IFIP–UNESCO–ITU

COMNET'85

SERVICES CONVEYED BY COMPUTER NETWORKS

Working Papers

»John von Neumann« Society for Computing Sciences
Budapest, Hungary

ITA/384

IFIP–UNESCO - ITU
International Symposium

COMNET'85

SERVICES CONVEYED BY COMPUTER NETWORKS

1–4. October 1985 Budapest

Organized by:

"John von Neumann" Society for Computing Sciences
with the co-operation of
Scientific Society for Measurement and Automation and
Scientific Society for Telecommunication

Sponsored by:

International Federation for Information Processing and
United Nations Educational, Scientific and Cultural Organization, International Telecommunication Union

Supported by:

Hungarian Academy of Sciences
State Office for Technical Development

Responsible publisher:
"John von Neumann" Society for Computing Sciences;
T. Szentiványi

Statistical Publishing House
Director: J. Kecskés

Printed in Budapest/Hungary

85-5262-09

PREFACE

Dealing with computer networks, the COMNET '85 conference is the third event in a series of meetings in this category. (The first one was held in 1977.) It is a pleasure for us to be host to this Conference which, although held in Central-Europe, will discuss problems arising in various parts of the world and draw participants representing almost every continent.

Regarding the development of networks the principal (technological and theoretical) operational questions are problems of the past, solved generally on the basis of international cooperation. In the present period two phenomena might be observed in connection with networks. One is embodied by the efforts to achieve a high degree of universality, in other words to have an unbound access to the different sources. The realization of the appropriate standards and protocols will ensure the necessary basis of this goal. The services provided by networks represent nowadays the other major field of interest. The questions are: How can networks efficiently combine individual and independent services rendered by single machines or even by groups of them? How can the range be widened, and what kind of difficulties arise in connection with these efforts? The goals of COMNET '85 are to present and discuss exactly these problems.

The year selected for COMNET '85 has a timely importance: It was 20 years ago that the thought of realization and practical usage of networks and the idea of packet-switching emerged for the first time. The early history of networks is marked by two names, Paul Baran of the USA and Donald W. Davies of the U.K., whose contributions are the most important in this respect. During the Conference a special session will be devoted to the commemoration of this anniversary.

I think a conference can only reach its goals if the necessary circumstances are ensured for participants to exchange views, furthermore to build and deepen personal contacts. Consequently, the Organizing and Program Committees laid great stress upon these in their activity. While the work of every Committee is highly esteemed, I wish to thank especially for their untiring efforts.

COMNET '85 could not have been efficient without the effectual support of sponsors of whom IFIP, the principal patron of this event and obviously the TC6, the international initiator within IFIP, have to be mentioned in the first place. The outstanding and generous support by UNESCO made it possible for specialists from a wide range of geographical locations to participate in COMNET '85. The allembracing contacts and experience of ITU helped us significantly in the preparation of the Conference. My special grateful thanks belong also to the John von Neumann Society which as main organizer of the Conference has provided a solid basis for the smooth running of the meeting.

T. Szentiványi

Hungarian TC6 delegate
General Chairman of COMNET '85

PUBLICATION COMMITTEE

Csaba,L.
Szentiványi,T.
Tarnay,K. /Mrs/

HUNGARIAN ORGANIZING COMMITTEE

Valter,F.	COMNET'85 Honorary Chairman
Szentiványi,T.	COMNET'85 General Chairman
Bakonyi,P.	
Balassi,I.	
Bottka,S.	
Budai,G.	
Csaba,L.	
Feczkó,I.	
Gergely,Cs.	Committee Chairman
Hetthéssy,G. /Mrs/	
Horniák,G.	
Horváth,P.	
Katona,J.	
Kovács,L.	
Lukács,J.	
Margitics,I.	
Mazgon,S.	
Németh,P.	
Nóbik,L.	
Parlagi,E.	
Tallóczy,I.	
Tarnay,K. /Mrs/	
Tóth,M. /Mrs/	Committee Secretary

INTERNATIONAL PROGRAM COMMITTEE

Bakonyi,P.	Committee Secretary, Hungary
Bazewicz,M.	Poland
Bochmann,G.V.	Canada
Boutmy,B.	the Netherlands
Boyanov,K.	Bulgaria
Csaba,L.	Publication Committee Chairman, Hungary
Danthine,A.	IFIP TC6 Chairman, Belgium
Davies,D.W.	UK
Gergely,Cs.	Hungary
Inose,H.	Japan
Königshofer,F.	IFIP Liaison Officer, Austria
Kulikowski,J.	Poland
Kutzkake,S.	UNESCO, Japan
Le Moli,G.	Italy
Maurer,H.	Austria
Meier,H.W.	GDR
Nakagome,Y.	Japan
Porizek,R.	Czechoslovakia
Pouzin,L.	France
Puzman,J.F.	Czechoslovakia
Samoylenko,S.I.	USSR
Spaniol,O.	FRG
Sunshine,C.	USA
Szentiványi,T.	COMNET'85 General Chairman, Hungary
Tarnay,K./Mrs/	Committee Chairperson, Hungary
Uhlig,R.P.	Canada
Vissers,Ch.A.	the Netherlands
Yakubaitis,E.A.	USSR

LIST OF REVIEWERS

Bakonyi,P.
Bochmann,G.V.
Bogomolov,L.P.
Boyanov,K.
Corc,R.
Csaba,L.
Danthine,A.
Ercsényi,A.
Ferenczi,P.
Foley,J.S.
Gabler,H.G.
Giessler,A.
Harangozó,J.
Hirst,I.
Horniák,G.
Horváth,P.
Johnston,R.B.
Koshelev,
Kovács,L.
Kubin,B.
Mahy,C.
Le Moli,G.
Löffler,H.
Margitics,I.
Meier,H.W.
Maurer,D.H.
Mazgon,S.
Nagy,Á.
Németh,G.
Paul,M.
Porizek,R.
Puzman,J.F.
Rayner,D.
Rubin,J.
Samoylenko,S.I.
Sarikaya,B.
Sugár,P.
Sunshine,C.
Szentiványi,T.
Tagányi,Gy.
Tarnay,K.
Tasker,R.
Telbisz,F.
Treijs,P.
Udalov,V.
Uhlig,R.P.
Urano,Y.
Vissers,Ch.A.
Yakubaitis,E.A.

TABLE OF CONTENTS

PLENARY PRESENTATIONS

The Videotex - Saga
H. MAURER - P-1

Architecture of Computer Networks for Large Offices
E.A.YAKUBAITIS - P-6

SESSION 1 - TYPE AND CHARACTER OF COMPUTER NETWORK PROVIDING SERVICES

The Management of Services on Computer Networks
E.B.SPRATT, D.J.CAUL, S.E.BINNS - 1-1

Computer Networks in the Academic Environment
M.BAZEWICZ, F.STUCHLIK - 1-8

The Design, Implementation and Information Services in Interuniversity Network (MSK) - Results and Experiences
M.BAZEWICZ - 1-21

User Characteristics of File and Job Services
R.KASZUBA, J.KWIATKOWSKI - 1-36

Strategy for Corporate Communication Networks within Philips
B.J.BOUTMY - 1-41

Architecture and Services of the Computer Network DELTA and Experiences with the System
E.JUNKER, H.W.MEIER - 1-50

How to Access DATA-STAR
L.J. VÁRSZEGI - 1-55

SESSION 2 - NON-TECHNICAL ASPECTS OF NETWORKING

Interworking Experiments with the Austrian and Hungarian Academic Networks
P.BAKONYI, L.CSABA, W.KUNFT, M.PAUL - 2-1

Impact of Networks on Database Users Education
M.R.MURASZKIEWICZ, Z.M.NOWICKI - 2-9

SESSION 3 - DATA COMMUNICATION ON PUBLIC DATA NETWORKS AND ISDN, TELEMATICS

Network Technology Independent Interface (NII)
K.SAMBOR - 3-1

Packet Switching in the Federal Republic of Germany (With a View to the Situation in CEPT-countries)
H.G.GABLER - 3-23

The ISDN Concept of the Deutsche Bundespost and the Integration of Non-voice-services
H.KUNZE - 3-37

Architectural Problems of ISDN
G.NÉMETH - 3-53

Winding Road to an ISDN
S.K.MAZGON - 3-61

Connection of Personal Computer Networks to ISDN
A.FILP - 3-72

Accessing Videotex through the ISO File Access Protocol
J.BARCHANSKI, G.V.BOCHMANN, P.DESJARDINS, J.GECSEI - 3-80

The BILDSCHIRMTEXT Service in the Federal Republic of Germany
H.KUNZE - 3-95

The Use of the Magazine-method in Interactive Videotex through Intelligent Videotex Decoders
I.SEBESTYÉN - 3-106

Accessing a Viewdatabase
L.KOVÁCS - 3-118

Implementation of the Hungarian Experimental Packet Switching Network
L.ALMÁSI, A.LÁBADI, P.HORVÁTH, A.KERTÉSZ - 3-128

SESSION 4 - DESIGN AND IMPLEMENTATION OF LOCAL AREA NETWORKS

The Development of an Ethernet Local Area Network for Scientific Data Acquisition
R.TASKER - 4-1

The Software Architecture of a Local Area Network
A.ARATÓ, I.SARKADI-NAGY, J.SULYÁN, F.TELBISZ - 4-9

LAN Simulation for Performance Prediction and Evaluation
K.H.KELLERMAYR - 4-23

The EXLOC Local Area Network
A.BALÁZS, M.GÁLL, R.JÁKFALVI, P.SUGÁR, J.SZENTNER - 4-32

Development of Distributed Applications for Experiment and Education Support Based on the NETEX Local Area Network
L.BORZEMSKI, A.GRZECH, A.KASPRZAK, L.KOSZALKA 4-38

A LAN Concept for a Physical Laboratory
P.E.GIESE, P.H.GIESE, A.V.ALFIMENKOV - 4-48

LOTUNET - A LAN for Process Control
H.LÖFFLER, D.GÜTTER, B.HACKLER, H.KÖNIG, Ch.KUBAS, G.NEIDHOLD, I.MEIER - 4-52

SESSION 5 - ARCHITECTURE OF COMPUTER NETWORKS

Communication Networks for Banking Service Delivery
D.A.SHEPPARD - 5-1

About Integration of Control and Commercial Information Systems
M.PAUL, W.KUNFT - 5-12

Mechanism Offering Multipoint Services on LAN-s Interconnected by Satellite
O.C.DUARTE, P.de LA MOTTE - 5-23

One Relational Model of Layered Network Architectures
R.PORIZEK - 5-35

Network Services vs. OSI/RM
J.F.PUZMAN - 5-43

An Implementation of a File Protocol for Computer Networks
Y.ATANASSOV, T.GOTCHEVA, T.STOINOV, J.GANOV 5-55

Management Service of Packet Switching Processors
R.PENZEL - 5-63

SESSION 6 - FORMAL DESCRIPTION METHODS

The OSI Transport Service and its Formal Description in LOTOS
G.SCOLLO, G.PAPPALARDO, L.LOGRIPPO, E.BRINKSMA - 6-1

SDL and ESTELLE: State Transition Oriented Specification Techniques in CCITT and ISO
M.STEINACKER - 6-25

Protocol Verification System Based on MODULA-2 Language
L.KOVÁCS, J.HARANGOZÓ - 6-41

VEDA: A Software Simulator for the Validation of Protocol Specifications
C.JARD, R.GROZ, J-F.MONIN - 6-50

Formal Description, Verification and Implementation of a Local Area Computer Network Transport Protocol
J.SZENTNER, L.KOVÁCS, J.HARANGOZÓ - 6-66

The PDL-System - an Unified Approach to the Specification, Verification and Implementation of Protocols
H.KÖNIG, M.HEINER - 6-87

PNPUO - A Petri-net Based Software Package for Protocol Validation
T.Y.CHEUNG, X.QIU - 6-103

Obtaining Normal Form Specifications for Protocols
G.V.BOCHMANN, B.SARIKAYA - 6-133

SESSION 7 - PROTOCOL TESTING EVALUATION AND MEASUREMENTS AND CONFORMANCE TESTING

Standardizing Conformance Testing for OSI
D.RAYNER - 7-1

The Use of LOTOS for the Specification and the Automatic Generation of Test Sequences
V.CARCHIOLO, S.PALAZZO, G.PAPPALARDO - 7-21

Description of Protocol Testing Scenarios
O.RAFIQ, J.HADDAD - 7-34

Testing Communication Protocols Using Random Legal Inputs
J.RUBIN - 7-50

Local Test for Distributed Testing
R.CASTANET, R.SIJELMASSI - 7-68

Performance Evaluation of ARQ Protocols in Ring Topology Networks
M.E.ANAGNOSTOU, S.A.BACOPOULO, E.N.PROTONATORIOS - 7-82

Performance of Unslotted Aloha with Partial ACK of Random Length Packets
J.WOZNIAK 7-92

Performance Measurements of the Communication Network Node
I.DUBILEWICZ, Z.FRYZLEWICZ, K.KOLESNIK, W.KOMOROWSKI - 7-102

On Measurements of the Packet Switching Network - Methods and Experiments
J.PEKAR - 7-118

SESSION 8 - SPECIAL TECHNICAL AND THEORETICAL PROBLEMS

An Analytical Model of a Local Network in Discrete Time
K.BOYANOV, P.RUSKOV, K.YANEV, B.DIMITROV - 8-1

OSI Communication Controller for Heterogeneous Computer Systems
A.HENNECKE, V.HEYMER - 8-10

Control Network Design Using MARK Microprocessor System
P.POPOV, I.JANOUSEK - 8-16

Problem of Inter-interference in a Two-hop Random Access Radio Network with BPSK Signals
A.T.KOZLOWSKI - 8-22

The Structure of the Reliable Control and Measurement System in the Computer Network
E.HUDYMA, E.KOSMULSKA-BOCHENEK - 8-29

Packet Communication on a T-User Collision Channel without Feedback via Cyclotomic Protocol Sequences
N.QUANG A., T.ZEISEL - 8-40

A Distributed Process Control Data Base Management System
G.JUHÁSZ, K.KOVÁCS, I.SÁRI - 8-54

Serviceability Analysis of Polish University Pilote Computer Network -MSK
M.M.BUDKA, J.J.GRZEGORCZYK - 8-67

Solution of the Problem of Standby for Data Circuits by Means of a Special Supplementary Hardware and/or Software
F.DLABOLA, J.PRIBYL, I.VAJDA 8-81

PLENARY PRESENTATIONS

THE VIDEOTEX-SAGA

H. Maurer

Institutes for Information Processing
Technical University
Graz, Austria

SUMMARY

The development of videotex reads like a tragedy or comedy, depending on one's point of view. Still, videotex is not dead, by no means, but still has the potential of developing into a major network for microcomputers. The author, who has been involved in videotex almost from the beginning, reviews the development of videotex and comments on its future.

VIDEOTEX DEVELOPMENT - A COMEDY OF ERRORS

Videotex originated in Great Britain in the early seventies ... as an offspring of unsuccessful attempts to develop a picture-telephone using ordinary phone-lines. The original idea of videotex (then still called viewdata) was simple and at least superficially appealing: take a television set with a remote-control unit as input device; add a little electrónics to it to convert it into a dumb (but colourful) terminal allowing the display of text and block-graphics ("mosaic graphic" in the lingo of videotex experts); connect this terminal via modem to a telephone and via the telephone to a computer (or a network of computers); provide on this computer-network a large variety of information by allowing arbitrary organisations to offer material for charge or free of charge; and what you get is an information system for everyone, inexpensive, easy to use and omnipresent. This was the theory behind videotex. And at a point in time when personal computers were not yet commonplace, colour monitors a rarity, and computer networks in their infancy the notion seemed to make some sense. However, reality was different: much of the information provided was advertismental and shallow, "gems" of information (which did exist) were difficult to find and, when looking for specific information of interest, chances were small to actually find it in the data-base. The incorporation of simple messaging and transactional facilities (albeit limited by the use of the only numeric remote-control key-pad) did not improve the situation significantly. Thus, when the first public videotex started as service with much fanfare in Great Britain in 1979 (under the trade-name Prestel), hopes were much too high: with a hodge-podge of some 100 000 screen-fulls ("frames") of essentially textual information who should be seriously interested in it? After all, a large general encylopaedia would already occupy 500 000 frames, the phone-books of Great Britain 300 000 frames, the major train connections within Europe 100 000 frames, etc., all this exemplifying the fact that fairly complete information on a fairly complete range of subject is quite out or reach of videotex technology, even today: a medium-scale library with 50 000 volumes corresponds to some 100 million

frames. This is some 100 times more than the capacity envisaged in any videotex-system to date! Such trivial calculations show that the notion that "videotex can provide up-to-date information for everything for everyone at any time" (I am quoting a popular early videotex-pamphlet) is nonsense, at least for some time to come. Hence it is a mute question whether at some stage videotex as a general information system can be a success (if so, we still have a long wait ahead of us). It can be a success, however, if it is used as an information system for certain kinds of information not otherwise readily available in up-to-date form, if it is used as network for communications, transactions and a variety of microcomputer-network applications.

It has taken a painful ten years for these facts to slowly being recognized. And even at the time of writing many videotex enthusiasts seem to be incredibly unaware of above basic points.

Let us briefly return, however, to the fascinating historic development of videotex. Although the British were not as successful at home with their Prestel-system as they had hoped for, they did manage to sell their system for trial installations to a variety of countries such as Germany, The Netherlands, Switzerland, Austria, Hongkong, Australia and others, just to name a few. It was recognized in many places that something was amiss with Prestel-videotex: after all, it did not develop as fast as was hoped for. The Canadians attributed the slow penetration rate to the poor quality of graphics, invented Telidon (a significant improvement as far as graphics go) ... but got so caught up with selling their graphics that they did not see the real problem: that videotex used only as a general information-retrieval system does not work. The French apparently recognized an other isolated but important point: that key-services are required to be successful, that the notion of selling someone videotex by offering a bit of everything is as ridiculous as a shop would be when offering "a package of one pound of bananas, fifteen assorted buttons and a few yards of barbed wire". The key service chosen in France was the electronic phone-book which is being pushed down the throats of hundred of thousands French and supposedly will form a good basis for other videotex services. In Germany, lack of interactivity was seen (correctly) as one other big problem with basic Prestel-videotex. Hence "external computers" linked via packet-switched networks were seen incorrectly as the only thing needed for a solution, allowing users a much enlarged spectrum of services ... overlooking the huge expenses involved, the slow responses to be expected and neglecting the question of better terminals.

In the meantime, discussions about presentation-level protocols (block-mosaic of Prestel versus geometric codings of Telidon) continued to escalate (as if the success of videotex would hinge only on the quality of graphics achieved!), leading to two different new developments: the NAPLPS-standard (pushed by AT+T with reluctant support of the Canadians in North-America) and the CEPT-standard of the Europeans. The CEPT-standard is complicated by the fact that it contains a basic (yet very sophisticated) level of block-mosaic graphics ("C0 level") and also contains higher ("C2 level") options allowing geoemtric graphics akin to the computer graphics standard GKS.

Despite the lip-service of all European countries to the CEPT standard only three countries have actually introduced CEPT, so far: Germany, Switzerland and Austria. And only Austria has introduced the CEPT standard including level C2 as obligatory. The reasons for Austria's choice are indeed noteworthy: it was recognized fairly early (in around 1979) that videotex and computer-technology would have to merge, if videotex had any chance to be widely accepted. For graphics this clearly implied the use of "geometric codings", i.e. GKS like-structures, as first tentatively introduced to videotex by the Canadians in their Telidon.

Indeed the author believes (maybe not quite objectively) that a number of crucial ideas which will help videotex to survive in the long run were recognized first in Austria: that the drop of the price of hardware would render the dumb-videotex-display terminals (be it using Prestel, Telidon, NAPLPS or CEPT standard) obsolete fairly rapidly, that the advance of home computers and personal computers would make videotex terminals look incredibly "dumb" and that, hence, the future of videotex would have to be based on the convergence of videotex terminals and personal computers into what one might call a videotex-computer: a videotex decoder of high quality which is a personal computer at the same time. Based on proposals by the author and supported by two federal ministries of the Austrian government the world's first videotex computer MUPID 1 was built in Austria, with first installations in the fall of 1982. The advent of MUPID 1 changed the Austrian videotex scene from the usual picture of sluggish growth into a situation where user demand for videotex is consistently higher than the installation capacity of the PTT: this has continued to be true throughout 1985 (and thousands of would-be videotex customers are still waiting to be hooked up at the time of writing).

The main point of MUPID 1 (i.e.of the notion of a videotex-computer) is that such a device not only is a videotex terminal and a personal computer but that the combination of the two functions provides totally new possibilities: first, software can be downloaded from the videotex network into MUPID, making videotex the potentially largest (and most easily updatable) library of computer programs ("telesoftware") ever to exist; second, using software, the navigation through the videotex data base can be much eased; third, software executed in MUPID can make use of data in the videotex-network and fourth, the whole communicative power of a general network can be used efficiently. The reader is referred for details to the literature Fellner (1984) and Maurer (1983a, 1984a, 1984b). In particular, the applications of videotex with MUPID to computer aided instruction Maurer (1983b) and societal important applications Maurer (1983c) all show the overriding importance of the use of videotex-computers such as MUPID as terminals. In short, "videotex needs PCs and PCs need videotex": dumb videotex-terminals do not offer enough possibilities; stand-alone micros collect dust in many cases: only the combination of micros and a network makes micros interesting: with electronic communication and a steady supply of new (tele)software.

That Austrian videotex has not made the big break-through which might be expected from above remarks is due to the painful transition from Prestel to CEPT standard, requiring a new system of videotex-terminals (MUPID 2 instead of MUPID 1). Indeed, the switch from Telidon to NAPLPS in North America and from Prestel to CEPT before having a solid base of videotex-users can be seen as a mistake,

obsoleting terminals, databases and protocols, annoying users and increasing the price of terminals at a time when prices for other PCs were dropping and dropping. The switch to CEPT in Europe was probably motivated by political considerations, introducing elements into videotex-graphics not readily available in ordinary personal computers (e.g. multi-phase flashing, colour-table flashing, 32 colours simultaneously, 480 x 240 pixel resolution, etc.), thus giving European industry a chance to produce suitable terminals without competing directly with all other personal computers on the market. From this point of view and considering the high quality of graphics obtained by CEPT (if the geometric level is included) the introduction of the CEPT standard may indeed prove beneficial in the long run.

A BRIEF STATUS REPORT ... AND A LOOK INTO THE FUTURE

As has been predicted already in 1981 by the author, Maurer (1981), videotex experiments based only on the notion of information retrieval have turned into failures, by and large. Exceptions are due to finding particularly useful niches (e.g. Grassroots for farmers in Canada) and due to particularly strong government support keeping things alive (e.g. Teletel in France with hundreds of thousands free Minitel-terminals being provided to customers, or Bildschirmtext in Germany, with the PTT not levying any frame charges and investing rather large sums of money).

Other systems which do better have switched from systems providing merely "ordinary" (textual or graphical) information to networks allowing broad electronic communication and the delivery of telesoftware. The Prestel system in Great Britain is a case-in-point: towards the fall of 83, with 20 000 users and deep in the red, British Telcom seriously considered abandoning (and closing) Prestel. As last ditch-effort Micronet 800 - a distribution center for telesoftware - was incorporated into Prestel: the rate of subscription went up (and continues to go up) significantly because of this move: today, with over 60 000 users forming a solid base, Prestel has stopped to loose money and is slowly starting to look like a viable system. Telesoftware has transformed the Dutch videotex-scene similarly in a positive fashion. Due to the fact that the Prestel standard is continued to be used in both Great Britain and The Netherlands many home-computers can be hooked up, forming a wide heterogeneous base of users, heterogeneous particularly concerning telesoftware (many pieces of telesoftware are available in a variety of home computers used as terminals).

The effect of intelligent terminals (= videotex computers) compared to dumb terminals (= display only terminals) can bee seen most dramatically when looking at Austria vis-a-vis Germany. Based on the countries'size and technological development, Germany should have 15 times more terminals than Austria, all else considered equal. Thus, the 3500 terminals installed in Austria by May 85 should correspond to some 55 000 terminals in Germany ... but only 27 000 terminals were in place, despite the fact that videotex is subsidized and promoted much more in Germany than in Austria...

All counted it is clear by now that videotex will only be a success if seen as a network of home - and personal computers, hopefully satisfying a reasonably high graphic standard (CEPT C2).

developments are supporting such direction (e.g. by replacing the ridiculously low transmission rates of 1200/75 Baud by 2400/2400 Baud) videotex will be a big success, both in the commercial and private arena. Ortherwise, videotex will fade away and will eventually be replaced by other micro-computer networks.

The question is not whether countries like Germany will have ten million networked colour graphic micros by 2000 or not, the question is only whether the majority will be hooked up to a videotex-network, or to a variety of other non-videotex networks.

REFERENCES

Fellner (1984): Fellner, W.D., Maurer, H.A. (Editors): MUPID 2 - eine Übersicht; Report B 47, IIG Graz (1984).

Maurer (1981): Maurer, H.A.: Bildschirmtextähnliche Systeme, Report B 11, IIG Graz (1981).

Maurer (1983a): Maurer, H.A.: Bildschirmtext - Omnipräsente Copmputerterminals; Elektronische Rechenanlagen 25 (1983), 177-183.

Maurer (1983b): Maurer, H.A.: New developments in videotex and their implications for computer aided instruction; Information Services and Use 3 (1983), 319-324.

Maurer (1983c): Maurer, H.A.: On some beneficial implications of videotex on society; Computer Compacts 1 (1983)

Maurer (1984a): Maurer, H.A.: The Austrian Approach to Videotex; Cybernetics and Systems Research 2, North Holland (1984).

Maurer (1984b): Maurer, H.A., Posch, R.: MUPID 2 - Durchbruch für Bildschirmtext; Report B 43, IIG Graz (1984).

ARCHITECTURE OF COMPUTER NETWORKS FOR LARGE OFFICES

E.A.Yakubaitis

A modern large office is characterized by a considerable base of computers with different productivity. These computers are entrusted the execution of a wide spectrum of information and computational tasks.

It is natural that in order to ensure the efficient work of the office, the computers must be capable of dynamically exchanging the information to provide the staff with the ability of operating with all the information and computing resources available. This requirement is fulfilled through the use of computer network whose architecture is considered below.

First we will assume that the computer network satisfies the requirements of Standard 7498 worked out by the International Standardization Organization /I/ which specifies the Basic Reference Model for Open Systems Interconnection.

According to Standard 7498, a system is a set of one or more computers, the associated software, peripherals, terminals, information transfer means and human operators. An application process is a set of procedures dealing with the processing of information. In accordance with these definitions, a subscriber system is a system which executes user application processes.

The logical structure of the subscriber system is shown in Fig. I. It follows from Standard 7498 that the system must have seven (I-7) layers of protocols. Above these protocols the set of application processes needed by the users is located (Fig. I shows the processes I, II, III). In conformity with the application processes, different services are introduced into the system which are determined by the application (7) and presentation (6) layers of protocols. They describe the procedures concerned with the creation of data banks, complexes of computational tasks, electronic mail, etc. The protocols of five (I-5) lower layers are combined into a single complex called a network access method.

The general structure of the computer network is shown in

Fig. 2. As the picture shows, the network consists of two types of basic components: the subscriber systems (rectangles I-7) and the communication subnetwork. The communication subnetwork is an aggregate of physical medium, hardware and, possibly, software that provides for the interaction of subscriber systems to one another.

The routing is a procedure of determination by the communication subnetwork of the paths of data transfer to their addresses. The selection is a procedure of selecting from a multiplicity of data units transmitted to the subscriber system those ones which are addressed to it. In this context, depending on the methods of data unit delivery, we identify /2/ three types of computer networks: networks with the routing of information, networks with the selection of information and mixed networks.

The type of the communication network employed determines also the type of the network on the whole. Thus, the network with the routing of information (Fig. 3), as its name implies, must have one or more components that ensure the process of routing. These components are called the communication systems or the packet switching nodes. In addition to the routing, each communication system executes also a number of other auxiliary functions related to this process. Corresponding to the number of communication systems, we identify two types of networks with the routing of information: one-node networks and multinode networks.

The logical structure of the communication system is shown in Fig. 4. The number of groups of protocols of the network (3), link (2) and physical (I) layers executed by the system is equal to the number of data transmission channels connected to this system. Fig. 4 shows four such groups. Above the network layer the network process is located which ensures the routing of information and executes a number of other service functions.

An example of the structure of one-node communication subnetwork is shown in Fig. 5a. Figure 5b shows the structure of the multinode subnetwork which includes four packet switching nodes. An example of the one-node communication subnetwork is the network developed by Northern Telecom /3/.

The boundaries of the communication subnetwork are determined by its interface points. The most widely used subnetwork interface is the standard known as the X.25 Recommendation /4/ of the International Consultative Committee on Telegraphy and Telephony.

A network with the selection of information is such one in which the delivery of data units from one application process to the other is provided through the execution of selection procedures. In such networks each subscriber system receives all data units transmitted to the multiplicity of subscriber systems. The system selects data units addressed to it and discards the rest of them.

The structure of the three types of communication subnetworks with the selection of information is shown in Fig. 6. A monochannel subnetwork is a subnetwork in which all information is transmitted over a single multipoint channel - a monochannel. The latter (Fig. 6a) consists of the physical medium and the medium access units (AU). The physical medium is provided by twisted pairs, coaxial cable or lightguide. The monochannel can transmit one data unit at a time only. The most typical example of this type of communication subnetwork is the monochannel of the Ethernet local network.

Along with the monochannel, the computer network can also use a polychannel /2/. The latter has the same structure (Fig. 6a as the monochannel. However, unlike the monochannel, it is frequency-divided into a number of subchannels. Each of the frequency subchannels together with the access unit connected to it forms an individual communication subnetwork. Thus, the polychannel, in contradistinction to the monochannel, enables the construction of a group of communication subnetworks rather than a single subnetwork. An example of this type of communication subnetwork is the polychannel created in the Wangnet /6/ local network.

The cyclic ring (Fig. 5b) forms the ring communication subnetwork. In this subnetwork /2/ the information circulates past all the access units thus providing for the subscriber systems connected to these units the ability of selecting data units on

the basis of their addresses. An example of the ring communication subnetwork is the Cambridge Ring /7/.

A mixed subnetwork is such one in which several different types of communication subnetworks are used so that one of the subnetworks performs the routing, while the other - the selection of information. Accordingly, in different parts of the mixed subnetwork different methods of transmission of data units to their destinations are employed.

We should note that the types of subnetworks considered (Fig. 3) can have different topology. Thus, the monochannel-based subnetwork can have the star, tree, ring or mesh topology. Therefore the topology of the subnetwork does not determine its class or type. For example, the class of ring subnetworks can include multinode, monochannel-based and cyclic subnetworks. Each of the communication subnetworks considered has standard interface points. At these points the subscriber systems are connected to the subnetwork.

The topology of computer location in a large office is determined by economic considerations and by the location of the points in which the information is sourced and consumed. An example of computer location in a large office is shown in Fig. 7. Computer locations are shown here by dark rectangles, Most of the computers in question are combined into compact groups (I, II, III). In each group the minimal distances between the computers (d_1, d_2, d_3) are rather small (1,5-2 km). However, the distances between the groups (l_1, l_2, l_3) are considerable and can amount to thousands of kilometers. In the office there will often be also the computers that (Fig. 7) do not fall within the compact groups.

For small distances very convenient are the communication subnetworks with the selection of information (monochannel, polychannel or cyclic ring). However, for long distances these subnetworks have proved inappropriate. In this case the subnetworks with the routing of information are used (one-node or multinode). These subnetworks can also be used for connecting closely situated computers.

Accordingly, three structural types of the networks for large offices are possible: one-node, multinode and mixed.

The one-node network has one switching node only (Fig. 5a) and therefore is the simplest one. Its node is installed in one of the buildings of the office and to this node all the computers are connected regardless of their location.

A more reliable and often economically practical is a multinode network. Here, if the office is spread over q geographic locations, in most cases the network has q small packet switching nodes (Fig. 5b).

Most effective is a mixed computer network. It is built on the basis of several communication subnetworks shown in Figs. 5 and 6.

Figure 8 shows a computer network based on three communication subnetworks. The first and third subnetworks are multichannel networks while the second one is the multinode network. We should note that in the network shown in Fig. 8 the packet switching node must be able to operate with three types of channels: the subscriber, the group and the internode channels.

The subscriber channel connects a single subscriber system to the node, while the group channel connects a number of subscriber systems. The internode channel links two switching nodes. The subnetworks are teamed into a unified communication subnetwork with the help of the interface systems. These systems provide here the necessary conversions of the network, link and physical protocols used in the subscriber systems connected to different subnetworks.

The logical structure of the interface system is shown in Fig. 9. It consists of three parts. The first part determines the network (3^{I}), link (2^{I}) and physical (1^{I}) layers that face the section I. The second part describes similar layers (3^{II}, 2^{II}, 1^{II}) directed toward the section II. The third part of the interface system is called a network process and couples the first two parts of this system. Here the network process ensures the execution of conversions needed for providing interaction between the two sections of the communication subnetwork.

We should note that the mixed computer network can be created in a step-by-step fashion. At the initial stage a multinode network can be created while at the succeeding stages the parts built on the basis of monochannel, polychannel or cyclic ring are added.

The topology of computer location in a large office (fig. 7) calls for a splitting of the mixed computer network into the standard sections. The structure of such a section is shown in Fig. IO.

The kernel of the section is the monochannel (Fig. 6a) to which the subscriber systems and the packet switching node are connected. The subscriber system consists of a host computer and a transport station (TS). The host computer provides or consumes information/computing resources of the network. Therefore we will call it a subscriber. Each network section can have n computers located at a small distance from one another. These computers and the packet switching node are connected through the A type transport stations (TS-A) directly to the monochannel. The group of m computers located at a considerable distance has the type B transport stations (TS-B) and connects to the monochannel through the packet switching node. It is through this node that the subscribers of the network section in question interact with the subscribers of other sections.

The mixed experimental computer network (ECN) was developed at the Institute of Electronics and Computer Science of the Latvian SSR Academy of Sciences; it consists of the sections shown in Fig. IO. The network is designed primarily for connecting the ES series mainframes (I022, I033, I035, I045, I060...) and the mini/microcomputers (SM-I300.0I, SM-I800, SM-4, SM-I420, ISKRA-226). However, it may incorporate also the computers of other types.

In the ECN the basic information and computing resources are provided by the ES computers. The logical structure of the subscriber system based on a ES computer is shown in Fig. II. It includes two access methods: a network access method and a telecommunications access method. The network access method ensures the interaction of application processes in the network (Fig. I). The telecommunications access method is used for controlling the multidisplay terminal complex ES-7920.

The terminal process service ensures communication of the ES-7920 terminals with all the ES computer-based subscriber systems connected to the computer network. This service allows:

- to establish a session with the subscriber system selected;
- to select a requisite service in the subscriber system;
- to carry out the work with the service selected.

The network job execution service distributes computational jobs to be executed in the network on the allocated set of ES computer systems. The basic functions performed by this service allow:

- to provide the users with the services of job preparation and execution on the ES computers distributed in the network;
- to carry out automatic control of job and result flows on the basis of information about the status of subscriber systems;
- to inform the user about the passage of its jobs through the network;
- to provide the collection of information about the functioning of the service;
- to store user files.

The electronic mail service encompasses the interacting mailboxes for the incoming and outgoing mail which are provided on all the subscriber systems. The electronic mail user can do the following:

- to prepare and edit a letter;
- to send a letter by putting it in the box for outgoing mail;
- to receive a letter from the box for incoming mail;
- to send a letter and to receive the acknowledgement of its delivery.

The three services described are the new ones and are not included into the ES computer software. However, at the same time the computer network provides the KROS, SRV and KAMA services which are standard for the ES computers. The KROS service controls the execution of computational jobs in a single ES-computer-based subscriber system and provides the following:

- optimal scheduling of the queues of jobs submitted to the system;
- automation of certain functions of the system operator.

The interactive time sharing service (SRV) provides the following functions:

- access to the resources of the subscriber system through the use of the instruction language;
- input, editing and retrieval of data;
- debugging and execution of user programs in dialog mode;
- dynamic control of systems resources.

In the SRV environment the UNIBAD information subservice operates. It provides for the following:

- creation of various data bases;
- unification and separation of data bases;
- search of information in data bases.

The information retrieval service KAMA+POISK executes the following tasks:

- creation of document files;
- addition of documents to the existing files;
- removal of documents from the files;
- retrieval of documents in the files.

The subscriber system of the ECN network (Fig. I2) consists of three types of basic complexes: the subscriber and the two types (A,B) of stations. At the first phase of the network project the subscriber is a ES computer or a minicomputer. The transport station is built on the basis of 'Elektronika' microcomputer. The application processes and the higher protocol layers (5-7) are implemented by the subscribers. The lower protocol layers (I-4) are implemented by the transport station.

The transport station of the ECN network is designed to enable direct connection of the subscriber to the monochannel or the packet switching node. Its structure is shown in Fig. I3. To the common station bus operating with the speed about 25 megabits per second are connected the 'Elektronika' microcomputer with its 64 kilobyte main storage, the 64 kilobyte station buffer memory and a group of controllers that provide for the connection of the station to the ES computer channel, minicomputer bus, monochannel or X.25 packet switching node. All the controllers have a direct access to the station buffer memory. The latter can be expanded up to I megabyte. The type A transport station does not have the X.25 (LAP B) controller, while the type B transport station does not have the monochannel controller. The general view of the transport station is shown in Fig. I4.

At layers I and 2 of the class A transport station (Fig. I2) the physical protocol, the monochannel access control protocol and the logical linkage control protocol are implemented. All these protocols comply with the IEEE-802 standard. The access to the monochannel is governed by the carrier sense and collision detection procedures. This monitoring is performed without organizing connections at the link layer. At the first stage of operation the network protocol practically is absent so that the services of layers I and 2 are made available to layer 4. At layer 4 the ISO transport protocol of class 4 is used /8/.

In the type B transport station (Fig. I2) the lower three layers (I-3) implement the X.25 Recommendation. At layer 4 the transport protocol is executed which is similar to that executed in the type A transport station. This allows the both types of stations to have the identical subscriber interface so that the subscriber is not aware of what type of transport station he has and how he is connected to the monochannel (directly or through the packet switching node).

The session protocol implemented by the subscriber (Fig. I2) corresponds to the ISO recommendations. The presentation and application protocols of the subscriber specify the network services shown in Fig. II.

The packet switching node has a small size since it is built on the basis of the SM-I300.0I microcomputer. The node has 256 kilobyte RAM memory.and switches sixteen data channels. The monochannel is based on the coaxial cable to which the access units are attached. The photograph of the access unit developed is shown in Fig. I5.

Currently the Experimental Computer Network of the Latvian SSR Academy of Sciences, as it is in October I984, encompasses I9 subscriber systems based on the ES, SM-4, Elektronika I00/25, Iskra-226 and SM-I800 computers. The network offers the following services:

- organization of information services;
- creation of data bases and the retrieval of requisite data in them;
- execution of computational tasks;
- electronic mail;

- local services (editing of texts, preparation of jobs, maintenance of job and result libraries, etc.);
- storage of texts in the network filestore;
- issuance of information about network operation and resources;
- storage of results until called for;
- development of new application processes in dialog mode.

The architecture, software and hardware available allow to create efficient computer networks for large offices. These networks enable the execution of practically all the activities needed by the offices which involve storage, processing and transmission of information.

References

1. ISO. 'Basic reference model for Open Systems Interconnection'. ISO 7498, 1983.
2. E.A.Yakubaitis. Information and computer networks. Statistika Publishers, Moscow, 1984.
3. Edward M. Deregulation heralds new era for digital PBX. Communication News, Dec. 1982, pp.32-36.
4. CCITT Yellow Book. Volume VIII.2. Data Communications Networks Services and Facilities. Terminal Equipment and Interfaces. CCITT VII th Plenary Assembly, Geneva, 1980, pp.83-93 (September 1981).
5. The Ethernet - A Local Area Network. Digital Equipment Corporation; Intel Corporation; Xerox Corporation. 1980.
6. Wangnet. Wang Laboratories Inc., 1982, pp.1-8.
7. Polynet. Network Manual. Logica VTS Limited, 1981.
8. ISO/DP 8073. Information Processing Systems - Open Systems Interconnection - Connection-oriented Transport Protocol Specification. April, 1983.

Abstract

The paper addresses the problems of development of computer networks for large offices. These offices are characterized by the use of a considerable number of computers grouped in different geographic locations, such as divisions, branch offices or bureaus. In this context the most efficient architectures of the networks created in large offices are analyzed.

List of illustrations

Fig. I. Logical structure of the subscriber system.
Fig. 2. General structure of computer network.
Fig. 3. Classification of computer networks.
Fig. 4. Logical structure of the communication system (packet switching node).
Fig. 5. Structure of the one-node (a) and the multinode (b) communication subnetwork.
Fig. 6. Structure of the monochannel (a), the polychannel (a) and the cyclic (b) communication subnetwork.
Fig. 7. Topology of computer location in a large office.
Fig. 8. Mixed communication subnetwork consisting of three sections.
Fig. 9. Logical structure of the interface system.
Fig. I0. A section of the computer network.
Fig. II. Logical structure of the subscriber system in the ECN network.
Fig. I2. Diagram of implementation of the subscriber system in the ECN network.
Fig. I3. Structure of the transport station in the ECN network.
Fig. I4. General view of the transport station.
Fig. I5. Coaxial cable access unit.

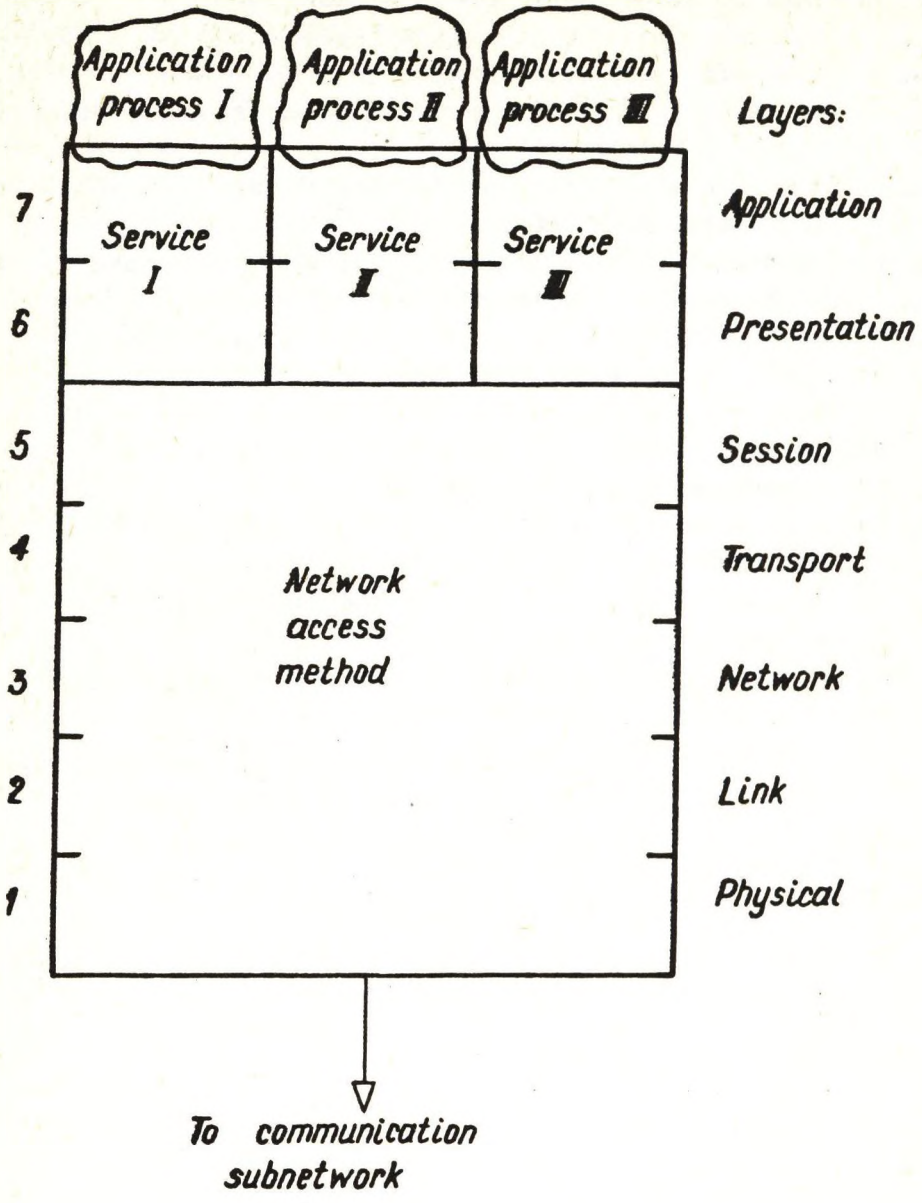

Application process I
Application process II
Application process III
Layers:
7
Application
Service I
Service II
Service III
6
Presentation
5
Session
4
Transport
Network access method
3
Network
2
Link
1
Physical
To communication subnetwork

Fig. 1

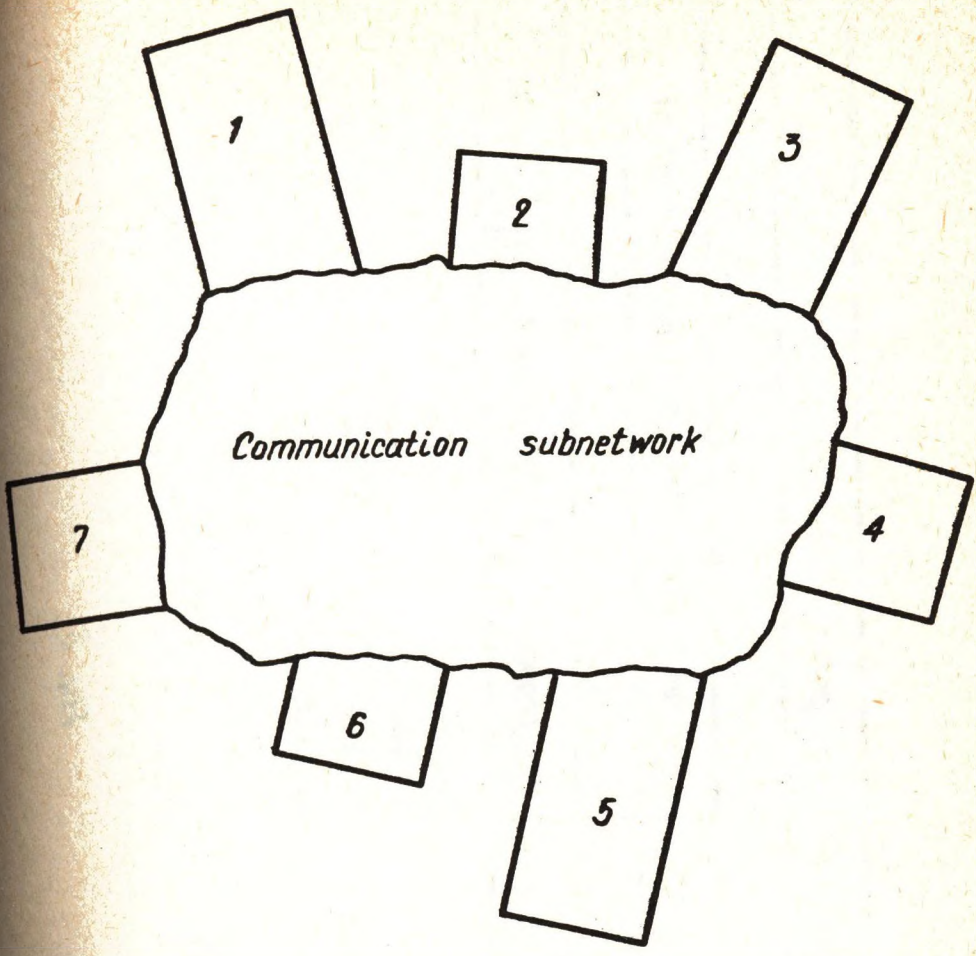


Fig. 2

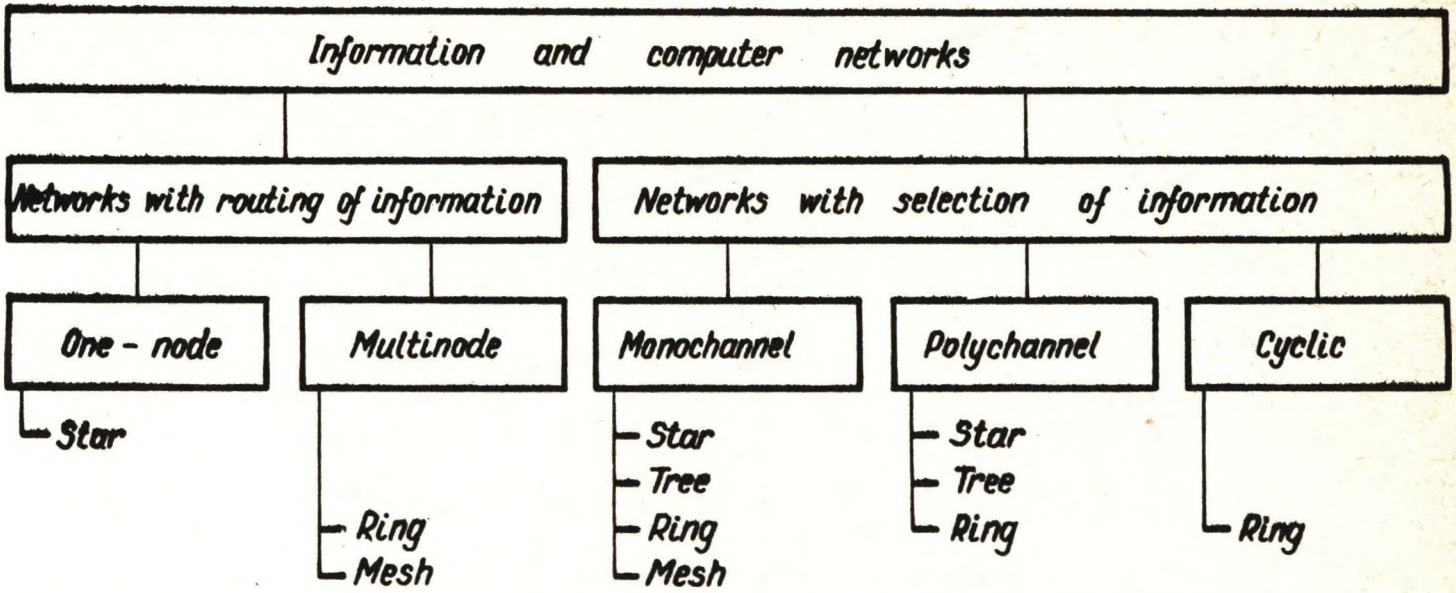

Information and computer networks
Networks with routing of information
Networks with selection of information
One - node
Multinode
Monochannel
Polychannel
Cyclic
Star
Ring
Mesh
Star
Tree
Ring
Mesh
Star
Tree
Ring
Ring

Fig. 3

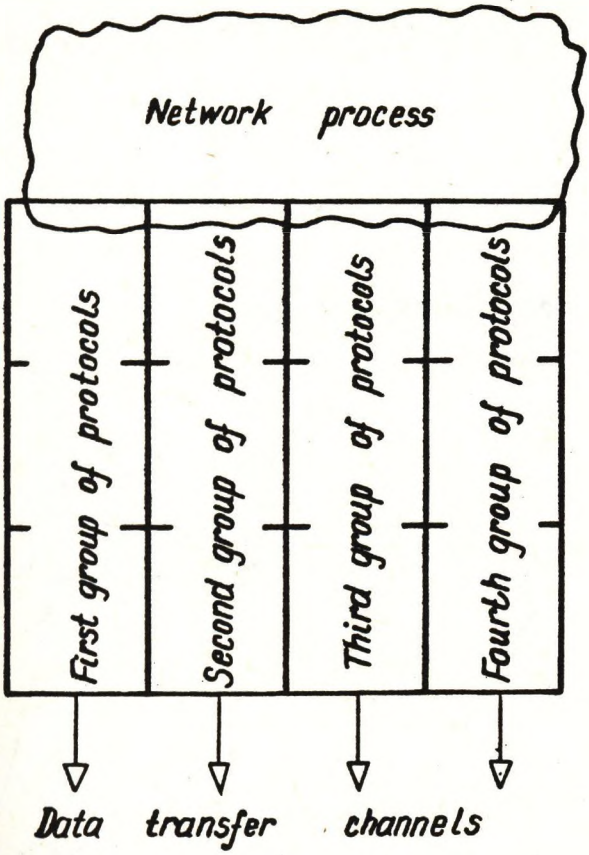
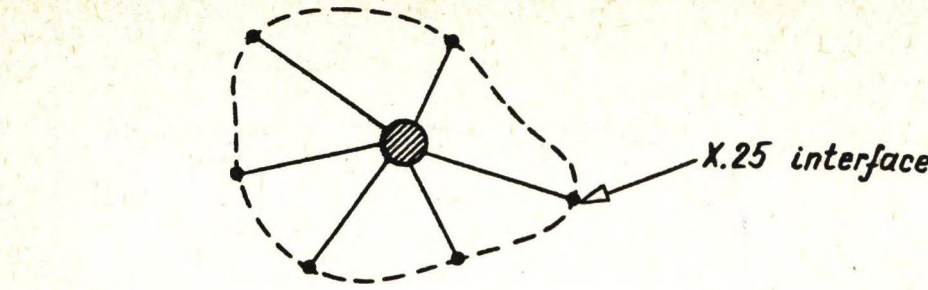
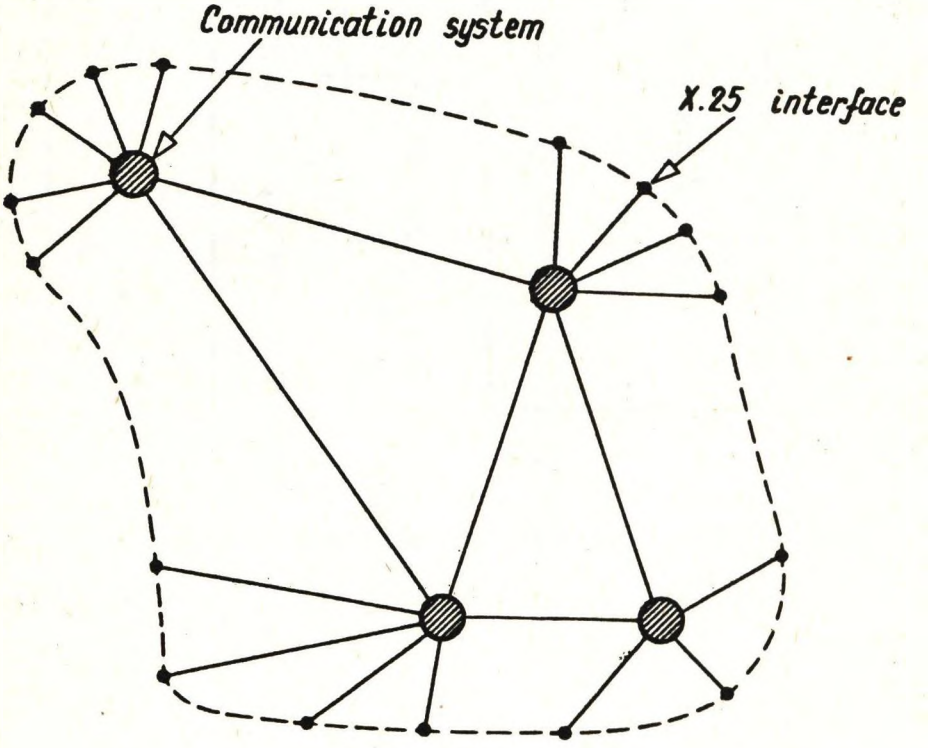

Network process
First group of protocols
Second group of protocols
Third group of protocols
Fourth group of protocols
Data transfer channels

Fig. 4

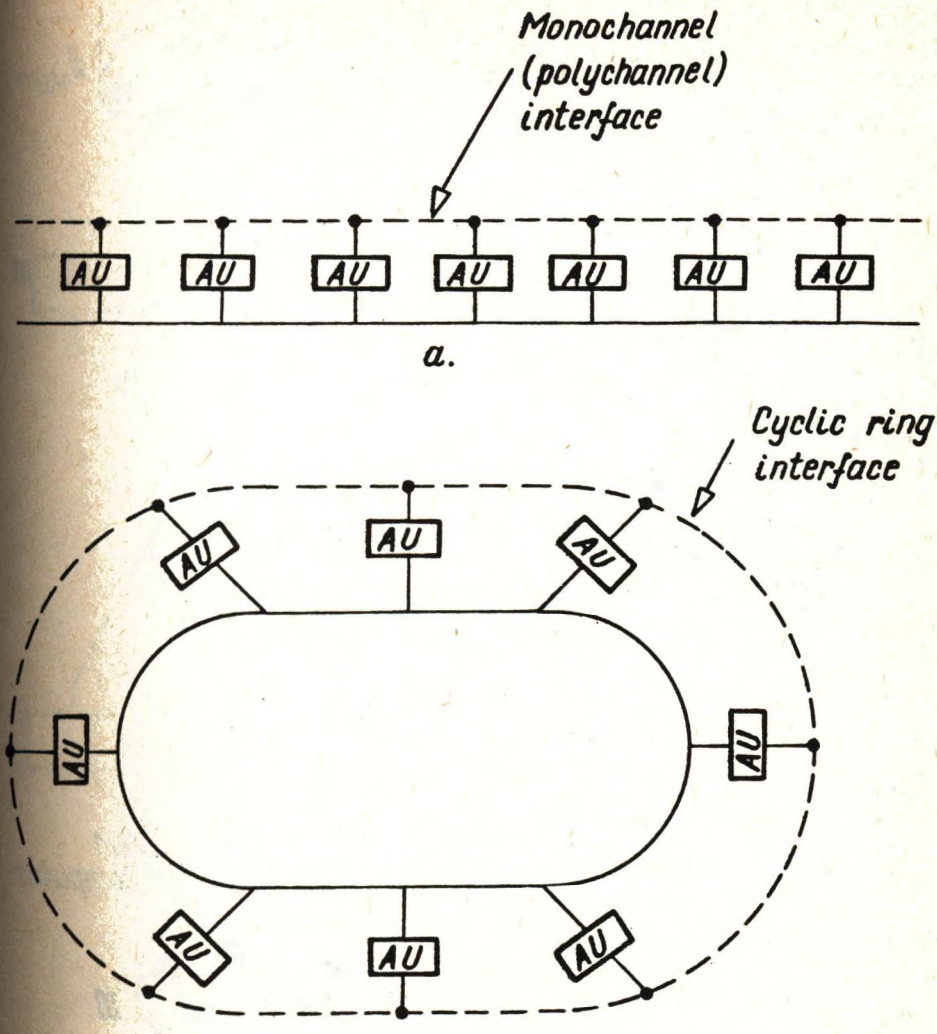


a.



b.

Fig. 5


Monochannel
(polychannel)
interface
AU
AU
AU
AU
AU
AU
AU
a.
Cyclic ring
interface
AU
AU
AU
AU
AU
AU
AU

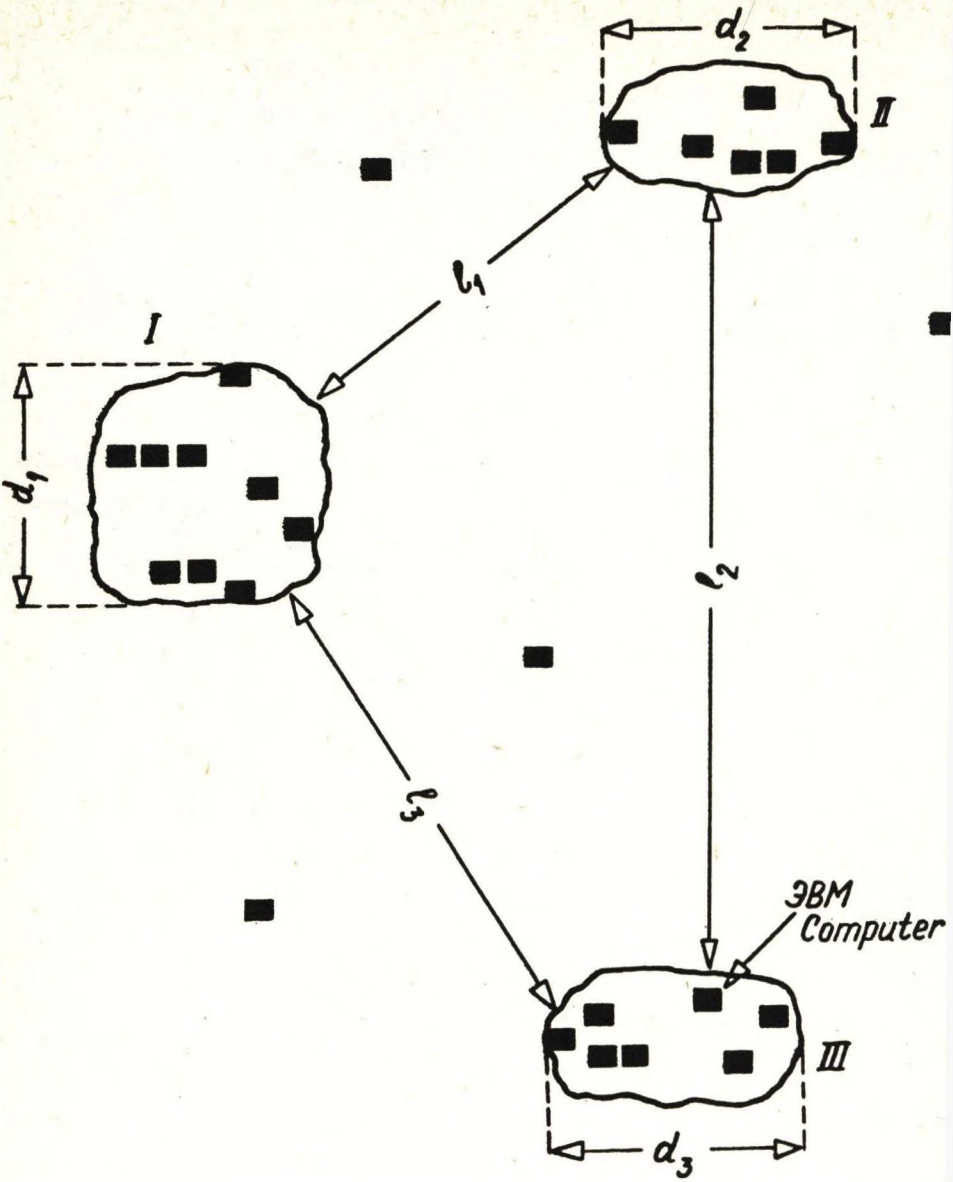


Fig. 7

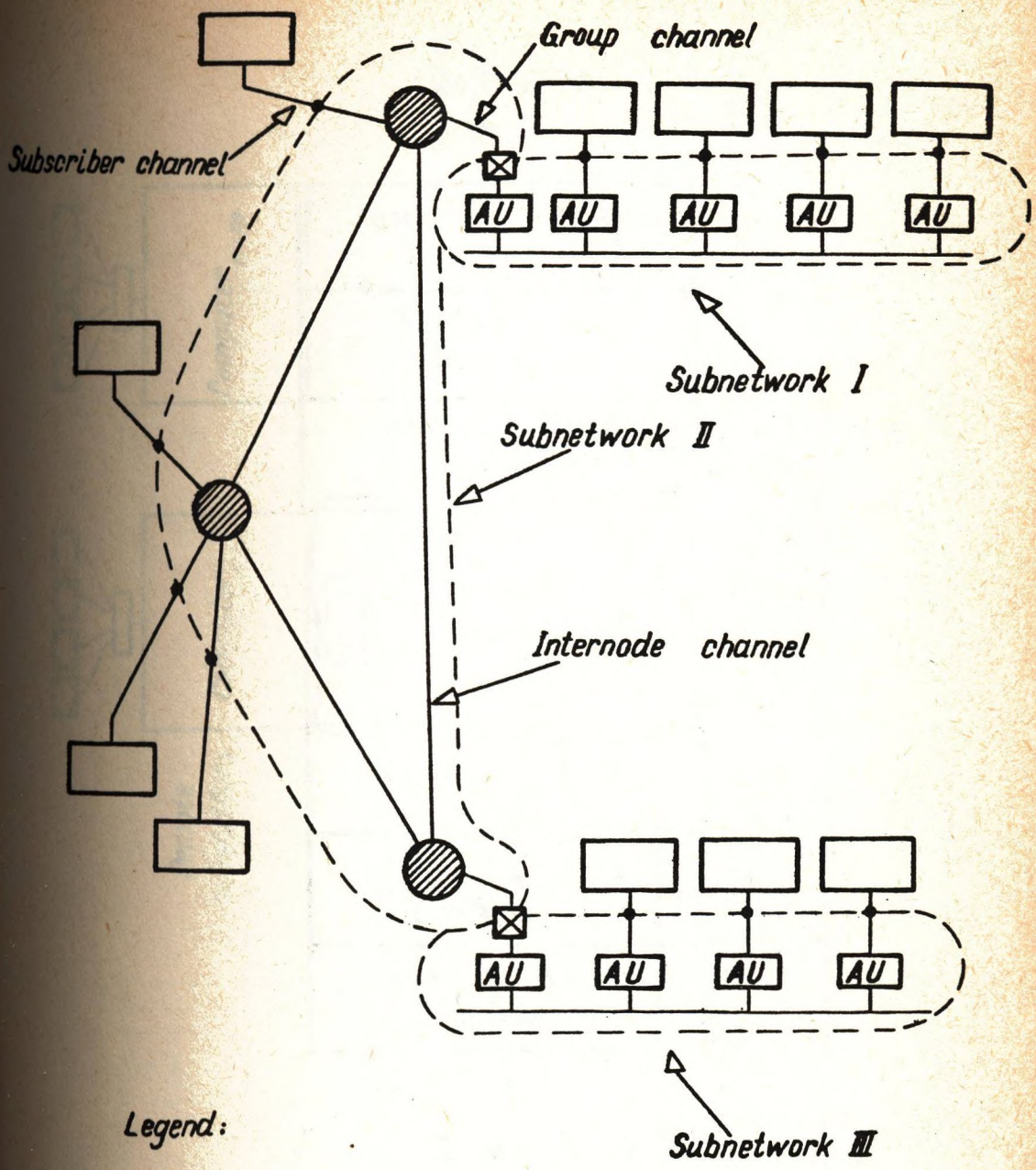

Group channel
Subscriber channel
AU
AU
AU
AU
AU
Subnetwork I
Subnetwork II
Internode channel
AU
AU
AU
AU
Subnetwork III
Legend:
- subscriber system,
- interface system.

Fig. 8

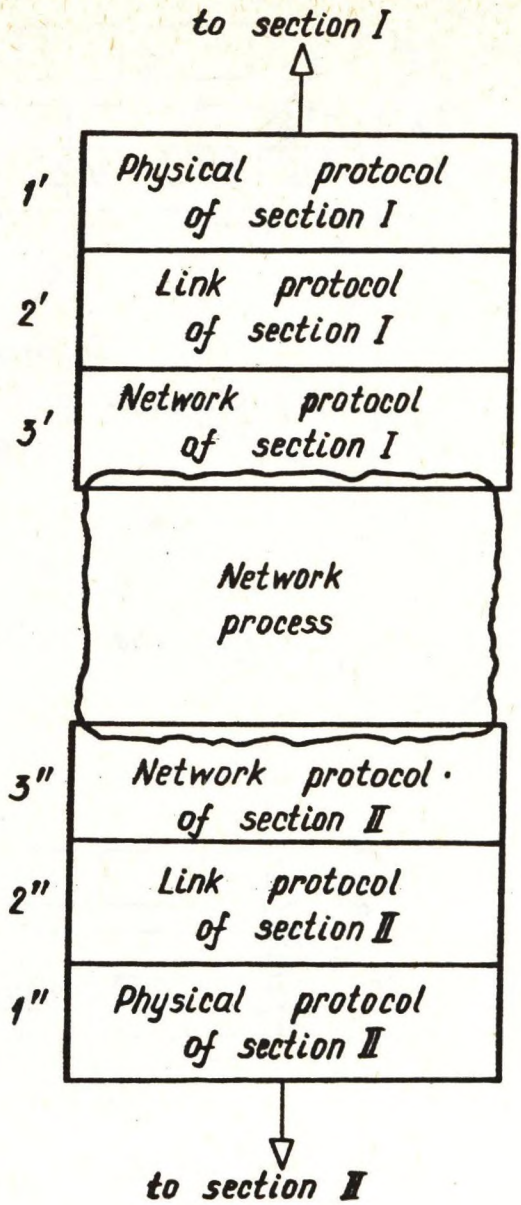

to section I
1'
Physical protocol of section I
2'
Link protocol of section I
3'
Network protocol of section I
Network process
3"
Network protocol of section II
2"
Link protocol of section II
1"
Physical protocol of section II
to section II

Fig. 9

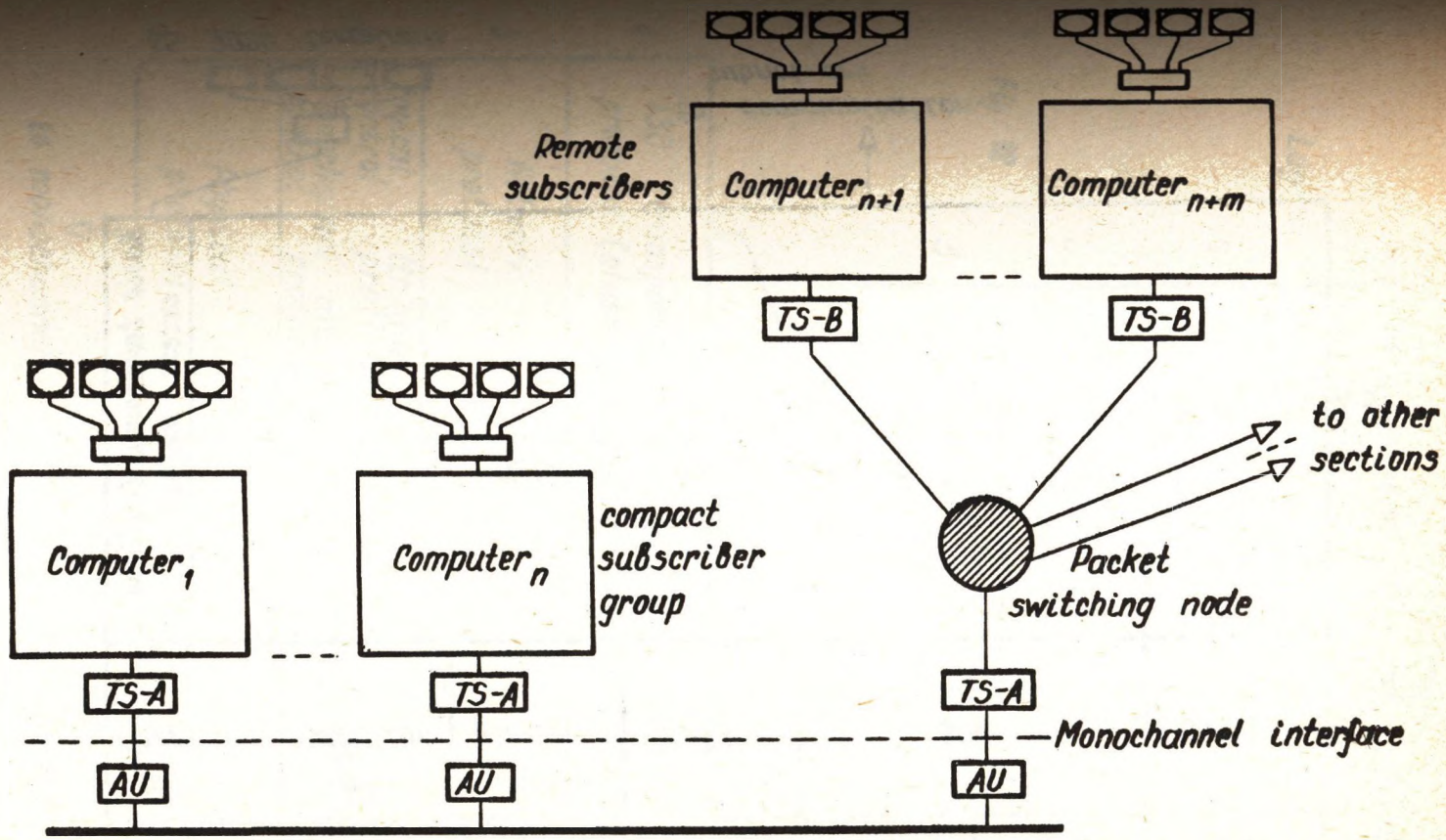
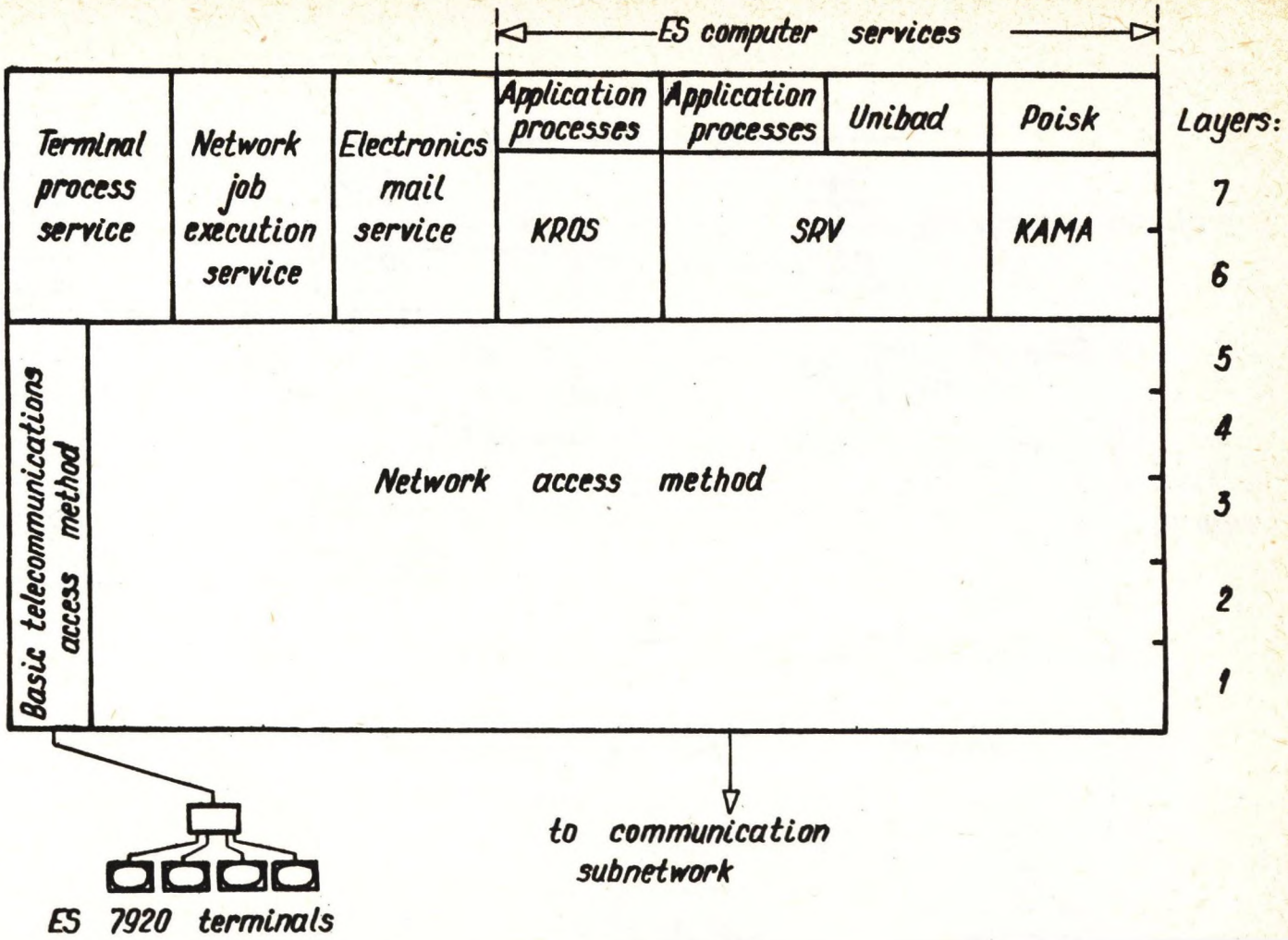

Remote subscribers
Computer n+1
Computer n+m
TS-B
TS-B
to other sections
Computer 1
Computer n
compact subscriber group
Packet switching node
TS-A
TS-A
TS-A
Monochannel interface
AU
AU
AU

Fig. 10


ES computer services
Terminal process service
Network job execution service
Electronics mail service
Application processes
Application processes
Unibad
Poisk
KROS
SRV
KAMA
Layers:
7
6
5
4
3
2
1
Basic telecommunications access method
Network access method
to communication subnetwork
ES 7920 terminals

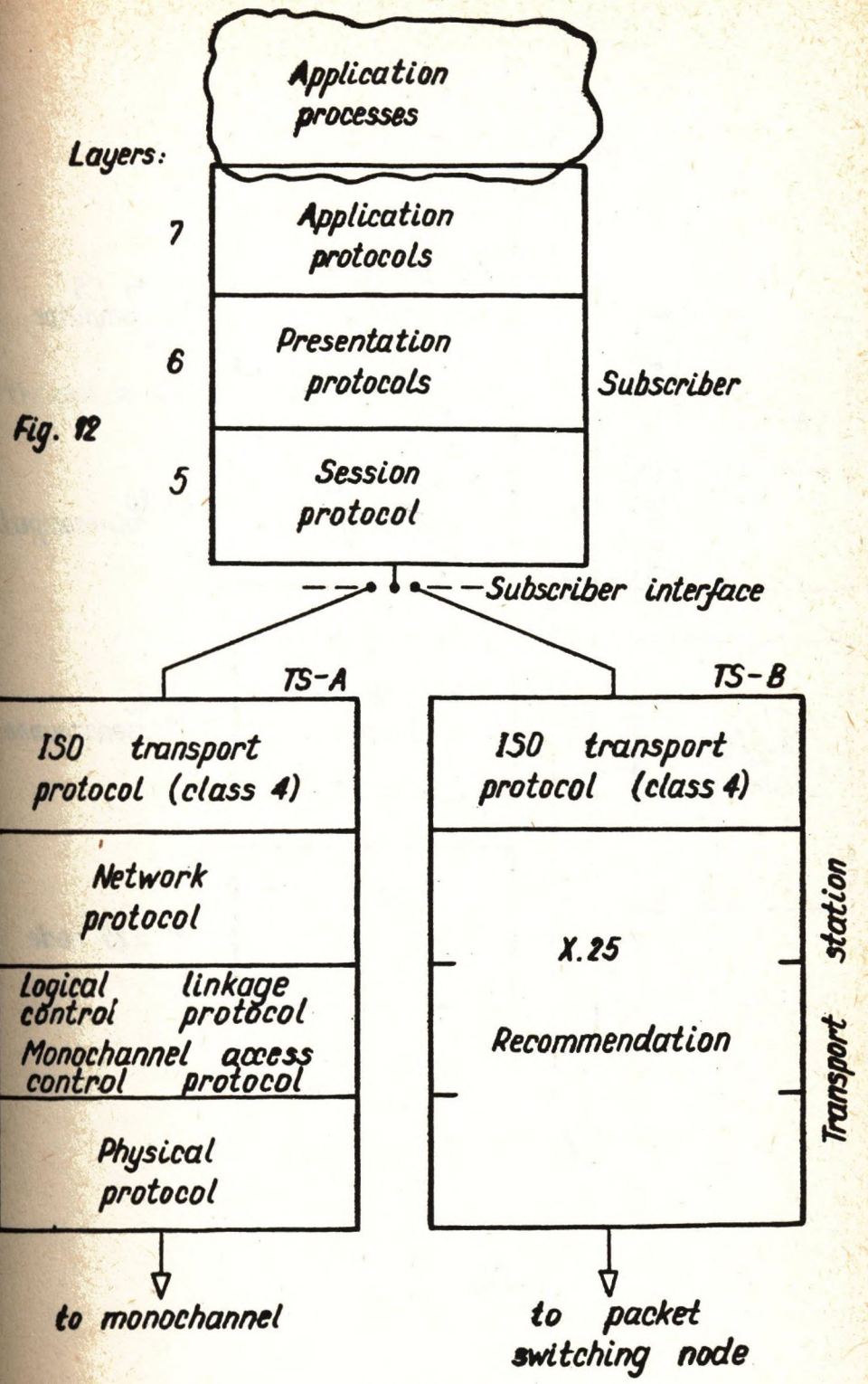

Application processes
Layers:
7
Application protocols
6
Presentation protocols
Subscriber
5
Session protocol
Subscriber interface
TS-A
TS-B
4
ISO transport protocol (class 4)
ISO transport protocol (class 4)
3
Network protocol
X.25
Recommendation
Transport station
2
logical control linkage protocol
Monochannel access control protocol
1
Physical protocol
to monochannel
to packet switching node

Fig. 12

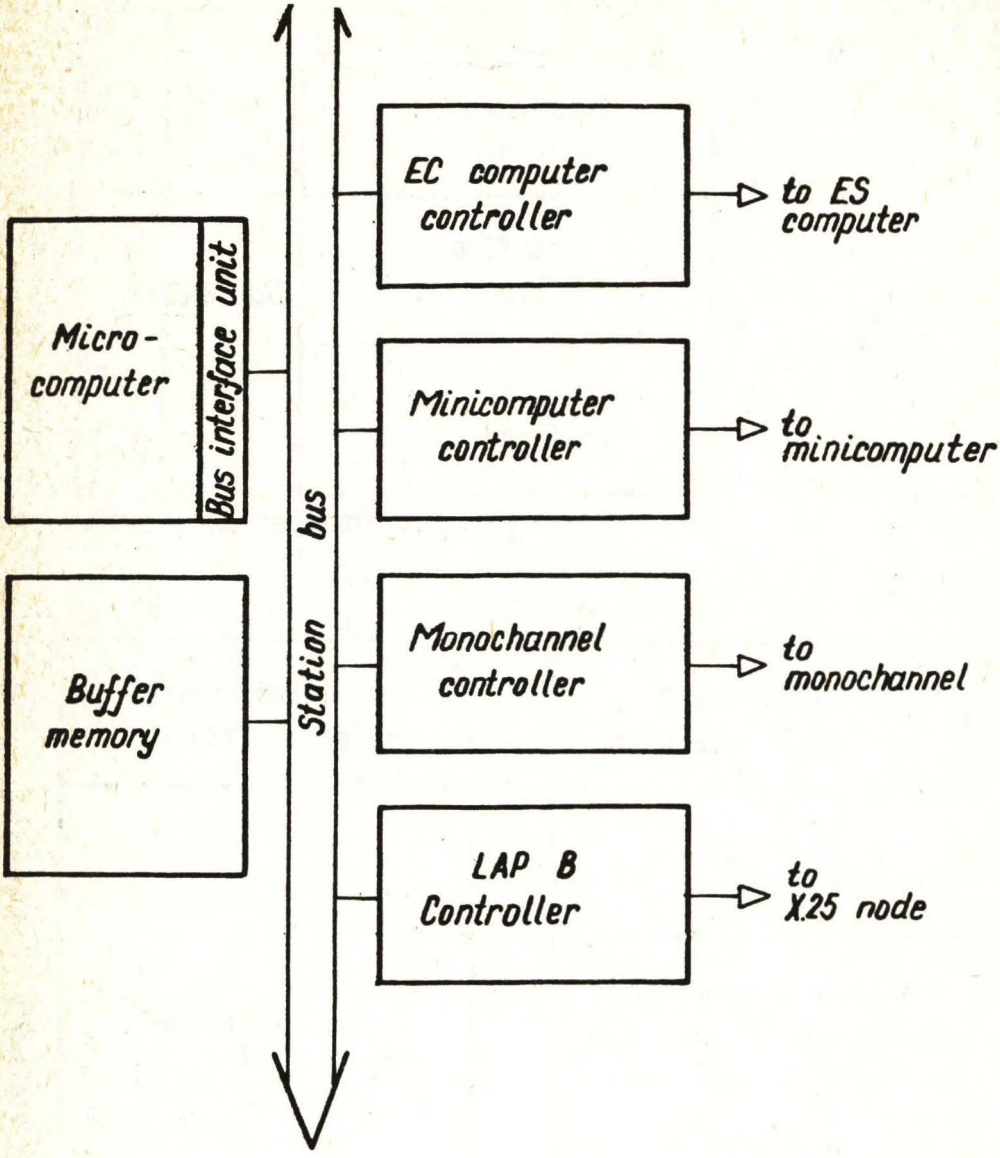

Micro-computer
Bus interface unit
Buffer memory
Station bus
EC computer controller
to ES computer
Minicomputer controller
to minicomputer
Monochannel controller
to monochannel
LAP B Controller
to X.25 node

Fig. 13

SESSION 1
TYPE AND CHARACTER OF COMPUTER NETWORK PROVIDING SERVICES

The Management of Services on Computer Networks

E. B. Spratt, Director and Professor of Computing
D. J. Caul, Head of Hardware and Operations Division
S. E. Binns, Head of Software Division

Computing Laboratory, University of Kent,
Canterbury, Kent, CT2 7NF, England.

ABSTRACT

The University of Kent has been using a fully operational Local Area Network based upon Cambridge Ring technology since late 1979. The network (UKCNET) has steadily grown and now provides communication facilities for all central services. The protocols being used conform to the standards laid down for the Academic Community in the UK for Open System Interconnection. This paper describes the user services being offered with emphasis on the management issues involved in providing such network based services.

Background

The Cambridge Ring is a high speed local area network system based on the empty slot principle. The base clock rate is 10M hertz that, excluding the packet overhead, gives a maximum data bandwidth of 4M bits/second. The empty slot system combined with a small packet size provides access to the medium within a very short time interval and fair sharing of the bandwidth even under very heavy load. However this access mechanism restricts the maximum point to point data rate to one third of this figure i.e. 1.3M bits/second. This figure has proved more than adequate for the current applications at Kent. The design and use of the Cambridge Ring in a major research environment is described by Needham and Herbert (1982).

A standard, called CR82 [2] and [3], has been defined by the UK Academic Community for Cambridge Ring equipment. This standard covers both hardware and protocol aspects up to a level corresponding to a Network Independent Network Service. This service is called 'Yellow Book' [4] in the UK from the colour of the covers the specification was bound in. For some time now the Yellow Book service has been provided between Academic sites over X25 networks. The higher level protocols in use within the Academic Community are all defined to be provided over the Yellow Book service. The existing high-level protocol set includes TS29 [5] which is an extension to X29 and provides for the connection of terminals over Yellow Book connection. File transfer is provided by the Network Independent File Transfer Protocol [6] on which are provided two further services: Mail transfer [7] and Job Transfer and Manipulation [8]. As ISO alternatives become available these will gradually replace the existing protocol set although the means by which this will be done is still a matter for detailed planning (and speculation).

The protocol set in use at Kent was originally based on protocols designed at the University of Edinburgh for the wide area network RCONET. Over the past three years, by an evolutionary development the Academic Standard set have gradually replaced them. The Yellow Book service is virtual circuit based which simplifies the connection between UKCNET and X25 wide area networks. However several services on UKCNET use a much simpler 'datagram' style protocol where this suits the requirements of the service being provided. Examples are the 'name server' function which maps service and device names into network addresses; access to the file server, and a polling function for checking the status of various network devices.

Much of the software effort at Kent has been devoted to the implementation of both high and low level protocols. Research work is currently in progress studying the difficulties in porting protocol implementations between differing operating system environments. This work chiefly relates to the low level protocols which are implemented within the operating system itself or within specialised communications subsystems.

UKCNET

The physical network itself consists of a set of Cambridge Rings connected by bridges. The bridges operate at the 'packet' level and are transparent to all applications software, this mode of operation is based on work by I. Leslie at the University of Cambridge (1982). UKCNET consists of one main distribution ring: the Campus ring, onto which are connected two other rings the Service ring and the Computing Laboratory ring to make up the main service components. Two other, smaller, rings are also connected; one

is used for engineering development and the other for software development and research. The Campus ring, as the name implies, links most of the buildings on the Campus, it is approximately 4 kilometres long. The other rings are confined to the computing buildings.

The physical partitioning of the network, which does not effect the logical view of the network as a single entity, is useful for several reasons: it improves reliability and maintainability; it increases the total system bandwidth; it may be used to improve security on parts of the network and finally it is a good model for the evolution of a local area network to encompass different media.

The Campus ring is the most vunerable part of the network; for reasons of expediency, on some sections, it uses lower standard cable than is normally specified because use was made of existing telephone cables. Although this deficiency is gradually being remedied with the planned introduction of fibre-optic links. Its power is also more liable to interruption from extraneous causes; electrical maintenance work, for example. The loss of part of the network does of course damage the user's logical view of the network as a single whole. The users then need to know something of the underlying structure to appreciate the significance of explanatory messages. Telling the users that the Campus ring will be unavailable for a short period for maintenance work has no meaning unless the users know what devices are connected to the Campus ring and will therefore be unavailable. There seems no easy answer to this problem, apart from improving the reliability and hence minimising the maintenance required for all parts of the network. Certainly the alternative of making the network structure visible, as would be achieved with a with a naming scheme for services that implied the structure, would create worse problems. The current naming mechanism is a 'flat' naming scheme over the whole campus, this enables services to be moved freely to different ring addresses and even different rings without affecting the users or application services. The translation of service names to network addresses is handled by 'name-server' devices and the ring bridges.

The way that multiple ring systems can be used to improve the network bandwidth will depend, in general, on the network traffic patterns. On UKCNET the heavy traffic is concentrated onto one ring, the service ring. This is because to obtain the best point to point data rate on a Cambridge ring the ring needs to be physically fairly small as an aggregate of the number of connected devices and cable length. The current model of network traffic is a small number of hosts which require a high bulk data transfer rate for file transfer between them. The majority of other devices connect to these hosts on a random basis for terminal access, printer access etc. As other localised traffic patterns develop, for example with the increase in departmental facilities and the expected introduction of personal workstations we would expect to add other departmental rings. This would also have the beneficial effect of improving network reliability as perceived by that local set of users.

The security aspect is less important in a University environment than in many other area where local area networks are likely to be installed. For some time the University administration at Kent had a separate ring connected to the campus ring. This gave the required physical security to the network components; they were installed in a secure area within the University administration, this included the bridge connecting them to the Campus ring. The host computer on this ring was able to guarantee the distinction between local connections (coming from a device on this ring) and connections being relayed through the bridge from another ring. There was no possibility of rogue systems claiming to be what they were not. Terminal access was only allowed from the local ring, although file and mail transfer were allowed from the network in general. The key point was that the local ring was necessary to guarantee the distinction between local (to that building) access and remote (general UKCNET) access. The administration has now changed their computing arrangements and currently does not have a UKCNET connection. This change was not due to any problems or dissatisfaction with the service provided on UKCNET.

Currently most of the ring equipment was installed before the CR82 standard was available and has some incompatabilities. To accommodate new equipment to the CR82 standard the Laboratory ring was installed. We expect that other network media will be installed either to provide a cheaper access method for micro computers or for systems that do not support a Cambridge Ring connection. An example of the latter is expected to be an ethernet system either within a department or as part of the central service.

Network Services

The network services, at least as far as the users see them, may be grouped into two main types with a further 'miscellaneous' group. The main types are host computers and terminal concentrators. The hosts are multi-access computers offering a standard interactive computing service. The majority of such hosts on UKCNET are mini-computers running the UNIX† operating system. The terminal concentrators, following X25 terminology called PADs, were designed at Kent, each connects up to 16 asynchronous terminals and one printer to UKCNET.

† UNIX is a trademark of AT&T Bell Laboratories.

The main service host is an ICL 2960 system running the EMAS operating system. Most of the host systems are connected to the ring with some sort of front-end processor which provides direct memory access to the host memory. The front-ends vary in functionality, in some the full network protocols are implemented thus removing this load from the host, in others the front-end is a simple micro-programmable device which just provides a packet service.

All terminal access to the host computers is via a PAD and UKCNET, with the exception of the minimum number of directly connected terminals necessary for maintenance purposes. Public terminals are now available in most buildings on the Kent campus, the PAD system suits the provision of clusters of terminals with a local printer, this is very suitable for teaching applications and also for a terminals in a set of offices located close together. For small numbers of terminals in the more remote corners of the campus statistical multiplexors are used to connect to a convenient PAD. As mentioned above, the protocol used for terminal access is the UK Academic standard TS29. Considerable development effort has gone into the implementation of TS29 drivers for UNIX and the EMAS system to provide the users with a consistent view of these systems and their responses to keyboard input. For example the same control characters are used to control the various flavours of 'breakin' and illegal characters are echoed and actioned in the same way. Both systems are capable of supporting visual (screen) mode editors on the network although the performance of the EMAS front-end system makes this mode of working unrealistic. This system is due for replacement in 1986 so it is not thought to be worthwhile expending effort in this area. Experiments are progressing with two cheap and lower performance networks: Econet [10] and Multi-Link [11] as replacements for the PAD. Simple micro-computers e.g. the BBC micro, would communicate across such networks to a gateway machine connected to the Cambridge Ring. Such a system simplifies the wiring of distributed systems and would be capable of higher performance than the PAD.

The PAD software was implemented at Kent, it is mostly written in the high level implementation language BCPL. The PADs contain a simple monitor in ROM which downloads the main code from a network server (the BOOT server). This simplifies the distribution of new object code to the PADs. This is a significant advantage with 23 PADs spread across the campus. Updating the PAD software by any other means would be a difficult logistical exercise. In principle PADs may be loaded across wide area networks although the speed of the wide area network links makes this unlikely. The bridge computers, that connect the rings together, use the same mechanism to loadup. The BOOT server can be instructed to load special versions of the software to particular devices, for development and testing, but in general the same code is loaded into all PADs. In a similar way, if the PAD software detects a failure it attempts to dump the current store image to a DUMP server. The DUMP server is one of the UNIX hosts, which has tools for performing a simple post-mortem. The PADs contain a simple hardware 'dead-man's handle' for detecting software stuck in endless loops, so that in these circumstances, dump and reload is an automatic operation. Many PADs are kept powered on all the time, so a mechanism is needed to force a reload when a new version of the software is available. This is a request packet sent on the network from a privileged host, the request is of two forms: a soft reload which instructs the PAD to reload when it is next not in use (no virtual circuits open); and a hard reload which forces an instant reload.

The request packet mentioned above is an example of a connectionless operation. The protocol used is called the single-shot protocol. All the hosts and PADs also respond to another single-shot packet that is used to poll the devices on the network by one of the hosts. This is used to produce a simple screen display of the state of the major network components. The image is available to the operators and to the users. For the hosts the number of logged on users is displayed and for the PADs the state of the printer and the number of terminals connected. When a host goes down this is highlighted with a flashing display for the operators. Such a system is necessary as most of the hosts only need operator intervention when they fail. The role of the operators is gradually changing they are becoming responsible more for the communications operations and less for the minutiae of running particular hosts. This has required a gradual training program for operators in post and a certain amount of rethinking of the design of network components intended for operator use. An example of the latter is a network operator station. This project is still under development, it currently acts as a multiplexor for the control consoles for several of the host systems. This has allowed us to dispense with several terminals in the machine room which were only used when a host was loaded or required maintenance. This system will be extended to log messages to the network file server and to accept messages from the network which signal error conditions to the operators. When this system is fully developed it will be capable of displaying information from many sources, and so much thought has gone into the best way of presenting the information in a structured and useful way to enable operators to respond to various errors or conditions signalled by network components.

Micro computers are currently connected via the serial lines on the PADs. They act as terminal emulators and use simple protocols to transfer files to and from the hosts. The Kermit package is used and also some earlier local packages. The use of serial lines is very slow, particularly as the PADs were designed to support fingers on keyboards and not high input transfer rates. However the current high costs of direct

network connections makes this the only feasible general connection method. The use of cheaper networks to connect micros, discussed above in the context of terminal connection, is expected to provide a much better solution.

The miscellaneous services include the following devices:-

* Gateways to X25 wide area networks
* Specialised device handlers: paper tape (rarely used), graph plotter, heavy duty line printers.
* Laser printer controller.
* Broadcast teletext service.
* Terminal booking service and administration server.
* File server.

The X25 gateways were originally developed at Edinburgh Regional Computing Centre and extended at Kent. The gateways include comprehensive accounting and accreditation facilities to control the user access to 'real-cost' networks. For interactive access the users logon to the gateways with their normal user code and password. The gateways provide protocol conversion from TS29 to X29 for those remote hosts that do not support TS29. Full logging of all calls to and from the gateways is provided. As the users have to quote their logon id to access the gateway this logging includes the user identification, the destination, time of call, duration and the traffic statistics. File transfers do not require the same authentication; it is assumed that the hosts generating the file transfer traffic will perform the necessary authentication before initiating the transfer.

The gateways do not allow the use of literal network addresses by the users, instead the gateways have a directory of names and network addresses. The directory also includes an access level for each name, each user has an access code and to access an network name the user's access code has to be at least equal to the access level for that name. This allows a controlled partitioning of the external networks. With the high cost of the networks like IPSS this mechanism is essential to avoid unexpected high network charges. A virtual circuit inadvertently left open across the Atlantic could accrue a charge of 50 pounds overnight. In fact this last occurrence is not likely to occur as the gateway operates timers on idle circuits and would disconnect the circuit after a few minutes. However at the moment there is a fashion for some individuals to attempt to obtain access to remote sites via any network connection they can manage to access (the hacker syndrome). Such access has been detected at Kent, originating both inside and outside the University. The controls and monitoring in the gateways proved invaluable in detecting and cutting down such use. In the academic environment we would prefer to operate in a more trusting and open mode but the need to avoid unnecessary network charges means that some controls are essential. Even for the UK Academic Network (Janet) for which no charges are made, the relatively low speed of our network connection (9.6kb) means that we need to limit inessential traffic so that the genuine users are not adversely affected.

The current gateways are stand-alone devices based on DEC PDP11 computers, with hard discs to hold the fairly substantial filestore required to support the features outlined above. The next generation of such gateways, we would expect to be provided on much cheaper hardware either a custom built micro computer or possibly a personal computer. In either case the filestore would likely be provided on some other device, possibly the network file server or one of the host computers. The advantage in the latter approach is that all the utilities required to administer the gateway files can more easily be provided than on the specialised systems. The utilities perform the updating of the various directories of network addresses and user information and the analysis of the logging information. Currently some of these operations take place on the gateways themselves.

The laser printer service has been developed as part of a research contract funded by the Science and Engineering Research Council. The laser printer itself is a *Canon LBP-10*, this is a modestly priced device, printing at a resolution of 240 bits to the inch (100 bits per cm). A controller has been built to connect the laser printer to UKCNET, this interfaces to the serial input to the laser printer. The controller is based on a 68000 micro computer with 1 Mbyte of store. Information is sent to the controller in a high level 'Document Transmission Format' the controller then produces a complete page image using fonts obtained from the file server and transmits the page image to the printer. Although this resolution produces images slightly inferior, in sharpness, to a good quality 'daisy wheel' printer, this is more than compensated for by the complete flexibility of image production. The image quality is more than adequate for locally produced documents and reports and for final drafts of documents destined for a commercial quality photo-typesetter. This facility has proved so successful (and reliable) that the University has funded an additional printer server for service use. Formerly a prototype service had been provided on the research equipment so that its validity in production use could be verified.

There are two management problems associated with the laser print service. One is the handling of fault messages from the printer itself, this is usually the 'out of paper' indication. As the printer normally

operates without manual intervention such faults are only noticed when the operators collect the output for distribution. The solution being adopted is to route error messages to the operator station on UKCNET. Error messages requiring operator action will be given a high priority for display so that the operators will see them quickly.

The other management issue is the fact that the formatting programs producing DTF output for the laser printer consume a large amount of machine resources. Of the order of 10 million machine instructions (32 bit) per page of output. Also the formatting programs e.g. troff on UNIX and TEX are very large and complex programs. On the main UNIX service machine the troff load was creating an intolerable degredation of the service offered to other users. The first solution to this problem was to force the troff program to be run in a batch mode. Although this did not lessen the number of runs demanded by the users, it forced them to be run consecutively rather than in parallel and minimised the impact on the remainder of the system. In fact the introduction of the batch system improved the throughput of the formatting jobs themselves by improving the overall system throughput The longer term solution to this problem is to remove this load altogether from the general service computer and to move it to a dedicated processor on the network. This operation is scheduled to take place by mid-summer 1985. The exact mode of operation has still to be determined. Their are two alternatives. Firstly, to simply transfer the batch system to the remote machine and to use file transfer to send jobs and job responses between the machines. The second alternative is to use a remote procedure call (RPC) mechanism to initiate the run of the formatting program on the remote machine. This second solution is more elegant but requires modification at the operating system kernel level to provide the RPC mechanism.

The prime advantage of a network system such as that provided at Kent is the flexibility it offers in terms of enhancing the system. Additional devices can be added without reference to the hosts and new hosts and specialised services added without affecting existing services. However one obvious problem arises if terminals are freely added to the network without any commensurate increase in host power. It is not feasible to ignore this problem and to allow the contention by users for the hosts to be a 'free for all'. This causes a degradation in the service offered to all the users i.e. unhappiness for all. A solution proposed by one of our users was to increase the number of hosts so that the contention would never occur. Unfortunately in the current situation of rapidly expanding demand for computing and shortfall in financial provision this is unrealistic to say the least. The alternative of trying to partition the user community onto different hosts so that peer group pressure could be used to limit access does not work. The problem here is that it would require all software to be provided on all of the hosts. The cost of licences for packages makes this prohibitive, it would also be a very inflexible system to operate. The solution adopted is, first of all, each of the hosts operates a limit on the number of terminals it will allow to simultaneously connect. This corresponds to the limited number of directly connected terminals in the non-networking situation. Now each of the hosts can provide a reasonable level of service to those users that can logon. However in the teaching environment we have a requirement to guarantee access to particular hosts at particular times for supervised teaching. This normally takes the form of a class or laboratory where, 16 terminals all want to logon to a particular host to use a certain package or compiler at a specified time each week.

To handle this situation a specialised host called the booking server has been added to UKCNET. The bookings are in units of 30 minutes throughout the day and guarantee a logon to a host. Separate bookings are kept for each host. The bookings are either on a per user basis, in which case the specified user is guaranteed the access independent of the terminal that is used, or the booking can be on a location basis. The location usually refers to a room on the University campus, in this case a number of terminals may be booked. This guarantees that up to the specified number of terminals at that location will have access to the particular host. The Booking Server is accessed by normal logon from any terminal, one privileged user has the capability of making the location bookings for teaching purposes, other users may make individual bookings or their own behalf, up to a set limit per week.

The Booking Server keeps track of the bookings that are made but the hosts have to administer the control mechanism. This takes the form of interrogating the booking server every half hour to obtain the current booking information. The hosts take no further action until the number of connected terminals reaches the specified limit value (the limit is read as part of the booking information). At this point only users who have booked, either explicitly or implicitly via a location booking are allowed to logon and users logged on who have not booked are liable to be forced off, after a suitable warning, to bring the numbers of logged on terminals back to the specified limit. The users that have been logged on the longest get discarded first.

This scheme has worked well for some years in spite of its simple nature. The main deficiency as far as the users are concerned is that all the bookings are of virtual terminal slots on the hosts. The user still has to find a terminal to take up the booking. It is not clear how the booking of physical terminals could be integrated into this sort of scheme. The method used at Kent is to leave this to manual methods at the areas where terminals are in heaviest use.

The booking of terminals by location requires a database to map from network addresses (of the form 'PADxxx/5' - the numeric suffix indicating the line number) to physical locations. This information is duplicated on each of the hosts. When new terminals are installed or existing ones moved the technicians log his information on one of the hosts using special purpose software. The change is then transmitted to the other hosts in the early hours of the morning. The database also contains information about the type of terminal attached to each of the network ports, this is required by various programs that produce screen image displays. On most of the hosts the screen handling is handled by a subroutine library so that applications code does not even need to know that different terminal types are supported. The effect on the users of this system is to allow them to use terminal dependent software such as screen editors without any knowledge of terminal characteristics or even that such knowledge might be required by the programs being used. This simplifies the task of the programming advisory service, makes the provision of documentation easier, and allows a flexible purchasing scheme for terminals as a wide range of types can be used.

With the increasing number of host systems and the steady expansion of computing into teaching applications the administration of the user community by the Computing Laboratory was becoming impossible by manual methods. The decision was taken some two years ago to move some of this administrative load onto the machine that acted as the Booking Server. The Booking Server machine was relatively lightly loaded, it had adequate disc space for the database required and it already included the network communications code to connect it to UKCNET. In addition a simplified version of the user database already existed on the Booking Server so it could validate user booking requests.

This exercise has now been completed. The Admin Server as it is called, is essentially a private database machine. The staff concerned with administering the users connect to the Admin Server from any terminal, and manipulate the data using a set of purpose built commands e.g. Createuser - to allocate a new user code, Accredituser - to allow an existing user to access another host etc. There are 'batch' facilities to accredit new users in bulk at the start of Academic courses. The hosts themselves contain their own mechanisms for accrediting and naming their users. The Admin server uses the mail system to communicate the necessary actions to be taken by the hosts. The security of this scheme is obtained by the hosts validating the network address of such messages.

The centralising of the administration has been most beneficial. The use of a dedicated machine (apart from the Booking Server load which is negligible) has meant that the response to the administrative staff is consistent and usually rapid. As this machine does not have a user population there is no problem of security, apart from keeping the administrator's password secret. The central database ensures consistency and means that the same login id is not used on different hosts for different people. However the system did require a considerable development effort because it was based on a system designed for communications work. Thus there were few utility programs available to aid in development. In particular the compiler had some bugs and diagnostics were non-existent. Extensions to the Admin Server would benefit from it being moved to a more standard environment for example a mini or micro computer running the UNIX system, this is under discussion.

Conclusions

It will be apparent that our solutions to the various management issues that have arisen from the development of the local area network at Kent are pragmatic in nature. This is in part due to difficulties in providing a computing service at the same time as the network facilities are being designed and implemented. It is also due to the overall lack of previous experiences in this area that we could draw on. So far we have had the advantage of running operating systems on the hosts that are available in source code. This is not likely always to be the case. For example, UNIX system are now appearing in 'binary' only form. This means that until system independent standards for management facilities are available, implementors will be forced to link together the existing higher level facilities . This is one of the reasons we have not adopted network level authorisation facilities. The computing system model in use at Kent is that of a cooperating group of autonomous systems. An advantage of this model is the ease of handling groups of specialised users, particularly for teaching. This improves security and simplifies the control of computing resources.

The network developments at the University of Kent have been a team effort from many people over a considerable period of time. The authors would like to express their appreciation to all the staff at Kent who have contributed to the the development of UKCNET. Finally we should like to acknowledge the considerable help that the University of Kent has obtained from The University of Cambridge Computer Laboratory, the Computing Science Department at University College London and the Regional Computing Centre at the University of Edinburgh.

References

[1] Needham, R.M. and Herbert, A.J., The Cambridge Distributed Computing System (Addison-Wesley, 1982).

[2] Sharpe, W.P. and Cash A.R., Cambridge Ring 82 - Interface Specification.

[3] Larmouth, J. (Editor), Cambridge Ring 82 - Protocol Specification.

[4] A Network Independent Transport Service (1980).

[5] Character Terminal Protocols on PSS, a recommendation on the use of X3, X28 and X29, (1981).

[6] A Network Independent File Transfer Protocol, (revised 1981).

[7] JNT Mail Protocol, (1982).

[8] A Network Independent Job Transfer and Manipulation Protocol, (1981).

[9] Leslie, I.M., A High Performance Gateway for the Local Connection of Cambridge Rings, in: Ravasio, P.C., Hopkins, G. and Naffah, N. (eds.), Local Computer Networks (North-Holland, Amsterdam, 1982).

[10] Econet - details available from Acorn Computers Ltd., Cambridge, UK.

[11] Grant, J.S., Multilink - A Ring that Provides a Simple Interface for Individual Outstations, in: Dallas, I.N. and Spratt, E.B. (eds.), Ring Technology Local Area Networks (North-Holland, Amsterdam, 1984).

References 2,3,4,5,6,7 and 8 are available through the Joint Network Team, Rutherford Appleton Laboratory, Chilton, Didcot, Oxfordshire, OX11 0QX.

COMPUTER NETWORKS IN THE ACADEMIC ENVIRONMENT

Mieczyslaw Bazewicz
Institute of Control and System Engineering
Technical University of Wroclaw
Poland

Franz Stuchlik
Sektion Rechentechnik und Datenverarbeitung
Technische Hochschule Magdeburg
G.D.R.

Most of our experiences on networking in the didactic and research laboratories have focused on the interprocess communication based on hardware/software/firmware technology solutions which must be strongly subordinated to the complex application environment. The growing number of PCs and information resources change the character and the functions of university CCs. Information services for the academic community need more sharing and specialization of the resources and the operations by hosts and networking services. An approach to the design and implementation of distributed data system services for complex environment will be presented. This contribution introduces new ideas and the foundations of a design methodology based on actual and future information technologies.

1. INTRODUCTION

Passing on the latest knowledge in various scientific and professional disciplines to their students universities and other higher education institutions /academies, polytechnics etc./ also teach them how to use it together with auxiliary media necessary to attain this purpose. This general statement became particularly important due to the appearance and fast wide applications of the extremely efficient auxiliary media which computers are. The consciousness of a need to project social computerization effects resulted in the necessity of answering the question "What should we do then?". And at last it is necessary to realize the fact that teachers should be able to cope with this development challenge and show their students the new tools and master the secrets that computers have and how they should be used.
Teaching computerization methods i.e. applications of computers is subject to the necessity of permanent changes. It results mainly from the rapid growth of computer and communication technologies. Whereas its effect is an expansive spectrum of disciplines that use computers as well as of qualitative and quantitative changes in the character of interdisciplinary relations - due to applications of computers. It consists among others in a more intensive passing from abstractness /reference models/ and fortuitousness to serviceableness and naturalness in computer applications. The significant factor of these changes is a revolutionary character of network and microprocessor techniques /VLSI/.
The consequence of that is a necessity to integrate huge information and programme resources aggregated by the techniques in the form of new methods for computer knowledge representation /VLDB/. Figure 1 shows the integration trend of system functions and information representation.

Computer Networks	distributed system, distributed computing, distributed service, expert systems	distributed data base, knowledge base	user-user communication, interprocess communication, knowledge representation
Multi-, Mini-Computer	time-sharing, multiaccess, parallel processing	data bases /VLDB/	specialized data processing word processing image processing
Main Frame Computer	centralized system, centralized service and computing	branch data bases	data storage, information retrieval,
μC	multi and microprocessor system	files	data processing, user-computer communication,
μP	multifunctional components /VLSI/	data	control process
Integration degree	Physical and logic components	System complexity	Data and information resource	Service and operational function

Figure 1 Technological and functional relations in computer information systems

These changes also involve the need to modernize the laboratory basis of computer sciences for teaching as well as the renewal of computer science school. The idea is to lay greater stress on the engineering of information technologies and on head-work technologies for users of computer systems. It is also very urgent to develop to a higher extent theoretical foundations of computer science with methods for modelling and evaluation of the adequacy of models of computer systems to reality, taking into account aspects of time, space, functions and behaviours of computer systems.

2. TECHNOLOGIES OF MENTAL-WORK

The evolutionary character of computer applications imposes the need to orient computer science methods more towards the man than towards hardware/software media. It should result in man's better functional integration and higher level of semantic compactness of information processes with thinking processes aided by computer systems. From among those processes we distinguish an automation of simple information processes such as collecting, transfer etc. and an automation of complex processes such as information processing and interpreting which we number among mental-work technologies. The difference consists in that the semantic value of a process does not change in the first technologies whereas in the latter there changes the semantic of the process. The change in the input value of the semantic is accompanied in an information

process by set of logic rules subordinated to pragmatism of human thoughts and actions. The specification of a mental-work technology is determined by human cognitive processes, a spectrum of the semantic of information services and the range of functions performing the logic of information processing rules provided by auxiliary media e.g. the computer.

There are some restrictions appearing in the applications of computers, however they are not so evident as in other cases where implementations concern one profession or individual disciplines or even parts thereof.The range of computer applications covers new disciplines and professions. For research as for teaching one can generally ascertain that they are applicable first of all to man's mental-work and information processes. That is just the context we mean while talking about "new technologies or methods of mental-work", where the attribute "new" should be identified with computer-aiding of that work. Computer-aided information processes and computer-aided mental-work technologies linked with them cannot be instantly available and that is why they must be first developed and then variously tried. It requires introductory solutions and trials which affects the development of local computer networks in universities and improvement of methods for teaching, research and engineering.

The correctness of this approach was confirmed in the early seventies by studies of methods for computer-aided work conducted by TH Magdeburg and the Technical University of Wroclav. These studies concentrated on computer-aided work methods for university teaching and research. The starting point for the studies was a computer-aided acquisition of measuring dimensions, initial processing of measuring data, storing, processing and a representation of results from the technology and construction Diesel machines, from the evaluation of mechanical compositions dynamics and of chemical processes etc.. Simultaneously with these applications computers aid the control in the real time mode cold-plating shop installations as well as production machines. The internal complexity of control of these processes required the following computer-aided modes:

1/ job mode - continuously computer-aided because it concerns the job entry, processing and solution
2/ disciplinary mode - concerning a discipline aided in a wide scope.

Next applications' consisted in connecting the mental-work aid with the experimenter`s workplace. The idea was to initiate, influence and use the object under investigation in the course of the experiment.

The computer-aided teaching covered such areas of mental-work as experimentation, recognition and identification of complex objects under examination and systems [5] in such areas as mechanical engineering, power industry, chemistry and medicine. It aimed at teaching new head-work technologies in the area of such method as: experiment control, multistage and multigradual identification, multimization etc.. Teaching tasks were performed the way which results from available functions of the structure and architecture of the computer system as well as from data bases /experimentation data bases/ and from libraries of engineering programmes. The latter were fully separated from the experimentation data bases. However, in general the necessary and suitable for the studies /in a semantic sense/ information resources were inaccessible for the experimenter and it was impossible to use techniques for access to centrally located resources. The computer-aided teaching was going on in accordance with needs in the real-time or dialog or batch

processing modes by time-sharing and multiaccess computers. At the same time there was a growing demand for necessary software because user needs could not be satisfactorily supplied. In consequence of the big computer load the user was unable to obtain the access to resources. However the experimentator or student did not have a chance to use such services of the system as: access and manipulation of remote data bases, transfer of files between data bases, access and manipulation of a joint /for many experiments/ data base, manipulation of jobs, graphical edition of results etc.. The making of proper software layers did not include only proper interfaces between layers but also a carefully adjusted interface with the user. It was the reason that the user was involved in the software developed in the software developing process since its very beginning. The preparation of specified dialogue programmes and its selection from the viewpoint of "menu", the addition of definite media for graphic data representation, the offer of the function "help" and other useful facilities became necessary to provide an effective utilization and involve researchers and students in the use of the new mental-work methods more and more perspectively.

The rich development of computer-aided research in such areas as mechanics of stresses, construction of gears and the like produced the need to develop and offer exquisite semantically-strong integrated information resources of different types as well as first attempts at developing computer-aided lectures were undertaken in the area of computer science /methods of programming, though to slight extent of the discipline, nevertheless 400 students could attend the lectures simultaneously. The usefulness of these developments pointed to great advantages particularly in integrating data with images and sounds since first experiments. By interacting sounds, images and data the new form of aided-lecturing appeared which thus caused the change in the character of these courses. Many changes in the direction of enriching lectures with experiments became possible to be introduced due to the following solutions.

The lecture hall shown in figure 2 was equipped with the following:

1/ a microcomputer with remote entry to computing facilites of large computer system /EC/, an SM-series minicomputer, a satelite digital graphics system with wide peripherals,
2/ a TV camera to investigate the course of actions in labs and computer rooms,
3/ wire and wireless phonic communication devices.

The hardware/software resources of this kind are used to: develop, edit in a source code and adapt programmes for the interpretative processing method, compile programmes including one task, give access to programmes and data stored as programme libraries and access to systems of data bases, graphical manipulation and representation of phenomenona and physical objects. By using information resources in a routine manner /and by adapting himself to being in contact with them/ the student of computer science or any other close discipline learns to find sources of errors in his programmes, can evaluate efficiency of his computer-aided methods and at last introduce gradually new mental-work technologies to his professional, research and learning activity. The introduction of moving pictures produced by the TV-camera or computer graphics system permits a demonstration of the behaviour of programmes and executions of commands as well as generating and holding interrelations between actions and reactions. The application of resources during lectures and classes is supplemented by individual forms of their utilization. These purposes are attained by special teaching labs equipped with terminals for group training and labs

for improving computer-aided mental-work technologies. The labs involve the necessity to give access to information resources which is important for problem solving. These are the programms for text editing, graphical representation and manipulation, search for information.
The student develops internally by his contacts with computer-aided design, expands his requirements for practical applications and implementations of new mental-work technologies /e.g. learning/ and therefore can stimulate their progress. From these various applications of information resources presented in this paper students train themselves simultaneously to create components of this kind which is a primary requirement for using new mental-work technologies.

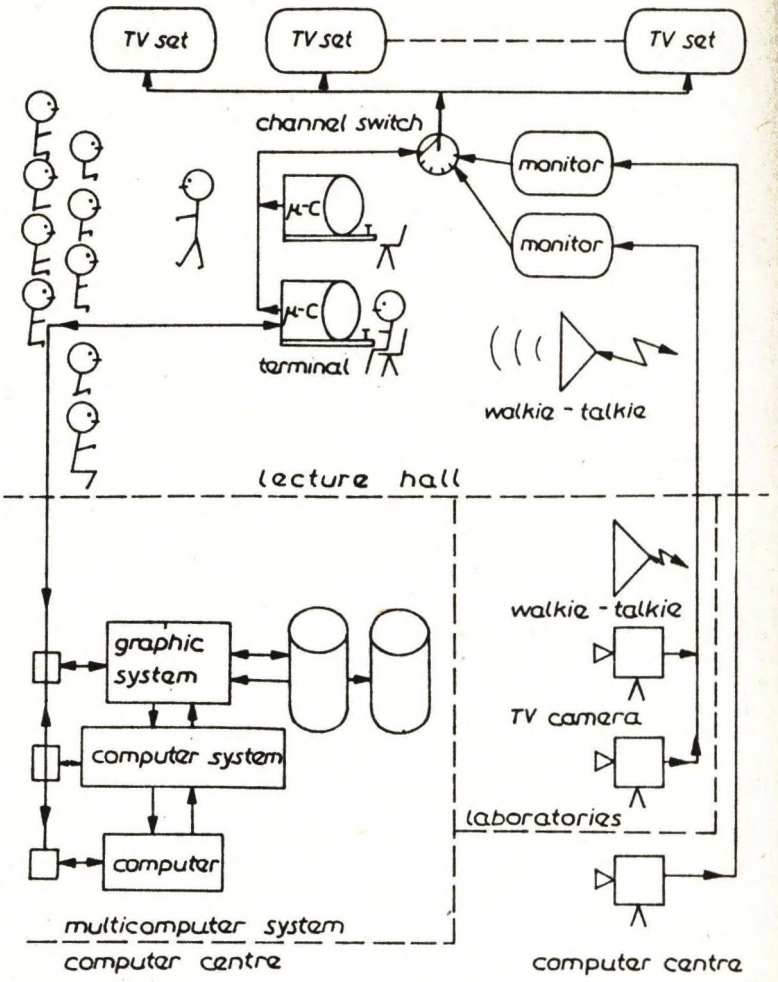


Figure 2. Computer-aided lecturing system for large numbers of students

3. FUNCTIONAL CHANGES IN THE UNIVERSITY COMPUTER INFRASTRUCTURE

As a result of the experience gained from applications of computers, used specificaly for this purpose to high-level training of students academic environment crystallized user requirements related to the development of network techniques. They set out the need to pay particular attention to developing higher layers in the architecture of the OSI/RM distributed data processing system. This can be considered and achieved only by strict relating of network solutions with potential university user requirements. Resources and their access priorities should be so arranged that they do not cause excessive waiting times and any unexpected interruptions in mental-work which may result in turning round student-user's attention or in dissociation of his concentration either in disturbances in the thinking process.

In recent years significant changes in the university computer infrastructure can be observed. Labs for teaching various disciplines and professions are being equipped with minicomputers having multiaccess to specialized and permanently growing programmed/information resources. They specialize in offering some computer services such as computer graphics, text processing, specialized engineering programmes /e.g. simulators/ etc.. However these installations lack mechanisms for mutual offer of services as well as access to university main Frame resources.

Recently the number of microcomputers and personal computers installed in universities for teachers and students has been rising fast. It changes with blind adherence to principles the character of user demands for large main frames.

Though they are sufficiently satisfied with computing facilities, which they possess nevertheless demands for information services such as access to large data bases and special programme libraries cannot be satisfied. These services are offered by large installations concentrated in huge computing centres. Changes in roles, functions and scope of information services are rendered by the computing centres. The centres get specialized in services. From them come new functions and activity forms in the processes of making, maintaining and offering information resources.

This fact stimulates the need to apply in academic environments network techniques where personal computers possessed by teachers and students will become multifunctional terminals in the distributed system. It will allow for access to differentiated information and programme services preserving, however, the independance of performing computations by their own micro- and personal computers.

Meeting these university environment needs lies in local computer networks /LAN/, which offer services of different classes and various types of information and programme resources based on mass and distributed memories and H/S mechanisms. As an example there can be devices to perform specialist processing and data presentation services such as: computer graphics equipment text-processing and editing devices components to collect data, their initial processing and edit results as CAMAC as well as components used to create and apply the knowledge computer representation called also expert systems. The latter are particularly important for teaching students the creative knowledge of computer science. Because the idea is to aquire skills oriented towards the philosophy of fifth-generation computer logic. It concerns designing and implementing problem-solving computer-aided methods as well as setting up an information basis to integrate models of behaviours, phenomenona and processes encountered in various

scientific disciplines. There are the knowledge bases and mechanisms of inference logic that serve optimization and global efficiency evaluation of phenomena and processes under examination and underlie the substance of future computer knowledge representation techniques.
In the seventies activity on developing a WAN-type network for universities e.g. the computer network DELTA in the GDR [8,9] and the computer network MSK in Poland was undertaken. Also the work on pilot LAN-type network projects was developed. These projects are based upon the seventies' technologies and used more to teach methods to control research experiments and methods to design and screen distributed processing techniques than to integrate of mass information and operational functions. See figures 3 and 4.

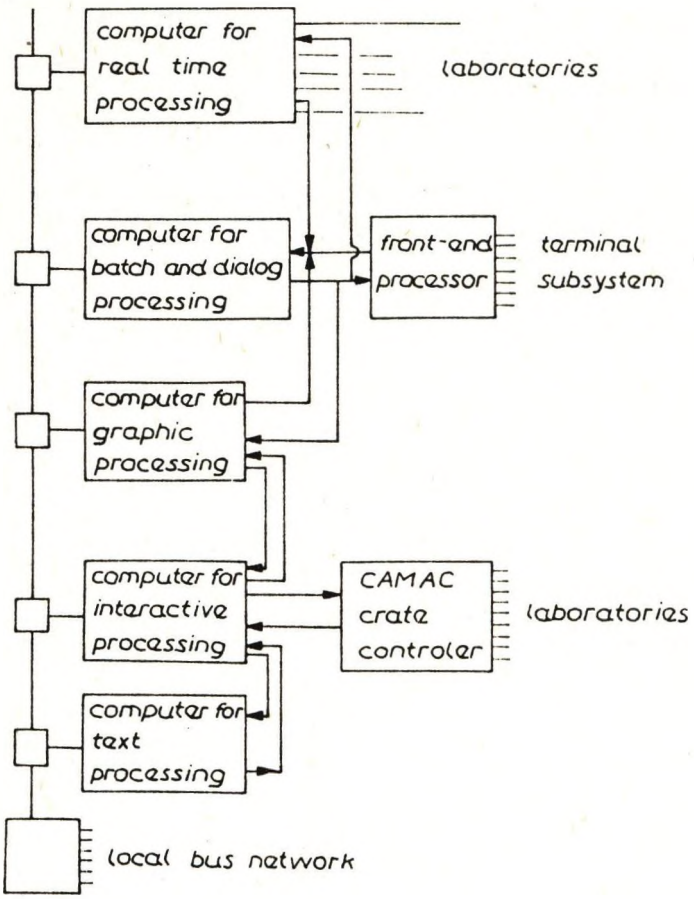


Figure 3 Multi-service local academic computer network

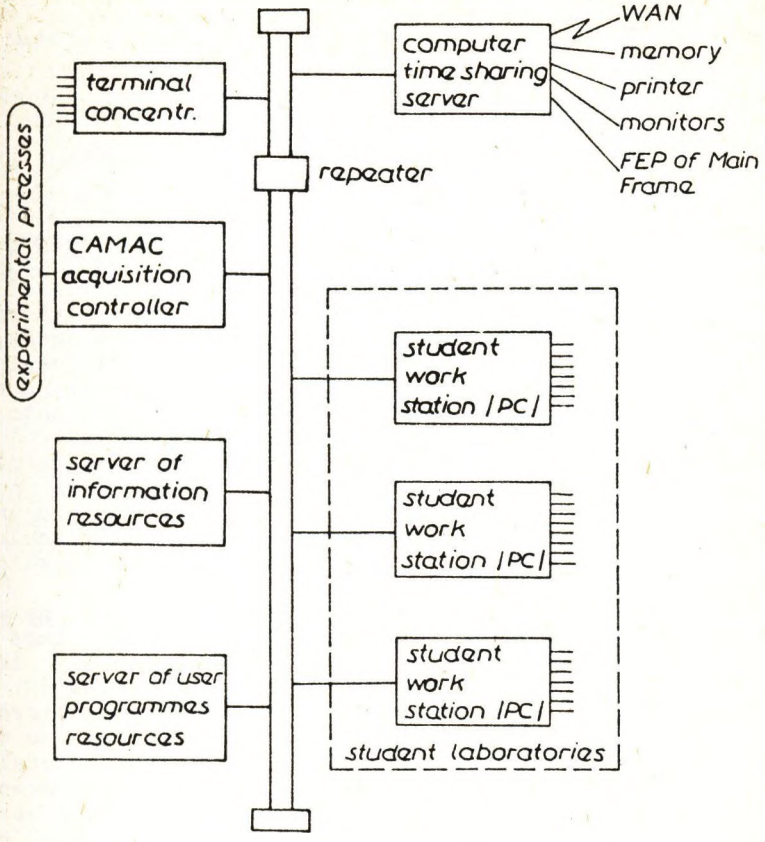


Figure 4 Multi-access local academic computer network
Note: PC - Personal Computer and video, printer, flopy disk, etc.

Significant restrictions in services offered by the LAN-type and WAN-type networks result from too scarce information and H/S resources. As an example of this statement the following characteristics can be quoted: too small number of terminals, mean data transfer speed less than 2.4-9.6 Kb/s, low transfer reliability of media, of channels and of multiplexer devices, inadequate range of operating system functions in hosts /in view of the number of simultaneous accesses from terminals to resources/, lack of effective time-sharing procedures and of bigger operating memory capacity as well as inflexibility in H/S modifications due to the growth of new implemented network services.
The presented situation of the information infrastructure as well as the characterization of information and mental-work technologies in academic environments points to a deep specification of methods for teaching computer applications and for developing computer science in the higher education system. It means that information systems expected by these environments should be perfect in respect

of user capabilities, nevertheless very complex in regard of their layout of H/S, information and programme resources. This problem relates not to computer and communication techniques but above all to the need of more perfect knowledge of design methods and development philosophy for computer information systems of this kind.

4. DESIGN AND TRAINING OF COMPUTER INFORMATION SYSTEMS

The up-to-date knowledge of methods for applications of computer science techniques related to designing of computer information systems is insufficient. It manifests itself particularly sharply while designing complex computer information systems that need network techniques, time-sharing techniques as well [4]. The key role is played here by some deterrents e.g. diversity of meaning of notions applied, and multicontextuality of their interpretations, decidedly static approach to constructions in hardware, software and firmware, unacquaintance with distributed processing methods and techniques and with the role of computer operating systems in these constructions. The equipment-oriented approach to designing and efficiency evaluation of systems being developed is still strong and traditional as before. The understanding of notions, roles and importance of interprocessor communication services as well as the idea of time-sharing techniques, updating, and synchronous data access mechanisms etc. are inadequate in information technologies. Also the incompetence in gaining profit from the techniques is often observed.

Multiaccess patterns in computers and computer networks are not complementary from the viewpoint of the architecture and policy to control data processing processes and interprocess communication. A lack of comprehension of distributed processing systems in terms of functions, time and space is noticeable [4,7]. As a result of the fact that the knowledge of systems design was based on traditional techniques, new professional network developments are slightly effective and non-progressive for academic environments. It also results from this that designers lack an adequate creative knowledge to realise the complicated information systems. Designers of the global as well as the local network eliminate from their field of vision problems of information services and resources, their specification and H/S mechanisms necessary for their efficiency. The designers` attention concentrates on interprocess communication problems provided that problems of processing techniques are obvious and they don`t need any modification or new solutions while new communication technologies are implemented.

In reality interprocess communication solutions impose functional changes on the mechanisms that cooperate with the processes of communication, processing, presentation etc. As an example of these mechanisms are the following: presentation, updating, storing, maintenance and others. It also concerns the organisational activity of computer centre host operators and the necessity of their adaptation to specification of services rendered by distributed systems and above all to creating and offering information resources /e.g. by specifying host service specializations/.

As result of these observations and experiments the new concept of understanding the nature of computer science and the philosophy of its methods and techniques. The idea of the accepted methodology is a withdrawal from the fuzzy set of multicontextual computer science methods $[H_M, S_M, A_M]$ on behalf of a monocontextual methodical approach to designing of hardware (H), software (S) and applica-

tions (A) i.e. $A_M[H,S]$. This approach concerns above all with techniques of computer science applications, assumption of a unmistakably defined notion of the information system to include as information technologies in IS, computer user mental-work in the computer information system /CIS/ as computer technologies in DPS [1]. The computer information system /CIS/ and its subsystems are treated - in respect of their semantic - in terms of attributes of a three-dimensional reference model of the distributed data processing system's architecture. Thismodel is defined by "services" in terms of its application, by "control" in regard of its processes and by "structure" in terms of its construction /in a sense of its statical optimization/. The approach to designing and implementing the system of this sort is user-oriented /The traditional design approach is only structure oriented./. It stands in hierarchical opposition to the approach assumed for interpretation of the OSI/RM architecture model and expands the model with new layers that define attributes of its environment /figure 5/. This approach and the CIS design methodology based on it was presented in detail in [1-3].

The idea of the CIS design methodology lies in the assumption of the definition basis that includes CIS user requirements, a three-attribute model of the CIS architecture as well as a uniform and semantically-coherent language of CIS notions. These rules underlie definitions of all classes within the model i.e. systems and their information processes, tasks performed by these systems, inside and outside interfaces of the system architecture, applied information and user mentalwork technologies for CISs, services and functions of physically-logic components and others.

The accepted methodology is characterized among the others by the fact, that LAN and WAN-type networks are treated as open projects

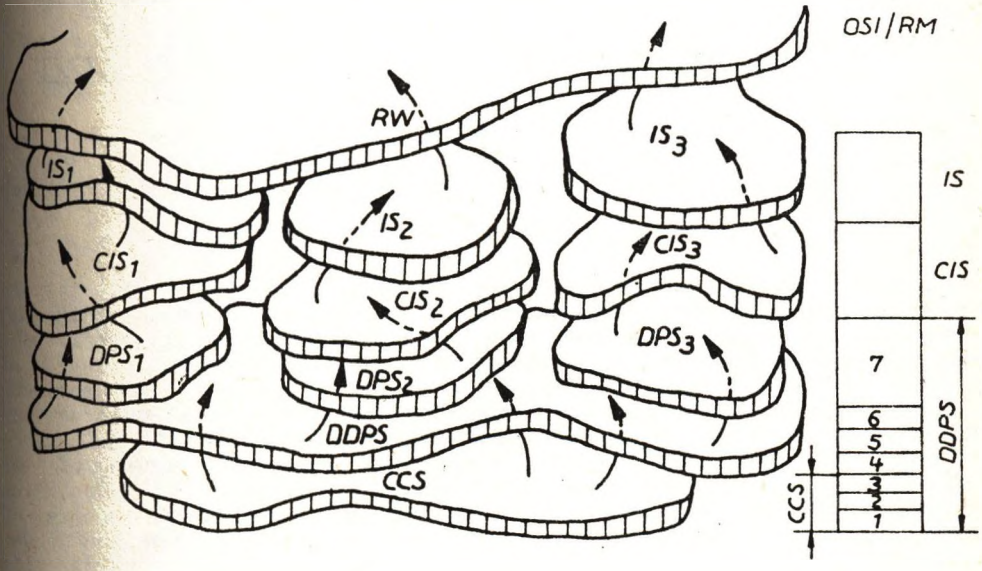


Figure 5a Multiayer model representing the two methodical design approaches to CIS: media-oriented approach

Note: the arrows mean information tasks provided by lower layer to upper layers and services provided by upper layer to lower layers.

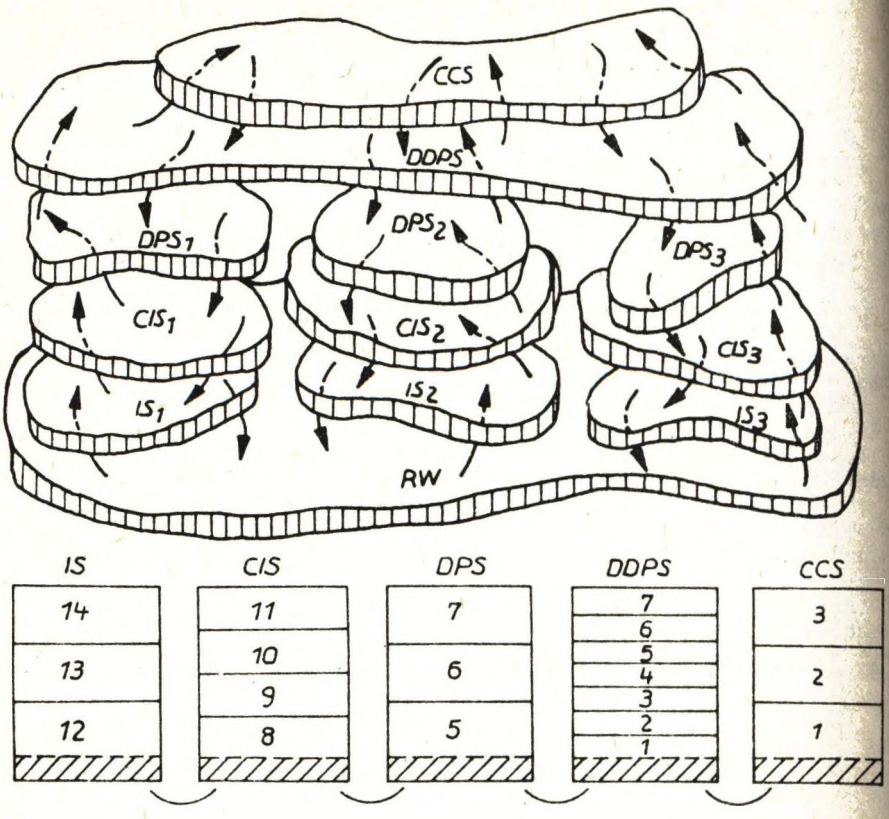


Figure 5b Multilayer model representing the two methodical design approaches to CIS: user and application-oriented approach

of distributed data processing systems with no restrictions regarding interprocess communication solutions. While analyzing and constructing these systems the emphasis is laid on H/firmware mechanisms and their operating systems that work at the level of higher OSI/RM layers independantly of their interprocess communication. Here the operating systems are treated as a certain type of distributed operating system implemented at all time-sharing Hs and servers at the level of adequate layers of the system architecture and its environment. Their properties decide on functional universality and flexibility of designed components of the LAN/WAN networks. The basic principles of the method refer strictly to the system architecture oriented towards its environment and the architecture of any subsystem /i.e. toward users/ whose characteristics are a reference point for making decisions about system's solutions regarding services, functions and protocols. The experiences gained from the design and constructions of highly-complex information systems indicate that using the method their designers do not confine themselves only to topology, structure and interprocess communication for distributed

systems. Besides this they can get help to solve such projects which could not be solved hitherto [6]. By including relations between the distributed data processing systems and the computer information systems /i.e. its environment/ the designers are strongly supported while solving and deciding on these projects whose solution results from an environment specification as e.g. determination of requirements for superiority of distributed systems over centralized systems, establishment of criteria to optimize design decisions, derivation of relations between particular parameters of the system and its reference model, evaluation of the resources allocation level assumed in the model, specifications and assessment of dynamic parameter regarding the distribution, checkup and validation of system design solutions, determination of applications and service quality, specification of information services in regard of the environment and internal components of the system, performance evalution, selection of the system user mode and task services, selection of techniques and strategies to control the distributed data processing at the physical, syntactic and semantic level etc..

The starting point for achieving practical usefulness of the presented methodology is a modification of courses and familiarization of changes in understanding creative aspects of the information philosophy. The beginning of these changes originates from teaching and project experiments. The observations that have been gathered hitherto indicate that familiarization with the knowledge of this methodology is faster and more effective by students that are familiar with foundations of microprocessor technologies then by professionals in computer science who were trained in the sixties and in the early seventies. That is why that not only the supply of equipment and spontaneous dazzle of people by microprocessor technologies but also university training of highly-qualified specialists in the methodology for creative shaping and applying new computer science techniques.

5. CONCLUSION

In the situation when the number of personal computers grows rapidly it is becoming obvious to try to join them with a network. It seems to be a technical problem that can be solved more easily than providing the numerous personal computers and its user consumer with information and user programme resources.
The trend to enrich network service due to the functional integration of the microprocessor technology /VLSI/ is accompanied by the trend to reduce the information entropy which consists of the integration of information collected and offered by networks. Spatially large information resources are integrated for the networks but this integration cannot be achieved only by the interprocess communication technology.
This integration relates with a new form of information aid called a computer knowledge representation which can get favorable facilites from microprocessor technologies and networking. This tendency expresses itself in the form of higher level of information services based on knowledge bases and logical inference mechanisms, enriched by new semantic values and presented graphically or as texts etc. This raises the need for other computer science thoughts and for a new computer science school. We are appreciative of the discussion of many problems mainly in multiprocess communication and pointing to interesting ideas in this area given by Dr. J. Pužman.

REFERENCES

[1] Bazewicz, M., Zur Projektierung Informationsverarbeitender Systeme, Dissertation (dr.sc.nat.), Technische Hochschule Otto von Guericke Magdeburg, 1984.

[2] Bazewicz, M., Stuchlik, F., Der Einfluss neuer Technologien geistiger Arbeit auf die Architektur und Projektierung Informationsverarbeitender Systeme, in Proceeding "Internationaler Kongress Datenverarbeitung im europaeischen Raum", Wien, Maerz 1984.

[3] Bazewicz, M., Stuchlik, F., Architektur und Projektierung von Informationsverarbeitungssystemen, Wissenschaftliche Zeitschrift der TH Magdeburg, 1985.

[4] Bazewicz, M., Stuchlik, F., Problems of computer system/network design and application methods, Wissenschaftliche Zeitschrift der TH Magdeburg 27, 1983, Heft 8.

[5] Borzemski, L. Koszalka, L., Statistical data analysis using a data base for experiment design and system identification, Proc. COMPSTAT 1984, Physica-Verlag, Vienna, 1984.

[6] Goos, G., Hartmann,J., Distributed Systems-Architecture and Implementation, An Advanced Course, Lecture Notes in Computer Science, Springer Verlag, Berlin, 1981.

[7] Puzmann, J., The relational approach to the computer-communication architectures, Wydawnictwo Politechniki Wroclawskiej, Seria: Biblioteka Informatyki Szkol Wyzszych, 1985 [in editing].

[8] Stuchlik, F., The strategy of preparing the applications of the computer network DELTA by the Management Information System of Higher Education in the GDR, COMNET`81, Budapest, North-Holland Publishing Comp.

[9] Bazewicz, M., The Design, Implementation and Information services in Interuniversity Network /MSK/ - Results and Experiences, COMNET`85.

THE DESIGN, IMPLEMENTATION AND INFORMATION SERVICES IN THE INTERUNIVERSITY NETWORK /MSK/ - RESULTS AND EXPERIENCES

Mieczysław Bazewicz
Institute of Control and Systems Engineering
Technical University of Wrocław
Poland

The paper contains conditions and solutions applied to construct a heterogeneous computer network of the WAN type for the requirements of the academic community of Poland. The designing and documenting methods, the experiences achieved during the implementation of communication protocols and the selection of hard and software components, as well of the consequences in network services resulting from the host heterogeneity are discussed. The means and directions of the network information service development the network management and its maintenance and the organization of shared using and of creation of distributed network resources are indicated.

INTRODUCTION

Since the middle seventies in Polish academic environments numbering totally 49 universities there has been the intensive concentration of computing facilities in the form of university computer centres. These centres render computing and information services in the batch and multiaccess mode. Beside them there are also specialistic student and research laboratories where smaller computer installations are set up for needs of various scientific institutes in the universities. The centers give also classes for students and in particular teach them how to use computers in engineering, economic, humanities, agricultural and educational disciplines.

Along with that development in 1977 the work on build-up of a main technical and organisational base for improvement of information services, mutual utilization to hardware/software, program libraries and information /data bases/ resources was undertaken in Poland. The activity was oriented toward the development of network techniques and its starting point was the training of specialist staffs in this area as well in methods for acquisition and access to information resources /these tasks have been achieved through Scientific Schools held jointly by Czech and Polish academic environments [2]/. Necessary were also the following: methodological knowledge in network designing and implementing [3], [4], capabilities of precise formulating requirements for the network [1], [6], [7], of defining the network architecture and user conditions such as: availability of computer and communication hardware, basic software and information resources [8], [10], [11]. The, work resulted in the implementation of the pilot Interuniversity Network

/MSK*/ in definite academic environments.

Parallel to that activity in 1981 the project on construction of the academic computer network /ASK**/ in Warsaw was initiated to meet complex needs of the Warsaw environment that owned large potentialities of various computer and information resources. The assumed solutions regarding the construction of the network differed: different were characteristics of hired telephone lines in city communications and in long-distance communications and different were also proportions between the projected batch processing share and the multiaccess share in utilization of both the networks. Both the mentioned networks were developed by two separate design teams.
However the MSK also admits of such solutions that allow for cooperation of the two networks in the mutual transfer of resources at the level of interprocess communication.

ARCHITECTURE OF THE MSK

The major design assumption for the MSK was to preserve the potentiality to extend gradually the network as only new hardware /H/ and software /S/ facilities would appear. Meeting this assumption was possible due to acceptance of the modular structure pattern for the network which was based on a seven-layer structure presented in the reference model OSI/RM. The acceptance of this model was a basis for option of ways to build the modules for different host access techniques - in respect of used operating systems - for ODRA 1305 and EC-System and thus to attain the network heterogeneity. This approach also allows for replacement of one layer of for its modification without the necessity of modifying other layers. The gained experience points to big applicability of the OSI model to designing of other networks. Considering the unpredictable development of sorts of resources, in particular H/S resources, the standard X25 was determined for communication services. As concerns processing services those of them were picked up which are provided by computers ODRA 1305 - equivalent of ICL 1900 with the operating system GEORGE3 + MOP - and by EC equivalent of IBM360 and 370 with OS/MVT + TSO. The two types of computers have differed and flexible modes of user task services, starting from batch processing through multiaccess up to remote jobs entry. Taking advantage of the facilities the following were assumed: virtual terminal services, file transfer, on-line access to data bases in the retrieval mode, dialogue mode and to a lesser extent batch-processing and remote jobs entry /RJE/. Rendering proper services to the academic environments by the network was a priority while processing services were designed. The environment of the network characterized as a set of users executing information processes was defined as the eighth layer in relation to OSI/RM [5]. This layer is called computer information system - CIS layer. The network services of this layer are defined by the notions of an information /DB/ service and of a user programme /UP/ service in relation to jobs demanded by users in the form of information tasks /IT/. These services [4] are characterized by the following attributes at the pragmatical, semantical and syntax level:

---

*/ MSK - Międzyuczelniana Sieć Komputerowa

**/ ASK - Akademicka Sieć Komputerowa

- Range, correctness and updating of data bases /DB/ user programmes /UP/ offerred by the network.
- Level of logic complexity of information tasks executed in the network that depends on a sort and readiness of H/S, DB and UP resources.
- Information quality i.e. size, form, time of editing results of network services.
- Rate of attaining services by users i.e. response time, execution time.
- Simplicity of the user-network cooperation mode /dialogue, retrieval, batch/ and of the language of accessing to network resources.
- Readiness, reliability and cost of the resource access.
- Supporting the user access to resources with the command HELP and the number and type of offerred DB and UP resources.

TOPOLOGY AND HARDWARE/SOFTWARE OF MSK AND ASK

Primary MSK communication nodes CCN are located in bigger university centres: Computing Centre of the Technical University of Wrocław in Wrocław, in Electronic Computing Technique Centre of the Silesian Technical University of Gliwice, and Computing Centre of the Institute of Computer Science at the Polish Academy of Sciences in Warsaw. The communication nodes of the network are minicomputers SM3/4 - equivalent to PDP 11/20 and microcomputers MERA 60 made in Poland. The software for the nodes was made the way not to derange the firmware solutions. The network Hosts /Hs/ are computers actually exploited in the time-sharing multiaccess mode in the universities. They are: ODRA 1305 /OM 256 Kwords, EDS 160 MB/, EC1032 /OM 1 MB, EDS 220 MB/, EC-1060 /OM 8 MB, EDS 920 MB/, EC-1040 /OM 0.5 - 1 MB, EDS - 200 MB/ and typical peripheral I/O devices. The ODRA-type Hs allow for parallel access to resources by 15 terminals during the response time required for the dialogue mode. The EC-1060, EC-1032, EC-1040-type Hs are allowed to work with 6 terminals for the response time shorter than 2 secs, and with 30 terminals for the response time of 2-30 secs. The EC-1060-type Hs permit to attain even a higher level of parallel terminals work. The Hs are linked with the communication nodes through front-end processors /FEPs/. This sort of connections causes that the Hs are relieved of execution of communication procedures and there is no need to interfere with operating systems of the Hs. For Hs as ODRA 1305 the computer ODRA 1325 /equivalent to ICL 1900/ Hs the FEP and for the EC-series Hs - the teleprocessing module EC 8371. The FEP software designed by the network design teams cooperates with H standard operating systems GEORGE3 + MOP and OS/MVT + TSO [8]. The terminals connected with the FEPs give communication as with a local H and its terminals as with the remote network H and its terminals.

The terminals cannot be connected with any H and its FEP but can be connected directly with the network through a proper concentrator of terminals for the service of individual or collective user groups /students/. Each terminal through the concentrator gives access to each H in the network and to any terminal joined with the FEP or concentrator. The concentrator of terminals in MSK has been implemented on the minicomputer SM4 /16 terminals/ with the multiplexer SMH-11p. Functions of the terminal concentrator may be also performed by the H ODRA 1325 and the teleprocessor EC 8371 /FEP/. The software of the concentrator functions does not differ from the

FEP software implemented on these processors, however then the FEP module assuring the communication with the H is not active.

The nodes in the network are connected one with another and with the FEP and the terminals concentrator through the communication line adapter /CLA/ ALS 11 which executes transformation of node interface signals /UNIBUS/ at pins S2 or V24 and reversely. It assures through the 1200/2400 b/s speed modems EC 8006 or EC 8013 the in-line synchronous duplex data transfer between the node processor and the telephone line. The node-to-node and node-to-FEP connections are made on hired /private/ four-wire telephone lines. The bit error rate measured in these lines at long distances /Warsaw-Wrocław/ amounts 10^{-6} - 10^{-5} for the transfer data of 9600 b/s.

The ASK topology is fit for the information infrastructure, H/S and telecommunication media available at the Warsaw academic environment [10]. It includes the need for offerring various universities and institutes the dispersed computing facilities of Hs EC-1040 and particularly of EC-1060 /operational memory of 8 MB and external disc memory of 920 MB/. The nodes of the network are based upon of the microcomputer MERA 60. The data transfer is realised at the speed of 48 Kb/s /by direct-current convertors/.

The MSK/ASK have a common communication node located in Warsaw on the minicomputer SM-4 /The Palace of Culture and Sciences/. The node allow presently the users of both the networks for inter-process communication at the level of the X-25 protocol. Regarding the specificity of network services the ASK was implemented with the basic software on the minicomputer SM-4 which performs the role of a RJE terminal. The terminal joined with the CN through the CLA ALS 11 executes batch processing functions by the H EC-1060. The ASK takes use of terminal concentrators with screen displays for the local dialogue mode and was made on the basis of the minicomputer SM-4 and of MERA 60-based-concentrators equipped with MERA 7900 screen displays - terminals.

PROTOCOLS AND FUNCTIONS

The significant problem to be solved at the first network design stage was a determination of standard protocols. It was difficult in so far as the process of international protocol standardization has not yet been completed. So those protocols were accepted which gained the international standard rank where only it was available. In layers where there were no international standards own solutions close to solutions applied in other networks or to subsequent standardization proposals [12] were used.

The physical layer makes possible to use real physical communication media. The unit of information transmitted through the medium is a bit. Protocols of any layer determine mechanical and electrical properties of hardware for a transparent transfer of the flow of bits between link stations preserving however the sequence of bits. The protocol X21 bis that uses the pins S2 defined in the CCITT V24 standard was accepted. The pin S2 makes possible to use known analog and direct-current modems working with rates from 50 b/s to 48 Kb/s.

The link layer contains procedures and functions for picking, holding and releasing the communication between stations. The unit of

information transferred by the link is a frame. Additional functions of this layer are to detect and correct errors that can arise in the physical layer and to control the flow of frames by adapting the transfer to receiver characteristics of the station within the network layer. The protocol X.25.2 in the LAP B version was assumed whereby stations at the link layer possess equal rights i.e. each of them can initiate and terminate the transfer. The purposefulness of selection of this protocol was confirmed by the fact that few /also in the GDR/ producers make integrated circuits adapted to hold the protocol.

The basic functions performed by the network layer are to select, hold and release network connections and replace through it network data units called packets between transport stations. The layer also performs functions necessary to mask variously in characteristics of various transfers. Services rendered to transport stations at the ends of a network connection are the same and a range of the services is negotiated during the connection initialization. The range of the services includes: error, reporting, holding the sequence of transferred packets and eliminating packets from the connection if any failure is detected. Also errors due to calls from the connectors layer are detected and corrected. The MSK utilizates the CCITT X.25.3 protocol for switched or direct logic connections which fulfills the mentioned functions and moreover notifies the transport station about errors being not corrected.

To transfer data between session stations the functions of selecting, holding and disconnecting transport connection are used. The transport connection quality is expressed in terms of a transport service i.e. the service rendered by the transport communication. The division of services into classes is designated by a subset of the following functions:

- mapping of transport addresses into network addresses,
- multiplexing transport communications,
- splitting of messages into blocks,
- transfer of fast data units,
- detection and correction of errors in the end-to-end relation,
- flow control.

The ISO DIS 8072 standard admits the application of 5 classes of services numbered from 0 do 4 in transport connections. The MSK uses the transport protocol defined by the DIS 8073 standard with classes 0 and 1 i.e. class 0 - Simple Class and class 1 - Basic Error Recovery Class. After the transport communication between transport stations is initiated communication conditions are negotiated i.e. one out of the service classes is selected. In class 1 it is possible to perform a normal or expedited exchange of data of errors signalized by the network layer and signalization of transport connection errors. Then a disconnection takes place. In class 0 it is possible to achieve a normal exchange of data, signalization of connection errors and then an unexplicit disconnection comes about. Transport stations in the MSK offer in principle class 0 and 1 services however there can be stations which render only class 0 services.

Since international standardization proposals for higher layers 5-7 differred significantly in subsequent versions it was determined to desist from the purpose of implementing independant protocols in session and presentation layers. The decision was made to base the two parallel protocols: for the virtual terminal and file transfer

on the transport protocol. Each of them performs the session and presentation functions. To file transfer a subset of the INWG86 protocol applied in the PSS network in England was used and for the virtual terminal an own protocol was developed.

The virtual terminal protocol applied in the MSK does not contain messages serving to start and terminate a cooperation of session stations but utilizes transport station services. So the type of the interaction terminal protocol is assumed to be present. After the transport connection take place the stage of parameter negotiation follows and next after their agreement - the stage of data exchange. At the negotiation stage messages are exchanged pointing to own parameters, questions about terminal parameters, demand for accepting parameters and confirming the reception of the parameters. The negotiated parameters are as follows:
- type of the code: EBCDIC or S16CCITT,
- impact of the interruption on disturbances in data output,
- type of the station: terminal or programme,
- way of editing of new line characters,
- mode of the terminal - programme cooperation,
- change in the transmission token,
- length of the message to be edited by the terminal.

The exchange of data may be carried out in one of the two modes: TWS - bidirectional synchronous /duplex/ or TWA - bidirectional alternate /semi-duplex/. Beside data the TWA mode also allow for exchange of the transmission token.

The file transfer protocol [12] used in the MSK is a modification of the INWG86 protocol version. File transfer stations make possible to transfer data bases between real H memories, between H memories and output devices such as line-printers, and between memories and job manipulation stations. Each performance of file transfer consists of three phases: negotiation, data transfer and termination. At the negotiation and termination phases the data transfer stations exchange between themselves commands, agree to values of attributes characterizing data bases and the way of the data exchange.

As distinct from the INWG86 protocol, the data transfer protocol implemented in the MSK does not allow for:
- defining real memory attributes /the way of data storing/,
- alterations in file editing control,
- data transfer control,
- transfer agreements in accord with properties of user groups /the group comprises users of the same type of H/.

Down to the network layer the protocols developed for the MSK are used in the ASK. At the moment an interprocess communication adapted to the operating system EC-1060 specificity is projected at the level of higher layers.

INFORMATION AND USER PROGRAMME RESOURCES

Beside H/S resources such as computing facilities, H memory capacity, the information /Data Bases - DB/ and user programme /UP/ resources determine the network service quality.

Simultaneously with the design activity on the network hardware and implementation of protocols the organisational and design work on

acquisition, selection and offering DB and UP resources in the MSK started [9], [11]. In order to provide this type of resources it turned out necessary to undertake actions aiming at the acquisition of new resources, updating of the existent resources and maintaining them in permanent exploitation.

The primary source of the new UP resources are user special programmes developed by specialistic groups in academic environments as well as those purchased abroad under scientific exchange rules. Whereas the source of information resources are bases of factographic and bibliographic data stored in universities by teaching and research centres and also these brought from abroad under scientific exchange programs and included in international information service. The UP and DB resources are implemented in the network in the form of programme libraries and data bases which are offerred to network users. The distribution of resource implementations and the mode to access to them requires a thourough analysis to eliminate unnecessary data transfers. To do it proper statistical techniques /e.g. file transfer journal/ are used in the analysis and evaluation. The problem of resources updating turned out very difficult to be solved when a resource maker is not an owner of technical media supporting the resource and is not directly linked with the network. The maker of any information and UP resource who passes the resource through H/S facilities of any computing centre joined to the network, passes also rights for its updating to the centre which becomes its exploiter. In this case the exploitation specifications should be extremely precise. That is why most often the resource is developed in specialistic institutes and then updated in batch processing. The problem can be best solved if such institutes are incorporated to the network as satellite installations.

The term "resource maintaining" is understood as actions aiming at proper service for resource users. The maintenance of resources in the network requires a permanent observation of resource exploitation and reactions to signals from users about the quality of the resources and the way to offer them. So it is necessary to keep statistics on their utilization and collecting /questionnaires/ remarks and conclusions from users.

The MSK/ASK information and UP resources are made by various universities throughout the country. The specialization of universities as based upon experience and scientific findings of the universities and moreover it aimed at the elimination of repetitions while making the information and UP resources.
The beneath listed universities were obliged to create and maintain the following resources:

1. Technical University of Wrocław
   - Bibliographic DB in engineering and exact sciences /excepting chemistry/,
   - factographic DBs in open-mining /lignite, ores of sulphur/,
   - programmes and DBs for planning and carrying out research experiments /in mechanics, chemistry/
2. Silesian Technical University in Gliwice
   - User and Tool programming for computer-aided designing and computer graphics,
   - factographic DBs and programmes for power engineering and metal industries
3. University of Warsaw
   - bibliographic DBs in humanities and social sciences,

- factographic DBs including data about standards, patents, translations, reports from foreign visits resulted in research papers

4. Nicolaus Coppernicus University in Toruń
   - user programmes for teaching and research in exact sciences /theoretical physics and chemistry/,
   - factographic DBs for theoretical physics and chemistry,
   - factographic DBs for selected areas of culture, sociology and geography
5. Rzeszow Technical University
   - user programmes for teaching and research in applied chemistry,
   - factographic DBs for applied chemistry
6. Academy of Mining of Metallurgy in Cracow
   - programmes for computer-aided designing in mining and metallurgy
7. Technical University of Poznań
   - programmes for computer-aided design of electrical devices and of automation
8. A.Mickiewicz University of Poznań and University of Wrocław
   - library of specialistic mathematical programmes
9. Technical University of Łódź
   - programmes for computer-aided designing of control and automation circuits,
   - factographic DB in textile industry.

The types and development directions of resources within the MSK by the universities are shown in fig. 1.
In conditions when not all the resources stored in the network cannot be simultaneously available because e.g. H memory shortages it is necessary to divide them into availability classes. In the MSK the following five availability classes are assumed:

1. Ever-available resources in the on-line mode,
2. Resources available after a certain time interval passes since the user called them,
3. Resources available during performance held in definite time intervals,
4. Resources available on a previous request,
5. Resources available on a previous request and confirmed to be available.

The utilization of the resources by users is then effective when the users know them. If the variety of resources in the network as opposed to resources disposed by a single host enlarges, the problem of announcing users about resources possessed by the network seems more and more important. Therefore various announcement forms on resources existent in isolated Hs and such as: user resources guides, training of potential users and publishing of booklets should be supplemented by automated systems of informing about resources. The MSK has a central information system for resources located in the whole network and local information systems for resources possessed by each H of the network. The resource information systems allow for getting on-line information about a state of the resource its availability e.g. location, availability class, availability time, utilization costs etc.

IMPLEMENTATION AND SETTING UP OF THE MSK

From the viewpoint of setting up and implementation the computer network comprises elements of various types such as: hardware,

operating /basic/ software, user programme and information /DB/ resources and administrative and technical personnel dispersed in terms of territory and interrelated. By the notion of the network implementation and setting up one should mean a complex process resulting in integration of all the elements into a network which offer its services to users. This complex integration process comprises the following:

- hardware setting and testing,
- operating software founding and setting,
- founding and setting of information and programme resources,
- organisation and training of operators and tenders,
- informing and training of users,
- conduction of a test exploitation,
- investigation of the network to check up its conformity and quality of services offerred to users.

For the MSK/ASK the following range of investigations is used:

1. Investigation of the functions:
   - entry of test tasks from terminals and its performance by addressed Hs,
   - editing of job results from the H and its realisation by other Hs,
   - transfer of files between Hs together with initiation of the transfer from the terminal and from the H,
   - transfer of control texts between terminals.
2. Definition of parameters and evaluation of their values
   - network mean time between failures in various exploitation conditions,
   - to/from communication node maximal transmission speed /for 9.6 and 48 Kb/s/,
   - communication node throuput counted in packets per second and packet-in-node delay time against a number of received packets /by the Intensive Testing Module/,
   - total subnetwork transit delay time /nodes, lines/,
   - readiness ratio as a quotient of the network readiness time to the total exploitation time /including also operators' proficiency in detecting and recovery of failure reasons/,
   - maximal transmission speed to/from FEPs,
   - number of retransmissions on intercity lines /that is an indirect measure of a group error rate/,
   - mean response time for the access to bibliographic DBs /working simultaneously with a definite number of terminals/,
   - mean time between failures for the whole network /together with its communication subnetwork/ in normal exploitation conditions,
   - mean service time for informing about resources, for counting service costs and for network statistics.

Generally speaking the subject of any implementation is a ready installation. But while implementing the pilot MSK the situation was different. The possessed equipment resources disabled designers to build the network that would provide full test exploitation conditions. So at the phase of constructing particular elements and mainly at the last test phase they faced the necessity to utilize installations of few centres and their personnel. Thus the implementation process coincided the last set up and test phase. As a date of the final setting up phase as well as the implementation commencement December of 1983 can be assumed when hired /private/ intercity lines between Wrocław, Warsaw and Gliwice were arranged. Then the situation looked as follows:

- nodes on SM-3 were tested in 70% of software and allowed to be exploited,
- front-end processors were tested in 80% of software and also permitted the exploitation,
- in the three centres: Wrocław, Warsaw and Gliwice there were H fit to work in the network,
- there were local informing systems in Hs in Wrocław and Gliwice,
- IDOL system was triggered in Wrocław to allow for access to the bibliographic DB.

The major task assumed for these conditions was to start, and implement the network understood as a process of integrating elements of the network, and also investigate the network work to recognize its exploitation characteristics and analyze services offerred to users. In the period of 1984/5 the exploitation investigations in the homogeneous and heterogenous version were carried out and then the installation and basic programmes of the network were tested and drawbacks removed. At the same time the investigation of exploitation properties was performed on the basis of IDOL services mainly in data retrieval from the bibliographic DB. The IDOL services were triggered three times a week in four-hour sessions. During these investigations:
- sessions were started 123 times, including 23 restarts after failures and 7 times the system failed in restart,
- total duration of the sessions took 25850 minutes,
- mean session duration took 275 minutes,
- 337 dialogues with users were performed,
- 2505 document descriptions were called from the DB,
- 2816 document descriptions were placed on after-sessions printouts,
- 2631 simple questions and 1053 complex questions were answered,
- one user dialogue averaged 8 simple questions, 3 complex questions, 7-8 descriptions transferred through a terminal and 8-9 descriptions on the after-session printout.

These investigations of the MSK exploitation allow for a formulation of these few remarks:
- reliability of the network is not a critical problem, the several hours network sessions lasted without break-downs,
- scarcity of available resources and redundancy of most of UP resources determined rather slight interest in network services,
- the IDOL /Information Database On-Line/ system under control of the operating system GEORGE3 allows 7 users to make use of it and the response time does not exceed few minutes,
- localization of break-downs and their removal takes more than half an hour what means that network operators and tenders are still low-qualified and diagnostic procedures do not work yet well,
- though transmissions were conducted on hired intercity lines, interruptions on these lines were one of main reasons for the network inefficiency.

The remarks allow the designers to draw the following conclusions:
- it is necessary to concentrate media to accelerate the implementation of further information resources in the network and provide efficient organisation to access to it,
- it is necessary to modernize the DB access mode to enlarge the number of simultaneous users,
- it is necessary to supplement node programmes with procedures that will improve the work done by operators and tenders.

ORGANISATION OF THE NETWORK EXPLOITATION

The basis for the MSK activity and exploitation is the agreement negotiated in March 1984 by five university environments in Łódź, Gliwice, Toruń, Warsaw and Wrocław on joint exploitation and utilization of network resources. The agreement participants are universities and institutes whose computer centers partake in setting up and exploitation of the MSK and ASK networks. The agreement is under the auspices of the early-founded Council for University Computer Centers.

To provide proper forms for the cooperation comprising starting, exploitation and improvement of network services the MSK and ASK Administrators as well as the MSK and ASK Computer Network Supervisory Body to fulfill advisory functions have been appointed [9] and [11].

The Supervisory Body is specialists' group whose composition is established by the Council for University Computer Centres in accord with parts of the agreement. Its tasks include the following:
- approval of exploitation conditions and giving opinions on accounting principles in the network,
- giving opinions on development plans,
- periodical evaluation of the network.

The MSK and ASK Administrators and Chief Network Operators elected one from each centre linked with the network enter the Executive Body. Its tasks are as follows:
- providing a proper reliability level,
- supervision of right exploitation cost registers and current accounting,
- supervision of network exploitation improvement,
- consultancy and inspection of readiness of new centres to be connected with the network,
- organisation of trainings for new users of the network,
- editing a network information bulletin,
- inspection of records and distribution of basic software assumed as the standard in the network.

The Network Administrator manages the functioning of the network, coordinates the performance of network services offerred by particular centres as well as exacts quality and the range of services rendered by owners of resources for needs of universities.
The owner of resources is here the computer centre or the institution collaborating with it that maintains and offers the resource to network users. Its duties comprise the following:
- maintaining resources in the network,
- transactions of deals with users for remote use of network resources,
- granting permissions for the use of resources and determining the resource protection level,
- collection of payments for the use of resources in accord with deals.

METHODICAL ASPECTS AND CONCLUSSIONS

1. The design and implementation studies of both the networks were conducted by a few research groups. Among them leading was a very numerous MSK design groups at the Technical University of Wrocław

whose task included a construction of the communication subnetwork and basic components of the MSK Hosts as e.g. communication nodes, FEPs for the homo- and heterogenous version, implementation of protocols up to the virtual terminal, file transfer, informing systems and network diagnostic system. The remaining groups dealt with the communication subnetwork and RJE terminals for the ASK Hosts, communication nodes for MERA 60 /University of Warsaw, and Silesian Technical University/ and terminal concentrators on SM-4 /Technical University of Poznań/. The parallelism of the so-widely planned projects pointed to the importance of a property substantial coordination of design solutions and exactness in formulating technical and exploitational /services/ specifications for components of the network.

2. Considering a larger functional flexibility the admittance of alternative solutions of some components as e.g. communication nodes /on the basis of SM-4 and MERA 60/, FEPs /on the basis of ODRA 1325, EC 8371 and MERA 400/, functionally differring versions of concentrators on the basis of SM, CLA, H programmes and systems informing about resources resulted in a bigger complexity of the start and implementation process. However it allowed for the step-by-step implementation closer to real conditions and for evaluation of network services from the viewpoint of hardware/software alterations [14], [15]. The important element for evaluation of dynamic properties of the network were diagnostic solutions assumed for it, techniques to control the access to equipment and programme resources, components to measure the efficiency of the network and communication services, of distributed processing and the access to DB [16]. It turned out to significant to abide consequently by accepted parameters. In the case of a necessary change in even one parameters if turned out important to proceed consequently with modification of remaining network parameters.

3. Together with the process of improving accepted solutions and network parameters, producers continue to modify typical installed network components in the area of firmware e.g. new types of operating systems etc. It impacts on the evolution of network parameters, on effective interactions of the components and the efficiency of improving network services [13].

4. Network designers /particularly those with big engineering experience/ face the problem of transforming their creative activity from traditional approaches fit to closed systems to new approaches adequate to distributed and open systems. It is also necessary to leave equipment-oriented solutions on behalf of system user-oriented solutions and focus on network services, creation of new DB and UP resources as well to integrate them in the form of knowledge bases. This kind of weakness manifests itself in the application of the experience in early seventies' network techniques by designers as well in wider applications of contemporary microprocessor technologies to network projects what would result in a jump of some recent technological evolution phases related to computer networks.

5. The inseparable element in constructing technical network components is a development of installed DB and UP resources and modernization of methods for the computer knowledge representation [14]. Since there is here a strict correlation of the development with the social efficiency of accessing to these resources.

6. The rich user software output possessed by academic environments is not applicable - in respect of its solutions - to direct implementations and offerring by the network /no proper classifications, modularity and descriptions how to use it and aid users/.

7. The role of network techniques consists not only in the integration and hire of territorialy dispersed resources. They are also the factor that sets in order and increase the software output efficiency. Besides they are stimulators improving the software quality, modernizing mental work methods of the man - the network user, changing thinking of computer specialists profesionalists of informatics - realizers of network solutions, of organizers of network services, changing the role of computer centres and their relations with makers of DB and UP resources and breaking with the wide isolation of many computer environments, groups and computer centres. Because the computer network as open systems belong to evolutional systems.

I want to acknowledge the cooperation in many years' realisation of the MSK project to all specialists. I express my respect for successful copying with many difficulties that appeared during the performance of this project. I appreciate co-authors of many publications devoted to network problems and lecturers from scientific schools "Computer Networks" [2]. This project has been sponsored by the Ministry Problem RI-14 financed, by the Ministry of Science, Higher Education and Technics and supported by Rectors of Technical University of Wrocław to who we owe the chance to develop the Project to the state that has been described in this paper.

REFERENCES:

[1] Bazewicz, M., Barchański, J.A., Preliminary project of the Interuniversity Computer Network - MSK /pilot stage/, Part I a.Part II. Reports 268 and 269, ICT Technical University of Wrocław, 1978 /in Pol./.

[2] Bazewicz, M. /Red./, Computer Network Properties and Functions, Part I: Protocols, Part II: Communication, Part III: Software, Technical University of Wrocław, Publications Serie: Biblioteka WASC, 1978-82 /in Pol./.

[3] Bazewicz, M., Time-sharing multiaccess computer systems, Państwowe Wydawnictwo Naukowe, Warszawa, 1977 /in Pol./.

[4] Bazewicz, M., Computer information systems design, Scientific Papers of the Institute of Technical Cybernetics of the Technical University of Wrocław, No 60, Monographs No 12, 1982 /in Pol./.

[5] Bazewicz, M., Zur Projektierung Informationsverarbeitender Systeme, Dissertation B, Technische Hochschule Otto von Guericke, Magdeburg, 1984; Wissenschaftliche Zeitschrift der TH Magdeburg, Heft 8, 1983.

[6] Bazewicz, M., Maroński, J., Mika, T., Computer Network for Universities and Institutes of Polish Academy of Sciences. Comm. No 40 of Technical University of Wrocław, 1977 /in Germ./.

[7] Bazewicz, M., Mika, T., Some Problems of Computer Network for Polish Scientific Centres. COMNET 77 Symposium, Budapest, October 1977.

[8] Bilski, E., Technical and functional project of the Interuniversity Computer Network /MSK/. Part 1. Technical resources. Report 15/82 ICT and CO Technical University of Wrocław, 1982 /in Pol./.

[9] Dębowy, J., Mika, T., Misiaszek, L., Organisational and functional project of the Interuniversity Computer Network - MSK, Part 4. Network Resource Administration, Reports 2/83 ISiTS Technical University of Wrocław, 1983 /in Pol./.

[10] Technical and functional project of the Academic Computer Network - ASK. Part 3. CPiZI and University of Warsaw, 1982 /in Pol./.

[11] Żurawski, B., Wiecaarkowski, K., Wiśniewski, J., Technical and functional project of the Interuniversity Computer Network - MSK. Part 2. Information and software resources. Reports SPR 28, 000 Nicolaus Copernicus University of Toruń, 1982 /in Pol./.

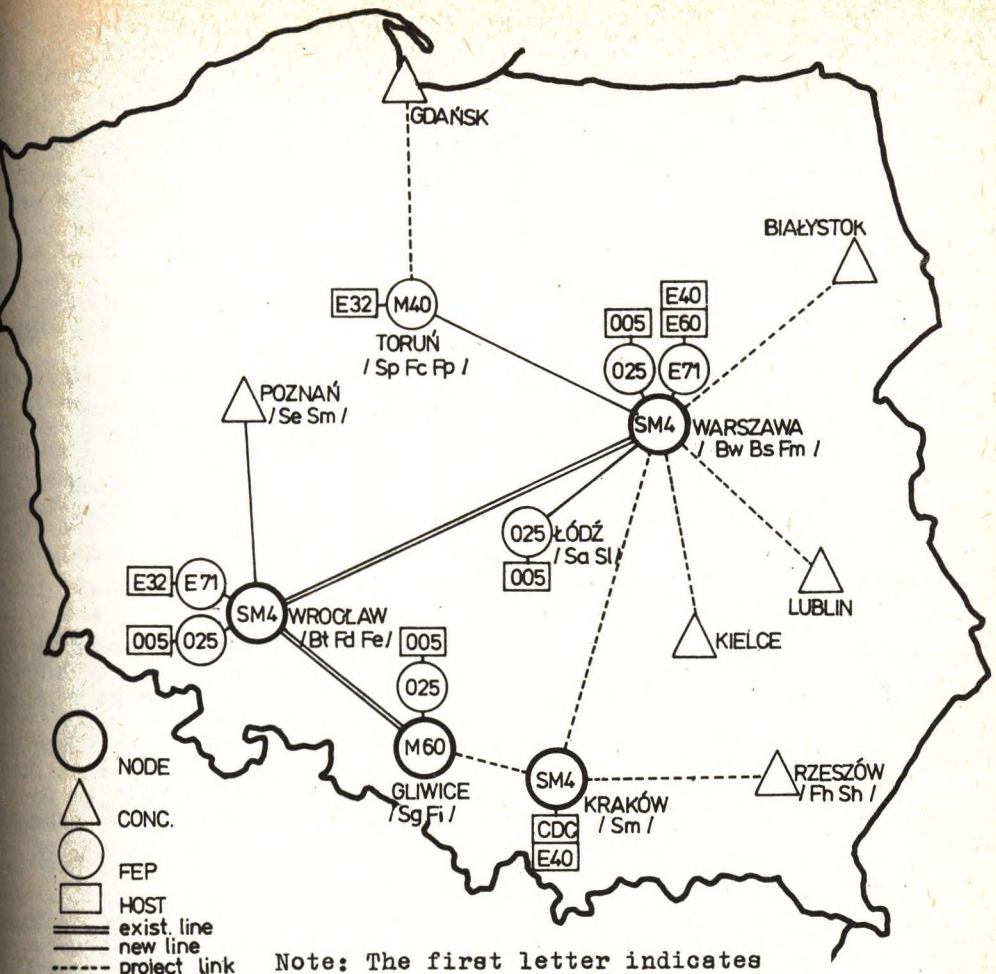
[12] Protocols in Interuniversity Computer Network - MSK, Edit. Bilski, E., Technical University of Wrocław, Publications, Serie: Biblioteka Informatyki Szkół Wyższych /in Polish, in editing/.

[13] Bazewicz, M., Architecture modelling of data processing systems, Reports of IIASA, Laxenburg-Vien, 1979.

[14] Bazewicz, M., Analysis of distributed data processing systems and their application parameters, Proceeding of Confer. "Počitačove Site'80", CSVTS, Bratislava, 1980.

[15] Bazewicz, M., Pužman, J., The role of user job handling process in the computer network design methods, COMNET'81, Budapest, 1981.

[16] Bazewicz, M., On methods of computer network performance evaluation, Proceedings of Confer. DPD'83, CSVTS, Karlove Vary, 1983.



Note: The first letter indicates the kind of the resources

Hardware/Software

005 - ODRA 1305
025 - ODRA 1325
032 - EC1032
040 - EC1040
060 - EC1060
071 - EC8371
SM4 - SM4
M60 - MERA60
M40 - MERA400
CDC - CDC6000

Information Resources

S - User Software
B - Bibliographical Database
F - Factographical Database

Sections

a - Chemistry
g - Computer Graphics
e - Electricity and Automation
a - Control and Automation
n - Mathematics
t - Technologies incl.Computer Science
s - Social Sciences
i - Industry
m - Mining
c - Culture
p - Physics and Chemistry
d - Management
k - Scientific Experiments
l - Textiles

Figure 1
MSK Network and Resources development

USER CHARACTERISTICS OF FILE AND JOB SERVICES

Roman KASZUBA
Jan KWIATKOWSKI

Computer Centre
Technical University of Wrocław
Wybrzeże Wyspiańskiego 27
50-370 Wrocław
Poland

The development of computer networks has create a new services offered to their user. The paper described the computing facilities of a computer network and a file/job transfer and manipulation services. The resourse consumption by file transfer is investigated and justification of optimization solution presented.

INTRODUCTION

This paper concerns file and job transfer services implemented in th Polish Interuniversity Computer Network (MSK). MSK is a wide area computer network, first of all used for information exchange betwee university computer centres and information retrieval, also for training, education and researches into computer network constructi The services offered by it are composed of general purpose faciliti and a data base access facility. The first include facilities for access to individual host computer services, for information transf and for running jobs composed of tasks executed simultaneously or in sequence on different computers. The network consists of host computers, front - end processors connecting host computers to a co munication subnetwork and handling terminals, terminal concentrator handling terminals and connecting them to a communication subnetwor and X.25 communication subnetwork. Host computers are of different computing power - from mainframes supporting multiaccess time - sha ring systems to minicomputers to be connected in the near future. Over the X.25 subnetwork access interface there are implemented: th ISO transport protocol [8], the MSK file transfer protocol [2], the MSK job transfer and manipulation protocol [1] and the MSK virtual terminal protocol [7]. The file and job protocols are somewhat modified British protocols: A Network Independent File Transfer pro tocol [9] and Job Transfer and Manipulation protocol [10]. The paper describes features of the general purpose facilities and give some remarks on their support.

USAGE OF BASIC FACILITIES

General purpose facilities enable user to access services of any computer in the network. He can access operating system/data base services by means of the virtual terminal service or transfer files and run jobs executed on many computers by means of the file/job transfer and manipulation service. That latest service consist of a file and job transfer facility, a remote file manipulation facili a remote job control and monitoring facility, and a job synchroni- zation facility. A user access the service through network job

control language (NJCL) commands, which constitute an extension to local job control language (JCL). The main features of NJCL commands is connected with the fact that almost each command execution involves data transmission over a network. Therefore the NJCL commands, interpreted during a job execution, only start appropriate processes which are executed in parallel with subsequent progress of that job execution (analogical situation can be found in a file access: program which has issued a file block/record read instruction can wait for transfer completion or proceed in parallel with it). This assures that a job issuing a NJCL command can proceed without unnecessary delay connected with data transmission. This NJCL command execution mode is very useful, particulary for interactive terminal users.

Considering that JCL and NJCL commands can be issued by a terminal user as well as by a process, and that each process can create new processes, a user has a powerful tool for generating a set of processes executed in parallel on many computers. To organize the processing of the set, which usualy contains mutually dependent processes, a user utilizes the job control and monitoring facility and the NJCL facility, described in [4]. THe set of processes is changing during execution because of subsequent user actions (process abortion or creation), natural process termination, creation of new processes by processes during execution, and network break-downs, which eliminate some processes (other are automatically retried after a network restart). The control considerations are complicated by a partial break-downs which suspend/eliminate only a subset of processes. So a user should take a special care when using the network general purpose facilities in full.

FILE/JOB TRANSFER AND MANIPULATION SERVICES

The implemented file/job transfer services can be described by a NJCL commands. These commands are added to local JCL commands in each host computer and can be issued in the same way (and context) as JCL commands. Their semantic and syntactic are identical irrespective of computer/operating system type in which they are implemented. The NJCL commands are:

TRANSFER
MODIFYORDER
WHATORDER
MAIL.

Each command may have parameters which specify the details of a command execution or refer to data accesses by that execution. The main command is the TRANSFER command, which is used to order services. Other commands are used to get information about ordered services or to modify previous orders. The purpose of the individual commands is as follows:

the TRANSFER command is used for:
- transfer of a file or a job or a document to be outputed between a couple of systems,
- transfer of a job and starting its execution,
- remote file manipulation create or delete a file, change an attribute ;

the MODIFYORDER command is used for cancelation/suspension/resumption of a TRANSFER command execution;
the WHATORDER command is used for interrogation of a TRANSFER command execution;
the MAIL command is used for listing reports from TRANSFER command

executions and replies given during user absence to MODIFYORDER and WHATORDER commands.

The TRANSFER command parameters refer to a source file and a destination file/job/document, authorization data, accounts, actions to be performed on a source/destination file. Parameters also determine conditions which must be fulfilled for command execution to be started (synchronization facility). Inclusion of parameters for the definition of data transformation (as described in [3]) is also considered.

The commands can be issued by an interactive terminal user or by any of created by him processes. Each command execution is treated as independent task, which is started if conditions stated by command parameters are fulfiled and which processing is under system scheduling. A user can log-in to the network, create some processes, and log-off. During his absence reports from his process executions are added to his mail-box, which he can investigate after subsequent log-in. He can also interrogate the process progres and make modifications of previous commands or create new additional processes. As processes can survive network break-downs he has the possibility of running a job composed of many processes by arbitrary long time.

RESOURCE UTILIZATION

One of the MSK goals posed a requirement of minimal resourse utilization. Connected with it was other which stated that the network service provision should not decrease the free-standing host computer computing power in a visible to users way. To this end the network services should not consume more than ten per cent of the host computing power. Therefore the services properties should allow implementations to be well fitted to properties of operating systems and such kind of external device as computer network, which is a very slow one. Likely, this requirements corresponds to the network service properties derived from considerations taking into account the duration of actions involving data transmission, mentioned above. For that reason, user orders implying transfer from tasks, which are processed independently. They are stored in a task queue in a system in which the orders have been issued. That system is responsible for their execution. The task executions are scheduled in order of better utilization of host/communication resources. Optimizing the resource consumption in MSK we have taken into account the transfer characteristics as well as possible saving algoritms, applicable at a host and subnetwork level.

Considering a single transfer one can find that a bulk data transfer occurs only during the data transfer phase, before which, during the negotiation phase, and after which, during the termination phase, a conversational mode trafic occurs. In that conversational mode trafic, the response time of computers is long, because in negotiation before a reply can be given the responder must not only analyse all requirements, but also make a lot of investigations in a filestore and a human operator action may be needed. In the termination phase operations connected with the commitment exchange can also proceed a long period. Then, in a file transfer, there are periods during which the trafic is very low and assigned to the established for transfer connection througput is not utilized in full. Moreover, during the data transfer phase, the trafic is one directional, so only a half of the connection througput is utilized. At the time

there are no posibilities to avoid that situation, as the subnetwork can not take into account periods of one directional trafic over connections. A temporary solution is that the subnetwork establishes more connections than it could fully support.

The optimization goal to make the trafic more symetric and to avoid periods of the conversational mode trafic could only be done in relation to data lines instead of connections on the host level. To accomplish these goals the system must be capable to perform a number of simultaneus transfers in a way which assures with considerable possibility the trafic symetrization and parallel occurence of both modes of trafic on the lines within the subnetwork. That yields the need for scheduling algorithm. We have adopted a very simple one, which is not ideal, but helps to a certain degree. The rule is: when the system chose a transfer task to be executed it search for a transfer to the same computer that one of the presently executed transfers is pointed, but in opposite direction.

The communication subnetwork carries two completely different kinds of trafic: the conversational mode trafic, mainly generated by terminal users accessing computing facilities and data bases, and the bulk transfer mode trafic generated by job/file transfer processes (outputs from data bases interrogation are transfered as files). Due to that and low throughput lines used it was decided to improve the subnetwork quality by separation of that two kinds of trafic [5, 6]. Such separation improves the subnetwork/network performance because for the conversational mode trafic the warranted round-trip delay is important while for the bulk transfer mode trafic the warranted throughput is. It was proposed to distinguish three classes of subnetwork users and to provide non-praemptive priorities at the entry to any communication link. The classes are: express letters (low warranted round-trip delay, window size w=1, the highest tariffs) , letters (medium warranted round-trip delay, w=1, medium tariffs) , and bulk data (no round-trip delay warranty, high throughput, low tariffs). Such separation is desiderable for terminal users to avoid unexpected delays, and it assures the high throughput for the bulk transfers.

CONCLUSIONS

The paper is concerned with the user characteristics of job/file transfer service in a computer network and the way in which information is conveyed over the network. It takes up the evaluation problem of networks utilization by various users and presents our aproach.

REFERENCES

[1] Huzar, Z. and Kwiatkowski, J., Job transfer and manipulation protocol for MSK, Technical Reports of Computer Centre Tech. Univ. of Wrocław, SPR 16/84, Wrocław 1984, Poland.

[2] Kaszuba, R., File transfer protocol for MSK, Technical Reports of Computer Centre Tech. Univ. of Wrocław, SPR 16/83,Wrocław 1983, Poland.

[3] Kaszuba, R. and Kwiatkowski, J., Experience in file and job transfer implementation using the British coloured books protocols, Procedings of International Conference IFIP/NETWORK 84, Sophia 1984, Bulgarie.

[4] Kaszuba, R. and Kwiatkowski, J., File and job transfer service access language, paper for IFIP TC.2 working conference " Future of command languages ", Rome, Italy, September 1985.
[5] Lewoc, J.B. and Bicz, T., Some remarks on OSI and X25 for communication networks operating at medium transfer rates, AFRICON 84, Tunis, 1984.
[6] Lewoc, J.B. and Bieleninik, E. and Stanisz, A., Influence of bulk transfer and interrogate mode trafic separation on performance of computer network, Procedings of International Conference IFIP/NETWORK 84, Sophia 1984, Bulgarie.
[7] Wietrzych, J., Virtual terminal protocol for MSK, Technical Reports, Institute of Tech. Cybernetics of Tech. Univ. of Wrocław SPR 21/83, Wrocław 1983, Poland.
[8] ISO, Information Processing Systems-Open Systems Interconnection Connection oriented transport protocol specification, DP 8073, March 1983.
[9] High Level Protocol Group and revised by File Transfer Protocol Implementors Group of the DCPU, A network independent file transfer protocol, FTP-B 80, February 1981.
[10] The JFT Working Party of the DCPU, A network independent job transfer and manipulation protocol, DCPU/JTMP 81, September 198

STRATEGY FOR CORPORATE COMMUNICATION NETWORKS WITHIN PHILIPS

Bert J. Boutmy

Director of Corporate Communications
Philips International Eindhoven
The Netherlands

ABSTRACT The rapidly growing internal need for communications within Philips makes the company more and more dependent on cost effective communication networks and services. Effective business communication networks have become a Corporate asset and are of strategic importance for the company.
This paper describes how Philips has organized and implemented its internal communication services for mail, voice, message, computer, data and images in an integrated manner.

EFFICIENT BUSINESS COMMMUNICATIONS PROVIDE A COMPANY WITH A STRATEGIC EDGE OVER ITS COMPETITION

This has always been so, but communications have become faster and more effective. Timely knowledge of technological and economic developments around the world has become of prime importance. This is especially true for Philips, which is active in more than 60 countries and employs some 350.000 people.
Strategic decisions and methodology of working are influenced by technological, economical and political events worldwide. The company with the capability to anticipate and react to these developments by using flexible communication facilities remains ahead of its competition.

Philips has production spread over more than 400 factories, research and development spread around the world and marketing organizations responsible for sales of over 53.000 million Guilders per year. The amount of information interchanged within Philips is large and of vital importance to the efficient operation of the organization and the quality of its products and services. This all requires an efficient Coporate Communications network which is reliable and, above all, secure. Timely interchange of information between the various Philips partners can reduce costs substantially.
Effective business communication is a corporate asset. A strategic weapon. And its importance is increasing rapidly.

OBJECTIVES AND ORGANIZATION OF THE CORPORATE COMMUNICATIONS DEPARTMENT

For a long time existing services for mail, telephony, telegraphy and data transmissions were scattered over various departments.

The penetration of computer technology in the various communication service areas, as well as the sharing of resources, set a trend towards integrated networks.

Corporate networks demand a strong organization on an international level. A communications network must be operated as a cost-viable service towards the enduser. To do this it requires a well planned infrastructure.

It was for these reasons that Philips decided to create one Corporate Communications Department with the objective of providing worldwide economic, efficient and reliable communication services for mail, voice, message, computerdata and image.

Corporate Communications is responsible for the communication policy in the company;
- specifies requirements
- buys services from the PTT
- operates the international networks
- establishes and controls the budgets

A Communications System Group supports the operational groups with among others:
- the analysis of requirements
- the setting up of specifications
- system evaluation
- implementation

The International Co-ordination function deals with the national organizations on budgets and administrative matters.

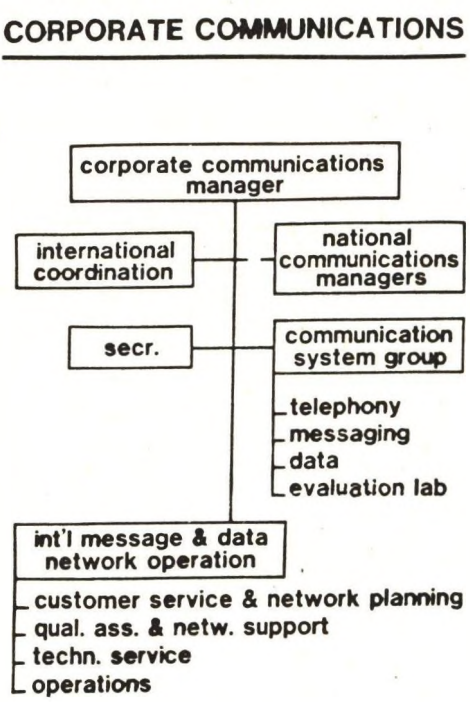


Fig. 1 Organization Chart Corporate Communications Department

In the various national organizations, National Communication Managers have been appointed, having similar tasks to those of the Corporate Communication Manager for their countries respectively. They report to the Corporate Communication Manager for international communication.

MAIL

Within Philips, traditional mail amounts to thousands of tons per year, to be sorted and delivered in the most effective manner.
Although electronic mail is gradually making an impact on business, the traditional form of mail will never disappear. There will always be parcels, trade magazines, circulars and other forms of printed matter, which will need to be sent and received. Savings have been achieved at the corporate centre in Eindhoven by limiting the number of envelope types used, sorting outgoing mail according to post code and introducing Communication Centres for integrated mailing services, including electronic mail, such as facsimile and text messages.
By coordination of logistics with various transportation agencies, an international mail service network is being maintained between Philips locations in Western Europe and the rest of the world. Courier services are used for local, national and international mail, where normal postal services are not satisfactory.

Communication centres

The Communication Centres are used as integrated services distribution points, handling all types of mail and messages, electronic and otherwise, including facsimile. The popularity of facsimile has grown over the last few years, mainly due to improvements made in quality and compatibility between differing types of machines. There are now hundreds of facsimile machines installed throughout Philips and the installed base is growing at an ever increasing rate. This has been stimulated by international standardization and the availability of Philips' own international facsimile directory.
The formation of Communication Centres by handling both 'traditional' and electronic mail has reduced both costs and delivery times.
It is anticipated that further reductions, especially with respect to external mail, can still be made.

TELEPHONY

The Central Telephone Department is operating 8 fully interconnected private exchanges with 20.000 subscribers in the Eindhoven area alone (see fig. 2).
The network handles an average of 140.000 calls per day. Most of them are desk-to-desk calls without operator intervention. During one working week of five days, the 20.000 extensions make as many calls as a small city of 20.000 inhabitants makes in two months. There are 65 telephone extensions per hundred employees, and the number of subscribers continues to grow at a rate of some 5% per year. The statistics given for Eindhoven are typical for the whole of Philips worldwide, indicating the importance of an efficient national and international telephone network service.

Philips possesses its own cable network in Eindhoven. Other locations in the Netherlands are connected via 150 leased PTT lines to the Eindhoven centre, and leased lines for international voice traffic are increasing rapidly.

The telephone network is used for many other purposes in addition to speech. Facsimile, messages, teletex, videotex and data transmissions are also sent over the telephone network, sometimes simultaneously with speech. Telephone extensions are used to connect terminals to mailboxes for electronic mail, access being granted by password control.

The vast number of calls made per day means that much time is spent on the telephone. Any facilities that reduce this time will increase office productivity and thereby decrease the hidden costs of telephone calls.
Faster connection, less operator intervention, improved operator facilities and higher external call connection success rate, all contribute towards a better service for our customers as well as internal efficiency.

Facilities such as Abbreviated Dialling for frequency called numbers, Auto-Ringback, for automatic connection to an engaged extension when both the caller and called party are free, Follow-me, for diverting calls to a temporary destination, when working there, instead of to the listed extension number; all increased office productivity.

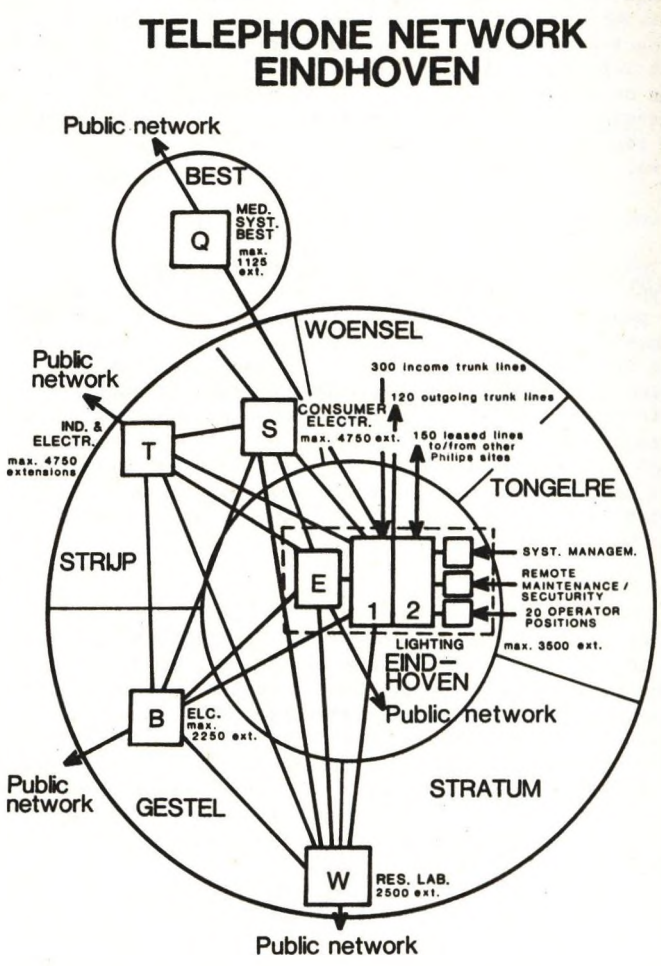


Fig. 2 Telephone network

In addition to the time-saving and convenience facilities, improved network management facilities make telephone operations much more efficient. Renumbering of telephone extensions after re-housing is accomplished in a matter of seconds, so that users are not cut-off from their communications life-line. Also, more accurate cost allocation for telephone usage can be made and statistics extracted, which enable network configuration planning to be implemented in a more efficient manner.

MESSAGING

Philips operates one of the largest private networks in the world, enabling a message to be sent between virtually every Philips employee. Covering the whole world, via private transmission media, it has more than 50 switches. Five international switching centres in Eindhoven, Paris, london, New York and Hong Kong are used as main routing nodes. The Eindhoven centre alone handles more than 40.000, 800 character average length, messages per day. Interfaces to the public telex network allow messages to be exchanged with third parties also. An international directory is available containing every Philips department messaging address. The naming and addressing scheme implemented provides automatic routing within the network. In addition to personal messages, the message network is also used for exchange of administrative and logistic information, which requires computer processing. A very simplified picture of the present network is shown in fig. 3.

The network was one fo the first to employ store-and-forward techniques for interconnection between its private switches and with public telex services, making efficient use of communication resources.

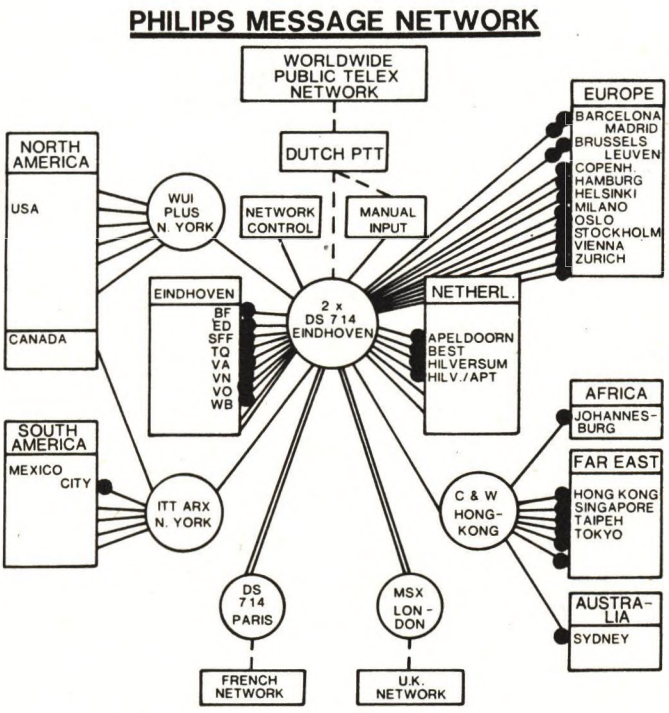


Fig. 3 Philips Message Network

System Enhancements in Communication Centres

Recent improvements on the message switching network are:
- faster and more realiable transport of messages (error free, complete)
- message submission and delivery directly to and from the offices
- connection of data-oriented terminals, communicating wordprocessors etc.
- extended character set
- connection to computer mainframes

These improvements typically indicate a development towards smaller machines and distributed message switching. The machine which fits in this development is the DSX40 office switch for text and telex traffic. It supports a variety of asynchronous circuits as well as teleprinters, data terminals and communicating wordprocessors.

In the Eindhoven area, but also in the National Organizations, the network has now been extended with these smaller type switches (DSX40; placed in the earlier mentioned communication centres), enabling connection of wordprocessing equipment as well as simple teleprinters. These terminals are placed directly into the offices. Messages can then be sent or received by the secretaries into personal mailboxes, avoiding mistakes by transcription and reducing delivery time considerably (see fig. 4).
Lateron connection of facsimile equipment is planned.

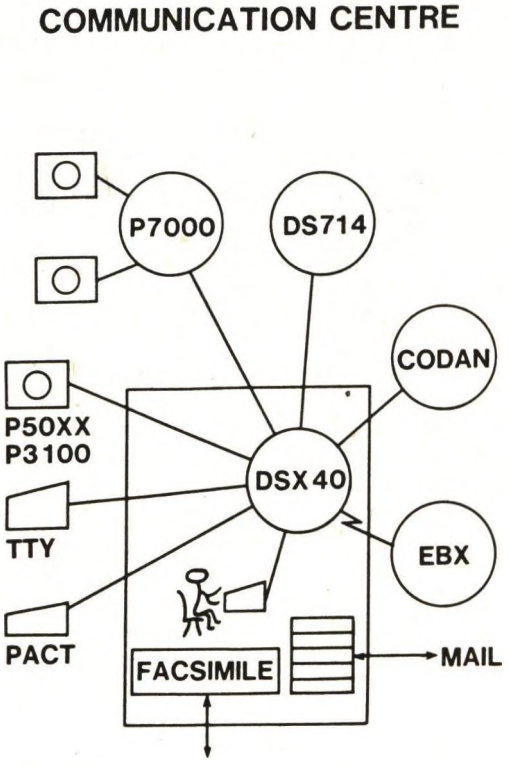


Fig. 4 Connection DSX40 in Communication Centre to Network

Siting of these switches in Communication Centres provides the added advantage of integrated control and support facilities for mail, facsimile and electronic messaging. In this way transitions between regular mail and electronic mail can be organized most efficiently. For example, electronic mail destined for non-terminal addressees can be dealt with, using the regular postal delivery available in the Communication Centre for the local area served.

COMPUTER DATA COMMUNICATIONS

During the past decade it was not only the telex message traffic which went through a spectacular growth. Within Philips also (computer) data communications spread its influence, which resulted in a variety of dedicated networks, consisting of leased lines, dial up connections, multiplexers etc.

However, the increase of leased line networks caused serious problems in network control and flexibility.

Therefore, in the seventies a project was started for a Corporate Data Network with the following main objectives:
- for international traffic
- sharing with other traffic, i.e. telegraphy, electronic mail, facsimile, digital voice
- limited access points per country
- first implementation for Belgium, France, Germany, Italy, the Netherlands, United Kingdom
- further extensions for other European countries and for intercontinental links

A set of requirements was defined with respect to the various aspects as shown below: (these requirements also formed the basis for the Philips Sohpomation network architecture)

General
- transparency
- connectability to other networks
- easy extensions
- independent network control
- multiple routing
- traffic measurement

End-user
- security
- general access
- types of service
- accounting

Environmental constraints
- existing hard-/software
- existing network applications
- conform to standards (CCITT, ISO etc.)

Performance
- availability
- reliability
- delays
- throughput

Network control

The Corporate Data Network (CODAN) is now one of the most advanced worldwide data networks which interlinks, via private lines, various national Philips data

networks and allows communications between computer systems and access from many thousands of authorised terminals.
Connections have been created to public data and telephone networks, allowing their facilities to be used in cases where dedicated data lines would be uneconomical.
The network contains some 30 nodes and offers and integrated 'open' communication service. Only an open network architecture allows the variety of applications and connection of equipment from different manufacturers.
A tremendous increase in usage of CODAN has been seen for many years, with traffic almost doubling every year.

CODAN, with nodes all over the world, is managed and kept-to-date centrally in order to guarantee network availability and service. Continuous adaptations of its capacity and transport facilities are made to meet the demand of the changing requirements from users. Also the CODAN network allows proper accounting and cost-allocation to users, which is necessary in its multi-discipline utilization.

The CODAN network in the European area is shown in figure 5. Moreover remote switching centres have been established in the Far East and the USA. The network has three gateway connections to public Tymnet, one in Brussels, one in California and one in London.

The CODAN network architecture is used to advantage for messaging.
Part of the message network traffic is now conducted over the CODAN network by connecting the DS714 message switch computer in Eindhoven to CODAN and to install DSX40 equipment, also connected to CODAN in the various countries. All defacto standards from IBM and DEC are supported by CODAN, as well as the OSI standards like X.25 etc.

The network is a packet switched session-oriented network, which does not perform any processing on the user data, although it is able to convert data link procedures and character codes. From the outside CODAN looks like a simple connect-through network. A user may connect his terminal to the network (dialled or permanent) and enter a username, password and host destination code. If he is authorized, the user obtains a virtual circuit through the network to its destination and from that moment on it looks to the user as if he were directly connected to the destination.

CORPORATE DATA NETWORK IN EUROPA

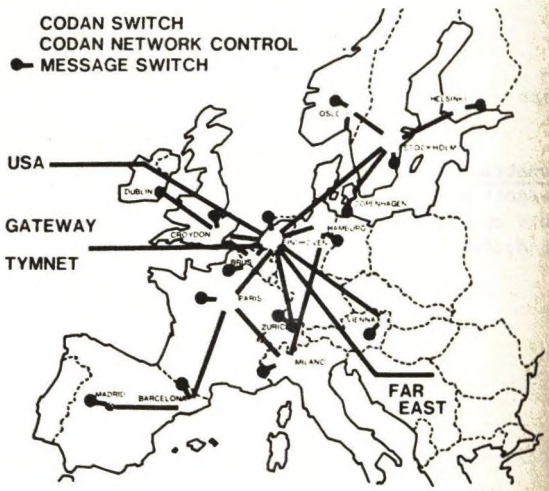


Fig. 5

Applications

The implementation of CODAN has contributed to much greater flexibility in use of computer applications, since differing types of computers have been interconnected. As a result of the implementation of CODAN and parallel with its growth, many applications in the fields of development, production, marketing and administration have been realized contributing to an increasing overall efficiency within Philips worldwide.

CONCLUSIONS

In the few years that the Corporate Communications organization exists within Philips it has been proven very successful.

Assessments of communication costs were made, tariff policies were made, communication budgets were put under control and plans for improvement or introduction of new services were established.

The international message network based on store and forward message switching principles, implemented on DS714 equipment, was merged with the data network based on packet switching principles.
Coupling to public national data networks has been realized.

For local area networks the introduction of communication centres equiped with DSX40 switches has been realized, combining floor services for message, data and mail (via communicating wordprocessors).
In the near future new public telecommunication services will become available. Especially ISDN, teletex and telefax will be of much interest to the company. Means and methods must be found to introduce these new public services into the existing private network services.

The integrated message and data network, together with the communication centres, form at the moment the communcation network infrastructure for worldwide data transfer within the company.

Architecture and Services of the Computer Network DELTA and Experiences with the System

E. Junker, H.W. Meier

Institute of Informatics and Computing Technique
Academy of Sciences of GDR

The computer network DELTA was developed in a national cooperation and is in operation since 1981. The paper deals with aspects of the DELTA model and of its services. Furthermore, there are given some experiences of the application of DELTA.

INTRODUCTION

In the beginning of the computer technique the efforts were concentrated upon the development of only one function of the computer, namely the processing function. Today, a broad spectrum of computer exists with a large range of computing power. With the up-coming of mass storage devices a second, the storage function was developed intensively. By solving the problem of cooperation of computer processes running in geographically distributed computer resources, the communication function of computer systems was lifted up to a higher level. Especially as a result of the computer network technology, the computer-aided telecommunication was developed in different forms as innovative applications.

During more than one decade in many countries a great deal of research work is going on intensively for the elaboration and analysis of computer networks. The fundamental idea of the computer network concept concists in that to give the user an access not only to an isolated computer system, but to a set of computer resources distributed over a set of non-locally distributed computer centres. Because of the integration of the above mentioned utilization functions into the network the user gets qualitative new possibilities for his applications.

Today, many computer networks are in operation in the world. Starting with closed systems like ARPANET /1/ or CYCLADES /2/ the development went to the evaluation of public data networks like TRANSPAC /3/ and to the setting-up of open systems basing upon the Reference Model of Open Systems Architecture /4/.

Taking into account the requirements for data and text communication as well as for information and information processing services, different groups of the Academy of Sciences of the GDR and the Technical University of Dresden developed the computer network model DELTA as a general concept of a global distributed computer system. The implementation of this logical model was carried out in several development stages capable for operating. The first stage of the network DELTA has been in operation in the beginning of 1981.

The present paper gives a short introduction into the DELTA model and describes the present stage of the network. Furthermore, it deals with several aspects of the DELTA services and of application experiences.

THE COMPUTER NETWORK DELTA

Main goals for the computer network DELTA are:

- To gain scientific experiences in the branch of computer networking
- To gather know-how in management, operating, and utilization of computer networks
- To improve the computing basis of the Academy of Sciences, the Academy of Agricultural Sciences, and the universities of the GDR by the integration of computer resources into a national computer network for research and education.

The general aspects of the system concept of DELTA were described in /5/. The DELTA concept is based upon a centralized control of hierarchically layered communication services. It is computer-independent and involves the following subsystems:

- The communication system KOMET is realized as a packet switching data network. It is based upon mini computers of the type KRS 4201, micro computers, and communication lines of 1.2, 2.4, 9.6, and 48 kbit/s.
- Different processing systems of the type BESM6 and EC computers are acting as hosts.
- Different types of terminal systems can be used upon the basis of communication lines of the manual switched data network with speeds of 1.2 and 2.4 kbit/s.

General aspects of the communication architecture of DELTA are published in /6/. Each layer of the communication architecture of DELTA makes a communication service available for the next higher layer and uses services of the lower layer.

The communication architecture of DELTA was elaborated parallely, but independently of the activities of the ISO for the Reference Model of Open Systems Interconnection. Nevertheless, because of the analogy in the design principles it is possible to map the communication architecture of DELTA to the OSI model and to compare both models.

Fig. 1 shows the interconnection of the subsystems of DELTA.

Today, there are integrated into DELTA the following hosts systems:

- 4 BESM6
- 1 EC 1055
- 1 EC 1040
- 2 EC 1035
- 2 EC 1022.

Furthermore, the interconnection of additional 5 EC computers is in preparation.

Corresponding to the increasing importance of computer-aided communication, additional to the above mentioned Operator Communication several other telecommunication services are implemented within the framework of DELTA.

Interactive communication is possible between:

- Terminals
- Terminals and the Terminal Interface Processor which has the function of a front-end computer of the terminal system to the processing systems
- Processing systems
- Communication computers of KOMET.

Storage-aided forms of communication services, that means MAILBOX services, exist in the framework of:

- The batch terminal system
- The processing systems.

The different communication means are utilized by computer operators and system engineers as well as by different user groups. Examples are:

- Planning of the cooperation of terminals, e.g. for user-to-user communication
- Supporting the operation of the local and the network production, e.g. by time-controlled instructions for operators, by the standard MAILBOX "OPERATOR" for general messages to operators, or by the HELP information service
- Communication and message distribution between users
- Supporting the cooperation of different user groups in development and research tasks.

EXPERIENCES

Upon the basis of the operation and utilization of DELTA the following technological conclusions can be drawn:

By increasing of the system as a result of the growth of the number of integrated components and of the degree of complexity the probability for a break-down of the system is increased. Therefore, it is necessary to have a high reliability and stability for all components.

In order to guarantee a continuous operation of such a complex system, it is necessary to install restart possibilities at many points of the Remote Job Processing system, and that especially in the phases of the cooperation of remote processes.

The coordination of the cooperation between different computer centres requires high efficient and reliable means for the man-man communication.

The introduction of a network application is essentially supported by their accordance to the local job processing. The activation of network functions can be organized by a small number of simple, but powerful additional functions. The existent operating systems and existent user software can be used without any changes.

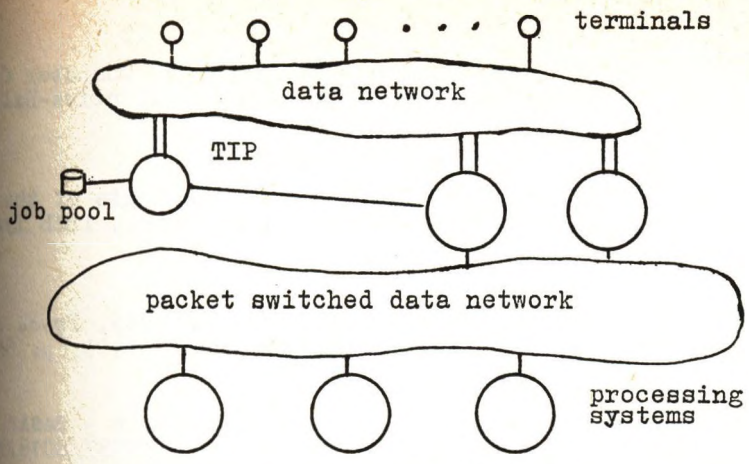


Fig. 1: Interconnection of DELTA subsystems

In order to study problems of the international data communication for the support of the international cooperation of users, there has been set up one node of the communication system of DELTA at Prague /7/. By the help of this node there was studied the system behavior of KOMET very intensively. Several EC computers will be interconnected to this Prague node during the next time.

SERVICES OF DELTA

During the practical operation of DELTA the following services turned out as dominating:

- Remote Job Processing
- File Transfer
- Telecommunication means.

The Remote Job Processing takes over the transmission of jobs from any batch terminal to any remote processing system or from a processing system working as a logical terminal to any processing system, the evalution of jobs, and the transmission of the results to a given output computer or terminal.

By the help of the File Transfer it is possible to exchange files between any processing systems.

The telecommunication service Operator Communication is used for organizing special forms of cooperations between computer centres, e.g. in the case of load sharing.

By the help of these basic service are realized:

- Data sharing
- Load sharing
- Resource sharing
- Reliability compound.

REFERENCES

/1/ L. G. Roberts, B. Wessler, The ARPA Network, Computer Communications Networks, Abramson a. Kuo, Eds., Prentice-Hall, Inc., 1973, p. 485

/2/ L. Pouzin, Presentation and Major Design Aspects of the CYCLADES Computer Network, 3rd Data Com. Symp., IEEE Tympa, Nov. 1973, SCH 511

/3/ M. Monet, The TRANSPAC Network Services in 1985, Proc. of the 6th Int. Conf. on Comp. Comm., London, Sept. 1980, p. 68

/4/ Data Processing - Open Systems Interconnections - Basic Reference Model. Draft Proposal ISO/DP 7498 (ISO/TC97/SC16/N719)

/5/ D. Carl, W. Dames, D. Hammer, V. Heymer, G. Hoffmann, H. W. Meier, C. Sattler, I. Wende, The System Concept of the Computer Network DELTA, Proc. of the 4th Int. Conf. on Comp. Comm., Kyoto, Sept. 1978, p. 703

/6/ D. Carl, W. Dames, D. Hammer, A. Hennecke, V. Heymer, G. Hoffmann, H. W. Meier, C. Sattler, R. Sommerfeld, I. Wende, The Architecture of the Communication Services Hierarchy of the Computer Network DELTA and Experiences in its Implementation COMNET'81, Budapest, May 1981, Conf. Proc., North-Holland Publ. Comp., Amsterdam 1981

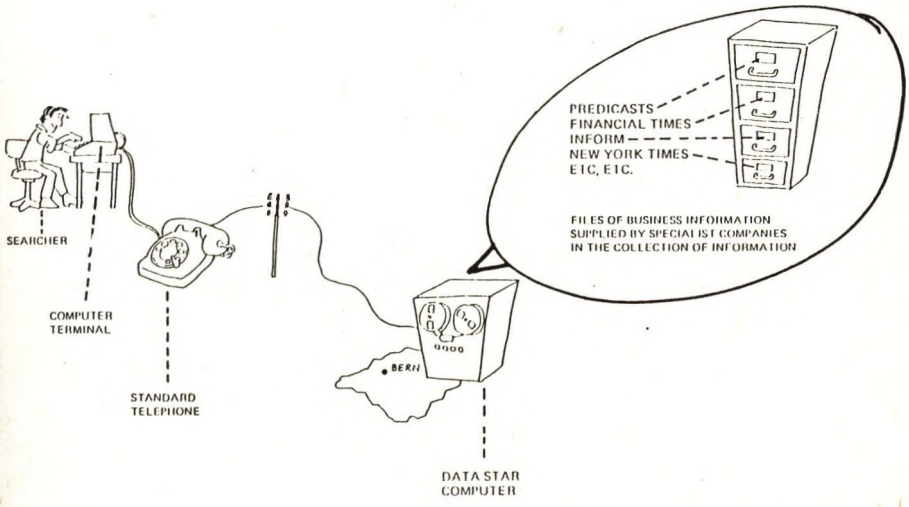
/7/ H.-M. Adler, D. Hammer, H. W. Meier, R. Sommerfeld, M. Novak, Datenkommunikation Berlin-Prag, rechentechnik/datenverarbeitung 20 (1983) 6, S. 19

HOW TO ACCESS DATA-STAR

Levente J. Varszegi

Head of Developments Telecommunications
Radio-Suisse Ltd. Berne
Switzerland

Data-Star is a data base service with millions of documents. An electronic information library for industry and science. The enormous volume is housed in mass stores at the operational centre of Radio-Suisse Ltd. in Berne. The user can call up the information interactively via the public telex, telephone and data neworks (X. 25). The access of the networks to DATA-STAR is then made automatically thanks to front end processors, and is continuously under control.



The System!

1. The company behind DATA-STAR

DATA-STAR is the brand name of an on-line data base service.

The DATA-STAR data base service was introduced in 1981 by Radio-Suisse Ltd. in Berne, and is today Europe's leading on-line data base.

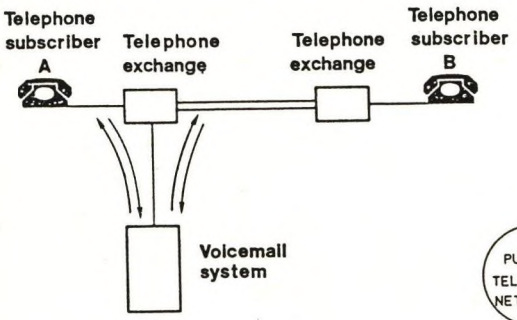
Radio-Suisse Ltd., the owner of this source of data and information indispensable for the society of the future, can look back on 60 years of experience in the operation of intercontinental telecommunication services. Radio-Suisse Ltd. has always been able to apply the most modern technologies in a competent and profitable manner in the organization of its services.

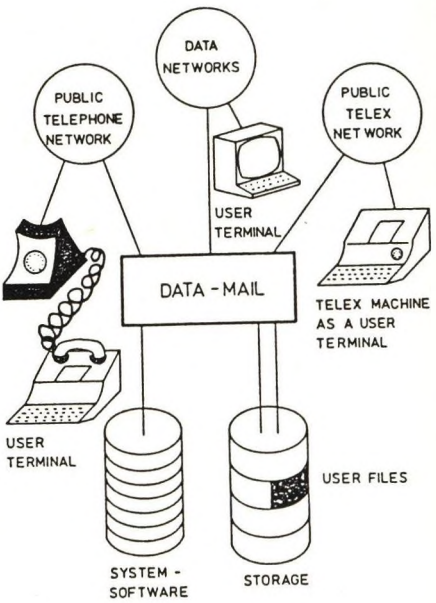
In recent years, Radio-Suisse Ltd. was the first firm in Europe to come onto the market with a revolutionary fully-electronic communication service. Both written an oral communications from person to person can be filed and polled in the central computer in Berne, completely independent of time and place.

Here the transport network used for these communications is the world-wide public telephone dialling network, mainly to access the public packet switched network.

The persons addressed can be reached anywhere in Europe (range about 2,000 km) with the aid of a personal paging service, the so-called VIP Line.

The service for written communications is called DATA-MAIL and that for oral messages VOICE-MAIL.





In the context of another modern service, concerning leased channels service, the specific communications need of international concerns located in Switzerland with worldwide branches can be individually, rationally and reliably covered.

These market services are supplemented by consulting activities and implementation of projects in the field of telecommunication and data processing in Switzerland and abroad.

The operational and technological expercience they have acquired permits our engineers to develop innovative special equipments not available on the market. These computer-aided special equipments are used primarily in our own operations, but since they fill a gap in the market, they can naturally also be produced and sold for the market. These are in particular applications for office communication, message transfer and data bases.

Radio-Suisse Ltd., which is behind the DATA-STAR service, is active not only in telecommunication and telematic. It is responsible, on behalf of the Federation, for the organization and execution of air traffic control in all Switzerland's airports. Its employees guarantee a safe, orderly and fluid handling of air traffic and are in charge of the ground installations and landing systems necessary for this purpose.

With this background: experience in the telecommunications, data telematic and data processing, continually seeking presence on the market and diversification, Radio-Suisse Ltd. could not fail to recognize the emergent need of the market for the timely acquisition of up-to-the-minute useful information.

It did recognize this need, and thus DATA-STAR was born.

2. WHAT IS DATA-STAR?

2.1. Who uses data bases and when?

Today's rapidly changing market situation, characterized by fierce competition, allows only those enterprises to survive which can gain competitive advantages for themselves.

One of the most important competitive advantages consists in the timely acquisition of up-to-the-minute, reliable and useful information.

As an example of how useful timely and comprehensive information can be, we may mention that German firms annually vainly invest more than a thousand million DM in developments which have already been taken up by their competitors. Losses of this extent are already of significance for the economy as a whole.

Taking these market needs into account, the first data bases appeared in the 1970s in the USA, where still today 75 % of the world market volume is supplied. Radio-Suisse Ltd. offered from the start in Switzerland and in Europe access to the American data bases by a rapid data transmission service (DATAC).

These are basically two kinds of data bases: text databases and fact databases.

The users of data bases are recruited mainly from industry, research and development, and marketing.

The sectors most in demand are as follows:

Industry	50 %
Tecnical	15 %
Biomedicine and chemistry	15 %
Others	20 %

2.2. Product definition

DATA-STAR is the designation of the data base service of Radio-Suisse Ltd., and represents a library of information built up on the principle of computer technique.

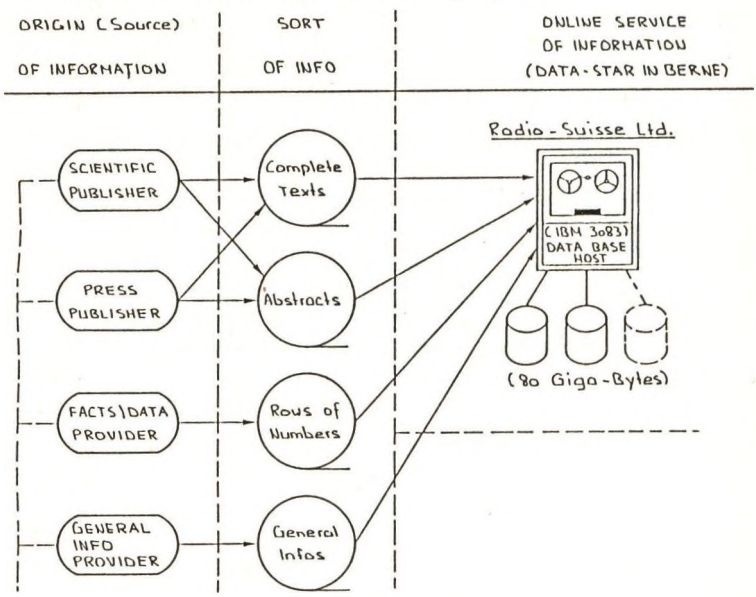
This electronic library today comprises more that 30 million documents, which are dialy added to or renewed.

The information gathered together comes from publishing houses all over the world.

Scientific articles appear mainly in books and periodicals.

Economic information is obtained mainly via the press (periodicals and newspapers).

So-called information suppliers now dial with the collection, classification and supply of abstracts and full texts and the registration of (factual) data and their storage on magnetic tapes (see diagram 1).



The information thus prepared in stored in an IBM computer in Berne. A leading programme developed by Radio-Suisse Ltd. allows the data formats as supplied to be rapidly converted into the form suitable for the computer.

The storage capacity of the data base installation amounts at the present time to 80 gigabytes.

This means:

80,000,000,000	characters (letters and figures)
	or
80,000	books of 500 pages and 2,000 characters per page
	or
35,000,000	literature citations

The market services offered by DATA-STAR at the present time cover the following partial markets/specialist sectors:

- Biology, medicine and chemistry
  (e.g. chemical abstracts)
- Technical
  (e.g. INSPEC, data base of the Institution of Electrical Engineers, London)
- Economic
  (e.g. Predicast, Dow jones).

A complete list of the data bases offered, arranged according to specialist sectors will be found in Annex A.

These data bases supply information in the form of:

- bibliographies, full texts of newsletters, which are generally available to any purchaser of information, and

- internal text data bases, offered by firms and associations to their employees or members as closed user groups.

Extensions of the offer into the fields of

- new data bases
  (e.g. SDA, Schweizerische Depeschenagentur)
  and

- fact data bases, mainly in the economic sector

have now been completed (SDA) or are just about to be introduced.

As an example of a news data bank, we give a brief presentation of the "ELSA" concept of the SDA:

The SDA, as the largest Swiss supplier of information to the mass media, will in the future not only send its messages via telex to its customers, all kinds of editors in Switzerland; it will also store the texts in their entirety on DATA-STAR in the Radio-Suisse Ltd. host computer. Thus SDA is the first Swiss data base operator whose information can be polled on-line via DATA-STAR. What is planned for this year is that the texts produced

daily (about 500 Kbytes) will be stacked via their own line directly in the host computer, which means that they will be available to the public with a delay of 30 minutes.

This example shows how by the growing together of data processing and telecommunications an information market with profitable use arises:

In fact information, once it has been stored, can be used for various purposes. If, for example, a journalist inputs his text directly into a computer system, then this text can not only be passed on to the photosetting installation - which would already result in a major internal rationalisation effect - but the text can also be assembled in data bases into a new information product. This in its turn can be used for the internal support of searches, but it can also be made available to the general or interested public. For the individual user to find the information he wants, a suitable software is also of central significance in DATA-STAR.

2.3. Customer structure and sales channels

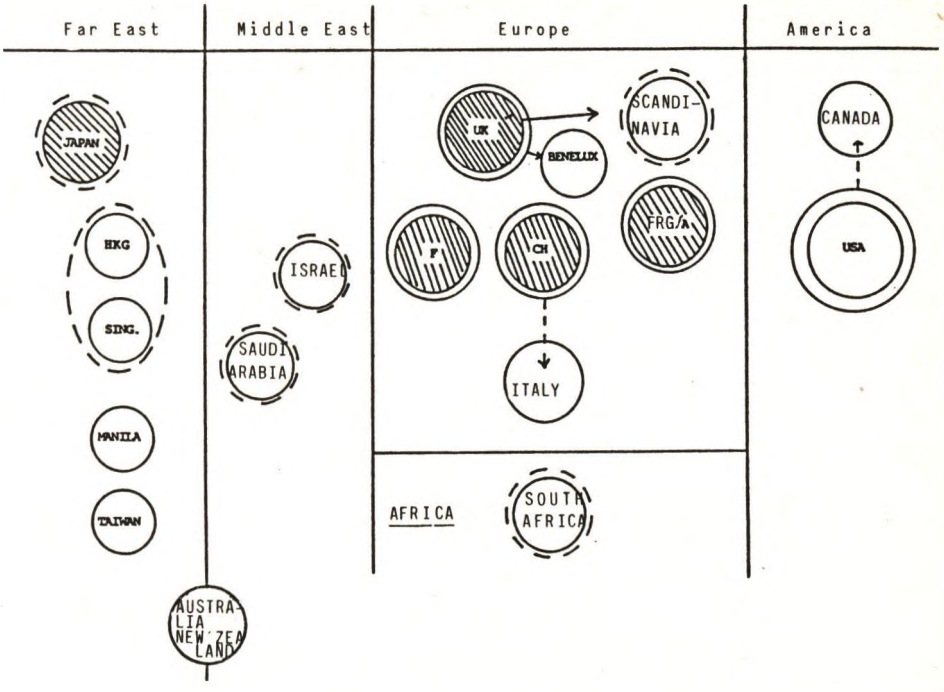
The current range on offer is especially addressed to the following purchasers of information:

- Market segment science:
  Documentalists (90 %) researchers and developers (10 %) from the fields of biomedicine, chemistry and technique.

- Market segment economy:
  Managers, management consultants, businessmen and insurance specialists in the industrial field. 50 % of customers are end users.

By the extension of the market services, additional and end users should be won in the near future in the field of biomedicine, as well as additional banks and insurance companies.

The diagram below gives information about existing and planned sales channels:



In certain countries, agencies are entrusted with the marketing and sale of DATA-STAR services:

West Germany:	Bertelsmann Datenbank GmbH
France:	G-CAM
Japan:	Mitsui Ltd.
Austria:	Bertelsmann Datenbank GmbH
Switzerland:	Radio-Suisse Ltd.
United Kingdom:	DS Marketing Ltd.

3. HOW TO ACCESS DATA-STAR

The DATA-STAR data base service uses as infrastructure Radio Suisse's own host computer in Berne. The infrastructure comprises:

	Capacity Jan. 1985
Processor IBM 3083-B24 Various discs Printer Siemens ND2200 Telecommunication IBM 3725 Own data base software (DSO, DSL)	6.0 MIPS 80 Gigabytes 7500 LPM 24 Ltgn of which 2 x 48 kbps

On this host, the data bases are now leaded, and then the following services are available to the user:

- ON-LINE access to the stored data bases.

- SDI service = Selectrive dissemination of Information. This automatically supplies the user with any new information on a certain question (specific search criterion).

- Enquiries printed OFF-LINE and delivered by post.

Basically, the purchaser of information has the following possibilities of communication to get at his information (see also next page).

a) With a telex or a terminal using a telex adapter via the worldwide telex network:

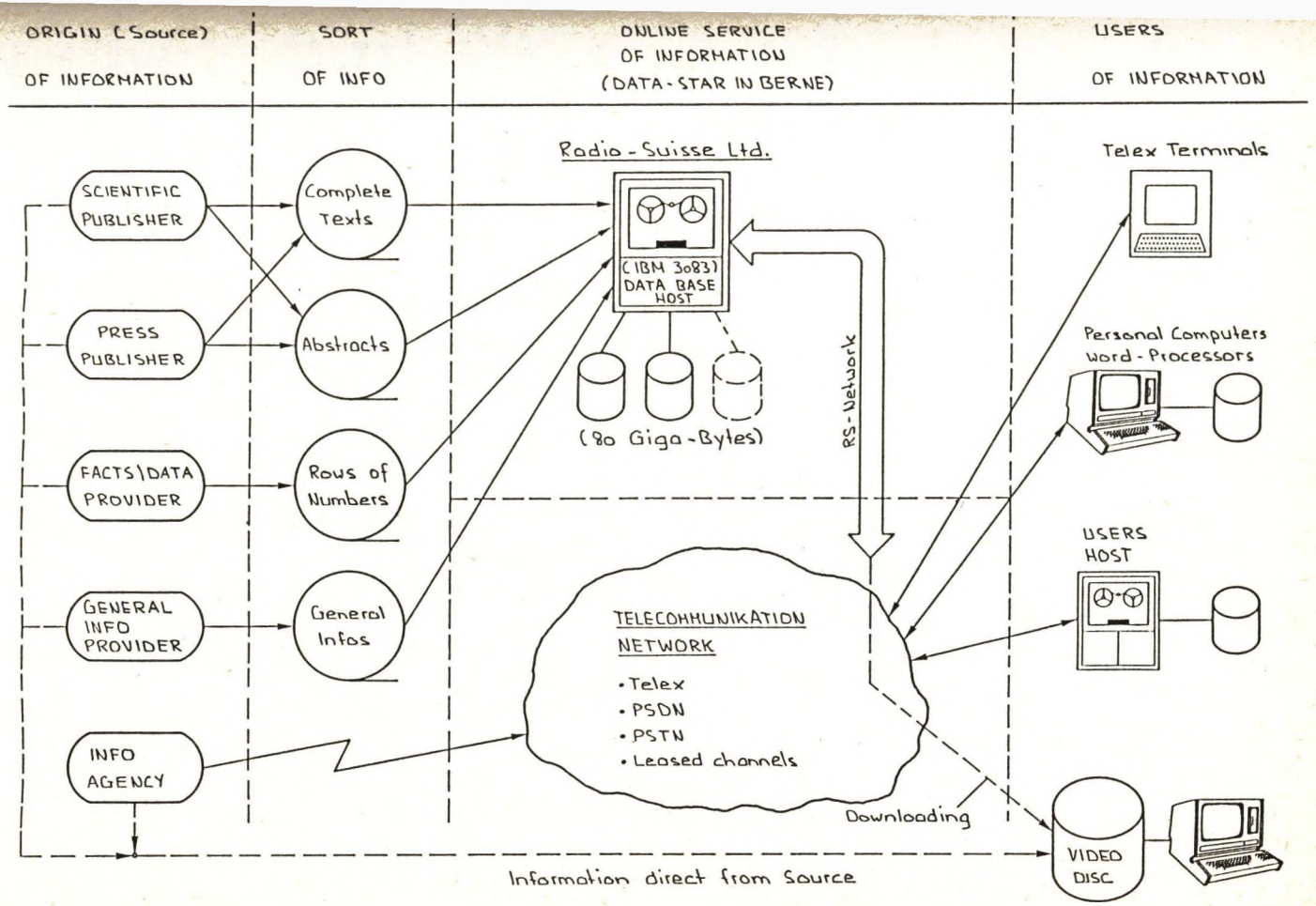
   This method is used by only about 1 %, in countries with a poor telephone structure (high bit error rate) and the absence of a data network. But it is suitable for the purchaser needing a small volume of information and not having a terminal appliance.

b) With a fixed terminal and modem via a leased line:

   Data transmission rate: 300 - 9600 BPS
   Frequency : 4 %
   Type of transmission : synch./asynch.

c) With a fixed terminal and selector modem or acoustic coupler via the telephone dialling service (PSTN):

   Data transmission rate: 300/1200 BPS
   Frequency : 4 %/6 %
   Type of transmission : asynchronous


ORIGIN (Source) OF INFORMATION
SORT OF INFO
ONLINE SERVICE OF INFORMATION (DATA-STAR IN BERNE)
USERS OF INFORMATION
SCIENTIFIC PUBLISHER
PRESS PUBLISHER
FACTS/DATA PROVIDER
GENERAL INFO PROVIDER
INFO AGENCY
Complete Texts
Abstracts
Rous of Numbers
General Infos
Radio-Suisse Ltd.
(IBM 3083) DATA BASE HOST
(80 Giga-Bytes)
RS-Network
TELECOMMUNIKATION NETWORK
• Telex
• PSDN
• PSTN
• Leased channels
Telex Terminals
Personal Computers Word-Processors
USERS HOST
VIDEO DISC
Downloading
Information direct from Source

d) With a fixed terminal via the data network (PSDN):

Data transmission rate: 300/1200 BPS
Frequency : 35 %/50 %
Type of transmission : Packet-switching

For using the DATA-STAR infrastructure for ON-LINE scanning, the user of Radio-Suisse Ltd. must pay a fee of SFR. 33 to SFR. 66 (according the agreement on volume). Or he pays the supplier of information a licence fee (SFR. 25 to SFR. 110 per hour, according to data base).

3.1. How an "ON-LINE" search is carried out

The user, with his possibility of communication, makes the connection to DATA-STAR in Berne.

He then chooses the data base he wants (i.e. the desired supplier of information) and starts his enquiry.

The search proceeds as follows: first of all logically connected key words (e.g. computer and toys) are inputted.

When the article is found, the system then gives an abstract of it with author, language, date of appearence, etc.

The following example shows the process in detail.

a) The DATA-STAR databases and how to find the right one you need

The DATA-STAR system contains files of information supplied by many independent information-gathering companies, each company covering the literature of a particular subject area, e.g. "chemistry" or a professional area, e.g. "management". DATA-STAR offers the major international databases together with complementary databases which enhance the subject coverage of the system.

Information companies process their original material (newspaper of journal articles, conference papers, scientific reports etc.) into records in the form of their choice, ranging from a one-line descriptive index to a long summary. They may have also added "descriptors" of key words or numeric codes to facilitate accurate retrieval of the information. Each company's file is unique in its format but all are searched by the same basic set of DATA-STAR commands.

DATA-STAR tells you wich databases have the information you need.

Each file of information has a particular speciality or emphasis which distinguishes it from others, but subject coverage frequently overlaps. Relevant information will therefore often be found in more than one database.

DATA-STAR has a directory file, called CROS, which tells user how much information each database will yield for each question. Even without a knowledge of DATA-STAR's databases, the user knows which DATA-STAR file(s) to choose to search for information.

The CROS file works very simply. When the user enters the CROS file, he is invited to specify whether he is seeking technical or commercial-type information.

```
ENTER DATABASE NAME   : CROS

1 ALL                    2 LIFE SCIENCES
3 PHYSICAL SCIENCES      4 BUSINESS
5 SOCIAL SCIENCES        6 USER OPTION:
```

User specifies "Business" information by typing "4". The computer then asks for the user's search and replies with the quantity of information present in each database.

```
                    DATASTAR-SEARCH MODE-enter query

user enters search subject- - - - - 1__: solar adj energy
                                       FNTL          16
databases labels with                  INFO         332
quantity of relevant - - - - - - - - - MGMT         142
records                                PTSP        1626
                                       etc
```

ADJ

"adj" specifies that words must be adjacent in same document and in this order.

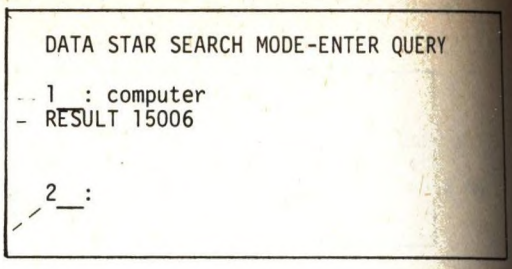
The user is now ready to search the individual databases where relevant information is to be found.

```
1__: ..change

ENTER DATABASE NAME: ptsp
```

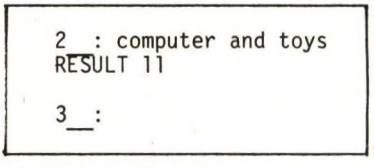
b) How to search information in a given database

The user selects the first file of information (database) that he wishes to search and types into the terminal the word or words which describe the topic of the information that he needs. The DATA-STAR computer searches through each word of each record in the database and replies by telling the user how many records contain the specific words.



On the basis of the quantity of information found, the user can broaden or narrow his original concept by typing in more words. When more than one word is entered at a time, a "connector" must be placed between word to indicate to the computer the relationship that the user wishes the words to have.
These "connectors" are:

AND both words must appear in the same record for it to be retrieved
SAME both words must appear in the same paragraph of the same record
WITH both words must appear in the same sentence of the same record
ADJ both words must appear adjacent to each other (and in the order specified) in the same record
OR at least one of the words must be present in a record
NOT one word must be present in the record and the other word must not be present in the same record



The computer also allows the user to ask for all words which begin with a certain stem but have different endings:

retrieves all records with words starting COMPUTER e.g. COMPUTERS, COMPUTERISED, etc.

```
3   : computer$
RESULT 18887
```

the word is limited to the stem plus one further character, permitting COMPUTER and COMPUTERS but excluding COMPUTERISED etc.

```
4   : computer$1
RESULT 16584
```

The computer searches through a database trying to "match" the words in the record to the word combination that the user has typed in. Because the process is a matching process, the user must consider carefully, before sitting down to the terminal, all the words which could have been used in the original literature to describe his topic.

The user must look for

synonyms and alternatives : computer, electronic: game, toy
plurals : game, games; toy, toys
spelling variations : computerised, computerized
parts of speech : computer, computerised

The flexibility of the DATA-STAR system makes it easy to compose a "search strategy" which takes account of all possibilities.
The user can build up the search, one concept at a time:

```
DS-SEARCH MODE-ENTER QUERY
1   : computer$4 or electronic
RESULT 17343
2   : toy$ or game$1
RESULT 1607
3   : 1 adj 2
RESULT 60
```

The user can request to see the information he has found at any point:

requests the system to print out at the terminal from the 3rd set found, all the elements of the first record

```
4__:.. print 3 all/doc = 1
```

The system prints out the requested record(s), then waits

```
                    AN 80-09886
author - - -  - - - AU ANONYMOUS
title - - -   - - - TI TALKING CHIPS.AN EXPLOSION
                        OF NEW PRODUCTS
source - - -  - - - SO BUSINESS WEEK. VOL: N2637.
                        PAGE: 44D-44F MAY 19, 1980.
abstract - -  - - - AB TEXAS INSTRUMENTS INC. (TI) HAS
                        FINALLY DECIDED TO UNLEASH THE
                        TECHNOLOGY BEHIND ITS
                        REVOLUTIONARY SPEAK & SPELL
                        TALKING ELECTRONIC TOY. TI IS
                        MOVING TO MAINTAIN ITS HEADSTART
                        WITH A MASSIVE CORPORATE EFFORT ...
```

The user may continue printing his results, resume the search, or change to another database.

Search sessions are ended by the user typing a closing command which makes the computer type out the duration of the session and cut the connection (see also chapter 3.3.).

3.2. Communications facilities to DATA-STAR

As already mentioned in Chapter 3, there are in principle four possibilities of access available to the user

- Telex network
- Leased line
- Telephone dialling network (PSTN)
- Data network (PSDN)

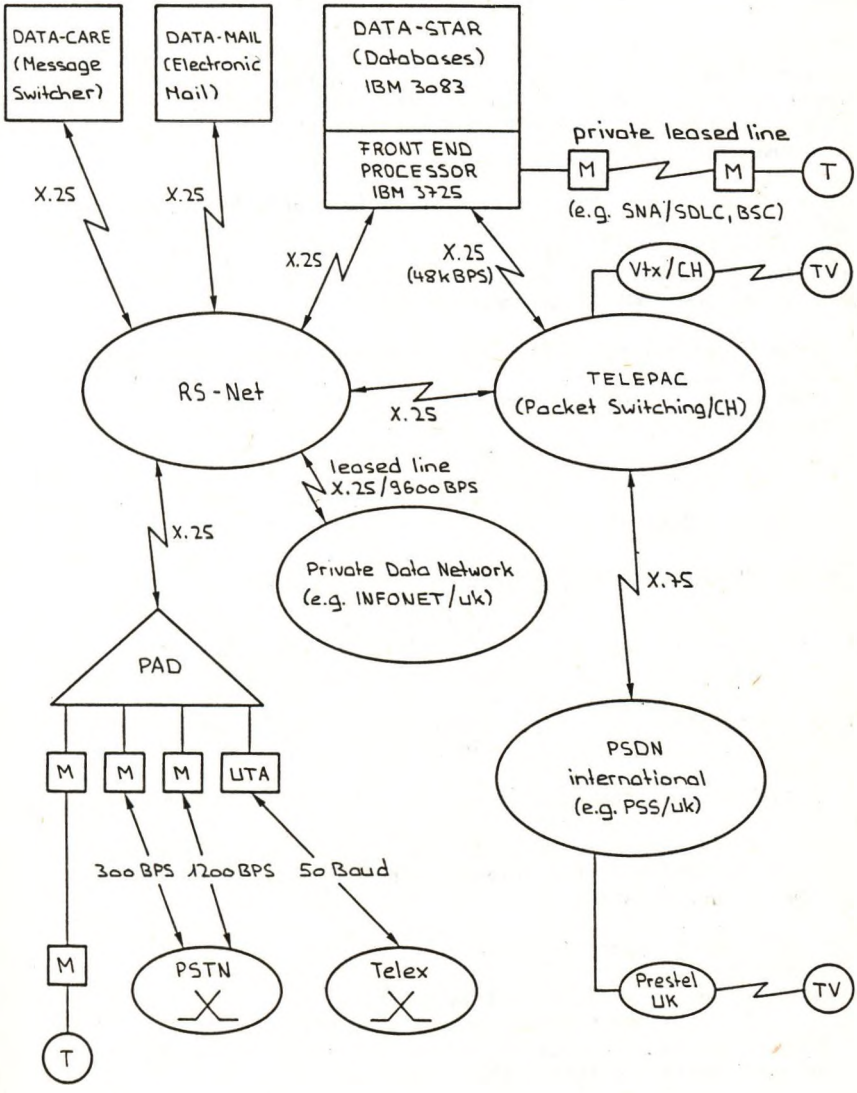
For access to the Value Added Services, Radio-Suisse Ltd. has build up its own RS Network (see next page). The RS Network provides the interface to the public network and permits the user to have automatic access to the following service:

- DATA-MAIL (Electronic Mail)
- DATA-CARE (Intercontinental Message Switcher)
- DATA-STAR (Database Service).

The DATA-STAR Host consists of two front-end processors, type IBM 3725, connected in series as a communications FEP. These are operated with ACF-NCP and NPSI (Network Packet Switching Interface).

Through this network operating system, the connections to DATA-STAR, independently of the means of communication used, can be built up, monitored and disengaged in the same manner.

RS - Network for Value Added Services



The FEPs can be reached direct via

- RS-Network
- TELEPAC (Packet Switching Network in Switzerland)
  or
- Leased Lines

The RS-Network is intended particularly for those users who reach DATA-STAR via the telephone dialling network of the telex network, and who want with the same terminal to use also the DATA-MAIL and DATA-CARE services.

TELEPAC is the data network of Switzerland and is available in particular to Swiss customers. Recently, however, numerous foreign data networks have been connected to TELEPAC, so that this possibility also exists in those foreign countries. Via the RS-Network, the TELEPAC user can also reach DATA-MAIL and DATA-CARE.

Leased lines are intended particularly for large Swiss users or large Swiss and foreign firms, who also store their private files in DATA-STAR and poll them intensively. But private or certain specific public data networks can also be connected to DATA-STAR via leased lines. The use of the data network offers some advantages:

- better availability, i.e. less often engaged or overloaded.
- less dropouts
- communication fees are more favourable than those of the telephone network

3.3. Session Control

Connections can be made by the following numbers.

- Telex network	:	33 917
- Telephone network	:	031/261 333 for 300 BPS
		031/260 871 for 1200 BPS
- TELEPAC	:	049/041 111 for 300 bps
		048/043 111 for 1200 BPS

(For data network connections abroad, other numbers must be dialled).

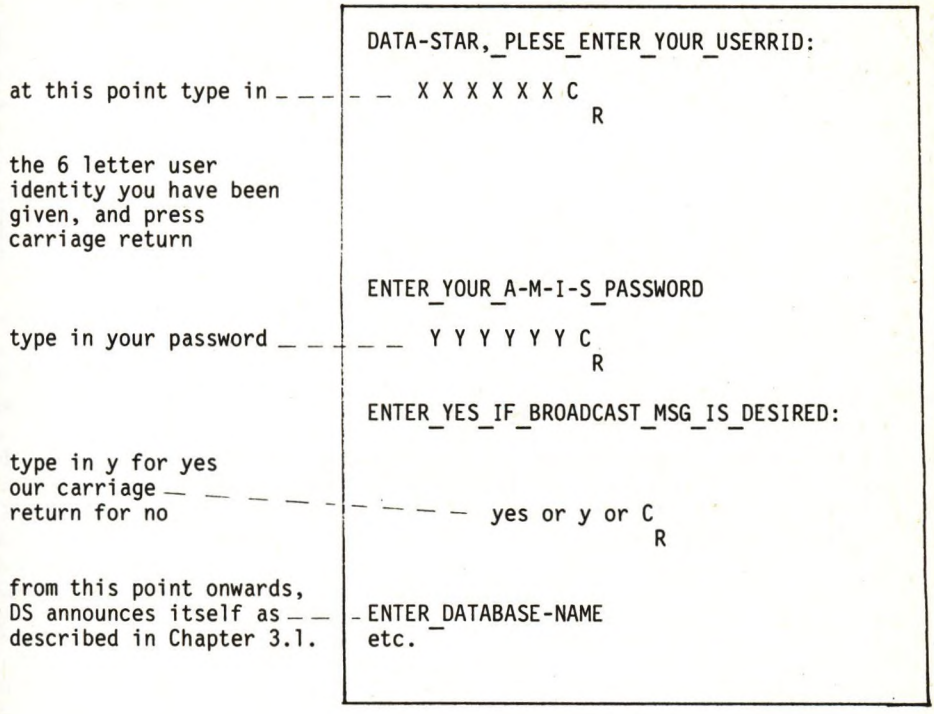
With the exception of the telex terminal, the terminal used must meet the following specifications:

- communication capacity
- ASCII Code
- Byte buildup: 1 start bit, 7 data bits
  1 even parity bit, 1 stop bit.
- Full Duplex Mode or Local Echo
- Key for Carriage Return without New Line
- Speed: 300 BPS or 1200 BPS.

After the connection to DATA-STAR the user has to follow the same "Log-on procedure" independently of the means of communication used.

a) Log-on

The DATA-STAR System announces itself with:



The numerous DATA-STAR commands such as Logical Operators, Truncation, Printing, etc. are not dealt with here. The user receives the corresponding documentation during his training.

b) Ending the session

After ending the dialogue with the computer, the user should end the session with "OFF" and disengage the connection.

He will receive the following announcement on the terminal:

signing-off by type _ _ _ _ _
on OFF

```
_ _ 22_:.. OFF

     CONNECT_TIME_PTSP_0:08:00_HH:MM:SS
     0.133_DEC_HRS_SESSION_141

     SIGN_OFF    09.04.56    30.09.85
```

To resume from same point later in the day type .. OFF_CONT.

c) Interruption without Ending the session

A break in the connection, or break-off by the user without SIGN-OFF, is recognized by the system. Statistically and from the point of view of tariffs, the session until the point of break-off will be treated as a concluded session.

REVIEW: DATA BASES

SECTOR

BUSINESS

ABBREVIATION:	Busi	Cind.	Ieab	Fntl	Harf	Hbro	Hope	Info	Mgmt	Ptar	Ptsp	Ptin	Ptfc	Ptts	Pais	Sgdb	Cmpt	Soft	Hoaa	Honl	Djns
Data base name	Business	Chem. Ind. Notes	Econ. Abs. Int.	Financial Times	Harfax Industry	Harvard Bus. Rev.	Hoppenstedt	ABI Inform	management Cont.	Predicast An. Rep.	Predicast Promt	Predicast Indexes	Predicast Forecasts	" Time Series	Public Affairs	Soc. Gen. de Banque	Computer	Soft	Hoppenstedt Austria	Hoppenstedt Nederl.	Dow Jones
Data base content																					
Business relations	x																				
Products		x	x	x				x	x	x	x	x	x	x		x	x	x			
Economic news		x		x						x								x			x
Information on firms		x		x			x			x	x	x						x	x	x	x
International trade	x	x	x	x						x		x		x	x	x					
Market reports		x	x		x	x		x	x		x	x	x	x	x	x	x				
Statistics/Forecasts/Trends		x	x		x						x		x	x	x	x	x				
Management science						x		x	x						x						
Technologie					x						x	x					x				
Legislation		x									x	x			x	x					
Banks/Finance				x										x	x	x					x
Biology																					
Biochemistry		x																			
Medicine																					
Pharmacology																					
Psychology																					
Chemistry		x																			
Sociology															x						
Physics																					
Environment			x					x	x		x	x									
Material processing		x	x					x	x		x	x									
Building/Architecture			x					x	x		x	x									
Machine construction			x					x	x		x	x									
Computers/Communications			x			x		x	x		x	x					x	x			
Electronics			x					x	x		x	x									
Geology/Mining			x								x	x									
Reference information			x	x	x			x	x		x	x			x	x		x			x
Data/Fact	x					x	x		x	x			x	x			x	x	x	x	
SDI Service (1)		x	x	x	x			x			x	x			x		x	x			x
Updated:																					
weekly		x		x							x	x									x
fortnightly																	x				
monthly	x		x					x	x						x	x					
every other month					x	x															
quarterly										x			x	x				x			
half-yearly							x														
yearly							x												x	x	

CONT. REVIEW: DATA BASES

SECTOR — SCIENCE AND TECHNIQUE

ABBREVIATION:	Biol	Canc	Emed	Medl	Prem	Psyc	Soca	Chem	Ceab	Comp	Cmpt	Dkfl	Eiem	Insp	Ntis	Ulit	Ufor	Ptdt	Mart	Dhss	Mdav
Data base name	Biosis	Cancerlit.	Excerpta Medica	Medline	Pre-Med.	Psychol. Abstracts	Sociol. Abstracts	Chemical Abstracts	Chem. Eng. "	Compendex	Computer Database	Doc. lorries	Eng. Meeting	Inspec	Nat. Tech. Inf. Ser	Environm. databases	"	Defense Market+Tech.	Martindale	Dep. of Health	Merck Index
Data base content																					
Business relations																					
Products												x						x	x		
Economic news											x										
Information on firms												x									
International trade																		x			
Market reports											x	x						x			
Statistics/Forecasts/Trends											x							x			
Management science																				x	
Technologie											x					x	x	x			
Legislation																x	x	x			
Banks/Finance																					
Biology	x	x	x				x									x	x				
Biochemistry	x	x	x	x	x			x	x							x	x				
Medicine	x	x	x	x	x	x	x									x	x			x	x
Pharmacology	x		x	x	x	x		x											x		
Psychology				x	x	x	x														
Chemistry	x	x	x	x	x			x	x							x	x	x	x		
Sociology						x	x									x	x				
Physics										x	x		x	x	x			x			
Environment			x				x			x		x	x		x	x	x				
Material processing								x	x	x	x	x	x	x	x	x	x				
Building/Architecture										x			x		x						
Machine construction										x	x	x	x		x			x			
Computers/Communications										x	x		x	x	x			x			
Electronics									x	x	x	x	x	x	x			x			
Geology/Mining										x			x	x		x	x				
Reference information	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
Data/Fact																					
SDI Service (1)	x	x	x	x		x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
Updated:																					
dayly																					
weekly			x		x															x	
fortnightly	x							x			x	x									
monthly		x		x		x			x	x			x	x	x	x		x			
every other month							x														
quarterly																					
half-yearly																			x		
yearly																		x			

SESSION 2
NON-TECHNICAL ASPECTS OF NETWORKING

INTERNETWORKING EXPERIMENTS WITH THE AUSTRIAN AND HUNGARIAN ACADEMIC NETWORKS

Dr. P. BAKONYI, Dr. L. CSABA
Computer and Automation Institute,
Hungarian Academy of Sciences

Dr. W. KUNFT, Dr. M. PAUL
Technische Universitaet
Wien, Austria

The paper gives an account of the cooperation between Computer and Automation Institute /CAI Budapest/ and Technical University /TU Vienna/ and the results obtained by experimenting with the Networks of the two sides.

INTRODUCTION

Scientific cooperation between Austria and Hungary in the field of networking started in the early seventies.

The first event was the establishment of a reciprocal remote job entry connection between the CDC Cyber 74 computer system at the Technical University Vienna and the CDC 3300 at the Computer and Automation Institute of the Hungarian Academy of Sciences, Budapest.

The next important development occured after the foundation of the International Institute of Applied System Analysis /IIASA/. An ambitious project has been launched to form a Computer Communication Network interconnecting the research community of the Member Countries belonging to IIASA. Austria and Hungary played an active role in designing and implementing the Network the so-called "IIASA Gateway System". This system became the basis of the experiments carried out by the Austrian and Hungarian Institutes.

Recognizing the importance of the crossborder data communication, UNDP/UNESCO initiated a regional project in 1982 called "Crossborder computerized data exchange in science and technology". The participating countries were Bulgaria, Hungary, Yugoslavia and Austria. The main objective of the project was to promote national-international networking.

In the framework of the UNDP project a bilateral cooperation has been formed between the Process Control Department of TU Vienna and CAI Budapest. The paper gives an account of this cooperation outlining the objectives of the joint research, a short description of the Hungarian and Austrian Academic Networks and the results of the internetworking experiments.

In the final part of the paper further plans and some technical problems of the implementation are mentioned.

OBJECTIVES OF THE JOINT RESEARCH

The Computer and Automation Institute and the Process Control Department of TU Vienna are working on the development of their Aca-

demic Network. Both Institutes take part in national networking projects, such as the establishment of Public Packet Switching Data Network, Videotex services, etc. The similar principle of network design followed by the two sides is the basis of the cooperation as the ISO seven layer architecture and the related standards are considered as a basic concept of network development. The ISO standardization process is in progress. Our Institutes take part in the elaboration of the standards.

The first important objective of the cooperation is to benefit from the experiences gained in the course of standards implementation and to evaluate the results and feed this information back to ISO standardization activity and national projects.

The second objective is a natural consequence of the first one.

Having networks with similar architecture, protocols and user community, interconnecting them seems feasible and worthwhile.

So for the long run the interconnection of the Austrian and Hungarian Academic network is envisaged.

THE HUNGARIAN ACADEMY NETWORK /HAN/

The design goals laid down for HAN in 1978 were to provide a service type network for the users of the Academy, and to have a set of hardware/software tools to take part in the development of standardized network products.

To fulfil these two contradictionary tasks we selected a very pragmatic step by step solution.

The first two steps were the design and implementation of a private packet switching subnetwork and the design and implementation of interface converter equipments for connection various hosts and terminals to the packet network. The system is based more or less on the principles of CCITT PAD /Packet Assembly and Disassembly of CCITT X.3, X.28/. The packet switching subnetwork is compatible with the CCITT X.25, X.75 recommendations and therefore in the following part of this paper we will not describe the details of it.

The layout of the network can be seen on Fig. 1. It consists of
- the private packet switching subnetwork /X.25 interface/,
- the terminal concentrators /PAD/
  - . for IBM 3275 compatible BSC terminals,
  - . for TTY compatible terminals,
- interface processors, which are able to connect various types of computers for instance IBM compatible Riad computers to the packet network,
- gateways to connect the LAN "COBUS" of our Institute to the wide area network.

The terminal concentrator in service is able to handle TTY-compatible display terminals, DEC VT52 type terminals. The software for connecting IBM compatible BSC terminals is ready and will be in service in 1985. The interesting feature of the terminal concentrator is that it can accomodate TTY compatible display terminals even if the host side is emulating the BSC type cluster controller.

The host interface processor can be connected to the hosts by real

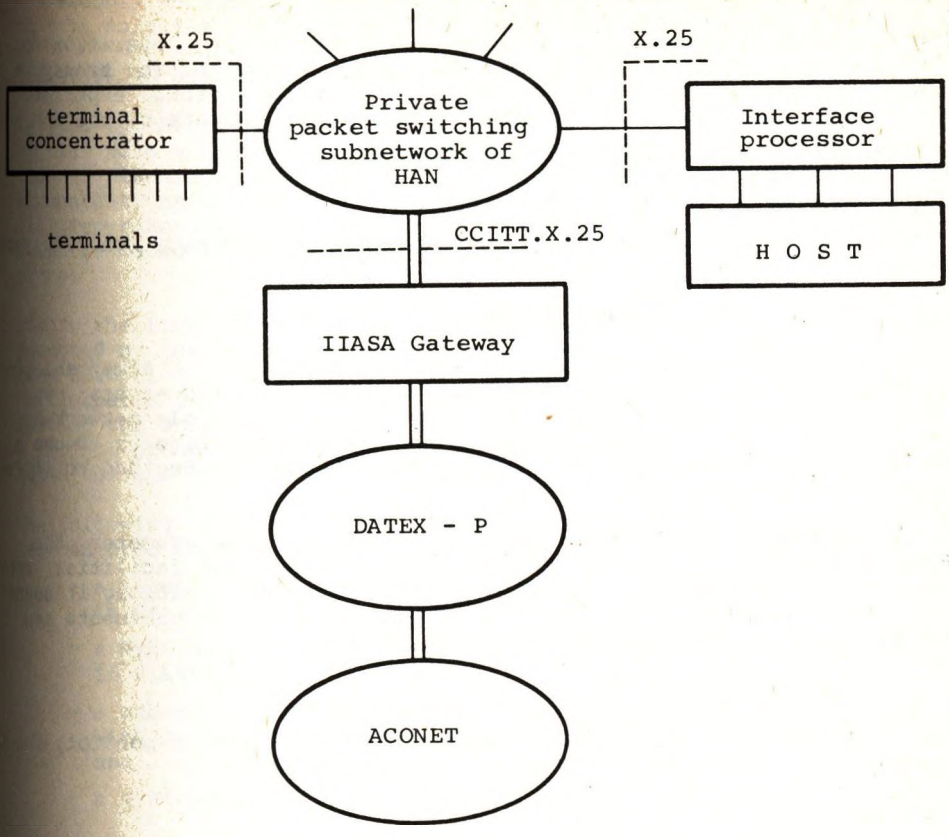


Figure 1. The Hungarian Academy Network (HAN) and the experimental connection

physical terminal lines and can emulate

IBM 2741
CDC 200 UT /display/
DEC VT 52

terminals and convert the traffic sent to them into packets and vice versa.

The interface processor can emulate IBM 3274 cluster controller equipped with several display stations. In this case the control unit of the cluster is emulated locally in the interface processor, but the data entry type display emulators are implemented in the remote terminal concentrators. The data flow between the cluster side and the display side is packetized and depacketized.

The interesting feature of this solutions is that the polling of the host is answered locally in the interface processor, and the polling of the terminal if it is IBM 3275 type is generated in the remote

terminal concentrator. By this way the terminal concentrator acts as a host from the terminal point of view.

The transport protocol has been implemented as the preparation for the next design phase which will be described later. The transport protocol implementation was tested in the international experiment as follows. Here we have to mention that the hardware of the HAN is

- the packet switching equipments
- the interface processors
- the terminal concentrators

is realized on our multimicroprocessor based communication processor system.

Transport Protocol implementation

As the connections in our network are of type B /with acceptable residual error rate and unacceptable rate of signalled errors/, we could choose between class 1 and class 3. As the whole network belongs to our Institute no tariff considerations oblige us to use a multiplexing class over the network connection, we decided to implement the transport class 1.

At the moment we don't use the Quality of Service parameter, the Network Receipt Confirmation, the Network Expedited facilities, concatenation and separation /in a class without multiplexing it seems to be unnecessary/. For the sake of international experiments and conformance we implemented class 0 as well.

Internal layer structure

The layer is separated into three main entities and a monitor, which handles the communication with the operating system.

The monitor

The monitor initializes the data structure for N, T and S-machines. It handles the buffer allocations, decodes all the incoming requests and indications to the layer and transfers the control to the appropriate procedure in the adequate FSM /Finite State Machine/.

The N-machine

The N-machine handles the layer 3/4 interface and the interface between N and T.

The T-machine

The T-machine is the actual Transport Protocol machine. It has internal interfaces with N, M and S. It processes the TPDU-s coming from the N-machine and the commands coming from the S-machine.

The S-machine

The S-machine is the transport service access point machine, which handles the service primitives and communicates with the T-machine. It handles the TSAP-ID tables too.

The implementation was realised in assembler and first tested locally. The next step in testing was to use transport service instead of network service in the interface processor to terminal concen-

trator communication which means that instead of packet assembly/ disassembly we applied TPDU assembly/disassembly. The network based on this solution worked properly.

THE AUSTRIAN ACADEMIC COMPUTER NETWORK

In 1980 the Austrian Ministry of Science and Research established a special work group to give consultative assistance in the coordination of planning and of research activities in the field of data communications at Austrian universities. Especially the planning and all preparatory work concerning the establishment of an Academic Computer Network /ACONET/ in Austria is supervised by this work group.

Within the framework of the Pilot Project ACONET Vienna, which is promoted by the Austrian Ministry of Science and Research, the Process Control Department /PCD/ of the EDP-centre of the TU Vienna participated in establishing an experimental computer network connecting the Universities of Graz, Linz and Vienna. For this purpose a project group of the PCD collaborated with project groups of the other universities concerned.

The following single computer systems of subnetworks were to be interconnected in the framework of the pilot project /see Fig.2./:

- a Local Area Network /ETHERNET/ at the Technical University in Graz,
- a Hybrid-Computer /EAI-PACER 600/ at the Technical University in Vienna,
- a subnetwork consisting of VAX and PDP-11 computers operating with DECnet at the PCD of the Technical University in Vienna and
- a system IBM 370/155 at the University in Linz.

The Austrian packet switching data network DATEX-P was chosen to connect the computers and subnetworks. However very heterogeneous systems had to be connected. Therefore the coordinating work group fundamentally decided that the interconnection should take place according to the rules of the Open Systems Interconnection /OSI/ worked out by ISO. Unfortunately, at the time the work was started three years ago, there existed only one stable set of OSI-rules, namely in the Transport Layer. So we decided to implement as a first step a Transport-Layer-Protocol on top of X.25. Layers 5 and 6 should be realized in a second step in the future.

The second fundamental decision made by the project groups was to integrate not only the subnetworks but also single computers via gateway-systems into the experimental interconnection. Relying on the gateway the project groups integrated the subnetworks at the TU Graz and at the TU Vienna and the single hybrid-computer at the TU Vienna. The IBM-system at the University in Linz was connected directly to DATEX-P via a front-end-system 3705. A PDP-11/34 and two PDP-11/23-Plus-systems have been used as gateways with the operating systems RSX-11M and RSX-11S. The PDP-11/34 is one of the computers of the PCD and is equipped with disks, all other gateways have no disks or floppies.

The software of a gateway can be divided into two parts, the "OSI-side" and the "local-side". On the OSI-side it contains X.25-software. This software was developed several years ago at the PCD and

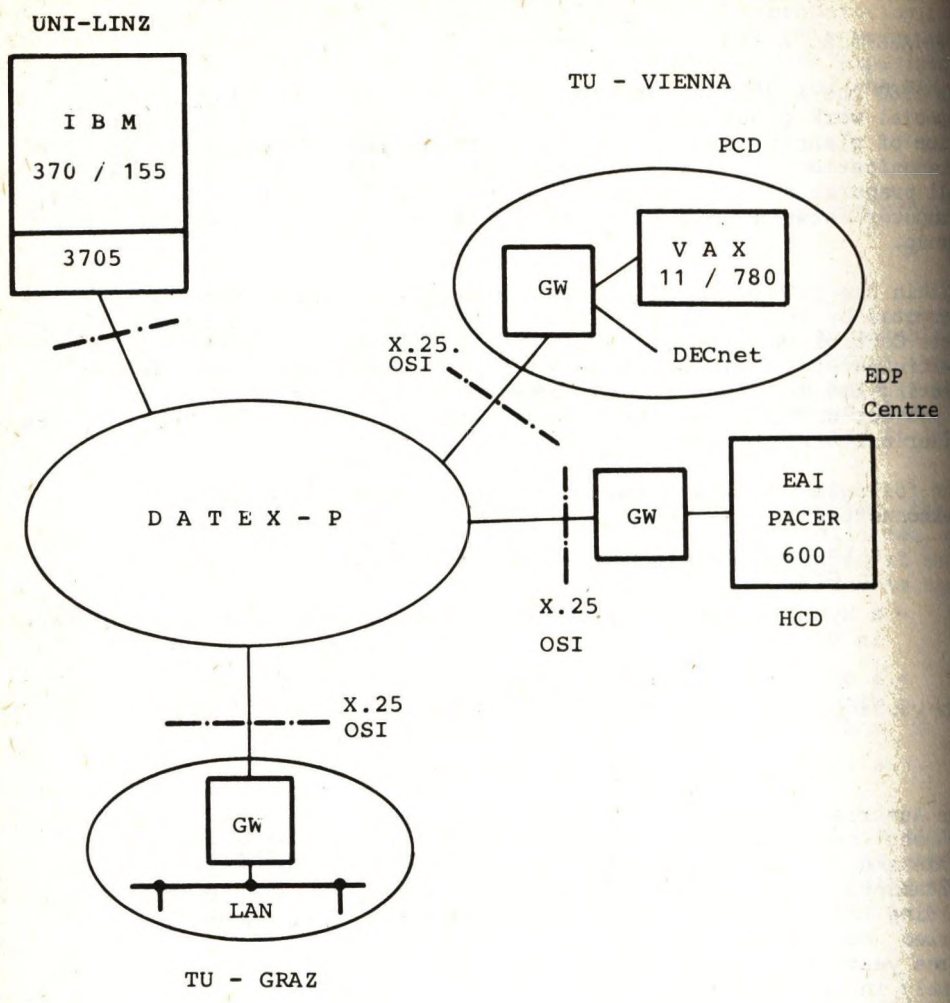


Figure 2. Construction of the experimental interconnection

GW..... Gateway
OSI ..... Open Systems Interconnection
LAN ..... Local Area Network
PCD ..... Process Computer Department
HCD ..... Hybrid Computer Department

has been adapted to meet the requirements. In the Transport Layer protocol class 0 was implemented.

The simple class is sufficient because DATEX-P provides acceptable error detection and error recovery in the first 3 layers. As mentioned above layers 5 and 6 remain empty in the first version of our gateways. This means that the application program in layer 7 in the gateway performing the relay-function directly uses the services of the Transport Layer. On the local-side the gateway contains the communication software for the subnetwork, e.g. DECnet.

The PCD's project group implemented the X.25 - and the OSI-software in the gateways. Additionally this group was responsible integrate its subnetwork in the experimental interconnection, to develop software and firmware to load a gateway via DATEX-P and to provide a simple central support for developing gateway-software via DATEX-P.

The protocols of X.25 and the transport protocol class 0 have been implemented as Finite State Machines.

An universal interface specification according to the Transport Service Definition ISO DIS 8072 has been worked out to define the interface between layers 4 and 5. This specification considers the application of all protocol-classes.

THE RESULTS OF THE EXPERIMENTS

The Fig. 1. shows that the packet-subnetwork of HAN is connected to the IIASA gateway, situated in Laxenburg. By this way terminal users of HAN are able to set up connections to the IIASA resources and vice versa. The IIASA gateway is connected to the DATEX-P network of Austria and therefore for the period of experiments the equipments of ACONET and HAN can communicate.

The cooperation between the Hungarian and Austrian work groups in implementing protocol class 0 of the Transport Layer was very precisious for both sides. The crossborder communication experiments allowed to test the compatibility of software developed independently by different groups.

We have tested our class 0 implementations very efficiently and resolving some minor problems, for instance the coding of not used fields of the headers, our systems were able to exchange information without any problem.

FURTHER PLANS

In the previous sections we have pointed out that the transport implementation is the preparation for further development of HAN. The direction of this development is to implement an experimental OSI System where the Application Level protocol is the FTAM protocol. We have decided to use a front-end processor such a way that the four lower layers are implemented in the multimicroprocessor system and the higher layers in the IBM compatible host.

The problem of having a real physical interface instead of an abstract one between layers 4 and 5 is investigated. As the result of our cooperation, the Austrian proposal for CCITT to design a Network Independent Interface /NII/ were used as a model to define the so-called Network Independent Transport Interface, where the problem is not only the coding but e.g. the solution of the interface flow con-

trol procedure.

The real transport interface implemented in the front-end and the host opens the way for FTAM experiments between the two cooperati Institutions.

IMPACT OF NETWORKS ON DATABASE USERS EDUCATION

Mieczysław R. Muraszkiewicz and Zbigniew M. Nowicki

Institute for Scientific, Technical and Economic Information
00-926 Warsaw, ul. Żurawia 3/5
Poland

One of the most convenient way of access to various databases from one place is to use a computer network. Therefore, networks become a significant tool in everyday information practice. This is the reason one has to provide adequate network training for future and present staffers of information centers and casual users. More efficient usage of information resources available via the network is an important effect of the training. A method of network users education based on the authors approach is presented in the paper. A stress is put on the end-user role in a man-network interaction.

INTRODUCTION

As a result of progress in computer technology society appears to be on the threshold of new era. It seems to be possible that many of us will be members of the so called "information society". In this context the role of information, its processing and people who are involved in information handling is supposed to increase dramatically. A trend of this kind has already occured in the developed countries where information is considered as a great national resource.

Undoubtedly, computers are not newcomers to information service. However, there is still a lack of information people understanding of the computer possibilities and functioning. This statement is much more true as far as computer networks are concerned. The aim of this paper is to present how computer education for information and library people in Poland is to be extended to include computer networks usage.

NETWORK ACCESS TO DATABASES IN POLAND

The MSK/ASK network is the most advanced project in the field of computer networks implemented in Poland at the moment. This is a joint project of high schools and the Polish Academy of Science. MSK/ASK is a heterogenous network with packet switching; currently it is composed of three nodes that are located at universities in Wrocław, Gliwice and Warsaw. The ODRA 1305 computers are used as hosts; they are connected to the network via the ODRA 1325 computers which work as front-end processors. It is expected that the RIAD computers that dominate on the COMECON computer market will be connected to the network, soon. It is also planned to increase the number of nodes; they will be located first of all in the cities where big universities carry out their activities.

One of the goals of the MSK/ASK network is to allow easy access to databases stored in different host computers. It is assumed that various databases produced in Poland as well as imported ones will be accessible via the network. At the current stage of the network implementation a part of the INSPEC database, section C is offered by the computer center of Wrocław University. Despite the fact that the network is working in an experimental mode it is already used by information center and library staffers.

Polish users have also accessed the DIALOG Information System since 1983. The system is offered by the DIALOG Information Services Inc., Palo Alto /Ca./ and available in the frame of the British Library Automated Information Service. Polish information people can communicate with DIALOG from a terminal located at the British Council in Warsaw. More than 180 databases are given at the user's disposal.

Poland participates also in the project conducted within the International System of Scientific and Technical Information /MSNTI/. At the moment, a pilot version of the MSNTI network is implemented. It consists of the nodes located in Moscow /USSR/, Prague /Czechoslovakia/, Sofia /Bulgaria/ and Budapest /Hungary/. There are preparatory works to connect the MSK/ASK network to the MSNTI one. For a while, the Polish users can access databases of MSNTI through a terminal located at the Institute of Computer Science, Polish Academy of Science in Warsaw.

Summarizing this brief outline one can state that information and library people in Poland have already access to databases in the network mode. There exist many difficulties and limitations now, however, they are considered as motivation rather than discouragement.

COMPUTER SCIENCE EDUCATION FOR STUDENTS OF INFORMATION SCIENCE

Before entering into discussion on computer networks education let us briefly present a current status of computer science education for information and library personnel in Poland. We focus our attention on university education.

In Poland the students who wish to join information and/or library service are educated in information science in secondary schools, colleges and universities. Particular stress on computers as tools for information processing and handling is paid at universities. Of 14 universities and high schools 3 conduct remarkable computer science training at information science and librarianship faculties. These are Computer Science and Management Faculty of Wrocław University, Institute for Information Science at Silesion University at Gliwice and Institute for Library and Information Science at Warsaw University /ILIS/. It should be noted that the nodes of the MSK/ASK network presented in the previous section are located in the three mentioned universities.

As an example of computer related courses offered to students let us mention those which are run at ILIS:

- Introduction to computer science /45 hours/
  More detailed information on the methodology and curriculum the reader can find in Muraszkiewicz and Nowicki (1983, 1984).
- Information systems. Analysis and design /60 hours/
- Programming languages /45 hours/
- Databases /45 hours/

COMPUTER NETWORKS CURRICULUM FOR INFORMATION SCIENCE STUDENTS

Needless to continue arguing for computer networks usage for the purposes of information service. Let us now proceed to a curriculum which is to be implemented at ILIS, soon. The main goals of the proposed course are as follows: /a/ to provide the students with basic notions dealing with computer networks; /b/ to provide the students with knowledge about the role of computer networks in information service; /c/ to provide the students with a context from which to consider possible future directions in networking and its applications. Let us have a look at the curriculum.

1. Computer network as a tool for automation of information systems	4
1.1. Computer network as an element of computer development line (cf. Muraszkiewicz and Nowicki, 1984)	1
1.2. Networks classification	1
1.3. An outline of computer networks development	1
1.4. Advantages of information systems implementation in computer networks	1
2. Computer network operating principles	8
2.1. Operating principles of a telephone network Network structure; multiplexing of multiaccess channels; communication through a satellite channel	2
2.2. Application of a telephone network for data transmission	1
2.3. Organization of data transmission in computer networks Circuit switching, message switching and packet switching; routing rules; protocols; transmission through a common channel	4
2.4. Data security in computer networks	1
3. Examples of information systems accessible through computer networks	12
3.1. Airline reservation system GABRIEL II	3
3.2. Bibliographic information system DIALOG	9
4. Computer networks development perspectives	6
4.1. Computer networks in Poland	1
4.2. Network access to the International System of Scientific and Technical Information	2
4.3. New telecommunication services Teletex network; videotext and teletext services	2
4.4. Concept of Integrated Services Digital Network	1
the total number of hours	30

The above course is to be supplementted with 30 hours of class discussions, tutorials and terminal sessions including practical training and contact with the DIALOG system and the MSK/ASK network. A textbook by Nowicki (1985) will be used as a manual.

FINAL REMARKS

The majority of information personnel in Poland exhibits a lack of computer background and skills, therefore, upgrading the skills is considered as an important

task to be done. The Center for Scientific, Technical and Economic Information in Warsaw is responsible for that in Poland. We have already submitted the above curriculum as a benchmark for preparing the final version of the programme addressed to information and library people who wish to learn computer networks and their applications in the field of information.

REFERENCES

[1] Muraszkiewicz, M.R. and Nowicki, Z.M., Model of Computer Science Education for Information and Library Staff, Proc. of Int. Conf. on Application of Internationally Available Databases to National Scientific and Technical Information Systems (OMIKK-UNESCO-FID, Budapest, 1983) 710-718.

[2] Muraszkiewicz, M.R. and Nowicki, Z.M., End-User Oriented Approach to the Development of Computerized Information Systems, in: van der Laan, A. and Winters, A.A.(eds.) , The Use of Information in a Changing World, (North-Holland, Amsterdam, 1984) 161-170.

[3] Nowicki, Z.M., Operating principles of Automated Information Networks, SINTO Materiały Metodyczne (CINTE, Warsaw, 1985)(in Polish)

SESSION 3
DATA COMMUNICATION ON PUBLIC DATA NETWORKS AND ISDN, TELEMATICS

NETWORK TECHNOLOGY INDEPENDENT INTERFACE (NII)

Klaus Sambor

Fernmeldetechnisches Zentralamt
of the Austrian PTT
Wien, Austria

This interface is designed to be applicable with already existing telecommunication networks (connection oriented or connectionless) as well as for forthcoming ISDNs (basic access) and LANs. It is a simple interface which is independent of specific networks and makes terminals independent of specific network access protocols. This interface is intended to be used only for short (some metres long) point-to-point configurations.

1. INTRODUCTION

The Austrian PTT got knowledge of the NII concept in March 1983. The original proposal of the NII concept [1] will be briefly described unde item 2.

The proposal was brought to the attention of the Sub Working Group CEPT-CD/LM in April 1983 and led at the next CD/LM meeting in June 1983 to a proposal for a harmonisation project for "intelligent DCEs" (including layers 1 to 3) using the NII as an interface between this DCE and a new type of DTEs (including layers 4 to 7). In the Working Group CEPT-CD (October 1983) it was decided to make a feasibility study about the NII concept at first within Sub Working Group CEPT-CD/IP, before starting with a harmonisation project. (CD/IP is responsible for layers 1 to 4.) Within this "one-year-feasibility study" 2 Drafts of an NII have been worked out. These Drafts are based on the original proposal and will be described under item 3.

In Oct. 1984 the Working Group CEPT-CD evaluated the feasibility study and agreed to a Study Point, reading.

"A Customer Interface Protocol using a physical interface between layer 3 and layer 4 (the Network Independent Interface), which permits the connection of any DTE to any telecommunications network. Applicability of this interface to LANs should also be considered":

At the same time that topic was discussed at an ISO meeting (ISO/TC97/SC6/WG5) and the ISO Member Bodies were requested to contribute to this question, in order to be able to decide if a New Work Item "NII" should be proposed.

At the next CD/IP meeting (January 1985) a third Draft was discussed and revised. The results will be described under item 4. It was agreed that the industry should be asked what they would think about the NII concept. A questionnaire was sent to EEC, ECMA and EUCATEL. It has been expected that these bodies will agree to discuss this topic at a joint meeting with CEPT (June 1985).

In the meantime the NII concept has been also discussed within the Austrian Member Body to ISO and it was agreed that a New Work Item "NII" should be proposed at the relevant ISO-meeting (March 1985). The Austrian contributions to this meeting and the result of the discussions are described under item 5.

Based on these discussions on the one hand and a detailed analysis between ISDN-layer-3 protocol (Q. 931) and the OSI network service (X.213) on the other hand, an NII Draft 4 has been prepared (see item 6).

In this item also a summary is given, where a comparison between the current interfaces and the NII concept should show that this new concept would have many advantages.

2. ORIGINAL PROPOSAL

In the real world global network many different interfaces exist. In the future this situation will becom even more complex with the introduction of ISDNs.

The NII is different from all existing interfaces of the CCITT series V ..., X ..., S... and so on, but is not in conflict with them. Neither the OSI-model is changed nor must existing networks be changed because of the NII concept. Also the existing V .. and X.. interfaces can be used further in the same network.

The NII uses only service primitives and not protocol elements of network dependent protocols. The principle of the NII is to realize a physical sparation at the place where the separation is logically provided between layer 3 and layer 4.

The NII is derived from the Network Service Definition as defined in CCITT Rec. X.213 and ISO 8348 respectively.

The functions of the operational phases

- connection establishment
- transfer of data
- connection release

are realized in one end system at the NII by transferring the relevant service primitives (with a limited max. size) as "action" (A) from one entity of one layer to the other entity of the other layer. Each action has to be responded locally by a "reaction" (R) of the entities involved.

It is important to differentiate between static entities and dynamic entities. A connection is represented in an end system by dynamic created entities which exist only for the duration of the connection. These dynamic entities are created by static entities which exist permanently in each end system for the sole purpose of creating and destroying dynamic entities.

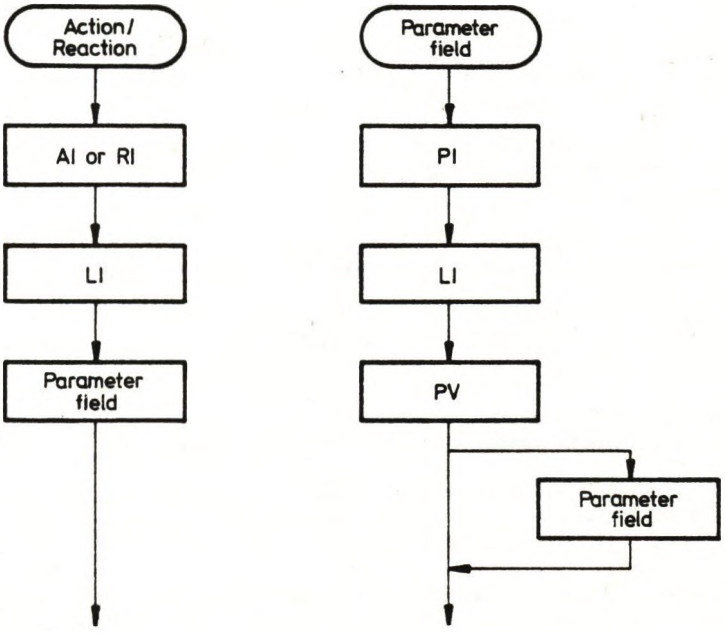
Over a service accesspoint (with a global address c) in each end system, exactly one static N-user entitiy communicates with one static N-provider entity. "Connection end points" (with an address c, k) at a service access point represent the resources which the static N-provider entity (at the calling end system) allocates to dynamic entities on both sides of the interface boundary (and the static N-provider entity at the called end system later on) during each connection establishment.

3. DESCRIPTION OF DRAFT 1 AND DRAFT 2 OF THE NII

Draft 1 was formulated in the form of a "Draft CEPT Recommendation" with the following contents.

- General
- Functional description
- Format and coding
- Interface characteristics
  . Interchange circuits
  . Mechanical characteristics
  . Electrical characteristics

The first two items contain in principle the original proposal. New in this first Draft is only the proposal for the format of the service primitives (based on CCITT Rec. T.62) and their coding. Fig. 1 shows the allowable sequences of units within an action or reaction. In addition, first considerations for an NII-prototype are mentioned. The first Draft covers only the needs for non-voice services.



AI ... Action Identifier
RI ... Reaction Identifier
LI ... Length Indicator

PI ... Parameter Identifier
PV ... Parameter Value

Fig. 1: Allowable sequences of units within an action or reaction

As for non-voice services, the problems were not expected to include the most stringent requirements for the realization of the NII concept, the "problem area ISDN" has been considered in the second Draft.

For an analysis 3 steps are necessary:

- Study of the Network layer for ISDN (CCITT Rec. Q.931, layer 3 specification)
- Proposals for enhancements of the Network Service Definition (CCITT Rec. X.213)
- On the assumption that the proposals for enhancements would be accepted, definit of the mapping between service primitives and protocol elements must be accomp

The First step shows how complex the ISDN layer 3 is. There are at present (Dec. 30 different "message types" which are made up of 25 different "information elements". Each of these elements consists of many parts and for each part many values are standardized.

For simplification purposes and to get an initial understandig of problems involv in the ISDN-layer 3 protocol, for the second Draft only "stimulus mode of operati and only "normal cases" (establishment, data transfer, release) have been consider

4. DESCRIPTION OF DRAFT 3 (REVISED) OF THE NII

The analysis of the stimulus mode operation has been continued and has now been completed, with the exception of some stimulus information elements which have also not yet been finally defined in CCITT Rec. Q.931.

Before going into details about what is new compared with Draft 2, it will be reported what has been discussed after the presentation of Draft 3 at the CEPT-CD/IP meeting in January 1985.

At first the whole problem area of possible implementations of the Network Service Definition with respect to the NII was discussed and the result was described in Fig. 2.

Investigations of the Networks described in Fig. 2 should be made in order to find out whether the mapping between the functions

- required by the network and
- required by the Network Service Definition (according to X.213 = ISO 8348)

would be possible. Because it has already turned out that in some cases difficulties are to be expected for this mapping, three different approaches to overcome these difficulties were discussed:

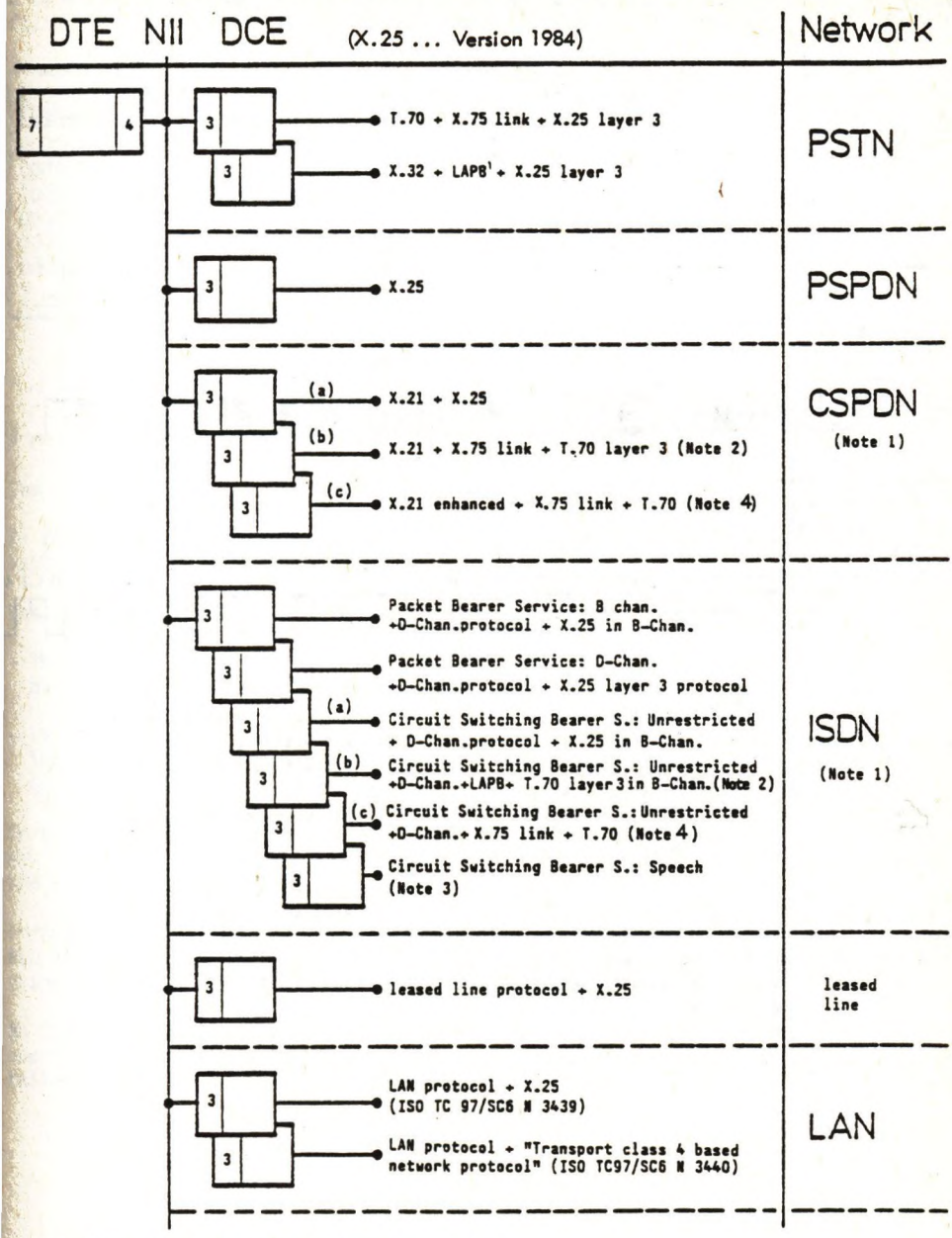
I. To stick to the current version of ISO 8348. Necessary changes or enhancements have to be made in the underlying layers and consequently in the corresponding ISO Standards and CCITT Rec. in order to fulfil the requirements of ISO 8348.

II. To enhance ISO 8348 in order to be able to provide what is additionalle require by the underlying layers of any network.

III. To minimize all necessary changes or enhancements by providing modifications to ISO 8348 and standards related to layers 1 to 3 wherever it is easier.

It has not yet been finally decided, but approach III seems to be the proper method to be adopted.

Some examples of possible implementations of the network service defined in IS 8348 with respect to the NII



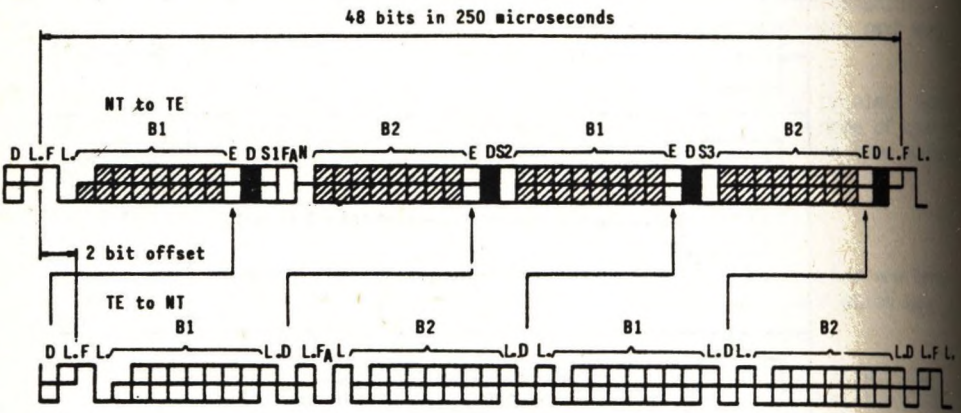
Note 1: Either a or b may be chosen after further study
Note 2: T.70 layer enhanced (See ISO TC 97/SC 6 N3150)
Note 3: The relationship of speech to the model in particular to X.213 is for further study.
Note 4: Requires further study
At the time beeing no need for an NII for Telex exists.

Fig. 2

The most important addition to Draft 3 (revised) is Appendix 7:

"Necessary speed at the interface". The considerations are based on CCITT Rec. I. 430 (Basic User - Network Interface, Layer 1 Specification).

Fig. 3 shows the frame structure at reference points S and T. The information transferred across the NII has to be "filled" in the frames (or "taken out" respectively of the frames) of 48 bits for 2 B and 1 D channels (further on called S-frame).



F = framing bit
L = DC balancing bit
D = D-channel bit
E = D-echo-channel bit
F_A = Auxiliary framing bit (= 0)

N = bit set to a binary N = $\overline{F_A}$
B1 = bit within B channel 1
B2 = bit within B channel 2

A = bit used for activation

S1, S2 = Reserved for future standardization

Dots mark those parts of the frame that are independently DC-balanced

Fig. 3

The information for the B1, B2 and D-channels can be transferred e.g. as shown in Fig. 4. Because of the big overhead (7 octets for 1 PCM sample for digitzed voice) a speed of about 1,2 Mbit/s would be necessary (each 125 µs 1 PCM sample has to be filled in the S-frame).

87654321	
10000011	AI: N-DATA.Request
00000110	LI: 6
00010000	PI: Connection end point address
00000001	LI: 1
00000111	PV: e.g. 7
00000000	PI: NS user data
00000001	LI: 1
1 octet	PV: One sample for digitized voice

Fig. 4

A reduction of this bit rate is possible by blocking the digitzed voice.

The question is what size should be allowed for blocks.

At first the assumption is made that blocks for voice and blocks for non-voice data have the same max. size.

The reason for this assumption is

- The dynamic N-user entities and N-provider entities respectively must be able to handle both voice and non-voice data.

- The max. delay of one "voice block" to the next should not be increased by the insertion of a "non-voice data block", whose size is larger than that of a voice block.

On the assumption that voice and data should be transferred in blocks of the same max. size, the block size will be dependent on the delay time max. acceptable to the voice terminal.

Further on the assumption is made that the transmission of digitized voice across the NII is of highest priority, so only one data block can be inserted between two voice blocks (and "signalling data blocks" respectively).

An example for a block size of an 8-bytes-voice block and 8 bytes overhead and a speed of 512 kbit/s at the NII is shown in Fig. 5 (review), and in Fig. 6 a detailed illustration of one part of Fig. 5 is given.

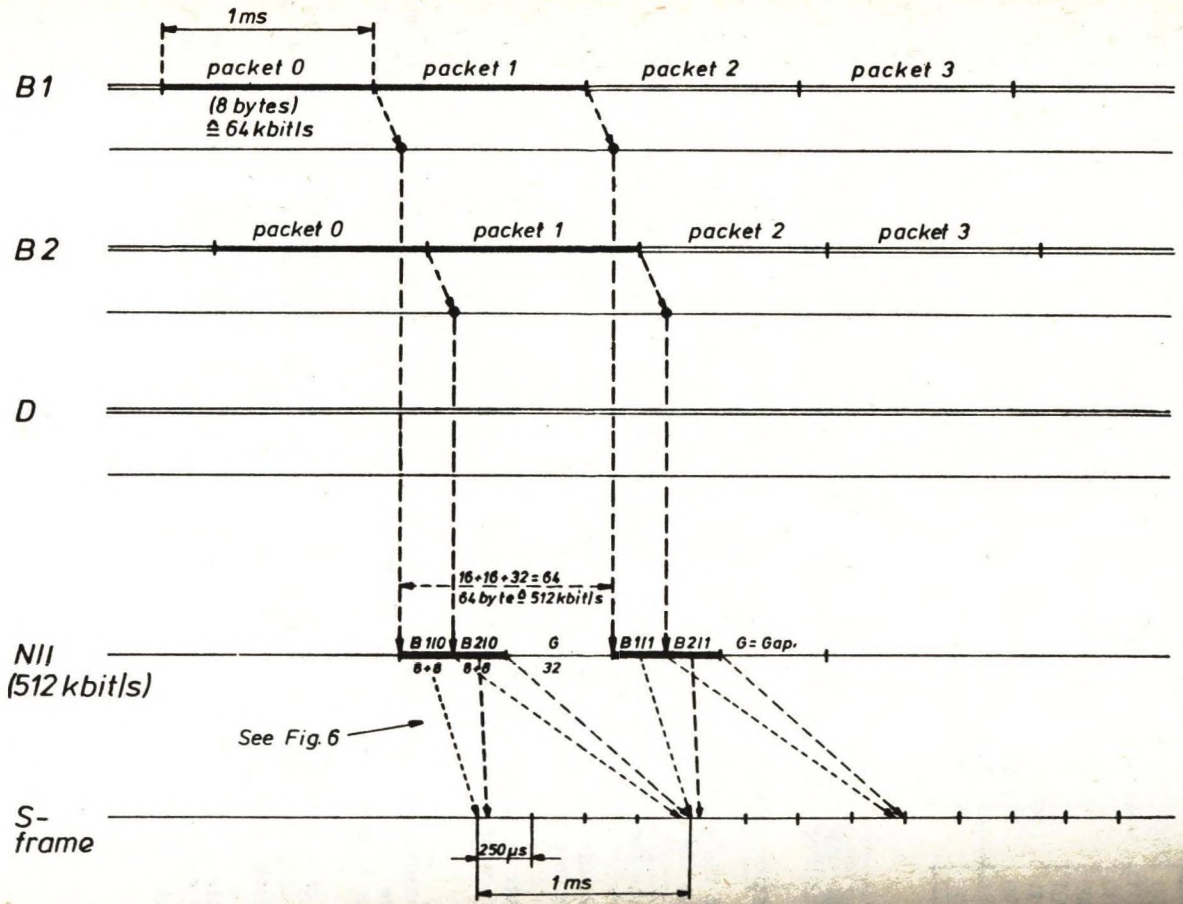

1ms
B1
packet 0
packet 1
packet 2
packet 3
(8 bytes)
≙ 64 kbit/s
B2
packet 0
packet 1
packet 2
packet 3
D
16+16+32=64
64 byte ≙ 512 kbit/s
NII
(512 kbit/s)
B110
B210
G
B111
B211
G= Gap.
8+8
8+8
32
See Fig. 6
S-
frame
250µs
1 ms

Fig. 5

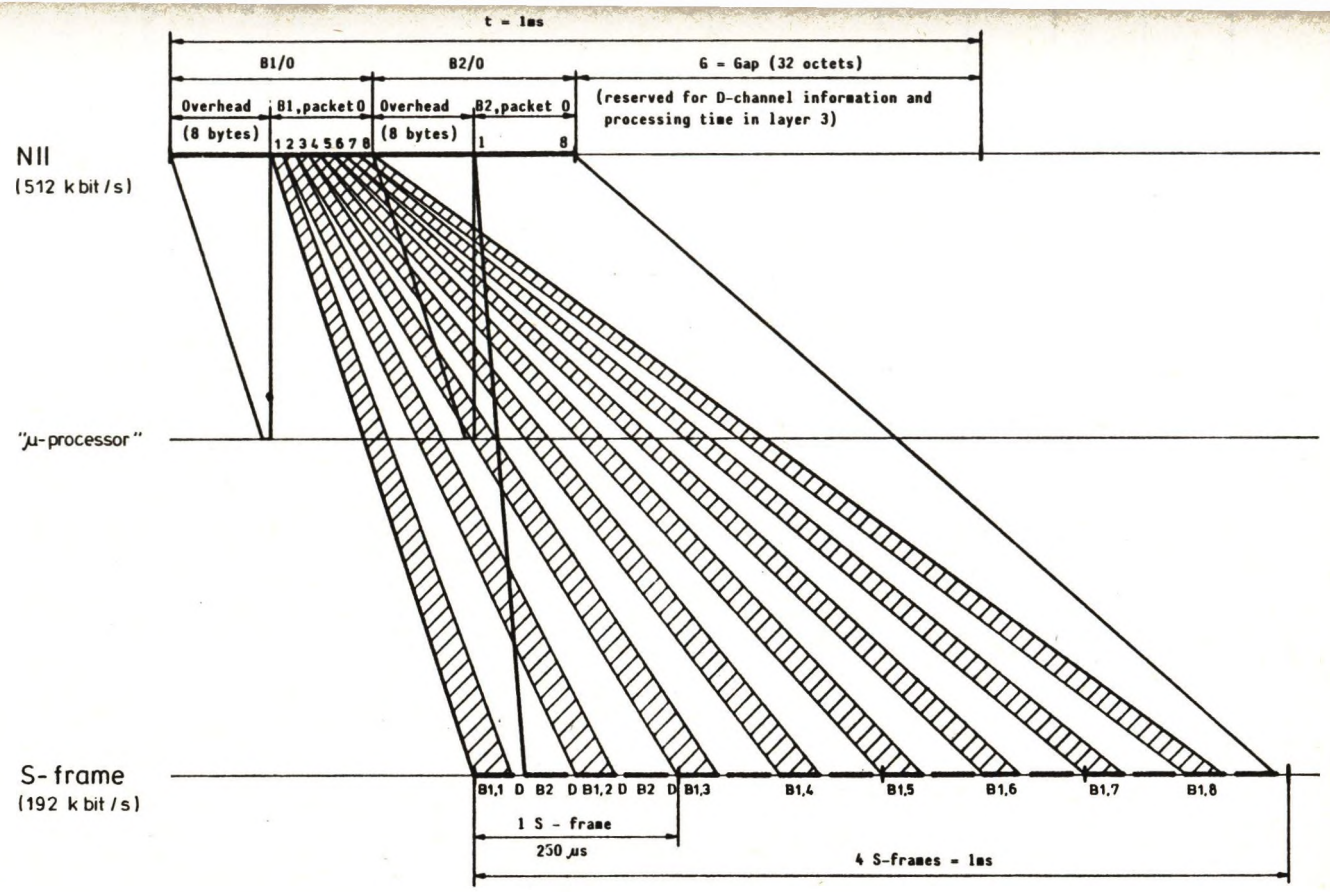
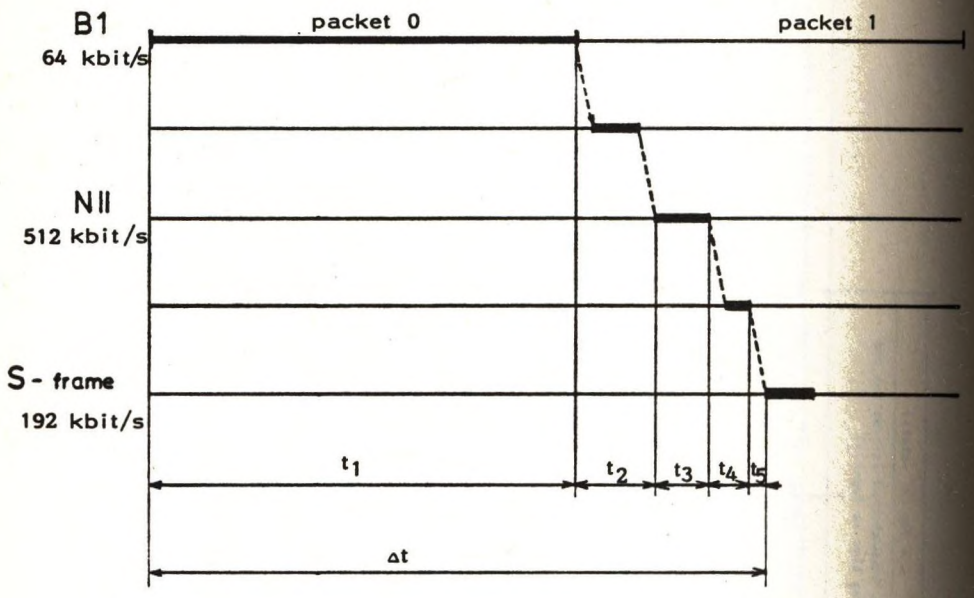

t = 1ms
B1/0
B2/0
G = Gap (32 octets)
Overhead
B1,packet 0
Overhead
B2,packet 0
(reserved for D-channel information and processing time in layer 3)
(8 bytes)
1 2 3 4 5 6 7 8
(8 bytes)
1
8
NII
(512 k bit / s)
"µ-processor"
S-frame
(192 k bit / s)
B1,1 D B2 D B1,2 D B2 D B1,3 B1,4 B1,5 B1,6 B1,7 B1,8
1 S - frame
250 µs
4 S-frames = 1ms

Fig. 6

In Fig. 7 the delay time Δ is dealt with, which lasts from the time of generation of a PCM sample until that sample will be inserted in the S-frame. 2 Examples for a block size of 8 and 128 bytes are contained in Fig. 7.



$$\Delta t[ms] = t_1 + t_2 + t_3 + t_4 + t_5$$

$$t_1 = \frac{x \cdot 8\ [bit]}{64\ [kbit/s]}$$ x .. block size [byte]

$t_2 + t_4 + t_5 = G_p$ ... Gap reserved for "processing time"

$0 < t_5 \leq 125\mu s$

$$t_3 = \frac{x \cdot 8\ [bit]}{V_{NII}\ [kbit/s]}$$ V_{NII} . speed at the NII

Example 1: Block size = 128 bytes
Speed = 512 kbit/s; Gp = 0,3 ms

$\Delta t = t_1 + t_3 + G_p$ = 16 ms + 2 ms + 0,3 ms = 18,3 ms

Example 2: Block size = 8 bytes
Speed = 512 kbit/s; Gp = 0,3 ms

$\Delta t = t_1 + t_3 + G_p$ = 1 ms + 0,125 ms + 0,3 ms = 1,415 ms

Fig. 7

Fig. 8 gives a general review of the relationship between the delay time Δt and the size of the voice blocks, assuming different speeds at the NII (an additional assumption is the value Gp = 0,3 ms).

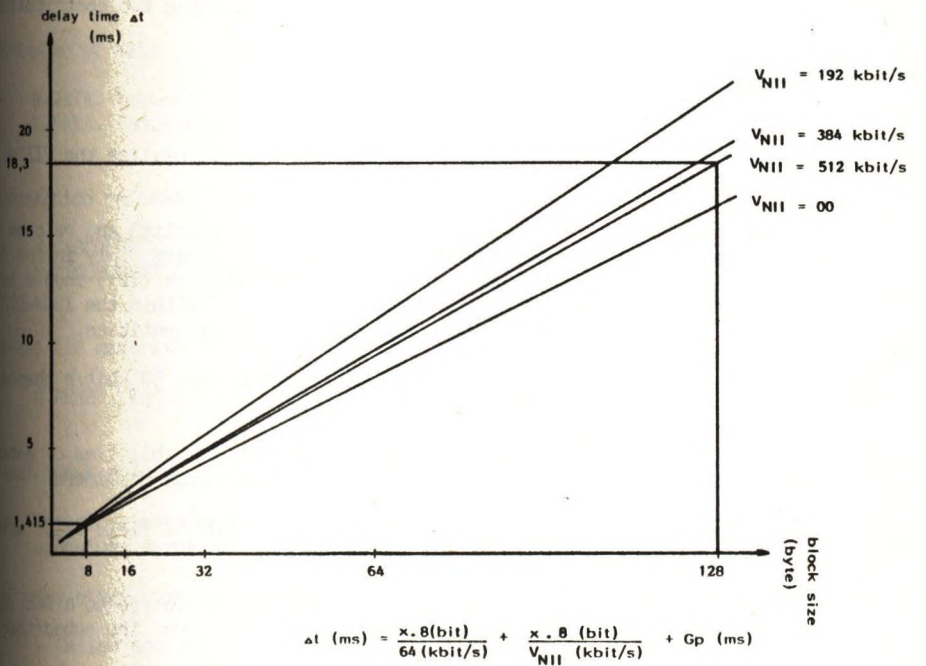


Fig. 8

The conclusion was: the decision on the speed to be used at the NII is dependent on

- the max. acceptable delaytime for voice terminals, and
- the wanted max. block size (larger block sizes require lower processor performance but higher speeds at the NII).

The results show the principal feasibility, but because of

- the high real-time requirements for speech and
- the wish to have a short time delay

a very sophisticated technology would be necessary for the realization of the NII.

5. ADDITIONAL IDEAS TO THE NII CONCEPT AND THE REACTION OF ISO

The Austrian Member Body proposed a New Work Item "NII" at the ISO/TC 97/SC6/WG5 meeting (March 1985), and besides this proposal 5 additional contributions from Austrian experts were presented at this meeting. The contents of 3 of them are more or less dealt with in items 2 to 4. The titles of the remaining two contributions are:

a)Proposal for an ISDN-NII Prototype
b)NII Prototype for the Austrian Videotex Service

to a) In this contribution a new method is described able to realize the NII with available technology on an economic basis.

The mechanism as described in Draft 3 (revised) to establish and release transparent connections between TEs would remain unchanged. Only in the data transfer phase of e.g. digitzed voice, an additional path (TP1) should be offered via the NII, which could be through-connected within the ISDN-DCE from the NII to the layer 1 without using any dynamical entities.

If there is a need of simultaneously using 2 transparent 64 kbit/s channels, 2 additional paths (TP1, TP2) will be necessary.

to b) In the public Austrian Videotex service 5 different possibilities of access to Videotex centres are planned for users and information providers.

For two of these possibilities, the PSTN and the PSPDN, a field trial using an NII Prototype (Subset of Draft 3) is planned.

The Videotex terminal will be connected via the NII Prototype to a DCE (BAG 1B) for the subscriber type 1B, or another DCE (BAG 2) for the subscriber type 2.

DCE "BAG 1B": Layer 3: Automatic calling (or manual via NII) for the PSTN to a Videotex concentrator. After connection establishment again automatic dialling, now according to X.25 (version 1980), will be done in order to get access to the Videotex centre.
Layer 2: Half duplex HDLC
Layer 1: 1200/1200 bit/s, synchronous, half duplex
DCE "BAG 2": Layer 3: Automatic calling according to X.25 (version 1980)
Layer 2: HDLC
Layer 1: "Voice plus data technique" in accordance with the CEPT Recommendation T/CD 01-013

The complete simple subset, which is necessary for Videotex application, is given in Table 1.

Interchange circuits: 2 wires for direction DTE —►DCE
2 wires for direction DCE —►DTE
1 wire for earth

Electrical characteristics: CCITT Rec. V.11
19,2 kbit/s, asynchronous

Mechanical characteristics: 9-pin subminiature connector
(ISO 4902, pin allocation not yet included)

TABLE 1: NII SUBSET FOR VIDEOTEX

Connection establishment

N-CONNECT. Request
N-CONNECT. Confirm

Transfer of data

N-DATA. Request
N-DATA. Indication

Connection release

N-DISCONNECT. Request
N-DISCONNECT. Indication

Reset (if generated by the network)

N-RESET. Indication

Additional rules:

- Any other service primitives (not included in this subset) or a missing mandatory parameter leads to a disconnection.

- Any other parameter than

  - Called address
  - Connection and point address
  - Reason for local reject
  - Reason (for RESET)

  will be ignored.

After the presentation of all contributions a long discussion took place at the ISO meeting about the architectural possibility to have a physical interface between layer 3 and layer 4.

It was stated that the NII concept, as described above, would not be in line with the OSI model. After an intensive exchange of arguments in favour of and against this view, the following "historical compromise" was found.

The description of the NII concept should be brought in line with the majority understanding of what would be allowed in the OSI model, but the contents of the proposal for the NII should be not changed.

Within the following text unanimously agreed upon for a New Work Item "NII" to be forwarded to the Sub-Committee approval procedure, Fig. 9 illustrates the new description of the NII-concept.

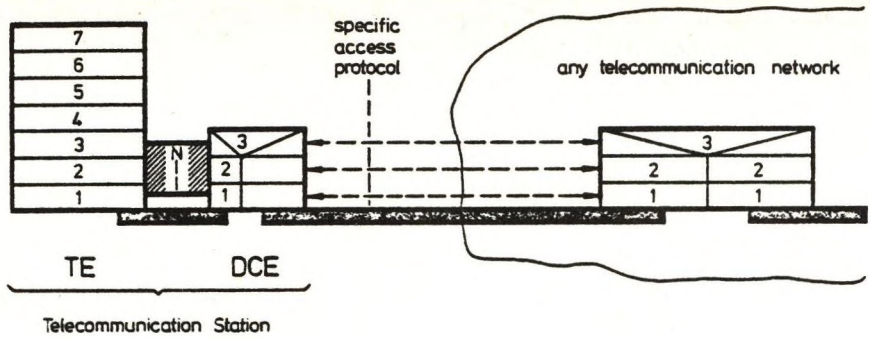


Fig. 9

6 DESCRIPTION OF DRAFT 4 (MAY 1985) OF THE NII

The main differences between Draft 3 and Draft 4 are the following:

- Draft 4 is brought in line with the majority understanding of the right architectural design of the NII.
- The question how to transfer digitized voice via the NII is solved in accordanc with an ISO/TC97/SC6/WG5 contribution as mentioned under 5 a).
- A contribution to the joint CEPT/Industry meeting (June 1985) with the title "Detailed analysis of the relationship between the ISDN user-network interface layer 3 protocol (Q.931) and the OSI network service (X.213)" is taken into account. (This analysis deals with stimulus and functional mode of operation.)

The contents of Draft 4 is the following:

Content

1. General

1.1 Scope
1.2 Definitions

2. Functional description

2.1 Layer 3
2.1.1 Non-voice services and signalling for voice services
2.1.2 Transparent transfer phase
2.1.3 Format and coding of the messages

2.2 Layer 2

2.3 Layer 1

3. Interface characteristics

3.1 Interchange circuits
3.2 Mechanical characteristics
3.3 Electrical characteristics

Appendix 1: Definitions
Appendix 2: Review of successful connection establishment
Appendix 3: Calling side: successful connection establishment
Appendix 4: Called side: successful connection establishment
Appendix 5: Format of the messages

The scope remains unchanged and reads as follows:

"This interface is designed to be applicable with already existing telecommunication networks (connection oriented or connectionless) as well as for forthcoming ISDNs (basic access) and LANs.

It is a simple interface which is independent of specific networks and makes terminals (TE) independent of specific network access protocols.

This interface is intended to be used only for short (some metres long) point-to-point configurations."

Fig. 10 gives a more detailed review of the NII between the TE and the DCE of a telecommunication station (TS). The network independent part (TE and a small part of the DCE) and the network dependent part are also shown in Fig. 10.

The DCE contains a mapping function which converts the service primitives of layer 3 into the protocol elements of the layer 3 protocol of the NII.

The service primitives are directly derived from the network service definitions as defined in CCITT Rec. X.213 (= IS 8348). Proposed additions to this network service definitions regarding ISDN are also taken into account. (Connectionless mode of operation is not yet included in this Draft 4.)

Layer 3 of the NII contains a simple protocol (elementary interactions). In layer 3 of the TE the corresponding mapping function which converts the protocol elements into the service primitives between layer 3 and layer 4 is included.

As the transmission error probability is expected to be extremely low, no transmission error correcting mechanism is needed in layer 2 of the NII. Only for the synchronous transmission via this interface a simple framing mechanism is needed.

For the short distance between TE and DCE no modem function is necessary in layer 1 of the NII, only input and output components for the physical media.

For voice services a transparent path (or 2 paths) will be established for digitized voice during the connection establishment phase (see also Fig. 10).

As an example, a successful connection establishment will be described more detailedly:

For this case 4 interactions are necessary, each consisting of an action and a local reaction. These actions and reactions will be executed via the NII by means of messages (protocol elements). The format is the same as given in Fig. 1.

First interaction:

A^1_c ... N-CONNECT REQUEST ACTION message (N-CO.RQ.A.m)

R^1_c ... N-CONNECT REQUEST REACTION message (N-CO.RQ.R.m)

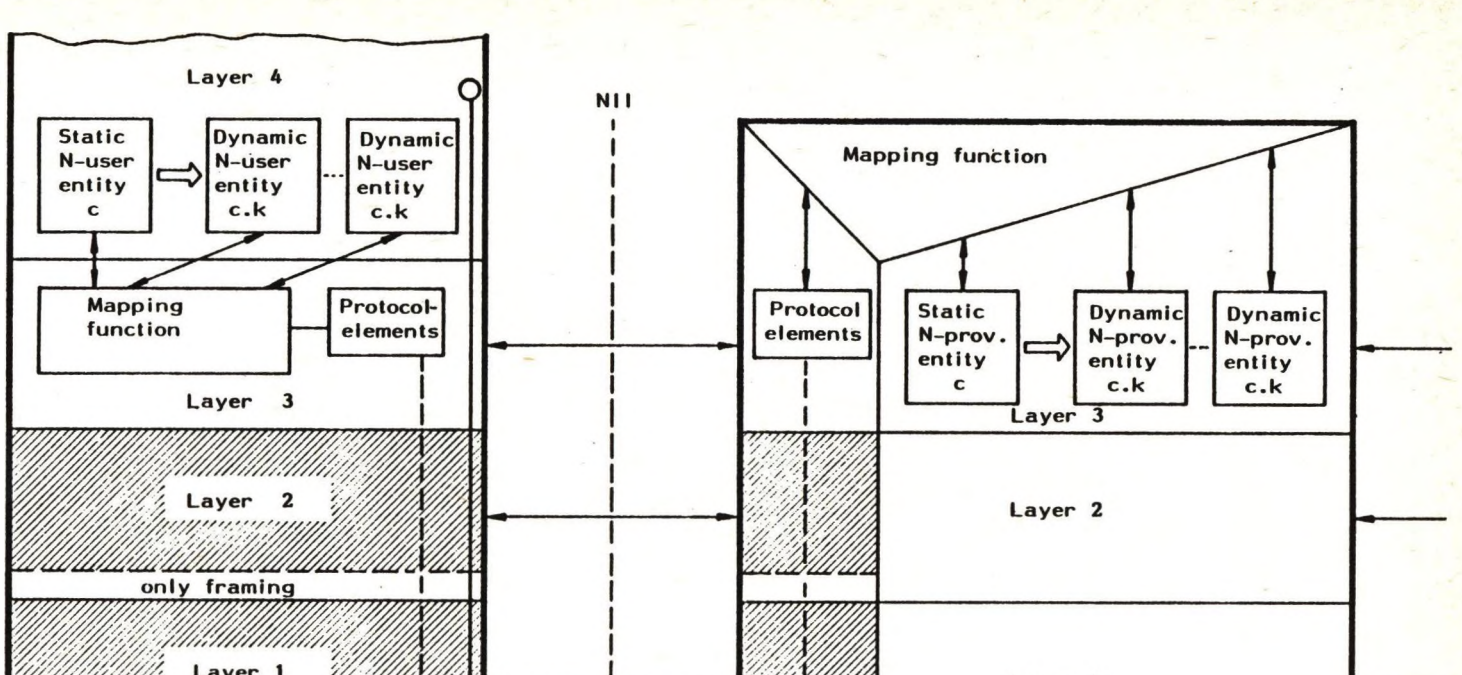

Layer 4
Static N-user entity c
Dynamic N-user entity c.k
Dynamic N-user entity c.k
Mapping function
Protocol-elements
Layer 3
Layer 2
only framing
Layer 1
Input / Output components
NII
Mapping function
Protocol elements
Static N-prov. entity c
Dynamic N-prov. entity c.k
Dynamic N-prov. entity c.k
Layer 3
Layer 2
Layer 1
transparent path for digitized voice
TE
DCE
TS
Network independent
Network dependent

Fig. 10

Second interaction:

A^2_c... N-CONNECT INDICATION ACTION message (N-CO.IN.A.m)

R^2_c... N-CONNECT INDICATION REACTION message (N-CO.IN.R.m)

Third interaction:

$A^3_{c,k}$... N-CONNECT RESPONSE ACTION message (N-CO.RP.A.m)

$R^3_{c,k}$... N-CONNECT RESPONSE REACTION message (N-CO.RP.R.m)

Fourth interaction:

$A^4_{c,k}$ ... N-CONNECT CONFIRM ACTION message (N-CO.CF.A.m)

$R^4_{c,k}$... N-CONNECT CONFIRM REACTION message (N-CO.CF.R.m)

Fig. 11 shows the first and second interactions, Fig. 12 the third and fourth interactions.

Fig. 13 is a more detailed figure of the calling side and Fig. 14 of the called side.

In case of connection establishment for ISDN service, 2 additional service primitives (which have not yet been included in the current version of CCITT Rec. X.213) are necessary. These service primitives are:

N-INFORMATION REQUEST
N-INFORMATION INDICATION

A complete list of necessary messages is given in Table 2.

TABLE 2:

1.	N-CONNECT REQUEST ACTION message	(N-CO.RQ.A.m.)
2.	N-CONNECT REQUEST REACTION message	(N-CO.RQ.R.m.)
3.	N-CONNECT INDICATION ACTION message	(N-CO.IN.A.m.)
4.	N-CONNECT INDICATION REACTION message	(N-CO.IN.R.m.)
5.	N-CONNECT RESPONSE ACTION message	(N-CO.RP.A.m.)
6.	N-CONNECT RESPONSE REACTION message	(N-CO.RP.R.m.)
7.	N-CONNECT CONFIRM ACTION message	(N-CO.CF.A.m.)
8.	N-CONNECT CONFIRM REACTION message	(N-CO.CF.R.m.)
9.	N-DISCONNECT REQUEST ACTION message	(N-DI.RQ.A.m.)
10.	N-DISCONNECT REQUEST REACTION message	(N-DI.RQ.R.m.)
11.	N-DISCONNECT INDICATION ACTION message	(N-DI.IN.A.m.)
12.	N-DISCONNECT INDICATION REACTION message	(N-DI.IN.R.m.)
13.	N-INFORMATION REQUEST ACTION message	(N-IF.RQ.A.m.)
14.	N-INFORMATION REQUEST REACTION message	(N-IF.RQ.R.m.)
15.	N-INFORMATION INDICATION ACTION message	(N-IF.IN.A.m.)
16.	N-INFORMATION INDICATION REACTION message	(N-IF.IN.R.m.)

Review on successful connection establishment

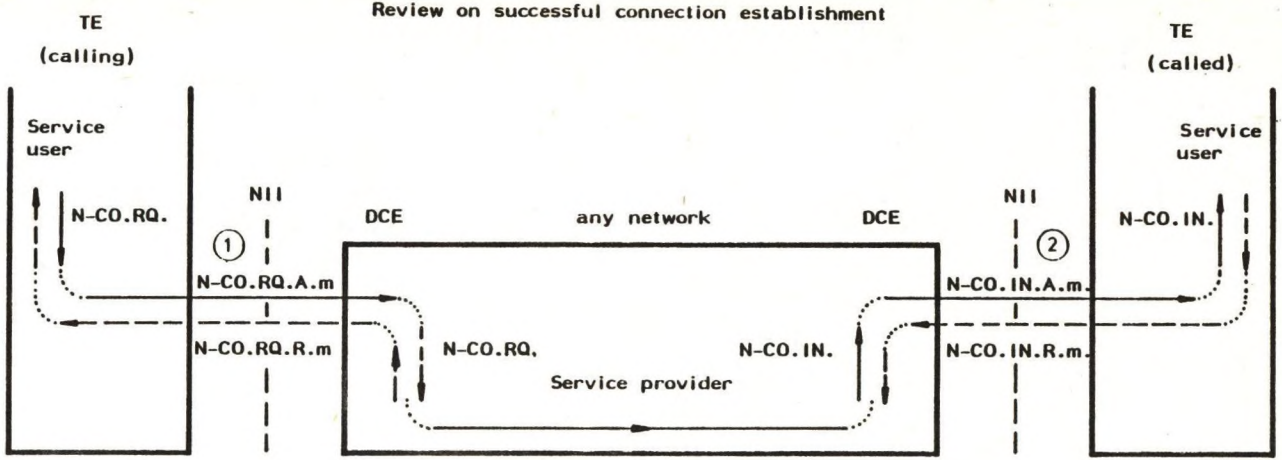


Fig. 11

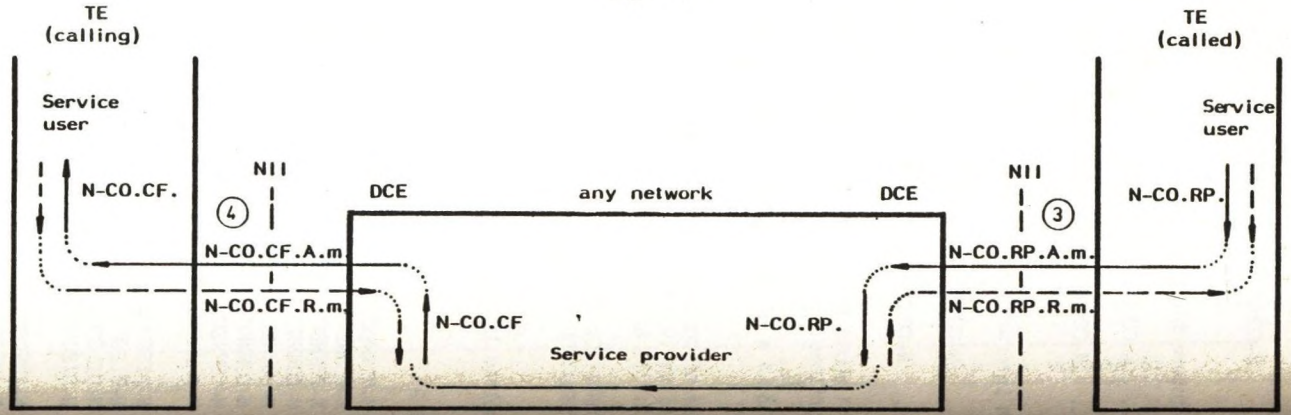


Fig. 12

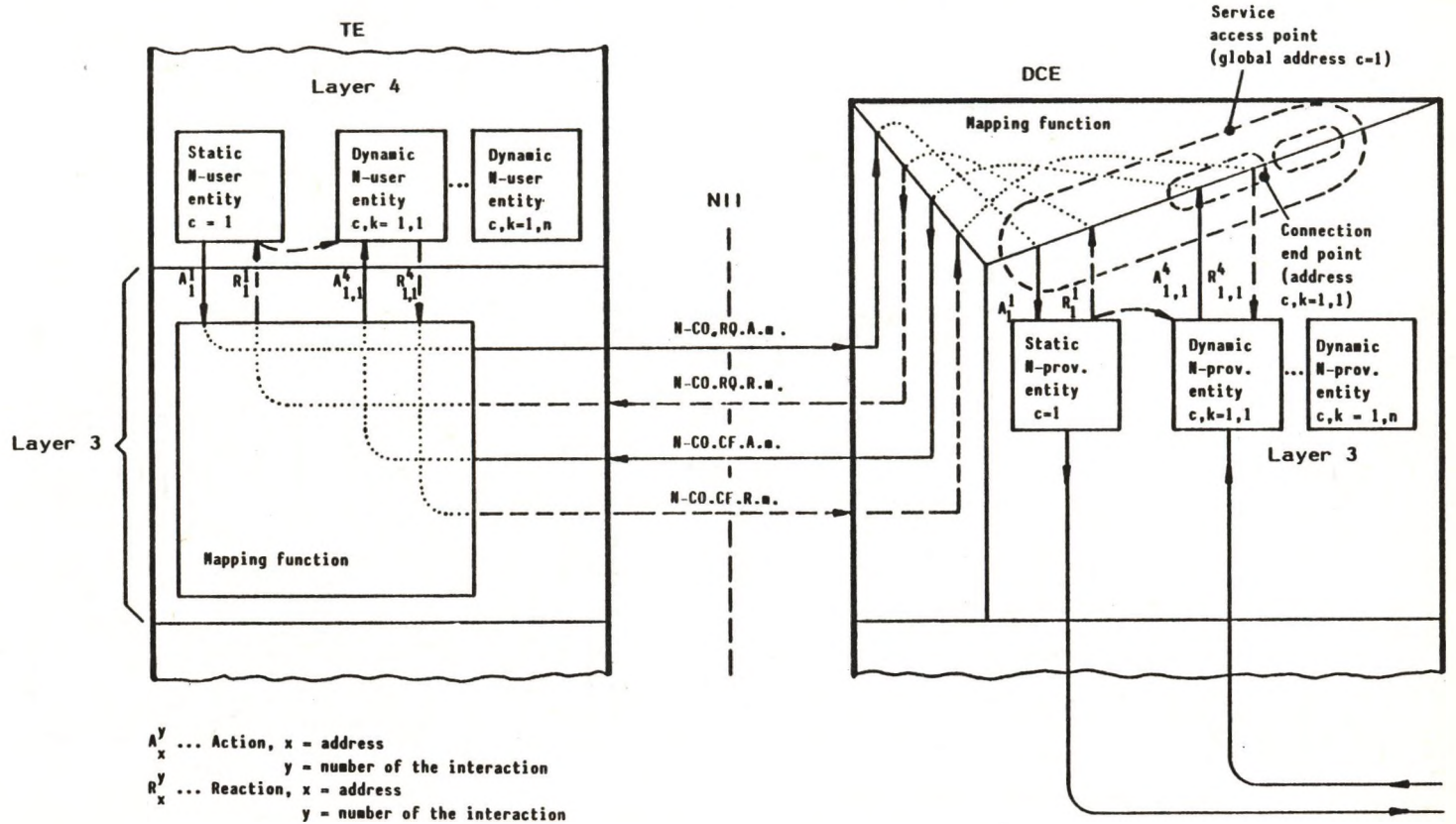

TE
Layer 4
Static N-user entity c = 1
Dynamic N-user entity c,k= 1,1
Dynamic N-user entity c,k=1,n
Layer 3
Mapping function
NII
N-CO.RQ.A.m.
N-CO.RQ.R.m.
N-CO.CF.A.m.
N-CO.CF.R.m.
DCE
Service access point (global address c=1)
Mapping function
Connection end point (address c,k=1,1)
Static N-prov. entity c=1
Dynamic N-prov. entity c,k=1,1
Dynamic N-prov. entity c,k = 1,n
Layer 3
A^y_x ... Action, x = address
y = number of the interaction
R^y_x ... Reaction, x = address
y = number of the interaction

Fig. 13

Called side: Successful connection establishment

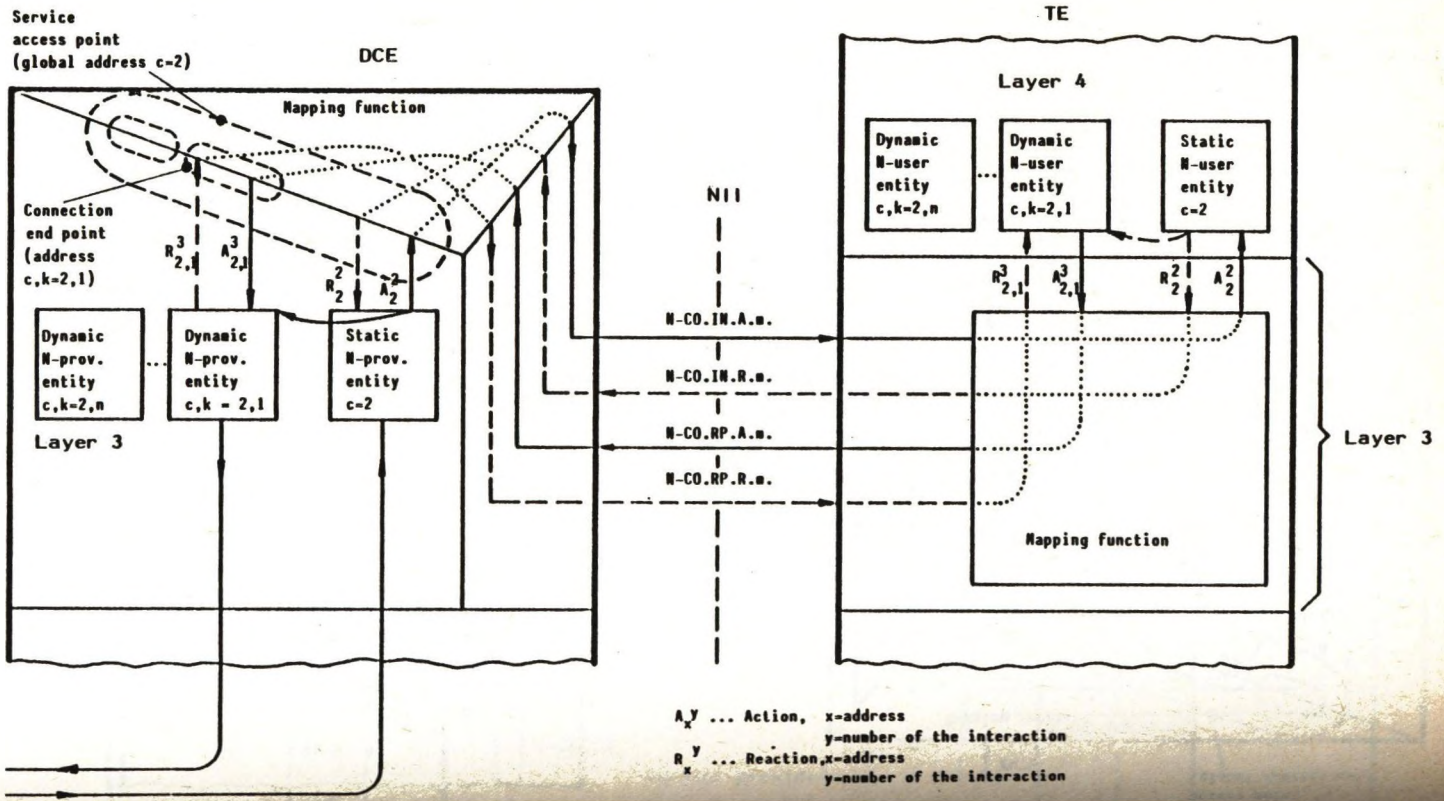


Fig. 14

17.	N-SUSPEND REQUEST ACTION message	(N-SU.RQ.A.m.)
18.	N-SUSPEND REQUEST REACTION message	(N-SU.RQ.R.m.)
19.	N-SUSPEND INDICATION ACTION message	(N-SU.IN.A.m.)
20.	N-SUSPEND INDICATION REACTION message	(N-SU.IN.R.m.)
21.	N-SUSPEND CONFIRM ACTION message	(N-SU-CF.A.m.)
22.	N-SUSPEND CONFIRM REACTION message	(N-SU.CF.R.m.)
23.	N-RESUME REQUEST ACTION message	(N-RU.RQ.A.m.)
24.	N-RESUME REQUEST REACTION message	(N-RU.RQ.R.m.)
25.	N-RESUME INDICATION ACTION message	(N-RU.IN.A.m.)
26.	N-RESUME INDICATION REACTION message	(N-RU.IN.R.m.)
27.	N-RESUME CONFIRM ACTION message	(N-RU.CF.A.m.)
28.	N-RESUME CONFIRM REACTION message	(N-RU.CF.R.m.)
29.	N-REARRANGE REQUEST ACTION message	(N-RR.RQ.A.m.)
30.	N-REARRANGE REQUEST REACTION message	(N-RR.RQ.R.m.)
31.	N-RESET REQUEST ACTION message	(N-RT.RQ.A.m.)
32.	N-RESET REQUEST REACTION message	(N-RT.RQ.R.m.)
33.	N-RESET INDICATION ACTION message	(N-RT.IN.A.m.)
34.	N-RESET INDICATION REACTION message	(N-RT.IN.R.m.)
35.	N-RESET RESPONSE ACTION message	(N-RT.RP.A.m.)
36.	N-RESET RESPONSE REACTION message	(N-RT.RP.R.m.)
37.	N-RESET CONFIRM ACTION message	(N-RT.CF.A.m.)
38.	N-RESET CONFIRM REACTION message	(N-RT.CF.R.m.)
39.	N-DATA REQUEST ACTION message	(N-DA.RQ.A.m.)
40.	N-DATA REQUEST REACTION message	(N-DA.RQ.R.m.)
41	N-DATA INDICATION ACTION message	(N-DA.IN.A.m.)
42.	N-DATA INDICATION REACTION message	(N-DA.IN.R.m.)
43.	N-DATA ACKNOLEDGE REQUEST ACTION message	(N-DK.RQ.A.m.)
44.	N-DATA ACKNOLEDGE REACTION message	(N-DK.RQ.R.m.)
45.	N-DATA ACKNOLEDGE INDICATION ACTION message	(N-DK.IN.A.m.)
46.	N-DATA ACKNOLEDGE INDICATION REACTION message	(N-DK.IN.R.m)
47.	N-EXPEDITED DATA REQUEST ACTION message	(N-ED.RQ.A.m.)
48.	N-EXPEDITED DATA REQUEST REACTION message	(N-ED.RQ.R.m.)
49.	N-EXPEDITED DATA INDICATION ACTION message	(N-ED.IN.A.m.)
50.	N-EXPEDITED DATA INDICATION REACTION message	(N-ED.IN.R.m.)

A comparison between existing interface concepts and the NII concept shows the advantages of this new interface.

Subject	Current interfaces	NII
No restrictions to the plannings of the network provider	Modifications of the network characteristics affect the DTE, therefore interim regulations/conversion facilities are re-	Modifications of the network would not affect the DTE, but only the DCE. Since the subscribers are not involved, the network would not be impeded.

Subject	Current interfaces	NII
	quired in the network: Evolution impeded by subscriber considerations.	
Competence: Network provider-subscriber	Difficulties with respect to maintenance, fault location, ownership problems becauce of theoretical boundary, which has not been technically realized.	Administrative boundary could be identical with the technical interface, therefore no conflicts of competence.
Flexibility, portability of the terminal equipment	Each terminal equipment has been designated for a certain network. The subscriber can choose another network if he is willing to meet the costs for additional conversion expenditure.	The terminal equipment could be used universally in connection with each network; in each case, however, the proper DCE would have to be used.
Transistion of existing terminal equipments/ services into the ISDN	Connection of existing DTEs (X.21,X.25..) via individual terminal adapters(TA). The TAs are costly, as the three lower layers must be used twice including a mapping between them.	NII-DTEs could be connected without any change. Only one type of DCEs with the S-interface would be needed.
Changes and/or enhancements of the S-interface	In spite of the fixed attitude of some countries, the international pressure can lead to changes of the S-interface. As a consequence the DTEs and DCEs have to be modified.	Changes of the S-interface would necessitate changes only in the DCEs, but not in the DTEs.
Promotion of new services	For Teletex (operating at 64 kbit/s), mixed mode and facsimile group 4 no proper network is available, therefore no market for such terminals. Existing service are affected by a future promise of new ISDN services.	NII-DTEs, as prepared for future use, could be connected to existing networks and thereby provide for a readily available subscriber's potential for ISDN.
Approval procedure	Each modification of a communication- and/or application part necessitates a repetition of the total approval test.	Due to the splitting up into the DTE and the DCE approval, the required test procedures would be simplified and partly dispensable.

REFERENCE:
[1]H.J. Burkhart, K.J.Eckert and T.H. Krecioch, Computers & Standards, The International Journal, Volume 2, No. 1, 1983.

Packet Switching in the Federal Republic of Germany (with a View to the Situation in CEPT-Countries)

Hermann G. Gabler

Deutsche Bundespost, Fernmeldetechnisches Zentralamt
Text and Data Communications Department
Federal Republic of Germany

SUMMARY

The Deutsche Bundespost introduced its public packet switched data network in 1980. The paper reports on the status of DATEX-P after four years of operation and gives a survey on the services offered, on the traffic characteristics arising from about 10 000 ports and on network performance (including data on availability, message transfer delay and throughput). The paper tackles the international traffic relations and touches specifically the situation in Europe. Some statements concern future planning. The essential characteristics of the envisaged next generation packet switch are highlighted.

1 INTRODUCTION

The integrated data network (IDN) in the Federal Republic of Germany is a communication network which combines data and text networks in common equipment of the PTT. In the network section DATEX-P, packet-switched data connections are established. The following sections provide an overview of the current status of the DATEX-P network, a preview of further expansion plans of the Deutsche Bundespost and a view to packet switching networks in Europe.

2 TECHNOLOGY

The DATEX-P network was taken into service in 1980 after successful completion of a pilot project. Transmission is based on the PCM technology which is used generally in the IDN. 64 and 128 kbit/s channels are used between the exchanges; in the lower network level, additional multiplexes ZD-A2 or ZD-A3 are used for connection of transmission speeds of 9.6 kbit/s. The switching computers in the network are type SL-10 computers of Northern Telecom Limited, whose offer in an international tender in 1979 came closest to the existing requirements. Concentrators are not yet used for switching purposes. Network management tasks are handled in higher level centers. In the DATEX-P network, these functions are split between the network control center (NCC, 2 SL-10 nodes) in Düsseldorf and the data collection center (DCC) in Darmstadt (Figure 1).

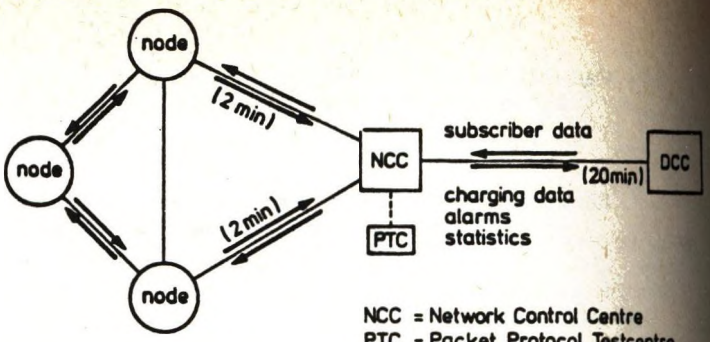


Figure 1
Structure of the
DATEX-P network

The NCC collects alarms, statistical data, and charging information and passes this at regular intervals to the DCC for further processing. As the most important information is displayed continously at the control position of the NCC, a comprehensive overview of the status and current load of the network is available there at all times. In the case of a fault, immediate intervention can be carried out in the NCC, in many cases without the use of local personnel, because all operator functions can be executed both from the NCC and from the exchange itself.

International traffic is split between three exchanges in order to spread the load and increase reliability.

In addition to the two computers for DCC functions, the network currently consists of 35 exchanges in 17 locations (Figure 2).

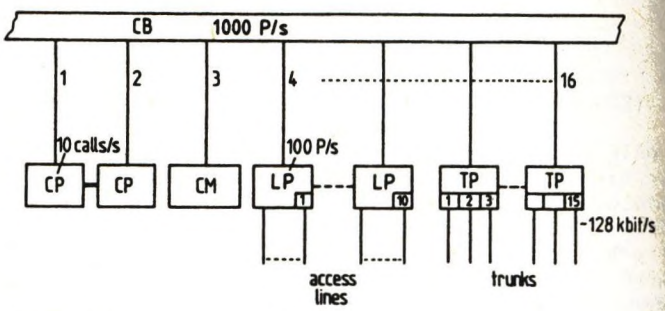


Figure 3
DATEX-P
Structure of an
SL-10 node

CB Common Bus — 4 × 48 kbit/s
CP Control Processor — 2 × 48 kbit/s + 62 × (≤) 9,6 kbit/s
CM Common Memory — 124 × (≤) 9,6 kbit/s
TP Trunk Processor
P/s Pakets per sec

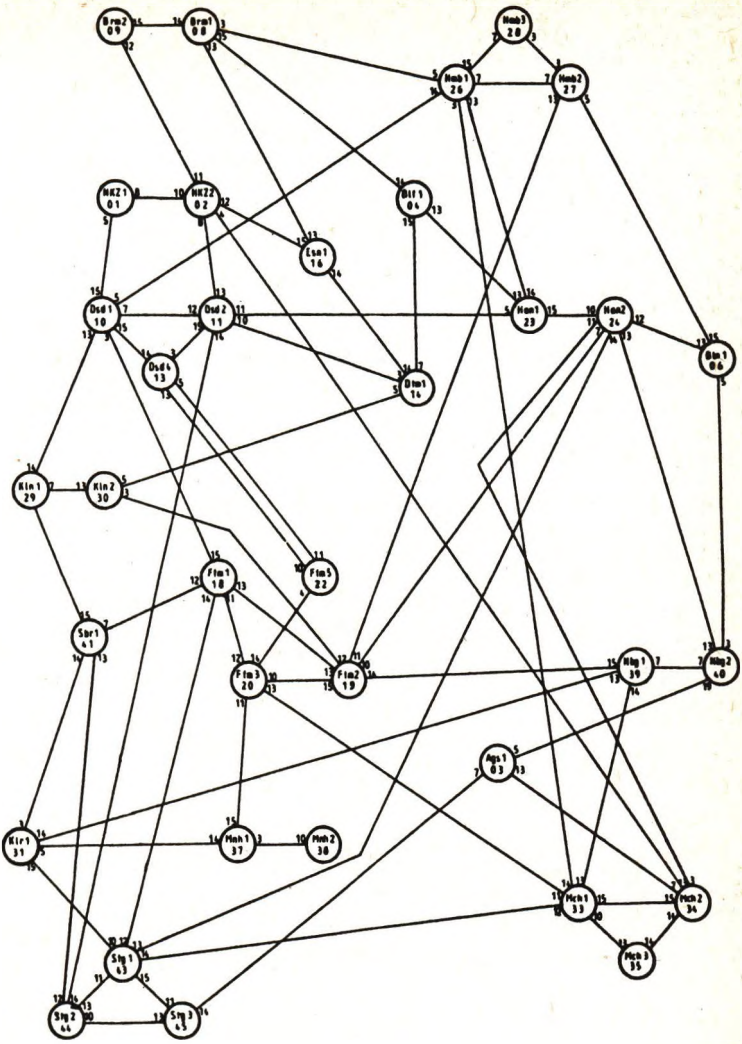


Figure 2
Network configuration DATEX-P
Status April 1985

The SL-10 technology (Figure 3) consists of a multiprocessor system whose modular structure makes it possible to match it easily to the existing application and the required configuration. Since introduction of the system, the manufacturer has carried out various improvements in the hardware and software which, together with a large number of optimization steps of the FTZ, have led to a major increase in the dynamic load carrying ability. Instead of the originally planned 170 ports, newest results have shown that in the current subscriber and traffic structure, approximately 500 ports (400 X.25, 100 asynchronous) can be handled by one exchange.

3 SERVICES

In the provision of its services, the Deutsche Bundespost complies as closely as possible with the recommendations of CCITT. This offers the best guarantee for compatibility of the data terminal equipment used throughout the world and greatly simplifies the international connection of such complex communications networks. However, innovations in the standardization also occassionally require modifications in the existing terminals. These modifications, although only slight, are apparently underestimated by the device manufacturers. Although, for example, the changes have been published in the interface recommendation X.25 since the Plenary Assembly of CCITT in 1980, and although the Deutsche Bundespost, in October 1983 - more than one year before introduction into the network, has informed the manufacturers of the necessity of modification, timely conversion has still caused problems. The Deutsche Bundespost is often in a difficult situation. On the one hand, it is under pressure to introduce internationally agreed innovations into its networks while, on the other hand, it must take into account the effects on thousands of its customers.

This short overview does not permit a description of all performance features which are offered. Table 1 provides basic information on these features; it shows the performance features of the basic service DATEX-P10, i.e. for connections between DATEX main stations which comply with interface recommendation X.25. Generally, these will be packet-oriented data terminal equipment. The packet facility can also be achieved by an intermediate device (called a black box, protocol converter, or similar).

Table 1: Facilities DATEX-P10 (X.25)

legend (x) = opening date not yet defined	on a per-call basis	for a contractual period
signalling rate of a main station 2400/4800/9600/48000 bit/s		x
User data in call request packet	x	
Connection type - switched virtual call (SVC) - permanent virtual call (PVC)	x	 x
single/multi logical channels		x
Number of logical channels - for switched and permanent virtual calls - for incoming, outgoing and two way virtual calls		 x x
Nonstandard window sizes (packet level)	(x)	x
Closed user group		x
Subaddresses		x
Reverse charching		x
Hunt group		x
Throughput class negotiation	(x)	
D-Bit	x	
Nonstandard packet sizes	(x)	(x)

Before data terminals can be connected to the network, they must be officially approved. For this, compliance with the interface conditions is checked with an automatic test system. Before approval is applied for, the manufacturers can have the terminals tested, in agreement with the FTZ, with a similar test system PETRUS. Previous experience has shown that newly developed X.25 data terminal equipment has only slight chances, without previous implementation tests with the FTZ of passing the technical approval tests. The test conditions of the system PETRUS were set up in cooperation with manufacturers and are continously refined. (Reference is made to a companion paper to COMNET 85 on conformance testing.)

4 SUBSCRIBER; UTILIZATION OF THE SERVICES

Figure 4 shows the increase in the number of subscribers up to date and the expected increases in the future.

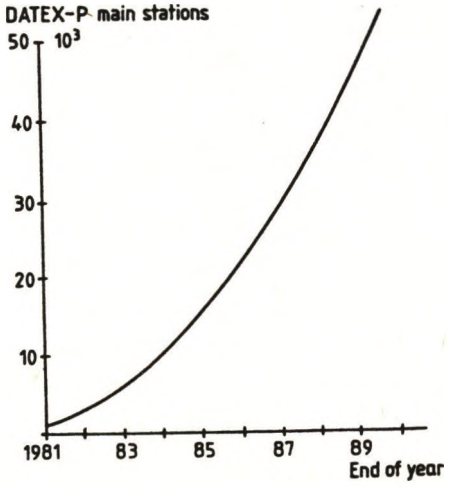


Figure 4
Growth rate of
DATEX-P main stations

Currently, there are approximately 10,000 main attachments, and this figure is increasing at approximately 5% per month in accordance with the forecasts, which means that no further connection bottlenecks exist, with the exception of occasional local problems. In addition to subscriber mainstations, many subscribers (the number is not known precisely but is estimated to be approximately 4000) use the packet switched network via the telephone network or the circuit-switched data network. Furthermore, over 1000 network user identifications (NUI) have been allocated for terminals which operate with interface recommendation X.28. The great number of ports per exchange, which lies far above the original manufacturer specifications, requires careful observation of the traffic behavior of the users. Figure 5 shows that the traffic variations in the last 12 months have certainly remained within manageable limits. The slight decrease in traffic per port is due to efforts of terminal manufacturers and subscribers who optimize their applications according to the properties of a packet switching network.

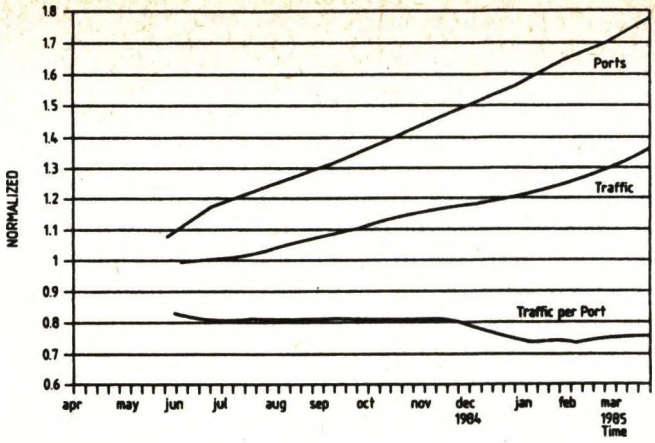


Figure 5
Traffic variations
in DATEX-P

DATEX-P is used in extremely different manners by the various user groups. 85% of the users generate only 25% of the total traffic of the network. A good proportion of the terminals generate only extremely weak traffic, i.e. there is often no traffic at all for several days at a time (Figure 6).

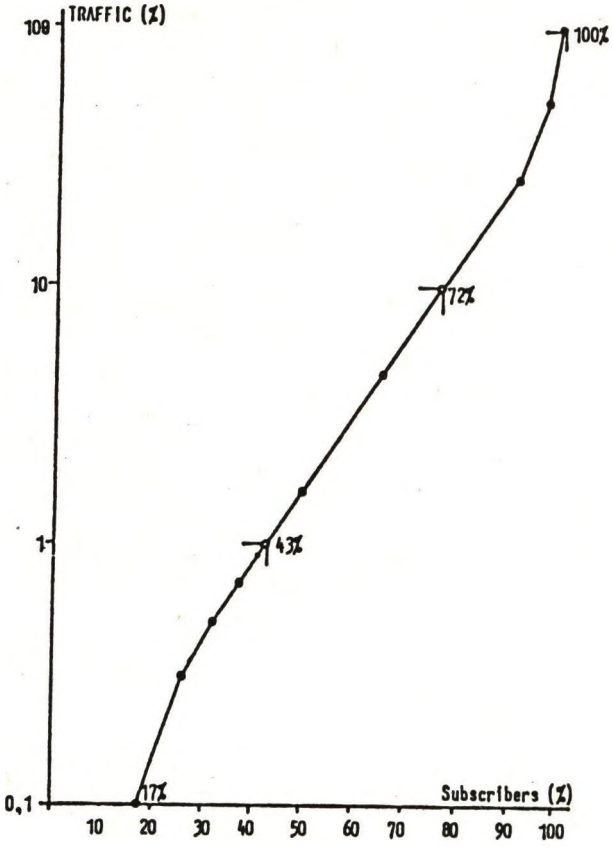


Figure 6
Distribution of
traffic volume

In contrast to the traffic structure in the public packet-switched networks of other countries, the proportion of X.25 main stations, with 84%, is relatively high in DATEX-P. Table 2 shows the distribution of the number of main stations per service and signalling rate.

Table 2

Number of main stations per service and signalling rate

service	signalling rate	distribution of main stations
main stations:		
P10H (X.25 terminals)	2400 bit/s	25,5 %
	4800 bit/s	32,0 %
	9600 bit/s	42,0 %
	48000 bit/s	0,5 %
	Total	84,0 %
P20H (X.28 terminals)	110-300 bit/s	
	1200 bit/s	
	75/1200 bit/s	11,0 %
P32H IBM 3270-type terminals	2400 bit/s	
	4800 bit/s	
	9600 bit/s	
	Total	4,0 %
P42H	1200 bit/s	
	2400 bit/s	
	4800 bit/s	
	Total	0,1 %
Total number of main stations		ca. 12 000

64% of the X.25 main attachments use switched virtual calls (SVC); the remainder - and this number is decreasing - use permanent virtual calls (PVC). The daily traffic curve (Figure 7) is an important parameter for correct dimensioning of the network.

The number of logical channels per main station depends greatly on the signalling rate and varies between an average of 31 with 48 kbit/s and 2.2 with 2.4 kbit/s. On average, there are 4.4 logical channels per port in DATEX-P.

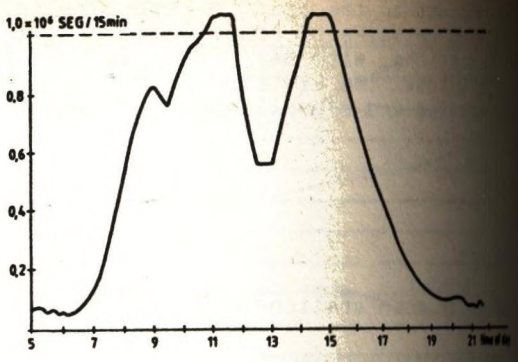


Figure 7 Daily traffic distribution (averaged) in DATEX-P for a business day (Nov. 84) Main traffic hour = 12.5% of the total traffic per day

5 NETWORK BEHAVIOR

AVAILABILITY

The availability of the DATEX-P network has, in the past, occasionally given rise ro justified user complaints. The incomplete redundancy of the switching technology had an unfavorable effect in the case of system faults. At the end of 1983, after careful analysis of all operational and technical weaknesses, and with generous support from the manufacturer of the switching technology, the availability of the DATEX-P network was considerably improved. In the first quarter of 1985 the average node availability has remained in the order of 99.99%.

NETWORK PERFORMANCE

Results of measurements of the most important network performance parameters have been published in /2/.

The DATEX-P network is designed such that there are normally never more than three transmission sections. This makes it possible to keep the network transit time low. This network transit time is defined as the time from complete reception of a data block in the originating exchange until complete storage of this block in the output buffer of the receiver's attachment line in the destination exchange.

Table 3 shows the network transit times measured in the DATEX-P network for data packets with 128 octets of user data as a function of the number of hops used within the network.

Table 3

number of hops	network transit time			
	Ø	min	max	95 %
1	110	74	172	≤ 141
2	126	103	181	≤ 144
3	141	132	197	≤ 162
4	173	158	224	≤ 193

Table 3: Network transit time of data packets in the DATEX-P network

Setting up of a call is a special form of the dialog between the participating data terminal equipments. Table 4 shows the relationship between the call setup times and the number of hops used within the network.

Table 4

number of hops	call set up times			
	Ø	min	max	95 %
1	340	272	494	≤ 405
2	380	319	644	≤ 427
3	400	332	643	≤ 448
4	418	384	465	≤ 461

Table 4: Call set up times in the DATEX-P network

New users in the DATEX-P network, who have transferred their application from the leased line network of the HfD into the switched network, often complain about the reduced throughput. The problem lies in the additional "intelligence" of the packet-switched network in comparison with the "unintelligent" lines of the HfD. A typical example of this is the fact that processing programms generally release user data blocks which are considerably larger than the maximum packet lengths used in the DATEX-P network. These are therefore divided down into the permissible packet lenghts of 128 octets before transmission and assembled again in the receiving DTE before they are passed to the processing program in this equipment. In order to improve the transmission reliability, the transmitting DTE often requests an aditional acknowledgement from the receiving DTE for a transmitted user data block. Only after this has ben received, is the next block released for transmission. This corresponds to a window size w = 1 for user data blocks. As further transmission

is stopped while the equipment is waiting for the confirmation, the possible throughput is reduced. The reduction factor increases as the size of the user data blocks decreases, i.e. as the number of times where the equipment has to wait for an acknowledgement increases. The following table 5 shows the dependency of the throughput on the size of the user data blocks.

length of user data blocks Octets	throughput DP/s
256	2.13
512	3.94
1024	4.59
2048	5.23

Table 5: Throughput as a function of user data block length

Measuring conditions for all measured values shown:

- TANDEM-R16 computer as data source
- TANDEM-R16 computer as data sink
- Window size w = 2 (packet level)
- signalling rate 9600 bit/s
- Full data packets (128 octets)
- One internal hop in the DATEX-P network

6 INTERNATIONAL TRAFFIC

More than 10% of the total traffic in DATEX-P is international. In the recent past, small and industrially less developed countries have been attempting, with the aid of packet switching technology, to obtain access to the information networks of the industrial countries. Table 6 shows the current status of the possible traffic connections between the Federal Republic of Germany and other countries.

Table 6: DATEX-P overview
Traffic connections with foreign countries (status March 1985)

Countries which can be reached via a transit country are marked with *.

Egypt *	Gabun *	New Zealand *
Alaska *	Greece *	Norway
Argentina *	Great Britain	Austria
Australia *	Guam *	Panama *
Bahamas *	Hawai *	Philippines *
Bahrain *	Hong Kong *	Portugal *
Barbados *	Ireland *	Puerto Rico *
Belgium *	Israel *	Reunion *
Bermuda *	Italy	Sweden
Brazil *	Japan	Switzerland
Chile *	Virgin Islands *	Singapore
Costa Rica *	Canada	Spain
Denmark	Camerun *	South Africa *
Dominican Republic *	Columbia *	Taiwan *
Ivory Coast *	Korea (Republic) *	Thailand *
Finland	Luxembourg	Trinidad and Tobago *
France	Morocco *	Tunesia *
French Antilles *	Mexico *	United Arab Emirates *
French Polynesia *	Netherlands	USA

Generally, however, it should be noted that the DBP initiates traffic with each new country as soon as it is capable of handling this traffic. If direct connection of the networks is not possible, the connection is switched through a transit country.

SITUATION IN EUROPE

Nearly all West European countries have implemented packet switching in their networks. Table 7 shows, to the best knowledge of the author, the present situation as far as the number of terminals and the vendors of switching equipment are concerned. Through the CEPT organization, most European administrations are in close contract and intend to harmonize their services. An important target of this harmonization work will be the mutual acknowledgement of terminal approvals. A prerequisite for these endeavors is to agree on the procedures to be employed for terminal conformance tests.

PUBLIC PACKET SWITCHING NETWORKS IN EUROPE
(Installed ports, year end 1984)

Country	Service	System(s)	Manufacturer	Ports	Carrier
Austria	Inter-nat.'I	EDX-P	SIEMENS	160	RADAUS
Austria	DATEX-P	SL-10	NT	1000	ÖPTV
Belgium	DCS	EDX-P	SIEMENS	3928	RTT
Denmark	DATAPAK	TP-4000	GTE	400	PT
Finland	DATAPAK	DPS-25, TP-4000	BTMC, GTE	1000	PTT, HTV
France	TANSPAC, NTI	CP-50	SESA, TRT	21000	PTT
Germany F. R.	DATEX-P	SL-10	NT	13000	DBP
Great Britain	PSS, IPSS	TP-4000 DPS-25	BTMC, GTE	13000	BT
Greece	HELPAC		SESA	100	PTT
Italy	ITAPAC	EDX-P	SIEMENS	2500	SIP, PT
Ireland	EIRPAC	SL-10	NT	300	PTT
Island	ISPAC	Eripac	LME	300	PTT
Luxem-bourg	LUXPAC	DPS-25	BTMC	300	PTT
Nether-lands	DATA-NET 1	DPS-1500	BTMC	1400	PTT
Norway	NORPAK	TP-4000	GTE	1000	TV
Portugal	TELEPAC	SL-10	NT	1000	CTT
Spain	IBERPAC, NID	HW716 TESYS	HONEYWELL SECOINSA	5500	CTNE
Sweden	TELEPAK	Tymnet Engine	TYMNET	1300	TV
Switzer-land	TELEPAC	SL 10	NT	500	PTT
Yugos-lavia (Slov.)	ERIPAC		LME	200	PTT
			TOTAL	69000	

7 FURTHER EXPANSION OF THE DATEX-P NETWORK

MEDIUM-TERM EXPANSION PLANS

The current technical status of the switching technology SL-10 permits, from the current point of view, expansion of the network to at least approximately 30,000 ports. Due to the increasing number of hops in a connection,however, it is necessary to reduce the number of exchanges which are involved in a connec-

tion. In the next few months, the DBP will start to form regions consisting of several nodes ("cluster formation"). The regions will then form a second network layer.

LONG-TERM EXPANSION PLANS

During the introduction of the SL-10 technology, it was already laid down that this system, in step with the technological advances, should be replaced by a more powerful technology when a level of approximately 13,000 users was reached. The system SL-10 proved to be more powerful than originally assumed, which meant, particularly with respect to the considerable introduction costs for new technology, that rapid replacement by new technology was not necessary. Due to the decision of the Deutsche Bundespost to implement the packet switching technology only in a second development step of ISDN, the requirement with respect to the capacity of the following system is defined. In order to cover the required number of ports until at least 1995, the system must be capable of handling the traffic of 150 - 200,000 users. Company tenders for a subsequent system are currently being evaluated. The new system is intended to be used in further network expansion starting in approximately 1987.

The following list shows the most important performance features which the Deutsche Bundespost must demand from this system.

- Switching technology	
Capacity	10,000 access lines 30,000 packets/second
Availability	99.997% (15 minutes/year total failure permitted)
Design	Modular construction which permits economical use even with small numbers of users Small space requirements Low power consumption
Operation	24 hours per day System modifications without interruption of services
- Network	Hierarchical structure Concentrator technology 2 Mbit/sec trunks
	Dynamic routing Measures for overload handling
- Network interface	Cooperation with the existing part of the DATEX-P network Compatibility with the existing services

LITERATURE

1 User handbook Dx-P

2 "Delay and Throughput Analysis for the German Packet Switched Public Data Network DATEX-P" by K. P. Steinruck, G. Gutmerow, Globecom '84 IEEE Global Telecommunications Conference, Nov. 1984, Atlanta, USA.

THE ISDN CONCEPT OF THE DEUTSCHE BUNDESPOST AND THE INTEGRATION OF NON-VOICE-SERVICES

Heinz Kunze

Deutsche Bundespost
Fernmeldetechnisches Zentralamt
Darmstadt
(F.R.G.)

Summary

Many administrations and organizations are concerned today with the idea of devising a concept for an Integrated Services Digital Network (ISDN). The CCITT makes every effort to adopt appropriate recommendations still during its current study period (1981 - 1984). In its capacity as the Telecommunications Administration of the Federal Republic of Germany, the Deutsche Bundespost has resolved to introduce an ISDN and defined its configuration for the initial phase. The equipment required will be tested as a pilot project, prior to its series-production and delivery to the Deutsche Bundespost.

This early phase of ISDN implementation will be followed by further development stages, in which the network facilities and range of services offered are planned to be extended. In the context of that expansion, the change from 64 kbit/s services to broadband services is also intended to take place. At present, the centre of all activities is doubtlessly the first phase, whose characteristic features will be described in the paper.

1 The way to the ISDN

The Deutsche Bundespost operates a variety of telecommunication networks in its service area - for example, a telephone network which also allows the transmission of a large amount of data traffic. In addition, it runs a circuit-switched data network (called DATEX-L) and a packet-switched data network (termed DATEX-P), combined in the integrated data network (IDN). Also the largest telex network the world is operated by DBP in this IDN. All these networks are integrated insofar als they make use of joint transmission systems. However, the switching, signalling and subscriber line equipment is service or network specific. The network structure therefore differs according to the traffic requirements to be fulfilled.

Depending on the range of services expected from an ISDN, any of these networks could form the starting point for an Integrated Services Digital Network, but a prerequisite for the evolution of the ISDN from the telephone network is the decision of the Deutsche Bundespost to digitize all the telephone transmission and switching equipment in the future. Although this resolution was originally

not influenced by the idea of an ISDN, it nevertheless allows the considerations of establishing an ISDN to be centred on the subscriber-line area, including the local exchange, thus assuming the transit exchanges and the transmission equipment between them to be 'existing'. The only new ISDN element to be introduced into the trunk area is the common channel signallig system with an ISDN user part (CCITT NO 7 with ISDN UP).

Based on this concept, an ISDN will emerge, in principle, from the modernization of the telephone network. Since, from the present point of view, the digitization of the subscriber-line area does not seem economically justifiable for a single telephone station without special supplementary services (i.e. only basic function 'telephony'), the digital local exchange can be equipped with two types of subscriber lines: an analogue access for the single-telephone user and an ISDN access line of initially two variants for customers who want to participate in serveral services, including two times telephony simultaneously (figure 1). The way to the ISDN thus is clearly defined. From this also follows the concept that the first ISDN phase must be oriented toward a transmission speed of 64 kbit/s derived from the speech coding standard according to the PCM laws as recommended by CCITT.

2 Services in the ISDN

As already mentioned, the ISDN concept of the Deutsche Bundespost will be 64 kbit/s-based in the first phase. In other words, it will initially cover those services which require a transmission speed of up to 64 kbit/s. This is, first of all, the telephone service. The possibilities of improving the comfort of the telephone service by network-supported subscriber facilities are shown in figure 2. This figure indicates all supplementary services which are of interest for subscribers. This figure shows some results of a market study performed on behalf of the Deutsche Bundespost where participants were asked to state whether they expect them to be an integral part of telephony or to be offered as an option.

For telephone services offered in the ISDN, new ISDN telephone terminal equipment is under development, which falls into three categories: instruments which largely fulfill basic telephone functions only and two convenience telephones with different degrees of implementation of supplementary services.

As an attractive non-voice 64 kbit/s service, facsimile (group 4) has been envisaged. At present, the access requirements at the ISDN subscriber interface at protocol layers 2 to 7 (OSI reference model) are being specified. Also Text 64 will be offered as a new service, which will be discussed later on.

Further ISDN-specific, that is, 64 kbit/s-oriented services such as a 64 kbit/s interactive videotex service called Bildschirmtext (Btx) are planned for 1990.

In the initial stage, the ISDN will also make it possible to handle a number of the services already existing today. These are, among others, the services offered over the telephone network - for example, the data transmission service with modem and Bildschirmtext. Since the continued use of the existing terminal equipment is assumed for a certain transition period, a terminal adaptor (TA a/b) has been included in the range of products to be developed.

In addition, it is planned to introduce services from dedicated networks or to offer the possibility of connecting the terminal equipment of these services to the ISDN. This applies, above all, to devices conforming to terminal equipment standard X.21 of user classes 4 and 12 according to CCITT Recommendation X.1. An appropriate terminal adaptor (TA X.21) is under development.

Given the above-described range of services, the following ISDN users are to be expected:

1) subscribers with two independent telephone lines,

2) telephone subscribers demanding higher comfort in the form of supplementary services,

3) subscribers who wish to use the telephone and Bildschirmtext services simultaneously,

4) subscribers who want to take part simultaneously in the telephone and data communication services (inc. facsimile),

5) subscribers who wish a combination of various services.

The examples given above show that the ISDN will initally be best-suited for business customers and 'advanced private customers'. The estimated number of ISDN subscribers is shown in figure 3.

The numbering plan of ISDN is recommended in CCITT E.164. Several services require the definition of a so called service identification number which must be added to the unique subscriber number of all services.

3 The technical ISDN concept

As already mentioned, the ISDN concept of the Deutsche Bundespost has directly evolved from the digitization of the telephone network, and all ISDN-specific new development - except for the common channel signalling system - involve the subscriber line area. Figure 4 shows the technical components for the basic and the primary rate multiplex access, which are under development at present and can both be connected to the same local exchange.

Figure 5 shows the structure and nomination of the different interfaces concerning the subscriber access. Whereas the basic access is of the multi-terminal type, to which a great number of 64 kbit/s terminals can be connected, the primary rate

multiplex access has been conceived for connection of medium-large-size PABXs. Both types of customer access offer 64 kbit/s information-carrying capabilities: the basic access 2 times 64 kbit/s, the primary access 30 times 64 kbit/s. The basic access comprises the following equipment: user/network interface, network termination, intermediate regenerator, multiplexer (BAMX), concentrator (BAKT) and terminal adaptors TA (a/b) and TA (X.21) and the network and service interworking equipment required. The primary rate multiplex access encompasses: user/network interface, network termination, intermediate regenerator and line terminal equipment. The following general specifications were required for the above-mentioned products:

- electrical characteristics and physical properties of the interface for the digital subscriber line (reference point u, figure 5).
- testing and maintenance of digital subscriber lines
- activation and deactivation of digital subscriber lines (for basic access only)
- D-channel protocol (levels 2 and 3)
- power feeding concept

Since a detailed description of all components and their technical specifications would go beyond the scope only the essential characteristics of some equipment will be discussed in this paper. The following digital interfaces (in addition to the possibilities of connecting existing analogue equipment) are envisaged on the subscriber's side of digital local exchange:

- interface for basic access (U_{K_o})
- interface for basic access via preceding multiplexer (V_{2MX})
- interface for basic eccess via prededing concentrator (V_{2KT})
- interface for primary rate multiplex access (V_{2PM})

3.1 Network termination

For both the basic access and the primary rate multiplex access the associated network terminations will perform layer 1 functions only. According to the CCITT terminology, they can thus be assigned to the group 1 terminations (NT 1). As a consequence, the user/network interface standards are to be implemented at reference point T. The network termination has to fulfill the following functions:

- line interface
- user/network interface termination
- transmission of activation/deactivation signals (for basic access lines)
- power feeding of the user/network interface and the connected terminals by a local 220 V source up to a limit value of 4 W (applies to network terminations for basic access only)
- provision of test loops (for maintenance purposes) for the bit stream incoming from the local exchange, activated by a layer-1 line signal
- detection of transmission errors of the signals incoming from the line and acknowledgement to the local exchange.

3.2 User/network interface

The electrical and physical standards of the user/network interface conform to CCITT Draft Recommendations I.431 and I.432 for the basic access and primary rate multiplex access, respectively. Two types of customer installation are feasible for the basic access: a point-to-point configuration for connection of single equipment - for example, small PABXs - and a passive bus configuration for connection of a maximum of eight terminals. To be able to use the same network termination for the basic access in both cases, a maximum range of 150 m has been specified for both configurations, which is mainly determined by the electrical characteristics of the requirements of the passive bus configuration.

For the primary rate multiplex access a point-to-point configuration is envisaged only.

3.3 Line interface for the basic access (U_{K_0})

In chapter 2 a description was given of the services offered. These are characterized by the demand for simultaneous communication of different services - among them, also two times telephony. According to CCITT Recommendation I.431, this possibility is offered for the user/network interface by the channel structure B + B + D(16). Consequently, the Deutsche Bundespost has decided in favour of the same channel structure at the line interface. Since both analogue and ISDN basic accesses are possible at the local exchange - and a transition from analogue to ISDN accesses must additionally be feasible if appropriate - the interface parameters and application requirements have been laid down such that 100 % compatibility is ensured in the subscriber line network without the need for switching measures in the cable network. As a result, a cable can also be equipped exclusively with ISDN access lines in the extreme case. In view of these conditions, the following characteristics have been adopted for the interface:

- duplex transmission of B + B + D(16) channels (144 kbit/s)
- one symmetrical pair per ISDN access
- Two-wire transmission with echo cancellation
- 160 kbit/s line bit rate
- line code MMS43 (4B3T)
- 120 kbaud line modulation rate
- maximum range:
  4.2 km with 0.4 mm conductor diamter (92 % coverage)
  8 km with 0.6 mm conductor diamter (nearly 100 % coverage)
- scrambler for B + B + D(16)
- 1 kbaud (service) channel

The work performed so far has shown that the provision of a very large scale integrated interface module constitutes a great technological challenge. This increasingly supports the Deutsche Bundespost's decision to dispense with an interim approach (e.g., with 80 kbit/s) and to concentrate all resources available on the 'full' ISDN solution using the B + B + D(16) structure.

3.4 D-channel protocol

Signalling between ISDN subscriber installtions - including PABXs - and exchanges takes place according to the ISDN D-channel protocol. For the definition of the protocol, the OSI 7-layer reference model is used, where signalling is restricted to layer 1 to 3 (figure 6). In layer 1, whose function is the 'physical transport of bit streams', the D-channel for the basic access operates at 16 kbit/s and that for the primary multiplex access at 64 kbit/s. The D-channel protocol for the ISDN user/network interface is composed of the following two parts:

- layer 2 protocol (data link layer protocol) according to CCITT Recommendation Q.920/I.441 used for transmission control, link-by-link protection, detection and elimination of transmission errors, and
- layer 3 protocol (network layer) according to CCITT Recommendations Q.929/I.450 and Q.930/I.451, for treatment of connections proper.

The basic architecture of the D-channel protocol is shown in figure 7.

Apart from the bit rate, the D-channel protocol for the basic access and primary multiplex access differs in parameters such as the abandonment of the point-to-multipoint configuration in the case of the primary multiplex access.

In the pilot project of the Deutsche Bundespost the D-channel protocol is tested for

- control of 64 kbit/s B-channels and, if required by the service features, also for control without B-channel reference
- transparent network termination (NT 1) (suitable for bus terminal equipment configuration)
- PABXs connected through n x basic access o primary multiplex access (for point-to-point configurations).

The D-channel protocol contains the applicable CCITT specifications and additionally takes account of the services as supplementary services conceived by the Deutsche Bundespost, which are partly shown in figure 2.

4 Integration of non-voice services

Non voice services of today are

- Telex (Tx)
- Teletex (Ttx)
- Facsimile (Fax)
- Videotex interactive (Bildschirmtext, Btx)
- Telebox
- Videotex broadcast (Videotext, Vtx)
- Data services (circuit-switched, cs; packet-switched, ps)

As an example only Ttx will be discussed. The normal Ttx-service is specified by CCITT Recommendations F.200/201 and T.60 - 62. Terminals of this Ttx-service must be connected also to an ISDN-interface So. As mentioned before for this purpose TA X.21 will be provided (figure 8). This figure shows the network transition between Ttx (IDN) and Ttx (ISDN, circuit-switched). From the point of view of the subscriber there should be no difference between the two types of Ttx. Main purpose is to offer the possibility of transition from IDN to ISDN without changing of all terminal equipment. Therefore figure 8 presents only a network transition. The service identification (SI) must be introduced by IDN-ISDN converter (Ttx UC '1001' - SI).

ISDN will offer a new text communication service with a transmission rate of 64 kbit/s (text 64). Within this service Ttx-Fax-mixed mode application will be implemented. The text 64 (64 kbit/s Ttx service) will be the same to Ttx of 2,4 kbit/s as far the service features are concerened. In figure 9 is shown a network and service transition between two different test communication services in different networks. This service text 64 is not yet specified. Therefore figure 9 only describes the basic function of this transition.

A pure service transition within the same network is shown in figure 10. The basic structure of a connection between Ttx (ISDN-CS) and Text 64 (ISDN-CS) requires a speed conversion 2,4/64 kbit/s (TA), but Ttx (ISDN-CS) will be transmitted within the ISDN in a non transparent mode using 64 kbit/s. Text 64 requires a transparent mode. Therefore a special equipment in the ISDN is necessary for this conversion.

Today a connection between Ttx and Dx-L-2400 is possible in the IDN of DBP. Therefore the same service could be offered in ISDN. Figure 11 shows the configuration of the network. It is similar to figure 8, the other way around. Difficulties arise from the different numbering plans of the two networks. Also connections from abroad to subscribers of the Ttx service are in the same situation (Ttx-IDN, Ttx-ISDN).

The best way to avoid some difficulties is the connection of similar networks only:

ISDN (DBP) - ISDN (foreign countries)

IDN (DBP) - foreign data networks.

The conversion always will be made in the country of the receiving terminal equipment.

5 Activities on the way to the ISDN

The activities undertaken by the Deutsche Bundespost to develop an ISDN have been influenced largely by the discussions of the CCITT bodies - especially Study Groups XI and XVIII. In this context, the Montreal meeting of the ISDN rapporteurs of Study Group XVIII has played a key role. There a viable ISDN concept was developed for the first time, which has quickly been taken as a basis for all CCITT standardization activities. The Deutsche Bundespost has followed this concept and resolved at the end of 1982, to introduce an ISDN. In a first standardization phase, the general specification was elaborated for the ISDN configuration to be used in the introductory stage. This specification is fully in line with the descriptions in this paper. A detailed specification was completed in mid-1983. As a result, the prerequisite wascreated to offer the customer series-produced ISDN access lines from the end of 1987 (respectively in 1988).

Since the development of the majority of the ISDN elements requires new technological ground to be broken, the Deutsche Bundespost has taken the decision to test most of the series-produced components in a pilot project under real operating conditions. The first orders for the equipment needed have already been placed with German telecommunications firms. The operational pilot trial will be conducted in 1986 and 1987. To this end, two local switching systems, each for a maximum capacity of 400 ISDN and 50 analogue subscribers, are being set up in Mannheim and Stuttgart. At the same time, tests are performed on Common Channel Signalling System No. 7. The two local switching systems at Mannheim and Stuttgart will be included in the trial of the largely national ISDN User Part of SS No. 7, whose specification was finished early in 1984:

With the described approach, which was developed in close cooperation with the German telecommunications industry, the Deutsche Bundespost feels to have taken all measures required to offer ISDN services from 1988 onward. The corresponding planning proceeds on the assumption that only ISDN-capable switching systems will be implemented under the procurement programme of 1988. In the initial phase, it is envisaged to equip about 5 - 10 % of the local exchanges with ISDN access lines. However, this is only a target value, which may vary depending on actual demand.

The aforementioned approach makes it possible to offer ISDN facilities on an country-wide scale. At the same time, it avoids prohibitive prepayments for the ISDN and cost penalties for the basic telephone service. In addition, optimum flexibility is achieved with respect to both the reaction of the market to the ISDN facilities and the advances in technological development. For no change in the basic concept will be required, even if the digitization of the subscriber line area for the basic telephone service becomes economically justifiable.

6 Further development of the ISDN concept

As has already been mentioned, the configuration for 1987 will indicate the 1st ISDN phase. With it the Deutsche Bundespost will create the basic ISDN concept, characterized by the step towards a universal telecommunication socket on the subscriber's side and the free use of the 64 kbit/s channel. The advent of the ISDN will be accompanied by a rising demand on the terminal market and the offer of new services for business and private applications. Emphasis will be placed on the 64 kbit/s channel. Novel network-supported supplementary services will be realized - for example mail-box services. However, also services with a bandwidth greater than 64 kbit/s will increasingly gain in importance. The services operating in the range of up to 1920 kbit/s ($n \times 384$ kbit/s with $1 \leq n \leq 5$), for which the values of n have still to be specified (first specifications by CCITT Study Group XVIII are already available), can largely be covered by the basic ISDN concept presented in this paper. However, this necessitates the implementation of the facility of throughconnection of several 64 kbit/s channels, while retaining 'digit sequence integrity' in the exchange. Alternatively, a broadband (e.g., 2048 kbit/s) switching network can be provided additionally in the exchanges. As a result, the equipment already envisaged for these services in the basic ISDN concept - particularly the subscriber installations can largely be taken over without modification and with partially extended functions only.

For the services operating in the range above 2048 kbit/s, where visual telephony is the only candidate for an individual duplex service at present, innovations must be introduced at least for the transmission media in the subscriber line area - most likely, optical fibre systems. As a consequence, appropriate broadband

switching networks will be required in the exchanges. Although the studies on these services have not yet been concluded, it has become ever more obvious that their introduction is economically justifiable only if their integration into the overall ISDN concept is ensured and an as wide as possible use of the ISDN components by the broadband services is possible, too. The time frame, then, is given: a 64 kbit/s ISDN will develop from the telephone network and pave the way for a broadband ISDN.

References

/ 1 / CCITT RECOMMENDATIONS OF St.Gr. XI AND XVIII

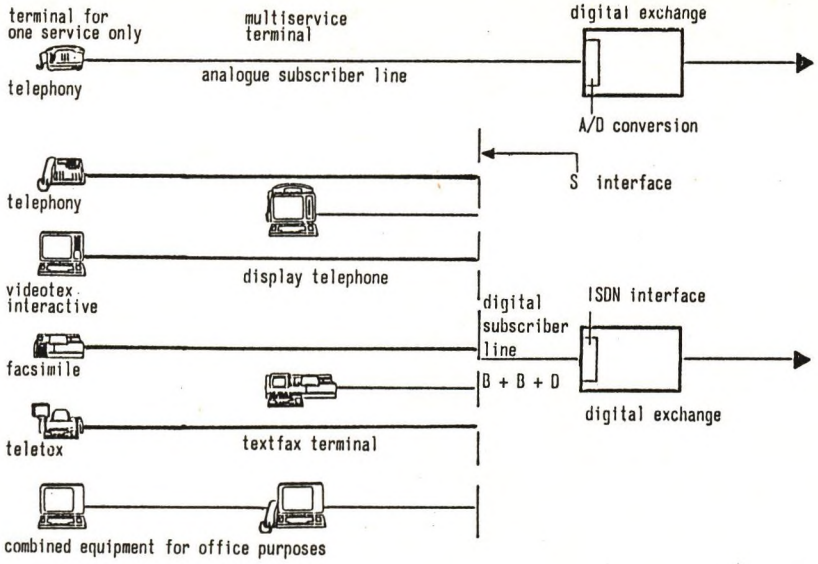
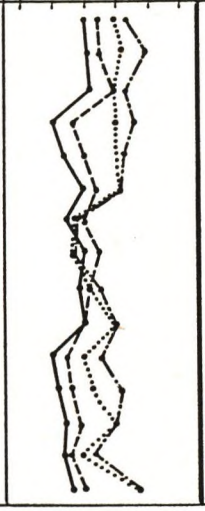


Figure 1: Subscriber access in ISDN (multiservice terminals)

USER'S INTEREST IN MORE COMFORT

Supplementary Service	1 2 3 4 5 6		ISDN
SIMPLIFICATION DIALLING		S	L
COMPLETION OF CALLS		O	Y
CALL WAITING		S	Y
CALL FORWARDING		O	Y
AUTOMATIC ANSWERING		O	L
RECORDING OF INC. CALLS		S	Y
CALL BARRING		O	Y
DO NOT DISTURB		O	L
COLLECT CALLS		O	Y
CHARGE INDICATION		S	Y
CONFERENCE CALL		O	Yn = 5 Ln = 10
THREE PARTY CALL		O	Y
DIRECTORY INQUIRY		O	Y
SERVICE SIGNAL		O	Y
MAINTAINING S-NUMBER		O	



PRIVATE

ENTERPRISES ≤ 9

ENTERPRISES > 10

O = OPTION S = STANDARD Y = YES L = LATER

figure 2

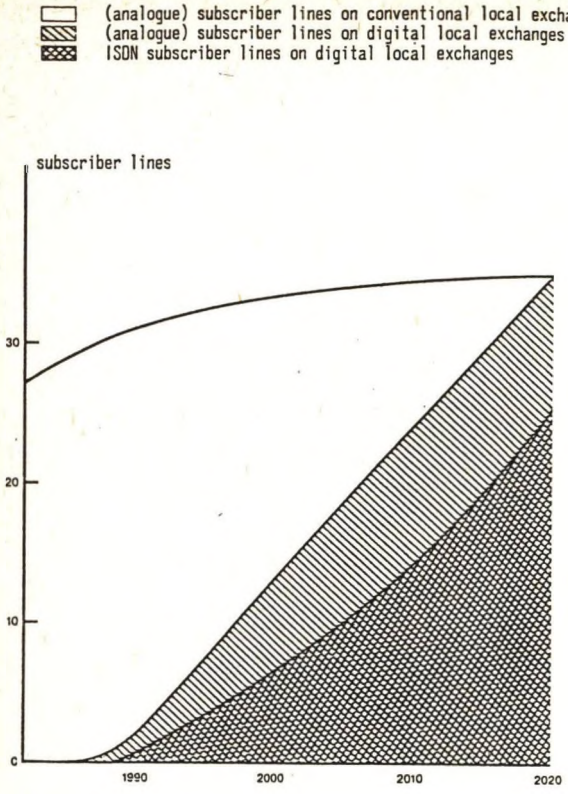


figure 3:
Transition from analogue to digital switching

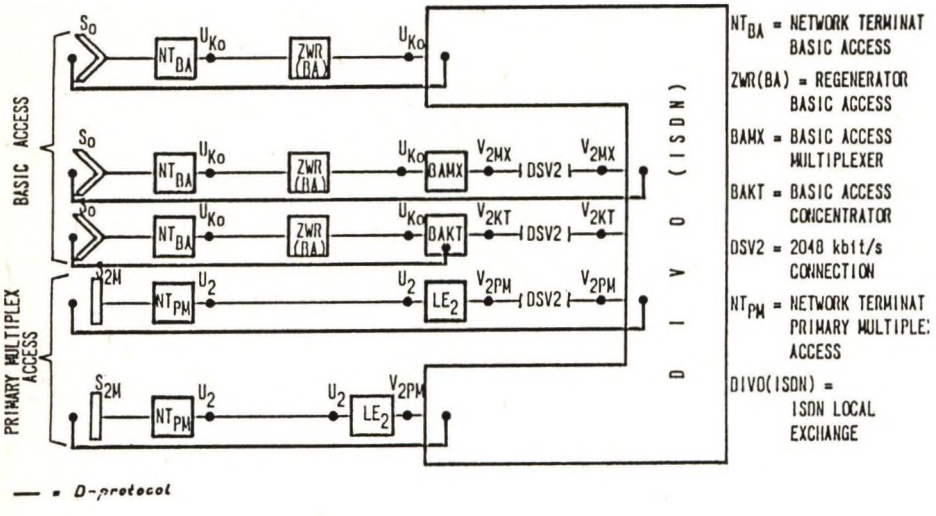
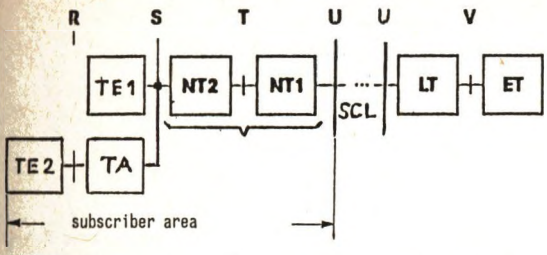


figure 4: Equipments for ISDN access



TE1	terminal equipment (So interface)	NT1	network termination (layer 1)
TE2	terminal equipment (a/b; X.21)	NT2	network termination
TA	terminal adaptation	LT	line terminal
SCL	subscriber line	ET	exchange terminal

figure 5: Structure of the ISDN subscriber access

Ebene Layer Nr.	Aufgaben function	Beispiele examples	
7	Verarbeitung Anwendungssteuerung	Funktionen im Lokalbereich	(1)
	application	data base, time sharing, electronic funds transfer, order entry	
6	Datendarstellung	Vorschriften für Darstellungssteuerung Codierung von graph. Zeichen	
	presentation	data structure formats virtual terminal protocol	
5	Verbindungssteuerung	Austausch von Kennungen und Parametern Unterteilung in "Sitzungen"	
	session	session management	
4	Transportsteuerung	Beginn, Ende Mux-Verbindungen Blockaufteilung Flußkontrolle	
	end-to-end-transport control	network independent interface	
3	Netzsteuerung (Vermittlung)	Paketverm. X.25, Ebene 3 Leit.verm. X.21 / S.70 Netzsteuerung für F-Netze	(2)
	network control		
2	Leitungssteuerung (Übermittlung)	Datenübertragungssteuerungsverfahren (z.B. HDLC)	
	link control		
1	Bit-Übertragung	Strom-/Spannungswerte Geschwindigkeiten Zahl der Leitungen/Stifte CCITT X.- und V.-Serie	
	physical control		

network < independent / dependent

figure 6:
The 7-layer-architecture model

INFORMATION ELEMENT MESSAGE TYPE CALL REFERENCE PROTOCOL DISCRIMINATOR	layer 3
· LAP D - unacknowledged - acknowledged communication · TEI assignment procedure	layer 2
D-channel: 16 kbit/s (basic access) 64 kbit/s (primary multiplex access)	layer 1

figure 7: Architecture of the D-channel

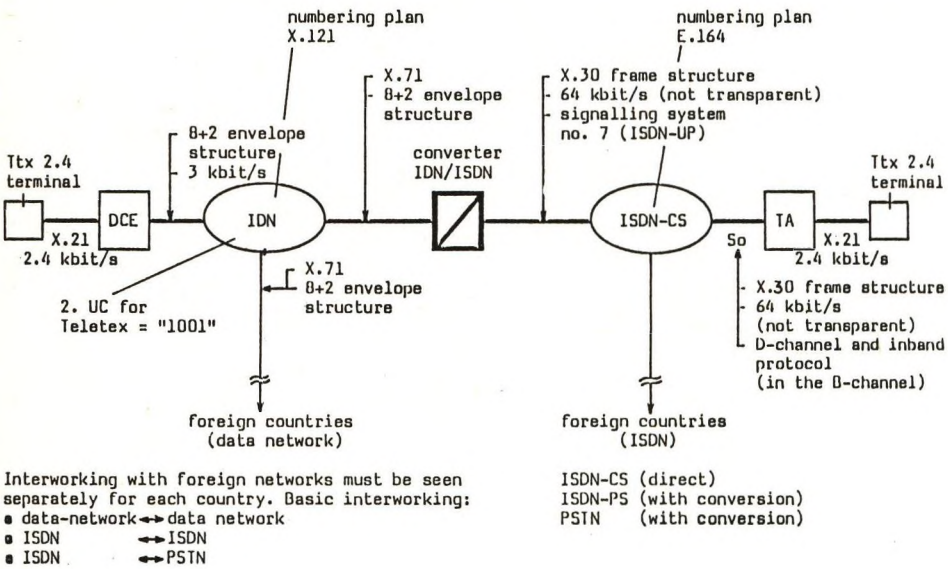


figure 8: Transversion between
Ttx (IDN) and Ttx (ISDN-CS)

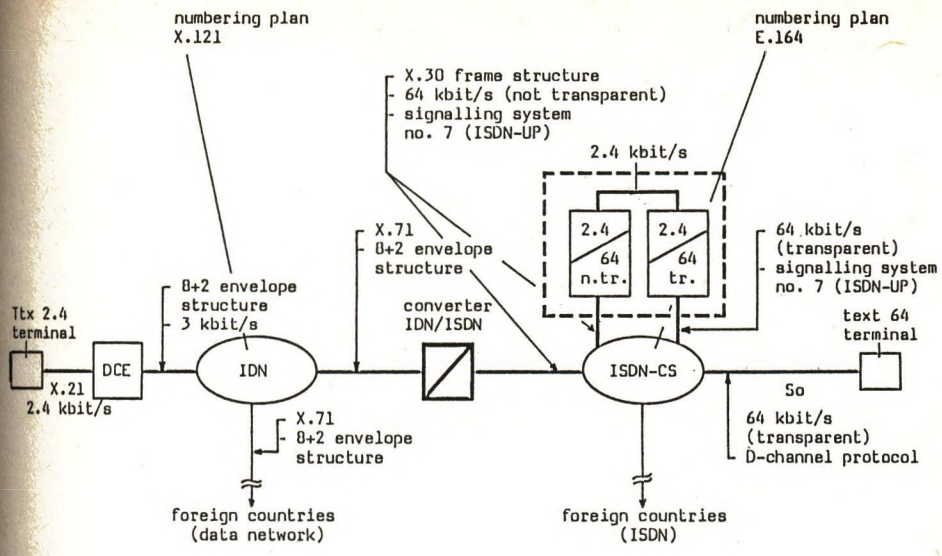


figure 9: Transversion between
Ttx (IDN) and Text (ISDN-CS)

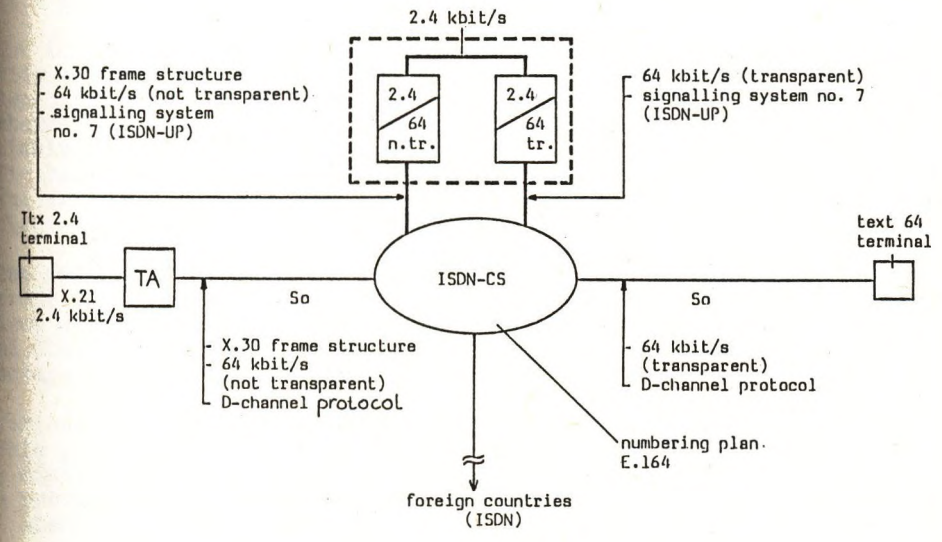


figure 10: Transversion between
Ttx (ISDN-CS) and Text (ISDN-CS)

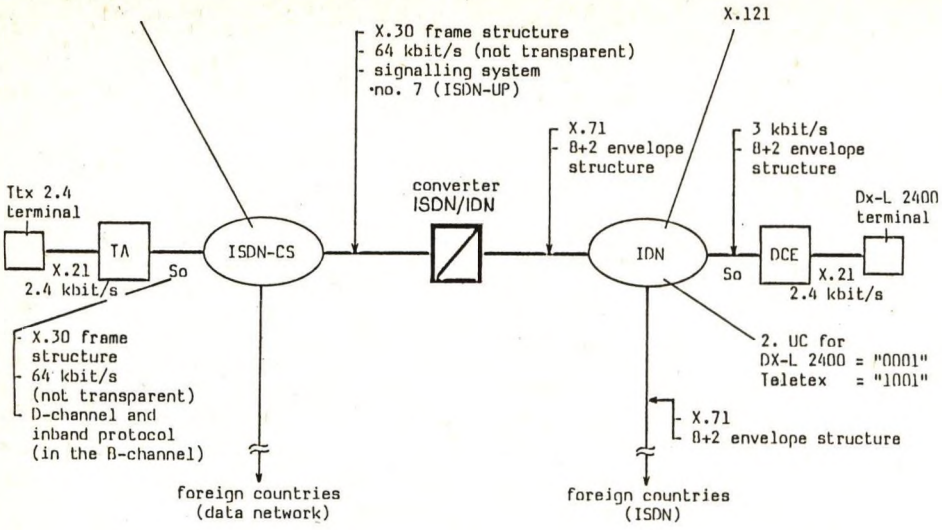


figure 11: Transversion between
Ttx (ISDN-CS) and Dx-L-2400 (IDN)

ARCHITECTURAL PROBLEMS OF ISDN

Gábor Németh

Budapest Technical University
Hungary

A distributed system with high level of modularity in functions, services and capacity is required for the Integrated Services Digital Network. While the channels, functional interfaces and protocols may be primarily based on technical and economic considerations, the services provided, access methods and physical interfaces will have a national flavor. After discussing the architectural problems the pilot project PRS, based on a multimicroprocessor controlled exchange is discussed in some details.

INTRODUCTION

One of the most significant trends of the last decade is the integration of communications and computer techniques [1]. This can be attributed to three main factors: advances in stored program control and digital PCM switching techniques, and of merging of speech- and data transmission.

The integration of communications and computer techniques can provide new and extensive services for large populations. The 8 bit telephony PCM standard provides naturally a 64 kbit/s capacity basic channel capable of transmitting not only phone, but data as well. A widely held opinion is today that the ISDN /Integrated Services Digital Network/ will evolve out of the digital phone network, and the various services for large populations will be based on these channels /and on their suitable multiples and submultiples/.

The framework of ISDN is determined /so far partly only/ by the relevant CCITT recommandations, however, within this framework the various countries will likely develop architectures with strong national flavors according to their present networks, priorities, and economic considerations [2].

Various definitions are given for the ISDN in the literature, and in this paper the essence of ISDN is considered as:
- digital point-to-point /transparent/ connection;
- provision of a multitude of services by using a limited number of standard interfaces.

STRATEGIC CONSIDERATIONS

The move toward the ISDN requires a long transition period even in the developed countries. Naturally the present networks and equipments installed create widely different starting positions in different countries. A good transition strategy in the most developed

countries may be the development of ISDN based on the Integrated Digital Phone Network.

In Hungary there is no basic digital phone network today. Because of this the digital point-to-point transparent connection can be realized only with severe restrictions. Although the videotex, teletex and facsimile services can be introduced into the analog network, its low transmission speed makes these services painfully slow.

In case of narrow channel services the creation of ISDN "islands" seems to be the advisable strategy [3]. In this case digital switches and PBX-es installed on the lowest network layer can provide new services /including connection to special networks/ for those subscribers connected to these islands. Naturally this approach severely restricts the digital point-to-point transparent connection requirement if no overlay networks are employed.

Our own digital exchanges and PBX-es for the lowest network layer are available already or in their final testing phase, and the previous digitalization of the trunks supports this digital island strategy.

An important question is where to make such islands first. In Hungary the new residential areas of cities and to some extent the rural areas pose the most pressing needs. Because in these areas virtually no phone or data services exist today, the new services can be installed economically, since there is no need to create an overlay network.

Of course, not only the network side, but the subscriber side should be carefully investigated. While in the most developed countries the majority of the users of the new narrow channel services comes from the business users, in Hungary the extension of services within an institution locality and the agricultural companies /the latter requiring a sizable share of mobile equipment/ may be the driving force behind the introduction of ISDN. So the need for cooperation with personal computer based local networks will be an especially important consideration.

The digital island strategy proposed tends to localize some new services offered. Besides this, selecting the new services is limited by economic considerations, too. If an offered new service is underutilized, the cost of the other services will increase. However, in the residential and rural areas some services may be important from social aspects, regardless of economy, for example access to the patient scheduling system of hospitals and district medical centers, or monitoring of aged people by remote measurement of the electric consumption.

INTEGRATION LEVELS

Considering the narrow channel services only, it is obvious that the division of functions among the exchanges /PBX, local, teletex, etc./ will vary by the advancement of integration. This fact necessitates the development of flexible hardware and software structure, with carefully selected and strictly function /and not technology/ based modules.

An important problem is the division of the services among the modules of the system. In case of low or medium level of integration

the use of circuit and/or packet switching is proposed for the transmission of messages, too. Besides, for the circuit switched and packet switched procedure the use of common subscriber lines and local exchange /or PBX/, with branching off at its output may be proposed /Fig. 1/.

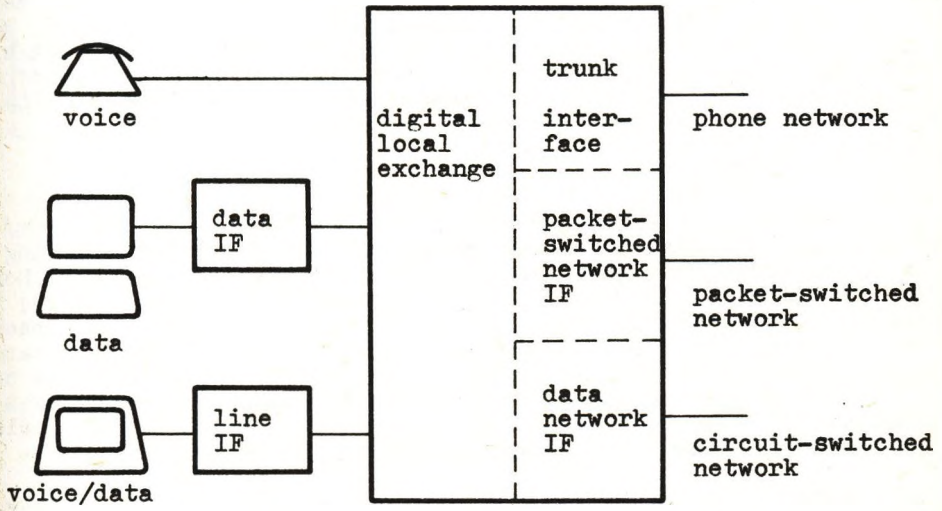


Figure 1
Multifunction local exchange

It should be noted that other service assignment structures may also be developed for the low or medium levels of integration. However, the structure proposed beside supporting the modular architecture of the local exchange, suits well to the frame organized transport protocol recommendation of CCITT for the ISDN. Thus the majority of the hardware and software control modules of the local exchange will be common for both procedures.

Another important problem of the service modularity is related to the control structure. Compared to the conventional phone service the signalling and control connections between the subscriber and the network should considerably be improved /consider, for example, the case a subscriber wants to make a phone and a data connection from his/her terminal at the same time with different partners/. Moreover, in case of several new services some control and signalling information should be transmitted between the users. These informations are transmitted, received and interpreted by the user terminals; the network should be transparent for them. Thus the handling of the control and signalling information - according to their types - should be divided between the subscriber terminal and the exchange/s/ of the network.

Let us consider, for example, the cooperation of ISDN and local area networks /LANs/. In this case a connection should be set up between a LAN terminal and an ISDN terminal, requiring a transformation between the LAN and the ISDN protocols /gateway function/. The problem

in this example is that the LAN protocols are determined by the topology and technology used /access methods and rights, various acknowledgement procedures, etc./, differing markedly from the technology and topology independent philosophy of the ISDN.

In developing a control structure for the ISDN the principle of the virtual resources can be used [4]. In our case a network providing various services may be regarded as a set of several virtual networks, each of them providing a single service. These virtual networks should be mapped onto the parts of the physical network/s/. In every phase of the introduction of the ISDN the actual form of mapping is determined by the available physical network/s/ and the required services /virtual networks/ in the current phase.

THE PILOT PROJECT PRS

The main objective of the pilot project PRS /PCM Remote System/ is to provide extensive phone and data services for widely varying number of subscribers without overloading the existing network. Moreover, in spite of increasing needs to connect new subscribers, the lack of floor space in the existing exchange buildings /or space for new buildings/, as well as the transmission loss problem in case of remote subscribers the extension of the existing exchanges is usually neither economical, nor preferable. Thus a pilot project was originated by the cooperation of the Research Institute for Telecommunications and the Budapest Technical University [5].

The control structure of an electronic digital exchange differs markedly from those of other fields. The exchange should be highly modular with respect to capacity /its cost should be directly proportional to the number of subscribers connected to it, and to the number of services provided/, and in case of failures the system should not break down, but degrade gracefully /continuing its operation perhaps with some limitations/. These two characteristics raise the possibility of a control system with multiprocessor structure. Besides, in an exchange the accomplishment of certain services is not strictly required /e.g. the dial tone may be given to the subscriber immediately after hook-off, but a 100 ms delay is also perfectly acceptable/. This third characteristic permits task queuing [6].

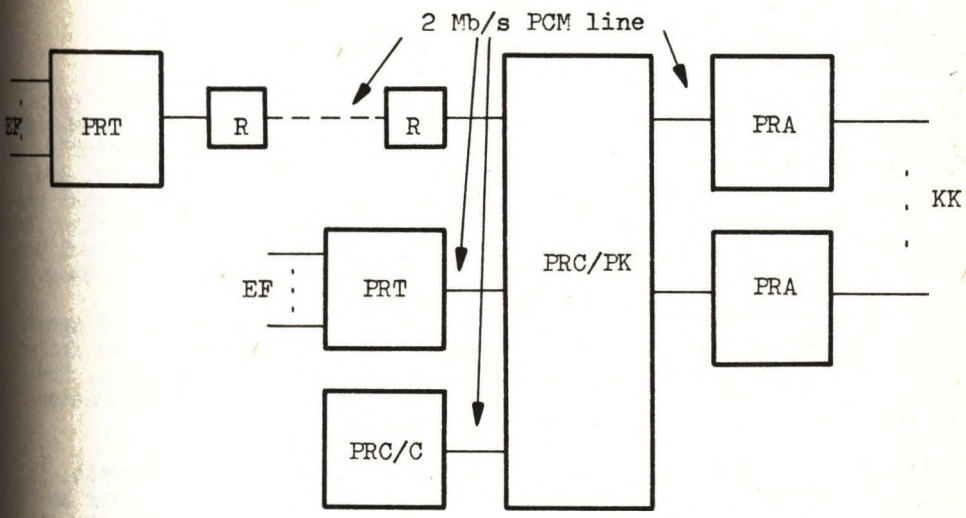
Based on the discussion above the main features of the PRS are the following: digital remote exchange concentrator with internal traffic, distribution of call processing functions, cooperation with the local exchange through common channel signalling, extensive maintenance software, and automatic reconfigurability in case of a failure.

The distributed architecture, multiprocessor-controlled digital exchange PRS is shown in Fig. 2.

The terminal unit PRT preprocesses the subscriber lines. It has the following basic varieties:
- analog subscriber line /each analog subscriber line has its own single-channel codec in the PRT/;
- digital phone subscriber line /the codec is in the phone/;
- data line;
- combined /multiservice/ digital subscriber line with 144 kbit/s or some multiple of it.

The common functional units are duplicated to increase the availa-

bility of the PRT. Every functional unit is separately duplicated, thus the PRT has no critical parts, and it is flexibly reconfigurable.



EF - Subscriber line
PRT - Remote Terminal /up to 500 subscriber lines/
R - Repeater
PRC/PK - PCM /TST/ switching unit
PRC/C - Central controller
PRA - Network Adapter
KK - Trunks

Figure 2
Structure of the PRS /PCM Remote System/ digital exchange

The PRT has static task allocation control architecture. In this case a processor provides a certain function for the whole unit, minimizing the interactions among the tasks of the processors, thus the software is relatively simple. The rapid increase of the capabilities of the microprocessors, and decrease of their prices make also preferable the static task allocation nowadays.

Up to two PRTs, with the equivalent of up to 500 conventional phone subscriber lines each, may be connected to a segment of the central controller PRC. A PRC segment consists of a PRC/C and a PRC/PK. The PRTs may be located either in the same location as the PRC /this is the typical arrangement in cities and PBX uses/, or remotely located and connected to the PRC through repeaters /this is typical of rural applications/.

The PRC may contain up to four segments, communicating with the units PRT and PRA through 2 Mbit/s lines /duplicated toward the PRTs/. The dinamically reconfigurable redundant switching stage segment PRC/PK performs time-space-time switching on these 2 Mbit/s lines.

The control of the switching stage, and the majority of the call processing functions are performed by the control unit segments PRC/Cs. The major aspects in selecting its suitable organization are the following:
- partitioning the control algorithm into tasks;
- allocation of the system resources;
- characteristics of the processor modules;
- communication methods among the processor modules; and
- reliability requirements.

Naturally these aspects are interrelated and quite often pose widely divergent requirements. Let us look at, for example, the problem of the system resources /by the way, they may be both hardware, and software modules/. On the one hand, more system resources make possible greater flexibility and system availability. On the other hand, more system resources mean more complicated allocation control procedures to be able to handle the increased competition for them.

There is a theoretically interesting problem, the so called sphere of influence. Diagnosing an error symptom we can remove the failed module from the system /in case of a suitable organization/. However, usually several modules interact, thus the reconfiguration must be performed for a proper environment.

This problem has architectural consequences, too. For example, in case of hardware modules, the segmentation should preferably be based upon functional isolation, space exploitation, capacity and service modularity from construction aspects, while the diagnostics and repair processes require a different segmentation.

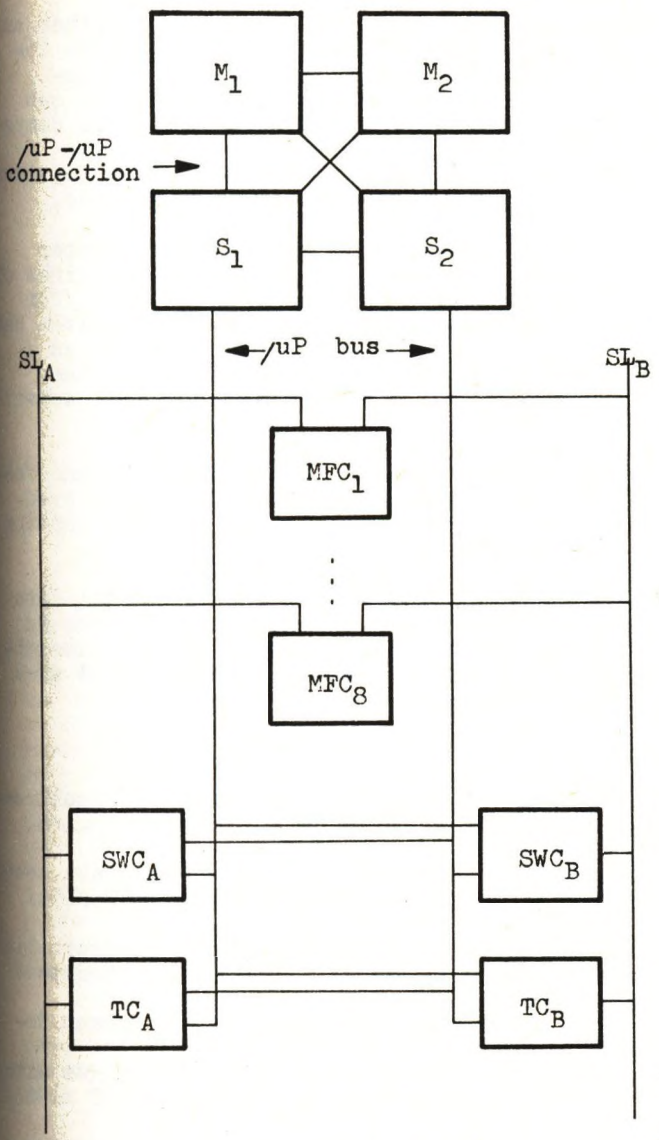
Based on the considerations above, the joint use of the static and dynamic task allocations was selected. The simplified structure of a control unit segment PRC/C is shown in Fig. 3.

The global functions of control are performed by the active master processor, while the other master processor performs self-testing and mutual testing. Likewise, the peripheries are controlled by the active slave processor.

Microprocessor-controlled digital multifrequency coded /MFC/ receivers and transmitters perform the signalling with the other exchanges. They form a resource pool, and the number of active MFCs depends upon the current traffic conditions. Note the redundant structure of the control unit.

The unit PRA connects the system to the higher level exchange of the network through digital lines. The man-machine communication /including the operational and global maintenance functions/ is provided also through this unit. /The detailed maintenance procedures are performed by the microprocessors in the units PRTs and PRCs. They are invoked either automatically, or by operator request./

The maintenance philosophy is out of the scope of this paper, so let us look only an important part of it. The reconfiguration of the system in case of a failure, and the field repair process requires some kind of diagnostics in the exchange. It is desirable to have a diagnostic software operating in a virtual environment, and analyzing the functions and not their implementation details. This principle deserves some explanation.



M - Master/uP
S - Slave/uP
SL - Service line /2 Mbit/s/
MFC - MFC receiver - transmitter /uP/
SWC - Switch controller
TC - Terminal controller

Figure 3
Structure of a segment of PRC/C

Theoretically the depth of diagnostics can be arbitrarily chosen, although economic considerations clearly limit it. However, the key factor here is the fast and continuous progress of the integrated circuit technology, making a lot of times economically attractive the hardware modifications during the usually long lifetime of the exchanges. Having a modular architecture with well defined interfaces however, only the internal structure of the modules - preferably the

WINDING ROADS TO AN ISDN

Sándor K. Mazgon

System Technics Department
Research Institute of the Hungarian
Post Office /PKI/
Budapest
Hungary

The many efforts devoted on the ideas assembled under the slogan ISDN implies a concensus: ISDN is feasible and desirable,but no concensus exists on in what way and how fast to proceed on developing an ISDN. There are many questions open making the ISDN itself questionable.For example the many new telecommunication services provided on existing networks tend to concentrate on dedicated networks. The funds invested into such dedicated networks need to be recovered before the integration of any is initiated into a digitalized telephone IDN.

Information Society, Information Based Economy, Information Age and the many synonyms widely used are all manifesting the importance of information in our everyday life, the increasing dependence on it, the demand for real time access to the valid information. In that respect the role of communication to access the information services, is indisputable. Telecommunication networks are the vehicles that make the information available and accessible where and when it is needed. Telecommunication networks were created originally to support a single service they were optimized for.Now there are many such networks in existence,some parallel to each other and redundant in many sense.Table 1 gives a summary of communication, classifying it in a manner natural for the designer of a network.What networks are created for, however, is not communication itself,rather a service provided to support it. Table 2 lists telecommunication services offered over public telecommunication networks.The trivial conclusion of such tables would be an integration scenario.

The idea to integrate the many networks, applications, user services and user network access interfaces was not new,nevertheless, the early methods and means were divergent, interfaces and protocols were incompatible and interest was not on the side of the integrators until the ISDN was defined.The environment is now changing: semiconductor technology, stored program control, digital handling of information, general acceptance of standardization, the developments of open systems interconnection architecture and protocol constitute a framework driven by the rapid evolution in new materials, fibre optics and satellite technique to name a few.

The convergence of communication and computer technique and the use of both in many if not in all the sectors of the economy increase the demand for data communication higher and higher. Data communication traffic is growing rapidly in private and in public networks as well and the general need to make data communication available everywhere justifies the ambition to integrate data

transmission with telephony on a wide sense and with video services in the further future. But there are the public digital data networks providing services not supported by the analogue telephone networks. Almost all European countries have plans to develop an established or an experimental public data network providing packet switching and/or circuit switching services.In Hungary there is a circuit switched data transmission service provided on a circuit swiched public data network.The phases of the evolution of the data communication services in Hungary are summarized in the Tables 3 to 6 and are planned to be provided until a full-scale coverage of the country is achieved by an ISDN.

The way,over which an Administration intends to approach an ISDN is the individual responsibility of the Administration. The choice involves the clear definition of the ISDN and the establishment of the means and end for that way. The aspects of that choice are summarized on Table 7.

Making the plan to approach an ISDN, the Administration have to solve the many problems of digitalisation and establishment of the many phases until a digital-telephone-network, an integrated-digital-telephone-network, a narrow-band-integrated-services-digital-network and last but not least a wide-band-ISDN is achieved. In Hungary, as for many developing countries, the situation is complicated by the limited capability of the national industry, by the financial problems constraining imports, by organizational problems, and by the manufacturer's interest diverging from the provider's interest and both being far from the user's need.

The prerequisites and conditions for an ISDN are many and lead to different interpretations of the concept of the ISDN itself. Is it a concept of development? Is it a single network or are there many types and forms of ISDNs with some common features? Is an ISDN directly achievable or must be developed for a long transitional period where many evolutionary phases are to be planned exactly and carefully? The factors to be taken into account when approaching an ISDN are too numerous to enumerate all,some are listed on Table 8. In Hungary one of the hardest problem seems to be the insufficiency of the telecommunications infrastructure /cables, ducts, manholes, buildings, repeater stations,distribution cables,etc./. Perhaps it would be easier to plan a very future-safe, economic, modern and sophisticated network not having any telecommunications at all. But there are some old and obsolete systems in use.And the telephone density is far behind the real need of the society.And traditionally the resources available for telecommunication development are far below the European average level /the fund in relation to the per capita national product or in relation to the per capita investment/. The shortages in equipment and/or financial resources may have the very effect of delaying the development and leading to uncoordinated measures taken at random disarranging time schedules and upseting the plans.

Another difficulty is created when partial digitalisation is performed and there is no balance between transmission technique and switching system used for it. Without digital switches there is no way to make a technologically integrated IDN type network. With any switches, not designed for handling of digital signals digitally, it is very hard to plan a full 64 kbit/s connectivity and an integration of voice and data traffic over the customer-to-customer digital connection.Where the multiplex structure of transmission and of the virtual highways of the switching fields need conversions for

data transmission services, it will need many more conversions for the introduction of any other non-voice service.

In many countries several dedicated networks are considered to be integrated in a single ISDN. These networks must have a common transmission plan,common synchronisation structure and many other common features at the time of planned integration. In Hungary there is a public telephone network used for telephony and for data transmission via modems; there is a public integrated data network used for telegraphy, telex and circuit switched data transmission services. These both cover the country but penetration of the services is very low, while the number of telephone sets connected to the network is over 1,3 million, the number of data stations of all kind is not more than 2000. Private networks are used for telephony,telex and data transmission as well.But structure and transmission plans are especially created for the special needs of that special networks Any plan of integration must ensure, that all of these networks will be capable of being incorporated into an ISDN. The difficulty to plan this is manyfold from the diverging interest up to incoherent requirements.

A reasonable way of digitalisation may be the one in which some isles of ISDN pilot projects become commertially operated and the nationwide coverage is planned to evolve from that isles.The next step is the most difficult to choose. Transit switching would be the natural choice. That means a digitalisation of the four-wire switches is made where in analogue swiching a four-wire switch is already used. So the transmission plan, the synchronisation plan, and other plans can smoothly go through the transitional period and the network inherent services and utilities are available for all the customers of the region concerned.

Almost all of the organisations planning an ISDN stress the fact that the transition period from analogue to digital, and from digital to IDN type, and then to ISDN type network will be quite long During that periods analogue and digital, dedicated and integrated, IDN-like and ISDN-like networks and network-parts will coexist in many countries. Independently from the strategy used in digitalisation the international traffic must be evolving in an undisturbed manner with constantly improving quality.

The ISDN concept implies that there is a high degree of independence between the transport mechanism and the services which it supports.New services,whose future utilization and traffic characteristics cannot be determined precisely in advance, could be integrated into the existing network without a major reconfiguration of its architecture.ISDN could carry services in the future which have not even been envisaged at the present time.

Since the ISDN is formed from a circuit switched and a signalling network, support for a circuit switched data transmission service is achieved by adoption mainly at the periphery of the network by terminal adapters /see X.30 Recommendation of the CCITT/. Full support of a packet switched data transmission service requires additional switching capabilities over that inherent in the telephony IDNs.Simple adaption at the periphery of the network is only consider to be practicable if the ISDN is used as a transparent access information path to a separate packet switching function or a dedicated packet switching function network /see X.31 Rec.of CCITT/. Later introduction of packet switching functions in the ISDN should be compatible with interfaces of B and D channels in the ISDN.

Longer term development could see the rationalisation of the functions of the packet and signalling layers by the introduction of common hardware configured to support a variety of physical, data link and network layer protocols as well as switching functions.

The high cost of access to a packet switched data transmission service through an analogue network involves the cost of the modem-pair and the analogue access circuit.Where the distances becomes short enough the modems can be of baseband type or of low cost variety. It is generally assumed, that the introduction of a leased circuit digital data service help to reduce the access costs further especially at the higher data signalling rates required for packet mode of operation in the terminals, to support many simultaneous calls over a single physical access circuit.As ISDN facilities become available these may also reduce acces costs to a packet switched dedicated data network /minimal integration scenario/ but much will depend on the tariffs charged for such access.In the IDN,where the common access circuit is used for many services and this access circuit enable customer to have access not only to his local telephone exchange,which might employ digital switching, but also to a packet assembly-disassembly device or packet concentrator co-sited at the local telephone exchange may involve the greatest cost reduction. Integration of a packet switched public data network into an ISDN is complicated because there is little commonality other than the transmission and the handling of signalling in the signalling network itself. Some Administrations plan to use new generation of digital swiching exchanges both in the packet switched data network and in their ISDN pilot projects, as in the latter special switching modules have packet switching functions or some other functions supporting the packet access at the ISDN interfaces. Countries,like Hungary,where packet switching is only planned for the near future, the ISDN appoach to data transmission must involve a leased circuit data transmission service and an access to the circuit switched public data network and extension capability for the packet switched services of the future.In that respect the integration of data services may be smooth technically but not organizationally, as many aspects are to be arranged well in advance. See Table 9.

It is very hard to conclude anything from the large amount of ideas given in the literature on ISDN which can be stated as surely relevant to a country like Hungary.It seems to be realistic, that:

- there are many ways leading to an ISDN but many more lead elsewhere;
- the circumstances can force a winding road rather than the direct one;
- there are many winding roads and not all are leading to ISDNs
- if there is a misleading road, then there is a force also, which by Murphy's low, compels us to drive on it. /Fig.1/

References

A.A.Dogterom: Is the ISDN concept realistic?
ISSLS-82 Conf.proc. pp 14-18

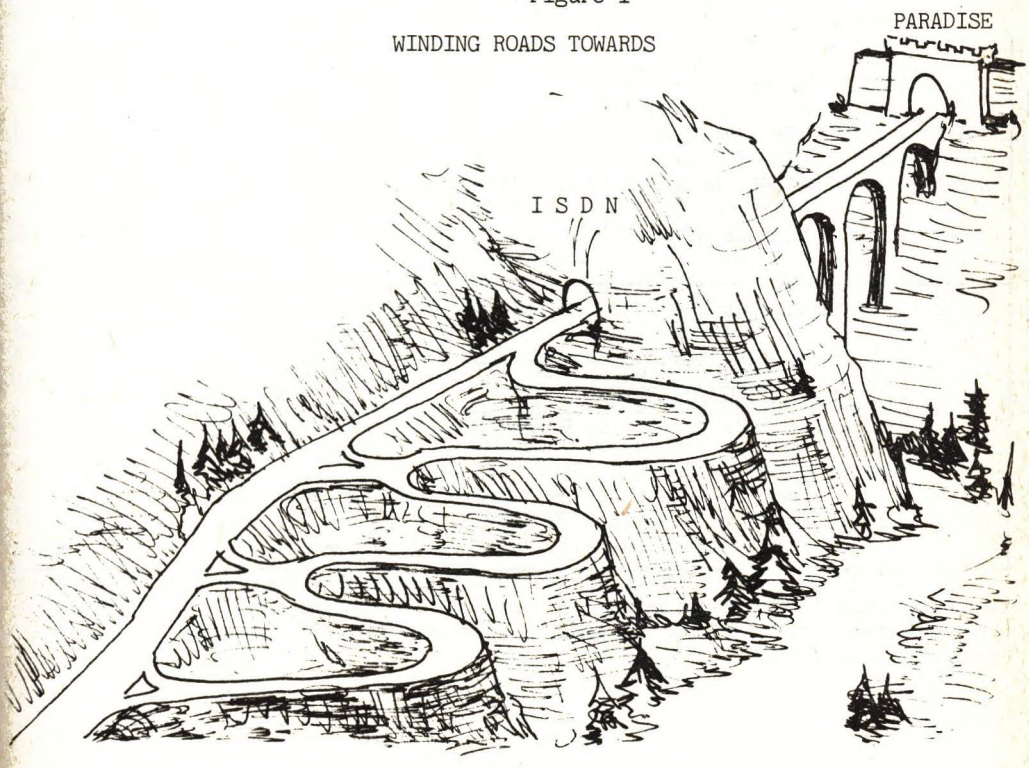
CCITT Red Book Volume III Fascile III.5 Integrated Services Digital Networks /ISDN/. Series I Recommendations.
International Telecommunication Union.Geneva.1985

L.Ackzell: Evolution of ISDN in Europe
ISSLS-82 Conf.proc. pp 104-108

M.Decina: International standards activites shaping ISDN
ICC-84 Links for the future. IEEE. pp 563

D.J.Kostas: ISDN overview. Telephone and Engineer Management Dec.1.1985p.153-9
Dr.S.Kent: Integrated services digital networks
Communication International 10.6.July.1983 pp 40-43
G.P.Oliver: The integrated services digital network
Conference proceedings pp 8 - 13
E.B.Holloway: The evolution of PSS within the ISDN
Conference proceedings pp 284 - 288
P.T.F.Kelly: Dedicated networks in an ISDN
Conference proceedings paper 3.14.3 pp 1-4
J.O.Wedlake: Customer services for the next decade
Conference proceedings paper 2.6.1 pp 1-8
A.Heiber: the road to ISDN
Telephone Engineer and Management.March 1.1985. pp 122 - 124
S.W.Johnston: Three-phase evolution to integrate networks
Telephone Engineer and Management.March 1.1985. pp 124 - 126
L.Mantelman: ATT's ISDN begins earth descent; BOCs and the ISDN; Beyond ISDN
Data Communications 14.5.May.1985 pp 45 - 64
L.Gimpelson: Network transition strategies - analogue to digital IDN to ISDN
Communication International 8.6.June.1981 pp 43 - 46
W.Staudinger: A 64 kbit/s model network as a first ISDN approach
ISS-84 Florence paper 22.B.5 pp 1 - 5
M.Decina: Evolving network technologies for the ISDN
London. Workshop on ISDN. Jan.1985 pp 1 - 5
Louis Pouzin: Will the ISDN promote standards or black boxes ?
London. Workshop on ISDN. Jan.1985 pp 1 - 4.

Figure 1



COMMUNICATION TABLE 1

1.	DIRECT COMMUNICATION	
2.	INDIRECT COMMUNICATION	
	2.1 MASS COMMUNICATION	
	DISTRIBUTION OF PRINTED MATERIALS	
	BROADCASTING OF PROGRAMS: RADIO, TV, ...	
	2.2 TRADITIONAL POSTAL DISRTIBUTION OF	
	CARDS, LETTERS, MAIL, PARCELS, PACKETS, ...	
	PRINTED MATTER, BOOKS,....	
	2.3 TELECOMMUNICATION	
	2.3.1 VOICE COMMUNICATION: SPEECH,TONES,MUSIC,STEREO,...	
	2.3.2 VIDEO COMMUNICATION:	
	STILL PICTURES: FACSIMILE, CODED FACSIMILE,...	
	MOVING IMAGES: SLOW SCAN TV, HIGH RESOLUTION TV,...	
	2.3.3 TEXT AND DATA COMMUNICATION:	
	TELEPROCESSING, CODED TEXT COMMUNICATION,GRAPHIC COMMUNICATION, DATA TRANSMISSION FOR TELEMECHANICS, REMOTE SENSING,REMOTE CONTROL,REMOTE ALARM,...	

NOTE: The above classification is not exhaustive and is intended only to demonstrate that neither the communication, nor the mode of it nor the representation of information constitutes a service or facility.

TELECOMMUNICATION SERVICES TABLE 2

1.	CCITT regulated services	
1.1	TELEPHONE SERVICE	/TF/
1.2	TELEX SERVICE	/TLX/ or /TX/
1.3	GENTEX SERVICE	/GX/
1.4	TELETEX SERVICE	/TTX/
1.5	TELEFAX SERVICE	/TFX/
1.6	BUREAUFAX SERVICE	/BFX/
1.7	VIDEOTEX SERVICE	/VTX/
1.8	CIRCUIT SWITCHED DATA TRANSMISSION SERVICE	/CSDTS/
1.9	PACKET SWITCHED DATA TRANSMISSION SERVICE	/PSDTS/
1.10	LEASED CIRCUIT DATA TRANSMISSION SERVICE	/LCDTS/
1.11	MESSAGE HANDLING SERVICE	/MHS/
	etc.	
2.	Facilities and features not regulated by the CCITT but sometimes referred as services	
2.1	ACCESS TO INTERNATIONAL DATABASES	
2.2	ELECTRONIC LETTERPOST	/ELP/
2.3	ELECTRONIC MAIL WITH DELIVERY /for closed user groups or as a special service for the public/	/EMD/
2.4	COMPUTER BASED MESSAGE SERVICES /may be a part of the message handling services/	/CBMS/
2.5	ELECTRONIC MAILBOX TO RETRIEVE /including text-mail, voice-mail, data-mail,etc./	/EMB/

NOTE: The above classification is not exhaustive and is intended only to demonstrate that the definition of services have to be network independent and consistent with the open systems interconnection /OSI/ when planned to be included in an ISDN.

TABLE 3

DATA COMMUNICATIONS DEVELOPMENT PHASE Ø
BEFORE THE ERA OF THE INTEGRATED DATA NETWORK /IDA/

TELEGRAPH SERVICES:	AVAILABLE SINCE:	1847
LEASED TELEGRAPH CIRCUITS ARE USED FOR DATA TRANSMISSION	SINCE:	1968
TELEPHONE SERVICE:	AVAILABLE SINCE:	1881
PUBLIC SWITCHED TELEPHONE NETWORK IS USED FOR DATA TRANSMISSION	SINCE:	1967
LEASED TELEPHONE CIRCUITS ARE USED FOR DATA TRANSMISSION	SINCE:	1968
TELEX SERVICE:	AVAILABLE SINCE:	1942
TELEX USED FOR DATA TRANSMISSION	SINCE:	1968
DATEX SERVICE:	AVAILABLE SINCE:	1971
200 BAUD SERVICE DESIGNED AND IMPLEMENTED FOR CIRCUIT SWITCHED DATA TRANSMISSION SERVICE ONLY		

NOTE: One of the first use of data transmission was the collection of the results of lottery evaluations. One of the first terminal designed for data transmission has been an automaton "TELEXDAT" providing the height data of the water level of the river Danube retrievable through the telex network.

TABLE 4

DATA COMMUNICATIONS DEVELOPMENT PHASE 1
AT THE INTRODUCTION OF THE INTEGRATED DATA NETWORK /IDA/

INSTALLATION OF THE FIRST ELECTRONIC SPC SWITCHING EXCHANGE IN BUDAPEST WAS THE FIRST STEP OF IMPLEMENTATION OF THE IDA:	1981
IDA INTEGRATES ALL THE SERVICES SUPPORTED BY THE TELEGRAM, THE TELEX, AND THE DATEX NETWORK BEFORE	1981
IDA OFFERS A CIRCUIT SWITCHED DATA TRANSMISSION SERVICE FOR ASYNCHONOUS TERMINALS OF USER CLASSES 1 AND 2 SINCE	1981
IDA OFFERS A CIRCUIT SWITCHED DATA TRANSMISSION SERVICE FOR SYNCHONOUS TERMINALS OF USER CLASSES 4 AND 5 SINCE.	1984
IDA OFFERS ACCESS TO INTERNATIONAL DATABASES IN THE USA SINCE	1982
AND IN 6 EUROPEAN COUNTRIES SINCE	1984
IDA CONSTITUTES AN OPEN NETWORK FOR FURTHER EXTENSION AND EXPANSION FOR PACKET SWITCHED AND LEASED CIRCUIT DATA TRANSMISSION SERVICES	
IDA WILL BE THE BASE OF INTRODUCTION OF TELEMATIC SERVICES	
IDA IS EVOLVING AS A SINGLE EXCHANGE NETWORK WHILE DATA TRANSMISSION CONTINUES TO DEVELOP ON THE TELEPHONE AND TELEX AND OTHER NETWORKS /TELETEXT APPEARED AS A DATA DISSEMINATION SERVICE OVER TV PROGRAM BROADCAST/	

TABLE 5

DATA COMMUNICATIONS DEVELOPMENT PHASE 2
WHEN EXPANDING THE IDA IN A MULTIEXCHANGE FORM OF NETWORK

INSTALLATION OF OTHER ELECTRONIC SPC DATA SWITCHING EXCHANGES FOR THE INTEGRATED DATA NETWORK /IDA/

SYNCHRONIZATION BETWEEN AND WITHIN THE IDA AND OTHER DIGITAL NETWORKS

EXTENDING THE DATA TRANSMISSION SERVICES IN 1985-87 TO SUPPORT TELEMATIC SERVICES BY

-- A PUBLIC TELETEX SERVICE OVER THE IDA
-- A PUBLIC VIDEOTEX SERVICE OVER THE PSTN
-- A PUBLIC TELEFAX SERVICE OVER TH PSTN
-- A BUREAUFAX SERVICE NAMED: POSTAFAX

IDA WILL OFFER PACKET SWITCHED DATA TRANSMISSION SERVICE FOR PACKET MODE TERMINALS OF USER CLASSES OF SERVICE 8 AND 9 IN 1985 - 86

IDA WILL EXTEND THE ACCESS OF INTERNATIONAL DATABASES TO OTHER EUROPEAN AND NON-EUROPEAN COUNTRIES HAVING DATA NETWORKS

IDA, AS THE HUNGARIAN PUBLIC DATA NETWORK WILL BE INTERCONNECTED TO PUBLIC DATA NETWORKS OF OTHER COUNTRIES TO PROVIDE INTERNATIONAL CIRCUIT SWITCHED AND PACKET SWITCHED DATA TRANSMISSION SERVICES AND OPTIONAL USER FACILITIES

IDA WILL BE USED TO OFFER LEASED CIRCUIT DATA TRANSMISSION SERVICE FOR NATIONAL AND INTERNATIONAL USERS

IDA WILL BE PREPARED FOR THE INTRODUCTION OF FURTHER TELEMATIC AND DATA TRANSMISSION SERVICES COMPATIBLE WITH THE OPEN SYSTEMS INTERCONNECTION PROTOCOLS

DATA TRANSMISSION CONTINUES TO DEVELOP ON THE TELEPHONE AND OTHER NETWORKS

TABLE 6

DATA COMMUNICATIONS DEVELOPMENT PHASE 3
NATIONAL COVERAGE AND SERVICE PROVISION BY THE IDA

IDA CONSTITUTES THE PUBLIC DATA NETWORK OF HUNGARY, IT INTEGRATES TELEGRAM, TELEX, CIRCUIT SWITCHED DATA TRANSMISSION, PACKET SWITCHED DATA TRANSMISSION, LEASED CIRCUIT DATA TRANSMISSION, AND DATABASE ACCESS SERVICES FOR ASYNCHRONOUS, SYNCHRONOUS, AND PACKET MODE DATA TERMINALS

IDA OFFERS FULL AVAILABILITY FOR TELEMATIC SERVICES AS TELETEX, SUPPORTS TELEFAX AND VIDEOTEX SERVICES ON PSTN AND BUREAUFAX AND ELECTRONIC LETTERPOST SERVICES PROVIDED BY POST OFFICES

IDA PROVIDES A UNIFIED QUALITY OF SERVICE FOR ANY USER OF ANY SERVICE IT PROVIDES, AND EQUAL AVAILABILTY OF SERVICES

IDA SUPPORTS THE INTROQUCTION OF FURTHER DATA AND TEXT COMMUNICATION SERVICES INCLUDING MESSAGE HANDLING,AND SUPPORT THE EMREGING ISDN ISLES WITH ALL THE SERVICES IT PROVIDES BY THE INTERCONNECTION FACILITIES TO THAT ISDNS

IDA SERVES AS A RELIABLE CARRIER OF INFORMATION FOR ANY COMPUTER-COMMUNICATION APPLICATION MEANWHILE DATA TRANSMISSION CONTINUES TO DEVELOP ON THE TELEPHONE AND OTHER NETWORKS INCLUDING ISDN /IF EMERGING/

NOTE: FURTHER DEVELOPMENT DEPENDS ON THE TIMESCHEDULE OF ISDN

TABLE 7

THE ASPECTS OF AN ISDN

ISDN READS:	INTEGRATED SERVICES DIGITAL NETWORK
ISDN IS INTERPRETED	AS THE INTEGRATION OF THE SERVICES OFFERED BY A SINGLE DIGITAL TELECOMMUNICATION NETWORK INSTEAD OF A SET OF DEDICATED SERVICES NETWORKS
ISDN MEANS	A TELECOMMUNICATION NETWORK COMPOSED FROM DIGITAL SWITCHING EXCHANGES AND DIGITAL TRANSMISSION LINKS THAT CARRY DIGITAL SIGNALS ON DIGITAL USER-TO-USER /END-TO-END/ CONNECTIONS,THAT SUPPORT A WIDE RANGE OF SERVICES AND USE A LIMITED SET OF CONNECTION TYPES AND MUTLIPURPOSE USER-NETWORK INTERFACE ARRANGEMENTS
ISDN IS FEASIBLE	WHERE THE TELEPHONE-IDNS AND NON-TELEPHONE IDNS ARE SPREAD AND HAVE THE NECESSARY INTELLIGENCE AND FUNCTIONALITY TO BE CONVERTED INTO SOME FORM OF AN ISDN
ISDN IS DESIRABLE	WHERE DIGITALISATION IS ITSELF ECONOMICAL, WHERE SERVICE INTEGRATION IN A SINGLE NETWORK CAN PROVIDE AND SUPPORT MORE SERVICES ECONOMICALLY THAT IN A SET OF DEDICATED SERVICES NETWORKS, WHERE IT IS OPTIMAL FOR THE MANY PUBLIC AND PRIVATE NETWORKS AND SERVICES TO BE SUBSTITUTED BY AND ABSORBED INTO A SINGLE NETWORK LIKE AN ISDN, AND WHERE THE NECESSARY RESOURCES AND MEANS ARE ALL AVAILABLE

NOTE: There are many more aspects not listed, sometimes more serious and therefore more significant to be taken into account when planning and implementing an ISDN.Table is intended to demonstrate the existence of such aspects.

TABLE 8

FACTORS OF APPROACHING AN ISDN

THE ISDN MUST BE BASED ON DIGITAL TELEPHONY - BUT WHERE THE ANALOG TELEPHONY IS OLD,INSUFFICIENT AND UNRELIABE? - AND THERE IS NOT ENOUGH INFRASTRUCTURE TO BASE ON IT A WIDER EXPANSION OF SERVICES? - AND THERE IS A QUITE UP TO DATE CIRCUIT SWITCHED DATA NETWORK ABLE TO OFFER ANY TELEGRAPH, DATA, TEXT, AND TELEMATIC SERVICES ON A HIGH QUALITY BASIS /COMPETING WITH TELEPHONY FOR THE VERY LIMITED FINANCIAL AND INFRASCTRUCTURAL RESOURCES/?
THE ISDN MUST EVOLVE FROM DIGITAL SWITCHING IN TELEPHONY TO A MULTISERVICE AND MULTIRATE SWITCHING FOR ANY OF THE TELECOMMUNICATION SERVICES - BUT WHERE NO MANUFACTURING OF LOCAL AND TRANSIT DIGITAL EXCHANGES EXIST AND NO IMPORT IS POSSIBLE? - AND THE DEVELOPMENT FOR DIGITAL SWITCHING SYSTEMS ARE MADE ONLY FOR TELEPHONE REQUIREMENTS? - AND THE IDEAS ARE EMERGING THAT ISDN CAN BE BASED ON A PABX WHICH IS DIGITAL AND OFFERS CENTRAL OFFICE SERVICES OR ON AN INTELLIGENT REMOTE CONCENTRATOR?
THE ISDN MUST BE A DIGITAL NETWORK WHICH OFFER CIRCUIT TYPE AND PACKET TYPE BEARER SERVICES FOR DIGITALLY CONNECTED USER TERMINALS - BUT WHERE THE SEMICONDUCTOR INDUSTRY DOES NOT PRODUCE THE NECESSARY LSI AND VLSI DEVICES TO IMPLEMENT THE TERMINALS AND ACCESS ARRENGEMENTS? - AND WHERE THE COST STRUCTURE OF ADVANCED TECHNOLOGY IS INVERSE TO THAT EXPECTED FROM IT /FUNCTIONS IMPLEMENTED IN NEW TECHNOLOGY COST MORE THAN IN THE OLD ONE/? - AND WHERE AT THE TIME OF DIGITALIZATION BY THE STRATEGY OF ISLES IS SOME AREAS THERE WILL BE NO DIGITAL TRANSIT EXCHANGE IN OPERATION?
THE ISDN MUST BE AN INTELLIGENT NETWORK USING HIGH LEVEL OF NETWORK CONTROL AND NETWORK MANAGEMENT - BUT WHERE NETWORK CONTROL AND NETWORK MANAGEMENT ARE PLANNED TO BE CREATED SEPARATELY FOR TELEPHONY AND FOR NON-TELEPHONY APPLICATIONS? - AND WHERE THERE IS NO IDEA TO USE CCSS AT ALL? - AND WHERE THERE ISVERY LITTLE UNDERSTANDING WHAT NETWORK MANAGEMENT IS AND MAKES FOR THE DIFFERENT SERVICES?

NOTE: QUITE AN UNLIMITED SET OF ADDITIONAL STATEMENTS CAN BE GENERATED ON WHAT ISDN HAVE TO BE AND WHAT DIFFICULTIES EXIST THAT MAKE ISDN DELAYED IF NOT DOUBTFUL AT ALL.

TABLE 9

KEY ISSUES OF AN ISDN

THE ISDN SHOULD BE PLANNED AND PREPARED TO PROVIDE ORDERLY EVOLUTION AND GROWTH,THEREFORE IT NEEDS

- FINANCIAL STABILITY AND AVAILABILITY OF EXTERNAL SUPPORT,
- MANAGEMENT STABILITY AND CLEAR STATEMENT OF MEANS AND ENDS,
- PRODUCT AND SERVICE STABILITY AND ORGANIZED COOPERATION BETWEEN USERS, MANUFACTURERS AND SERVICE PROVIDERS.

THE ISDN SHOULD BE ACCEPTED AND DESIRED

- BY ITS INTENDED USERS WHO PAY FOR IT AND STATE THE REQUIREMENT TO BE MET,
- BY ITS INTENDED SUPPLIERS WHO INVEST IN RESEARCH AND DEVELOPMENT AND MANUFACTURING,
- BY ITS INTENDED "REGULATED SERVICES" PROVIDER WHO INVEST IN THE BEARER NETWORK,
- BY ITS INTENDED "VALUE ADDED SERVICES" PROVIDERS WHO INVEST IN THE PROCESSING CAPABILITIES OF THE ISDN.

THE ISDN SHOULD BE BASED ON CLEAR DISTINCTION BETWEEN REGULATED AND COMPETITIVE MARKET SERVICES

- TO LET THE USER GAIN MOST FROM THE COMPETITION ON THE VALUE ADDED SERVICES,
- TO LET THE USER GAIN MOST FROM THE MONOPOLY OF THE REGULATED BEARER SERVICE,
- TO LET THE USER RESELL IN A REGULATED MANNER ANY PORTION OF THE SUBSCRIBED BUT UNUSED TELECOMMUNICATION CAPACITY.

THE ISDN SHOULD BE BASED ON A TARIFF STRUCTURE REPRESENTING COSTS

- TO BE INDEPENDENT FROM THE TYPE OF INFORMATION TRANSFERRED /VOICE,DATA,TEXT,IMAGE,FACSIMILE DOCUMENT/,
- TO BE INDEPENDENT FROM THE CONNECTION TIME /IF THAT TIME DOES NOT MEAN TIME USING RESOURCES/,
- TO BE DEPENDENT OF THE AMOUNT OT RESOURCES USED,
- TO BE DEPENDENT ON THE NUMBER OF BITS TRANSMITTED.

THE ISDN SHOULD BE BASED ON METHODS ELABORATED IN DATA COMMUNICATION

- ON DIGITAL INTERFACES BETWEEN DTES AND DCES /TES AND NTS/,
- ON THE OPEN SYSTEMS INTERCONNECTION ARCHITECTURE MODELS,
- ON THE INTEGRATION OF CIRCUIT SWITCHING AND PACKET SWITCHING,
- ON THE UNLIMITED CAPABILITY TO BASE SERVICE OFFERINGS ON REGULATED TELECOMMUNICATION SERVICES: THE POSSIBILITY TO SUPPORT VALUE ADDITION.

CONNECTION OF PERSONAL COMPUTER NETWORKS TO ISDN

András Filp
Budapest Technical University
Hungary

The paper deals with the connection of a small local area network comprising personal computers, to ISDN. Design aspects of the terminal adapter are discussed in terms of the ISO-OSI model. The necessary protocol and speed conversion problems are discussed in some detail.

ISDN AND LAN PROTOCOLS

The ISDN specifications, given as CCITT recommendations allow to connect different devices and networks to ISDN. For that purpose the recommendation specifies some reference points, and functional groupings. The configuration available at the customer's premise is shown in Fig. 1.

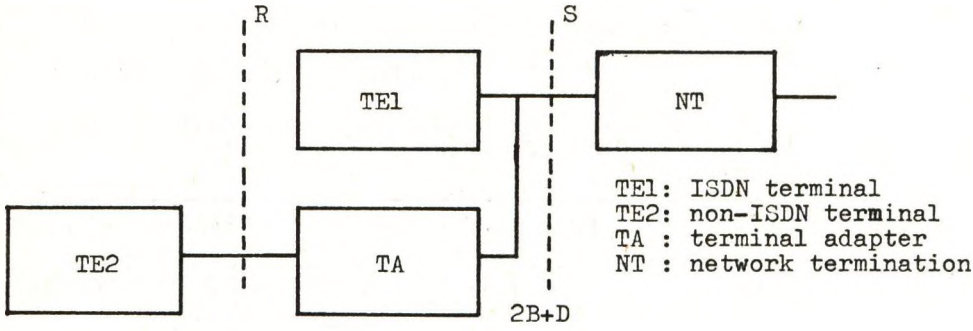


Figure 1
User-network interface reference configuration

Devices, that do not comply the ISDN recommendation are connected to reference point R and a terminal adapter TA is used to map this point onto reference point S.

The ISDN recommendations specify the available functions as follows:

NT functions:
- network termination
- timing
- 1st layer line maintenance
- 2nd and 3rd layer protocol handling

- maintenance

TE functions: - protocol handling
- maintenance
- interface functions

There are two types of terminal equipments, TE1 and TE2. TE1 performs all TE functions and contains the ISDN-user interface, while TE2 doesn't have the latter.

Not only the functions but the transfer channels are specified, too. At reference point S, the so-called basic ISDN access contains two type B channel, with 64 Kbit/sec transfer rate and a D channel, with 16 Kbit/sec transfer rate. Channel B can be used for transmission of speech, data and other digital information. Channel D is used first of all, for signalling purposes and data transmission, too in circuit switched transmission, and in packet switched systems for data transmission as well. Features of the basic ISDN access are defined in terms of the ISO-OSI seven layer model.

Physical layer functions include transmission, contention resolution, power feeding, activation/deactivation and maintenance. Data link layer functions in ISDN refer only to the D channel, and are called Link Access Procedures on channel D. The LAPD supports multiple terminal configurations and multiple layer 3 entities. The network layer protocol for channel D gives the possibility to establish, maintain and terminate connections within the ISDN. These connections can be circuit-switched, user-to-user signalling and packet-switched. The layer 1, 2 and 3 ISDN protocol are shown in Fig. 2.

As far as the LAN protocols are concerned, there is no general international standard or recommandation for them. These protocols, in general, have the following features:

- LAN protocols depend on the topology of the network
- there is no strict distinction between data link and network layer functions
- they can be either byte or bit oriented
- they have different line access methods
- they have different acknowledge systems

F	L	B_1	L	D	L	F_A	L	B_2	L	D	L	B_1	L	D	L	B_2	L	D	F

F: framina bit
L: DC balancia bit
D: D-channel bit
B_1: 8 bits of B1 channel
B_2: 8 bits of B2 channel

Figure 2.a. Frame structure for the S-interface

Flag 8 bits	Address 8 bits	Control 8 bits	Information n*8 bits	Frame checking sequence 16 bits	Flag 8 bits

Figure 2.b. Data link protocol frame structure

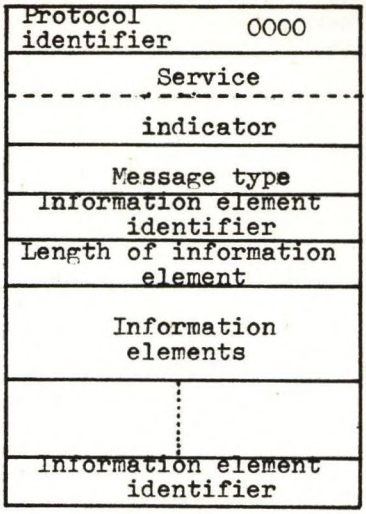


Figure 2.c. Network layer message structure

Figure 2
ISDN protocol formats

These features don't agree with the ISDN protocols of any level. A characteristic LAN protocol, the Ethernet protocol can be seen in Fig. 3.

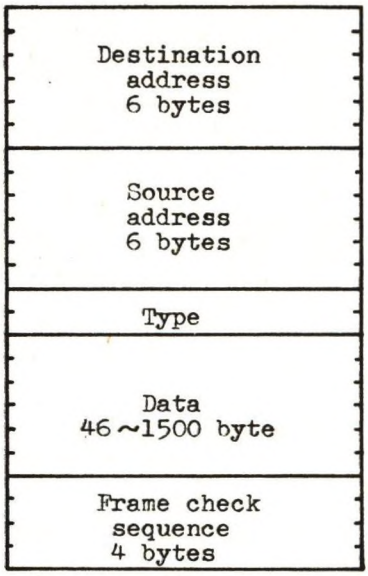


Figure 3
Ethernet data link layer protocol

ISDN-LAN CONNECTION

There are tow basic kinds of ISDN-LAN connections. In the first case the LAN terminals and users intend to connect to an ISDN terminal, or want to use an ISDN service. This is the so called "gateway" function, and is shown in Fig. 4.a. The second kind of connection is when two LAN terminals are connected through an ISDN. This is the "bridge" function, and Fig. 4.b. shows it.

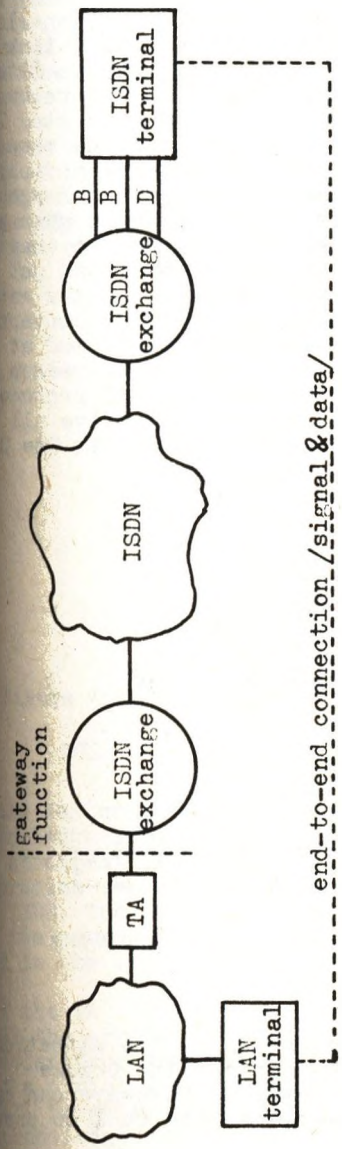


Figure 4.a. TA in a "gateway" configuration

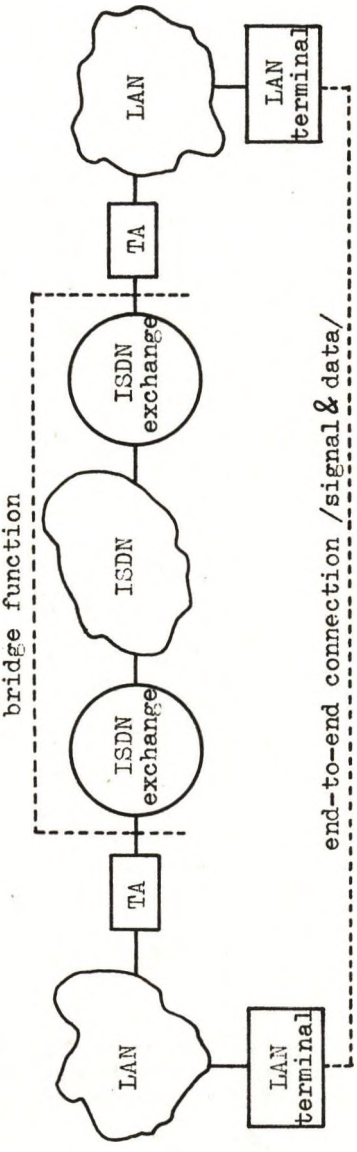


Figure 4.b. TA in a "bridge" configuration

Figure 4
Connection of LAN and ISDN

What should a TA perform in these cases?

The TA as a gateway should:

- perform conversion between ISDN and LAN protocols
- translate address from LAN to ISDN and in reverse direction
- equalize the different speed of the two networks
- provide appropriate physical interface for both networks

In a gateway configuration the terminals of the LAN can consider the ISDN terminals and services as if they were connected to the LAN, and were all represented by TA. So the LAN terminals address the TA and request a connection or a service. Addressing method is the same as for "real" LAN terminals or units. This message arrives to TA in three "envelopes": physical, data link and network layer envelopes of the LAN. The TA unit has to perform all physical, data link and network layer functions of LAN in order to get the message itself. From the innermost /network layer/ envelope and from the requested service, the TA can define the exact address of the requested ISDN terminal or service. This address is then placed into the message and the message is given to the ISDN network, data link and physical layer functions. Data transfer in the opposite direction happens in the same way. /The situation would be somewhat different when an ISDN terminal initiated a message towards the LAN, but we consider a LAN as a private network, so a standard ISDN terminal can not address LAN terminals and services./ As can be seen, in the gateway configuration the most important TA functions /protocol conversion and address translation/ can be most economically performed at any of the upper four levels of the ISO-OSI model. The main reason for this is the significant difference between ISDN and LAN protocols, which reflects the fact that in a LAN all stations receive all messages /on the physical level/ while in the ISDN don't. The OSI model of the gateway TA is shown in Fig. 5.a.

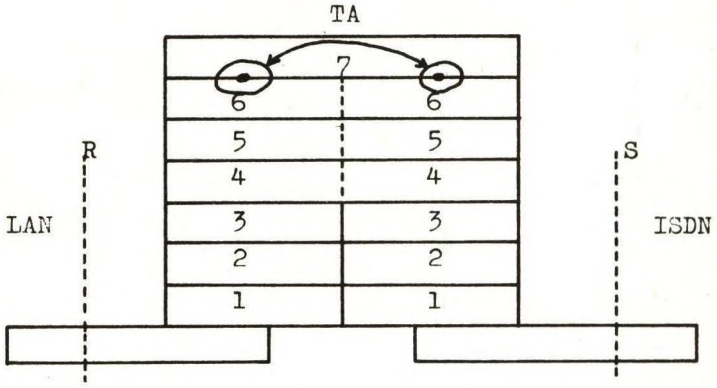


Figure 5.a. OSI model of the gateway TA functions

When the ISDN serves as a bridge between two LANs, the situation is somewhat different. There are two ways to perform the bridge functions: two TAs, configured and programmed as gateways can operate as a bridge connection, or the TA serves as the tool to provide complete transparency of the ISDN.

When two gateways are used as a bridge there is just a little modification in the original gateway functions, namely in the address translation. While in a true gateway configuration, the LAN address

is translated into an ISDN address, in the bridge configuration an ISDN address is derived from the LAN address, but the LAN address is transmitted in its original form, too. This is necessary in order to find the addressee of the message. Otherwise the transmission process is the same as the above mentioned.

There is an other way to perform the bridge function. The main feature is that the physical layer level LAN message is considered as the application-transport layer of an ISDN message. This is an unbalanced connection of the LAN-ISDN transport functions. This solution is simpler than the previous one, as there is no protocol conversion at all. The TA performs only a limited set of a layer 2 functions on the LAN side. This limited set of functions are necessary in order to derive the ISDN address.

Although the message, transmitted through the ISDN is longer /it contains the level 1, 2 and 3 LAN envelopes, too/ the simpler TA functions can overwhelm this disadvantage, and such a TA is almost completely insensitive to the changes in the LAN protocol. This unbalanced connection is modelled in Fig. 5.b.

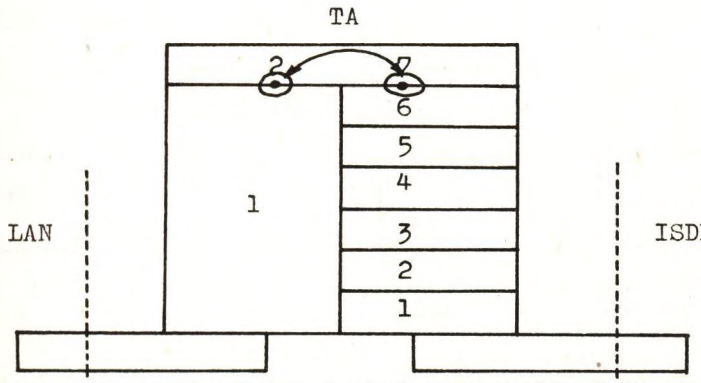


Figure 5.b. OSI model of the "unbalanced bridge" TA functions

Figure 5
ISO OSI models of TA functions

In the design process of a LAN protocol an important question is the acknowledge system. This is the area, where the ISDN connection has the strongest influence. Either in the gateway or in the bridge configuration the message proceeds through at least two networks, LAN and ISDN. The data link layer acknowledgement, in general is to signal the correct or errorneous reception of a message. When a TA is used in the LAN, the following problems arise:

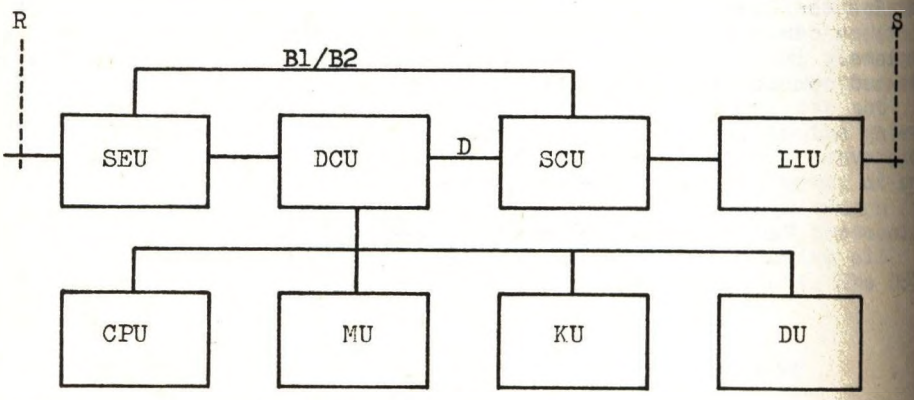
- if the TA acknowledges a message to the sender immediately, then all the errors, that occur in the ISDN or in the other LAN, are hidden from the sender, so there is no way to repeat the message,
- if a strict acknowledge chain is implemented, for example in the LAN1 terminal - TA1 - ISDN - TA2 - LAN2 terminal path, then the acknowledge message have to pass this way backward, and the acknowledge time can be considerably longer than the same parameter in the LAN,
- even if this long acknowledge time is acceptable another trouble occurs in the TA functions, namely that a TA can emit a negative acknowledge answer in two cases, first when detects an error in

the message from the LAN terminal, and second when such an answer arrives from the ISDN.

All these difficulties suggest, that in a LAN which is to be connected to an ISDN, the acknowledgement should not be implemented in the data link layer protocol. The overall qualification of the transfer can be implemented on the higher /upper four/ layers of the LAN.

THE STRUCTURE OF A TA

In a LAN of personal computers the TA itself is an intelligent device, a microcomputer, too. It can be implemented with 8 main units, as can bee seen in Fig. 6.



SEU: speed equalization unit
DCU: D channel control unit
SCU: S interface control unit
LIU: line isolation unit
CPU: central processing unit
MU : memory unit
KU : keyboard unit
DU : display unit

Figure 6
Functional structure of a TA

Four of them are conventional microcomputer units:

- central processing unit /CPU/
- memory unit /MU/
- display /DU/
- keyboard /KU/

These are responsible for the functions of the layer 2 - layer 7 in the OSI model, as well as for the above detailed tasks. Some other functions, for example maintenance, traffic monitoring, are also implemented.

There are some special units in the TA.

In order to perform the physical layer functions of reference point

S, an S interface control unit, SCU is applied. In connection with the S interface a line isolation unit is also necessary, to isolate the TA from the line and to perform the necessary electrical signal--conversions. The D channel link access procedures are executed by the D control unit, DCU. This unit is based on a HDLC controller element, and can itself contain some intelligence.

A speed equalization unit, SEU can balance the different speeds of LAN and ISDN provide the physical interface for the LAN, on reference point R.

ACCESSING VIDEOTEX THROUGH THE ISO FILE ACCESS PROTOCOL

J. Barchanski*
G.V. Bochmann**
P. Desjardins**
J. Gecsei**

* Brock University, St.-Catharines, Ontario, Canada
** Université de Montréal, Québec, Canada

The paper describes the design of an OSI - structured access protocol whose purpose is to provide a uniform way of querying different page-oriented external videotex databases. The leading principle of the design is that the external database is unaware of the identities of requesting users. The application layer of the protocol consists of a subset of the ISO File Access Protocol proposal extended by some videotex-specific primitives. The user interface used with the protocol is based on CCITT Recommendation F.300.

1. Introduction

It has been repeatedly argued [Clar 81, Gecs 83] that for a videotex system to be successful, it is essential that it provide access to a number of services "external" to the system itself. Hence the need to develop standardized protocol for such access. There exist several gateway protocol implementations [Gilb 81, Edwa 82]; the novelty of the work reported here is in the use, to the maximum possible extent, of some existing CCITT and ISO protocol proposals for OSI [OSI 83] to construct a videotex database access (VDBA) protocol. Its purpose is to provide a uniform way (as seen by the user and his access machine) of querying a class of page-oriented external databases. The main problem in designing such a protocol is to reconcile the contradictory requirements of creating a simple and uniform user interface, without a substantial loss of service features present in the individual databases. For example, how to map the "document page" feature of Telidon databases and the multiframe pages in Prestel and the different command structures of these databases, onto a unified interface? This paper outlines a solution to this problem which has been implemented on a VAX/750 computer.

The main principles underlying our solution are: (1) The external computer is shielded from the identities of the requesting users. In effect, the external database is used as a "frame store"; all user-dependent information is handled in the access machine. (2) To achieve this, all session-dependent information (also called user context) is kept in the access machine, and thus need not be maintained in the external computer. Data from external sources are requested normally by sending the identifiers of the demanded frames. However, the user's main retrieval tool is menu selection. (3) In addition to

the above approach, the possibility of a transparent mode between the user terminal and an external application is provided. In this case items 1 and 2 above are not achieved, but only the ability to logon to the external machine via a simple selection from a menu of databases is provided. After logon and service selection, as handled by the access machine the users will see the external database without any adaptation, and they will have to interact with it in terms of its native command structure. Note that such is also the principle of the iNET network [Edwa 82] .

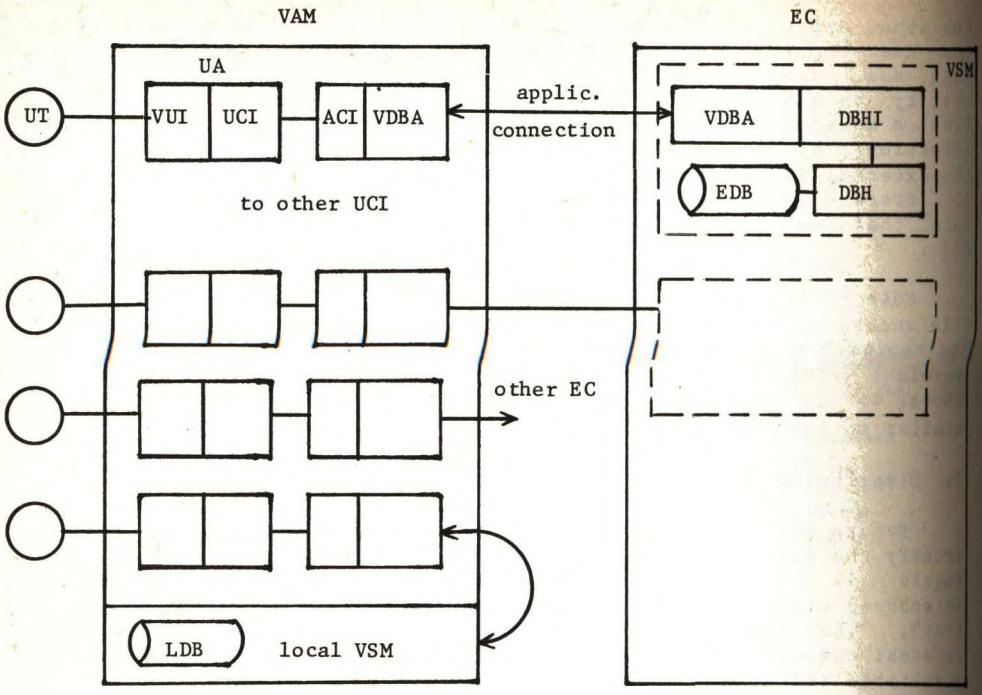
Note that our solution with points (1) and (2) above is a file access protocol in the OSI sense. This is in contrast with the approach used in Architel [Arch 82], which defines a virtual terminal protocol. However, our transparent mode, point (3) above, can be considered as a primitive virtual terminal protocol similar to the one used in iNET .

2. Distributed Videotex Network

Before describing the OSI-based access protocol, we discuss briefly the generic network under consideration which is shown in Figure 1. User terminals (UT) are linked (e.g. through telephone) with a service computer called videotex access machine (VAM). The external (third party) databases are running on external computers (EC); these communicate with the VAM through a suitable transport link (see below).

The access protocol to be described is implemented in VAM and ECs by videotex database access (VDBA) modules which maintain an application connection. One of the functions of VDBA modules is to keep track of multiple frame requests and of their correct routing to the users who originated them. The database handler interface (DBHI) interprets the commands and frame requests received through the VDBA and generates the control information (including links to subsequent frames) to be sent back to the VAM. DBHI is basically an adaptor hiding the particularities of the external database and its handler (DBH) and maintaining a uniform interface with VDBA. Each instance of VDBA and associated external database is called a videotex server machine VSM. There may be several VSMs in a given EC; each VSM is served by a separate application connection (possibly multiplexed over a single transport link).

The VAM maintains a link with a given VSM through an access module (AM), composed of VDBA and a multiplexor (ACI) whose function is to return the data received from the VSM to the appropriate user. Another component found in the VAM is the user agent (UA); there is one UA per active user terminal. The main function of a UA is to maintain session an user- specific information in a data structure called user context; this latter is centered around the control information of the currently accessed frame (essentially the list of available menu selections and corresponding frame identifiers). Additional UA functions include maintaining an access trace (including transitions between databases), and statistics/accounting functions.



VAM : Videotex Access Machine
EC : External Computer
UA : User Agent
UT : User Terminal
VUI : Videotex User Interface
UCI : User Communication Interface
ACI : Application Communication Interface
VDBA: Videotex Database Access
DBHI: Database Handler Interface
DBH : Database Handler
EDB : External Database
VSM : Videotex Server Machine
LDB : Local Database

Figure 1: System configuration and structure

Finally, a connection of a UA with a particular AM is routed through a user communicaiton interface (UCI). The setting of the UCI is determined by the current service (database) selection made by the user in interaction with the UA. At the beginning of a session, a user is connected to a local database (LDB); later he may be switched to a VSM through the intermediary of a gateway page and of VDBA. The typical (and minimal) use of the LDB is a directory of available VSMs.

In the above scenario, the effect of VDBA, DBHI and UA is that the external database's interface is masked to the videotex users. Therefore we call this access protocol nontransparent. The alternative is transparent access , in which the roles of UA and VDBA are reduced to the creation of a transport connection between the UT and the EC.

3. Protocol Levels

The protocol used for nontransparent access between the EC and VAM is oriented towards the Open Systems Interconnection (OSI) architecture [OSI 83] developed by ISO and CCITT and is based on a generalized videotex database structure [Berg 84]. The overall structure of the VAM-EC connection is shown in Figure 2. For the lower protocol layers, including the transport layer, two possible configurations are provided: (a) in the case that DECnet communication software is used by both computers, the NSP communication service is considered to be the transport service. (b) In the case that OSI protocol standards are used for the lower communication layers, the use of X.25 and an ISO/CCITT transport protocol is foreseen. For nontransparent access protocol only one transport connection is required between the EDB and AM; therefore protocol class 0 or 1 (the latter providing additional reliability) can be used. In the case of transparent interworking where a separate transport connection is required for each interactive user, the use of protocol classes 2 or 3 is more suitable because they allow multiplexing over a single network connection.

The higher level protocols are modeled after working documents of ISO and CCITT [OSI FP, OSI SP]. For the session layer, the minimal two-way simultaneous subset may be used. However, in the initial version of the implementation, a null session layer was chosen. The presentation layer functions include a facility for seclecting the coding used for the information to be displayed. In the initial version of the implementation, this facility is included in the application layer protocol. The application layer includes two sublayers: (a) A subset of the ISO File Transfer protocol, and (b) the adaptation of this subset to the particular requirement of access to a menu driven page-oriented videotex database.

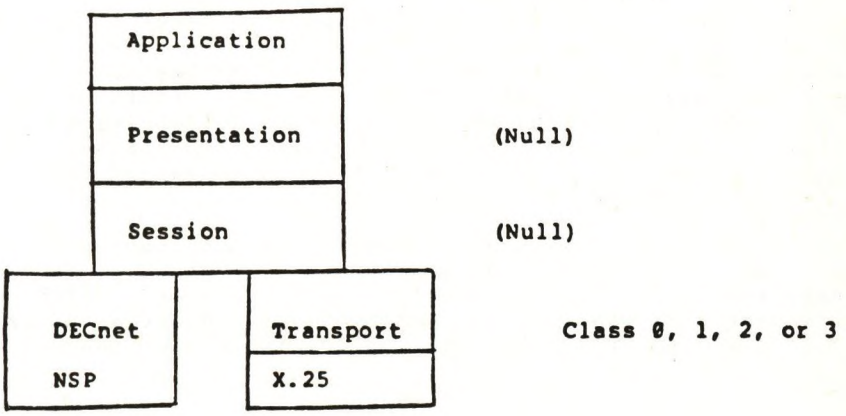


Figure 2 VAM-EC Connection Structure

This adaptation is reflected mainly in the service primitives A-READ and A-Data (see below). The first contains the identifier of a frame to be retrieved from the EC. The second is used to transfer the requested frame back to the VAM, along with a number of "links"; each link contains among others, a (selection label - selection frame specification) pair. The selection label is a menu choice to be displayed and possibly selected by the user; the frame specification is the corresponding frame identifier to be sent back to the EC, if its selection label is entered by the user.

User input is treated as follows: first the links of the current page are searched for a selection label matching the user's selection; if found, the input is considered a menu selection and the corresponding frame specification is sent to the EC as the frame request. If no match is found, the user input is checked to see if it is a local interface command, and if not it is sent without any change to the EC. In this way, commands specific to the external database (e.g. keywords or Telidon "F" commands [Teli 81]), or answers to prompts appearing on the screen as text (but not present in the frame's control information) can be correctly treated by the DBHI.

4. VDBA service and protocol

4.1. General

The VDBA protocol provides a subset of the service defined in the ISO File Service Definition [OSI FP] and is based on the ISO File Protocol Specification [OSI SP]. It contains only those functions which were considered necessary for access to external videotex databases. However, these functions are specialized to the case of menu-driven selection from page-oriented databases. Morever, the protocol incorporates some features for negotiation of data representation, providing thereby a basic presentation service.

Protocol operation proceeds as a sequence of interactions between two application entities residing in a videotex access machine and an external computer. The application entity in the VAM initiates all interactions and is therefore called the initiating entity. The application entity residing in the EC and being connected to one or several databases will be called the responding entity. These two entities correspond roughly to the UA-AM pair and the VSM, respectively.

The initiating and responding entities cooperate over an application connection established at the start of the protocol operation. An EC may contain several databases which may use different data representations. Users requesting access to the same database share the same application connection between the VAM and the EC.

4.2. VDBA Service Structure

The VDBA is connection-oriented. Operations over a given connection are structured into the following phases:

a) creation of an application connection between VAM and an initial VSM;
b) target server selection, which identifies a VSM and establishes an association with it;
c) target server opening, which establishes the environment for the data transfer;
d) data transfer, which performs and manages the movement of the data;
e) target server closing, which releases the environment for the data transfer;
f) target server release, which releases the association with the selected database server;
g) application connection release which breaks the

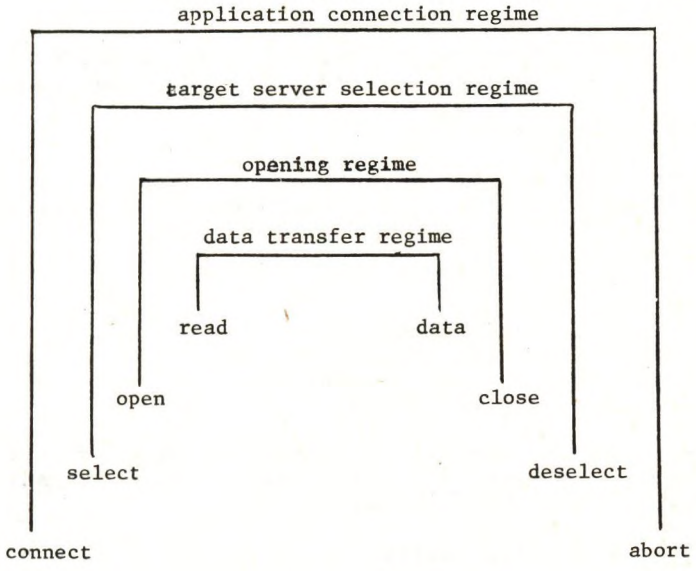


Figure 3 VDBA service regimes

association between the MPX module and the server.

These phases give rise to the following regimes, which are shown in Figure 3:

1 -- application connection regime;
2 -- server selection regime;
3 -- server open regime;
4 -- data transfer regime.

These regimes are nested in the given order : terminiation of a lower numbered regime implies termination of any higher numbered regime. There is at most one instance of each regime at one time, but a regime may be released and a different regime of the same kind established as often as required. For instance, a given application connection may be used, alternatively for communication with several database servers.

4.3. Service primitives

This section provides a list of primitives used by the VDBA Service. Details on these primitives can be found in [OSI FP]. The primitives for the data transfer regime contain videotex-specific extensions, and are described in more detail in [Barc 84].

Application connection regime:

- A-Connect (request, indication, response, confirm)
- A-Abort (request, indication)

Server selection regime:

- A-Select (request, indication, response, confirm)
- A-Deselect (request, indication, response, confirm)

Server open regime:

- A-Open (request, indication, response, confirm)
- A-Close (request, indication, response, confirm)

Data transfer regime:

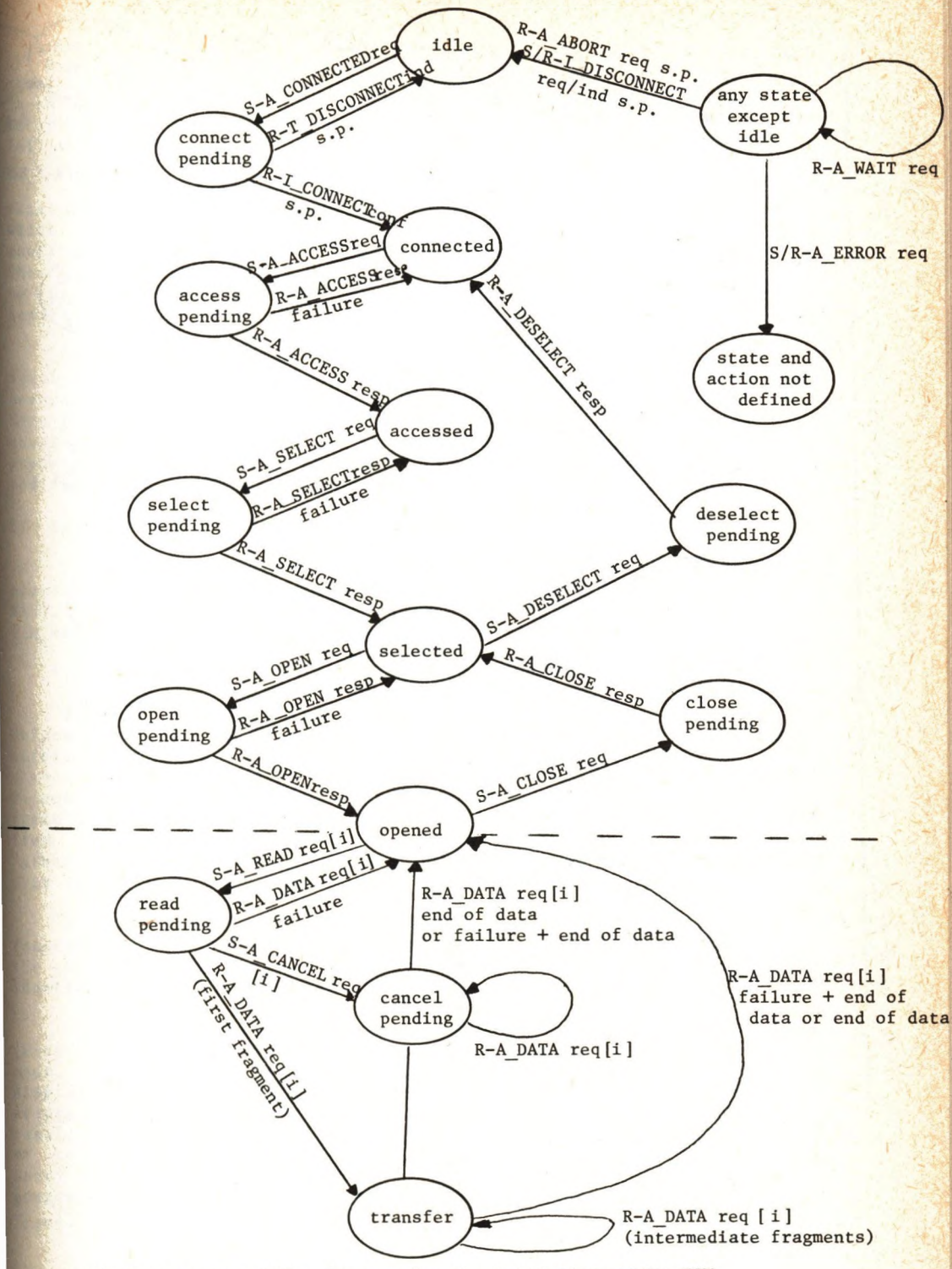
- A-Read (request, indication)
- A-Data (request, indication)
- A-Cancel (request, indication)
- A-Exception (indication)

The user may cancel his request to read a frame using an A-Cancel request primitive, which may be issued at any time after the A-Read request service primitive.

The A-Exception indication primitive is used by the service provider to signal exceptional conditions which have perturbed the transfer of data. The primitive carries a diagnostic parameter indicating the nature of the exception.

4.4. Overview of the Service Operations

The MPX module and the server processes are considered here as finite state machines FSM interconnected through the initiating and responding application protocol entities FSMs. The service primitives are the I/O functions of the FSMs. The behaviour of the MPX module and the server processes in terms of state transitions is represented by the detailed state transitions diagrams on Figures 4 and 5. The upper parts of these diagrams, between the "idle" and "open" states pertain to the application connection, server selection and opening regimes



g. 4 Detailed state transition diagram for the initiating entity FSM.
S-sent; R-received; req-request; resp-response; ind-indication;
[i]-reference number; s.p.-service primitive.

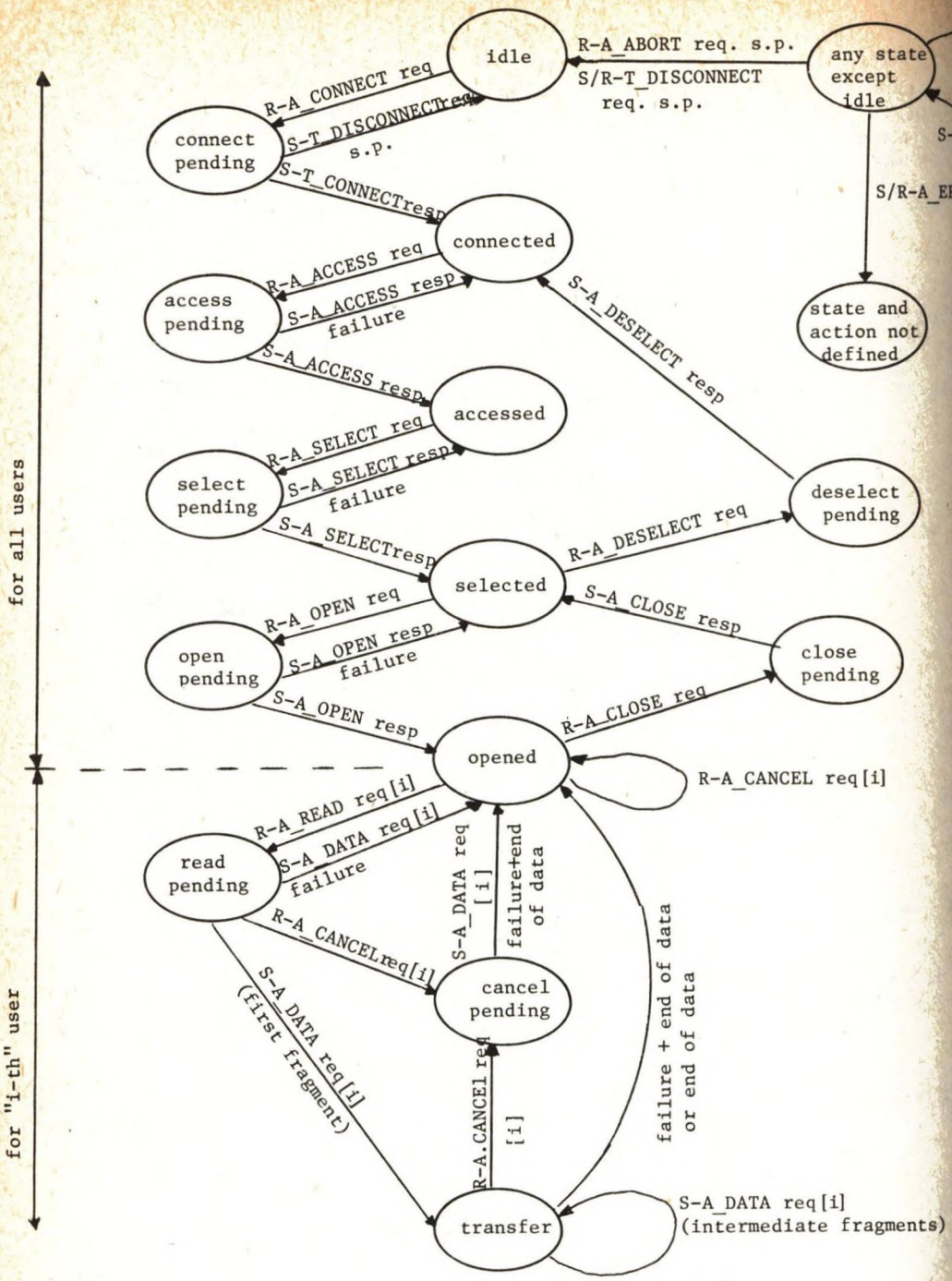


Fig. 5 Detailed state diagram for the responding entity FSM.

S-sent; R-received; req-request; resp-response; [i]-reference number; s.p.-service primitive.

of a given connection which can be shared among several terminal users. The lower parts represent the behaviour of a particular "i-th" user agent, as identified by a particular value of the the reference number. The sequence of service operations is as follows.

After establishing an application connection, a specific database server is selected. The different data representations which may be handled by this server are indicated by the A-Select confirm primitive. The MPX module selects (if at all) one of them and indicates the choice in the A-Open request. If there is no matching data representation, the MPX module deselects the server.

An "i-th" user agent may get access to a particular frame by issuing an A-Read request service primitive. Each user may issue several such requests, without waiting for a response. He can have thus several views of information from one or more databases quasi-simultaneously on the same screen (if this parallelism is supported by the user agent in the VAM and by proper presentation coding).

5. Database organization and user interface

This section concerns two problems complementary to the VDBA protocol: a possible structure for the local database (in the VAM), and the (uniform) interface presented to the user by the VAM while querying local or external databases.

5.1. Generalized videotex database structure

This section presents a simple database structure for menu-driven retrieval which can be considered a generalization of many existing videotex databases. The user interface for this structure, described in more detail below, is based on the CCITT recommendation F.300 [F.300]. It allows the user to select a frame by absolute or relative access, to retrace or to get help from the system. We call a database using this structure a "generalized videotex database" (GVDB). An implementation of a GVDB is described in [Berg 84].

A GVDB is composed of logical units of information called frames, interconnected through a network of links determined by the information provider. Each frame of a given GVDB is uniquely identified by a frame identifier.

Each frame has a parameter specifying the frame's type (display, sound, software, gateway, etc.). When accessed by a user, a frame will provoke some action depending on its type. A display frame generates a display on the user's terminal according to the presentation coding used. A software frame results in the execution of a program carried in the frame itself or referenced by it. A gateway frame contains a pointer to the entry frame of another database. The end result of accessing a gateway frame is the automatic access of the referenced entry frame.

There is no constraint on the structure of possible links; no underlying hierarchical arrangement is assumed. Mnemonic names (identifiers) or numbers (as in Prestel and Telidon) can be explicitly given to frame identifiers; otherwise a system-assigned identifier is generated for each new frame. At any rate, each frame has a unique (absolute) identifier which can be used to access that frame at any time. In addition, if frame A is linked to frame B by a (directed) link, then B can be accessed from A through a menu selection. Selections can be numeric or alphanumeric, and not all available selections need to be displayed to the user. Some selections may be obvious from the application context, others may be standard (e.g. the "Next frame" command).

In a more formal way, GVDB can be thought of in terms of a relational database consisting of two relations:

```
FRAMES:  frame_id   x    content
LINKS:   frame_id   x    selection x next_frame_id
```

where "frame_id" and "next_frame_id" are frame identifiers, "content" stands for presentation information, and "selection" is a character string typed in by the user in response to a menu displayed on the frame identified by "frame_id". Further, each tuple in the first relation contains a Boolean parameter specifying if the frame in question is an entry point to the database. Each tuple in the second relation contains a parameter specifying if the link in question is a stage (see below). However, this relational model does not imply that the database need to be implemented as a relational database.

5.2. Videotex user interface

The user interface for GVDB described above consists of a number of functions commonly found in videotex systems. In the following, we list the operations available for the users of the interface. All of them are contained in CCITT Recommendation F.300, with the exception of PREVIOUS_STAGE. Each operation consists of a request primitive initiated by the user and a confirmation primitive returned by the user agent, and which is based on data retrieved from the database. Request and confirmation primitives consist of lists of parameter values.

GET_ABSOLUTE

request parameter: frame_id
confirmation parameter: content of the frame, or "Frame does not exist".

This operation results in a query in the FRAMES relation and returns the content of the specified frame, if found in the relation.

GET_RELATIVE

request parameter: selection
confirmation parameter: frame content, or "Invalid selection" or "Frame does not exist".

After a user has made a menu selection, the operation first generates a search into the LINKS relation. The frame_id used is the identifier of the current frame, maintained in the user context. The resulting next_frame_id is used to search the FRAMES relation and the resulting frame content is sent to the terminal. If either search fails, the appropriate message is generated.

NEXT_FRAME

The operation is a special case of GET_RELATIVE

RETRANSMIT

request parameter: none
confirmation parameter: last frame content.

PREVIOUS

request parameter: n (positive integer).
confirmation parameter: n-th last frame content (in reverse time order).

PREVIOUS_STAGE

request parameter: none
confirmation parameter: content of the frame pointed to by the last link specified as "stage", found in the chain of links of this session.

Stages are important reference points in the user's path through the database, intended as navigational aids. If no stage is found, an appropriate message is sent to the user.

PREVIOUS_ENTRY

request parameter: none
confirmation parameter: content of the entry frame used when entered this database. If the current frame is already the entry frame, then the preceding entry frame will be displayed.

LOGON

request parameter: password or account number
confirmation parameter: "OK" or "Failed".
This operation would not be used when a host system logon is required before reaching videotext access machine functions.

GET_USER_INFO

to obtain information on connect time, page count, charges to date, etc.

HELP

request parameter: none
confirmation parameter: frame(s) with user assistance information.

LOGOFF

DISPLAY_FRAME_ID

request parameter: none
confirmation parameter: frame identifier or the current frame.

CANCEL

cancel reception of current frame

5.3. Syntax of interface operations

The above interface operations are coded as short sequences to be entered by the user. While most videotex systems use basically similar sets of operations, their syntax is widely differing.

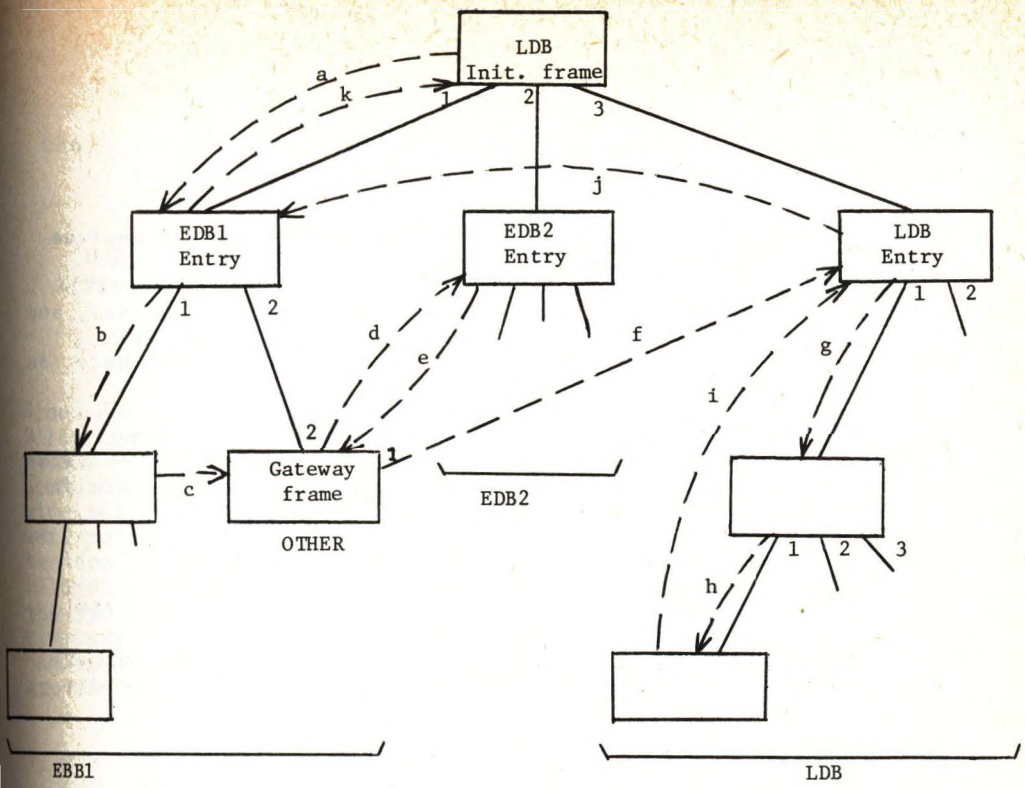
Table 1 summarizes the syntax of Telidon, Prestel and our implementation (see VUI in Fig. 1)

6. Summary

The protocol described in this paper is an OSI-based solution to the problem of external database access in videotex. Two modalities of the protocol offer "transparent" and "nontransparent" access, respectively. The latter offers a uniform user interface to all accessed databases; this interface is based on a "generalized videotex database", also described in the paper. In effect the of indentity of individual databases in hidden; the users perceive essentially a single virtual database. This is illustrated by a session example in Fig. 6. The protocol has been implemented on a DEC VAX/750 machine.

Acknowledgements

This research was supported by a strategic grant No. G1073 of the National Sciences and Engineering Research Council of Canada, and a contract with Digital Equipment Corporation. This paper is based on [Barc 84]. The authors acknowledge the contributions of J.-L. Landry to the development of the protocol.



a - Selection_label (1)
b - Selection_label (2)
c - Frame-id (OTHER)
d - Selection_label (2)
e - Previous
f - Selection_label (1)
g,h - Selection_label
i,j,k - Previous_entry

Figure 6: Sample retrieval session.

Operation	Telidon	Prestel-style	VUI
GET_ ABSOLUTE	frame-id	frame-id	frame-id
GET_RELATIVE	selection-label	selection-label	selection-label
NEXT	>	#	/Next
RETRANSMIT	"	*00	/Retransmit
PREVIOUS (n)	<[n]	*#[n]	/Previous
PREVIOUS_STAGE	(<<)	(**#)	/Stage
PREVIOUS_ENTRY	(<<<)	(***#)	/Entry
GET_USER_INFO	F7	(*7)	/Info
HELP	(...)	(***)	/Help
DISPLAY_FRAME_ID	F8	(*8)	/Display
CANCEL	F	(*)	/Cancel
LOGOFF	F6	(*6)	/Logoff

NOTES:

Character strings in paranthesis are possible extensions of the respective syntaxes.

Character strings in brackets [ ] are optional.

Table 1.

REFERENCES

[Arch 82] Architel, "Protocole de presentation", Note Technique NT/PAA/DIR/746, CNET, France, 1982.

[Barc 83] J.Barchanski, G.v.Bochmann, P.Desjardins, J.Gecsei, and J.L.Landry, "Access protocol to external videotex databases", Doc. de Travail #143, Dept. d'IRO, Univ. de Montreal, 1983.

[Berg 84] D.Berger, "Logiciel de gestion de banque de donnees videotex generalisees applique a l'enseinement assiste avec micro-ordinateur", Masters Thesis, Universite de Montreal, 1984.

[Clar 81] K.E.Clarke and S.J.Wilders,"Implementation of third-party database systems in the UK", Prestel to Third Party Databases Conference, London, March 1981.

[Edwa 82] G.C.Edwards, "A national videotex intelligent network and its role in the information industry", Videotex '82, pp. 149-158.

[F.300] CCITT, "Recommendation F.300: Videotex service", Geneva, 1980.

[Gecs 83] J.Gecsei, "The architecture of videotex systems", Prentice-Hall, 1983.

[Gilb 81] D. Gilbert and I.Taylor, "The Bildschirmtext gateway facility: How does it work?", Prestel to Third Party Databases Conference, London, March 1981.

[OSI 83] Special issue on Open Systems Interworking, Proc. of the IEEE, Dec. 1983.

[OSI FP] ISO TC97/SC21 DP, "OSI - File Transfer, Access and Management", 1984.

[OSI SP] ISO, DIS "OSI - Connection oriented Session protocol specification", 1984.

[Teli 81] "Telidon host database system design specification manual", DOC Canada, 1981.

[Barc 84] J. Barchanski, G.v. Bochmann, P. Desjardins, J. Gecsei J.-L. Landry, "Access Protocol to External Videotex Databases", Publ. No. 500, Département d'informatique et de recherche opérationnelle, Université de Montréal, February 1984.

THE BILDSCHIRMTEXT SERVICE IN THE FEDERAL REPUBLIC OF GERMANY

Heinz Kunze

DBP, Fernmeldetechnisches Zentralamt Darmstadt (F.R.G.)
Division T: Text And Data Communication, Terminal Equipment

Since June 1980, the Deutsche Bundespost has been conducting Bildschirmtext field trials in the areas of Duesseldorf and Berlin/West. The results obtained so far have encouraged the Administration to offer the Bildschirmtext service on a nationwide scale. The hardware and software required for this new network were put up for tender nationally. After evaluation of the submitted tenders a contract for the development and supply of the first Bildschirmtext centres was concluded. The facilities realized in the system are based on CEPT Recommendation T/CD 6-1. In the strategy of introduction special emphasis has been placed on reaching nationwide coverage as soon as possible. A report is given on the experience with the new system during the first year of operation.

1. THE BILDSCHIRMTEXT FIELD TRIALS CARRIED OUT IN DUESSELDORF AND BERLIN/WEST

In autum 1977 the Deutsche Bundespost presented for the first time at the International Radio Exhibition in Berlin its new service 'Bildschirmtext' based on the British Viewdata system. The success of this demonstration encouraged the Administration to start at the beginning of 1978 a nonpublic technical field trial which had the main purpose to make information providers familiar with the new service.

The hardware used for update centres and satellites were GEC-computers of the type 4082.

The software was based on the Viewdata software. In addition, a special software was developed which allowed the connection of external computers to the Bildschirmtext Centres.

So the following types of services could be offered to the subscribers:

- Retrieval of information, e.g. for news, timetables, sports, games etc.,
- On-line dialogues with Bildschirmtext centres or external computers for e.g. tourist office and mail order business,
- Message service for the exchange of information between Bildschirmtext subscribers.

At the end of the trial phase more than 8000 terminals were in operation. About 4 million connections were established to the Bildschirmtext centres. 2000 information providers had offered

some 350 000 Bildschirmtext frames. 53 external computers were integrated, more than 60 are still under test.

The field trials were, of course, observed and the experience gained therein carefully evaluated. The acceptance during the test was excellent, 89 % of the interviewed subscribers expressed their intention to continue using the Bildschirmtext service after the field trials.

So the decision taken by the Federal Cabinet in June 1981 to introduce the Bildschirmtext service on a nationwide scale as a means of individual communication was fully justified.

2. THE CEPT STANDARD

The CEPT Administrations recognized that the then existing forms of representation did not meet all important demands of the users. Therefore, they decided to extend the facilities of the alpha-mosaic mode considerably. After long and sometimes controversal discussions they defined the specifications for the new service in CEPT-Recommendation T/CD 6-1.

Based on CEPT-Recommendation T/CD 6-1 the Deutsche Bundespost published a specification for a basic Bildschirmtext terminal which was supplied especially to terminal manufacturers but also to many other interested parties. The attitude of the Administration was to ensure that every Bildschirmtest terminal should comply with the recommendations laid down there in order to maintain the multiplicity of the CEPT-Standard.

3. THE NEW NATIONWIDE BILDSCHIRMTEXT NETWORK IN THE FEDERAL REPUBLIC OF GERMANY

3.1 Invitation for tenders

After the positive decision of the Federal Cabinet, the Deutsche Bundespost laid down functional specifications for the nationwide Bildschirmtext network and invited manufacturers for tender. These specifications gave enough freedom to propose not only hardware and software but also concepts for this network. After evaluation of the submitted tenders, the Deutsche Bundespost concluded a contract with IBM for the development and supply of the first Bildschirmtext centres.

3.2 Structure of the Bildschirmtext Network

Figure 1 shows the structure of the Bildschirmtext network which at the moment has a two level hierarchy.
At the top we find a main data base and control centre the core of which are two IBM computers 3083 B 16 with two times 16 MByte store capacity and the possibility to connect two times 16 channels. The tasks of this centre are

- to store all the frames in a legally binding form

- to store the information of the search tree

- to store the data concerning subscribers and information providers

- to store and to process the data concerning statistics and system operation and

- to centrally store and process the data concerning charges, so that the bills for the subscribers and the remunerations for the information providers can be produced.

Each information provider may claim a remuneration from the subscriber who calls up data supplied by him. The charges payable for the Bildschirmtext frames will be recorded by the Bildschirmtext system, collected from the subscriber by the Deutsche Bundespost through the telecommunications invoice and credited to the information provider's account against payment of a fee (see 3.3).

At the second level we find Btx-Centres of the A-type which are directly connected to the main data base and control centre via an infra network consisting of fixed data channels. The data flow in the infra network is controlled both by network control computers and by gateway computers. Besides this network an additional computer for network management functions is in operation.

Up to six line computers each with a maximum of 96 subscriber ports may be connected to a data base controller. Up to two of these line computers can be installed at remote locations, thus forming Bildschirmtext Centres of the type B.

In the data base of a line computer where some 60 000 frames are stored, the subscriber will find the desired Bildschirmtext pages in as many as 90 per cent of all cases. If the requested information cannot be obtained there, it will be searched for automatically and without perceptible delay in the data base computer of the same Bildschirmtextcentre or called up from the main data base.

At the right side of fig. 1 we see an external computer which is connected with the Bildschirmtext network via the packet switched data network.

The test and development system at the left side of fig. 1 is neccessary for various reasons: to test new software releases, new facilities and also to train the staff responsible for the operation of the system.

3.3 Costs and charges

It was the intention of the Deutsche Bundespost that Bildschirmtext should be a mass service as soon as possible. Only this could guarantee that manufacturers would offer terminals to a price which could afford a broad public and stimulate the information providers to offer sufficiently interesting programmes. At present, i.e. in the introductory phase of the service, the price for a built - in CEPT - standardized decoder still amounts to about DM 600. However, w expect that the costs will decrease to some DM 300 - 400.

For the above mentioned reasons the charges for the subscribers were kept as low as possible. So charges in the amount of DM 8 per month are levied for the use of the Bildschirmtext connecting unit provided by the Deutsche Bundespost. The transmission of a message from one subscriber to another will amount to DM 0.40. The charges for information providers ar listed below:

Countrywide entry page	DM 350,--/Month
Regional entry page, 1. region	DM 50,--/Month
Further regional entry pages	DM 15,--/Month
Storage of Btx frame countrywide	DM 0,075/Day
Connection of external computer	DM 250,--/Month
Transmission of Btx frame to EC	DM 0,01/Frame
Transmission of response frame	DM 0,30/Frame
Input of frames online	DM 0,10/Frame
Input of frames delayed	DM 0,05/Frame
Processing of IP's remuneration	DM 20,--/Month
Surchage	2 % of remuneration

4. NATIONWIDE BILDSCHIRMTEXT SERVICE

4.1 Introductory Phase until the End

The introduction of the new service was pushed by the Deutsche Bundespost with great efforts and large investments.

At the end of 1984

12 Btx centres of type A
9 Btx centres of type B were

installed (see fig. 2) with a total of 66 line computers or about 6600 ports. Some areas are served by ports connected via multiplex links with line computers of distant type A centres.

The Bildschirmtext subscribers gain access to the ports via special telephone exchanges for information and dialogue services (INDI-exchanges).

More than 400 INDI-exchanges made it possible that already at the end of 1984 95 % of the telephone subscribers (and potential Bildschirmtext subscribers) were able to reach their Bildschirmtext centres at the extended local call charge rate.

4.2 Expansion Phase from 1985 onwards

The Deutsche Bundespost's planning activities are based on the assumption that the number of Bildschirmtext subscribers will increase until the end of 1987 to 1 million. For this purpose 53 Btx-exchanges, will be installed at the end of 1986.

Consequently

- new Bildschirmtext Centres only of the A type will have to be installed
- the existing Bildschirmtest Centres of the type B will change their sizes to the A type
- At the end of 1985 only A-type-exchanges will be in service. On economically reasons the Deutsche Bundespost will not install further B-type-exchanges. Access will be made only via INDI-exchanges and data multiplex equipment.
- new INDI-exchanges will be neccesary. Those 1500 local areas which serve more than 3000 telephone subscribers will finally be equipped with their own INDI-exchange. The access number is always 190.
- smaller local areas will be connected via lines of the regional long distance network using the access number 01910.

Up to 1986, the Deutsche Bundespost will invest approximately 500 Million German Marks in the Bildschirmtext technology and the associated adaption of the telephone network.

5. INTERNATIONAL BTX-CONNECTIONS

Unfortunately the Videotex standards in Europe are different. In France we have the Antiope system with Minitel terminals. In Great Britain up to now PRESTEL is in service. The Deutsche Bundespost introduced the full CEPT standard with smooth and line graphics and the possibility for DRCS (direct redefinable character sets).

Therefore we have to install an international gateway-system with different tasks. The general structure shows figure 3. All transitions between different videotex interactive networks and eventually other telematic services will be developed and installed in steps. The main modules of this gateway computer is to be seen from figure 4.

The first realization is the connection of Btx and Teletel (fig. 5).

Both-way traffic allows

- access for French Teletel (Ttl) users to Bildschirmtext (Btx) and
- access for German Btx-users to Ttl.

The DBP gateway works

- in the direction Btx-Ttl as an external computer in Btx and as PAV (point d'acces videotex) in Ttl.

- In the direction Ttl-Btx as a PAD (packet assembler/disassembler). The status of this PAD within the Btx-system is: no freedom of movement, closed user group (CUG) for international access). This access is similar to a normal subscriber access. Only on this way it is possible to get all informations from the data bases in
  - Btx exchanges
  - LZ (central Btx control) and
  - EC (external computers)

The gateways in the first phase have to support at least 10 connections at the same time.

6. CONCLUSIONS

There will be a great variety of users of the new Bildschirmtext service.

- normal households
- small and medium-sized enterprises that will no longer have disadvantages due to their location in comparison with large firms in densely populated regions.

But there are also indications that

- large companies will make extensive use of the Bildschirmtext service.

The Deutsche Bundespost has provided a solid basis for the service by

- making the access easy and economical
- providing it quickly to most of the potential Bildschirmtext subscribers and by
- defining reasonably low charges for the various facilities.

Last but not least, the introduction of this service offers the manufacturers the chance to develop, produce and sell equipment that is necessary to meet the demands of subscribers and information providers.

STRUCTURE OF THE BTX NETWORK

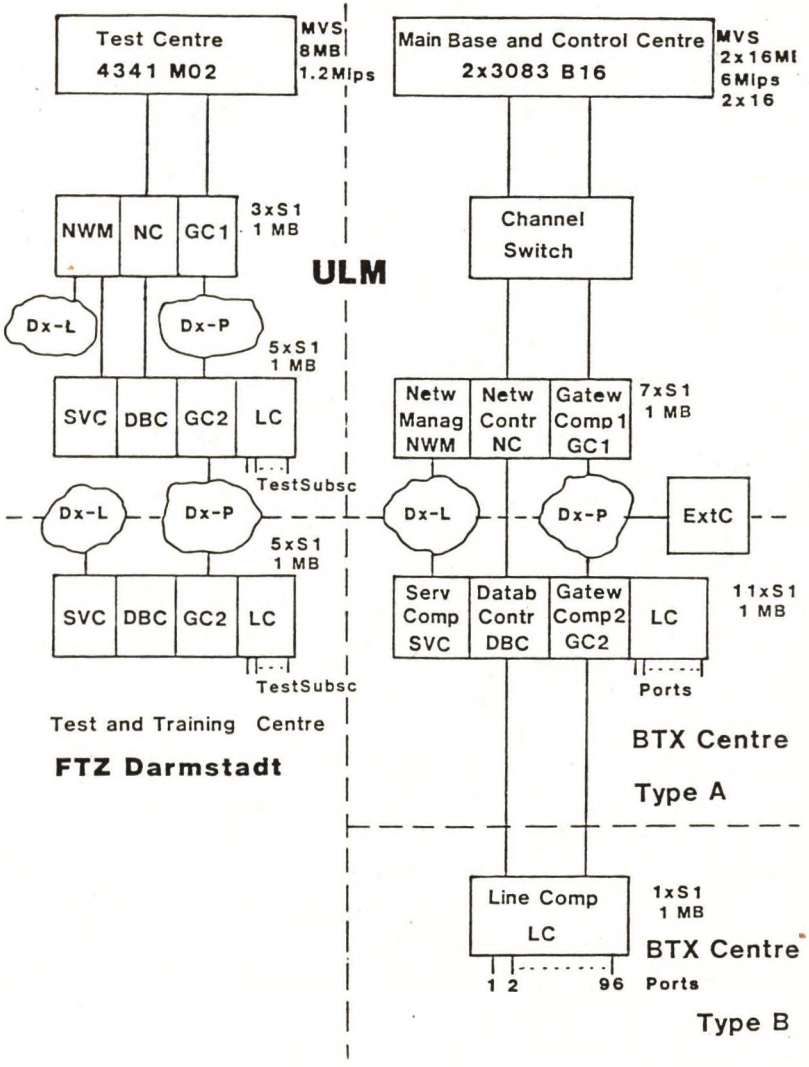


Fig. 1 Structure of the Btx Network

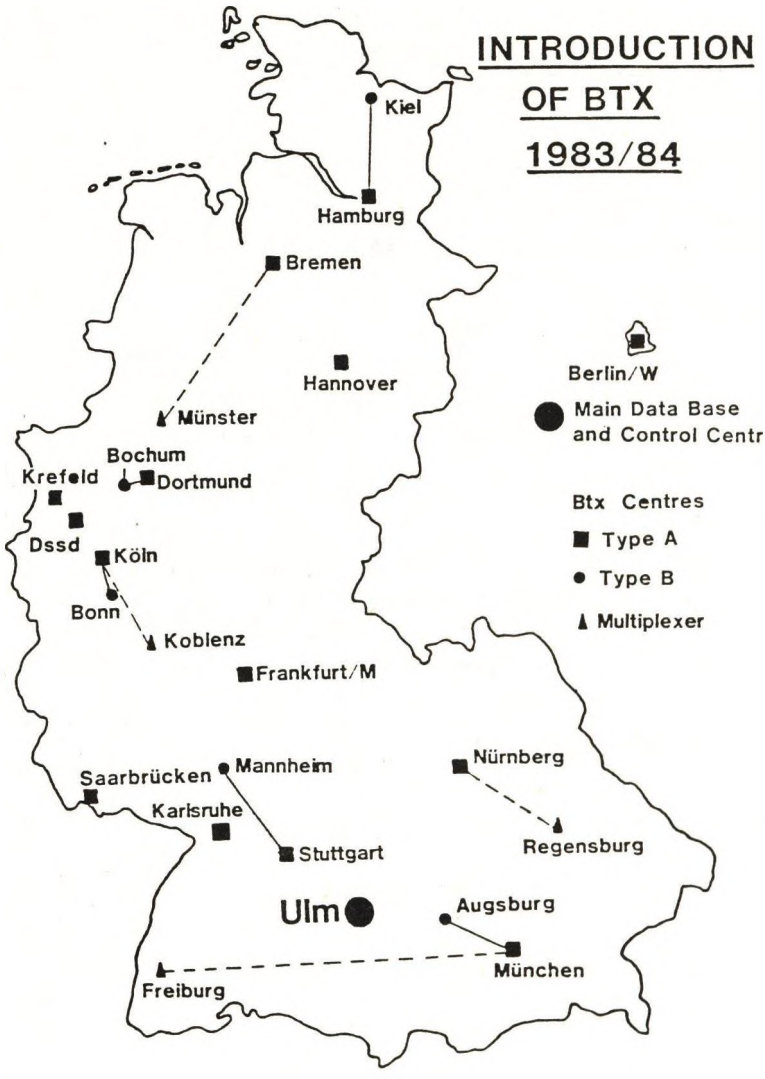
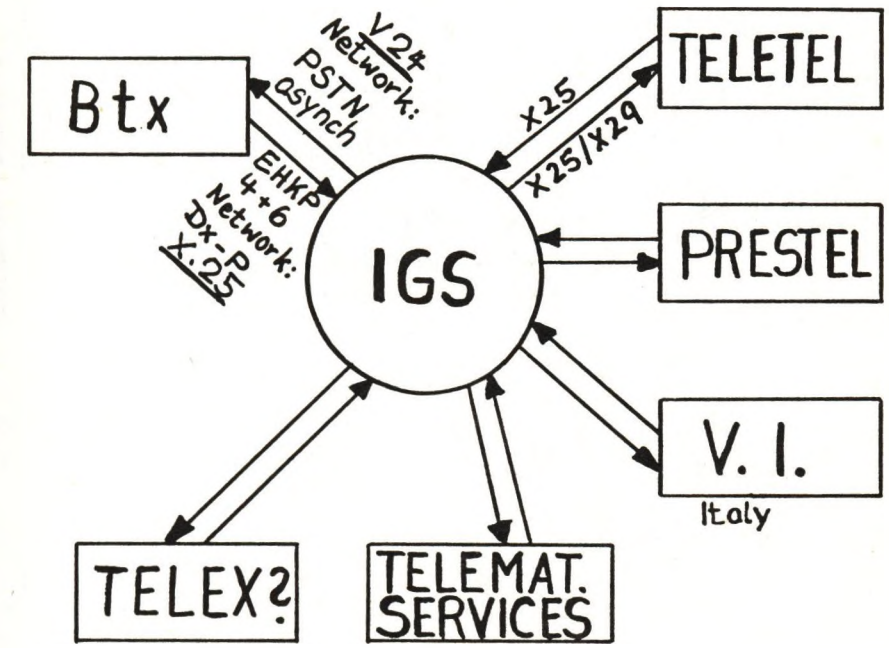


Fig. 2 Introduction of Btx 1983/84



→ Arrows indicate direction of user request

PSTN : Telephone Network

Dx-P : Packet Switching System

Fig. 3 General structure of IGS (international gateway system)

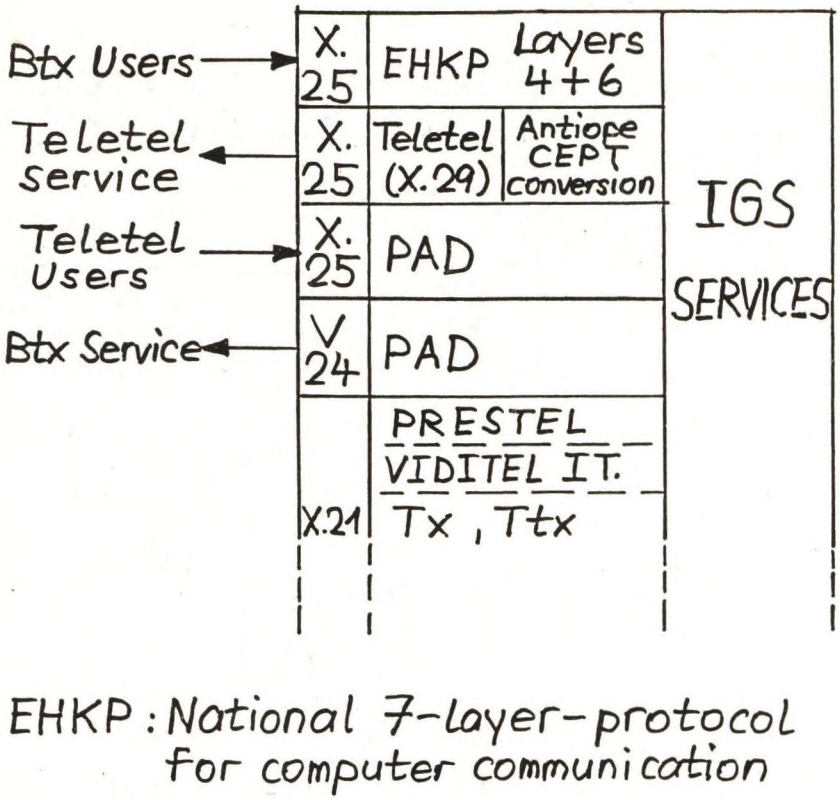
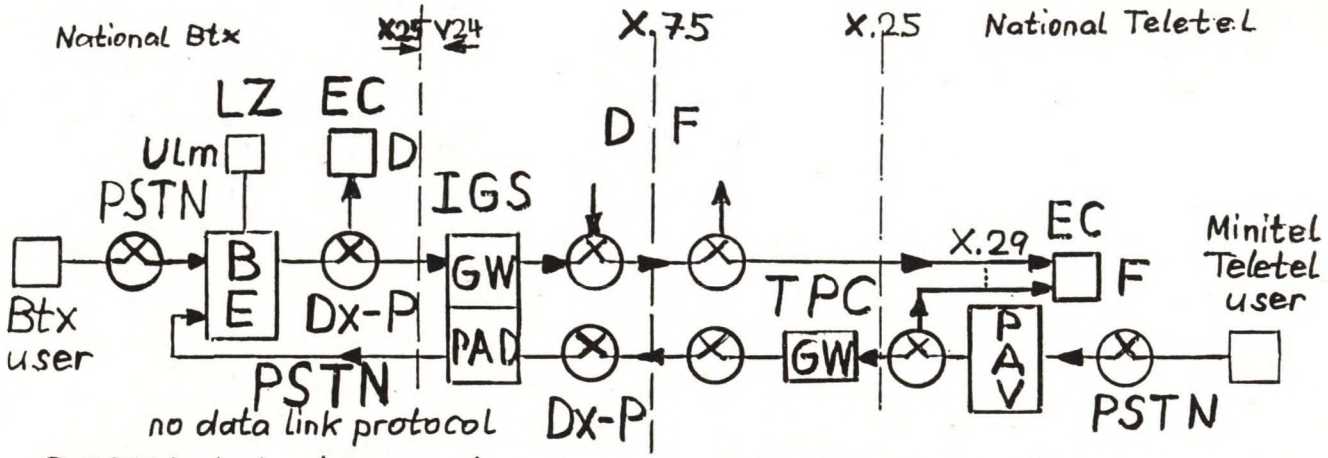


Fig. 4 Modular structure of IGS



PSTN: telephone network
Dx-P: packet switching system
TPC: Transpac system

LZ: Central. Btx-Database

BE: Btx-Exchange
IGS: Internat. Gateway-Sy
GW: Gateway
PAD: Packet-Ass.-Disass.
PAV: Point d'Accès Vidéotex
EC: External Computer

Fig. 5 Interworking Btx - Ttl

THE USE OF THE MAGAZINE-METHOD IN INTERACTIVE VIDEOTEX THROUGH INTELLIGENT VIDEOTEX DECODERS

István Sebestyén

The Institutes for Information Processing
Technical University of Graz and
Austrian Computer Society
Graz, Austria

This paper is based on development work at the Technical University of Graz implemented for CEPT videotex environment. The work was supported by the Austrian Federal Ministry for Science and Research. A so called magazine-method is used to download a prepared magazine of some ten to one hundred single information frames from a central videotex system into the MUPID II intelligent videotex decoder for subsequent off-line information retrieval. The method leads to substantial savings in telecommunication costs, improves the speed data transmission over the videotex network and potentially enables the use of more sophisticated data base search methods than used in classical videotex systems.

INTRODUCTION

Videotex systems have been around for a number of years. Yet noone can claim that public videotex systems have achieved the "breakthrough" experts were dreaming of in the early eighties when videotex started its penetration in the business- and the home market. Why, what happened?

The idea of videotex was born - as is well known - in the early seventies in Britain. At that time computer networking was in its early infancy, microcomputing was simply non existent, computers and computing were expensive and it was the privilege of only a few. The utmost state of technology was at that time "time-sharing", where computer users accessed large central mainframes with their video and teletype terminals. Time-sharing allowed expensive common resources to be shared - such as computer time and data storage capacity - while giving the individual users the impression that they were using the central mainframe exclusively all the time. This was a "programmer's dream", which in the late seventies became reality with the introduction of microcomputers - also called personal computers. The emergence of personal computers also influenced the development of new generation videotex systems where the terminals were not mere time-sharing terminals anymore but fully-fledged personal computers that could also be used as videotex terminals - leading to substantial improvements in videotex systems. A major development making this new generation of videotex rather different is that these videotex terminals with built in intelligence are able to take over or upgrade many - up to now central - videotex system functions, such as enhanced searching or local storage of videotex information frames. In addition, they are able to bring in new concepts and services such as the heavy use of telesoftware /1/.

It should be remembered that with the introduction of time-sharing and computer networks practically all - obviously "unintelligent" - applications now known

from videotex could principally already be supported. There were, however, some differences that distinguished them from videotex systems , such as a lack of higher level presentation protocol standards like PRESTEL or CEPT, or simply differences in the number of users (far fewer) and the types of service.

The principal original idea of videotex was to build a distributed information and transaction data network by using two widespread familiar devices: the home television set and its remote controller as computer terminal, and the telephone network as data carrier. Through this approach it was hoped to create a low cost time-sharing information and transaction network, easy enough to be used by thousands of "ordinary", non-trained users. It was hoped that by downgrading some hardware and software functions of a time-sharing data network to its absolute essentials a new mass media would be created, which was expected to achieve its breakthrough and maximum market penetration during the eighties. Up to now this battle is neither won nor lost and it is still not easy to predict its outcome. What happened?

First of all it turned out that the simplified time-sharing network and service - videotex - not fulfill all expectations could from the functional point of view. The terminal - a color TV set with a remote controller keypad and a built in videotex decoder for datacommunication and character representation - turned out to be far too rudimentary. Numerical keypads, suitable for looking up information frames in the simple numerical tree-like database of videotex, were just not enough for entering alphanumerical text such as for more complex search keywords or textual electronic messages. Actually a major problem of videotex turned out to be that, although information frames could be easily addressed and retrieved, due to the simplified rigid tree-like information management system the right information could often simply not be found. As mentioned above intelligent videotex decoders are able to improve the data retrieval capability of the terminal considerably in a number of ways. This paper describes one of them, the so called magazine-method, and presents some experience with practical implementations gained on CEPT videotex systems.

THE MAGAZINE-METHOD

The principle of the magazine-method in videotex is not new /2/. It is a way of distributing information nicely packaged by an information provider for both one- and two-way videotex systems. According to /2/ the two major categories of videotex systems are simmilar in that in both systems centrally stored information is delivered to a large audience of videotex users. In one-way systems this is done so that the centrally stored full information content is cyclically fed into the datacommunication channels - be it cable, fiberoptics or broadcast transmission - and the videotex terminal picks out the required frame information from the continuous stream of data. The flow of data is one-way from the videotex center to the users. The selection of information is local without any feedback to the center. In contrast, the selection of data in two-way videotex is done by the user who asks the videotex center to address and deliver the searched information. In terms of real implementation, information retrieval applications may in principle be built on both principles. In cases, however, where real interaction between videotex center and users or among videotex users is required two-way videotex is the only solution. For example, message sending of banking transactions can only be done on two-way videotex systems. The mere distribution of information, however, can principally be done in both ways, depending on factors such as given design possibilities and requirements, implementation time or costs. In practical realizations, interactive, public telephone based videotex services, such as PRESTEL in Britain, TELETEL in France,

BILDSCHIRMTEXT in Germany or Austria, are two-way videotex systems. Broadcast TELETEXT - utilizing the so called vertical blanking intervals of TV-broadcasts for data transmission - are typical one-way videotex systems. Information in most teletext systems is also grouped in magazines of about hundred information frames each - up to eight magazines - but this type of magazine-method is only remotely related to the one presented here.

The main principle of the magazine-method used here is that information magazines stored in one- or two-way videotex systems may be downloaded, frame after frame, into the intelligent videotex decoder - a dedicated personal computer - and be locally stored in RAM (or perhaps on diskette, if avalible) for subsequent "off-line" information retrieval by the intelligent videotex decoder (Figure 1 and 2).

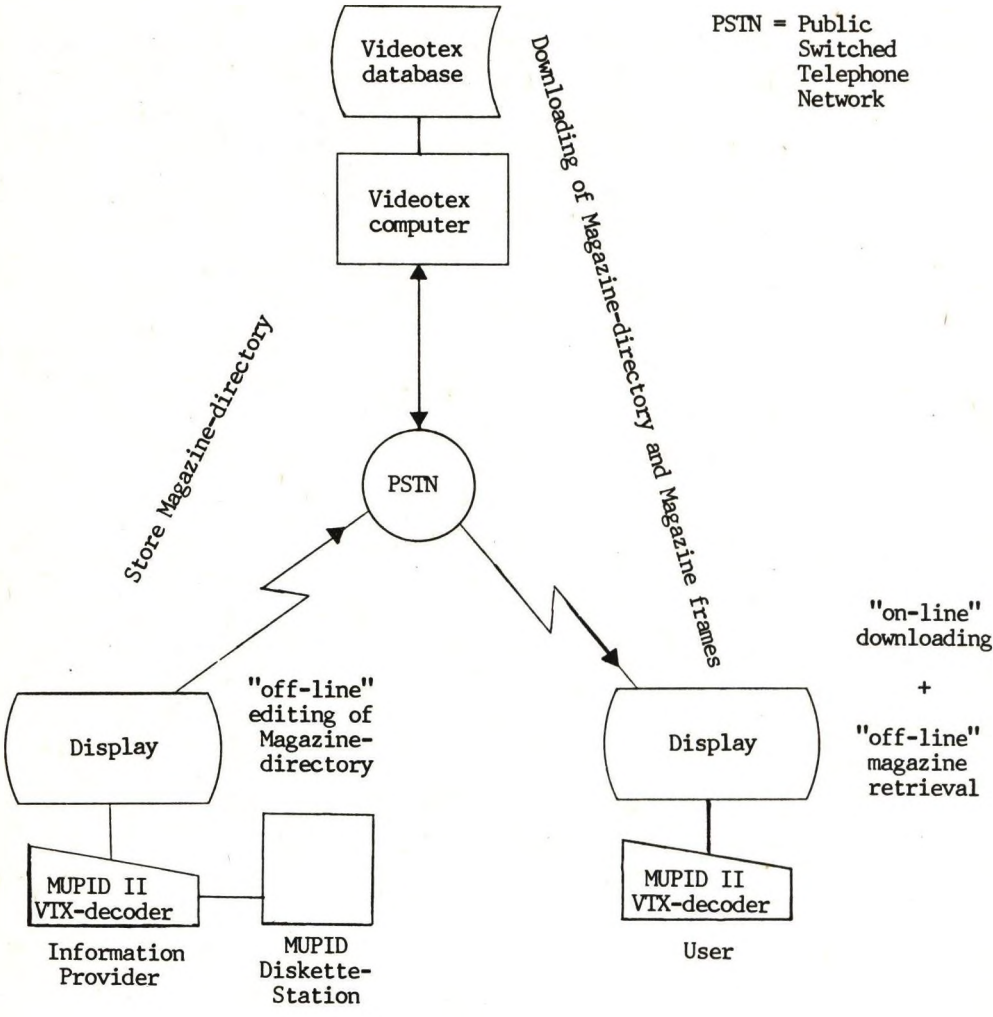


Figure 1
Principle of the magazine-method for two-way videotex

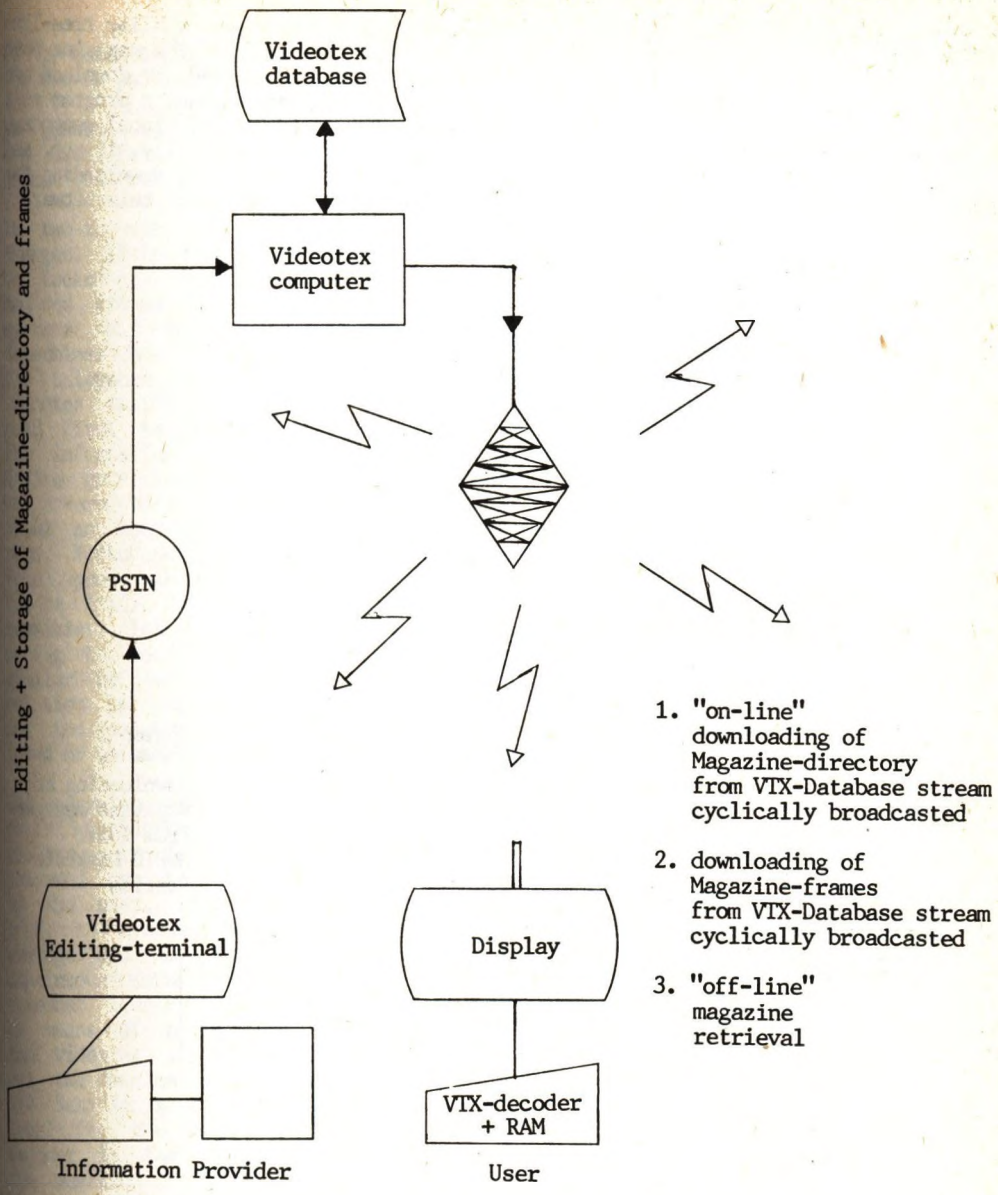


Figure 2
Principle of the magazine-method for one-way videotex

The magazine-method can be applied for both one-way and two-way systems. First, however, the information magazine has to be edited by an information provider such as a local newspaper. The information provider must do two things. First he must put together individually all the information frames that will make up the information magazine. Second the individual information frames have to be organized in a proper structure for the subsequent off-line information

retrieval. A suitable form of organization is to use a similar tree-like structure as in a classical videotex network. When downloading the magazine both information frames and their linkage information will be stored. To download and retrieve such a magazine, however, an appropriate magazine-manager program must first be loaded into the intelligent videotex decoder. If local secondary storage, such as a diskette, is provided this can be done locally, if not, and this should be the most frequent case it can be done by downloading the magazine-manager from the central videotex database as a piece of telesoftware (Figure 3).

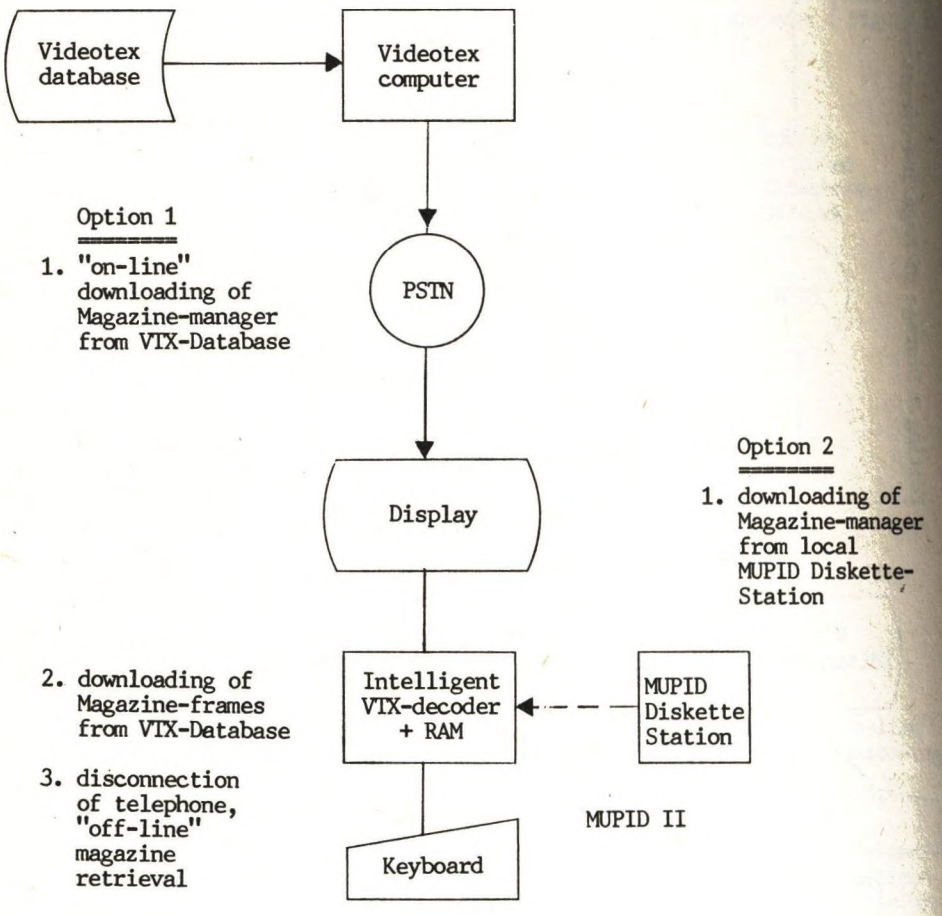


Figure 3
Loading the magazine-manager into users' terminal

The magazine-manager has two functions: first to download the information magazine and then to support information retrieval after the information is loaded.

Why use the magazine-method for information retrieval? Are the traditional searching methods for classical two-way videotex not sufficient?

It is quite clear that the magazine-method has it special application areas where its use is particularly suitable.

In general such a usage can be recommended if the package has to be "read" by the user from "cover to cover" such as is often the case with a newspaper or an educational course.

In two-way videotex a substantial part of the usage costs are linked to telephone charges. If in a videotex session a rather well-defined set of frames is going to be looked at - such as the electronic issue of a local newspaper where sixty to a hundred frames of information might be viewed - then in classical videotex systems this would be done in a fully interactive on-line mode. The user determines how long he would be viewing the individual frames while keeping up the telephone connection. From the cost point of view if one takes a classical videotex tariff scheme - such as that of Germany or Austria - a royalty has to be paid first to the information provider, here the local newspaper, for putting up the information and storing it on the central videotex database. Second the Post Office gets its revenue from the user on the basis of the telephone charges. If we assume that the average viewing time of a local newspaper on videotex takes about an hour then the costs of using the telephone for about an hour have to be paid. Fortunately - at least in Austria - all telephone charges for videotex are local phone charges, thus "only" 35 Austrian Schillings (some US $ 1.75) have to be paid. On top of this ,remember, the publisher earns its money from the royalties, let's say 5 Schillings for every "copy" of the electronic newspaper, making the total costs for the user about 40 Schillings. Assuming that the rather similar printed version of the paper also costs 5 Schillings, then the cost relation between the electronic and the printed version would be 8 to 1. This is not too encouraging considering it is the same type of information, although the speed of information access and its convenience is certainly worth something.

The magazine-method saves money for the user by helping to reduce the telephone costs considerably. Connection to the central videotex database has to be kept on-line only for the time required to download the magazine. Thus all the time consumed can be fully devoted to telecommuncation, and not to the "reading time" by the user. With the classical datacommunication speeds for videotex, 1200/75 baud, an average magazine of some 60 pages (with about 500 Character/page on average and 10 bits/Character) can be downloaded in about 250 seconds if continuous data transmission is assumed. In the practice this is, however, somewhat longer since each frame is accessed by the intelligent videotex decoder by means of a teleprogram separately. Furthermore, the search of the frames by the videotex center is also performed on a frame by frame basis. However, all in all the downloading process lasts on average some 5-7 minutes, equivalent to only 3-4 Schillings of telecommunication costs, which seems to be an acceptable cost level for the more convenient service. An advantage of using an intelligent decoder is that the magazine-manager can be programmed in such a way that while the magazine is being loaded into the decoder - in a sort of "background" mode - the user can already look up - in the "foreground" those frames that have already been loaded into the decoder. Since the information frames are linked through a tree-like structure the frame downloaded first are those that the user would logically read first. In such an arrangement, information retrieval can start immediately after the first frames of the magazine have arrived in the decoder. After the loading of the magazine is completed, the telephone connection can immediately automatically be switched off and all magazine frames can be searched in an "off-line" mode. Since the search for data is now done independently of the central videotex database the way of searching can be either the same as used with the central system - built on a tree-like frame structure - or even more complex, using more sophisticated data base searching techniques, such as in a

relational database. The former has the advantage that the user can use the same searching commands for both "off-line" and "on-line" searches. The latter has the advantage that more convenient and complex data search techniques can be applied in a videotex environment - a piece of information can more easily be found. In our practical implementation of the magazine-method the first approach has been adopted, simply bacause it was easier to implement.

Another advantage is that the presentation times for individual information frames on the screen can be speeded up. It is well known that in videotex systems information can only be shown on the screen at the same speed that data are downloaded from the central system, encoded and presented to the display unit by the decoder. In case of more complicated graphics, such as DRCS (Dynamically Redefineable Character Sets), or facsimile graphics with large storage needs - thus long transmission period - the search and presentation time of such a frame can take half a minute or even longer. With the magazine-method all information is retrieved from the fast local RAM storage of the videotex decoder at higher speeds - thus less annoying waiting times!

The magazine-method also has some disadvantages. When using intelligent decoders, the use of telesoftware opens up some particularly useful and unique applications /1/. An important aspect of the MUPID videotex decoder used is the so called "hidden" telesoftware. "Hidden" telesoftware is stored on ordinary videotex frames on the central videotex database like any videotex information frame. The only difference is that when retrieving such a page the intelligent videotex decoder automatically recognizes that the content of the frame contains an executable teleprogram and instead of decoding and showing its content on the terminal screen it loads its code directly into the memory of the decoder for immediate or subsequent program execution. The only effect of this automatic mechanism for the user is that he realizes that the performance of his system is suddenly considerably increased. When applying the magazine-method such "hidden" telesoftware can as a rule not be part of the information magazine, however, since their loading would destroy the magazine-manager program. The magazine-manager itself, however, can still be stored on videotex as "hidden" telesoftware. In the long run when the storage capacity of the videotex decoders increases further, it will be possible to load the magazine-manager into a separate memory bank, and the teleprogram frames could then be also loaded into the decoder to other memory banks. Obviously appropriate interfaces between the magazine-manager and the teleprograms will have to be made. At this point such systems are not yet implemented, as a general rule no other teleprograms can be combined with the magazine-manager implemented by us.

Another aspect of the present magazine-manager system is that with regard to data transmission security it assumes that the present CEPT systems, such as the German or Swiss take care of all data transmission errors. In principle this should be the case, in practice, however, from time to time one or two information frames are not correctly downloaded into the decoder. In such cases, at "off-line" retrieval from the magazine those information frames could be presented with their data transmission error. It can be expected, however, that such errors will not occur when CEPT videotex services become more mature.

Another disadvantage (or for some information providers even an advantage) is that by using the magazine manager the user is "forced" to consume the very same information package that the information provider designed for him. In classical videotex systems all information is offered under a type of "umbrella" by the PTT, whereby the user could easily switch from a piece of information provided by one information provider to information provided by others. Incidentally, this effect is considered by many to be one of the advantages of videotex, because it brings more selection freedom for the users. However, selection freedom often

means that the searched information not can always be easily found. Very often it is not in the interests of an information provider that a reader of his information package switches over to a "competing" information provider. With the "off-line" magazine method it can be ensured that the user first goes through the information content of the magazine and only if some additional information - perhaps from others - is required will renewed access to the videotex network be made.

An interesting new aspect of the magazine-method is pricing. In classical videotex systems every information frame can be separately priced. Prices are determined on the "value" of the information provided, whatever "value of information" means, or simply on the basis of how the "production costs" of such information are expected to generate their return. Thus an information frame with valuable marketing forecast information may cost a lot, while general or advertising type of information may come free of charge to the user. In practice it has turned out that if an information frame costs more than a Schilling the user is already selective in accepting it.

A newspaper or a monthly magazine - such as BYTE - not only provides both of the above types of information; they are in fact linked - due to the nature of newspaper - into a sort of common "magazine". Ofcourse a single price has to be paid for such a "magazine", which usually varies for a newspaper between 5 and 10 Schillings; for a weekly, monthly magazine between 20 and 100 Schillings. It is estimated that 60% of the income for a typical newspaper comes from advertisements paid by the advertisers, who are a type of "information provider". In addition to this the users pay only about 40% for both types of information. If the magazine-method is applied in videotex systems, the same, well known, used and accepted pricing structure can be applied. Up to now not much experience has been gathered about how much a user is willing to pay for downloading a single magazine. I would guess that the price of a daily newspaper would on average still be acceptable to the users. In order to keep the magazine costs down - where appropriate - advertising frames could also be included as for newspapers.

For classical videotex systems to apply the same philosophy - in my view - is more problematic. The combination of "primary" information - the actual information the videotex user searched for - and "secondary" information - a few lines of advertising is well known and actually common practice with some information providers. However, it is not so certain that the income from the advertisements is being deducted from the frame charges. Second, little is known about how users react if a part of the relatively small information window (in videotex usually 24 lines with each 40 characters) is "wasted" for ads. The case gets even more problematic if the advertisement uses more complicated graphics that not only occupy parts of the screen, but take time and leads to delays and increased telecommunication costs when they are downloaded and presented. If such pages are retrieved more than once during the search process, then the situation might be even more annoying for the user.

This should not be misunderstood: I am not against putting ads on videotex frames. My message is that the magazine-method is on the whole better suited to absorb the problems advertising presents for videotex and to utilize its advantages, i.e. save costs for the user and provide them with actual advertisement information in a well-structured way. How to do this, however is not the subject of this paper.

IMPLEMENTATION

The magazine-method has been implemented for the MUPID II intelligent videotex decoder /3/ for the CEPT environment.

MUPID II has an extensive BASIC including sophisticated graphic capabilities to generate text and picture according to the CEPT C0-C2 levels. Since the "hidden" nature of telesoftware is one of the specifics of MUPID, and furthermore MUPID is a dedicated personal computer directed towards the videotex environment, special BASIC commands take care of program interaction between MUPID and the central videotex network.

Most of the coding for the magazine-manager programs was implemented in MUPID Basic. Only some system-specific parts - such as storage from and to the different RAM banks - were written in Z80-Assembler. MUPID II has an extremely strong and suitable BASIC and an appropriate development system /4/, using for some extremely useful testing and tracing purposes the so called Bank 3. - part of MUPID's RAM for programming - of the decoder. This made the program development work rather difficult because for storage economy purposes all magazine-managers were designed in such a way, that Bank 2. of MUPID contained the executable BASIC and Z-80 ASSEMBLER codes, where as Bank 3. included primarily the buffer for the magazine. Thus the testing aids of the BASIC development systems could not be utilized. It should be noted here that the memory of MUPID II is organized in memory banks of 32 KByte storage capacity each. Access to the different memory banks is done by switching through software. In the basic configuration of MUPID out of its four memory banks only Bank 2. and Bank 3. can be used for telesoftware and magazine-data. Most magazine-manager program versions only support the access to these two banks. In the present, let us say standard, version a maximum of some 57 KByte storage capacity is reserved in MUPID to download the magazine, Bank 3. stores the first 32 KByte of it and Bank 2. the rest in its remaining space after loading the magazine-manager program and building up some necessary directory tables for the data base management system. With extremely tight programming it was possible to save some 25 KByte capacity in Bank 2. for storing.

Depending on the size of the individual frames to be stored, one can on average store some 60 to 100 frames. If, however, the magazine only contains complex DRCS frames (some require about 5 KByte storage) the storage capability can go down considerably. But this is not the standard state of operation.

It is perhaps interesting to mention that we have also implemented for testing and demonstration purposes a "JUMBO" version of the magazine-manager by upgrading the storage capacity of MUPID II to 512 KByte by using 256 Kbit RAM chips. With such an arrangement we were able to extend to the memory banks of MUPID up to sixteen. The manager program was organized in such a way that when downloading Bank 3. was filled up first, then Bank 4. and so on until Bank 15. Finally even the remaining part of Bank 2. could be utilized. The practical use of this experiment was obviously rather limited since the standard MUPID II will go with only 128 KByte, with Bank 2. and 3. Apart from test- and demonstration purposes (where we were able to download all CEPT demo-frames we had on diskettes - taking more than half an hour and we were not able to fill up the memory beyond Bank 6.) it became quite clear that such high magazine-storage capacity is only meaningful if the delivery channels are fast enough. In our case we used a local MUPID diskette station with 4800 b/sec speed for downloading the demonstration frames. Translating this for normal videotex systems, means that with a speed of 1200 b/sec the same amount of information (which only filled up the magazine to some 25% of its full capacity) would require a downloading time of some 2 hours and perhaps some 8 hours for the total. For the present two-way videotex systems

this is certainly not a viable alternative. Two-way videotex magazines of that size will only become sensible in the future ISDN era at much higher data transmission speeds. For one-way videotex over cable networks or through TV broadcasts, data can be circulated with a speed of some 5 Mbit/sec per transmission channel. With such a speed the "JUMBO" magazine of a MUPID II-like decoder of some 500 KByte could be filled with the processing speed of the decoder processor. It would certainly be below one minute. These figures should illustrate how suitable the magazine-method for one-way high speed media is.

During the development work, a number of versions of the magazine-manager were developed. One "manual" version allowed the user to enter his individual "user magazine" in interactive mode for subsequent downloading and "off-line" retrieval. Another "manual" version was implemented for both "off-line" retrieval and automatic rotating of frames mainly for demonstration purposes, for example for use in a supermarket chain where the information magazine on "special offers of the day" could be rotated throughout the day.

More significant are in my view however those versions that were primarily described in this paper, i.e. an information magazine edited by an information provider for subsequent downloading and "off-line" viewing by a user. Two major programs had to be written, one the above described magazine-managers for the users and a magazine-editor for Information Providers.

The magazine-editor - written also in MUPID-BASIC and Z-80 ASSEMBLER - allows one to enter or to modify a magazine directory. Such a directory contains all the names or addresses of the individual videotex frames to be downloaded and the way these frames are linked together. Finally after the editing session the magazine-directory is stored on a specific videotex information frame. When downloading the magazine the directory will be retrieved first by the magazine-manager, then all loading information, such as frame number and their linkages will be processed and finally according to the instructions on the magazine directory the individual frames downloaded, one after the other in prescribed order to allow the "foreground" retrieval of the already downloaded frames.

In our versions of the magazine-manager up to 100 frames of information could be stored in a magazine, whereby each information frame could be branched up to ten other magazine frames.

The magazine-editor allows the entry, modification and deletion of any frame and any linkage thereof. At design, specific effort has been made to secure an extensive plausibility control of the tree-structure of the magazine. It was felt that the frame editors of classical videotex systems were particularly weak from this point of view. In addition, special effort was devoted to make its use as friendly as possible by means of built in help functions.

After loading the magazine-manager it asks for the address of the magazine-directory on the central videotex system. When the directory frame is loaded into the RAM memory of the videotex decoder the first two frames of the magazine get loaded. After that the "foreground" retrieval of the frames already loaded can start. To serve the "foreground" inquiries fast enough the user has the highest priority for the magazine-manager, and is thus is served first and immediately. Since MUPID II has only one microprocessor as CPU it can either store information from the input buffer of MUPID into the RAM or display a frame from the RAM onto the screen. Parallel processing of loading frames into the input buffer and displaying frame is however possible. For this reason with normal use and not too frequent display requests by the user the "foreground" retrieval and the "background" loading works in practice very satisfactorily. If the display requests were, however, too frequent the processor would have less

time to store frames into the RAM and the downloading phase would take longer.

On any frame retrieved in line 25 (one line below the last line with meaningful videotex information) the choices to the linking frames are given. If the branch information is however a part of the videotex frame itself, the display of the branching information in line 25 can be switched off by the Information Provider when editing the directory. In such a case the magazine editor puts a certain flag into the magazine directory, which will be taken into consideration by the magazine-manager.

In addition, on-line help aids and a directory of the magazine content support the users.

CONCLUSIONS

The magazine-managers implemented are brand-new developments, thus it is too early to report on concrete field experiences in usage.

It can however be hoped that in both one- and two-way videotex systems users will quickly find and implement those videotex applications that are suitable for the magazine-method. It should be stressed again that not all videotex applications are appropriate. In cases where real interaction between the central videotex system and the user is required it can not be applied. It is ideal for information retrieval if the combination of "background" downloading and "foreground" viewing turns out too be more economical and convenient than the "on-line" retrieval and viewing of the same.

Finally a last personal remark by the author. When testing the magazine-manager with the Swiss and German public videotex systems both systems were in my view annoyingly slow in serving users. This fact was practically "confessed" by both systems: on most frames the message "please wait" came first. One bottleneck of the present CEPT systems seems to be the data transmission infrastructure, the 1200/75 b/sec speed is for CEPT certainly too slow. Using the magazine-manager however this slow response time simply disappears. The magazine-manager takes over from the central mainframe all interaction with the user, and this at internal processor speed. The display of the downloaded CEPT frames is instantaneous without waiting for network response. The downloading time of course is still "CEPT-speed" but this is at least invisible for the user. The magazine-method thus also helps to speed up CEPT as it were PRESTEL.

Note:

This paper was written by the author when he worked as Visiting Professor at the Institute for Information Processing of the Technical University of Graz, Austria. The author is particularly grateful to Professor H. A. Maurer for his invaluable suggestions and comments when implementing the magazine-method presented.

REFERENCES

/1/ Maurer, H.A., R. Posch, MUPID: An Austrian contribution to the development of videotex, IIG F87 (1982), Graz, Austria

/2/ Maurer, H.A., I. Sebestyen, One-way versus two-way videotex, Electronic Publishing Review 4(1982) & 1(1983)

/3/ Fellner, W.D., H.A. Maurer, MUPID 2 - Überblick, IIG B47(1984), Graz, Austria

/4/ Milner H., J. Stögerer, CEPT MUPID BASIC Sprache und System, IIG B46(1984), Graz, Austria

ACCESSING A VIEWDATABASE

László Kovács

Research Institute of the Hungarian PTT

Budapest, Hungary

This paper focuses on some technical questions of access to a national public videotex service. An evaluation of some aspects is given. Future access scenario is described. Guidelines for approaching access problems are discussed. Arguments for the realization of multistandard vtx are underlined.

INTRODUCTION

This paper is based on the results of a long term teamwork, which supported the introduction of a trial vtx service in Hungary. A PTT team has evaluated many technical specifications issued by other administrations, and numerous system descriptions provided by manufacturers, suppliers and other partners, which were involved in the preparation activities of contracting of equipment and software for the trial service. During the preparation activity members of the team held frequent consultations with specialists of administrations and manufacturers. After a certain study period it became clear, that features and facilities of existing public and private vtx systems are too different, and there is no universal solution at all for the realization of a public interactive vtx system. Facilities to be provided are largely supplier-dependent. Features supporting vtx network functions cannot be yet realized on the basis of standardized OSI protocols, or protocols elaborated for other telematic services because generally they do not provide all the necessary options for supporting interactive vtx operation mode. So the team aimed first of all to find a good compromise between basic facilities and features, to be provided, and options which are desired or accepted. This paper is intended to describe some interesting topics associated with acces which were evaluated during the preparation of the chapter "Facilitie of technical specification for suppliers of the trial system.

FACILITIES OF A PUBLIC INTERACTIVE VIDEOTEX SERVICE

Facilities offered by a given vtx system are the most relevant part of the whole parameter set which defines the system from the user and provider point of view. One can say that sophistication of a vtx system depends largely on how many facilities it provides, how many functions it incorporates. Although it is basically correct to try to realize facilities in a public vtx system as many as possible, in some cases or in the case of some facilities certain limitations seem to be needed. On the other hand some facilities require more attention because of their importance in a public environment. The following description of some facilities associated with access possibilities

is focused on those important system functions which are not yet generally accepted by all service providers.

ACCESS TO THE SYSTEM

A public vtx system must support several types of access in order to ensure a wide range of access paths available for users throughout the world. Each access type requires an appropriate interworking protocol /or gateway protocol/ which must support economic data exchange between the System and the terminal located elswhere in the world.

Some of the possible interworking protocols may set specific limitations on some application /and-or presentation/ level functions. As an example the PAD /Packet Assembly-Disassembly/ protocols may be mentioned. Standard PADs on PSPDNs do not support local control of full-screen oriented applications, therefore the host computer /in our case the System, i.e. the public interactive vtx system/ must perform a tight "remote" control of e.g. cursor movements on the screen, causingin almost all applications long delays in echoing characters entered by the user, accompanied with considerable increase in data transfer volume-based charges /See also "PAD versus GATEWAY"/. For the application of an interactive editor this is quite unacceptable and probably the System must ensure that a user communicating with the System through a PAD will be limited in access to applications only suitable for handling through a PAD device.

Another example is a non-standard application running on the System. Such an application may be largely incompatible with existing gateway protocols /because of e.g. alphanumerical input in Line 24./ and therefore access to that particular application from gateway connections /i.e. from other public vtx systems, or from External Computers - ECs/ must also be barred. So the System must recognize the type of access for each user session and act accordingly by limiting the user in accessing certain applications or by providing versions of a given application compatible with interworking protocols involved in the given access type.

THE SET OF GATEWAY PROTOCOLS

An important aspect of access is the set of gateway protocols to be provided in a national vtx service. On one hand interworking with other public or private vtx systems will cause the need for the realization of at least three well known existing gateway protocols:

- the german EHKP
- the Prestel Gateway Protocol
- the french ARCHITEL

On the other hand there are special implementations of some of the protocols above mentioned. In Hungary for example exists a limited IBM-327x/BSC-based implementation of the Prestel GW, which runs on IBM 370/x-compatible host computers providing External Computer /EC/ function for private vtx systems. His counterpart, the functional equivalent of the protocol machine implemented normally in a public vtx system, was realized in a small microprocessor-based terminal concentrator. Participants of the trial service who already have these

implementations will require appropriate connections to the System in order to eliminate additional hardware/software developments. So the System must be able to handle among others this particular type of interworking in order to increase the acceptance of the public trial service.

It must also be mentioned that existing gateway protocols provide only one-sided /"unbalanced"/ communication in their basic form. This means that:
Generally users of the public service may retrieve frames from ECs and interact with applications running on an EC, but users connected to an EC can gain access only to frames and applicatons available on that particular EC. However there are suitable technical solutions eliminating this problem /e.g. the implementation of "balanced" versions of gateway protocols if both ends use the same protocol and the same presentation layer data syntax with similar man-machine interface/, and it seems to be rather a problem of regulation to provide access from local terminals /or even dial-up terminals/ of an EC to frames and applications of the public service. If that problem will be solved then both the provider of the public service and the group of users located at an EC involved in such an interworking will benefit because:

- Users wouldn't be forced to set up separate PSTN connections to the public system, and terminals could be cheaper by saving modem costs.

- There are regions from where no access to public vtx services via PSTN will be available. So in these regions at least users of some local EC will be able to communicate with the national public vtx service.

- The provider of the public service will save capital investments by the decrease in the number of terminal concentrators to be installed in the own vtx network. In addition savings in costs of network maintenance also are expected.

PAD VERSUS GATEWAY

The fourth basic interworking protocol will be undoubtedly the CCITT X.3/X.29 used throughout the world in PSPDN's PAD devices.
PAD interworking capability of the System may have enormous importance due to the worldwide existence of interconnected PSPDNs.

Recent public and private vtx systems are basically isolated. National public /or public trial/ services do not yet ensure international GW facilities available to all users. PADs are accessible throughout the world, and they provide a more economical access with higher quality to remotely located vtx /and other/ hosts. So the only recent alternative, the long distance phone call may be in this way eliminated. However in the future international gateway connections will become a reality. Some PTTs have already made enormous efforts for the realization of their first international gateway connections, showing that the lack of a common interworking protocol and the considerable differences between accepted /but incompatible/ presentation layer data syntaxes may cause serious problems in interconnecting vtx systems. These efforts will result in the /may be far?/ future in live public international gateway connections. At that time the importance of PAD access to vtx systems will decrease significantly because of some basic disadvantages against gateway protocols:

- PAD protocols do not provide data compression capabilities as e.g.

the german EHKP does /through proper handling of the so called "Presentation Image" technique and "Format Service" facility/.

- PAD devices do not provide error correcting data link protocols on the terminal side /strongly needed for higher level presentation layer data syntaxes/ therefore the host must take care of remote control of the data link layer through appropriate handling of PAD parameters and processing of received data packets which contain control characters of the data link protocol implemented in the terminal.
In this case however serious problems may arise for longer delays in data transfer through the PSPDN and some limited implementations of the data link protocol in the terminal /caused e.g. by limited buffers/.

- PAD protocols do not support full-screen oriented applications.
They can't be controlled by the PAD properly due to the limited parameter set defined for PAD facilities. So for these applications the host computer must perform functions normally implemented /in computer networks/ in terminal controllers or even in higher level terminals.
In these applications PSPDN characteristics cause often unacceptable effects /long echoing delays/ and considerable increase in data transmission costs.
CPU load on the host side also becomes higher due to more frequent packet processing in this mode of operation.

- Each user session being established between a single host computer and users connected to a single PAD device requires a separate SVC /Switched Virtual Circuit/. That results in higher costs on both the terminal and the host side:

  - on the terminal side time based charges will be higher,

  - on the host side monthly charges increase because the need for the rental of more logical channels, /gateway protocols normally support multiplexing of more user sessions on a single SVC/.

- PAD protocols do not handle administrative elements, as e.g. price request-acceptance in existing gateway protocols. Therefore users communicating with a national vtx system through a PAD from abroad must be registered on that system in order to get access to charged frames and facilities /See also "ANONYM USERS, UNIDENTIFIED USERS"/.

ACCESS FROM AN X.25 DTE

An X.25 DTE in the following considerations is basically a host computer or a terminal connected to a PSPDN directly, without any interconnected protocol converter.
An EC being an X.25 DTE and providing access to vtx services through the PSPDN is to be regarded as a special case of this access type:

It is a host computer that performs through appropriate vtx-specific higher level protocols functions of a terminal cluster controller and of a front-end processor /i.e.: higher and lower layer protocol conversions, control of keyboards and screen contents etc./. In addition handling of administrative elements also can be supported by that computer. Even anonymity of users to the remote System /on which runs an application controlling the given EC/ may be supported in this special case /See as above!/. But generally an X.25 DTE must not to be an EC known by the given vtx system /e.g. through analyse and validation checks on data of "Call Request" packets/ in every case. In other sence

an EC or other hostcomputer which supports "balanced gateway" mode of operation with one vtx system, and is known to him, also can support the same interworking with other vtx systems which are accessible through PSPDN and provide compatible protocol versions. In this case however anonymity may suffer if there is no charging agreement between the operators of the interworking vtx systems. That results practically in the need for subscribing each user who wishes full access to a given vtx system to that particular system. So that is a situation very far from todays practices with ECs.

Approaching the future scenario one can find an evidence that a very frequent event could be the use of an X.25 terminal which is capable of handling appropriate higher layer protocols. As a general purpose protocol, ensuring both access to most of /non-vtx/ host computers like databases, Message Handling Services etc., and some of existing vtx services like the british "Prestel", could be regarded the PAD CCITT X.3/X.29 protocol discussed earlier in this paper.
Sure, his only advantage is generality /and may be a relative simplicity of implementation?/. However one of his disadvantages, the lack of an error correcting data link protocol between the PAD device and the terminal, seems to be eliminated in the case of an X.25 DTE which has an integrated PAD function. For a stand-alone X.25/PAD terminal there is no need for a separate local data link protocol at all: both X.25/PAD and terminal controller protocol machines run on the same CPU. In this case layer II. of X.25 ensures error free data transmission between the terminal and the PSPDN.

For an X.25 host or for a LAN which provides an X.25 gateway, data link protocols on the terminal side are a local matter.

Other disadvantages of PAD interworking cannot be eliminated because of the lack of advanced /e.g. screen oriented/ versions being already standardized of PAD control functions in host computers. Surely, virtual terminal protocols will replace PAD interworking in the future, therefore there is no reason for making significant enhancements of PAD protocols.

In vtx accesses at the beginning existing gateway protocols will also have a great importance due to the facts that they exist, they are well specified and already severally implemented.
For vtx access from an X.25 DTE special implementations of these existing gateway protocols could become a reality /e.g. an "EHKP" without "Format Service"/. There is nothing to say against the realization of such interworking between even a stand-alone terminal and a public vtx system. Many of existing Personal Computers are not less powerful than gateway servers in existing vtx networks. Most of PCs provide enough main memory and processing power for the proper handling of X.25 and higher layer protocols in parallel. A serious bottleneck in serving a large number of such advanced terminals will be undoubtedly the gateway server of the given vtx system. Surely, both number and processing power of these network elements must be increased when advanced X.25 terminals begin to penetrate, when there is already a number of such terminals active on the given vtx system comparable with the number of ECs interworking with that particular vtx system.

An important aspect is that neither the PAD nor the gateway-type access does require additional resources on the lowest level of a vtx network, on the level which serves PSTN access directly. This level seems to be a very heavy /hardware/ part of a larger vtx system, in addition being to specific /i.e. in most of cases clearly vtx dedicated/ and expensive

both in upgrading and maintaining. Therefore savings on that level could be very useful from a service provider's point of view. Deep penetration of X.25 terminals will require however upgrading of existing PSPDNs and significant increase in throughput on international trunk lines.

Independent of vtx evolution, penetration of advanced X.25 terminals is to be regarded as an evidence, so the only way to follow this tendency in vtx is to provide X.25 based access with effective higher layer protocols for users of suitable advanced X.25 terminals, for groups of users on ECs, LANs and other private or corporate computer systems. In this way processing power of a vtx system as a whole will be largely redistributed: many functions /protocol conversions, screen and keyboard control etc./ recently performed by lower level elements of vtx systems will be moved to terminals and to computer systems serving own terminals, and that will result both in higher quality of service on the terminal side and decrease of CPU load on the network side.

ANONYM USERS, UNIDENTIFIED USERS

Both categories are to be associated with a special user status, which means that the identity of the given user is not known to the end-system serving him. However there is a slight difference between anonimity and unidentified status. The first mentioned is a very frequent event in accesses to applications running on ECs. The user whose terminal is connected to the public system may be anonym to an EC if he decides so /by rejecting collection of his personal data, available to the application through appropriate system values in so called "User Action" frames/. May be the applications activated on an EC during a session do not require collection of personal user data or other identification procedure /e.g. password input/ at all. In this case the given user is identified only by the public service, but he will be charged for frame accesses on ECs too. Gateway protocols provide technical basis for this type of charging, so there is normally no need in registering a user on all ECs. The same will be true for future international gateway connections because of very similar interworking protocol solutions.

The second status means that neither the user nor his terminal /like a coin operated one/ is identified. So there is no possibility to charge that user per time base or for frame accesses. However the possibility of this specific status may have a great importance first of all at the beginning of a trial service. In an initial phase there are not too many coin operated or credit-card terminals available. So future users can get mainly access to terminals located near their working place. And that will mean that this specific status enable users not yet registered on the System to interact with the System, to see what there is available. That may result in a higher acceptance of a trial service.

The first and second status each causes certain limitations in access of applications available on the System. Neither the anonym nor the unidentified user could gain access to closed user group informations, to personal services like "Mailbox", and specific system frames like statistics or password manipulation. In addition an unidentified user cannot retrieve charged frames. It seems to be logical also to limit session time of such a user. Or probably the maximum number of parallel unidentified sessions may be limited depending on the current traffic load of the System.

The question of response frames to be provided for unidentified users is not yet clear enough. E.g. in the austrian public vtx service unidentified users have no access to response frames at all. Surely, there is no danger of running out all the storage space /the "Message Pool"/ because of some tricky users with their PCs programmed for such activity and making this using the unidentified status. However it seems to be very useful to provide at least a separate smaller set of response frames for unidentified users /e.g. for messages to the service provider or to other public authorities/. And that must be realized with explicit limitations both on the number of messages sent per session, and the size of a specific pool, which contains all messages sent by unidentified users. So this category of users will be able to come in touch at least on an initial level with one of the most important facilities of a vtx system: Interpersonal Messageing!

Limitations caused by the unidentified status will be rather useful than disturbing. Probably they will influence users to decide for a registration on the national public vtx service in order to get full access right to all non-closed applications of that service.

PRESENTATION LAYER FACILITIES

The greatest problem in vtx networking seems to be the lack of a unified world-wide presentation layer data syntax. Studiing existing syntaxes - each of them being incompatible with other syntaxes - one can see that there is often no possibility to realize proper syntax conversions for interworking between vtx services, or for the connection of terminals other than initially specified for the given vtx system. This situation is caused rather by significant differences in presentation layer capabilities associated with each syntax, than differences in syntaxes themselves. In addition some syntaxes are associated with largely device dependent presentation layer capabilities.

A good example for both a large difference between presentation capabilities /levels of terminals/ and presentation layer functions hard to implement in a terminal, is the CEPT alphamosaic display specification /6/. Although terminals of levels CEPT A.1 through A.4 use the same syntax, downward conversions from the highest CEPT alphamosaic level /A.4/ to lower levels is practically impossible because of Dinamically Redefinable Character Sets and additional attributes used on that level. In addition conversions from level A.3 to levels A.2/.1 also may cause problems with parallel attributes and accented letters.

Device dependancy of level A.4 is caused by DRCS and too complex attributes. Both of these presentation layer capabilities require very complex hardware to be developed, or one can buy a VLSI decoder chip /like eg Philip's "EUROM"/ which eliminates earlier day's hardware cemetaries in terminals.

Full software-emulation of the above mentioned two groups of presentation layer capabilities is recently not possible due to the following two facts:

- Personal Computers available recently mainly provide bit-map structures which are largely incompatible with both DRCS-raster and DRCS-colouring requirements of CEPT A.4 terminals.

- CPUs in today's PCs and terminals are not yet powerful enough to

process in real-time all dynamic effects which may result from certain attribute settings in level A.4 terminals.

But all the above according level A.4 does not interest users if terminal costs are acceptable, and PTTs if terminals are available on their national market. All the implementation problems caused by excellent presentation capabilities of a CEPT A.4 terminal are matter of manufacturers. They can solve them indeed.

As a contrary to CEPT alphamosaic syntax one can mention NAPLPS developed by ATT. This presentation layer protocol is largely device-independent and also provides higher level alphageometric presentation capabilities like TELIDON and CEPT C-levels do. May be it was the right solution unfortunately not accepted in Europe? In fact in Europe recently broadly used are only different alphamosaic standards. Alphageometric terminals are far from a deep penetration. Therefore a european PTT or other european provider of a vtx service must now concentrate on the problem of alphamosaic presentation layer syntaxes and different levels of alphamosaic terminals used in intra-european and extra-european vtx services.

THE BASIC PRESENTATION LEVEL AND SYNTAX TO BE USED

In order to ensure full interworking with other vtx services which use an alphamosaic presentation layer data syntax the basic terminal of a national vtx service must be able to display frames coded in the highest currently existing alphamosaic syntax: CEPT A.4. Full interworking shall mean that two-way access to both frames retrieval and transactional facilities of interconnected vtx services is properly realized. That can be true for services with the same A.4 level and CEPT syntax of their reference terminal.

Another advantage of CEPT A.4 on the user level is attractivity. In addition to a powerful standardized alphanumeric character set it provides a large set of block- and smooth-mosaics and other miscellaneous characters together with DRCS and many additional attributes, format effecting possibilities, all that not available in earlier lower level syntaxes. Upward conversions from lower levels /even with incompatible syntaxes/ do not cause serious problems. Only some additional software modules in gateway servers are needed, and probably a slight increase in CPU-load of these servers is to expect.

OTHER PRESENTATION LAYERS AND SYNTAXES TO BE SERVED

This is not a problem of intermediate nature. Although many european PTTs declared their decision to introduce in the near future CEPT presentation layer data syntax, CEPT incompatible vtx and non-vtx terminals remain a strong reality independent of the developments of vtx in Europe.

The greatest disadvantage of today's vtx systems is probably that they support with a few exceptions only one presentation level with a single syntax defined to it. All terminals with a lower presentation level or an incompatible syntax are excluded from these services. That could be avoided in future vtx systems first of all in order to ensure the possibility of international interworking of vtx services. Due to the fact of long term existence of CEPT-incompatible terminals and of the lack of a worldwide unified vtx standard future vtx systems could incorporate multilevel and multisyntax presentation layer facilities even if

that will cause a burden in information provider's activity, as it was in the case of the german "Uebergangsloesung" /Intermediate Solution/. In other words: There is a need for multiprotocol presentation layer capabilities of future vtx systems both on terminal side and on gateway connections.

There is a significant difference between multiprotocol function on the terminal side and on gateway connection. The first problem is accented mainly on character sets, attributes, format effectors etc., the second may be associated rather with higher level functions within the same layer hidden from a "dumb" terminal. Such functions can be: Handling of well structured presentation elements /different kinds of fields/, control of a complex "Presentation Image", use of a "Format Service" and others, like defined in the german EHKP /1/. Application layer functions also are part of a gateway-side protocol machine. There is a clear correspondence between keying sequences and promptings on the terminal side, and application layer functions on the gateway side. The latter relation provides services to an application similarly to OSI Virtual Terminal and File Access protocol machines. Gateway protocols even may be /but are not/ transparent to presentation layer functions on the terminal side if necessary. Practically there is a possibility to split presentation layer functions in two separate parts /a kind of sublayers/:

- presentation layer data syntax on the terminal side,

- presentation layer protocol machines on gateway connections.

Multiprotocol capabilities of the latter and that of the application layer are basis for a flexibility in vtx-networking. The first part also requires multiprotocol capabilities in order to ensure proper realization of international gateway connections associated with multilevel protocol conversions and probably with multistandard databases, or multistandard parts /international subsets/ of national vtx databases. If there will be no general solution found for this mode of operation, and that first of all for the service of lower level alphamosaic terminals /ASCII TTY and VDU, Prestel initial level, CEPT A.1/.2, Antiope-CEPT A.3, TTX and others/ than only a "store-and--retrieve" type of interworking between vtx services with different presentation standards will be possible, using a partially common and very limited subset of both control and displayable characters of existing standards /See "Annex A" of /5/: "Initial level of the Interworking Data Syntax for Videotex"/. So for users of the system with a lower level data syntax basically only a "mailbox-to-mailbox" type interworking could be made available. Due to the nature of interactive videotex this is probably unacceptable, therefore at least serious further studies are required in order to support the realization of multiprotocol vtx systems of the future.

LITERATURE

/1/ BTX Rechnerverbund, Protokollhandbuch fuer den Anschluss Externer Rechner /EHKP Vtx Specification, Deutsche Bundespost, 1983/

/2/ Prestel Gateway Protocol Specification /British Telecom, 1981/

/3/ CCITT X.3 /1984 Draft/

/4/ CCITT X.29 / " " " /

5/ CCITT T.101 /AP-VIII, 1984/

6/ CEPT T/CD 06-01, Videotex presentation layer data syntax /1983/

7/ Functional Specification for Bildschirmtext Terminals /Deutsche Bundespost, 1983/

8/ Prestel Product Definition /Annex VIII of GEC's proposal, 1983/

9/ Tender Invitation of the Austrian PTT /1982/

10/ BTX Benutzerhandbuch fuer Anbieter /BTX User's Manual for IPs, Deutsche Bundespost, 1985/

11/ DATEX-P Charges /DBP, 1984/

12/ PSS Charges /British Telecom, 1983/

Activity of the Hungarian PTT in the field of public data networks

L. Almási, A. Kertész, A. Lábadi and P. Horváth

Computer and Automation Institute
Hungarian Academy of Sciences

Central Telegraph Central
Hungarian PTT

To satisfy the growing needs in data transmission facilities the Hungarian PTT builds an integrated digital data network. Some equipments are imported, others are developed and manufactured in Hungary. The Hungarian Academy of Sciences in one of the partners of the PTT in the developments.

The main features of the experimental packet switching node developed by an academie institute for the PTT is describer below.

Introduction

With putting into operation of his circuit switched public data network in 1981. the Hungarian PTT has closed with good results a many years development period, and has made the first step towards realization of a giant plan for several decades. This plan is aimed at

- development of public data network and introduction of data services based on this network;
- introduction of new services, providing sophisticated information handling capabilities.

The last services - telematic and message handling systems services use network services of the afore mentioned data network, but some of them, in order to ensure the most popular use, will be accessible for users via the public switched telephone network too.

This presentation is aimed at giving information on networking issues.

General aims and results of present activities

The opening of the public data network and the present activities show, that the managemenet of the Hungarian PTT took notice of the fact, that for many years the development of the public data network can be considered as the most effective way of satisfaction of the rapidly growing subscriber's needs in data transmission. The bottlenecks of the present state of the data transmission service can be characterized by the fact, that this year the number of data stations to be put into operation has exceeded the number of operating data stations. The speed of growth of the waiting list seems to remain higher even in longer term as the speed of growth of operating data stations. It's clear, that the present telephone network can not offer neither the necessary transmission capacities, nor the required quality of service.

Consideration of this facts has lead to the following basic decisions:

- it's necessary to build a basic data transmission network using TDM systems with fixed and elastic /statistical/ capacity allocation to increase the effectiveness of use of local cables and long distance transmission systems;

- based on the TDM transmission hierarchy three kinds of data network services shall be provided
  leased line service
  circuit switched service
  packet switched service;

- the data network services are considered as the basic means for rising the quality of service;

- the quality of service in case of data transmission over the telephone network can be enhanced only together with the further development of the telephone network; this process is too slow from the point of view of teleprocessing and computer networking;

- the data network shall offer full service to the users, e.g. standardized digital DTE/DCE interfaces shall be provided, liberating the users from the struggle with data transmission problems;

For proving that the above mentioned decisions are on the way of realization, the following facts can be listed:

- the basic plans of the integrated digital data network /eg. plan of reference connections and allocation of network parameters, network synchronization plan, network maintenance plan/ are ready and accepted or near, to the acceptance;

- the specifications for multiplexing, transmission and other equipment to be developed and manufactured in Hugnary are ready;

- the development of many equipment is started /V/X converter, asynchronous TDM, synchronous DCE/, the first prototypes will be tested at the end of this year; some development contracts with the telecommunication industry are in preparation;

- a powerful bit error rate measuring and protocol testing equipment is already on the market;

- the circuit switched data network will be extended next year by more than 400 stations; it will essentially ease the tension between the requirements and the PTT's possibilities; and last but not least;

- in the first half of this year the experimental packet switching node of the PTT has been put into operation.

The last item - in spite of his experimental nature - is very important, since it serves as a drill ground for the PTT's personnel in getting the necessary skill in usage of the packet switching technology, and at he same time it is used for covering real subscribers requirements.

The experimental packet switching node

The experimental packet switching node consists of two basic units - the switch, based on a hungarian 16 bit mini computer TPA- 70, and - the MS 600 microprocessor based PAD.

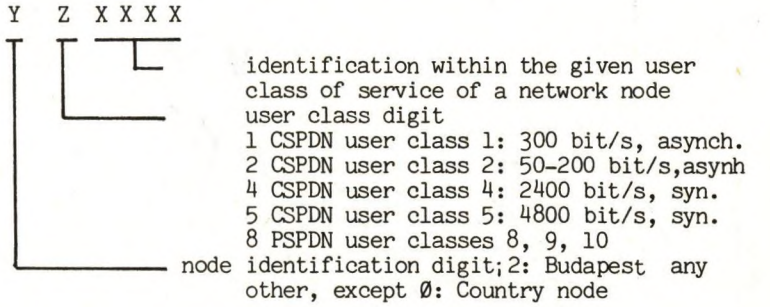
The TPA 70 packet switching software and the MS PAD were developed at the Computer and Automation Institute /CAI/ of the Hungarian Academy of Sciences. The TPA node itself contains a PAD too. The last is used for the interconnection with the asynchronous data network, and the MS PAD is used for the interconnection with the telephone network and with the asynchronous data network.

Synchronous DTE-s can be connected to the node either directly, using X.25 interface, or via the synchronous circuit switched data network. In the last case after the transparent network connection has been established to the node via the CSDN, the DTE starts with the X.25 procedure.

The network configuration is shown on figure 1.

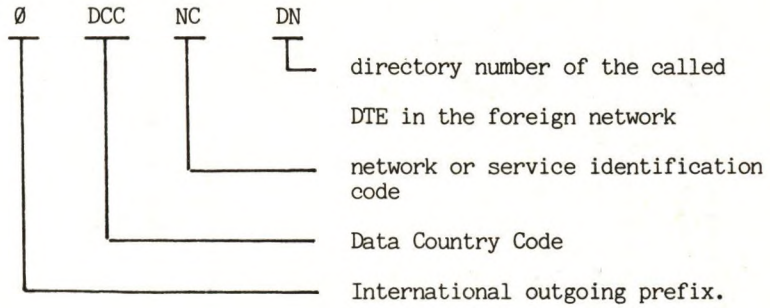
The figure shows the direction of establishment of connections between the three different switching systems /telephone switch, packet switch and data circuit switch/, the signalling systems used. It's to be seen, that subscriber signalling is used between the packet switching system and the circuit switching system, so the first one acts as a subscriber to the circuit switched system. As this signalling systems can not transfer the called number, all connections via the CSDN /and also via the PSTN/ are established using two step selection.

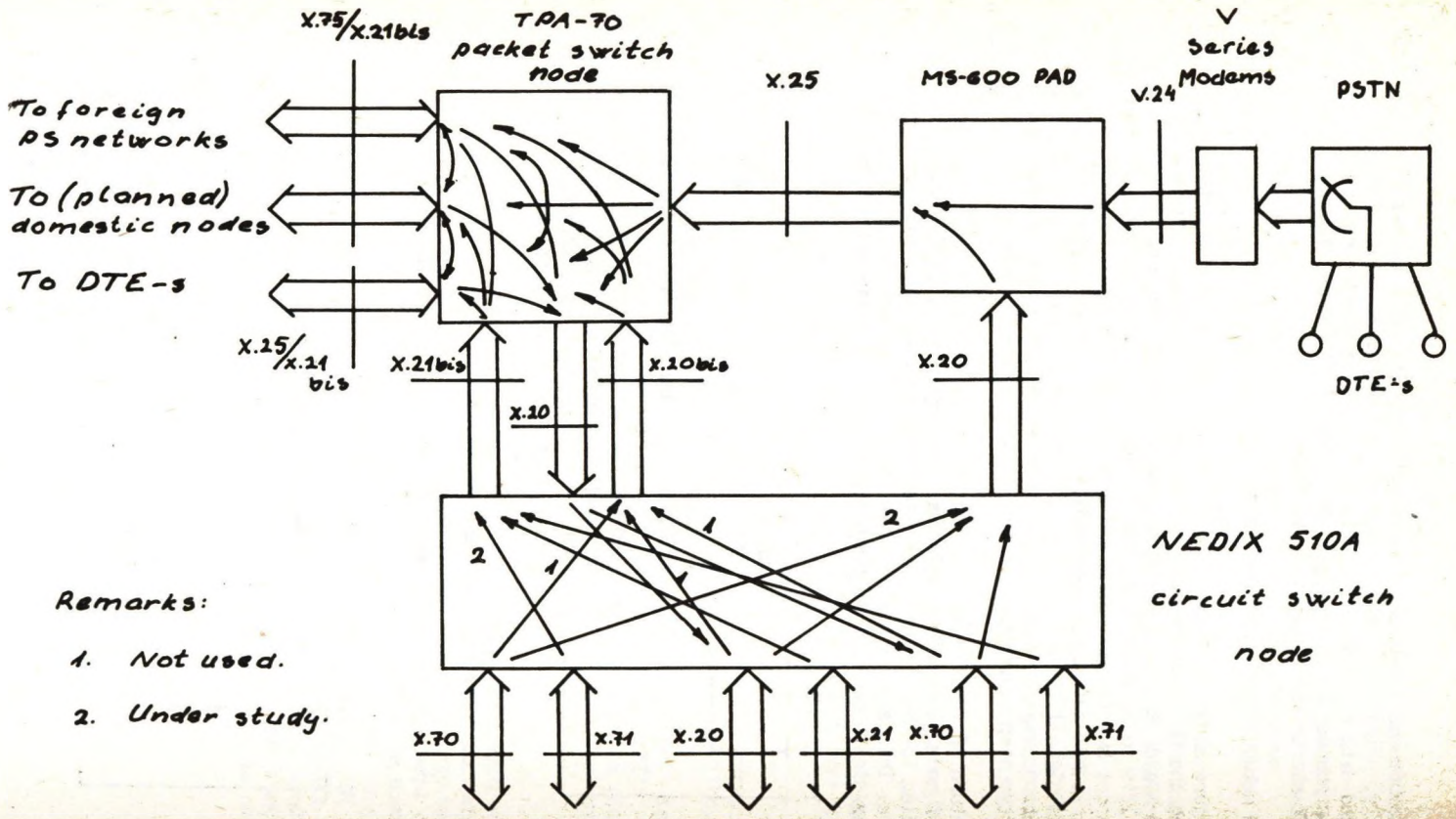
The return of traffic to the originating network is prevented in the TPA node. The node uses fixed routing tables, filled out according to the numbering scheme, accepted for the Hungarian Data Network. This scheme corresponds to CCITT Recommendation X.121. The six digit domestic numbering scheme is constructed as follows:



The Ø as first digit means international outgoing prefix, and shall not be transmitted on the international trunk.

An international outgoing call shall use the following structure of selection signals:





Remarks:

1. Not used.
2. Under study.

Figure 1. Call routing possibilities

To Hungary DCC= 216 is assigned in Recommandation X.121. The Hungarian PTT has applied for assigning to his CSPDN NC=0, and NC=1 to his PSDPDN.

The switch is based on a Hungarian made minicoputer, the TPA 70. It is a general purpose mini with 64 kbytes of memory. It has a 5 Mbytes disc, a printer and a console terminal. Beyond the usual V.24 asynchronous lines for the PAD it is equipped with DMA network interfaces for the bitstuffing frame I/O. Due to the characteristics of the machine we can not expect more communication lines connected than 20 and higher throuput than 30-40 kbit/s.

Though the actual configuration may vary according to the needs a typical setup can look like the following:

- an X.25 line /4800 baud/ to the MS 600 which concentrates some 10 PAD lines of the asynchronous data network and the telephone network for incoming calls /4800 baud/
- connection - with some 5 ports - to the asynchronous data network for outgoing calls in user class 1
- at least one local PAD line for the operator of the switch /9600 baud/
- a few X.75 lines for the international connections /4800-9600 baud/
- some 10 X.25 lines for the domestic users /leased or circuit switched, 2400-4800 bp/s.

Compatibility with the Standards

The X.75/X.25 Interface essentially follows the 1980 Recommendation of the CCITT. There are certain restrictions. Only virtual calls are supported. /There is no datagram service and permanent virtual circuit./ In the network facility field only the three mandatory facility fields are /call id, throuput class, transit network id/ are maintained correctly by the switch, all the rest as well as the user facility field are ignored, but transferred transparently.

The interface can be DCE or DTE /STX/ST/. It is a special feature that the line can run in regular synchronous or asynchronous mode with character stuffing rather than bitstuffing. In that case the frame format is similar to that of the transparent IBM BSC.

The PAD Interface follows the 1980 Recommendation of the CCITT. Network User Identification is asked for which is valid for the whole session. A session may last during several calls. It starts at carrier-on and terminates at either carrier-off or at the explicit "DISC" command. The NUI has two meanings:

- user identification
- permission

Operation of the switch

The switch is fully automatic, no operator is needed. However it runs a network callable process, the Monitor to supervise the operation - if needed. In fact the Monitor has two entry points. The one is the Super Monitor, the other is the User Monitor. The supervision of the switch is done through the Super Monitor accessible usually by the local operator terminal/s/ only. The access of the Super Monitor /that is the full control over the switch/ trough network lines is also possible. It is a matter of permission. The operator can check the load and the status of the lines, may put lines in or out of service, can initiate the down load of the dayfile information assembled so far, change the password file, and so on. Certain network test functions are also included in the Super Monitor such as the "echo", the "emitter", the "sink". The User Monitor - with a limited command set - is available for the network users. It provides information about the status of the lines, addresses used, etc as well as makes it possible to send messages to the operator.

The switch maintains a dayfile /log-file/ to store reports about every major event. A block of the dayfile may contain information about a call establishment, a clear, an accounting etc. Others report about the changes in the status of the lines or about some failures like frame reject on the LAPB level, hardware error on an interface, an so an. The dayfile is downloaded into a mainframe for further processing. It can be done either via direct connection or through the floppy disc of a microcomputer connected to the switch.

Permissions

Two levels of security are applied concerning access permission. The first is the absolute security. It is related to a network address and defines those physical lines - if any - which may not access that network address. The other is the dynamic security, applied by the PAD only. It is also related to a network address and defines a security code and those physical lines which can acces the network address only if their dynamic security conforms to the security code. The dynamic security of a PAD line is set according to the NUI entered.

The absolute security has priority over the dynamic security.

Routing

The routing is done by the routing tables at call time. The switch searches in its tables till finds one, matching the destination address of the call packet. Than it checks the security, updates network facility field - if any - , performs required transformation on the call packet and puts the call to the outgoing line. The routing is fixed for the duration of the wirtual call. To use alternate routes for different calls is possible, but not yet used.

SESSION 4
DESING AND IMPLEMENTATION OF LOCAL AREA NETWORKS

THE DEVELOPMENT OF AN ETHERNET LOCAL AREA NETWORK FOR SCIENTIFIC DATA ACQUISITION

Robin Tasker

Science and Engineering Research Council
Daresbury Laboratory
Warrington, Cheshire WA4 4AD
England

This paper describes the development of a data acquisition network for the Synchrotron Radiation Source of the Daresbury Laboratory. The development was based upon the evaluation of Ethernet technology and resulted in the specification of a Remote Procedure Call protocol to provide the required service. This protocol is described together with the measurements of the network performance, where data rates in excess of 60 Kbytes per second are consistently seen.

1. INTRODUCTION

The Synchrotron Radiation Source (SRS) of the Daresbury Laboratory offers research facilities for the scientific community. The facility is organised as a number of experimental areas located on the various SRS beam lines. Each experimental areas has an associated experimental station which is based around either a PDP 11/04 or an LSI 11/23 computer. During experimental runs, these computers are fully utilised controlling the experiment, collecting and storing data to disk. A data acquisition network exists to link these experimental station computers to a central data concentrating computer. This configuration is necessary as the experimental station computers do not have the power to carry out data analysis, and do not have the storage capacity to cope with the quantity of data that is collected. Currently, the data is forwarded to the Laboratory mainframe computer for long term storage from the data concentrating computer, although a certain amount of data analysis is carried out here. The current network configuration between the various experimental computers and the data concentrating computer, a VAX 11/750, is reaching its limit with respect to the number of experimental stations that can be supported, and the data rates that the network can deliver. In the future there will be additional experimental stations to support and they will wish to transfer considerably more data and to do so at considerably higher data rates.

A requirement was produced defining the needs of a new data acquisition network. The requirement of the network included the ability to transfer data from an experimental station to the VAX, to transfer data between any two experimental station computers, to allow control message passing between experimental stations and to provide for terminal traffic. It is the data transfer requirements which are primarily responsible for the high data rates required of the network. One experimental station wished to transfer 8-Mbyte files in less than 5 minutes at regular intervals, and a second wished to transfer 400 Mbytes over an 8 hour period. This was generalised to a requirement to provide a peak working data rate of 45 Kbytes per second and a guaranteed data rate of 30 Kbytes per second.

An evaluation exercise was carried out to determine whether an Ethernet local area network would be a suitable medium on which to base the new data acquisition network. The evaluation was based upon the idea that data throughput available to an application was the single most important criterion for judging the suitability of a particular technology. Experiments were carried out to measure the performance

between an experimental station computer and the VAX. A very simple software protocol was designed and written which guaranteed a reliable data transfer, and provided the means to vary transfer parameters so that sensible decisions could be made in the design of the final software protocol. These experiments are discussed below but as a result it was proposed to install an Ethernet local area network as the basis of the new data acquisition network.

The software for the new network clearly had to be more sophisticated than the experimental protocol, but it had to retain sufficient simplicity so that the high data rates we had recorded could be maintained. We investigated the use of several light-weight protocols but none seemed ideal solutions to our particular requirements. The concept of remote procedure call seemed to us to be best suited. Unfortunately the existing schemes seemed overly complicated for our requirement of high speed, reliable data transfer. As a result we defined our own remote procedure call (RPC) protocol and the new network has been based on this work.

A working definition of the protocol is described below, together with details of its implementations. The developments of this network is now complete and performance measurements have been carried out to determine whether the final product performs to the initial requirement. The results indicate that this is indeed the case.

2. ETHERNET EVALUATION

2.1 Methods

The data concentrator, a VAX 11/750 running VMS version 3.2 was connected via an Interlan Unibus interface to a dedicated Ethernet cable running at 10 MHz. Attached to this cable via an Interlan Q-bus interface was an experimental station computer, an LSI 11/23 running RT11 version 4. All measurements were taken when no other applications were running in either machine, and the results therefore reflect only overheads imposed by the Ethernet hardware and software interfaces in addition to the normal system overheads.

The software protocol was a minimal packet-sequencing scheme to ensure that data was neither lost nor received out of sequence. The ability to vary an acknowledge window size, and the Ethernet packet size were also included. The acknowledge window size was defined as the number of packets sent from the LSI to the VAX before an acknowledgement packet was required in the opposite direction. Received packets, generated by the LSI were checked to ensure that the sequence numbrs they contained matched that which was expected. In a similar fashion, the acknowledgement packets were checked for consistency. Data transfers wre performed from LSI memory to VAX memory, from LSI memory to VAX disk and from LSI disk to VAX disk. In all cases the time taken to transfer a similar quantity of data was recorded, for a range of Ethernet packet sizes and for a range of acknowledge window sizes. As a result, data throughput was computed as a function of packet size for each of the acknowledge window sizes defined.

A full description of these experiments and an analysis of the results may be found elsewhere [1].

2.2 Results

Table 1 shows the data rates [Kbytes per second] recorded for each type of data transfer from the LSI to the VAX for acknowledge window sizes of one, two, four and eight for the maximum packet size allowed by the Ethernet specification [2]. In all cases the data throughput showed a steady decline with decreasing packet sizes. The limit of eight as the maximum acknowledge window size was set as any higher value resulted in data loss by the Interlan interface board.

Table 1

	Acknowledge window size 1	2	4	8
Memory to memory	59	79	96	107
Memory to disk	48	65	79	89
Disk to disk	30	42	49	61

Data throughput [Kbytes/second] recorded for transfers between an LSI 11/23 and a VAX 11/750 for a range of acknowledge window sizes, and for a variety of sources and sinks for the data with an Ethernet packet size of 1480 bytes.

The results show that the data throughput increases with increasing acknowledge window size, regardless of which type of data transfer is considered. The maximum throughput achieved was 107 Kbytes per second for memory to memory data transfer, and the worst was 61 Kbytes per second for disk to disk data transfer.

Table 2 shows the data rates recorded for each type of data transfer from the LSI to the VAX, but with a packet size of 64 bytes. The results show a similar pattern to those of Table 1 but at a very much reduced level. The maximum throughput recorded here is 16 Kbytes per second compared with 107 Kbytes per second with the maximum packet size.

Table 2

	Acknowledge window size 1	2	4	8
Memory to memory	9	12	14	16
Memory to disk	7	11	13	15
Disk to disk	5	6	8	8

Data throughput [Kbytes/second] recorded for transfers between an LSI 11/23 and a VAX 11/750 for a range of acknowledge window sizes, and for a variety of sources and sinks for the data with an Ethernet packet size of 64 bytes.

2.3 Discussion

The most important conclusion to be drawn from these results was that an Ethernet was capable of providing the data rates that the requirement specified. This can be seen from the disk to disk throughput with an acknowledge window size of eight. On the basis of this result the recommendation to use Ethernet was made.

As one would expect, Table 1 also showed that the disk to disk data transfer was considerably slower than the memory to memory data transfer. This is a reflection of the overheads involved in disk accesses. The increase in throughput with increasing acknowledge window size rsults from the decrease in processing imposed by the software protocol. Acknowledgements are sent from the VAX to the LSI less frequently with an acknowledge window size of eight and this has a cumulative effect on the data throughput. These results are for the maximum Ethernet packet size allowed. Table 2 shows the dramatic effect of transferring 64 bytes per packet on the total data throughput. Clearly there is a fixed overhead per packet transferred from the LSI to the VAX, and the larger the packet, the less the number of packets that need to be transferred.

These last two points were important to us when designing our network-level protocol so that data throughput might be maximised.

3. SOFTWARE DEVELOPMENTS

3.1 Software Protocols

It was decided to implement as the link control software the logical link control (LLC) protocol as defined by IEEE P802.2 [3]. This standard describes the function and the features and the protocol of the logical link control sub-layer in the IEEE 802 Local Area Network Protocol. The LLC sub-layer constitutes the top sub-layer in the Data Link Layer and is common to the various medium access methods that are defined and supported by the IEEE 802 activity. The data link protocol defined here is defined as Class 1 to provide a data-link-connectionless service.

We decided that the use of International Standards to be sensible wherever possible so that future network-level developments could be integrated without disruption to existing services.

For the network-level protocol we designed and implemented a remote procedure call protocol (RPC) [4]. The basis of RPC is that a process in one network machine is able to invoke a procedure or set of procedures in some other network machine. This generally involves the exchange of two packets across the network. The initiator of the exchange, called the requester, transfers the request which contains the necessary parameters, to the remote procedures, called the server process. As a result of this request, the response is returned and this may or may not contain returning parameters. To this basic request/response protocol, two addition extensions have been added. Firstly before any such exchange occurs, the requester must issue a multicast on the Ethernet containing the name of the service it requires, or the name of the required network machine. Secondly, during data transfer, not all requests are required to be acknowledged. The former extension is include so that each requester need not know the location of the services on the network. As location is equated with physical Ethernet address, this allows interface boards to be changed without the necessity of re-configuration. The latter extension is a direct result of the evaluation measurements we had previously made. Higher throughput can be achieved if not all data packets are acknowledged.

Figure 1 shows a typical RPC exchange between a requester and a server which will give filestore access on the remote machine. The requester issues a multicast containing the filestore server name. As the remote machine knows of this server type, it replies and in so doing returns its physical Ethernet address. The requester issues a create server request to which the remote RPC manager replies success when the process is created. The requester opens the remote file and transfers data. Not all the data packets sent by the requester elicit acknowledgements from the server thus speeding up data transfer. When the data is successfully transferred, the requester sends a close file request which must be acknowledged. Finally a delete server request is sent, and its acknowledgement completes this RPC exchange.

An Ethernet does not inherently provide a reliable service as there is no built-in assurance that a packet successfully transmitted has been successfully received. The service we wished to provide required a reliable data transfer mechanism, so RPC was designed to detect and recover from network-level errors.

Although RPC is inherently a connectionless operation on the network, our initial application (of reliable, high speed file transfer) does imply a connection at the application level. Each RPC packet contains a sequence number which is validated at the remote end of the exchange to detect lost packets or packets received out

f sequence. A "Single-shot" mechanism is defined within RPC to allow for true onnectionless applications and this is the subject of future developments.

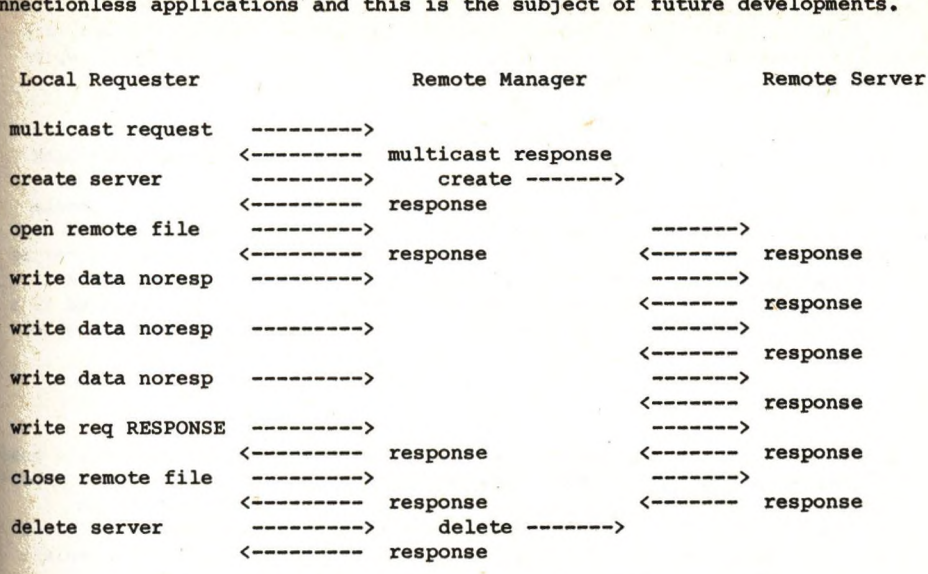


Figure 1. A typical example of an RPC exchange.

Requests can be classified into two distinct groups, non-repeatable requests and repeatable requests. If a non-repeatable request is lost, some repeatable request is initiated, and if a repeatable request is lost it is repeated upto some specified limit. If this limit is reached, the worst is assumed. To aid error recovery, RPC defines a check status request which may be transmitted if an error is detected. This returns from the remote process details of the server process and of the state of the RPC network connection. This allows a requester to synchronise its activities for recovery. The check status request is the fundamental example of a repeatable request being invoked as a result of the failure of a non-repeatable request.

3.2 Software Implementation

The implementation details varied between the LSI and the VAX. Initially, the experimental station software - the LSI - required only requester software, and the VAX required only server software. Also the LSI implementation was limited by the available space. RPC can be logically divided between requester software, server software and an RPC manager. The first two entities have already been defined, while the latter is no more than the software to implement the rules defined by the RPC protocol specification.

The LSI requester software has been implemented as a set of subroutines to perform the tasks involved in an RPC exchange. The software is structured to perform the tasks involved in an RPC exchange. The software is structured into layers with access available to the user at each layer, the idea being that each layer adds more sophistication to the service that is provided. The lowest level of access allows data acquisition programs to be tailored to the particular hardware or for high performance. Here a knowledge of RPC and of Ethernet is required, though the subroutines provide the access to the software driver of the Ethernet interface. The second level provides routines to be includd as an integral part of data acquisition programs. Here all details of RPC and Ethernet are hidden. The highest level is designed for the novice user, and provides a set of utilities to perform the whole of a required RPC exchange.

In this way a distinct RPC manager does not exist and the functions of the RPC manager are distributed between the subroutines making up the requester software. The task is simplified as only one requester will ever be active at any one time in an RT11-based experimental station computer. The full details of this implementation are described elsewhere [5].

The VAX implementation has an RPC manager process permanently running, and this process controls all parts of the RPC software currently implemented. The link control software is started by the RPC manager at system start-up, while the server processes are created in response to a received request from some remote requester. The design allows new server types to be easily added, and for a requester to have the exclusive use of a server. When more than one requester is active, the total bandwidth of the Ethernet will clearly be shared, but once in the target machine - the VAX - it is intendd that further delays are reduced to a minimum. The full details of all the VAX-based RPC software are described elsewhere [6].

4. RPC NETWORK PERFORMANCE

We have all along taken the pragmatic view that data throughput is the single most important factor in assessing the worth of this network to a user. We have therefore equated the measurement of throughput with that of performance. These measurements show firstly the throughput a user might experience under different network loadings, and secondly a measurement of network performance. A complete description of these measurements can be found elsewhere [7].

4.1 Methods

The results are based on the time taken to transfer a file of 1000 Kbytes from an experimental station computer to the data concentrating computer. The experimental station computers were all LSI 11/23's running RT11 Version 5.1 using Interlan Q-bus interface boards to connect to the Ethernet via an Interlan transceivr. The data concentrating computer is a VAX 11/750 running VMS Version 3.7 using an Interlan Unibus interface board again connected to the Ethernet via an Interlan transceiver. All the Ethernet equipment used was designed to the IEEE 802.3 specification. In all cases there were no additional users of either the computer systems or the network.

Times for the data transfer were recorded for the different experimental stations when there was no other user of the network, one other experimental station transmitting, and two other experimntal stations transmitting. The results represent the average of at least five separate timings for each station under all the different experimental conditions.

4.2 Results

Table 3 shows the time taken to transfer the test file from the different experimental stations to the VAX, together with the average for all stations.

These figures show a degree of variability between the experimental stations, but generally are consistent at about 16 seconds. Table 4 shows the quivalent data throughput per station as more experimental stations access the network simultaneously. The throughput per station falls from 62.5 Kbytes per second for an experimental station having sole use of the network to 26.8 Kbytes per second when two other experimental stations are sharing the network. The total network throughput increases from 62.5 Kbytes per second for a single active experimental station to 80.4 Kbytes per second when three stations are active simultaneously.

In the case of two and of three experimental stations transmitting simultaneously it was noted that errors were detected by the RPC manager, and the requester processes invoked the error recovery procedure outlined above. These errors

Table 3

	Number of stations transmitting simultaneously		
	1	2	3
Station 1	15.5	24.9	not tested
Station 2	17.4	26.1	32.0
Station 3	15.2	28.5	40.0
Station 4	15.9	28.9	39.8
Mean time	16.0	27.1	37.3

The time in seconds to transfer a 1000 Kbyte file from an experimental station computer, an LSI 11/23 to a VAX 11/750 under different network loadings determined by the number of experimental stations simultaneously transmitting.

Table 4

	Number of stations transmitting simultaneously		
	1	2	3
Data rate per station	62.6	36.9	26.8
Total network data rate	62.5	73.8	80.4

Data throughput in Kbytes per second measured per station and for the total network under different network loadings determined by the number of experimental stations transmitting simultaneously.

occurred very infrequently and we recorded at most three such instances during the transfer of the test file. In all cases recovery was successful, and the test file successfully transferred.

4.3 Discussion

The variability in time taken to transfer the test file between the experimental station computers can be explained by the different Winchester disk systems that they possessed. There were no other differences between the computer systems.

The average data rate for an experimental station having sole use of the network was slightly higher than the equivalent measurement in our evaluation. The difference, 61 Kbytes per second on evaluation and 62.5 Kbytes per second here, is minimal, but the sophistication of the RPC software is far greater. Clearly the evalution software was a little crude, but there seemed no point in adding to it when it showed that Ethernet was a suitable technology.

The decline in data rates as more experimental stations transmitted was expected and it was clear that it was the VAX which was becoming the limiting part of the network. The errors detected resulted from the VAX part of the network missing packets, or not processing the packets fast enough, and the requester software timing out on the request. Our experience with RPC during its development has meant that the latter reason for network errors has been all but eliminated. We have adjusted the timeout period of the requester software to minimise the occurrence of timeouts during data transfer.

The fact that network errors were detected suggested that a degradation in performance should have been visible. While the total data throughput is increasing, the rate of that increase is diminishing. This is in part a result of

the error recovery procedures that RPC imposes on requester software after an error has been detected. However the major contributor to this degradation is the reduced speed of processing of the VAX. The acknowledgements are sent and received within the timeout period, but at the end of that period. As a requester is not allowed to transmit until the previous acknowledge has been received, this has implications for the data throughput.

5. CONCLUSIONS

These measurements confirm that the original requirements have been met, and the software is now installed in a service environment. To date only one experimental station is using this software, but other stations are due on-line in the near future. Already further developments are planned, in particular to add requester software to the VAX so that servers may be accessed from this computer.

6. REFERENCES

[1] Rake, F.M., Tasker, R. and Kummer, P.S., Performance measurements of an Ethernet local area network, Interfaces in Computing 2 (1984) pp 221-227.

[2] A bus utilising CSMA/CD as the access method. IEEE Stand. P802.3 (Institute of Electrical and Electronic Enginers).

[3] Local Area Network Standard, Logical Link Control. IEE Stand. P802.2 (Institute of Electrical and Electronic Engineers).

[4] Kummer, P.S. and Tasker, R., Remote Procedure Call for Ethernet: Protocol Specification DL/CSE/TM35 (1984).

[5] Hines, D., System and implementation guide to the RT-11 RPC code, Internal Document (1984).

[6] Rake, F.M., Tasker, R. and Kummer, P.S., VAX/VMS Remote Procedure Call System Documentation, Internal Document (1984).

[7] Tasker, R., Rake, F.M., Kummer, P.S. and Hines, D. The use of a remote procedure call protocol for high speed data transfer on an Ethernet, in press.

THE SOFTWARE ARCHITECTURE OF A LOCAL AREA NETWORK

A. Arató, I. Sarkadi-Nagy, J. Sulyán and F. Telbisz

Central Research Institute for Physics
H-1525 Budapest, P.O.Box 49
Hungary

A local area network has been developed, whose intended main application field is laboratory data acquisition and process control. The network uses a modified version of CSMA/CD. The transmission speed is 1 Mbits/s. The software architecture of the network is described and compared to the Open System Interconnection architecture of ISO. The transport and the logical link control layers are discussed in detail. Measurements concerning the task to task throughput rate as well as the network file transfer are presented in this paper.

1. INTRODUCTION

In the Central Research Institute for Physics a local area network has been developed with laboratory automation and process control as the primary fields of application. The name of the network is LOCHNESS, which stands for LOCal Highspeed NEtwork SyStem. The architecture of the local area network takes into consideration the requirements of such systems.

Process control systems are generally distributed systems. They consists of several subsystems, where each subsystem is controlled by a minicomputer and is devoted to special control functions, some kind of data acquisition or man- machines communication. In our systems 16-bit TPA minicomputers, developed in our institute, are used as subsystem control computers. A subsystem may contain several microprocessor based Intelligent CAMAC Crate Controllers (ICC-s) as control stations.

The subsystem control computers can exchange data among themselves without restriction, but a subsystem control computer only has access to the control stations of its own subsystem, and control stations belonging to different subsystems can not communicate among themselves. The only exception is the process control data-base management, to be mentioned later, which has access to the data-base in any subsystem control computer or control station belonging to any subsystem.

The communication medium is a coaxial cable both within a subsystem and between the subsystems. A sample lay-out can be seen in Fig. 1. Each subsystem uses a separate coaxial cable segment and the subsystems are connected by a common segment. The length of a cable segment is a maximum of 1 km., the transfer rate is 1 Mbit/s, and the cable is isolated from the machines up to 15kV DC.

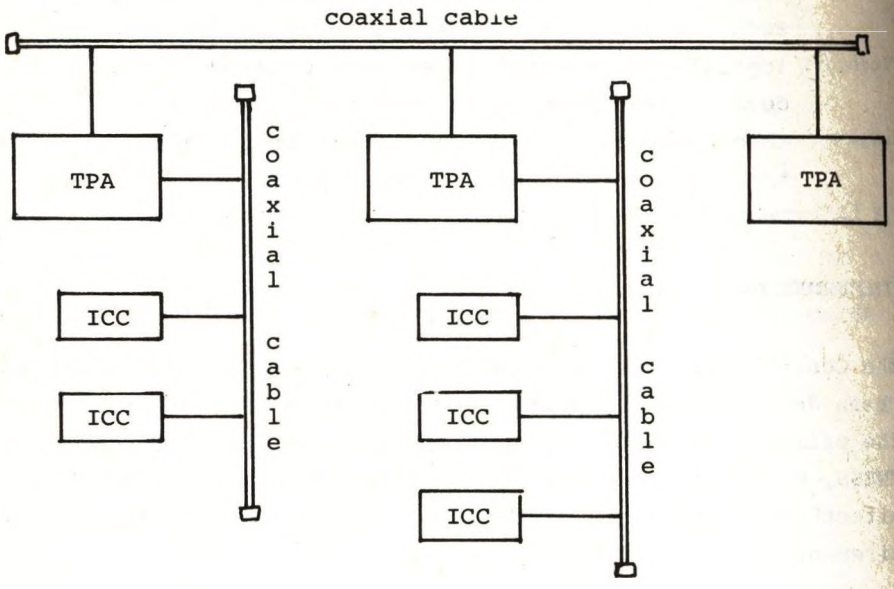


Figure 1: Sample layout

The line arbitration mechanism used in LOCHNESS is a modified CSMA/CD algorithm. The CSMA/CD technique is supplemented with a slotting system, which provides for fast acknowledgement and quick retransmission of the frames in the case of transmission errors. The slotting scheme can be used to implement contention free multiframe

control sequences too. This last one is a very important feature and it will be discussed with the logical link control. The arbitration method is described in [1].

2. NETWORK LAYERING

LOCHNESS is designed according to the layered structure of the ISO Open System Interconnection architecture [2], but we treated the recommendations as architectural guide lines only. We deviated from the reference model in all cases when some gain was expected either in efficiency or in the simplicity of the implementation. Some layers are not implemented, if their functions are not required, while others with strongly reduced functionality are incorporated into the neighbouring ones. Our protocols are clearly different from the OSI protocols.

The correspodance between the OSI reference model and LOCHNESS is shown in Fig. 2. In the figure the three layers of the IEEE 802 project are shown at the place of the two lowest OSI layers.

<table>
<tr><th>O S I</th><th>L O C H N E S S</th></tr>
<tr><td>Application</td><td>- Terminal Communication
- File Transfer Services
- Remote Terminal Facility
- Virtual Device Package
- Process Control Data-base</td></tr>
<tr><td>Presentation</td><td rowspan="2">Not implemented</td></tr>
<tr><td>Session Control</td></tr>
<tr><td>Transport Control</td><td rowspan="2">LOCHNESS Service Task
and
the program
of the ICC interface</td></tr>
<tr><td>Network Control</td></tr>
<tr><td>Logical Link Control</td><td>Driver + the microprogram
of the communication board
and
the program
of the ICC interface</td></tr>
</table>

OSI	LOCHNESS
Media Access Control	Interface hardware and microprogram
Physical Layer	Interface hardware

Fig. 2: Correspondance of OSI and LOCHNESS layering.

The components of the application layer of the intelligent CAMAC crate controllers are specific ones and thus they are not discussed in this paper. On the minicomputers four services are available for the users:

- With the help of the File Transfer Utility program files can be transferred between any two minicomputers connected to the network The File Transfer Utility supports remote directory listings too.

- With the help of the Remote Terminal Support program any terminal can be connected to a remote minicomputer and can be used in the same way as a local one, connected to that computer.

- With the help of the Terminal Interchange Utility messages can be exchanged, or a dialogue can be pursued, between any two minicomputer terminals.

- With the help of the Virtual Device Package a remote physical device (disk, magnetic tape or printer) can be used troughout the network in a completely transparent way.

Most or all of these services are generally offered in network software systems. A further very important and most interesting application layer software is a distributed Process Control Data Base system. It can manipulate on data in the memory of on the disk of a local or remote minicomputer, or in the memory of a remote intelligent CAMAC crate controller. Data can be accessed throughout the network in a transparent manner. The distributed Process Control Data Base system is described in a separate paper [3].

The presentation and the session layers are null layers in our case and are omitted.

The transport layer is, in our opinion, one of the most important point of our network architecture. It is implemented as a privileged system task. Its work and relation to the logical link layer will be discussed in section 3.

The network layer -- as in most LAN-s -- is very simple in our network. We do not need convergence functions and the routing is rather simple too [4]. As a cable offers full connectivity between the nodes connected to its, no routing is needed between the subsystem control computers. There is complete connectivity within a subsystem too. In this network architecture, as was mentioned earlier, the intelligent CAMAC crate controllers are only accessible from within their own subsystems and are hidden from any other subsystem. The only required routing is -- for the time being at least -- the selection of the proper cable interface. Thus the required network layer functions could be easily incorporated into the transport layer.

The Logical Link Control and Media Access Control is described in [1]. We should like to mention here only the most important features. The media access control as well as a considerable part of the logical link control is implemented in the communication boards equipped with a processor. The processor of the interface board can manage both the traffic control tables and data buffers. In the case of the minicomputers these are placed in the host memory, while in the Intelligent CAMAC Crate Controllers the communication board has sufficient memory to store both the control tables and the data buffers. Contention free multiframe sequences are supported by the media access control. The transfer of a message is carried out by one or more line sequences, each sequence consisting of the exchange of a few (three or four) frames between the message source and destination stations. The use of this way of operation will be discussed in chapter 3.

3. THE TRANSPORT LAYER AND THE LOGICAL LINK LAYER

LOCHNESS is a connection oriented network. Only a connection oriented logical link service is available even at the lowest layer too. As neither the logical link layer nor the transport layer has a

multiplexing-demultiplexing function, the connections of the logical link layer and of the transport layer can be mutually mapped with respect to each other in a one-to-one relationship. This means that no separate connect-disconnect procedure is needed for the two layers. The most important functions of the transport layer are the creation and deletion of the connections or logical links. The transport layer is implemented in the minicomputers as a special privileged task, named as LOCHNESS Service Task.

A logical link is dynamically created and deleted between two processes running in the minicomputers, if both processes agree to communicate. The two processes must have a preliminary dialogue with the help of the tasks representing the transport layers. A user process can turn to the LOCHNESS Service Task (LST) by using special I/O executive requests. In this three party handshaking procedure (the two processes and the transport layer) each program recognizes and agrees to be linked to the other task. A schematic diagram of the connect procedure is shown in Fig. 3, where Task 1 in Node A wants to connect Task 2 in Node B.

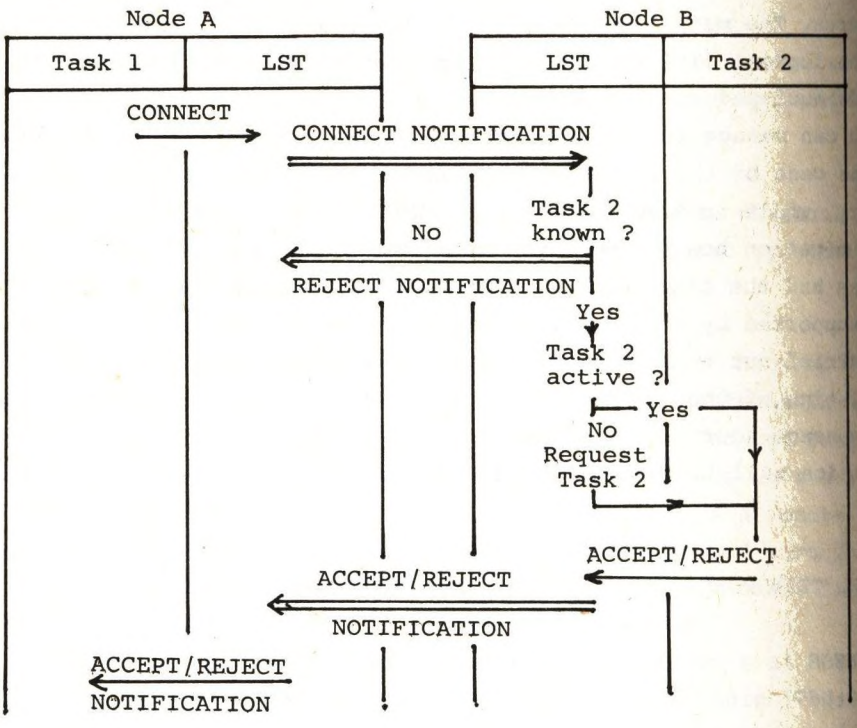


Fig. 4. Schematic diagram of a connect procedure.

With the cooperation of the transport layer the following steps are carried out in each minicomputer:

- A transport Service Access Point (SAP) is allocated to the logical link by the transport service layer.

- A logical link Service Access Point is also allocated to the virtual circuit. As this logical link Service Access Point is not multiplexed, a one-to-one correspondance is established between the transport layer and logical link layer Service Access Points.

- The logical unit number (LUN) designed by the task is attached to the logical link, i.e. this logical unit is assigned to the device driver. In all subsequent data transfers only this LUN should be specified by the communicating processes issueing executive requests for sending or receiving data. The transfer request is immediately passed to the device driver by the Executive, thus no lengthy task switching is needed.

While the flexibility in the setting-up of virtual circuits is necessary in the case of the minicomputers, it is not needed for the intelligent CAMAC crate controllers, where a more or less permanent supervisory control function is provided by the subsystem computer. Therefore we have introduced the concept of static (prefixed) logical links. In this case all the allowed pairing of processes are listed in a control table and only the process name and the free or busy state of the SAP is checked by the Transport Layer in the minicomputer. If the SAP is not used for any existing connection, it is made available for the requested logical link. If the SAP is already used by another task, the connection is rejected.

The allocation of the Service Acess Points for the prefixed logical links is the task of the system manager(s). It can be done during the generation of the control tables describing the configuration of the communicating systems. Nevertheless all this is hidden from the user processes, they should request the connection in the same way in both cases.

A part of the logical link layer is implemented as a device driver, but most of the logical link control is implemented in the interface

boards. By using the assigned LUN, the Executive immediately passes the send/receive requests to the driver. The main advantage in this architecture lies in the fact that the transport layer is completely bypassed during the exchange of data. As the data acquisition processes occasionally may have to transfer a very big quantity of data in a relatively short time, it is very important to offer them the physical transfer rate with as small software overhead as is possible.

Another important gain in speed could be achieved by the implemented logical link control protocol. As data can only be sent over on existing connections, a selective flow control could be used, which works on each connection independently from the others, i.e. the buffer requirements on the individual connections do not interfere with each other. This selective flow control ensures that no data will be transmitted along a connection until a receiving buffer is available. The transfer of a message is started by a "send request" frame. If the receiving buffer is allocated already, the destination station replies with an OK, and the data are immediately sent over. This is shown schematically in Fig. 4a. If there is no receiving buffer at the Service Access Point, the destination station delays the transmission of the data with a receiver not ready response. When a receiving buffer is available, the sender will be "woken up" by a receiver ready message and the data will be immediately transmitted. This sequence is schematically shown in Fig. 4b. As the traffic control tables, as well as the buffers, are accessible for the processor of the LAN control board, the availability and the length of a receiving buffer can be tested by a few instructions. This procedure is much faster than the average bus contention time.

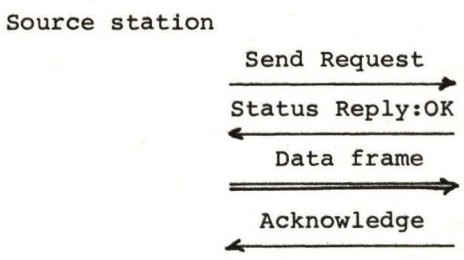


Fig. 4a. Transmission of data with receiving buffer available.

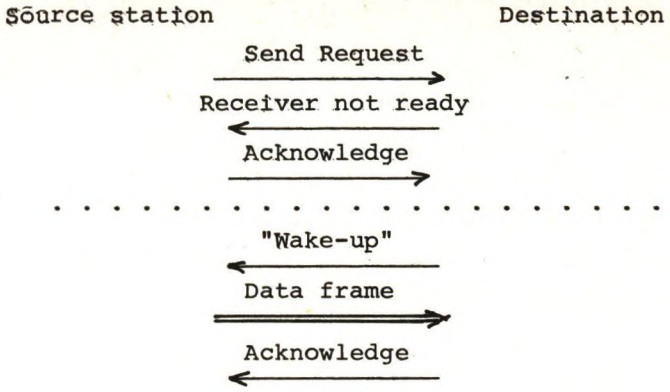


Fig. 4b. Transmission of data with receiving buffer not available immediately.

The last frame in each sequence is an acknowledge frame issued automatically by the hardware if the previous frame was received without error and the acknowledge request bit (the poll bit) was set in that frame. If this acknowledge is not received by the transmitting station within a preset time, it immediately repeats the previous frame, thus achieving a fast error recovery. (See ref. [1]). Thus after a successful transmission the message can be assumed safely placed in the buffer of the receiving task.

Error recovery by repetition is done by the driver too, and only in the case of irrecoverable transfer errors will the error indication be forwarded to the transport layer, where the logical link will be disconnected.

The direct one-to-one correspondance between the transport layer and logical link layer sevice access points has a further advantage: the selective flow control works for the transport connections too. Practically no limiting length is applicable either for the transport, or for the logical link message length (the limit is 8 Kbytes), and therefore no blocking or deblocking is needed between the transport and logical link protocol data units. Thus the LAN interface units can immediately fetch the data from the user's buffer and they can place them into the buffer of the receiving process, neither internal copy nor intermediate buffers in the memory are needed, This fact increases the speed of the data transfer between the user processes.

The efficiency of this method will be seen in section 4.

Although the logical link control SAP-s are not multiplexing ones, there is one exception. There is a special SAP, which is used by the transport layer itself for internal communications between the LOCHNESS Service Tasks. All the messages having another LOCHNESS Service task as their source or destination are transferred through this SAP. These may be service information as well as expedited data messages relayed through the transport layer.

4. MEASUREMENTS

With the measurements our aim was to obtain information concerning the data throughput between two connected hosts. In some respects these could be more important, from the point of view of the applications, than the measurements concerning the overall traffic characteristics, network utilisations, etc.

Our measurements can be divided in two groups. In the first group we wanted to determine the througput between two user process, while in the second one we tried to obtain some information concerning the speed of the file transfer operations.

Our experimental setup consisted of four TPA minicomputers, connected by a coaxial cable, running at 1 Mbits/s. During the experiments no other acticity was going on the computers.

To find the first group of measurements a program was written which exchanged mesages with an identical one running in an other node. The program provided the option to vary the length of the messages, the number of buffers and the duration of the measurement. No acknowledge window was needed, as our logical link protocol immediately acknowledges each packet [1]. The packets contained a sequence number, and the program checked for any duplicate or lost packets.

Three series of measurements were made. In the first series two processes in Nodes 1 and 2 were exchanging messages, in the second one two processes in Node 2 were exchanging messages with two other

processes on Nodes 1 and 2, resp. In the third series two independent pairs of processes exchanged messages between Nodes 1 and 2, and Nodes 3 and 4, resp. The logical setup may be seen in Fig. 5. In all of the three experiments the total throughput between the processes was measured. The results are shown in Table 1.

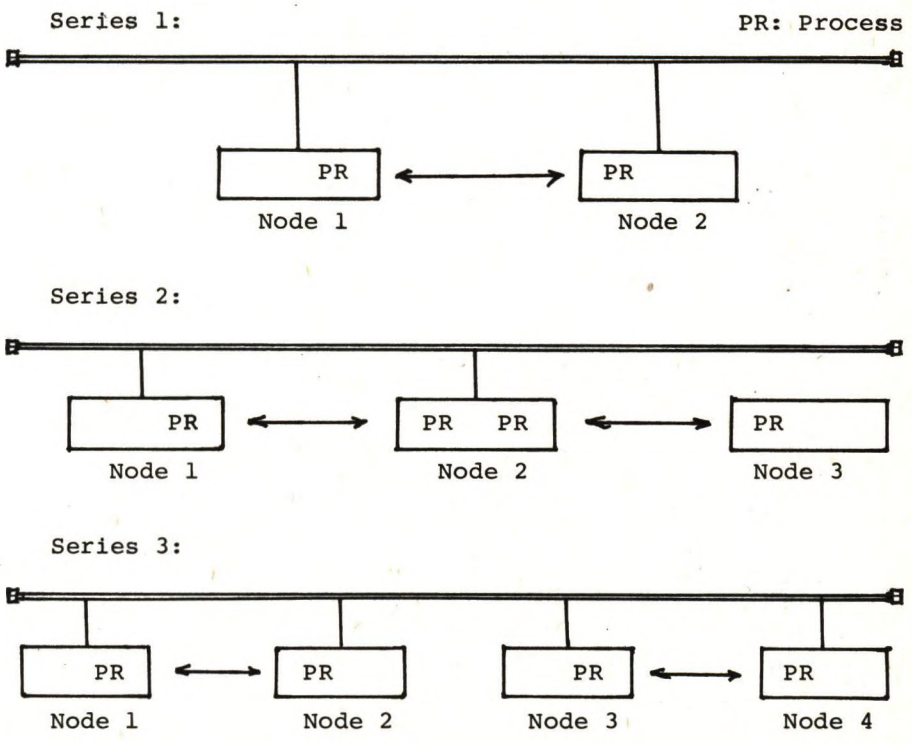


Figure 5: Experimental setup.

So it can be seen from the Table, that for a small packet length the software overhead is the main limiting fator, while for a higher packet length the bus capacity becomes the predominant one. At a message length of 512 bytes two minicomputers can utilize the cable capacity by 65%, and two independent pairs by 79%. We could not measure the packet error rate, as the hardware automatically retransmits the erroneous packets, but we could obtain an upper limit for the lost or duplicate packet rate. After eventual retransmissions this rate is less than 2.7 E-7 at 512 bytes useful data length.

This transfer rate could be compared to other available measurements. One experiment can be found in [5], where a 10 Mbits/s Ethernet cable was used. The througput between two tasks in this experiment was of the same order of magnitude as in our case. As another data reference, we have measured the throughput between two PDP-11 computers, connected by 1 Mbits/s full duplex DMR-11 intefaces. If the useful message length was 256 bytes, the throughput rate turned out to be 27720 bytes/s, roughly half of the throughput of our measurements with similar conditions (series 1).

Table 1.

Message size (bytes)	Total throughput rates (bytes/s)		
	Series 1	Series 2	Series 3
1	185.5	178.1	364.7
32	5879.8	5941.0	11719.3
256	47567		80805
512	80981	80184	99003
2048	109884	110953	117273
8192	121729	122312	123117

In the second group of measurements we compared the disk to disk transfer times within the same computer and between two different computers. We used the same file and disk copy utilities in both cases. For the file transfer to a different computer we used the utilities via the network virtual device package. The results are shown in Table 2. In the fourth experiment in the table a special network disk copy utility was used to copy a complete disk.

Table 2.

Measurement description	Total time within the same computers	Total time between two different computers	Approx. data rates bytes/s
Copy of a big file (about 1.54 Mbytes)	45 s	100 s	34133
Copy of 100 small files (about 50 Kbytes total)	116 s	128 s	500

Table 2. (continued)

Measurement description	Total time within the same computers	Total time between two different computers	Approx. data rates bytes/s
Copy of a complete cartridge disk (about 2.5 Mbytes)	92 s	186 s	13200
The same with special network utility (about 2.5 Mbytes)	-	35 s	70200

5. CONCLUSIONS

From the results of the measurements we can conclude that between two computers a 1 Mbits/s coaxial cable can offer similar throughput rates as a 10 Mbits/s Ethernet cable, the CPU speeds and the software overheads being more decisive factors than the transmission speed. A 10 Mbits/s Ethernet cable, however, can support many more simultaneous connections than a slower cable. But while the Ethernet is designed for up to several hundred nodes, in a laboratory environment, 5 - 20 nodes are connected to the local area network.

We should like to express our thanks to Mr. G. Csuka, who helped us with the measurements.

REFERENCES

[1] A. Arató, I. Sarkadi-Nagy, J. Sulyán, F. Telbisz: A Local Area Network Architecture Tailored to Laboratory Environments. Interfaces in Computing, 1, (1983), 255-262.

[2] Basic Reference Model for Open System Interconnection. ISO IS 7498, 1983.

[3] G. Juhász, K. Kovács, I. Sári: A Distributed Process Control Data Base Management System. KFKI Internal Report.

[4] C. Ware: The OSI Network Layer: Standards to Cope with the Real Word. Proceedings if IEEE. Vol. 71, No. 12, (1983), 1384-1387.

[5] F. Rake, R, Tasker and P. Kummer: Performance measurements of an Ethernet Local Area network. Interfaces in Computing, Vol. 2, (1984), 221-227.

LAN SIMULATION FOR PERFORMANCE PREDICTION AND EVALUATION

Karl H. Kellermayr

Institute of Systems Sciences
University of Linz
Linz-Austria

Due to the increasing demand being made upon distributed information processing, local area networks (LANs) have captured the interest of a host of people across the world. A LAN system designer of developer who invests in LAN architecture, implementation or realization without first addressing questions of capabilities, qualitative and quantitative aspects of alternative concepts in terms of performance analysis, prediction and evaluation, as well as system overhead analysis studies, may be in for an expensive lesson. To address these issues, one must be capable of evaluating an entire system prior to prototyping and testing the system. Computer simulation can provide powerful tools for that purpose. Such a tool is presented in this paper. Examples on how to use ist, are included.

INTRODUCTION

Since the introduction of ETHERNET in 1976 (1), LANs have been given a great amount of attention. Numerous LANs have been proposed, designed, implemented and installed. In order to achieve communication between any two devices, a number of communication tasks have to be performed at each device. These tasks have been classified into seven categories or protocol layers by the ISO OSI reference model. In order to achieve compatibility between communicating devices, standards must exist for each of the protocol layers. For local area networks, the IEEE 802 standard project spans such standard protocols for layers 1 and 2 of the ISO OSI/RM and for higher layers to an extent necessary for internetworking. LAN application areas as well as technology are moving very fast. So, even in the IEEE 802 standard project, there is no proposed single set of standards, but a family of standards. Such a family of standards provides a balance between the large number of different and incompatible networks for specific applications on the one hand and a standardized but general approach which fails to satisfy specific application and/or cost goals on the other hand. Four types of medium access technologies are defined within this family for a variety of alternative physical media. The local area networks covered by the specifications of IEEE 802 are intended to operate with transmission rates in the range of 1 to 20 Mbit/s (2). These specifications were developed primarily for office automation environments and not for heavy industrial environments or to support process control. However, for research activities such as the investigation of capabilities, performance, reliability, availability and serviceability aspects, they can be used as an excellent framework for the very broad range of potential LAN application areas. Figure 1 shows the relationship among the six documents of IEEE 802.

LAN SIMULATION

Today there exist much more than 100 different LANs. To investigate their performance characteristics as well as other important aspects, there are quite a lot of specific models available. However, to my knowledge, there does not exist a

comprehensive approach for the overall prediction and evaluation of system aspects of different design alternatives under realistic application assumptions.

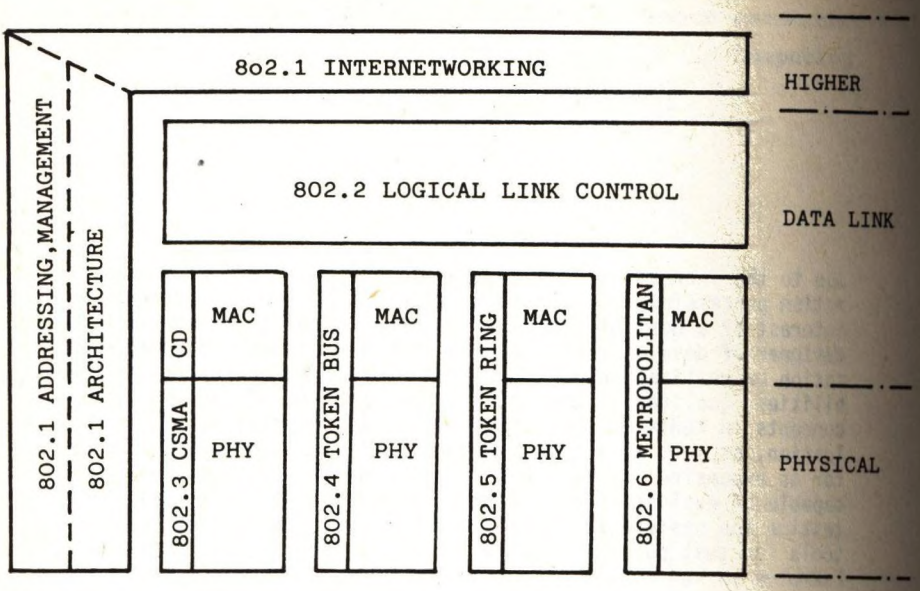


Figure 1: IEEE 802 Standard Architecture

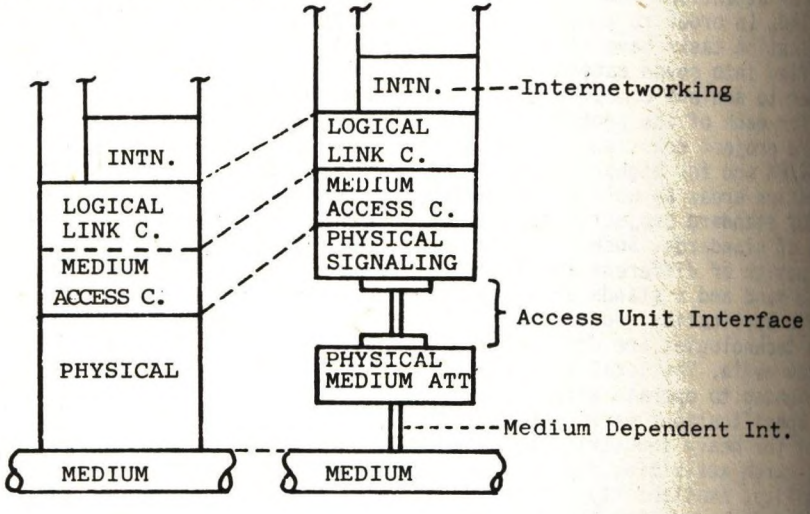


Figure 2: IEEE 802 Architectural and Implementation Reference Model

The Department of Systems Sciences at the University of Linz is developing and refining such an approach. This work was started in order to support the development of a performance optimal star shaped optical bus system. Simulation results for that system as well as the systems engineering based design approach for the simulation model were presented in several papers (4,3). The intention of our recent work is to extend this simulation model to a more general evaluation and performance prediction tool. In order to have a common base for the different design alternatives ranging from ETHERNET, different CSMA, TOKEN, bus or ring and PABX type systems, we use the IEEE 802 architectural reference model and implementation reference model as our architectural frame(figure 2).

LOCAL AREA NETWORK PERFORMANCE EVALUATION AND PREDICTION: LAN-PEP

In order to evaluate or to predict the performance of a LAN, one must be able to experiment with it. In the situation where the systems do not exist, the experiments must be performed with a model which represents the systems to the degree, that the evaluator may perform the desired experiments and extract the desired results. However the development of models for complex systems such as LANs is a difficult task and thus it is costly to design, validate, implement, realize and analyse them. Frequently their development is considered more an art than a scientific or engineering process. In order to overcome some of the well-known problems in this area by providing a scientifically founded operational basis we have chosen a systems engineering-based approach (3). We are pursuing both the development and implementation of models and their application as tools for optimal LAN design (parametric systems-engineering), for the analysis of optimal LAN selection and/or the comparison of different existing or proposed LAN systems.

The tool is designed as an open tool and its development is still on its way. It is named Local Area Network -Performance Evaluation and Prediction: LAN-PEP. Two different interfaces are maintained by LAN-PEP as shown in Figure 3: one to provide assistance to the user of the tool and the other to the implementer of new concepts (physical media, topologies, protocols). So far, we have already implemented quite a broad spectrum of LAN physical media (fiber optic cables, twisted pair, coaxial cable), topologies (bus, star, ring) and network protocols (different versions of CSMA/CD, TDMA, 3rd Generation PABX schemes, slotted ring, register insertion ring, TOKEN passing).

One of the most important questions in the process of designing and implementing the tool is how to develop a level of confidence in prediction i.e. model validation. This was done by first developing the models to represent existing systems and validate the results of the model by comparing them to measured or well accepted results from the actual system. Once the level of confidence is developed, the user can make alterations in the model, perform experiments and predict the performance of the actual systems if the same alterations were done to it.

As reference systems we have so far implemented the following LANs:

ETHERNET
CAMBRIDGE RING
IBM TOKEN RING
HASLER SILK
3rd Generation PABX

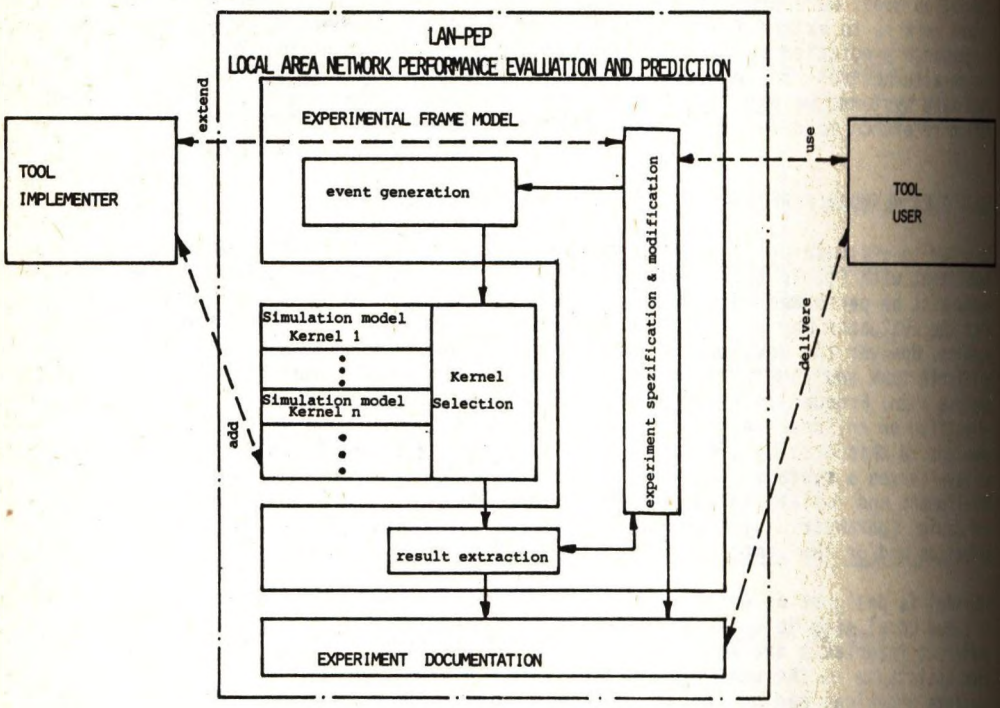


Figure 3: The LAN-Performance evaluation and Prediction tool LAN-PEP

SOME SIMULATION EXPERIMENTS

Now some results of simulation experiments will be presented, in order to demonstrate the application area of LAN-PEP.

Experiment 1: Data link layer performance comparison of different protocols under rising traffic load

Experiment 2: Different protocols under heavy traffic load and with different resolution levels for result extraction

Experiment 3: CSMA/CD and Token protocol under rising data transmission rate

EXPERIMENT 1:

This experiment is done for a star like optical bus medium. Four different medium access protocols are compared: two stochastic contention protocols, one collision free deterministic and one adaptive limited contention protocol. These protocols are described in some detail in (5). Traffic is being raised by keeping packet length constant, but decreasing the mean interpacket arrival time from 100 to 1 ms (traffic is generated with negative exponential distribution). In Figure 4 a diagram of the maximum waiting time is presented for this experiment. CSMA-SP is a very simple CSMA procedure. For low traffic loads waiting time is very good, but as traffic increases, waiting time escalates due to a very poor collision recovery mechanism within this protocol.

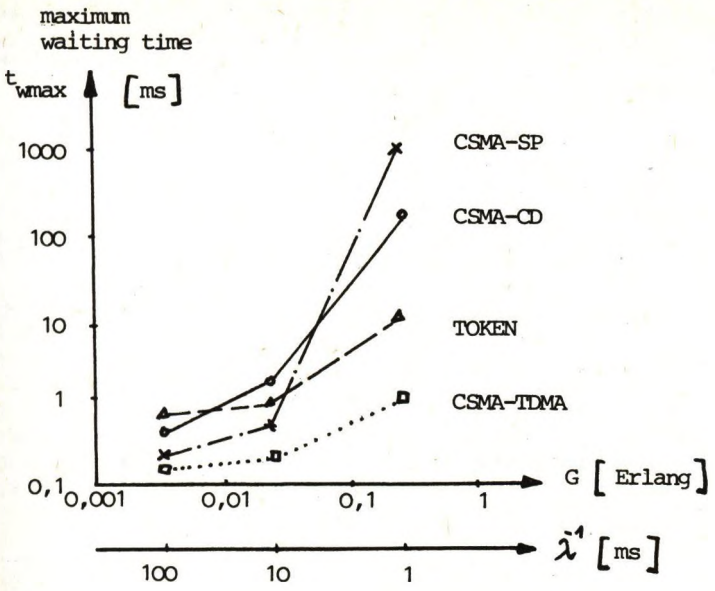


Figure 4: Comparison of the maximum waiting time under raising traffic load

The rather complicated limited contention protocol CSMA-TDMA is a combination of a CSMA and a deterministic TDMA protocol. It is a very efficient protocol which takes advantage of good performance of CSMA under low traffic load and the good performance of TDMA under high traffic load. Figure 5 represents the channel efficiency as function of traffic load for that experiment. One can see very clearly, that CSMA-SP rapidly becomes inefficient if traffic load exceeds a certain limit.

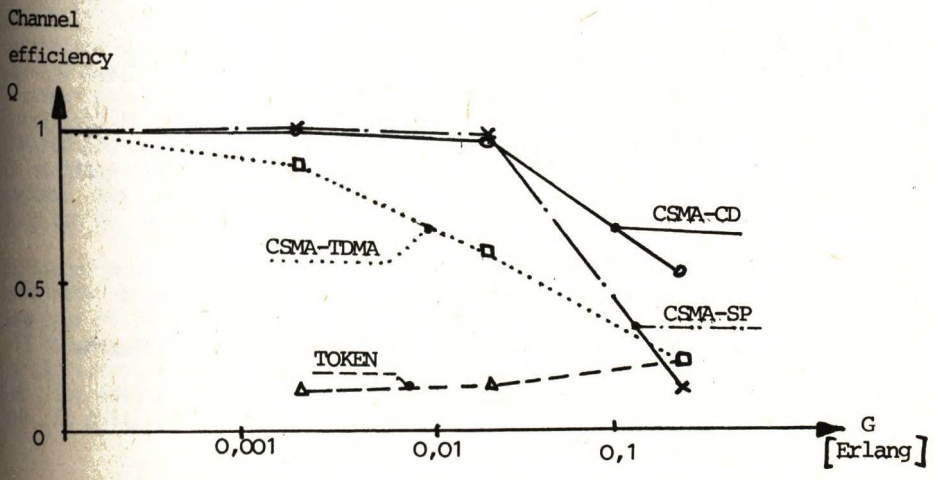


Figure 5: Channel efficiency

Channel efficiency of stochastic contention protocols and collision free deterministic protocols can not be defined in a consistent way, so comparison of both is not useful. However, one can see in figure 5 very clearly that channel efficiency improves if traffic load is raised for the TOKEN protocol. This performance is directly opposite that of the CSMA type of protocols.

EXPERIMENT 2:

In this experiment two media with different MAC protocols are considered. The one is a coaxial with ETHERNET protocol and 10 Mbit/s datarate, the other is the star-like optical bus and the protocols are as in Experiment 1 (datarate 16 Mbit/s). (This experiment is discussed in great detail in (5).) 16 stations are connected to the medium. 8 of them have rather low terminal dialogue type of data traffic, 4 of them have uniformly distributed real time function computer traffic, 1 station is considered a function diagnostic computer generating data with constant packet separation, and 3 stations generate bursty behavior by causing a brief flooding of the bus about once every second with three times its capacity (like a file server). The performance evaluation, it must be noted, depends strongly on the resolution level of result definition (interpretation). Experiment 1 shows, that the rather complicated limited contention protocol CSMA-TDMA has very good performance for both low and high traffic. In figure 6 one can see, that the total system performance of CSMA-TDMA is not good at all for this example, wherein the bus is flooded for short periods of time.

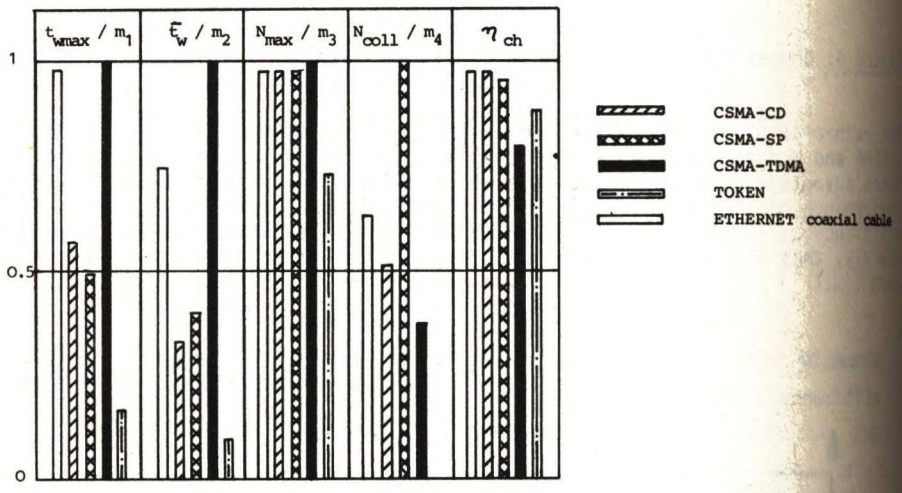


Figure 6: Total simulation results for the optical bus.

t_{wmax}...maximum waiting time ; $\bar{t}_w$...mean waiting time

N_{max}....maximum queue length ; N_{coll}...number of collisions

η_{ch}....channel efficiency

m_1=598,6 ms ; m_2=49,7 ms ; m_3=99 ; m_4=34089

But is total system performance important, or is it more useful in this situation to consider performance at a more refined resolution level. For example, to consider performance for each of the classes of stations connected to the system. Figure 7 is a diagram showing maximum waiting times in the four station classes. We can easily see that with this refined observation, the situation looks different. CSMA-TDMA shows that the 3 bursty stations do not adversely effect the others, so it can be considered very good. ETHERNET, as designed to be a "fair net", does not show good results. Overall delay is the price for fairness.

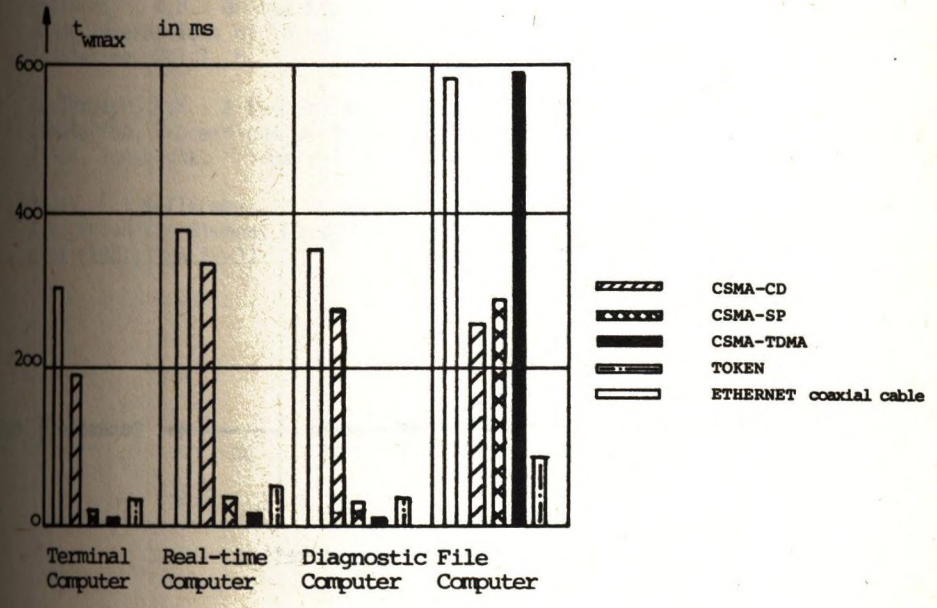


Figure 7: Comparison of maximum waiting times t_{wmax} in the four station classes

EXPERIMENT 3:
With this experiment I would like to point to a problem in comparing stochastic contention and deterministic protocols. While for the first class of protocols packets may be lost if the number of collisions is to high (depending on collision recovery procedure), this may not be the case in the second class. This has to be considered if one compares such protocols. In this experiment an ETHERNET like CSMA/CD protocol is compared with a TOKEN protocol on a star like bus system. 10 stations offer the negative exponential distributed data traffic load to the systems, and data rate is raised from 5 to 20 Mbit/s. For 5 Mbit/s the traffic load can be considered as very high.

In figure 8 one can see that for the CSMA/CD protocol the number of unsuccessfully terminated packets is very high for 5 Mbit/s but decreases, so that it gets 0 in the area of 20 Mbit/s. Obviously, for the TOKEN protocol the number remains 0 for all the data rates. In figure 9 the maximal waiting times are presented. An adequate comparison of the performance on data link layer for these protocols should be made only for adequate behavioure, which means in this experiments for data rates higher than 20 Mbit/s.

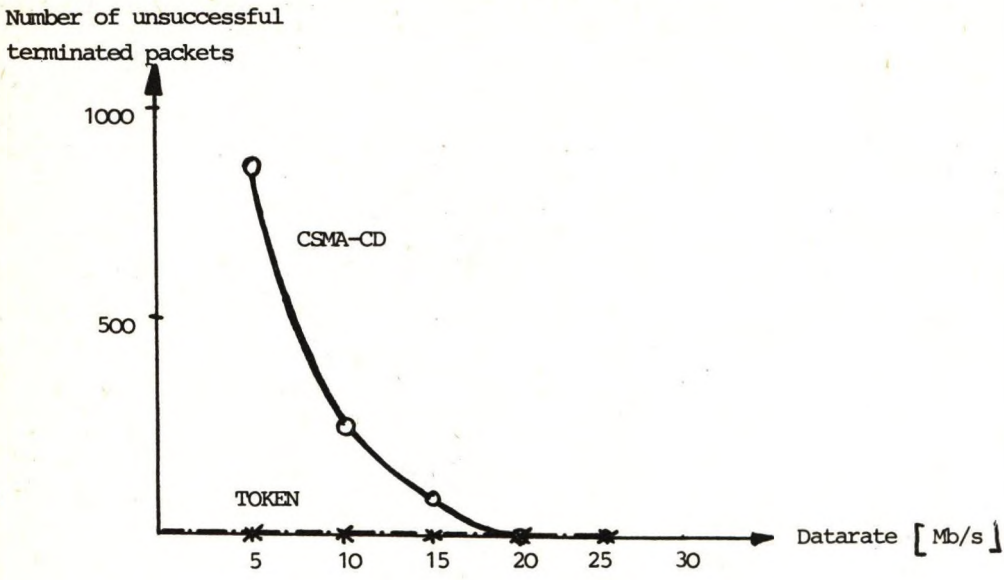


Figure 8: Number of unsuccessfully terminated packets

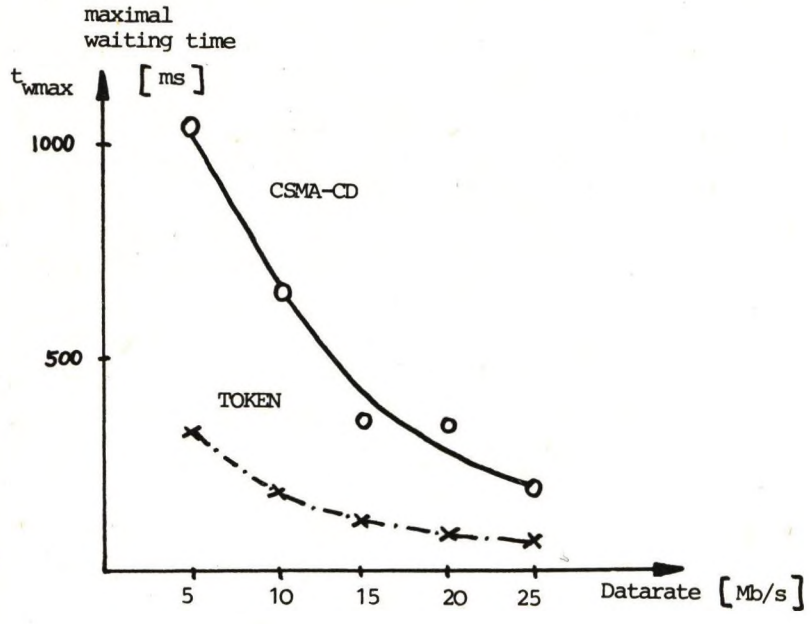


Figure 9: Maximal waiting time

REFERENCES

(1) Shoch, J.F. et al: Evolution of the ETHERNET Local Area Computer Network, IEEE Computer (August 1982), pp 10-27

(2) IEEE Project 802: Local and Metropolitan Area Network Standard, Draft IEEE Standard 802.1 (Part 1) - Overview and Architecture, Revision B (June 1983)

(3) Kellermayr, K.H.: On the Simulation of Local Area Networks: Mutual Performance Interactions of Integrated Services, Systems Analysis Modelling Simulation, Vol.1, No.5 (1984), pp 309-409

(4) Kellermayr, K.H.: A Systems Engineering Approach to Local Area Network Simulation, Proceedings of the Second IASTED International Symposium, AI'84, Innsbruck, February 14-17, 1984

(5) Baues, P., Kellermayr, K.H. et al: Simulation of Access Methods in Local Area Networks, Siemens Forschungs- und Entwicklungsberichte, Band 12, No.1 (1983), pp 28-33

THE EXLOC LOCAL AREA NETWORK

Balázs A., Gáll M., Jákfalvi R., Sugár P., Szentner J.

Department of Computer Communications
Videoton Development Institute
1021, Budapest, Vöröshadsereg útja 54.
Hungary

The paper presents a local area network, called EXLOC, developed in the Videoton to support distributed processing, office automation and CAD/CAM. The EXLOC system is based on the Videoton's lately developed microcomputers, the VT32s, and their UNIX compatible operating system. The main goals and considerations of the network are described. Some experiences and future plans are also presented.

INTRODUCTION

EXLOC (Experimental Local Area Network) is the name of the VIDEOTON's local area network, aimed to integrate first of all the VIDEOTON's VT32 microcomputers. Though the name EXLOC comes from our original experimental puposes, by now the system has matured to be a real product. The paper presents the main considerations of the development, the main goals and characteristics of that, and our plans.

GOALS

a) A communication vehicle of high speed and short distance for VT32.

VT32 is a 16/32 bit microcomputer based on a microprocessor compatible to M68000. It has an operating system uSOS, which has a user interface compatible to UNIX V.7. Thus, VT32 can theoretically host all the UNIX V.7. application programs. Since VT32 has been developed to support mainly office automation and CAD/CAM applications, besides the standard uSOS/UNIX applications, the high speed/short distance approach seemed a reasonable goal for the communication vehicle to be developed.

b) Transparency to the application programs

We believe that, besides the sharing of the network-wide resources, it is an advantage, that is mostly appreciated by network users. It means that users need not rewrite their application programs already available for a single machine, because of the introduction of the new network environment. Interfaces remain the same, application programs don't bother themselves with the problem if they execute in a single machine, or in a network environment with distributed resources possibly referenced by them.

d) Transparency to the Kernel

It means that a new verison of the uSOS (UNIX) Kernel theoretically does not imply any modification in the network software. It can result a certain level of portability.

e) Readyness to any possible future modifications and extensions regarding both the range of the machines connected and the communication medias.

THE EXLOC APPROACH

To meet the requirements detailed in the previous section, we followed the main lines below.

a) For the high speed/short distance communcation, we have taken the Ethernet local area network approach. It is particularly suitable for office and CAD/CAM applications, it has been standardized well with supporting chips available from several manufacturers.

b) For the uSOS/UNIX environment, it seemed reasonable to follow the UNIX concept, i.e. the way "to see the world, as the UNIX does it". It will be a bit more detailed in the next paragraphs.

c) For the transparency to the application programs, the protocol architectture should include the highest level protocols too. Considering the poor interprocess communications vehicle of the UNIX V. 7., unlike many other network designers, we did not wished to develope a global network-wide interprocess communication. Instead, we have introduced the Network Directory Tree and the Network File concepts, taking the elegancy and the generality of the UNIX file concept into account. UNIX considers any kind of resources (data file, program, I/O device) as files and provides the same mechanism to access them. In this way, UNIX does inherently provide application processes a certain level of transparency to resources. UNIX has a file-directory tree structure, which contains a root directory, several subdirectories (non terminals) and files as leaves (Fig. 1). Each directory or file can be accessed from the root-directory tracking the path from the root (or the current directory) to it, i.e. by identifying the path name.

We have defined the Network Directory Tree as the union of the directory trees of the different machines and by creating an extra Network Root Directory. The new root is connected to each directory node which was earlier a local root for a particular machine related (Fig. 2). These particular directories represent their related machines in the network, indeed. Thus, every particular file (resource) of the network can be accessed by any application process much the same way as in a single machine environment. The only difference is that the path name is related to the global Network Directory Tree.

d) For the transparency to the Kernel, we have decided to develop most modules of the EXLOC software outside the Kernel, as user level processes. In fact, these processes can be considered as new UNIX command utilities, thus at this point we have also followed the UNIX concept.

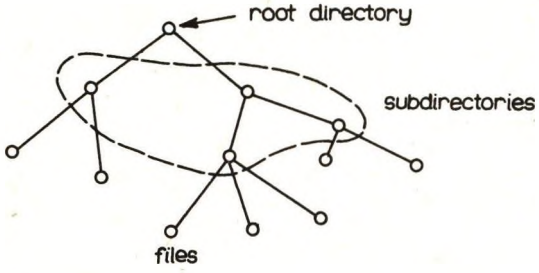


Figure 1.

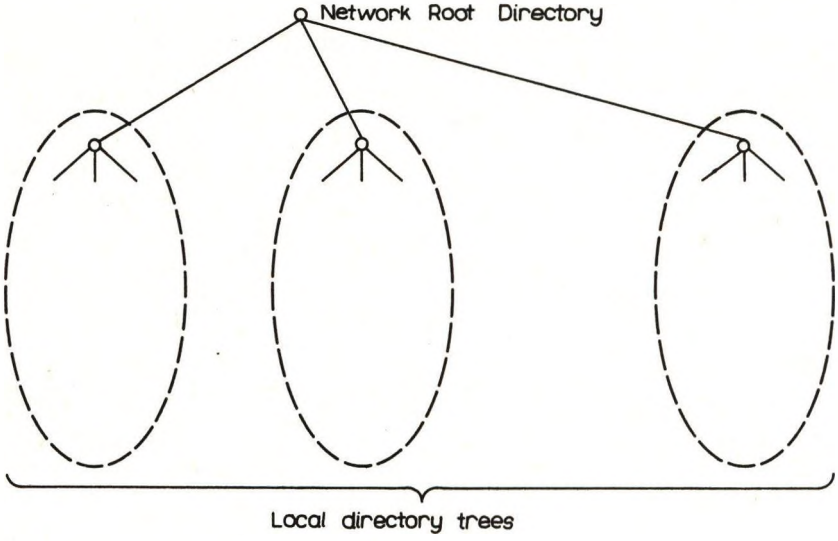


Figure 2.

e) For the openness, we have followed the ISO OSI reference model with its layered structure. For the network and transport layers, we have taken the Xerox Network System (XNS) approach [1] with some modifications in its protocols. Besides, we are sure that in the future some more appropriate new protocols could be applied either.

PROTOCOL ARCHITECTURE

The Global File Access Protocol (GFAP) program takes care the Network Directory Tree. It captures all the file access commands of the application processes. In the case of an open file command, it decides if the file related is remote or local. If it is local, then the GFAP passes the request to the local Kernel. Else, it initiates a session to a remote GFAP program, which in turn controls all the incoming access requests to the files local to that. During normal resource usages, GFAP is responsible for the consistency of the file system, by detecting any possible abort condition of its partner and handling such situations.

The Service Connection Protocol (SCP) controls the cooperation between processes wishing to use remote resources and processes advertising their services. In general, we consider an EXLOC network as a set of network wide services, like print service, terminal emulation service, file store service, global file access service, etc., and a set of service users. Our approach to their connection is shown in Figure 4. A particular service in a machine is represented by its Service Listener process (SLP). An SLP can declare or clear the associated service to the network service registration, which itself is a service too. After SLP opened the service at its local machine, it can accept any incoming "connect to server" requests related to the service represented from remote user processes. It can check some service specific attributes of the request (e.g. access right, version) and it may refuse or accept that. In the latter case, SLP creates a server process representing the particular remote user process at the local machine further on. Afterwards, that particular user-server connection will realize the related service usage. As soon as the connection is closed, the related server process terminates.

The Reliable Datagram Transport Protocol (RDTP), like the XNS Sequenced Packet Protocol, provides the correct, sequenced delivery of packets, the protection against duplicates and packets from inappropriate sources. It also provides fragmentation into 543 bytes blocks and flow control. Since it does not deal with connections, it can be considered an acknowledged connectionless protocol.

The Internetwork Datagram Protocol (IDP) is a slightly reduced version of that of the XNS. Its function is to address, route and deliver standard internet packets. It gives its best effort to deliver them, but it cannot guarantee that they are delivered once and only once, or their correct sequence. It uses the standard XNS IDP packet format with the address triple of the network address (32 bits), the host number (48 bits) and the socket number (16 bits). Sockets are bidirectional structures, able both to send and receive internet packets at the same addresses. Certain socket numbers are considered well-known, that is, they are allocated statically to the network service, or more precisely to their related Service Listener Process. Thus, in the current EXLOC network we have allocated the

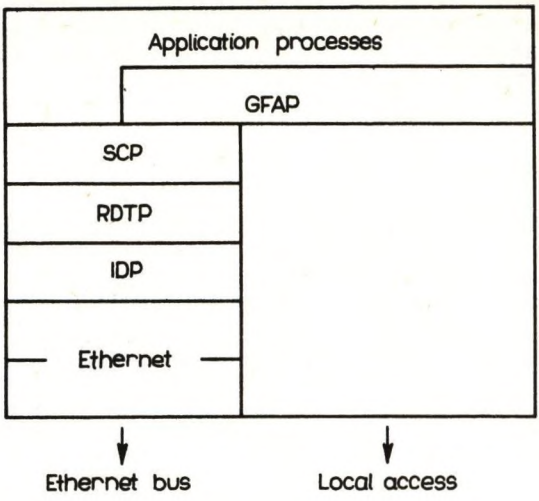


Figure 3.

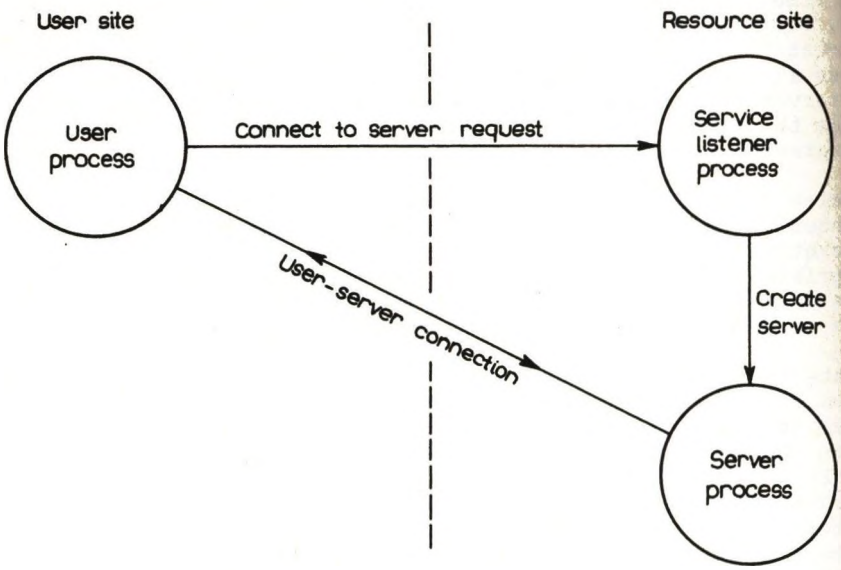


Figure 4.

Network Service Registration SLP and the Global File Access SLP the well known sockets 4 and 5, respectively.

The two lower layers have been realized in our VT32 Ethernet controller of a single board connected to the VME compatible bus of the VT32.

EXPERIENCES

Our first system was in experimental stage in 1984. It had three VT32 machnies connected to a network cable of 25 meters. Each VT32 had a number of interactive terminals and floppy or winchester disk units. For the first version, only the Global File Access service had been implemented; since first we wanted to address the problem of the transparency to the application programs in homogeneous uSOS/UNIX environment. For the time being, the Network Service Registration and the IDP routing function had been implemented in a minimal level.

The network software needs 60-70 Kbytes in core on average, but its size depends on the number of current services and user-server connections. Recently, we deal with the integration of the shell and the uSOS/UNIX utilities. It is possible, since they execute in the form of application level processes too, including the shell, which interprets the users' interactive commands. Thus, the user, who wishes to process file A by a program utility B and to output the result into file C, need to know neither at which machines the files A, C and the program (file) B reside, respectively, nor the fact that he is working with a network. He need simply type his commands in his terminal in the same way as in a single machine environment. The ability to execute utilities on another machine is also significant, because it allows there to be just one copy of any such program in the entire network, thus avoiding the duplication of system utilites in the network and freeing disk space. Since most UNIX systems needs about 8 megabytes for the full set of UNIX utilities, a system of 20 machines with 80% of all utilities stored on a single machine would result in a saving of 128 megabytes on disks in the network.

There is under development an Electronic Mail application, too, following the CCITT MHS standard in some respect. It is also a single machine program which can be applied both in a single machine or in network environment.

FUTURE PLANS

To reduce the overhead, we plan to develop a programmable Ethernet front-end controller, which could realize most parts of our protocol architecture. We also plan to integrate Videoton's megamini machines, called R11. Some new services are also under development, as for instance the IBM 3276 Terminal Emulation Service (TES), by the means of which IBM mainframes could be accessed from the network.

REFERENCES

[1] Internet Transport Protocols - Xerox System Integration Standard (Xerox Corporation, Stamford, XSOS 028112, Dec. 1981).

DEVELOPMENT OF DISTRIBUTED APPLICATIONS FOR EXPERIMENT AND EDUCATION SUPPORT BASED ON THE NETEX LOCAL AREA NETWORK

Leszek Borzemski, Adam Grzech, Andrzej Kasprzak, Leszek Koszałka

Institute of Control and Systems Engineering
Technical University of Wrocław
50-370 Wrocław, Poland

In the paper we are interested in the qualitative and quantitative statements of the requirements analysis of the local area network applications which are met in the university environment. In local area network called NETEX being under development in the Technical University of Wrocław, the following requirements are taken into account: service availability, human interface, user-tolerant and user convenient architecture, personal computer use, distributed data processing and conformance to standards. Based on the requirements analysis, the development of the network is presented.

INTRODUCTION

The planning and designing of any computer system is always confronted with sets of often conflicting factors and requirements, and a host of increasing technological alternatives, therefore a disciplined approach to define the problem and obtain a solution must be applied. In such approach, the greatest amount of effort is concerned with the requirements analysis, directed at extracting of consistently defined requirements from the users' community. The objective of the requirements analysis is to codify the needs and desires for a system in a form useful for designers and providers of network and network services. Needs are conditions that must be satisfied regardless of costs, while desires are conditions that may be satisfied based on some arbitrarily chosen criteria.

In the requirements analysis, two types of needs and desires may be identified, i.e., qualitative and quantitative statements. The former provide scope and direction to the development of a solution space, while the latter govern the development of a solution space. The network planning is based on planners' and users' cooperation. Network planners interact with their customer, explaining to the customers the planners view of the problem, and being informed by the customer what realities must be considered in the planning process.

In our case, the planner is at the same time the user, so the understanding of the requirements is reached and is based on past experience in defining some of our information processing needs or applications. Performing both of activities mentioned above gives some handicap in the network development.

USER REQUIREMENTS IN THE UNIVERSITY ENVIRONMENT

In defining our applications, we emphasize the characteristics of the activities that tend to differentiate the following applications education, research, engineering design and local networking development. Aforenamed applications are the most important among those we are interested in.

The first application, education support, deals with students' education in automatic control and computer systems areas. Problem-oriented and complex education in the areas requires to have the following possibilities (activities);

- data collecting (from real-time and simulation processes),
- data converting (a/d and d/a converting),
- data transmission (from the data sources to the data destinations),
- distributed disconnected and distributed hierarchical data processing (processing of previously collected real-time and/or simulation data using the preestablished and known procedures),
- programs and data storing (data bases and program libraries for specialized and general procedures and programs), and
- distributed simulation.

Based on the listed above possibilities, the student education may be organized and performed in three following steps:

- education problem analysis and acquintance with resources dedicated to problem solving,
- application of known resources to solve some preestablished education problem (within the boundaries of own resources), and
- development of new education resources (based on available activities).

The second application is scientific research. Since the education and research are (or should be) closely related, the requirements for both of them are similar. What distinguishes them, is the different way to ask questions: what kind of education problem can be solved based on available resources whereas what kind of resources are desired to solve determined research problem. Because the question referring to resources is equivalent to the question about available computer system activities, only some basic research requirements can be determined a priori. It is also evident, that the amount of computer system resources desired in solving some research problems may be different to that used in education problems. It must be noted, that the research desires should create some new needs, after that they may be implemented in improved education support.

The third application is computer-aided engineering design. This application is characterized by the following activities, that should be available:

- data storing (libraries of standards and large data bases, mostly specialized),
- program storing (program libraries),
- distributed data processing, and
- remote job entry type traffic.

The respectively large demand of available procedures, large programs and standard libraries, distributed simultaneous processing

and well-organized management are characteristic for this application. The CAD process is based on performing some preestab-*lished procedures in the boundaries given by standing standards.* The amount of input data is relatively small, but the data are multiple processed in specific to (driven by gain of design) designing process manner. Moreover, the designing procedures are relatively simple, the performing designing process requires the procedures to be well-organized and synchronized. The next important need is the output form of the designing results which must follow the previously determined formats.

In the first three applications, the user is interested in the problem formulation (in some language) and in the data output; so, he treats the computer system as a "block box".

In the following, fourth application, the user requirements are, in the greater part, different - the computer system (local area network) is the subject of education, research and development. Then the application is the local networking development as a result of rapid changes in the implications of distributed processing with mini/microcomputers and automated data processing network structures. The application covers two main areas of interest. The first is the education, and the second is research and development.

Local area networks using low-cost, high-speed transmission capability as the basis for a general-purpose data transfer network seems to be very attractive tool in universities environments. Education and research interests are directed into the following problems:

- network control structures and access methods,
- protocols for local area networks,
- local area network performance measures (performance requirements),
- application of local area networks, and
- interconnection of local area networks with other (local area and/or wide area) networks.

The gain of education of local area network is to understand the advantages resulting from integrating the idea of distribution and communication as well as the study of the way of doing it. The main interest in local area network development is directed into allowing the existing and anticipated users needs and desires to be met. Local networking development is conditioned by extended monitoring and testing systems within existing local area network and nested network operation program structure.

THE SYNTHESIS OF USERS REQUIREMENTS

In the previous chapter, four basic applications, the most important at the Institute of Control and Systems Engineering, have been presented. The framework of the requirements analysis is defined by the scope of applications to be considered and the timeframe over which the applications are to be considered.
The requirements are functional, and at the same time generic in form. They are related to specific university applications among which the education and research (including the local networking development) are the most important. The needs that must be satisfied give also possibilities to meet some users desires (for example: administration supporting).
The collected requirements include known and some anticipated needs and desires. What is characteristic in our approach is that the

process of requirements collecting is in progress, i.e., all known needs are always a subset of anticipated needs and desires.
The requirements collection is not following any codification and has no preestablished procedures.
The requirements analysis realized in our Institute for development purposes is not limited by any timeframe. Among the requirements, some of them will be distinguished to be satisfied in the first step of network development. The experiences coupled during realization of the first step may influence the framework and the requirements subset selected to be satisfied in the next steps.

The framework of requirements analysis and requirements analysis itself are the basis for functional description of the possible automated data processing (ADP) within the computer system. The rapid advances in computer technology and the requirements analysis enable to recognize that the distributed processing with mini and microcomputers and ADP network structure will be the best solution to satisfy the requirements.

The main functional characteristics of desired ADP system are as follows:

1. Service flexibility - accommodation of a variety of service types, ability to access multiple data sources and possibility to improve and expand the services,
2. Human interface - ability to interact at different levels with the user by means of user-oriented language to meet individual requirements,
3. User tolerant and convenient architecture - providing the users with the network as a "black box" as well as the system of several accessible and distinguishable components,
4. Personal computer use - satisfying urgent needs for own versatile processing and storage capabilities,
5. Distributed disconnected and distributed hierarchical data processing - possibility to provide both functional alternatives, each serving in different managerial, technological and economical environments as proposals against centralized support in rather local geographical scope spanning,
6. Conformance to standards - following this feature, the data processing may present well-suited, validated and internationally proven solutions.

Other ADP characteristics can be also stated in technological and economical categories taking into account the cost, technology state of arts and benefits. Performance factors should be deeply synthetized for ADP system, as well. All above identify a profile for the ADP system, which gives needed /desired principal system characteristics.

There are many reasons to develop a local area network for the applications characterized in this paper. The principal characteristics of such network called NETEX (NETwork for EXperimentation) are presented in the next chapter, whereas the other sections describe the hardware and system architecture, focussing on some components designed to meet user requirements.

THE PRINCIPAL CHARACTERISTICS OF LAN NETEX

Taking into account the requirements analysis and its framework, the quantitative statements which govern development of a solution space

should be formulated in terms of environment definition, traffic characteristics, performance requirements and interface requirement

The environmental requirements deal with identifying "what" is "where" and seek a precise characterization of network nodes. In our case heterogenity of personal computer use is assumed. Two kinds of network nodes - workstations and resource nodes (servers)- may appear to be nodes of interest, realized on mini and micro-computers. The development of LAN is based on internationally accepted standards which will allow in the future to connect our network with other LAN's and wide area network MSK (Interuniversity Computer Network).

The traffic characteristics describing the stochastic nature of the information to be exchanged can be determined after some experience Even though, the different types of the anticipated traffic could be distinguished, the message types, message lenght distributions, traffic origination and destination patterns and inter-arrival time distributions can be obtained after implementing the applications. The simulation study cannot substitute this. Knowing the importance of this item while the real characteristics are not available yet, in the first stop of network design we assume traffic characteristics known from the literature for the same and similar applications. The performance requirements, specifying grade of service of the network, depend on many factors, including the technology, concept and design. In our case the expected performance requirements are the following:

- reliability/ availability - achieved by consequent application of proven design methods and by defining the maximum amount of time a user could wait for an unavailable network to become available,
- time characteristics - achieved by defining the network delay and response time for number of distinct application traffic types such as remote job entry, interactive, transaction-oriented etc.

HARDWARE AND SOFTWARE OF THE NETEX LOCAL AREA NETWORK

Network hardware

NETEX is a small CSMA/CD bus network developed at the Institute of Control and Systems Engineering, the Technical University of Wrocła for connecting a number of PDP-like systems and 8-bits, 16-bits microcomputers together for offering several services in networking as well as for studying local network protocols and distributed systems architecture, especially database architectures.

The NETEX LAN offers an integrated possible low-cost way of carryin many different network services. In the design the emphasis has bee concentrated on availability of a wide service spectrum, low hardware cost, low entry cost and wide range of expansion in hardware and services.

The Reference Model for Open Systems Architecture (ISO OSI/RM) (Day and Zimmerman (1983)) was our framework in designing of the network. Considering the results obtained in local network protocols standarization, the IEEE project 802 (1981) has been chosen as a basis in constructing physical and link layers.

The components of the NETEX network segment are: - coaxial cable

(50 ohms) as the transmission medium, - terminators at both ends, - taps to connect the stations to the cable without cutting, - transceivers which contain line drivers, line receivers and collision detection mechanism.

The basic component is the Controller which has two Z80A microprocessor boards. The first board performs the framing, bit decoding encoding, address comparison, CRC calculation and checking, the contention resolution algorithm and has four frame buffers (one for sending and three for receiving frames). The carrier sense multiple access with collision detection mechanism (CSMA/CD) is used as the means by which stations share a common bus transmission medium. Transmission is unmodulated baseband with the transmission speed 1 Mbits/s. The size of frame can be changed and takes the size of user data of 512, 1024 and 1518 bytes. The second board is an adapter to the micro- or minicomputer through the RS 232C interface (V24) with up to 50 Kbits/s and it also provides the functions of the Logical Link Control (LLC) sublayer in the context of both connection-oriented and connectionless services.

Service primitives are of the same generic types as described in the IEEE LAN Reference Model, namely: requests (R), indications (I) and confirmations (C). The service primitives associated with the unacknowledged connectionless transmission procedure are DATA-R and DATA-I. The first is for data transmission among remote stations, exactly to the Link Service Access Point (LSAP). DATA-I indicates the arrival of data at local LSAP from the specified remote LSAP. Seven pairs of service primitives are associated with the connection oriented service: CONNECT-R and CONNECT-I, DATA-CONNECT-R and DATA-CONNECT-I, DISCONNECT-R and DISCONNECT-I, RESET-R and RESET-I, RECEIVE-READY-R and RECEIVE-READY-I, RECEIVE-NOT-READY-R and RECEIVE-NOT-READY-I, CONFIRMATION-C and REFUSAL-C. The connection is established after issuing CONNECTION-R, and then by means of DATA-CONNECT-R the data can be transmitted over an existing connection. The connection can be RESETed or DISCONNECTed. RECEIVE-READY-R informs the LLC Layer and remote LSAP that the local LSAP is ready for data receiving. CONFIRMATION is to inform the local LSAP when the remote station sends positive acknowledgment concerning an action started at this local LSAP, that is when connection is distinguished, restarted or terminated, or data transmission is completed. REFUSAL is sent to local LSAP when negative acknowledgment comes.

Network configuration

The configuration of the NETEX which is being developed (shown in Fig.1), includes the stations (workstations) and the server stations (servers or resource nodes). The first are micro- or minicomputers which provide general capabilities in processing, data manipulation and remote access for the users communicating with the network through the devices of that station. The servers are machines (or subsystems) designed to provide a specific, well-defined service to the users, located at the place or at remote stations. Among the stations are two minicomputers SM4 (PDP 11 like), serving also as a remote job entry processors, and several microcomputers. Our first effort has been concentrated to connect 8-bits CP/M based microcomputers involving Intel 8080, Intel 8085 and Z80A microprocessors. However, as additional 16-bits microcomputers become available at the university, they will be also included into the network.

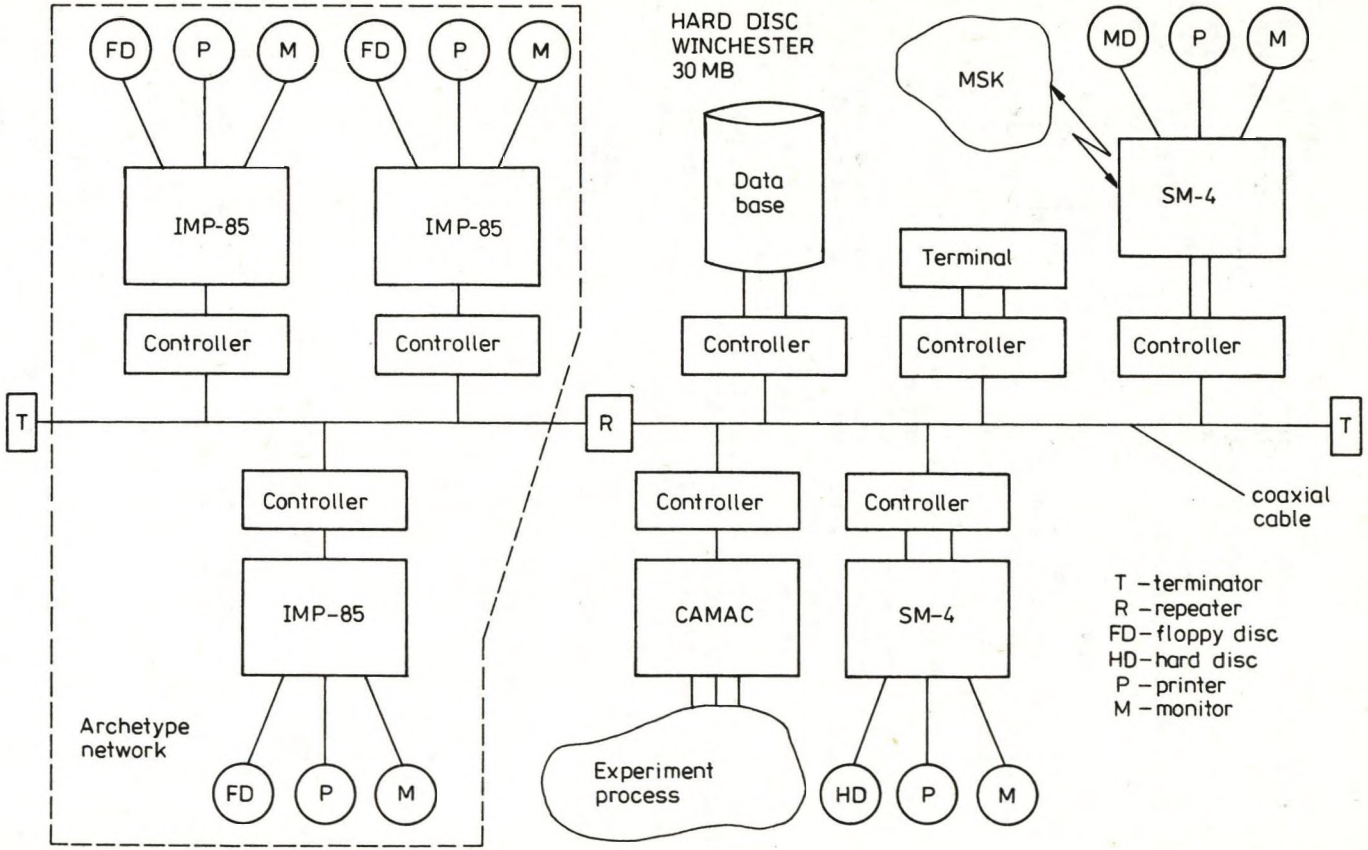


Figure 1

Structure of local area network NETEX

- CAMAC server supporting data acquisition, local data processing and storage, and object actuation at the Control Systems Laboratory,
- Database server maintaing common back-end relational database with directory and accounting system,
- Printing server capable of bulk printouts,

The CAMAC is also working under individual control of the operating system of SM-4 and is not considered as a common network service but as a device of that minicomputer.
In the future development, the connection between NETEX and MSK will be provided via gateway based on microprocessor board or one of SM-4 minicomputer.

System architecture

The conformance of the NETEX architecture to the ISO OSI/RM is presented in Fig.2. The main feature which characterizes the way how the architecture is implemented consists in the definition of a process. The operations between components of the network are performed in the context of processes. It is achieved at the user level, as well as at lower layers, as presentation, session and transport. Thus the NETEX supports a number of local and remote processes in a common environment. Processes can communicate with each other and interface with this environment in a manner, which is independent of process localization. The management of this is performed by the Communication System (CS) which is a kernel for interprocess communication.

The kernel consists of core-resident supervisors, each implemented at the individual station or server. The supervisor provides management of local processes in the sense of process initialization, process scheduling, local interprocess communication (via send/ receive interprocess communication mechanism), process timeslicing and storage allocation. It also dispatches interrupts (from devices and clocks) and recognizes if remote processes are required. The process address space is common for all processes in the network and each supervisor is capable of recognizing which network destination address is required for given process. The supervisor is also a process but it cannot be explicitly addressed by other processes.

The process which governs the remote interprocess mechanism is a session process. Single occurrence of this process must be activated in the network node needing some communication. It performs communication between remote processes by means of send/receive mechanism in similar way as in well known concurrent programming languages ADA and CHILL (Borzemski et al. (1984), Sas et al. (1984)). But interprocess mechanisms from these languages cannot be just taken into distributed environment. They assume reliable centralized environment with well-designed proven processes. So we have added possibilities to recognize soft failures and message lost by means of positive and negative acknowledgments.

The processes send the messages to other processes (individual, multicasting or broadcasting addressing) as signals of specific types. Each process can declare several types of signals. The signals can have different attributes informing the session process about data representation, relevance to common data representation

ISO layers	NETEX architecture	Realization in the NETEX
application	electronic mail, management of back-end database, experiment management in laboratory with distributed resources, monitoring station, printing station	Applications
presentation	protocol conversion, data format conversion	Communication System CS
session	interprocess logical connection	
transport	multiplexing, addressing, connection management, message segmentation, sequencing, error control, end-to-end flow control	
network	not used in extended version - internetwork addressing	
link	access control, framing, addressing, error detection, correction, link management, node-to-node flow control	IEEE 802 Standard
physical	transmit and receive information specification, bit synchronization, coaxial cable medium dependent signals, modulation techniques and medium specifications	

Figure 2
ISO layers and the NETEX architecture

available at the presentation layer and about storage requirements. The messages are mailed to appropriate signal cases, addressed by the process name and case name. The sending process can wait for the acknowledgment if the message has been inserted in the destination case, or can run immediately after issuing send command. The priorities of the messages allow the kernel to transmit the messages in required order through other lower communication layers as well as to add the message at the proper order in the signal case. The signal cases are served and the message is received by the destination process when the process issues receive case request. The process can specify waiting time for a message if the signal case is empty.
In this mechanism no indication action is available. However, well-defined concurrent processing does not require such mechanism.

Comparing to standard session mechanisms, proposed communication can be theoretically less flexible, for instance, there is no possibility to maintain constant session connection after issuing individual send. This mechanism also requires several sophisticated actions performed by the session process, what may cause more space usage.

CONCLUSIONS

In this paper the development of distributed applications in the university environment has been presented. Four areas of interest for university community have been distinguished, namely education, research, engineering design and local networking development.

The user requirements have been stated for each application and they were synthetized to obtain the requirements for the design framework and ADP system characteristics. Being network planners and network users we can simultaneously use and improve the network services. This gives the possibility to meet the user requirements and increase system performance.

The local area network called NETEX designed to meet studied requirements has been presented. In particular, the communication system allowing for transparent interprocess communication is presented. On its basis the development of several described distributed applications is being expected to be very efficient. A three-node version of NETEX is being currently developed and is expected to operate in late 1985. It will provide basic functions of CS, that is a datagram-type transport service in homogeneous environment. The controllers will provide the connection-oriented LLC protocol.

REFERENCES

[1] L.Borzemski, A.Grzech, A.Kasprzak, The interprocess communication system for the NETEX local area network, Internationalen Problemseminar Lokale Rechnernetze, Nassau (October 1984).

[2] J.D.Day, H.Zimmerman, The OSI Reference Model, Proceeding of IEEE, Vol. 71 (December 1983).

[3] J.Sas, L.Borzemski, T.Kobylański, Kernel and session functions in the NETEX local area network, Working report, Tech.Univ. of Wroclaw (December 1984).

[4] IEEE Project 802, A status report. Local Network Standards Committee, Draft B (October 1981).

A LAN CONCEPT FOR A PHYSICAL LABORATORY

P.E. Giese*, P.H. Giese*, A.V. Alfimenkov**

* Central Research Institute for Physics - Budapest, Hungary
** Joint Institute for Nuclear Research - Dubna, USSR

In scientific laboratories with a great number of on-line experiments the centralized storing and processing of experimental data is a general problem. In the solution presented the experiment control computers are interconnected by a coaxial cable via single-board intelligent Network Interface Units. The implemented general purpose communication subnet and the specialized remote terminal and file transfer transport service is described.

INTRODUCTION

In the Neutron Physics Laboratory of the JINR about twenty small computers of the PDP 11 family are used for data aquisition and measurement control. The measured and preprocessed data must be transferred to the PDP 11/70 laboratory computer for storing on disk or tape and for further processing. Since 1982 a data file transfer system has been implemented for this aim using standard serial interfaces [1].

In order to improve the efficiency and the possibilities of computer communication a LAN concept has been developed and implemented. The main goal was a simple and low cost solution with high data throughput.

COMMUNICATION SUBNET

A single-board Network Interface Unit (NIU) based on Zilog Z80 microprocessor with serial I/O (SIO), parallel I/O, CTC chips and memories has been developed [2]. The NIU provides an UNIBUS interface with two hardware and unlimited number of software registers and direct memory access to the host computer using Z80 block transfer operations. This interface realizes the two bottom layers of the OSI reference model and the IEEE LAN 802.3 standard respectively. The transmission speed over the coaxial cable is determined by the Z80 SIO chip with 0.5 Mbit/s.

The media access control is implemented by software using the carriersense and collision-detection techniques. In our realization each node on the line has equal priority; in case of collision random timeslots are used.

The logical link control provides a simple unacknowledged connection-

less data transfer service. The packets are transferred directly between the host memory and the line without intermediate buffering. The additional functions as framing, transparency, cyclic redundancy checking, address and error detection are fulfilled by the SIO chip logic.

The NIU control software is a set of interrupt reaction routines (1.5 kbyte EPROM and 0.25 kbyte RAM).

The realization of the layers covered by the network interface unit hardware and software is illustrated in Table 1.

ISO - OSI	IEEE LAN 802.3	function	realization
data link	logical link control LLC	- send/receive frames	NIU software
		- framing - transparency - address recognation - error recognation	Z80 SIO chip
	media access control MAC	- line arbitration	NIU software
physical	physical	- send/receive bits	Z80 SIO chip
		- bit coding - carrier sense - collision detection	NIU hardware (line adapter)

Table 1

Network Interface Functions

TRANSPORT SERVICE FOR A SPECIFIC APPLICATION

In the first phase of the application of the general purpose NIU based communication subnet the following user services are offered:

- access to the PDP 11/70 operating system from all connected small computers in remote terminal mode,
- file transfer.

These services are realized in a privileged task under the PDP 11/70 operating system RSX-11M-PLUS and in the corresponding tasks at the on-line computers. At the beginning of each remote terminal session a virtual terminal in the PDP 11/70 will be created.

While the subnet provides a datagram service the transport protocol layer guarantees an error free data transmission. It establishes, maintains and deletes an end-to-end connection (logical channel) between processes, i.e. between the remote and the virtual terminal,

and fulfills buffering and multiplexing. The logical channel will be established after the exchange of unnumbered commands (connection request, acceptance) initialized by the small computer. The data flow is controlled by the PDP 11/70 in a master-slave relation sending numbered commands and receiving its responses. The data packet transferred at this layer contains protocol information as command type, logical channel number, sequence number and optional data.

There are two kinds of numbered commands for data transfer:

- data output command: receive output data contained in the packet; the slave sends a response after correct reception;
- data input command: input data is requested; the slave responses with the requested data.

Error recovery is given by retransmission after timeouts. The sequence number check prevents the duplication of packets.

Figure 1 presents the information flow on the PDP 11/70 node.

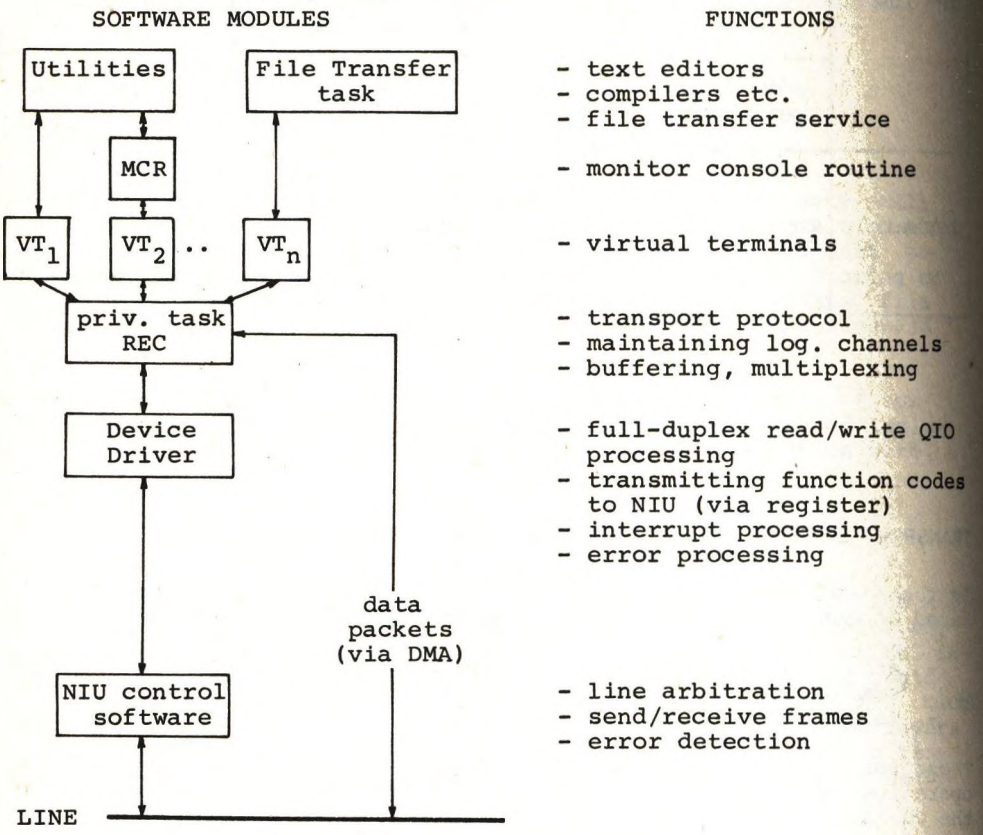


Figure 1

Information Flow on the PDP 11/70 Node

The main advantage of the presented solution is the direct high speed data transfer between the memories of the communicating computers. The bottleneck of the data transfer is only the executing time of the RSX read/write system directive QIO, which takes about 3 msec. On transport level for the transfer of acknowledged packets of 512 bytes 55%, for 4 kbytes packets 90% of the line speed have been achieved.

CONCLUSIONS

The described system has been realized with an effort of 3 man-years, including NIU test programms, NIU control software, device drivers for the operating systems RSX-11M, RSX-11M+, RT-11 and the transport service. The aim of the first realization phase was to give a working system to the physicists as soon as possible and to test the communication subnet in a real environment. The modular structure of the solution permits further developments and new implementations under modified conditions.

REFERENCES

[1] Aleksandrova, I. et al., JINR Preprint 10-82-407 (Dubna 1982)

[2] Giese, P. et al., Setevoi adapter dlia lokalnih setei, Proceedings of the LAN Conference (Riga 1984)

LOTUNET - A LAN FOR PROCESS CONTROL

Helmut Löffler, Dietbert Gütter,
Barbara Hackler, Hartmut König,
Christa Kubas, Günther Neidhold,
Ingolf Meier

Department of Information Processing
Dresden Technical University
G. D. R.

LOTUNET is a concept for modular structured microcomputer-oriented LAN. Main application fields are industrial process control, CAM, automation of scientific experiments and office automation. LOTUNET has a layered stepwise-architecture. Different services, transmission media and topologies can be selected. In order to achieve good real-time behaviour, most of the services provided by LOTUNET are connectionless.

INTRODUCTION

Generally, the control of technological processes or systems as well as scientific experiments is realized by means of distributed computer systems and in particular by local area computer networks. Microcomputer-oriented local area computer networks for industrial process control, computer-aided manufacturing (CAM) and for the automation of scientific experiments are specialized networks [1]-[4] . They differ from general-purpose local area networks in the following items:

- Usually, the computers in systems for automatic control and for computer-aided manufacturing are incorporated into, or placed physically adjacent to the equipments or the machinery they control. The computers are equiped by dedicated peripherical devices, e. g. sensors, controllers, actuators and they work in real-time environments. Moreover, each of the microcomputers controls several subprocesses. As a consequence, task-oriented real-time operation systems are necessary.
- The response time is one of the most important performance criteria of real-time control systems. Therefore, the communication system of the corresponding local area computer network must provide high data transfer rates during short intervals of time. Hereby it is advantageous that usually short messages are to transmit, e. g. state informations etc.
- A high-level local communication system has to support quasiparallel message transmissions of several application-tasks via logical channels.
- Because of the microcomputers integrated into the specialized LAN and on account of the real-time conditions, the resource requirements of the communication software may be low. It is necessary to achieve a reasonable compromise between a high-level communication support, the overhead caused by the communication software, effective implementation technique, resource requirements and the performance criteria demanded.

In the following chapters we present a concept for modular structured microcomputer-oriented LANs and some implementation experiences as well. The LAN-concept takes into consideration the above mentio-

ned aspects of design, realization and application.

THE MODULAR LOTUNET-ARCHITECTURE

LOTUNET - the LOcal Technical University NETwork - is a modular concept of LANs for process control. Modularitiy refers to the following points of view:

- architecture, i. e. logical structure,
- topologies,
- communication services and protocols.

Logical structure: One of the fundamental preliminary conditions of modular structured computer networks is the layered architecture. That also holds for local area computer networks. Networks for process control applications are rather closed systems than open systems. Nevertheless, the architecture of LANs may be described in terms of the Reference Model of Open Systems Interconnection [5]. The layered structure of the LAN LOTUNET demonstrates figure 1. Two features are remarkable: the five layers and the stepwise-architecture.

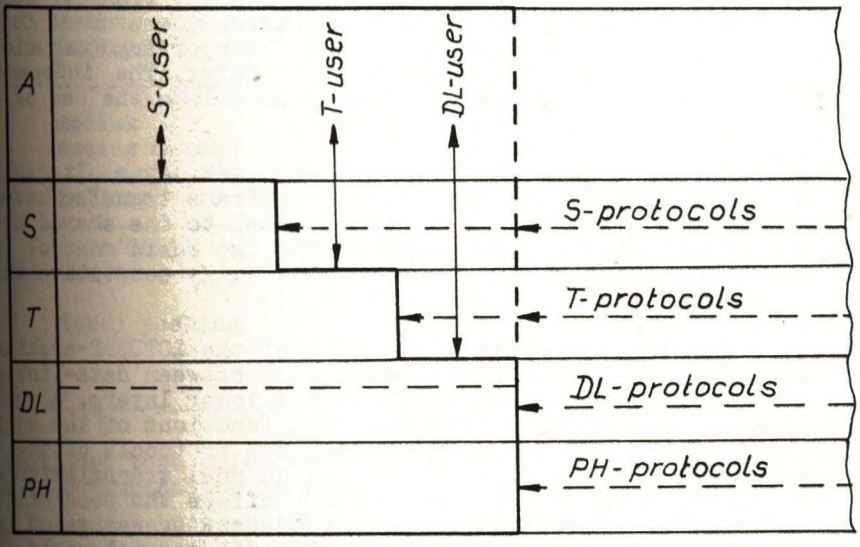


Figure 1
Layered architecture of the LAN LOTUNET

At first we will describe the layers of LOTUNET and then the reasons, advantages and disadvantages of the stepwise-architecture. Services and protocols provided by LOTUNET are described in the next chapter.

In its basic version, LOTUNET comprehends the application layer (A), the session layer (S), the transport layer (T), the data link layer (DL) and the physical layer (PH).

The application layer provides application-oriented services to the users, e. g. message and/or file transfer service, mailbox service etc. All the modules which perform process control functions would be considered part of the application layer. LOTUNET does not provide any specific process control protocols, as these are application dependent, but does provide general protocols as mentioned above.

In the LOTUNET-versions presented here, a presentation layer does not exist, because we use homogeneous computers. If heterogeneous computers are used, the presentation layer is necessary. The implementation of the presentation layer into LOTUNET prototype systems is possible, as the interfaces between adjacent layers are well defined. Up to now there is no need.

The purpose of the session layer within the architecture of LOTUNET is to provide the means necessary for cooperating application entities to organize and synchronize their dialogue and manage their data exchange. LOTUNET provides two classes of session-services: connectionless and connection-oriented message transfer (see next chapter).

The transport layer of LOTUNET is the highest layer which is concerned purely with data transfer tasks. The transport service performed by the transport layer provides transparent transfer of data between transport service users. It relieves these users from any concern about the detailed way in which supporting communications media are utilized to achieve this transfer. The independency of the underlying communications media allows the use of a variety of channel access methods.

Below the transport layer there is the data link layer. It realizes two basic control functions: control of the frame transfer between the interconnected stations and access control to the shared transmission medium connecting the stations. The two basic control functions correspond with two sublayers which jointly constitute the data link layer.

Finally, the physical layer at the bottom of the LOTUNET-architecture provides the means for bit transmission between data-link-entities. By the co-operative work of the two lower layers, a communication system is available providing the functions of the equivalent layers of the OSI Reference Model. The protocols of the two lower layers first of all depend from the channel transmission medium. The architectural concept of LOTUNET offers the possibility to utilize a set of different access methods and transmission media: for example polling/coaxial cable, CSMA/CD/coaxial cable, token bus/coaxial cable, token ring/fibre optic cable. In every case, the two lower layers of the corresponding LOTUNET-version form an unity. The advantage of that consists in the availability of different complete local communication systems. An important consequence for the realization of such an architectural concept are well designed and carefully defined interfaces between the data link layer and the transport layer.

There may be reasons which constrain to dispense with high-level communication support. In the field of process control engineering such reasons usually result from specific real-time conditions. In order to satisfy specific user requirements, the modular concept of LOTUNET involves the possibility to realize stepwise-architec-

tures (see fig. 1). Stepwise-architectures offer the following advantages:

- direct access of application processes to low-level communication services,
- possibility of effective implementation,
- low overhead by the communication software and short response times.

One of the disadvantages of stepwise-architectures consists in the restriction that only users of the same level are able to exchange messages (e. g. S-user tasks cannot communicate with T-user tasks). Dependent upon user requirements of the LAN LOTUNET, different steps and the corresponding communication services can be generated. As mentioned above, the utilization of the two lower layers is obligatory.

Topological structures: From the point of view of network topology, the LOTUNET concept is open for a variety of structures. Nevertheless, the basic topologies of LOTUNET are serial bus structures and loops. Reasons for these structures are: In local area networks, the topological structures must be considered in connection with the medium access methods. Serial bus structures and ring structures are suitable for polling, CSMA/CD and token passing, resp. (see table 1).

Table 1: Medium access methods, media and topologies of LOTUNET versions

medium access method	physical medium	topology
polling	coaxial cable	serial bus
CSMA/CD	coaxial cable	serial bus
token passing	coaxial cable	serial bus
	fibre optic cable	loop

The mentioned topologies and access methods allow the usual patterns of communication in process control systems:

- one to one - between two specific components
- one to many - a message is sent to multiple destinations
- any to one - any stations send messages to a determined station.

Meshed structures are not considered. As a consequence, switching functions and routing functions in our architectural concept are not necessary. Therefore, a network layer is not present in the actual architecture of the LAN LOTUNET. In case of internetworking, a conceptual network layer will be available. These problems are under study.

SERVICES AND PROTOCOLS

One of the basic motives of the LOTUNET architecture consists in freeing its users from all details of communication. As mentioned

above, LOTUNET provides a stepwise-architecture for specific applications. The main application of LOTUNET is process control and CAM. Generally, the interconnected microcomputer stations work in real-time environments and they exchange short messages, e. g. state informations and control commands. In order to support fast message-based interprocess communications, the communication software of the LAN LOTUNET always enables connectionless communication services. Additionally, some architectural layers offer connection-oriented services.

Because process control protocols are application dependent, the application layer of all the versions of the LAN LOTUNET does not provide application-oriented protocols. For specific purposes it offers a file transfer protocol and a mailbox protocol. Generally, application processes use the session service (S-service).

The S-service provides the transfer of messages between the S-user processes which are distributed within the network. The messages are transmitted as S-service-data-units (SSDU). Access to the S-service is offered at S-service-access-points (SSAP) which are located by means of its adresses. The S-service does neither restrict the content, format, or coding of the messages to transmit, nor does it ever need to interpret its structure or meaning. S-users can select two services: a connection-less and a connection-oriented service. An advantage of the connection-less service is that communication resources are necessary only during the transfer of messages. It also may be disadvantageous, e. g. if at the sending process and at the receiving process all recources are occupied and for some time transmissions cannot be initiated. In case of connection-oriented S-service, S-connections are established, maintained and released. During the existence of S-connections, message transfer is every time guaranted on these connections.

The operation of the S-services may be described by a set of S-service primitives. Figure 2 shows a time-sequence diagram of the connection-less S-service. The S-service makes use of the services provided by the transport layer.

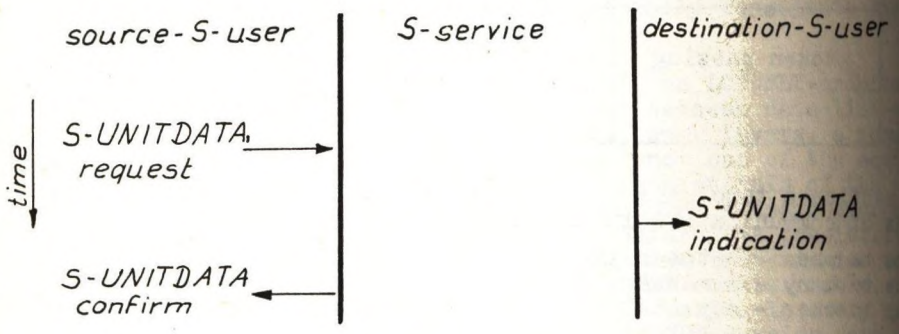


Figure 2
Time sequence diagram of connection-less S-service

Generally requirements for the transport service (T-service) are:

- End-to-end significance. The T-service provides for the transfer of data between two T-service users in end systems;
- Independence of underlaying communication resources;
- Transparency of transferred information;
- T-service user adressing; i. e. the T-service utilizes a system of adressing which is mapped into the adressing scheme of the

supporting DL-service;
. Quality of service selection: T-service users demand a quality of service which is specified through the selection of values for service parameters such as transit delay, throughput, residual error rate and failure probability.

The T-service of the LAN LOTUNET is connection-less. During the generating phase of the communicating software, for each T-service user a number of T-service-access-points (TSAP) is defined. Throughout these TSAP the TS-users have access to the T-service. The number of parallel data transmissions is determined by the number of existing TSAPs. A set of T-service primitives enables T-service users to organize data transmissions (table 2).

Table 2: T-service primitives

function	service primitives	parameters
transfer request for a data-unit	T-UNIDATA request	function code receiving TSAP, sending TSAP, length of DU, user-data, ...
indication of reception of a data-unit at the destination	T-UNIDATA indication	function code receiving TSAP, sending TSAP, length of DU, user-data, ...
confirmation of the transfer result at the sender	T-UNIDATA confirm	function code receiving TSAP, sending TSAP, user-data, ...
cancelation of data transfer	T-UNIDATA abort	function code, specification of canceled primitive

The T-service realizes transmissions of transport-service-data-units (TSDU) by means of transport protocols. At the adressed TSAP the destination-TS-user accepts the user-data.
LOTUNET offers the possibility to select two transport protocols: Basic Transport Protocol and Extended Transport Protocol. The Basic Transport Protocol organizes error-free transfer of TSDUs between source-TSAPs and destination-TSAPs. Source-TS-user are informed about the result of the transfer. Additionally, the Extended Transport Protocol supervises the correct reception of data by the addressed TS-users. The source-TS-users get an information about the data reception. The utilization of the Basic Transport Protocol is favourable if user-data of short lengthes (less than 240 bytes) are to be transmitted. It is also advantageous in case of cyclically transfered messages. In case of longer messages, the Extended Transport Protocol supports the synchronization of the interacting T-users and it provides error-free communication.
The transport layer of the LAN LOTUNET has been modelled by closed

queuing systems using the mean value analysis. Some results are shown in figure 3.

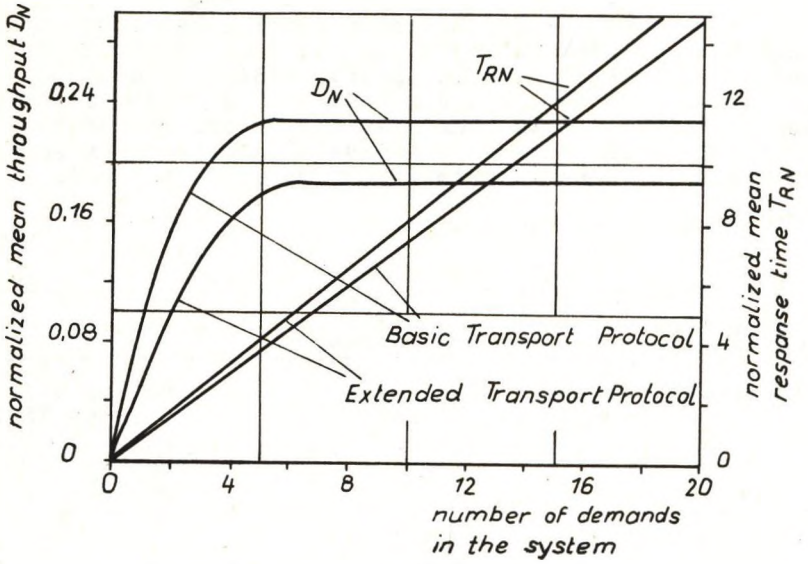


Figure 3
Normalized throughput and normalized response time of two transport protocols versus number of demands in the system

Mean normalized throughput and mean normalized response time of the transport layer are drawn versus the number of demands in the layer. This number corresponds to the number of protocol-data-units. As can be seen, the Basic Transport Protocol enables higher throughput than the Extended Transport Protocol because the overhead of the latter is greater. By the same reasons the mean response time of the Basic Transport Protocol is shorter.
The T-service uses the services provided by the DL-layer. The DL-layer realizes three important functions:

- It receives DL-service-data-units (DLSDU) from the DL-users and realizes the delivery to the P-layer. Taking over of DLSDU involves error-check (errors of addressing and length-errors),
- Taking over of DLSDUs received from the P-layer and delivery to the DL-users. That includes checking of errors occured within the P-layer,
- Control and management of medium access.

The DL-service of LOTUNET is connection-less. As mentioned in the preceding chapter, the modular concept of LOTUNET allows the utilization of different medium access protocols, topologies and communication media (see also table 1). In each of these versions, the DL-layer and the P-layer form an unity. Upwards, i. e. to the T-layer, this unity represents a black box of given properties and performance.
The different versions of the LAN-concept LOTUNET (table 1) has been realized experimentally. In each case 8-bit-microcomputers are interconnected. The normal transmission speed on the physical

medium is 500 kbit/s. The maximum cable length is of the order of magnitude of several hundred meters. Up to 255 stations can be integrated into the LAN.

CONCLUSION

LOTUNET is a concept for modular structured local area computer networks. Modularity refers to layering, services, protocols, access schemes, physical media and topologies. Experimental realizations of several LOTUNET versions using microcomputers demonstrate the validity of the modular architectural approach.

References

[1] Franta, W.R., Jensen, E.D. et al, Real-Time Distributed Computer Systems, in: Yovits, M.C. (ed.), Advances in Computers (Academic Press, New York, 1981), Vol. 20, 39-82.

[2] Thurber, K.J., Local Computer Networks, in Yovits, M.C.(ed.), Advances in Computers (Academic Press, New York 1981), Vol. 20, 73-114.

[3] Sloman, M.S., The CONIC communication system for distributed process control. Informatik-Fachberichte, Kommunication in verteilten Systemen - Anwendungen und Betrieb, Bd. 60, Springer-Verlag, Berlin-Heidelberg-New York 1983, 637-657.

[4] Löffler, H., Gütter, D. and Meier, I., An upper-layer proposal for specialized local area networks.Proc. IFIP-TC 6 Symposium Networks in Office Automation, Sofia 1984, 493-503.

[5] ISO/DP 7498: Data Processing-Open Systems Interconnection-Basic Reference Model.

SESSION 5
ARCHITECTURE OF COMPUTER NETWORKS

COMMUNICATION NETWORKS FOR BANKING SERVICE DELIVERY

D.A. Sheppard

Canadian Imperial Bank of Commerce
Toronto, Canada

Communication networks have become essential for the delivery of financial services to customers and for the exchange of financial information among banks. An infrastructure is now emerging that permits banking messages to travel securely and efficiently virtually anywhere in the world. Few business activities illustrate the potential of services conveyed by computer networks better than the financial services industry. An overview of banking services and network requirements is provided, and a scenario for future banking networks is proposed. The role of Open Systems Interconnection in the establishment of an "open" Universal Banking System Environment is described and the primary characteristics of such an environment are presented. The activities of ISO TC68/SC5 in the standardization of telecommunication messages for banking are described, and some recommendations for future development are offered.

INTRODUCTION

Over the past decade the effective use of automation has become critical to the operations of the financial industry, with software emerging as one of the valuable technical assets of a bank. Initially, the deployment of computers and their access via telecommunications was a support oriented strategy - until the cost, risk and competitive disadvantage associated with a computer-assisted, paper-based operation became prohibitive. Recently, rapid advances in technology have led to significant new business opportunities - that is, the "services" offered by banks to their many customers can now exploit technology in ways that would previously not have been profitable. Examples include the calculation of interest on account balances on a daily basis, the customization of banking services to suit each customer's needs, the development of customer-activated automated banking machines (ABMs), the presentation of services in multiple languages, and the gradual introduction of Remote Access Banking.

This paper explores the present and future role of communication networks in the delivery of financial services and proposes that an architecture be adopted for the harmonization of network evolution. Firstly, financial services are characterized and high level networking requirements derived. A conceptual model for the financial transaction cycle is then used to characterize a Universal Banking System Environment. Trends in network development for financial services are described, and major scenarios are projected. One of the critical success factors in the creation of an integrated financial networking infrastructure is the standardization of the messages that convey financial information and commands between the communications nodes that will inevitably include a variety of manufacturer's equipment. Current efforts in this direction by TC68/SC5, the ISO standards committee dealing with banking telecommunications, are reviewed.

This paper presents the opinions of the author and, while generally based on the Canadian environment, does not necessarily represent policies or plans of the Canadian Imperial Bank of Commerce or the Canadian banks. The Canadian Imperial Bank of Commerce is one of Canada's largest banks and is also a very large user of telecommunications in Canada. The terms "bank" and "financial institution" are used interchangeably.

BACKGROUND

Few businesses illustrate the power of "services conveyed by computer networks" better than the financial services industry. Indeed, few businesses are better positioned to exploit the capabilities of computer networks and to benefit from the Information Age. Banks are essentially information businesses, and those that have realized this are rapidly integrating the tools of the "third wave" into their operations. The implementation of such facilities as centralized account files, customer information facilities, event-driven account posting, and intelligent teller workstations is changing the bank branch from a relatively isolated "micro-bank" into a fully integrated service delivery and marketing outlet.

Even current levels of network usage have resulted in telecommunications costs that are a significant portion of total "non-interest" expenses and these costs are projected to continue to grow rapidly. These factors have led to an increased focus on both the value added and the manageability of telecommunications. In some cases the "Chief Communication Officer" is now a member of executive management.

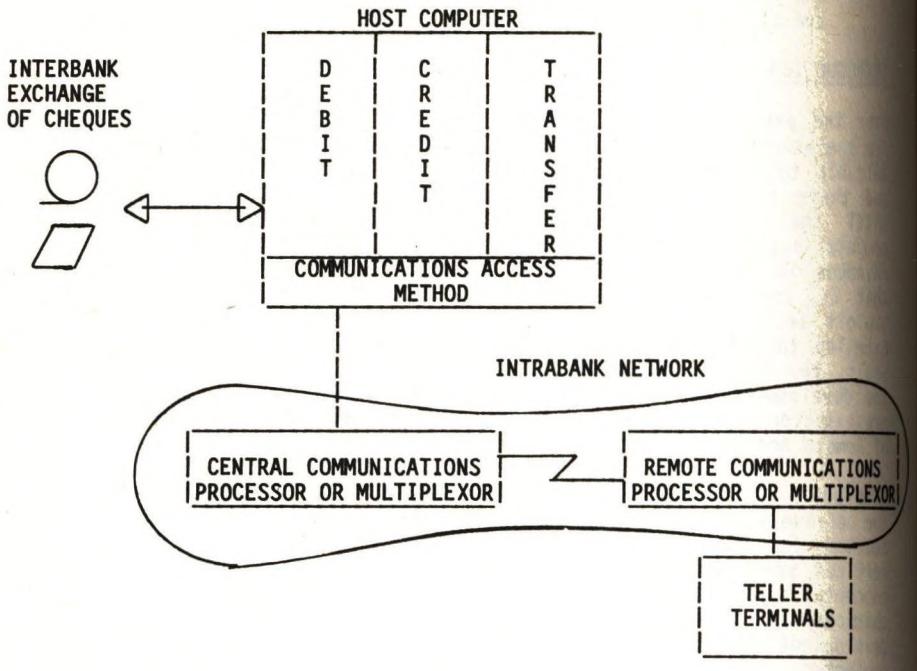


FIGURE 1: A Traditional Branch Communications Network

A typical early network for service delivery to branches is illustrated in Figure 1. These networks are straightforward heirarchical computer networks that provide teller support services and interconnect to large central databases for account and customer data. They are typically based on dedicated circuits and network concentrators. Initially the terminals were simple mechanical devices, but these are now being replaced by intelligent workstations that vastly improve the presentability of the data and significantly reduce the volume of data transported. Facilities for initiating the transfer of funds from an account in one bank to a different bank (normally internationally) may also be included. Branch networks are usually restricted to use by bank employees, and were originally implemented for the computerization of manual branch procedures. Branch networks constituted the "first wave" of automation in banking service delivery and serve to illustrate a strategy based on a "computer-centred view of the universe".

Traditionally, banking services were "delivered" only at the branch (even though data processing was centralized) and the customer was required to present himself at the branch to execute his business transactions. In this environment all telecommunications resources are under bank control and physical security can therefore provide substantial protection against unauthorized access to the networks. It has now become increasingly necessary for banks to provide "external" interconnections - to consumers, corporations, suppliers, governments, other banks, etc. Early examples of this were SWIFT (Society for Worldwide Interbank Financial Telecommunications), VISA and Mastercard authorization networks, and magnetic tapes containing cheque payment data. All of these relate to the interconnection of different banks, and represent the "second wave" of automation. The "third wave" involves communication links to customers. Current examples include corporate cash management, payroll data entry, point of sale, and ABM's. Clearly, these systems move some of the telecommunications resources outside bank control and necessitate much more formal security and authentication procedures.

To achieve any significant degree of external interconnection, banks must ensure that a variety of transport mechanisms can be used in safe, convenient and economical ways. Ideally, the complexities of interconnecting different network types (LANs, WANs, etc.) will be hidden by the definition and adoption of a common Financial Transport Network. Additionally, for interconnection between banks and "the outside world" to occur at all it is essential that all transported data be understood by all parties. This requires commonly accepted data definitions and encoding rules. Finally, common approaches to access control, security and authentication are necessary in order to protect customer confidentiality and minimize fraud losses. Banks are in the process of "opening" their networks and, wisely, are doing this in a stepwise, highly-controlled fashion. To achieve this goal, the banks are beginning to adopt a "network-centred view of the universe". It is hoped that responsiveness to changing service delivery requirements will be perceived by customers as a sign of excellence in the financial industry.

BANKING SERVICE DELIVERY

Over the next decade banking services will increasingly be delivered directly to the customer at his own location. It is expected that these services, which can be called Remote Access Banking, will include most of the familiar banking services other than cash withdrawal (balance inquiries, transfers, bill payments, credit applications, securities transactions, etc.) and may also include non-traditional banking services such as personal financial planning, insurance, mortgage applications, financial analyses, and others. In some cases, a bank could even become a specialized "information broker"

(for instance, a videotex system operator) and, in other cases, may participate jointly in systems sponsored by another party (such as home shopping with direct payment from a bank account).

In fact, increased convenience through service delivery to customer-activated terminals has already become a competitive issue, although not yet in the home market. Some of the benefits of Remote Access Banking are:

- transaction off-loading reduces branch traffic and eliminates terminal costs;
- convenience of access is a product differentiation and promotional tool;
- potentially provides a fee generating service; and
- service usage is promoted.

Remote Access Banking will involve a very large number of delivery points (in addition to branches) - the home, retail outlets, public locations such as hotels, and the office are some examples. All of these require communication networks that allow the interconnection of multiple device types, at all times of the day, which are widely distributed geographically. It is projected that Remote Access Banking will lead to a terminal population in the millions, with a correspondingly large transaction rate. At present only telephone networks are this order of magnitude in size and distribution.

Banks will ultimately be unable to dictate methods of attachment or anything other than basic device characteristics to their customers, and hence must utilize standards that are acceptable to the majority of participants (much as is now the case for telephone sets in Canada).

Delivery of banking services using Automated Banking Machines (ABMs) is the current high growth area, with shared ABM networks beginning to emerge as the dominant direction. A shared ABM network can deliver banking services to three classes of terminal device:

- non-bank devices (retail store in-house ABMs or third party service bureaux);
- bank devices with another bank's card used (e.g. public location, bank-owned ABMs); and
- bank devices with the same bank's card used (e.g. in-branch ABMs).

The devices may be connected in either a "back end" or a "front end" configuration depending upon whether the transaction is processed first by the host computer of the bank who services the ABM or is routed directly to the cardholder's bank. The shared ABM service shown in Figure 2 illustrates both scenarios. This topology is also valid for Electronic Funds Transfer at the Point of Sale (EFT/POS) applications.

A banking transaction generally proceeds in a series of distinct phases that are separate as far as network sequencing is concerned but linked in terms of the progress of the business activity. For example, a cash withdrawal from an ABM may include the following independent messages which must each be transported and yet together they constitute the completion of a single banking transaction:

- Card Accepted and Validated;
- Requestor Identified;
- Withdrawal Authorized and Limit Decremented;
- Withdrawal Executed and Logged;
- Withdrawal Completion Notified; and
- Interbank Settlement effected.

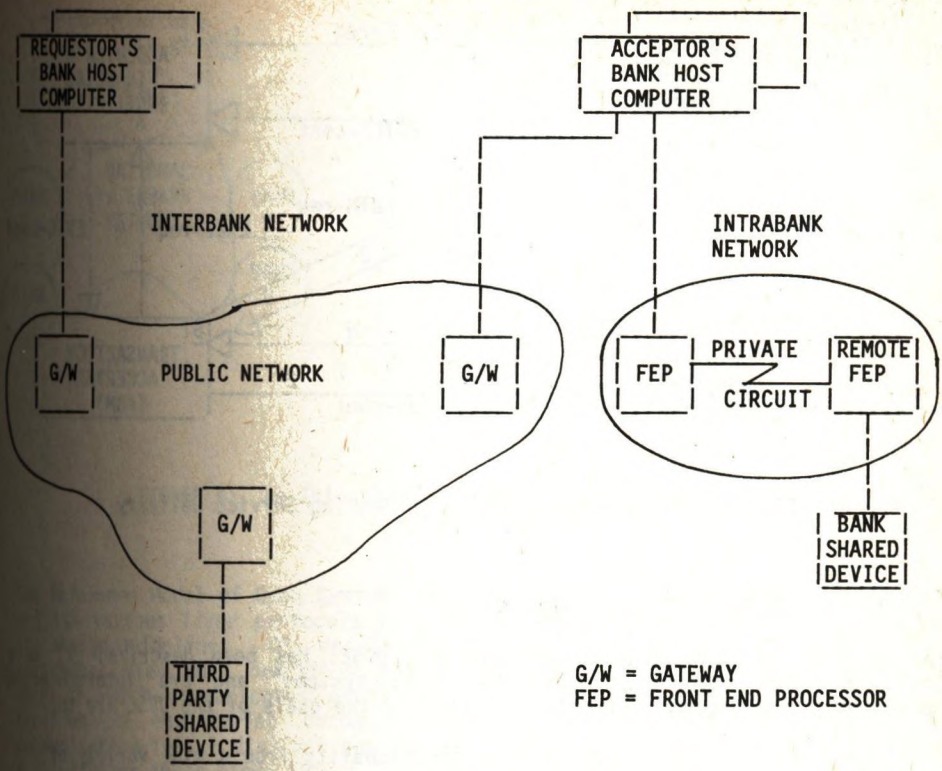


FIGURE 2: A Typical ABM (or EFT/POS) Network

Since these banking functions are independent of the delivery mechanism, it is possible to derive a conceptual model for banking transactions. This is illustrated in Figure 3. The four active elements in the transaction are:

A - the transaction requestor (customer);
B - the transaction acceptor (ABM device and related device support);
C - the acceptor's financial institution; and
D - the requestor's financial institution.

The five paths for information flow, for the ABM cash withdrawal, are:

1 - Places funds into an account for subsequent access;
2 - Customer withdrawal request;
3 - Confirmation that funds exist and can be withdrawn;
4 - Funds dispensed from ABM-owner's account; and
5 - Interbank settlement of accounts.

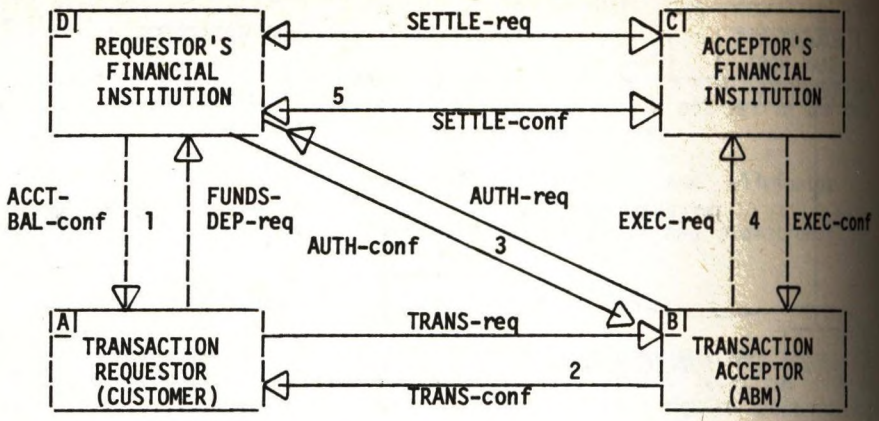


FIGURE 3: Conceptual Model for Banking Service Delivery

UNIVERSAL BANKING SYSTEM ENVIRONMENT

A "Universal Banking System Environment" (UBSE) has been described [1] as a set of banks that are viewed as "open systems" and are interconnected nationally and internationally. In general, the goals of the UBSE are to:

a) allow customer access to banking functionality from a wide variety of delivery points, anywhere in the world;

b) enable banks to exchange financial information electronically;

c) minimize costs and development efforts associated with inter-system interfaces;

d) permit banks to adopt an "event-driven" approach to processing; and

e) maintain high reliability, integrity, security and availability.

In addition to providing a high quality of service for its basic data transport activity, the UBSE must also satisfy the following criteria:

a) overall cost minimization at the institution, national and global levels;

b) strict access control and customer privacy;

c) universality of interconnection and interface;

d) hiding of details of physical equipment through standardization of the Financial Transport Network Service;

e) general acceptability of design (i.e. no class of user favoured in implementation complexity); and

f) generally modular, expandable, dynamically configurable, etc.

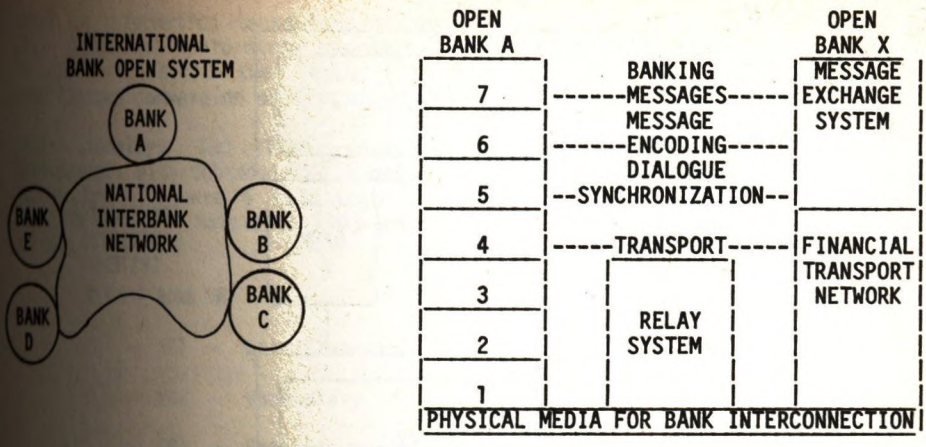


FIGURE 4: An Interbank Message Network Based on OSI

The Reference Model of Open Systems Interconnection (ISO 7498 and CCITT X.200) and its various layer protocols have been created with these criteria among its design guidelines. OSI models the communications process in two ways: a horizontal view that describes the interconnection between "open systems", each of which communicates using OSI protocols; and a vertical view that provides an engineering decomposition of the tasks of an instance of communication into 7 distinct subsystems, or layers, of functionality. Figure 4 illustrates these views [2 - 5], placed in a banking context.

The OSI model provides four layers of protocol for transporting the messages and three layers for the definition and control of message interchange. Banking message standards fit into the Application Layer (Layer 7) and, in OSI terms, define the syntax and semantics for the information to be exchanged. The communicating parties must also agree on an encoding for the data (the Transfer Syntax). This may be negotiated and agreed to by means of an OSI Presentation Layer (Layer 6) protocol or may be determined by bilateral agreement [6, 7].

THE COMMUNICATION SYSTEM

Communication networks for banking service delivery are required for terminal support, for intrabank transport, and for interbank message exchange. To date there has been a tendency towards dedicated, private networks for intrabank communications, closed user group public networks for interbank networks, and dial access to gateway systems for customer-owned terminals.

There is a need for a generic architecture that provides universal access and very high connectivity [8]. The shared ABM system, for example, could involve a variety of banking participants, all of whom may need to communicate with the same terminals. Similarly, a customer at a Remote Access Banking terminal may require access to several financial institutions. An architecture that satisfies these needs for a large number of participants is illustrated in Figure 5. Note that no attempt has been made to imply specific domains of ownership or administration.

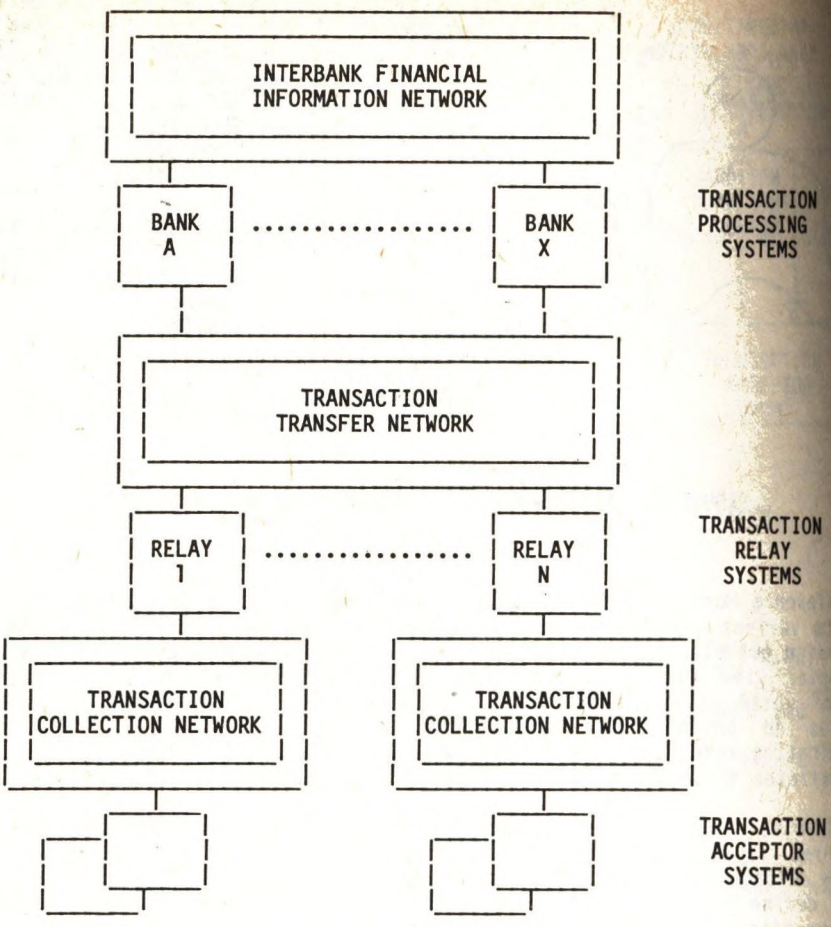


FIGURE 5: General Banking Communications Architecture

In Figure 5, the three network subsystems may physically be the same or they may be distinct and use different technologies. Relay Systems may be offered by banks or third parties, as can the Transaction Acceptor Systems. OSI protocols may be used between any or all of the systems and networks. Different transaction phases will require different types of service. For instance, AUTH messages should be realtime and interactive (i.e., synchronous) and are likely to be short. SETTLE messages are usually based on end of day totals and could use store and forward or file transfer facilities (i.e., asynchronous). In the case of high value transactions, immediate settlement may be desired, however.

BANKING STANDARDIZATION EFFORTS

Standards are considered to be the most effective way to integrate various banking networks into a unified financial services system. The development of a strategy for the selection and adoption of communication standards, for

users of information technology in general, has been discussed in Reference [9]. A gradual conformance approach is recommended for the banking industry in order to avoid the "traumatic" changes that could arise from any simultaneous conversion of systems.

Internationally, ISO TC68 provides leadership in banking standards development, with Subcommittee 5 being responsible whenever telecommunications is involved. Figure 6 illustrates the current organization of TC68, with emphasis on SC5. Reference [10] provides further information on business use of OSI.

TC68 - BANKING
- WG1 - Data Elements
- WG2 - Vocabulary
- SC1 - Bank Documents and Access Devices
- SC2 - Bank Operations and Procedures
- SC4 - Securities
- SC5 Telecommunication Messages in Banking
  - WG1 Bank Card Originated Communications
  - WG2 Telex Messages
  - WG3 Bank Telecommunication Messages
  - WG4 Applications of OSI in Banking
  - WG5 Chip Card Messages

FIGURE 6: Current Organization of TC68 with Emphasis on SC5

For card-based applications such as shared ABMs, POS and Visa, ISO DIS 8583 (Bank card originated messages - Interchange message specifications content for financial transactions) has already been adopted as the basis for message design [11] in several countries.

Within TC68/SC5, WG4 was established in 1984 with the following proposed Terms of Reference:

"Standardization in the area of communications and services for the interchange of financial information in an Open Systems Interconnection Environment (OSIE).

This will include, but not be limited to:

a) Adoption of a design methodology for use by TC68 for the development of banking-specific OSI communications standards, including functional modelling and descriptive techniques for services, protocols and data syntax;

b) Make recommendations to the appropriate sub-committees for the enhancement or modification of current work in TC68 to align banking standards with OSI;

c) Selection of options and facilities within OSI to satisfy the specific needs of banking;

users of information technology in general, has been discussed in Reference [9]. A gradual conformance approach is recommended for the banking industry in order to avoid the "traumatic" changes that could arise from any simultaneous conversion of systems.

Internationally, ISO TC68 provides leadership in banking standards development, with Subcommittee 5 being responsible whenever telecommunications is involved. Figure 6 illustrates the current organization of TC68, with emphasis on SC5. Reference [10] provides further information on business use of OSI.

TC68 - BANKING
- WG1 - Data Elements
- WG2 - Vocabulary
- SC1 - Bank Documents and Access Devices
- SC2 - Bank Operations and Procedures
- SC4 - Securities
- SC5 Telecommunication Messages in Banking
  - WG1 Bank Card Originated Communications
  - WG2 Telex Messages
  - WG3 Bank Telecommunication Messages
  - WG4 Applications of OSI in Banking
  - WG5 Chip Card Messages

FIGURE 6: Current Organization of TC68 with Emphasis on SC5

For card-based applications such as shared ABMs, POS and Visa, ISO DIS 8583 (Bank card originated messages - Interchange message specifications content for financial transactions) has already been adopted as the basis for message design [11] in several countries.

Within TC68/SC5, WG4 was established in 1984 with the following proposed Terms of Reference:

> "Standardization in the area of communications and services for the interchange of financial information in an Open Systems Interconnection Environment (OSIE).
>
> This will include, but not be limited to:
>
> a) Adoption of a design methodology for use by TC68 for the development of banking-specific OSI communications standards, including functional modelling and descriptive techniques for services, protocols and data syntax;
>
> b) Make recommendations to the appropriate sub-committees for the enhancement or modification of current work in TC68 to align banking standards with OSI;
>
> c) Selection of options and facilities within OSI to satisfy the specific needs of banking;

d) Act as a focal point for interaction between TC68 and TC97 in the area of OSI as it applies to banking."

It is intended that WG4 will formulate Application Layer protocols to correspond to SC5 message definitions. Future tasks may include interbank network management, a generalized message exchange syntax, banking directories, and schemes for naming and addressing in the Univeral Banking System Environment.

In addition, the following activities of TC68 are of interest:

- 68/1/3 - Bank Cards - Magnetic Stripe Data Content (ISO 4909-1978)
- 68/2/2 - Banking Procedures - Test Key (DP 8376)
  - Key Management (DP 8376/PART 3)
- 68/2/6 - Personal Identification Number: Management & Security
- 68/4/2 - International Securities Messages

The banking community is following with keen interest the work of TC97 (especially SC6, SC20, and SC21) in the area of Open Systems Interconnection protocol standards and security standards.

CONCLUDING REMARKS AND RECOMMENDED ACTIONS

In this paper we have characterized a Universal Banking System Environment and described banking services that require effective communication networks for convenient, secure, economical delivery. The concept of "openness" for banking telecommunications has been introduced and a scenario for the interconnection of open banking systems described.

It has been observed that standards, both for the banking messages (syntax and semantics) and for data transfer, are essential to the successful establishment of worldwide electronic banking. Activities within ISO TC68 and TC97 were reviewed very briefly to illustrate the directions being taken in the standards community. The Open Systems Interconnection approach has been recommended as a general framework for communication and the use of card-oriented messages based on DIS 8583 is proposed.

It is easily concluded that computer networks are now an essential part of banking operations, and that planned co-ordinated developments are required if unnecessary problems in interworking are to be avoided.

The following areas are felt to be critical to future success and are recommended as areas for early attention by planners and developers:

a) formal adoption of the OSI Model as a framework for inter-system communication and the selection of lower layer service elements;
b) harmonization and standardization of data element definitions and encoding rules (both within the banking industry and with major classes of customers) based on ISO DIS 8583;
c) implementation of naming/addressing/directory conventions for banking (i.e. customer and bank identification codes);
d) formalization of security and authentication requirements (again both internal to the banks and externally) based on the work of TC97/SC20, TC68/SC2 and TC97/SC21/WG1; and
e) development of Application Layer protocols, including specific application service elements and management.

REFERENCES

[1] Sheppard, D.A. and Dickson, H.C., A View of Banks as Interconnected Open Systems, ICCC 82

[2] ISO 7498 - Information Processing Systems - Open Systems Interconnection - Basic Reference Model

[3] Sheppard, D.A., OSI provides framework for network planning, Canadian Datasystems, June, 1983

[4] --, Ways to Open Communication, Siemans AG, October, 1983 (order number U1474-J-Z72-1-7600)

[5] Rosenberg, R., Closing in on Open Systems, Electronics, May 31, 1984

[6] Melendez, W.A. and Petersen, E.I., Tutorial on the Upper Layers of the OSI Reference Model, Commission of the European Communities, June, 1984

[7] Larmonth, J., Introduction to the Technical Content of OSI Layers 6 and 7, September, 1984

[8] Branscomb, L.M., Computer Communications in the Eighties - Time to put it all Together, ICCC80, North-Holland Publishing Company 1981

[9] --, The User View of Communication Standards, Information Technology Standards Committee for Private Sector Users, c/o The National Computing Centre Ltd, Oxford Road, Manchester M1 7ED

[10] Foley, J.S., Business Data Usage of OSI, Proceedings of the IEEE, December, 1983 PP 1442-1445

[11] ISO DIS 8583 - Bank Card Originated Messages - Interchange Message Specifications - Content for Financial Transactions, February, 1985

ABOUT INTEGRATION OF CONTROL AND COMMERCIAL INFORMATION SYSTEMS

M. Paul and W. Kunft

EDP-Center of the
Technical University of Vienna,
Vienna, Austria

An outline is given on the communication demands of a major industrial enterprise. The possibilities are presented to show the integration of information systems in various divisions of an enterprise by connection of local area networks to a wide area open systems interconnection. After a short introduction into ISO's rules for an open systems interconnection the realization of such a connection is discussed using ACONET, an experimental network to combine the EDP-centres of the Austrian universities.

1. INTRODUCTION

The dramatic increase in computing power and memory space has favoured developments which, considered until recently, would be unrealistic in the near future. The decentralization of computing power is t h e answer to those requirements. Thus it would no longer seem utopic for every employer to have his own computer at his place of work, as he has his telephone today.

The convenient office computer will have various additional provisions depending on the requirements of the working place. Thus, a computer in a laboratory will have various process input/outputs for connection with measurement equipment, while a computer for a supervisor will have direct access to the control equipment of a part or the whole of the factory. Computers for office workers will possess powerful word processing programs. However, certain standardized essential functions will be available at every place of work, as e.g. calculator arithmetic, electronical dictionary, basic graphics to visualize important figures of the process and, last but not least, an extensive number of utilities to ease the handling of the computer by the user.

Decentralization makes sense only if the easily accessable information in the centralized system still remains so. That means that there must be a great amount of information exchange between the various office computers. With this exchange we have to distinguish between system oriented and user oriented demands. A system oriented demand arrives, when an office computer needs resources, which are not available at the place of work because of economical or technical reasons, e.g. noise. For instance, you install peripheral equipment only once or only in subcentral stations if it is used only rarely and if its price is (still) fairly high as colour graphic plotters, letter quality printers, etc.

This kind of system oriented communication demand will decrease in future as the price of the various peripheral devices becomes lower and lower. At the same time, however, the amount of communication between the decentralized computers will increase rapidly, as these computers will have to be fed with (special) programs to secure the data basis and to support the remote failure analysis and remote computer maintenance. The user oriented communication consists of the

information exchange between the users themselves (electronic mail), but in particular of the information exchange between cooperating user programs. This is when in widely distributed areas complex user tasks have to be performed with task synchronisations, with bi- or multilateral supervision, etc.

The present communication demand will grow intensively in the future and has to be taken care of by efficient networks. With catch-words like LAN (Local Area Network), WAN (Wide Area Network) or MAN (Metropolitan Area Network) the technical user will get the impression that all his communication problems have already been solved.

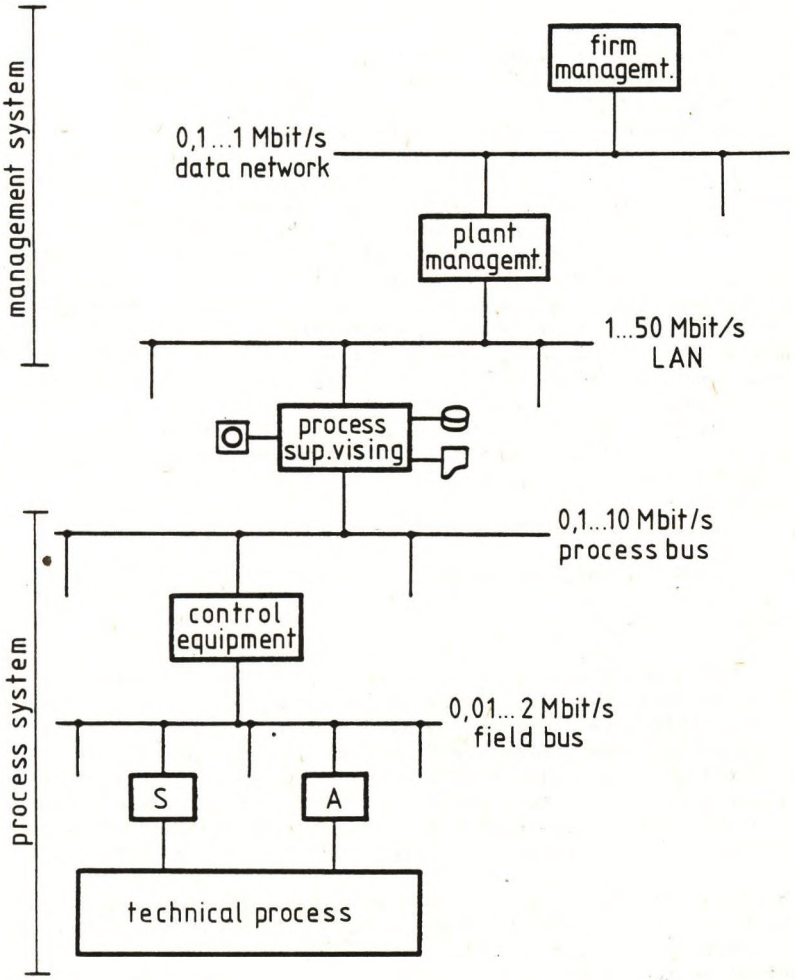


Fig. 1 Hierarchy of bus systems and networks in an industrial enterprise

(S...sensor, A...actor)

The whole communication demand of an enterprise can generally be divided into four layers, fig. 1. The lowest layer consists of a field bus which connects the control equipment with intelligent sensors and actors. Applying this will save great amounts in cabling costs, [1]. The transmission speed can reach the Megabit/s-area, [2]. In the layer above various process control equipment will be connected with each other and with the process supervising computer. Apart from special products of firms to realize a process bus the international draft standard PROWAY has good chances of being generally adopted.

In the layer between process and supervising control it is possible to use local area networks for the first time. While the lowest two communication layers are outlined and performed as a line bus due to technical constraints, the possibilities in the third layer are essentially greater. Based on the topology we distinguish meshed, star, ring and line shaped networks, based on the control system centrally and decentrally controlled systems. Although it is conceivable to combine each kind of control system with each kind of topology, it has appeared that a central control system best fits a meshed or star like network, while a decentral control system is better combined with a ring or line shaped network [3]. Up to now only decentrally controlled networks have gained importance as local area networks. Currently the most widely used local area network is ETHERNET [4]. Ethernet is a line shaped network. Line shaped networks in comparision with ring shaped networks have the advantage of a real "passivity", i.e. the break down of a station does not affect the whole system. Furthermore this topology allows an easy extension of the network and makes it possible for a message of a transmitter to reach all receivers more or less simultaneously, i.e. without delay. The disadvantage of this topology is that transmitter collisions may occur and this problem has to be solved by randomly delayed retransmitting times, [5]. At the highest level of communication, which in any case encompasses more than a single enterprise, we can find public networks, as DATEX-P, DATEX-L or in future ISDN. The connection of these networks with the local area networks still imposes a number of unsolved problems and in particular justifies the search for an answer by using pilot implementations.

2. INTEGRATION OF THE INFORMATION SYSTEMS

2.1 Architecture of the Information Systems

The various EDP firms competing with each other could take varyingly large shares of the market depending on the sector of economy. The share of the market a firm could take additionally differs within a sector of economy depending on whether the EDP applications are oriented more commercially, administratively, technically, etc. A great share of the market of an EDP firm in one sector mostly leads to a "quasi-standard" for hard- and software. The EDP user is then often very reluctant to change the standard although other EDP firms offer a better and more cost efficient solution for certain new tasks.

The commercial branch of an enterprise needs the latest production numbers in order to adapt the production figures to market demands, to secure quality demands, to detect production bottle necks very early, etc. On the other hand the technical sector of an enterprise requires the various planning data of the commercial sector in order to adjust the priorities within the production deadline. All this means that there is a strong interlocking of the various sectors of an enterprise. Every sector already runs extensive EDP systems nowadays, which ought to or will have to communicate with EDP systems in other sectors. If the EDP systems in the various sectors are not from the same EDP producer, one will have a lot of adapting problems which can only be solved with a great amount of time and money. A general solution for these problems is only conceivable when we switch to open networks.

2.2. OPEN NETWORKS

2.2.1 The Basic Reference Model for Open Systems Interconnection

The aim of an open systems interconnection (OSI) must be the definition of the necessary rules so that user programs in locally separated computer systems of different producers are able to co-operate with each other. An open system is defined by ISO as a system which behaves exactly according to the ISO rules, [6]. The basic reference model forms the heart of OSI. At first there is an essential abstraction. The internal behaviour of a real system within the open systems interconnection is not of such great interest but its external behaviour. Therefore, the external behaviour can be laid down if one does not consider the real system but only an abstract one, whose internal structure is not needed to be known. A model of abstract systems is, therefore, taken to define the rules of cooperation within an open systems interconnection.

The basic reference model defines the basic rules which the communication in an open systems interconnection has to follow. It defines all necessary functions and their structure. By hierachical division of the communication functions, which are necessary in an open system, we get subsystems. An (N)-subsystem has to exchange information only with its neighbouring subsystems, i.e. the (N-1)- and the (N+1)- subsystem. If we consider subsystems of equal functions which take part in the interconnection we get a function layer.

Each of these layers has to furnish the next higher one with very certain services. These services are defined for each layer in the basic reference model. An entity of a layer uses the services of the underlying layer via so-called service access points. An entity of the (N)-layer has the following possibilities to produce the required service:

1) using its own functions
2) cooperating with entities of the same layer
3) calling on the service of the (N-1)-layer and by it on the services of all layers underneath.

In order that the entities of the (N)-layer can cooperate with each other to produce the required service, a connection between them must be established. This occurs by the aid of the service of the (N-1)-layer, which itself establishes an (N-1)-connection between the cooperating (N)-entities. The precondition for establishing a connection between two systems is the existence of a communication medium. (N)-entities which are not directly coupled via the communication medium can establish connections with each other by using a special function, viz. the so-called relay function, which forwards a message from an (N-1)-connection to another one. Global names for entities throughout the interconnection as well as addresses of service access points by which entities of higher layers can be reached form the basis to establish connections between entities of one layer using appropriate directories, relay functions and functions for routing.

The basic reference model defines seven layers: The Physical Layer consists of those functions which enable the bit transmission between neighbouring systems via the communication medium. This layer copes with the whole variety of communication media. However, the higher layers do not have to take care of this variety.

The Data Link Layer is necessary as the communication medium may be not free of errors and the bit transmission in the physical layer may, therefore, be incorrect. The data link layer, therefore contains functions and the necessary protocols for error detection and error recovery. This layer provides the higher layer with the service of a secure bit transmission between systems directly connected by the communication medium.

The Network Layer takes into account that systems not directly connected by the communication media must also be able to cooperate. These problems exist in all data networks. This layer, therefore contains all those functions such as relay function, routing, multi-use of a single connection of the data link layer (multiplexing), etc, which make it possible to transmit a message from the source system to the destination system via one or more other systems lying between. According to ISO the network layer offers a service to the layers above, the functionality of which is independent of the data network used.

The task of the transport layer is to offer an information transport for the layers above it with a quality required by these layers. While the functionality of the network layer's service is independent of the network used, the quality of the information transport, of course, depends very much on the specific network. The transport layer, therefore, consists of functions and protocols which help to bridge the gap between the required and the given transport quality offered by the data network used. The result is a service for the upper layers comprising all relevant matters of information transport and its quality.

The Session Layer provides the means necessary for cooperating programs in different systems to organize and synchronize their dialogue. This layer essentially contains functions to:

1) confine as well as to control and supervise activities on a session-connection and the dialog units belonging to such activities,
2) administer communication-rights,
3) define synchronisation points in order to reopen the dialogue at a defined position under defined conditions if a failure occurs.

The Presentation Layer offers the above layer two categories of services, viz. the control of the information exchange and the negotiated and agreed presentation and manipulation of structured data. The first service is not offered by the presentation layer itself, but by calling on the services of the session layer. The second service allows the layer above to negotiate and to administer so-called presentation contexts, by which the type and the structure of the information as well as the operations, which are to be applied on the structured data, can be defined.

Finally the Application Layer is the highest layer. It contains the application processes which process the information exchanged via presentation connections. Thus application processes in one system can cooperate with those in other systems of the interconnection to solve complex applications. This layer does not offer a service itself but represents the total of all possible applications of the services of the six lower layers.

2.2.2. Standards to Define the Services of Each Layer

The basic reference model only contains fundamental definitions, whose services have to be offered by each layer. It is, therefore, necessary to have a further set of OSI standards which define how the services of the several layers can be called on and how the producing of the required services are to be reported to the layer making the request.

For this purpose standards have been developed for each layer, defining service primitives and their parameters, which are used to communicate between entities of neighbouring layers in the same system to use and to provide services.

2.2.3 Specification of Protocols in Each Layer

Neither the Basic Reference Model nor the Service Definition Standards of the different layers contain the detailed rules to allow different entities of the

same layer to communicate. For this reason a further set of standards is necessary to specify one or more protocols for each layer which are to be applied to allow communication between the entities of the layer concerned.

2.3 CONNECTION OF LOCAL AREA NETWORKS TO WIDE AREA NETWORKS

In spite of ISO's very strong attempts to achieve a quick completion of the standardization of OSI's seven layers it can not be denied that this will still take a long time. Fully functioning network architectures developed by notable EDP-manufacturers compete with OSI which exists today only fragmentarily. The replacement of such manufacturer-dependent networks by "OSI-networks" can not, if at all, be expected in the near future. Rather a future coexistence of OSI- and Non-OSI-networks must be assumed. Connections between both kinds of networks will be made possible by gateways.

2.3.1 Possibilities to Establish Gateways

Today national and international standardization groups discuss different possibilities to connect local area networks to an open systems interconnection, eventually by using public data networks. There are essentially two variants:

Fig. 2 shows the first variant: A gateway contains only the first three OSI-layers. A relay function in the network layer serves to forward information from the local area network to the wide area open interconnection and vice versa. This variant requires in each endsystem of the local area network communication programs according to the OSI-standards in all layers above the network layer to provide a cooperation with endsystems of other subnetworks within the supra-regional open interconnection. If it is necessary to communicate with other systems according to non-OSI-protocols too, e.g. CCITT X.29, the corresponding protocols above the network layer must also be implemented in all endsystems of the local area network.

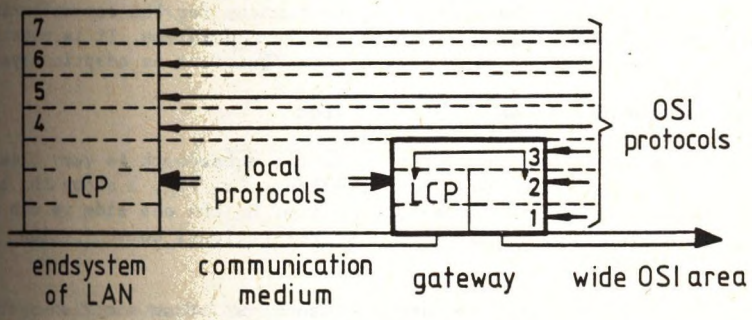


Fig. 2 Gateway according to the first variant

LCP Local Communication Programs

Fig. 3 shows the second variant: A gateway contains not only the first three layers up to and including the network layer but also the higher layers. The relay function is realized in this variant not within the network layer but within the application layer. Gateways according to this second variant are suitable for the interconnection of networks which apply different communications-standards. Such gateways will be named "adaption- systems" in the following. The properties of such adaption-systems and their utilization to integrate subnetworks of an enterprise within an open systems interconnection will be analysed more detailed in the following paragraphs.

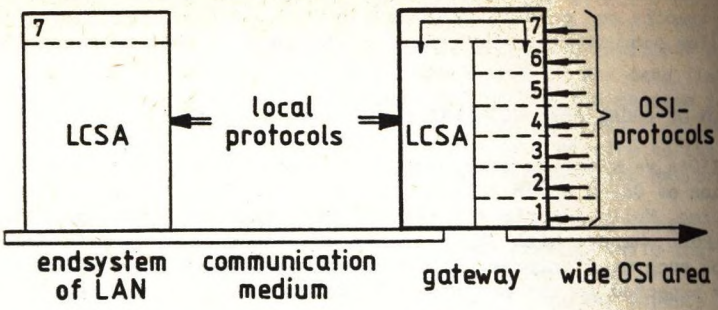


Fig. 3 Gateway according to the second variant (adaption-system)

LCSA Local Communication Software of any structure

2.3.2 Functions of an adaption-system

The purpose of an adaption-system is to offer as many services from as wide an area as possible to a local subnetwork. The cooperation with endsystems of other subnetworks within the wide area open interconnection is only one example. In principle, it should be possible to cooperate with systems of other data networks which do not conform to the OSI standards. Additionally it may be desirable to supply the local area with the access to other public services like videotext or teletex by an adaption-system. Further it should be possible to run communication programs within the local area which are tailored to the special requirements and technical possibilities of the concerned area. Generally such programs will not behave according to the OSI standards.

Each adaption-system contains on the one side local communication programs to communicate with the systems of its own region and on the other side communication programs for the wide area, that means for the communication with adaption systems of other regions or with single endsystems. It is practical to use computers of the same type and of the same equipment as adaption-systems.

2.3.3 Software Structure in an Adaption-System

The interface between an adaption system and its subnetwork is very flexible and can easily be adjusted to the local requirements. Fig. 3 shows the basic structure of the software in an adaption-system. On the one side we can see the necessary communication software for the supra- regional communication according to the rules of OSI.

The most important function in the application layer of an adaption-system is a relay-function which forwards the information flow from the local to the wide area and vice versa. Access-control, accounting and network management for the local as well as for the wide area are functions which are sensibly contained in an adaption-system.

On the other side of the vertical demarcation line in the adaption-system in Fig. 3 we see the software for the communication with systems of the local area. This software is structured as required by the local area network. It may consist only of a simple communication program for asynchronous terminals or may be a complex emulator for intelligent terminals or for the communication procedures applied within the local network.

Of course it is possible that the communication within a local subnetwork is also performed according to OSI-rules. In this case the communication programs

of the layers in the adaption-system can be used for the supra-regional communication to OSI-conforming systems as well as for the local communication to the endsystems of the local area.

3. PILOT APPLICATION

3.1 Conception of the Experimental Interconnection

An experimental interconnection between EDP-centres of Austrian universities has been created by connecting different subnetworks, or single systems of the universities involved via the public packet switching data network DATEX-P, fig. 4. The supra-regional interconnection was outlined as an open data network according to the rules of OSI, [7]. The integration of the subnetworks and the PACER 600 in the open interconnection was performed by the use of adaption systems, i.e. of gateways according to variant 2 as described in chapter 2. PDP-11/23 Plus and PDP-11/34 served as adaption-systems. Except the AS of the DRDP they are not equipped with mass storage devices due to financial reasons. The system 370/155 in Linz was not connected by using an adaption-system but by applying a front-end processor 3705.

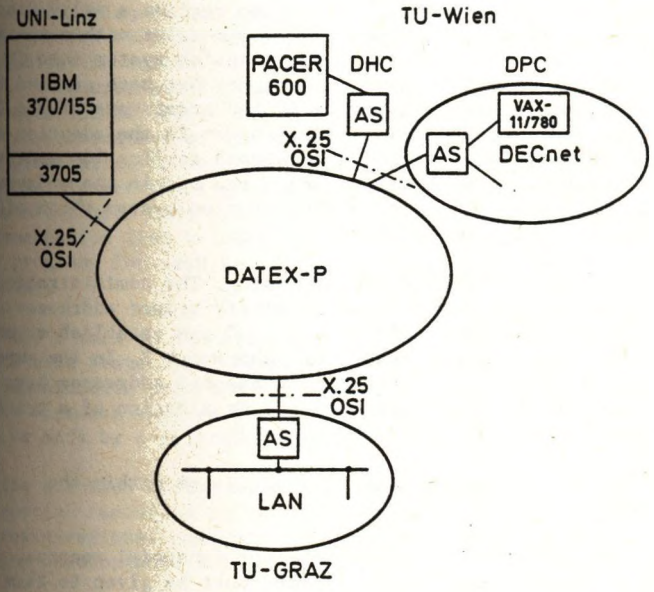


Fig. 4 Experimental interconnection between EDP-centres of Austrian universities

AS	Adaption-System
LAN	Local Area Network
DHC	EDP-Centre, Department for Hybrid Computing
DPC	EDP-Centre, Department for Process Computing

3.2 Implementation of the Communication Software of an Adaption-system

The software of an adaption-system has a structure as already described in chapter 2. On the wide-area-side it contains network access software for DATEX-P as well as communication programs for the higher layers according to the OSI-standards.

From the higher layers only the transport layer has been implemented in the pilot project for the present. That means, that the application programs in layer 7 use the services of the transport layer in a direct way. Because of the absence of the session- and the presentation layer the functions of these layers must be performed by the application programs themselves for the present. The project group of the Technical University of Vienna implemented the communication software for the wide area for all adaption-systems. To connect the adaption-systems to DATEX-P existing communication software according to CCITT X.25 could be used. The user interface of this software was adapted so that network connections according to X.25 could be used just like logical devices as provided by the operating systems RSX-11M and RSX-11S respectively.

Because DATEX-P, which carries the supra-regional interconnection, is a data network with a high quality, the simplest protocol class, namely protocol class 0, was sufficient in the transport layer.

3.3 Implementation of the Communication Software to Integrate a Subnetwork

As a rule the communication software applied in the local area is from a certain manufacturer and therefore a given fact. The main task of the adaption program in layer 7 in the adaption system is to realize a relay function. This can be done in cooperation with proper software-modules within the endsystems by transforming the user-interface from the adaption system into the endsystems. For this purpose a set of subroutines has been created for the experimental interconnection which reside in the endsystems of the DRDP's subnetwork and cooperate with the adaption program in the adaption-system according to the rules of an especially developed application protocol on top of DECnet. This enables the application processes within the endsystems to use the services of the transport layer in the adaption-system which offer a supra-regional information transport.

Special attention must be directed to addressing. The administration of the open, supra-regional interconnection must specify proper addresses so that an application program A in endsystem B of subnet C can establish a connection to an application program X in endsystem Y of subnetwork Z. In the experimental interconnection hierarchically constructed transport addresses have been defined which uniquely specify the source and destination of a transport connection (layers 5 und 6 not yet implemented).

3.4 Support for Adaption Systems by a Central System within the Interconnection

If adaption-systems are employed which are, for financial reasons, not equipped with mass storage devices, additional support must be given to them within the interconnection. So it is reasonable to store the operating systems and all necessary programs of all adaption-systems on a disk of a central system. If required by an adaption-system its operating system and its communication software can be loaded from the central system into the adaption- system via the carrier network (e.g. DATEX-P) of the interconnection.

As mentioned above it is sensible to use systems of the same type as adaption-systems. Thereby the problem may arise that an adaption-system is a heterogeneous system within its area and does not find the necessary local support for program development. To enable the group which is responsible for the concerned area to develop adaption and communication programs according to the local requirements, central support for program development for adaption-systems must be available. In the experimental interconnection this problem was solved thus: Users of a region have access from their terminals to the VAX-11/780 of the DRDP via DATEX-P and can develop the software for their

adaption-system there. After that the adaption-system concerned can be loaded with its operating system and all the communication and adaption programs. For the downline loading of an adaption-system via DATEX-P a suitable bootstrap loader for PDP-11/23 Plus systems has been developed and stored in EPROMs. This bootstrap loader is activated automatically if the adaption-system is switched on or if a restart is performed. It establishes a network connection to a loader task of the central adaption-system and loads the operating system and all programs from the disk of the central adaption-system via DATEX-P into the adaption-system of the region concerned using a simple loading protocol.

4. EXPERIENCES AND RESULTS OF THE EXPERIMENTAL INTERCONNECTION

The adaption-system approach avoids unnecessary parallel work and enables a clear separation of responsibilities:

The enterprise's region concerned has the responibility for:

- the development, maintenance and documentation of the communication software for its local subnetwork and
- the connection of an adaption-system to the local subnetwork.

A central organization of the enterprise has the responsibility for:

- the adaption-systems, their maintenance and the supra-regional communication software within them,
- the maintenance and further development of this software,
- the creation of new services for local regions of the enterprise,
- the administration of the services of the open interconnection and
- the preparation and updating of documentation regarding the supra-regional interconnection and its available services.

It is also incumbent upon this central organization to specify and document exactly the demarcation line between the supra-regional and the local communication programs for each service, which is made available for a region of the enterprise.

At the moment ISO's work on the standards, especially of the layers 6 and 7 has not reached a stable state. That means that modifications of already implemented communication programs must be still take place. In doing so the adaption-system approach appears to be very favorable, because such necessary adaptions can be made by a central team.

Additionally the adaption-system approach enables the integration of small systems with special resources in the interconnection without over-loading them by extensive supra-regional communication software and thus disturbing their operation in any unacceptable way.

Generally the transformation of the OSI-user-interface from the adaption-system into the endsystem to integrate a subnetwork of an enterprise's region in the interconnection, will reduce the quality of the communication service. The amount of the loss of quality depends on the quality of the communication service which is available within the subnetwork. With the example of the integration of the DRDP's DECnet-subnetwork in the experimental interconnection it could be demonstrated that this loss of quality can be minimized if the application protocol for the information exchange between the adaption-system and the endsystems of the region has been worked out very carefully.

5. FINAL REMARKS

The work to establish an experimental interconnection of EDP-centres of Austrian universities has been sponsored by the Austrian Federal Ministry for Science and Research, section research, by three coordinated research projects. Within the framework of an administrative cooperation agreement between the

Federal Ministry for Transport and the Federal Ministry for Science and Research the Austrian Postal Authorities made available the necessary DATEX-P connections free of charge.

REFERENCES

[1] Färber, G.: Konzepte und Anwendungen von Feldbussystemen mit verteilten Prozeßperipherie-Modulen. Interkama 83. Fortschritte durch digitale Meß- und Automatisierungstechnik. (Springer-Verlag, Berlin 1983) 495-510.

[2] MacWilliam, P.D. u.a.: Microcontroller serial bus yields distributed multispeed control. Electronics 57 (1984) 125-130.

[3] Fromm, I.: Local Area Networks. Bull. SEV/VSE 74 (1983) 11, 585-589.

[4] Shotwell and Associates: The Ethernet Handbook, 2. Edition. (North-Holland Amsterdam, New York, Oxford 1983)

[5] Park, C., and J. Lynch: Relative Performance of Token versus CSMA/CD. (Hewlett Parkard, Palo Alto 1980)

[6] ISO/TC 97: Information Processing System - Open Systems Interconnection- Basic Reference Model. Draft International Standard ISO/DIS 7498, Febr. 1982.

[7] Kunft, W.: Akademisches Computernetz in Österreich. ÖHZ 36 (1984) 7/8 39-40.

[8] Paul, M., und W. Kunft: Zur Integration regelungstechnischer und kommerzieller Informationssysteme. E und M, 102 (1985) 1, 31-36.

MECHANISMS OFFERING MULTIPOINT SERVICES ON LANs INTERCONNECTED BY SATELLITE

Otto Carlos DUARTE
Project NADIR c/o INRIA
UNIVERSIDADE FEDERAL DO RIO DE JANEIRO
Depto. de Eletrônica-Ilha do Fundao
Rio de Janeiro - RJ -BRAZIL

Pierre de LA MOTTE
Project NADIR c/o INRIA
Rocquencourt B.P.105
78153 Le Chesnay Cedex
FRANCE

The TELECOM1 satellite system offers to european organizations the possibility of interconnecting their remote sites. These sites will probably be equiped with LANs. One must take advantage of the broadcasting capabilities of both satellite and LAN to offer multipoint services. This paper presents four mechanisms which take into account the different application needs. Two of these mechanisms were studied and developped within the NADIR project and they are described here in detail.

1- INTRODUCTION

Computer networks began to arouse considerable attention in the mid-1970's. The commercial potential of packet switching, satellite and local area networks (LANs) technology and the declining cost of hardware have made computer network an important field.

In 1984, the French PTT have launched the TELECOM1 satellite. The main objective of this system is to offer digital links to the European organizations.

NADIR is a French research project (started in 1981) that aims at investigating new types of computer applications using satellite links. One of these research works is the interconnection of LANs through a multipoint satellite link (fig. 1).

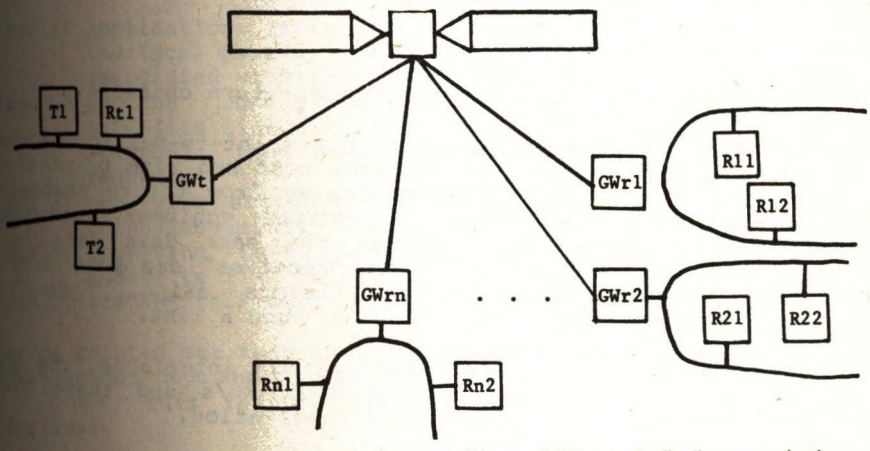


Fig.1- General configuration: one sending LAN and "n" receiving LANs

After a brief presentation of the TELECOM1 satellite system, this paper describes four protocols architectures offering services with different quality of services depending on the application needs. These architectures conform the functional structures defined by the Open System Interconnection Reference Model.

2- THE TELECOM1 SYSTEM

The objectives of the TELECOM1 satellite system [1] are:

- to provide flexible digital communication capabilities to French and European organizations (business service);
- to provide telephone and television links between mainland and overseas French departments;
- to develop videocommunication;
- to provide militar applications.

Among these objectives, the business service is the most important and five 25 Mb/s transponders are dedicated to it. These 20 W transponders are working in the 14/12 GHz frequency band, allowing the use of small antennas (3.5 m).

Access to satellite channels is made by means of a Time Division Multiple Access (TDMA) with demand assignment. This mechanism shares out the total channel capacity among terrestrial transmitting station (on a demand assignment basis).

2.1 - Business Service

The Business Service provides customers with three kinds of digital switching circuits:

- simplex or full-duplex point-to-point links;
- simplex point-to-multipoint links;
- point-to-multipoint links with "summed" return channels.

This third kind of link is made up of a point-to-multipoint channel in the transmitter-receivers direction, combined with point-to-point return channels. Information flows coming from return channels are "ANDed" when reaching terrestrial transmitter equipment, and hence at the access interface, the link has the same data rate in both directions (Fig. 2). Thus, only one "receiver" can send data at a given time, otherwise logical collisions arise. Up to 32 participants can be interconnected using such a link.

These links are accessed by X21, V35, or V24 interfaces. The offered data rates range from 2400 b/s to 1920 Kb/s and there are two establishment modes: call-by-call and reservation.

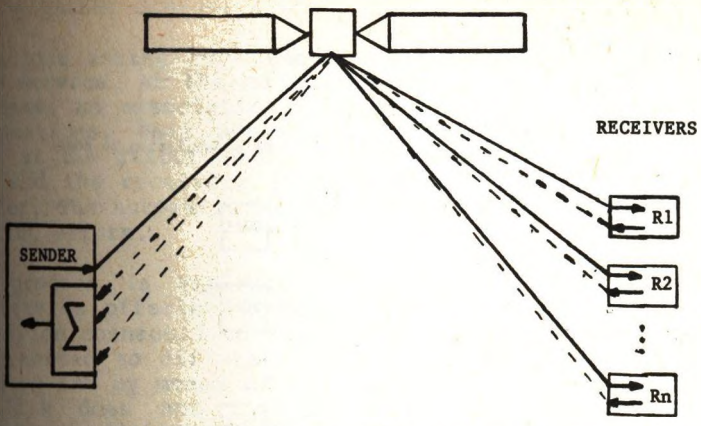


Fig. 2- Point-to-multipoint link with "summed" return channels

3- THREE STEPS COMMUNICATION

The first step consists in locally accumulating data to be transmitted in the sending LAN gateway. During the second step, when the satellite link is set up, these data are transmitted towards receiving LANs gateways. Then, within these LANs, gateways broadcast the received informations (Fig. 3).

This mechanism allows a more efficient utilization of the satellite link and consequently reduces its effective cost.

This link may be set up either when enough amount of data is accumulated in the transmitting gateway (call-by-call set up) or at set times (semi-permanent fashion). Set times can be also selected according to off-peak hours rates (three times cheaper).

On the other hand, a drawback of such a mechanism is that it is not possible for a transmitting application to have short-time acknowledgement (positive or negative) for the transfer. If an acknowledgement is necessary, it can be built from a sequence of partial acknowledgements as follows :

- stage 1: applications asking for a transfer set up a connection with the local gateway and transmit to it all the concerned data associated with the transfer acknowledgement request;
- stage 2: sending LAN gateway sets up a point-to-multipoint satellite link with receiving LAN gateways and transmit towards them both data and transfer acknowledgement request;
- stage 3: receiving stations provide receiving LANs gateways with acknowledgements;
- stage 4: receiving LANs gateways take advantage of the next satellite link to send acknowledgement responses;
- stage 5: transmitters receive acknowledgements.

It may be pointed out that the time between the data transfer and its acknowledgement depends on the time between two successive satellite link set-ups. This feature cannot meet the requirements of some applications.

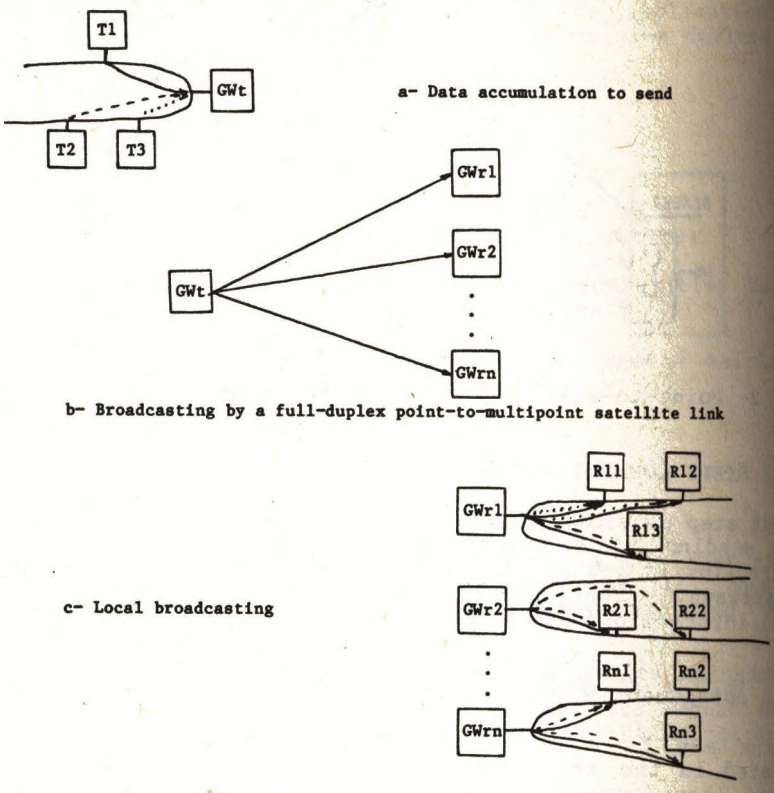


Fig 3 - Three Steps Communication

4- INFORMATION TRANSPARENCY vs TIME TRANSPARENCY

The X25 protocol,used in packet switching networks, guarantees the transferred data integrity but the response time is not assured. Applications [2] such as digital voice transfer, image transfer or remote real time control ask for a regular throughput but can tolerate possible errors in the information if the frequency of these errors is kept under a certain threshold.

Thus, in case of error in a semi-animated image transfer it is better to use a local mechanism which may display the preceding line instead of the lost one, than to wait 600 ms (two times the propagation delay of a satellite link) for a retransmission. This approach is based on a slow variation of information in adjacent lines. Possible approaches to control process applications and voice sequence transfer are, respectively, the repetition of the preceding state of the transducers, and the introduction of silences.

Neither the typical connection-mode service nor the connectionless-mode service suit these applications. The former one does not guarantee a regular throughput and the latter one neither guarantees the data sequence nor the throughput.

5 - TIME TRANSPARENT COMMUNICATION

The applications asking for time-transparency can be supported by a stream-mode service. At the establishment of the communication, this service creates an association between the sending station and the receiving stations. This association aims above all two things : reservation of the resources to insure a regular throughput between the sender and the receivers ; and establishment of a path for the data transfer. The Automatic Repeat Request (ARQ) scheme is not used for correcting errors.

The NADIR project is implementing a centralized multipoint data transport service offering a stream-mode service [3]. This service allows a station connected to a LAN to sending messages to many other stations connected to different LANs. The remote LANs are connected to the sending LAN by means of a point-to-point satellite link. This satellite link does not have any return channels, and so, no senders-receivers dialogue is possible. Thus, the time association which is created does not require any response from the receivers. There will be neither "connect response" nor the "disconnect request" send by the receivers. Obviously, no kind of negotiation and no communication control are possible.

The association is established (or released) by means of one or many protocol messages emitted by the sending station at the beginning (or at the end) of the communication. These "BEGIN" (or "END") messages guarantee the reservation (or the release) of the necessary resources for the communication. We can take advantage of this association to use an identification mechanism [4] that will allow to avoid overheads due to addressing resolution and transmission.

If all the sites need to transmit information, then a "token" mechanism [5] can be employed in which each LAN, by turn, becomes the sending LAN. In this case, a report of the preceding received transfers is sent by each station when the LAN, to which it is connected to, becomes the sender.

5.1 - Communication bit error rate

The global communication Bit Error Rate (BER) depends on each subnetwork BER. The BER of ETHERNET LAN type is better than 10 E-9 and the expected one for TELECOM1 is better than 10 E-9 during 90% of time [6]. The LAN's physical BER specified in IEEE 802 is less than 10 E-8 and the equivalent one for TELECOM1 is less than 10 E-6 during 99% of time.

In the architecture described here for time transparent communication there is no return channel and so, an Automatic Repeat Request (ARQ) error correction scheme is not possible. In spite of this, the BERs are good enough for a lot of applications.

5.2 - Flow control

This communication is performed through CSMA/CD LANs offering a 10Mb/s data rate (ETHERNET) and interconnected by a 64Kb/s (or 128Kb/s) satellite link. As data retransmission is impossible, we must avoid any loss of message due to the difference of data rates through the interconnected subnetworks.

Theoretically, we cannot guarantee a constant throughput in a LAN using CSMA/CD access technique. But, we make the assumption that the

LANs are available enough to support the data rate imposed by the satellite link. Furthermore, we will consider that the gateways can receive, process and transmit information at the data rate of the satellite link.

So, data flow is critical between the sender and the sending gateway and it must be controlled.

5.3 - Used protocols

At physical and data link layers of the LANs, we will use the connectionless mode protocol defined by IEEE 802 because this protocol is the only standardized one that provides multipoint communications.

The satellite physical layer is based on the X21 CCITT recommandations, which has been adapted to point-to multipoint circuit. At the data link layer a simplified HDLC procedure (no flow control and no error correction) is imposed by the lack of return channel (Fig. 4).

The transport and network layers use stream-mode protocols which guarantee the reservation of the necessary resources.

The applications using this service know "a priori" the main characteristics of the satellite link, for instance, the maximum offered throughput. The sending station cadences the transfer by means of a timing mechanism.

A more flexible service will be offered by introducing the multiplexing facility. This facility permits to share the satellite link among the stations connected to the sending LAN. In this case, the network protocol negotiates a throughput at the establishment of the association. The gateway response allows the sending station to know the instantaneous amount of spare satellite resources.

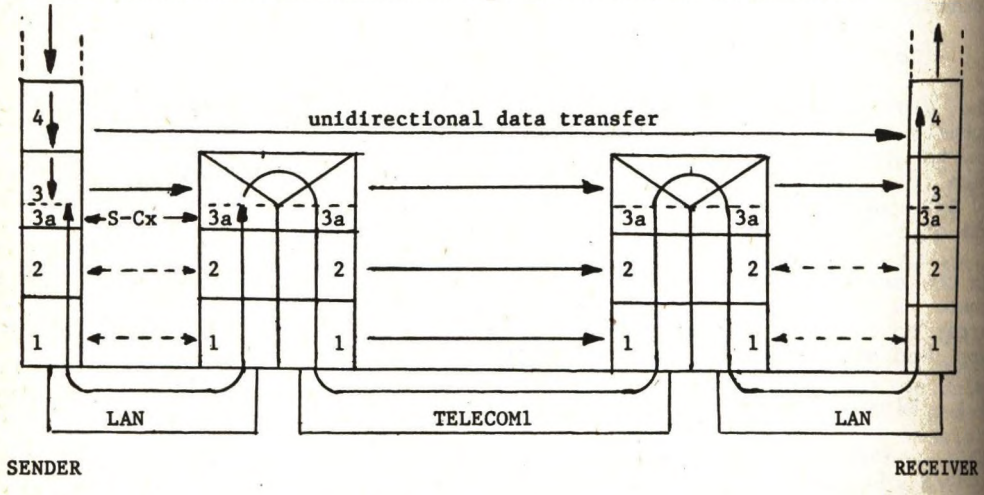


Fig 4 - Stream mode communication architecture (only one transmitter and one receiver is represented)

The throughput offered to the sending application must consider overheads due to the protocols (heads,FCS and zero insertions

realized by the bit transparent HDLC procedure). The timing flow control can be regulated to the worst overhead condition to avoid congestion in the sending gateway, but this approach leads to a satellite channel subutilization.

A more efficient flow control performed by means of a sub-network protocol (only at the sending LAN) is possible, but the time-transparency could be no longer respected.

6 - INFORMATION TRANSPARENCY COMMUNICATION

Bulk transfer, file transfer, coded image transfer, remote job entry are applications that claim for data integrity. They ask for a correct and ordered data transmission.

A service to be offered to these applications must solve all errors and flow problems.

6.1. Error correction

In order to minimize the communication residual bit error rate (BER), an Foward Error Correction (FEC) code can be used. Nevertheless this kind of approach does not allow to recover a frame which have been lost due to a flow problem (lack of memory resources at the gateway, for instance).

Another approach based on the transmission of redundant information [7] allows to recover, within certain limits, even an entire lost frame. Assuming that there is no correlation between the frames lost by each receiver, this multipoint recovery method shows a better efficiency when compared with the classical method which retransmits the non unacknowledged frames, after a first transmission, in various phases until all the frames will be acknowledged by all the receivers. This better efficiency is obtained with a greater complexity and the use of more processing time by the sender and the receivers.

Moreover, the continuous Automatic Repeat Request (ARQ) procedures (reject and selective reject) have been extended to be used in a multipoint satellite environment.

Finally, the error correction procedure using negative acknowledgements performs well for high data transmission in multipoint mode. The negative acknowledgements are often used associated with to surveillance and final positive acknowledgements [8]. The surveillance acknowledgements allow to distinguish wether a silent receiver is receiving all the frames correctly or it is off.

6.2 - Example of information transparency communication

The NADIR project plans to implement a centralized multipoint connection-mode transport service.

The aimed data rate (64 Kb/s or 128 Kb/s) and the low BER of the used networks encourage us to use a simple recovery mechanism. Thus, an extended multipoint version of the HDLC procedures will be used to realize both the error recovery and flow control.

We ought to use a bidirectional point-to-multipoint satellite link to offer this service. As it has been said before, whithin the TELECOM1

system, this type of link has ANDed return channels and so collisions can take place.

The synchronisation of the return channels implies hardware problems which we do not want to solve now. So, we will use a slightly modified "pure ALOHA" mechanism. This mechanism reaches its maximum efficiency (18%) when the offered traffic is equal to half the total channel capacity [9]. Furthermore, a collision is detected by the broadcasting gateway only when the return channels are ANDed, so the retransmission must be requested by this gateway or sent after elapsing a time-out.

So, we will minimise the return channel traffic, limiting it to short control messages. There is a trade off between the collisions and the frequency of acknowledgements needed for a proper flow control.

Although a bilateral satellite link is employed here, the information transfer is unidirectional (sender to receivers). The same mechanism described in paragraph 5 must be used if all the sites need to send informations.

6.2.1 - End-to-end control vs hop-by-hop control

We compare in this section the advantages and inconvenients of an end-to-end control (transport layer) to a hop-by-hop control (network layer).

These controls take place at different layers and so, they have different functions. An end-to-end control insures a good transfer from the sending station to the receiver stations while the hop-to-hop control guarantees a sub-network transfer.

An error correction mechanism at both network and transport layers seems heavy and redundant for a lot of applications. Hence we will apply flow control and error correction at only one of these layers.

The two majors advantages of an end-to-end control are :

- it assures an end-to-end transfer ;
- its utilization implies simpler gateways because flow control and error correction are performed at the extremities, by the sending station and by the receiving stations.

Nevertheless, the hop-by-hop control (at each sub-network) (fig. 5) has the following advantages :

- the frames are acknowledged at each sub-network, and hence the satellite return traffic does not depend on the number of receiving stations but on the number of receiving LANs. So, less collisions and a better response time can be obtained;

- The satellite return traffic is symmetrical;

- The overhead due to the retransmission of an erroneous message takes place only through the sub-network where the error has occured;

- The shorter response time implies a better flow control;

- An easier and more efficient multiplexing of the satellite link can be implemented.

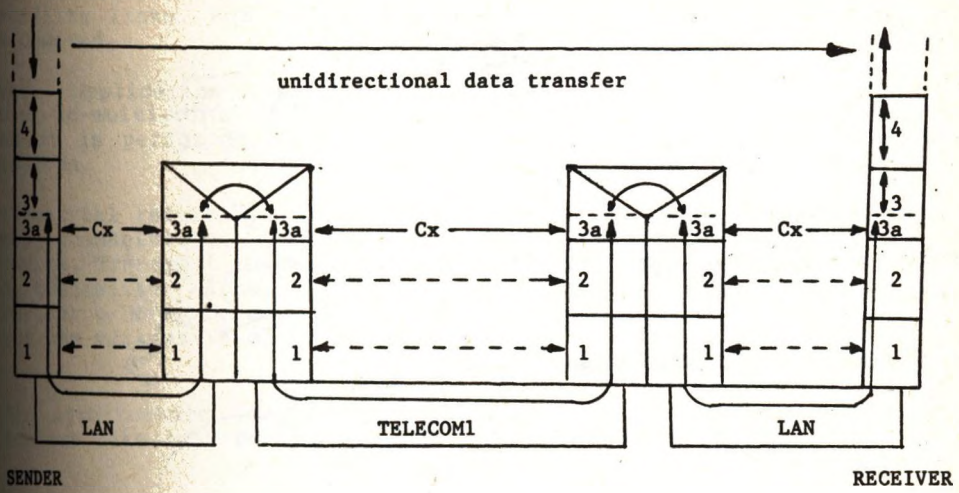


Fig 5 - Hop-by-hop Network control (only one transmitter and one receiver is represented)

6.2.2 - The used protocols

At physical and data link layers connectionless IEEE 802 protocols are used (see paragraph 5.1.3.)

The CCITT X21 procedures were adapted to the bilateral point-to-multipoint (with "ANDed" return channels) satellite environment. The satellite data link layer uses a simplified HDLC protocol.

All sub-networks employ a connection mode protocol performing a sending window flow control and a go-back N error correction procedure.

7 - COMMUNICATION THROUGH MIXED NETWORK

Another solution, currently under study, is the use of TELECOM1 network combined with terrestrial links. We call this combination "Mixed Network". Terrestrial data networks, as TRANSPAC packet switching network in France, provide services that are complementary to those provided by satellite network : conversational oriented service and low or medium data rates. Although communication system based on such a combination increases in complexity, it is flexible and it meets all the external communications requirements of a LAN (but telephony).

7.1 - Studied configuration

The study is based on two subnetworks : TELECOM1 and TRANSPAC. This architecture allows a station to communicate with others either through TELECOM1 or through TRANSPAC (Fig. 6) or through both

subnetworks at once.

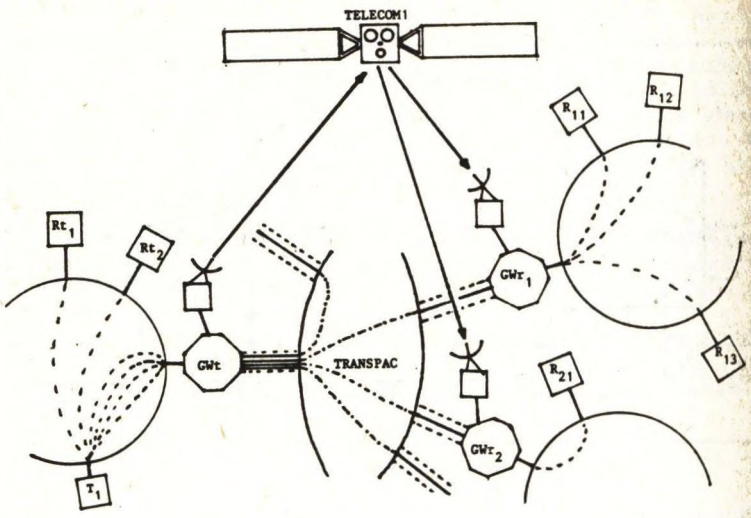


Fig 6 - Mixed Network

Communications are achieved through a gateway which gives on access to the two subnetworks (multinetworks gateway Fig. 7), from one hand through a connection with a TELECOM1 station, from the other hand through a link with a TRANSPAC access point. Several logical channels can be multiplexed on this link. In fact, the transmitting gateway manages as many virtual circuits as there are receiving LANs.

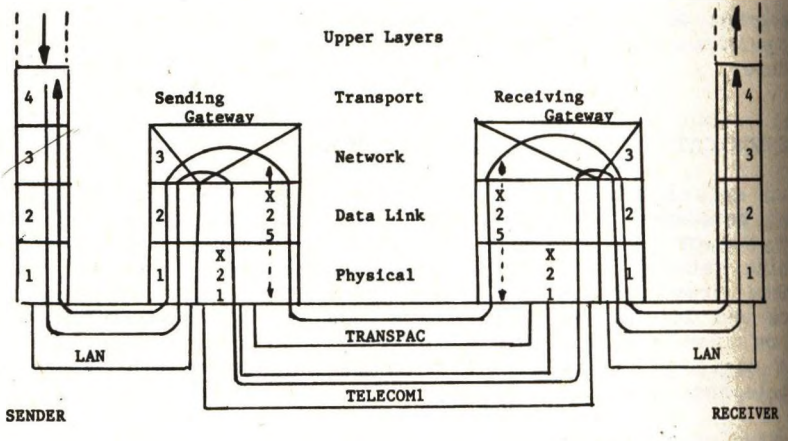


Fig 7 - Protocol Architecture of the Mixed Network (only one transmitter and one receiver is represented)

7.2. Synchronous use

This architecture is similar to the one based on duplex multipoint satellite links. But the drawbacks of collisions on return channels is removed.

From application standpoint, this structure provides a point-to-multipoint data transfer. This transfer is unidirectional and it is performed from link initiator station towards the other stations.

Terrestrial return channels are used only for communication control. This communication is supported by an unidirectional Transport Service. Transport protocol is connection-oriented and is in charge, on transmitter side , of receivers supervision. Network Layer provides a Network Service based on a hop-by-hop control performed over the various involved links.

This architecture leads gateway to resolve non trivial management problems like synchronisation between both subnetworks.

8. CONCLUSION

The choice of the protocol architecture to be used for interconnecting LANs through TELECOM1 depends on the concerned application and the required Quality Of Service.

All communication services which are described in this paper provide unidirectional data transfers from one transmitting station towards several receiving stations. It is due to the centralized way of the multipoint link provided by TELECOM1 network.

For informations which can be transmitted in a deferred way, applications may use of the "three steps" protocol architecture, thus taking advantage of the satellite link off-peak hour rates.

The respect of data integrity implies the use of a duplex link. As collision can occurs among satellite return channels , we preconize hop-by-hop error correction and flow control, to minimize and to make symmetrical the traffic on these channels.

"Time transparent" oriented communications take advantage of the satellite broadcast ability without needing duplex links and, by that, satellite link cost is cut down by half (TELECOM1 rate).

When the use of terrestrial links is allowed (for example when the links are already used for communication between LANs) we can use mixed network architecture eliminating by that the collisions problems.

9 - BIBLIOGRAPHY

[1] D. LOMBART, P. RAMAT, F. RANCY - "Télécom1 : multiservice network, actes du congrès satellite et téléinformatique", pp.3 - 11, Versailles, France, Avril 83.

[2] C. HUITEMA - "Réseaux locaux et Télécom1", Congrès de Nouvelles Archi tectures pour les Communications, pp. 319-324, Paris, France, Sept.84.

[3] O.C. DUARTE, J. RADUREAU - "Interconnection of local area networks through a multipoint satellite link", Networks 84, Madras, Inde, 1984.

[4] O.C. DUARTE - "Mécanismes d'identification de connexion multipoint", Congrès de Nouvelles Architectures pour les Communications, pp. 279-284, Paris, France, Sept.84.

[5] P. de LA MOTTE, I. VALET - "Multidestination channels on satellite systems". Proceedings of the 7th ICCC, pp. 527-532. Sidney, Australie, Nov. 84.

[6] P. RAMAT, C. HUITEMA - "Répartition des taux d'erreur sur les liaisons satellites Télécoml et l'influence sur des protocoles de transmission", CNET, note technique, 20p, 1983.

[7] J.J. METZNER - "An improved broadcast retransmission protocol", IEEE Trans. Comm., Vol. Com 32, N°6, pp. 679-683, June 1984.

[8] H. FALLOUR and AL. -"Transport de masses d'information en multipoint à haut débit par satellite". NADIR SPE 3.503.2, 38 p. Déc. 81.

[9] N. ABRAMSON - "The ALOHA system, another alternative for computer communications", AFIPS Proc. FJCC conf., vol. 37, 1970.

ONE RELATIONAL MODEL OF LAYERED NETWORK ARCHITECTURES

Radoslav Pořízek

Institute of Applied Cybernetics
844 16 Bratislava, Hanulova 5A
Czechoslovakia

A relational model of network architectures is introduced as a new methodological tool, or a framework, for analyzing, comparing, and specifying of network architectures. Services, communication functions, methods of communication functions, communication protocols, and modules were chosen as basic network architecture elements. Assignments of these elements to layers and existence of various dependencies between the elements are expressed by means of appropriately defined binary relations yielding sufficiently structured but yet clear specification of a network architecture. The structure of influence of the network architecture elements on network performance is covered as well.

INTRODUCTION

Network architectures are complex systems containing too many elements at various levels of formal or functional hierarchy, with too many dependencies existing among these elements. Network architectures may be, and are, specified at various levels of detail /e.g. at levels of services, functions, and protocols/, depending on the purpose of the specification. Though verbal specifications prevails, [1, 2, 3] several formal specifications of network architectures have also been elaborated, aimed at different goals. It is due to the different requirements that various specification approaches and tools have been developed, and all are used. The relational model of network architectures described in this contribution is just one more such tool. It should provide for some new views, and for more comprehensive but yet clear specification of network architectures.

Quite often two network architectures are compared in a "geometric" way, as in Fig. 1. This approach, though very expressive and quite sufficient for several purposes does not provide for defining any details. On the other hand, detailed comparison of all communication functions and even the mechanisms used /see e.g. [4] / is very extensive and may result in a loss of the overall imagination about the network architecture and its influence on a network performance, In order to prevent such a loss, and to add more semantics to schemes like Fig. 1, the relational model has been devised giving, hopefully, a useful framework.

Also, a lack of the formal definition of a network architecture

reminding the definition of, say, a finite state machine or a communication protocol [5], gave an impetus to the work on the relational model of network architectures.

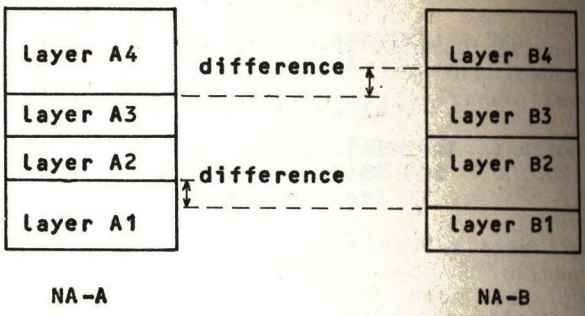


Figure 1
A very simplified, "geometric" comparison of two network architectures
/NA = network architecture/

THE BASIC RELATIONAL MODEL OF LAYERED NETWORK ARCHITECTURES

Further, a network architecture will be defined as a mathematical system consisting of several sets, containing assigned basic network architecture elements, and several binary relations, expressing both the assignment of the elements to individual layers, and various dependencies between the elements in individual layers, and also between adjacent layers.

$$na = \left(\left[L, \leqq \right] ; SV, CF, MCF, CP, MD; \sigma, \varphi, \varsigma, \pi, \mu; \delta, \varkappa, \lambda; \alpha \right), \quad (1)$$

where $[L, \leqq]$ is a linearly ordered set $L = \{l_1, \ldots, l_N, \ldots, l_R\}$ of R layers; SV, CF, MCF, CP, and MD are sets of basic network architecture elements assigned to the network architecture na; $\sigma, \varphi, \varsigma, \pi$, and μ are binary relations expressing the assignment of the elements to individual layers; δ, $\varkappa$, λ are systems of binary relations expressing dependencies among elements in individual layers; and α is a system of binary relations expressing the dependencies among elements in adjacent layers. The graph representation of the relational model is in Fig. 2.

SV, CF, MCF, CP, and MD are the sets of services, communication functions, methods of communication functions, communication protocols, and modules, respectively, assigned to the network architecture na.

σ, φ, ς, π, μ are binary relations expressing the assignment of services, communication functions, methods of communication functions, communication protocols, and modules, respectively, to individual layers, i. e. the following binary relations:

$$\sigma \subset L \times SV, \quad (2)$$

$$\varphi \subset L \times CF, \quad (3)$$

$$\varsigma \subset L \times MCF, \quad (4)$$

$$\pi \subset L \times CP, \qquad (5)$$

$$\mu \subset L \times MD. \qquad (6)$$

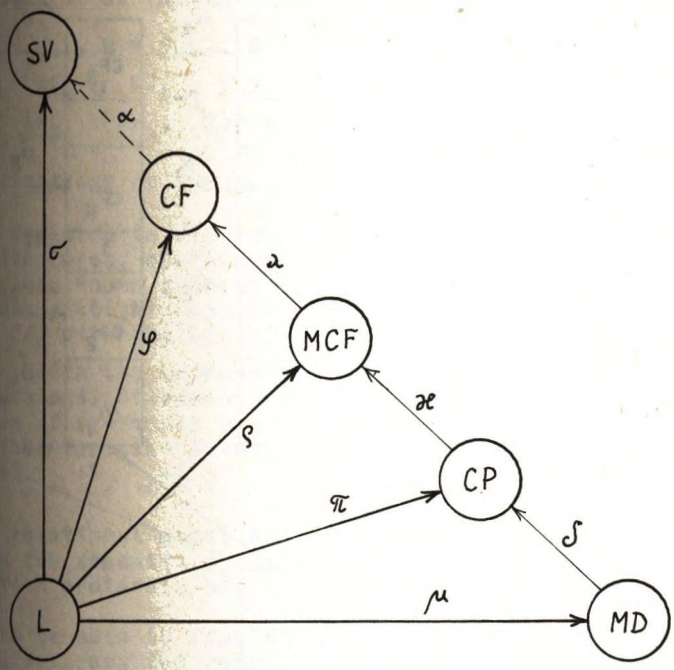


Figure 2
The relational model of a layered network architecture

The binary relations σ, φ, ϑ, π, μ assign to each layer $\ell_N \in L$ the subsets SV_N, CF_N, MCF_N, CP_N, MD_N respectively, where $SV_n \subset SV$, $CF_N \subset CF$, $MCF_N \subset MCF$, $CP_N \subset CP$, $MD_N \subset MD$.

It means that each layer ℓ_N, N = 1,...,R, may be specified as follows:

$$\ell_N = (SV_N, CP_N, MCF_N, CP_N, MD_N). \qquad (7)$$

The structure of these assignments is clearified by Fig. 3.

The main difficulty with sets, and subsets, of basic network architecture elements lies in "fuzzyness" of their definition, because there are no generally accepted lists of services, communication functions, communication protocols, etc. But there are important efforts in this area, e. g. the lists of services and functions in the RM OSI standard [1]. Moreover, the relational model, as a methodological tool rather than a simple reflection of reality, expresses what should be known about network architectures.

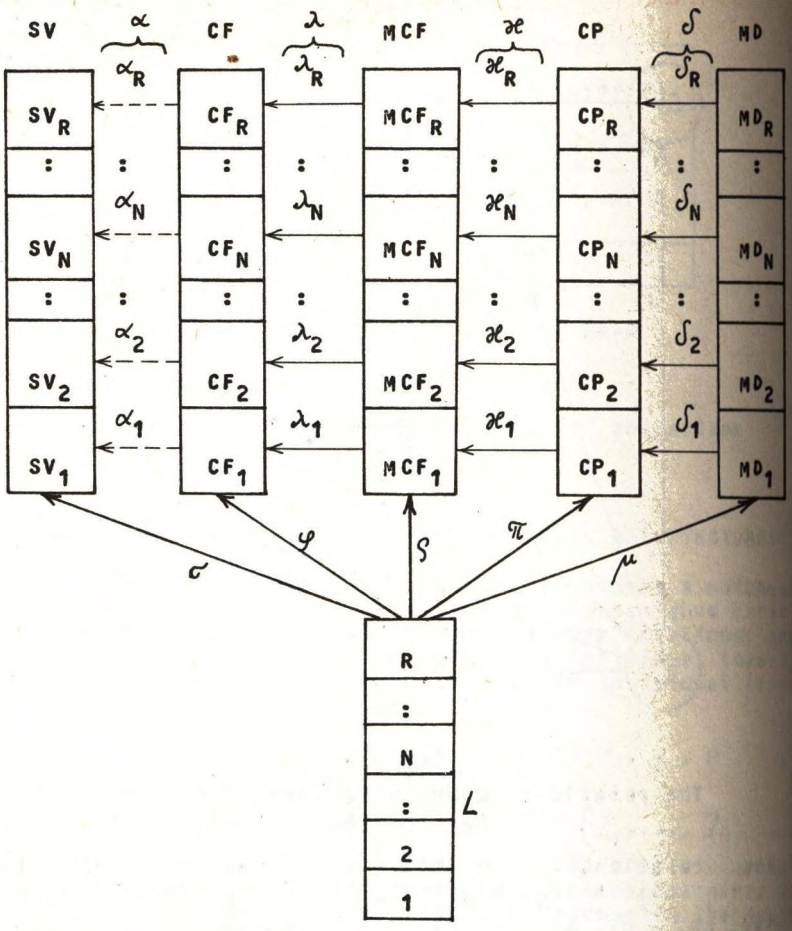


Figure 3
The structure of the binary relations with respect to layers

δ , $\varkappa$, λ are systems of binary relations δ_N, $\varkappa_N$, λ_N expressing existence of dependencies between modules and communication protocols, communication protocols and methods of communication functions, and methods of communication functions and communication functions, respectively, i. e. the following systems and binary relations:

$$\delta = \{\delta_N\}, \quad \varkappa = \{\varkappa_N\}, \quad \lambda = \{\lambda_N\}, \quad N = 1,\ldots,R; \tag{8}$$

$$\delta_N \subset MD_N \times CP_N, \tag{9}$$

$$\varkappa_N \subset CP_N \times MCF_N, \tag{10}$$

$$\lambda_N \subset MCF_N \times CF_N. \tag{11}$$

α is the system of binary relations α_N expressing the existence of dependencies between services required by a higher layer, and communication functions assigned to a lower layer together with services rendered to this lower layer by adjacent, even more lower, layer; i. e. the following system and binary relation:

$$\alpha = \{\alpha_N\}\ ,\ N = 1,\ldots,R; \tag{12}$$

$$\alpha_N \subset (CF_N \cup SV_{N-1}) \times SV_N,\ \text{for } N = 2,\ldots,R,$$

$$\alpha_N \subset CF_N \times SV_N,\ \text{for } N = 1. \tag{13}$$

SOME MODIFICATIONS OF THE BASIC RELATIONAL MODEL

The model depicted in Fig. 2 primarily serves to analysis of network architectures, characterized by known detailed elements and, possibly, less known aggregate elements. For synthesis of network architecture a slighty different relational model is to be used /see Fig. 4/, with the binary relations δ_N, $\varkappa_N$, λ_N and α_N /and, also, with the systems of these relations/ having reversed their directions. Of course, not only directions are reversed in the process of synthesis but the content of relations is changed as well. Therefore the notation is δ', $\varkappa'$, λ', α' /and not δ^{-1}, $\varkappa^{-1}$, λ^{-1}, α^{-1} /.

The basic relational model contains several semantic redundancies reflecting the semantic overlaying of basic elements, e. g. of communication protocols and communication functions.

In order to be able to cope with all real specifications of network architectures, the redundancy may even be enhanced, as shown in Fig. 5.

The resulting enhanced definition of a layered network architecture is as follows:

$$na = \left([L, \leq]\ ;\ SV,\ CF,\ MCF,\ CP,\ MD;\ \sigma, \varphi, \varsigma, \pi, \mu;\ \delta, \varkappa, \lambda, \beta, \omega, \psi;\ \alpha, \xi\right) \tag{14}$$

The meaning of the binary relations β, ω, ψ, and ξ is obvious. Such a redundant scheme may assist to find both missing and redundant elements in the specific description of a network architecture.

EXPRESSING THE INFLUENCE ON PERFORMANCE

Let us define the set PF containing network performance variables and, also, network capabilities. These variables and capabilities are influenced, besides others, by a network architecture. This influence may be structured as shown in Fig. 6. Again, binary relations Γ, χ, Θ, Ψ, and E are used to express the existence of the influence of individual network architecture elements. These relations show, which performance variables are influenced by which network architecture elements. They do not show the nature of the dependencies, only their existence. Even so simplified, the whole scheme is very complex. But the structure of existing relations is expressed clearly.

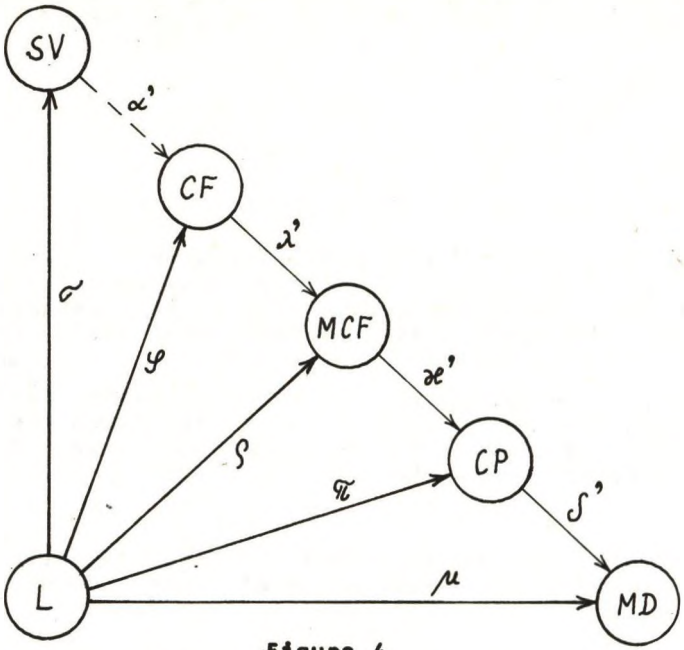


Figure 4
Synthesis oriented relational model

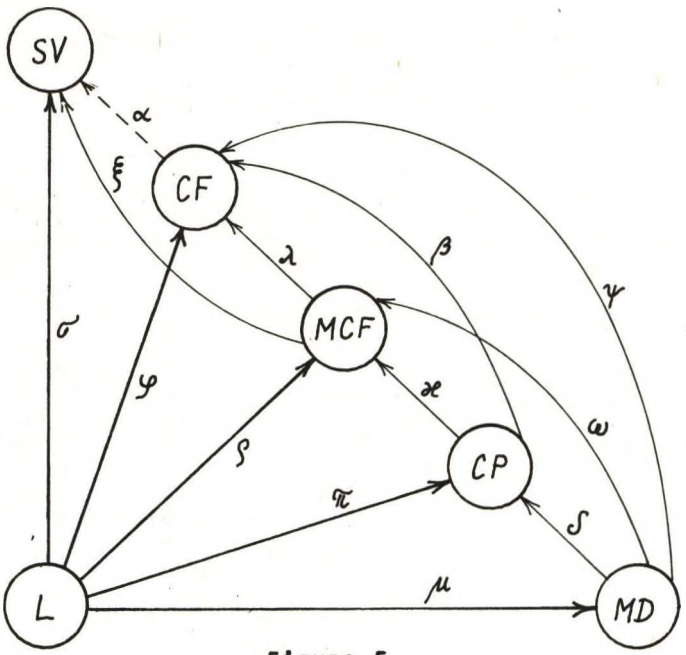


Figure 5
Redundant relational model of network architectures

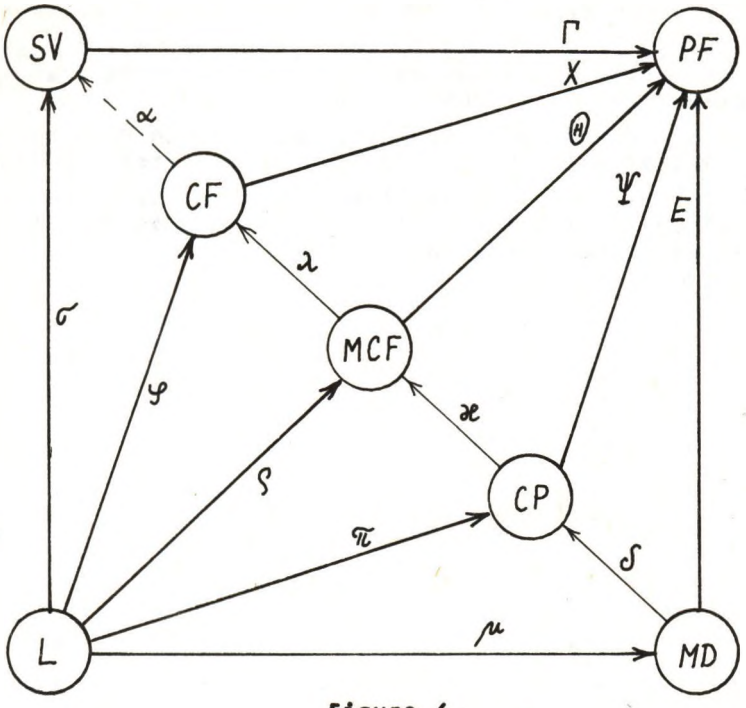


Figure 6
Structuring of influence of network architecture on performance

CONCLUSION

The specifications of network architectures may be more or less detailed, according to the purpose of the specification. The most detailed specification /at the module level/ is, of course, required for the implementation and for the product description. Such specification is very extensive.

The different approach to the expressing of network architectures by means of binary relations is described in [7] .

The described relational model of network architectures may serve as a framework enabling to keep up the overall picture of such complex system. It also enables to define the network architecture in a formal and exact way which may lead to removing of at least some ambiguities.

REFERENCES:

[1] Information Processing Systems - Open System Interconnection Basic Reference Model. IS 7498. /ISO, 1983/.

[2] Jakubajtis, E. A.: Computer network architectures /In Russian/ /Statistika, Moskva, 1980/.

[3] Cypser, R. J.: Communications architecture for distributed systems /Addison - Wesley, Reading, 1978/.

[4] Synoptic table of X.25, X.75 - D-channel protocol and signalling system No.7, mapped on layered Reference Model. CCITT Study Group XI, COM XI - 10-E /CCITT, December 1984/.

[5] Ideguchi, T. - Mizuno, T. and Matsunaga, H.: Automatic implementation of communication protocols, ACM Computer Comm. Review /SIGDC/, 1/1981/ 40-56.

[6] Pužman, J. and Pořízek, R.: Communication Control in Computer Networks /J. Wiley and Sons, Chichester, 1980/.

[7] Pužman, J.: The Relational Approach to the Computer-Communication Architectures, to be published in Problems of Control and Information Theory.

NETWORK SERVICES VS. OSI/RM

Joseph F. Pužman

Prague

Czechoslovakia

The paper deals with services provided within the Reference Model of Open Systems Interconnection (OSI/RM). The term "service" appearing in various contexts is discussed in detail and discrepancies in the understanding of this term are pointed out. New kinds of telematic services are mentioned and compared to the services provided by layers of OSI/RM. The evaluation of services in terms of performance criteria and quality of service parameters is also discussed with the aim to accent the user's and provider's standpoints.

INTRODUCTION

When the first proposals concerning the layering of complex computer-communication architectures appeared, hardly anybody believed that it would be possible to find out a reference model sufficiently universal and valid for all purposes and for an appropriately long time despite the rapid development of the computer-communication branch. Let us remind the discrepancies that occurred in the past between works based on ISO and CCITT. As soon as the basic document "Open Systems Interconnection - Basic Reference Model" was issued and approved as an international standard [1] the era to come was envisaged as a new revolution of the seven layered architectures (for the mystique of Seven see [2]). Although the Reference Model of Open Systems Interconnection (henceforth abbreviated as OSI/RM) was initially criticized, it is evident now that this model is suitable for all data transmission and computer-communication systems and networks, public, private as well as firm ones. This model may seem redundant for special purposes; however, it should be born in mind that it has to be generally universal. Necessary modifications are justified if they are of assistance in simplifying the design and operation of network structures.

Much work has been done within the framework of CCITT during the study period 1981-1984 which culminated last year by the VIIIth Plenary Assembly at Malaga-Torremolinos. A great number of new recommendations of the X-Series were approved there [3] , which made for a good start for further work. In addition to the documents concerning public data networks, many other recommendations from the telematic services area appeared, some of which were in harmony with OSI/RM while some were not. Nevertheless, these documents for the present have revealed no principle obstacles in the application of OSI/RM even to such systems.

In this paper it will be demonstrated how the layers of OSI/RM can be understood from the system, hardware, software, and users viewpoints; the widely used term "service", which acquires various meanings depend-

ing upon the context,will be discussed,and the connection of services with their performance (quality) will be outlined, again from the user´s and provider´s standpoints.

It should be noted that the contribution expresses the author´s personal views and it does not represent the official attitude of any institution.

SYSTEM INTERPRETATION OF THE OSI/RM LAYERING

The OSI/RM standards [1,3] offer a philosophical background and open the door to modelling complex computer-communication systems/networks. Their main aim is to coordinate the development of future standards (some of them having been already approved, others being in preparation) for the interconnection of the systems/networks under consideration.

Since OSI/RM is a layered model and many abstract terms are to be employed for its development (and understanding), the standards and recommendations are largely concerned with the requisite terminology. Even if the documents relating to OSI/RM contained nothing more than that (they actually contain much more) they would have the merit of having brought the Babel confusion which for a long time governed the separated computer and communication clans. Now the "Chinese walls" seem to have been torn down, and the first fruits of the cooperation are getting ripe in the form of a great number of standards, recommendations, and drafts.

The principle of layering, sometimes considered the most important feature of any architecture, is of a great practical value in view of the complexity of the existing data processing systems.The hierarchical arrangement of layers is logical, although many other architectures exist which can avoid it (cf. [4]). Only the layered architectures,in particular OSI/RM, however, will be our concern in the paper.

First, the question arises as to how to interpret the functions and contents of concrete layers from several viewpoints. As for communication functions within layers as well as communication services conveyed between layers the reader is referred to the relating documents [1,3] which require no comments. However, the interpretation of several views is left to the designer and user.

Let us show now an example of interpretation.Table 1 depicts the system view on the OSI/RM layering, the implementation of hardware/software into layers, and the user´s viewpoint divided further into data units to be transferred and/or processed within the particular layers, mental processes, and corresponding languages. Table 1 is a modification of the table taken from [5, p.5] and should be looked upon as a subjective opinion. The boundaries which are sufficiently strictly specified in OSI/RM become ambigous (even fuzzy) in other interpretations and do not serve as a generally adopted guide. The table also does not substitude the missing links in the approved documents.

Omitting the implementation which is the designer´s task and may vary in a wide spectrum, as well as the user´s viepoint which is beyond the scope of this treatment, we shall focus our attention on the system view. The lowest and simplest system involves physical connections (PCs) and data circuits in circuit-switched public networks - CSPN (telegraph, telephone, data). These require only the establishment/release for creating and disconnecting a connection between two physical entities.

Table 1 Several views on the layering within OSI/RM

System view	OSI/RM layering	Implementation views		User views		
		Hardware	Software	Data units	Mental process	Language
data processing system (distributed)	application	HOST computers	user applicat. process control RJE,VTAM,TCAM	data bases	semantics, inference mechanisms	artificial language
integrated services digital network - ISDN	presentation	personal computers terminals,VDU	library routines,data base management, VTP, FTP	transformed messages, files	syntax (compression, encryption)	high level programming, machine oriented, control, command, job control languages
computer-communication subsystem	session	front-end processors	operating system, network operating system	process messages, transactions	dialogue (interprocess communication)	
PC,CSPN / data link / packet switched public data network,LAN / private data network, value-added network	transport	concentrators	transport station	DTE´s messages, letters	cost of transfer, resource allocation	black box
	network	device driver, node computer, network processor	executor / firmware / communication module	packets, datagrams	black box	
	link	interface DCE/DTE / line adapter UART, USRT		blocks, frames		
	physical			signals, bits, characters		

The data link layer corresponds to the data links (hence its name) and their establishment, maintenance, and release. Note the difference between a physical connection (line) and a data link which is a data circuit equipped with control units of data stations controlling the data transfer among network entities. Data links are either private or constructed on the basis of leased lines or switched lines established in circuit-switched public data networks. The third layer is needed for relaying data flows through intermediate switching nodes, i.e. for packet/message switching. Three lower layers usually form the packet-switched data networks (including local area networks - LANs), and the interface between the network and transport layers can be considered as a demarcation line between the network provider (say PTT) and the network user (subscriber).

The transport layer functions serve to provide performance values requested by users with respect to services provided by a network (packet-switched or line-switched). For the services and their performance see next sections. All transport protocols operate between end systems and support transparent data transfer between session entities. Transport services provided to higher layers are similar to these provided by the most private data networks or some value-added networks (VANs). The computer communication subsystem, despite its vague definition may involve all layers up to the session layer because its main purpose is to organize the cooperation of presentation entities.

Since the integration of services in the telecommunication sense (ISDN - integrated services digital network) involves i.a. the data (audio, image, text) structuring ISDN must encroach upon the presentation layer. Finally, according to the definition of the open system (system of hardware, software, etc. performing information processing and/or information transfer [1,3]) data processing system or distributing data processing system should possess the seven layers architecture including the application layer.

While the system hierarchy up to the transport layer is unambigous and has been verified by the practice (thanks, i.a. to the adopted precise specifications in almost all firm architectures), the higher system hierarchy suffers from some speculations. More correct specifications and definitions of higher-layer data systems/networks on the one hand and of layer functions/services on the other are needed for mutual understanding.

Table 1 does not involve the well known DCE (data circuit-terminating equipment) and DTE (data terminal equipment) except for the DCE/DTE interface in the hardware column. The two devices are defined in the CCITT recommendations from the viewpoint of the data station structure. In fact, the term DCE refers to the border device of a data network and DTE is usually a user's equipment joined to the network through DCEs. Apart from their hardware/software content, the question arises to how DCE and DTE, regarded as systems, should be interpreted within OSI/RM. This is simple to answer if the CCITT X-Series recommendations are consulted. DTE may comprise all layers or at least the four lower layers up to the transport one (in order to guarantee the end-to-end communication between systems) while DCE covers at most the three lower layers and constitutes the relay-network internal node which stays near the user's DTE and negotiates the DTE's access into the network. Such an interpretation roughly satisfies the X.25 protocols which serve to control data exchange between the user owing his DTE and the network seen by him through one or several DCEs (Figure 1).

Let us return again to the interface between the network and transport

layer. It has been mentioned that the transport layer plays the most significant role for the separation of the data transmission from the data processing. The lower layers form a domain for a telecommunication access to the data processing facilities. Despite the very strict organization of the data communication services (functions) within the three lowest layers, their service capabilities vary according to the communication media used. Different functions are performed as soon as the access is provided via leased lines, switched lines, local area networks. For example, leased lines need all link layer services while circuit-switched lines must be, in addition to it, supported by the physical layer services. The packet-switched access as well as LAN access should be fully equipped up to the network layer, however, the architecture of LANs does not yet completely coincide with that of OSI/RM.

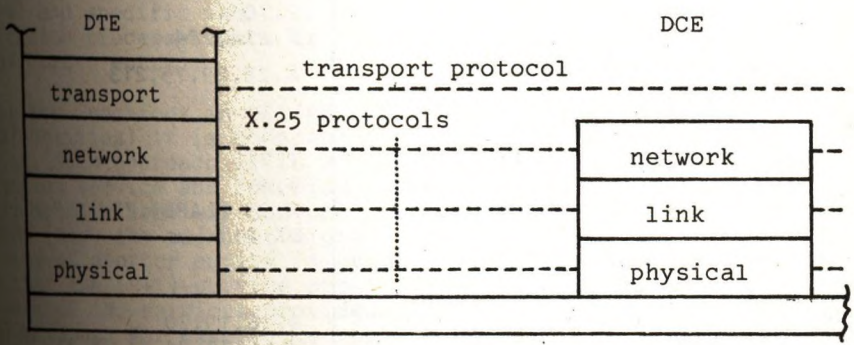


Figure 1
The position of DCE and DTE in OSI/RM

Moreover, if public networks (public switched telephone network -PSTN, circuit-switched public data network - CSPDN, packet-switched public data network - PSPDN) are employed for the access, it is under the control tied internationally by the series of CCITT recommendations (e.g. X.10 [6] which generally states the classes of access via all communication media except for LANs). Another example can be taken from the recommendation T.70 [7] where the transport service for the telematic purposes is described with a discussion of protocols which may or may not support the three lowest layers. Also the main requirements of the transport service are specified (network independence, end-to-end significance, transparency, error free delivery, and cost efficiency) which again proves the importance of the transport layer.

When we have mentioned the standards concerning three lower layers of OSI/RM it should be of use to check how all architectural layers are covered by ISO standards and CCITT recommendations. Thus, in conclusion of this section, Table 2 is attached surveying the standardization activity at the end of 1984. Besides standards and recommendations being already approved also their drafts (D) and draft proposals (DP) are included in order to show the current state of the development. The table cannot give a complete view on gaps, most of which are under studies, because of the absence of relationships between the documents and the relevant logical structure of OSI/RM. Also not all standards (particularly these which were issued before 1980) fit easily to the present concept of OSI/RM (e.g. IS 1155 and the family of character oriented link protocols, V.41, etc.).

Table 2 The covering of OSI/RM by the standardization activity

Layer	ISO standards	CCITT recommendations
application	DP 8571,8649,8650	T.5,60 X.400,401,408,409,410,411, 420 DX.217
presentation	DP 8822,8823	T.6,61,72,73,100 DT.101 DX.216
session	DIS 8326,8327	T.62 X.215,225
transport	IS 8073 DIS 8072	T.70 X.214,224
network	DIS 8348 DP 8208,8427,8473, 8648	X.25,29,75,213
link	IS 1155,1177,1745, 2111,2628,2629, 3309,4335,6159, 6256,7809 DIS 7478,7776 DP 7777,8471	T.71 V.41 X.25 (LAPB),75 (SLP,MLP) DX.212
physical	IS 2110,2593,4902, 4903,7480 DIS 7477,8481 DP 8480,8482	S.16,19,20 V.10,11,24,25,28,31 X.20,21,22,24,26,27,28,30, 31,32 DX.211

SERVICES AND "SERVICES"

The term "service" is dealt with in nearly all documents concerning OSI/RM. Still, this term is very often confused because of the professional approach in different branches. For the computer people, the service is processing capability of the computer system,sometimes lacking any detailed specifications. Communication people, in particular the PTT bodies, put a sign of equality between the general term "service" and services provided by the private or public communications carriers.

In the OSI/RM documents [1,3] the service is defined as the "abstract capability provided by a layer to the next higher layer" which differ from both of the meanings mentioned above (this fact appears in the notes accompanying the layer specification from the network layer beneath). Such a definition tacitly involves services provided by layers in the layered architecture (hence the name (N)-service). Nevertheless explanation in terms of the vague "abstract capability" is insufficient unless a more detailed specification is further worked out.Although much has been done in a deeper specification of (N)-service with relation to all the seven layers of OSI/RM, we shall add several remarks which should be taken into consideration in the stage of a "further study".

Generally, the (N)-service is composed of (N)-facilities and/or of elements of services. No unambigous boundary exists between the facil-

ities (e.g. connection establishment/release, normal/expedited data exchange, exception reporting/recovery) and the elements of services (e.g. routing, flow control, error control); distinguishing services from their elements (facilities) thus seems to be redundant.The CCITT approach is more justifiable (X.2 in [6]) because it utilizes the same pair of terms (service-facility) but in the hierarchical sense. While the user service denotes the common feature (leased circuit, circuit-switched, packet-switched, virtual call, permanent virtual call) the user facility means the subservice provided within a service (multi-point, direct call, closed user group, packet retransmission, fast select, etc.).

On the other hand the CCITT document [8] has introduced other elements of service applied in the application layer. These are categorized into common ones (providing capabilities generally useful in all applications) and specific ones (required to satisfy the needs of specific application processes such as file transfer, job transfer, data base access, order entry).

As for the functions in OSI/RM ((N)-functions as part of the activity of (N)-entities) it is comprehended always to be the other side of a service. Such approach is in accordance with the common notion of communication service and function (studied in [9] or elsewhere). An entity acts in some way and provides a function which is seen from outside (say, the adjacent higher layer) as a service supporting the activity of another entity (staying in that adjacent higher layer). Note that the (N+1)-service provider in the (N+1)-layer is in fact the user of (N)-services provided within the (N)-layer by means of (N)-functions. For example, error notification is a service provided by, say, the network layer and this service supports the function of the end-to-end error recovery in the transport layer.

Thus, the protocol of a layer can be designed and implemented based on the knowledge of services provided by a lower layer which support completely the functions within the layer in question.

In the CCITT X-Series recommendations, services are further divided into service-primitives (abstract interactions between service-users and service-providers) such as request, indication, confirmation, response, which are classified into mandatory and optional (user/provider). While service elements can be still regarded as services,primitives concern the command exchange and relate to protocols.

Since services in the sense discussed above serve as an abstract tool of OSI/RM, we shall refer to them as architectural (this name is not new, it is widely used in the CCITT documents [3] and does justice to the general relationship between entities). It remains, however, to define those attributes of architectural services that are connected with their quality. We shall touch the quality of service (QOS) in the next section. Here it is necessary to mention that QOS sometimes appears among services as a special item (in three lower layers),which again confuses the meaning of service.

Now, let us draw attention to the other side of the coin. In a common sense, services are considered in relation to the PTT provider as transfer of messages of different nature and type with a large variety of formats, viz.

- audio (TV, picturephone, facsimile),
- graphics (telex).
- data (data transmission).

For our purposes only digitized services will be dealt with. These

services are offered under many different names and most of them form a family of the co-called telematic services. We speak about teletex and teleconferencing (alphanumeric exchange), videotex and videoconferencing (graphic exchange), telewriting (handwritten information exchange), etc. Of course, these services are very strictly specified since all of them are under international control.

We have met with data transmission services in the early public data networks. Let us mention the classes of services defined in the oldest recommendations of the X-Series (X.1 and X.2 [6]). These services pertain to the type of switching (circuit vs. packet), calls (virtual vs. permanent virtual), connection (connection-oriented vs.connectionless), transmission rate, synchronization (start-stop vs.synchronous), and are referred to as user services. They represent a category of data transmission and are, from the provider point of view, network dependent.

The user, on the other hand, is not acquainted with the nature of the facilities offered by the various data transmission services and he speaks about communication or message exchange services. In order to avoid confusion with communication services within OSI/RM (as the opposite to communication functions), we shall keep to the latter term- - message exchange services.

Returning to OSI/RM, message exchange services may be identical with those provided by a certain layer of network architecture which is seen by the user at his access point. We have said that the network layer used to be an appropriate interface to public data networks; still, this is not the only solution. With the help of the newest CCITT recommendations and their drafts, Table 3 suggests a way of distributing the main message exchange services among layers of OSI/RM (cf. Table 2).

Table 3 Message exchange services within OSI/RM

Layer	Message exchange service
application	message handling system (MHS), document reference information
presentation	document interchange, videotex (presentation on the TV CRT), mixed mode teletex and facsimile, teleconferencing, telewriting, coding/conversion, encryption
session	telematic services identification (e.g. format in teletex, colouring in telefax)
transport	telefax, bureaufax
network	telefax, telex, packet switched data transmission services (virtual call)
link	leased circuit data transmission services (point--to-point, multipoint), permanent virtual circuit service
physical	circuit switched data transmission services

The lowest layer assures data transmission over public switched telephone (telegraph) networks and users handle the interface V.24/V.25 (S.16/S.19,20). In CSPDNs the physical connections are established, maintained, and cleared by means of the X.21 (X.20) protocol.The leased circuits provide message exchange services up to the link layer (data communication is supported by data stations and link protocols and the same may intrinsically hold for the permanent virtual circuit service in PSPDNs. The network layer is needed for packet switching services in general as well as for telex and telefax, if they are supported by message switching (due to the necessity of storing,routing, and forwarding). Teletex and Group 4 facsimile (facsimile over data networks) services require additionally the transport layer which guarantees the block formats, and sometimes the session layer for the service identification. The session layer and associated protocol serve for the exchange of information identifying practically all telematic services. Above the session layer the printing or display of stored information is necessary to make it understandable to human beings (interactive videotex, mixed mode teletex and facsimile, teleconferencing, telewriting, etc.). This is the role of the presentation layer which also seems to be optimal for data encryption for security purposes. The highest layer - application - supports the message exchange services for interpersonal communication (message handling system - MHS), and storing and retrieval of texts represented in electronic form (document reference information) universally applicable to all the existing text communication services.

Table 2 is not closed up and the topic calls for more study.The reader will find many stimulations in [10] and relating CCITT documents.

QUALITY OF SERVICES

As mentioned above,each service (architectural as well as message exchange or communication/processing) should be evaluated in terms of some numerical value (values). Regardless of the nature of a service, the user requires services chosen at a prescribed level of performance at appropriate cost. We underline "appropriate cost" since this is often replaced by "minimum cost". Figure 2 will serve as an explanation.

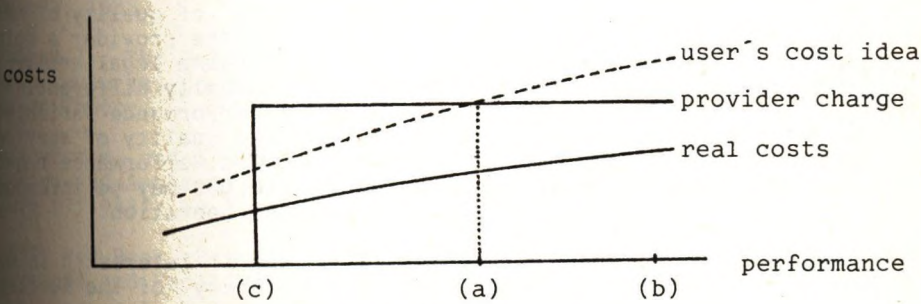


Figure 2
Examples of cost/performance curves

The user appoints a value of performance which he (she) considers acceptable (point (a)),and expects that the provider will take pains to keep the performance above this value. On the other hand, the service provider guarantees his performance value (say, point (b)). If this coincides with or is better than the user-acceptable value, the two

partners will be satisfied. If not, the user may either refuse the service or implement his (her) "own private layer" above the provided network which accommodates the network performance to the necessary requirements.

Actually,however, cost discrepancies may occur. In reality,the higher performance the higher costs and vice versa; the provider, however, usually charges a constant tax for a service according to a tariff policy stated in advance. He, of course, repudiates the charge if the performance decreases under a prescribed level (point (c)) but the level may no more remain suitable for the user.

The user, however, can have a different idea of the cost dependence (in fact, the user is not familiar with the network structure). An example of the user's idea is drawn in Figure 2 in a dashed line. If the acceptable value is identical with the point of intersection of the ideal and charge curves, the corresponding value of costs could be considered as the "user minimum".

Moreover, the provider cannot guarantee that the originally negotiated performance value will be maintained throughout the whole lifetime of the service provided.This is associated with the fact that the performance is largely expressed in random variables the values of which fluctuate over a wide range and whose changes are hardly explicitly signalled. It must be admitted that the user may claim a lower cost against the network provider but this is ex post facto and loses its sense as soon as the loss in unsatisfied communication cannot be undone.

Until now we have looked upon performance as an abstract attribute.In fact, performance has to represent several (if not all) of the user's requirements of performance criteria.From this it follows that performance is a vector whose components represent performance variables (we use the term "variable" since this term has a mathematically more specified meaning than the comonly used term parameter).For example, accuracy is a performance criterion and may be expressed through RER (residual error rate), BER (block error rate), block loss rate,extra block rate,block misdelivery rate,etc. in terms of accuracy variables (for a complete analysis of accuracy see [11]).

The performance should be the uniform measure of quality of message exchange services both from the user's and the provider's points of view. However, the user usually is not the same legal entity as the provider so that their evaluations may be mutually different.For this reason the network used to be evaluated by performance variables representing the user's requirements,and by the quality of services (QOS) relating to the services provided by a network.Performance requirements are focused on particular applications while QOS may be influenced by the network internal structure, control, and operation.

Let us return to the above example. The user is interested in message transfer and his only requirement is accuracy of the successfully transferred messages. The network handles packets (blocks, frames) rather than messages and then error rates referring to transmitted information units (say, packets) can be measured and used for the evaluation. A one-to-one correspondence must exist between the components of performance and those of QOS, otherwise the two partners could not come to agreement.

It is desirable to have a set of general QOS variables at one's disposal. A complete list of QOS variables relating to public data net-

works is included in the CCITT X.140 recommendation [12] which is based upon the Federal Standard 1033 [13]. Unfortunately, no QOS variables which would reflect the performance of layer services/functions of OSI/RM have been yet proposed within the ISO standards. Although performance variables have actually been mentioned in some layer specifications, the necessary definitions are still lacking. On the other hand, the general QOS variables from [12] are not always suitable as attributes of (N)-services, and the more or less complex conversion tables are a tenuous way out from this impasse.

CONCLUSION

In the paper some open questions concerning a wider application of OSI/RM have been set forth. These problems surely do not detract from the value of OSI/RM, and they should not discourage the reader from his or her OSI/RM study and development. The author believes that the OSI/RM approach will survive the rapid development of the new message exchange services and remain as a useful tool for the design, operation, and maintenance of sophisticated computer-communication systems for information management.

ACKNOWLEDGMENT

The author expresses his gratitude to M. Bazewicz from the Technical University of Wroclaw (Poland), and to F. Stuchlik from the Technical University of Magdeburg (GDR), for their suggestions and advice, in particular in preparing Section 1 and Table 1 of the contribution.

REFERENCES

[1] Information Processing Systems - Open Systems Interconnection - Basic Reference Model, International Standard IS 7498 (ISO,1983).

[2] Cohen,J. and Postel,J., The ISO Reference Model and Other Protocol Architectures, in: Mason,R.E.A. (ed.), Information Processing (North-Holland, Amsterodam, 1983).

[3] CCITT Red Book, Fascicule VIII.5, Data Communication Networks; Open Systems Interconnection, System Description Techniques (UIT, Geneva, 1985).

[4] Pužman,J., The Relational Approach to the Computer-Communication Architectures, to be published in: Problems of Control and Information Theory.

[5] Bazewicz,M. and Stuchlik,F., Problems of Computer System/Network Design and Application Methods, Wissn.Z.Techn.Hochschule Magdeburg 27(1983), No.8, 1-5.

[6] CCITT Red Book, Fascicule VIII.2, Data Communication Networks; Services and Facilities (UIT, Geneva, 1985).

[7] Network Independent Basic Transport Service for the Telematic Services, CCITT Recommendation T.70, in: Red Book, Fascicule VII.3 Terminal Equipment and Protocols for Telematic Services (UIT, Geneva, 1985).

[8] Final Report on the Work of SG VII During the Study Period 1981--1984, Pt. IV: Proposed Study Programme of SG VII for the Next Study Period, CCITT Report R 40 (CCITT, Geneva, May 1984).

[9] Pužman,J. and Pořízek,R., Communication Control in Computer Networks (J.Wiley, Chichester, 1980).

[10] Questions Entrusted to SG VIII "Terminal Equipment for Telematic Services" for the Period 1985-1988, CCITT SG VIII Contribution VIIIth Plenary Assembly (Malaga-Torremolinos, Oct. 1984).

[11] Pužman.J., On User Accuracy Parameters in Computer Networks, in: Proc. IFIP Intern. Symp. COMNET'81 Networks form the Users' Point of View (North-Holland, Amsterodam, 1981), 5/38-5/51.

[12] General Quality of Service Parameters for Communication via Public Data Networks, CCITT Recommendation X.140, in: Red Book, Fascicule VIII.4, Data Communication Networks; Transmission, Signalling and Switching, Network Aspects, Maintenance and Administrative Arrangements (UIT, Geneva, 1985).

[13] Telecommunications: Digital Communication Performance Parameters Int. Fed. Standard 001033 (NBS, Washington, 1979).

-0-0-0-

AN IMPLEMENTATION OF A FILE PROTOCOL FOR COMPUTER NETWORKS

Yuri Atanassov
Tchavdar Stoinov
Tzvetana Gotcheva
Jordan Ganov

Program Product and System Corporation
Central Institute for Program Product and Systems
Sofia, Bulgaria

The implementation of a file protocol for mini SM-4 operating in a computer network in conformity to ISO Standards is desribed. The basic principles of ISO Standarts /Draft Proposals/ defining the File Transfer, Access and Management Protocols in a computer network are viewed. The file protocol design is discussed. Its block structure and the basic functions performed by its modules are presented. Mapping the file protocol to the local file system is described and attention is called to some particuliarities in the implementation. Future trends of research efforts in this area are outlined.

1. INTRODUCTION

The large-scale management and general purpose information systems are characterised by numerous user tasks which can be resolved by processing large amount of data which have been collected and stored by different computers. In the basic reference model of open systems interconnections, standardized by ISO, the File Transfer, Access and Management /FTAM/ Protocol is one from a set of protocols at the application level. The FTAM protocol allows access to open systems resources and manipulation of files or data sets which may be treated as files, independent of their physical location on the computer network.

This paper describes the implementation of a file protocol for computer networks. Section 2 briefly presents the background. Section 3 discusses the basic characteristics of protocol design and system structure. Some experimental results are produced in Section 4. Finally, Section 5 concludes the paper with a brief discussion of outstanding problems.

2. FILE PROTOCOL PRINCIPLES

The principles of the File Transfer, Access and Management /FTAM/ Protocol, described below, are based on papers /1/, /2/, /3/, /4/.

The FTAM Protocol provides users with file services, independent of the physical station of files /whether at the local station to which users are connected or at a remote station on the computer network/.

Two types of actions are made available by the FTAM Protocol:

- actions of complete files;
- actions for file access.

Actions on whole files include the services:

- create file;
- select file;
- change attributes;
- read attributes;
- open file;
- close file;
- delete file;
- deselect file;

Actions for file access include service:?

- locate;
- read;
- insert;
- replace;
- extend;
- erase.

The FTAM protocol is devided to the following subsets:

- subsets of reliable file service;
  - file transfer service subset;
  - file access service subset;
  - limited file management service subset;
  - enhanced file management service subset.
- Subsets of user correctable file service;

  The user correctable file service is defined to include all the subsets of reliable file service. In addition, it also defines an error recovery service subset, in which additional primitives are available for error recovery purposes.

The graphical representation of the subsets and conditional indications of the above services are given in Fig. 1.

The FTAM protocol makes it possible for the users to define a group request including several requested services. The group request gives the users several advantages: to decrease the traffic being processed by the presenation, session, transport, network and physical levels; to decrease the traffic being transferred across the communication network; to simplify processing handled by both systems / the local and remote stations/, which perform the functions of a FTAM protocol.

The FTAM protocol gives users the possibilities of error recovery as information is being transferred. To meet this requirement, the FTAM protocol may be designed in two forms. With the first, error recovery is performed by the users themselves. With the second, these functions are done by the program module, which implements an error recovery protocol. The FTAM protocol may be designed in such a way as to allow users to take their choice between these two forms.

F-CONNECT, F-RELEASE, F-ABORT
F-SELECT, F-DESELECT
F-OPEN, F-CLOSE
F-READ, F-WRITE
F-DATA, F-DATA-END, F-TRANSFER-END
F-CANCEL
F-BEGIN-GROUP, F-END-GROUP
FILE TRANSFER SERVICE SUBSET
F-LOCATE, F-ERASE
FILE ACCESS SERVICE SUBSET
F-CREATE, F-DELETE, F-READ-ATTRIB
F-CHANGE-ATTRIB
LIMITED FILE MANAGEMENT SERVICE SUBSET
F-RECOVER, F-CHECK, F-RESTART, F-CANCEL
ENHANCED FILE MANAGEMENT SERVICE SUBSET
ERROR RECOVERY SERVICE SUBSET

Fig. 1

3. FILE TRANSFER PROTOCOL DESIGN

System analysis of ISO Draft Proposals /1/, /2/, /3/ and /4/ shows that the systems realisation program will be asymmetrical in structure. The block diagram of the separate structure parts and their interaction is shown in Fig. 2.

The TERMINAL INTERFACE /TI/ establishes the connection between the terminal user and a system structure part FTAM protocol user /FPU/. It offers the user a syntax that facilitates him to formulate the request. If necessary, TI displays an additional information about services, parameters and their suitable values. The dialogue between the user and TI is realized by displaying prompting information on the screen that defines the desired services and the admissible response range /in case of numeric values/, or admissible answers /for numeric values/. The formal and/or logical control is performed on the user answers, keyed in the terminal.

Information about previous requests which is refferred to the same file is saved by the TI. This feature of the TI relieves the user from unnecessary multiple input of the same service information relevant to the same file which has been previously requested. In case of negative completion of the request the user can correct only those parameters preventing the requested service execution and put the next request.

In the TI-part of our system is realized the idea for a complex request that allows the user to define the type of requested services for the FTAM Protocol and makes it unnecessary to keep in mind the previous FTAM protocol entity state completion. Each complex request is composed of a set of primitives. What kinds of parameters concerning a new request should be input by the user are prompted by the TI, depending on the old FPU entity state and the new desired services. This information is prompted by the user display screen during the TI-user dialogue. The idea for the complex request is different from that of the group request. The basic purpose of the complex request is to facilitate the user.

The FPU receives the complex request from the TI, which is coded as a sequence of primitives $described in /3/ and begins to execute the primitives step by step. Before the execution of each primitive some checks are made if the requested action is to be performed on a file. If this file resided in the local station the FPU calls the corresponding subroutines for processing of the requested action on the file. If a request is made for the remote station, a corresponding Protocol Data Unit /PDU/ is produced and transmitted to the remote station.

Whenever the complex request execution is successful or a negative result is received on the intermediate stage of the primitive execution /impossible requested action/, a response is produced to inform the user of the normal request completion or to determine the abnormal cause for termination. The response is delivered to the TI, afterwards it is supplied with the appropriate explanatory text and displayed on the user terminal.

The FTAM PROTOCOL SERVER /FPS/ receives a PDU or a set of PDU's /in case of group request/ from FPU and performs requested actions and produces answers containing the obtained result. If the requested action refers to the file the corresponding subroutines are called by the FPU to perform the desired services.

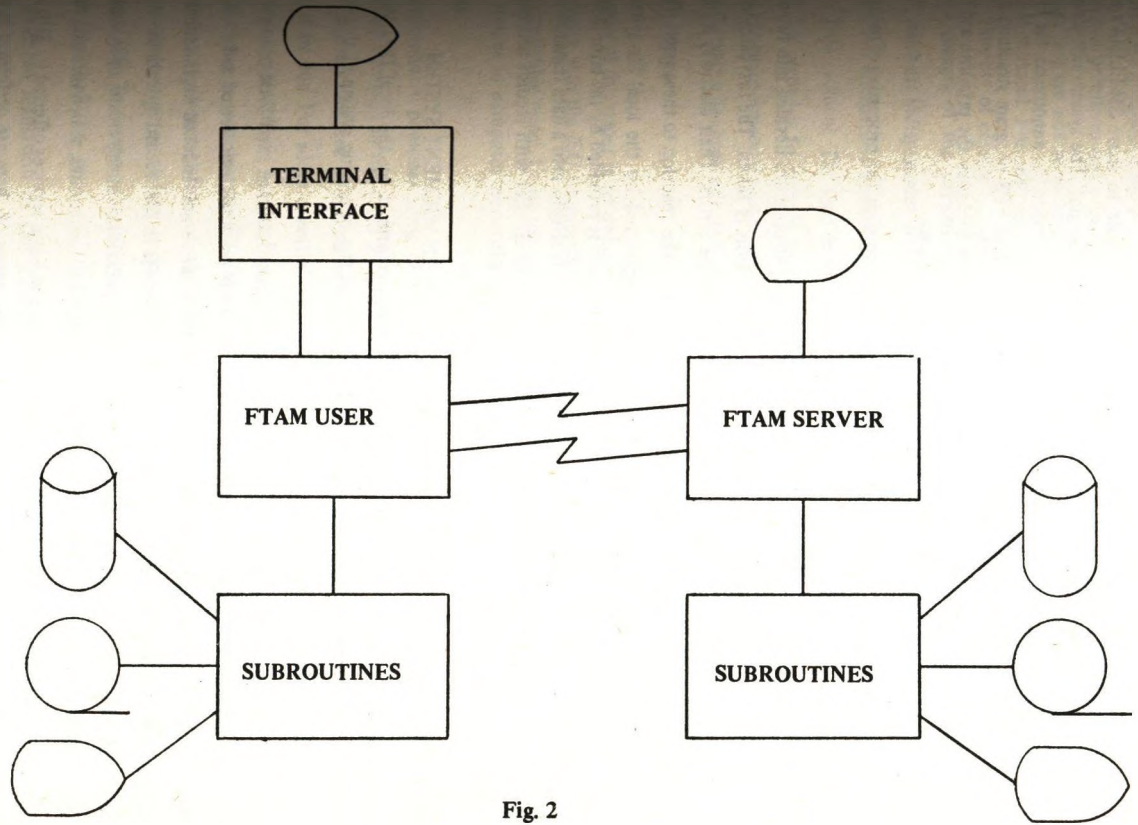

TERMINAL
INTERFACE
FTAM USER
FTAM SERVER
SUBROUTINES
SUBROUTINES

Fig. 2

Primitives and PDUs exchanging between FPU and FPS in task-to-task communications manner is done by a network task. A system called SM-ELIMA /5/ for SM-4 computers /or compatible ones/ has been developped by our team. This system has been developped on the basis of ISO Draft Proposals for Session, Transport and Network levels. The task-to-task communications require that FPU and FPS tasks to be known to SM-ELIMA-task and determined as ones being able to share SM-ELIMA's resources. Declaring FPU is made by starting the FPU-task from the user terminal. The system administrator runs the FPS-task on the system terminal. When a FPS-state "out of service" is announced, the FPS-task is aborted and the resources occupied are freed. Each of the FPU and FPS program structures consists of two parts - control and processing sections. The control section includes the primitive and the PDUs separators and common subroutines for abnormal situation processing.

The processing section consists of a set of subroutines that accomplish the processing of each input primitive or PDU.

The errors that are to be debugged for the reliable file service purpose are devided into two categories, depending on where error checking is made and appropriate action is taken. The first category includes the protocol errors /protocol procedure deviation/ and parameter errors. Their discovery is provided by an entity that performs the corresponding service. F-ABORT is the reaction to these errors. The second category includes logical errors, lack of resources, errors discovered by the local file system and so on. The error recovery reaction depends on the error character and is provided by the error recovery entity. This entity realizes the synchronization point distribution algorithm into a data transmission stream, determines the error recovery point, performs the control of the docket with records necessary for the error recovery activity.

4. MAPPING THE FILE PROTOCOL TO THE LOCAL FILE MANAGEMENT SYSTEM

The file access and management in every user's system is performed by the local Data Management System /DMS/. The mini SM-4's DMS, called SUD, is part of the operating system of the minicomputer.

The comparative analyses of services necessary for the implementation of the FTAM protocol and functions offerred by SUD demonstrate good conformity between possibilities offerred and necessary functions. SUD provides: management of files with sequential, relative and indexed organisations; three record access modes: sequential, random, access by record's address in file; record operations - connect, delete, disconnect, find, flush, get, put, rewind, update; file operations - close, create, open, erase, extend. The FTAM protocol envisages to use only the first two record access modes, which increases the time for the execution of subroutines /Fig. 1/. As mentioned above the basic services offerred by SUD are in conformity with those envisaged by the FTAM protocol, but some of them are performed in a different sequence. For instance, after create operation the FTAM entity is in the state "selected", but the SUD file is in the "data transfer idle"state; for the completion of a delete operation, SUD requires to be in "data transfer idle" state, while the FTAM entity must be in "selected" state. For transferring data

from/to SUD file, after open file service was made, connect to SUD file must be accomplished first, while the FTAM protocol does not provide for the service connect to SUD file.

There are some differences between SUD and FTAM protocol related to file and record attributes and parameters of primitives and protocol units. The differences concern the secondary parameters mainly. Parameters that are not supported by the local file system are not included in the implementation of the FTAM protocol. For instance, Read Attributes service's parameters: "date and time of last read access", "identity of creator", "current filesize", "future filesize" are not supported by SUD. SUD does not allow changes of the parameters "filename", "access control", "file availability", "legal qualifications", which are indispensable to Change Attributes services.

There are two alternatives concerning services and parameters supported by SUD, but not envisaged for the FTAM protocol. The first is the acceptance of standardized values for some parameters required by SUD. The second is to include these parameters in an implemented FTAM protocol, thereby making a certain allowance for some non-conformity to ISO standarts. Since secondary parameters are concerned the first alternative has been selected in most cases.

5. CONCLUSION

The purpose of this paper is to discuss a realization of a file transfer, access and management protocol in accordance with the ISO Standarts. The protocol is intended to be implemented on SM-4 minicomputers /or compatible ones/. Using the described system on these minicomputers makes it possible to provide file services in computer nerworks or to use the services which are presently available in computer networks.

The implementation possesses some characteristic features. A terminal interface has been treated which has not been provided in the draft proposals of ISO but is convenient for the users working with FTAM protocol and sets him free of the necessity to remember the rules of the actions as well as the parameter values that are due. For some essential services some differences were established about the consistency of the actions provided in the draft proposals of ISO and those in the local file system. To overcome those differences in mapping the FTAM protocol to the file system in the responsible transitory points, an association or dissassociation of some primitives has been applied. On the other hand the local file system proposes some secondary services that could be realised only by some deviations of the draft proposals of ISO. To preserve the full correspondence between the realized FTAM protocol and the draft proposals we had to give up the usage of those services in the network environment.

The so described realisation of the FTAM protocol requires: 64 KB of memory / an overlay structure is used as well/; SM-4 mini computers under control of DOS/RV-B-3.2 operating system. The programs are written on a MACRO assembler.

Future efforts will be focussed on several points. First, it will be necessary to create the corresponding structural parts of the FTAM protocol, on the analogy of those discussed above, which are to be used

on ES-computers /or compatible ones/. This will make possible to provide file services in heterogeneous networks, incorporating SM-4 and ES computers.

Users connected to the corresponding local computer via terminals will have the possibility for interactive work with files in a computer network.

The second step will be to develop FTAM protocol program interfaces, allowing the user application programs to use the file services in the computer network, i.e. to use files sited in any connected to the network computer.

The next step will be to program some simple functions for remote job entry with the aid of FTAM. For this purpose the command file will be transferred to and executed on the remote computer.

Finally, a user correctable file service with an error recovery service subset will be incorporated in the FTAM protocol. This will allow the users, if they wish, to define by themselves the location of the control points and the actions required for maintaining the transferred information integrity /without the employment of the error control protocol/.

After the completion and introduction of the development project referred to in this paper, the new FTAM protocol will provide the possibility for using all services set out in the ISO Standards concerning heterogeneous networks, consisting of both ES-computers and SM-4-computers.

REFERENCES

/1/ ISO/DP 8571/1 Information Processing Systems - Open Systems Interconnection - File Transfer Access and Management - Part I: General Description.

/2/ ISO/DP 8571/2 Information Processing Systems - Open Systems Interconnection - File Transfer, Access and Management - Part II: The Virtual Filestore.

/3/ ISO/DP 8571/3 Information Processing Systems - Open Systems Interconnection - File Transfer Access and Management - Part III: The File Service Definition.

/4/ ISO/DP 8571/4 Information Processing Systems - Open Systems Interconnection - File Transfer, Access and Management - Part IV: The File Protocol Specification

/5/ Dimitrov St., B. Dimitrova, J. Mindov, Tz. Gotcheva, An Implementation of ISO Protocols in SM-4 Host Computers, IFIP Conference 1984, Bulgaria Sofia.

MANAGEMENT SERVICE OF PACKET SWITCHING PROCESSORS

Rainer Penzel

Institute of Informatics and Computer Techniques
Academy of Sciences of the G.D.R.
Berlin, G.D.R.

This paper deals with management functions, which are connected to a data circuit-terminating equipment within a packet switching processor by a transport station and an internal data terminal equipment. There are many possibilities for communication between packet switching processors and privileged user terminals. Management functions are determined by requirements, which arise from system control, operating, maintenance and network management.

INTRODUCTION

Packet switching networks (PSN) are of increasing importance. They form an important base for public data communication. According to the performance requirements of the user a permanent high availability and reliability of such a network has to be guaranteed. Means for carrying out experiments should also exist to support the further development of the networks respectively the general growth in knowledge regarding the behaviour of information processing systems. The demand for an efficient control, operating and management of a PSN, the demand for management functions follows as a result.

Management functions have to make allowance for the special properties of PSN as distributed systems /1/. System engineers, operators on network control centres or service engineers use these functions usually.

CLASSES AND TASKS OF MANAGEMENT FUNCTIONS

The tasks of management functions of packet switching processors (PSP) are mainly determined by user requirements:

- guarantee of a high reliability and availability of the network;
- realization of a high troughput with low packet delay and cost;
- reconfiguration (adaptation to existent conditions, changing of system parameters, installation of new services).

Keeping up a high quality of service of the PSN requires the solution of four problems:

- system control;
- operating;
- maintenance;
- network management.

The quality of the service of a PSN includes beside availability, packet throughput, packet delay and reliability also special facilities of the PSN like priorities, troughput classes or closed user groups. A permanent control, registration respectively correction of important parameters of network components is necessary to keep up a high quality of service of the PSN.

Network control can be organized by distributing at each PSP or centralized at a network control center. Among other things system control covers control of system states, network load, quality of service and connection datas e.g. fees or statistics. Not only hardware based states of the PSP, modems, data transmission links and links to host computers respectively terminals should be controlled but also the states of the processes working in the PSP.

Operating encloses all measures which influence the work of the PSN in an activ manner. It involves the operating of the PSP and the switching typical operating among others changing of user status and link characteristics e.g. on/off.

During the development of a network management functions are necessary to influence the packet flow through the network by changing timeouts, counters and other parameters. With help of these functions every kind of exceptional circumstances or working conditions can be simulated on the real network. The functions should be implemented on each PSP and actions on remote PSP should be possible. For installation of new versions of software real situations have to be simulated. That means to adjust a variable network load by generating different packet flows through the network and possibilities to observe the network behaviour. With help of the results the efficiency of the installed configuration regarding the requirements at the time can be estimated and conclusions e.g. for increasing the throughput can be made. Management functions should support the efficient carrying out of experiments like:

- the test of new routing strategies;
- the preparation of new respectively additional connections to the PSN;
- the investigation of the influence of new communication mediums or
- the carrying out of performance analyses on selected network components.

Furthermore one task of operating is the line-up of the network or of single PSP. Both the local start of a single PSP and the common start of all network components should be possible. The geographical distance between PSP requires the automatic line-up of the PSN. A high degree of automation is necessary because /2/:

- a fast line-up increases the availability of the network;
- the packet switching inside of the network must not be impaired essentially;
- the status of the PSP isn't known exactly at any point of the network during line-up.

The maintenance of a PSN requires management functions for preventive system checks, error detection and checkup of new components. In this context the remote checkup of components is very important because in this way a centralization of maintenance staff can be obtained.

Functions for the network management solve problems like collection, storage and transmission of informations for the accounting of fees or statistics.These functions don't interact in the systems work.

THE COMMUNICATION BETWEEN MANAGEMENT FUNCTIONS OF REMOTE PACKET SWITCHING PROCESSORS

The reference to international standards in the stage of planning and implementation of PSN is very important for the application and the interconnection of networks. The ISO reference model for Open Systems Interconnection is general accepted in the field of data and computer communication. The lowest three layers form the basis of the data communication service of a PSN. With the CCITT recommendations X.25 and X.75 exist the instructions for the implementation and with recommendation X.121 exists the instruction for addressing.

According to the ISO reference model management functions are special management-application-entities and the protocols reside in the application layer /3/. Furthermore functions of the layer-management can exist in every layer and a part of them form a subset of the systems-management. The protocols reside in the application layer too.

Within a PSN a PSP works as relay open system. Therefore the functions and protocols which support the transmission of data are provided by the three lowest layers, i.e. only a data circuit-terminating equipment (DCE) is implemented in a PSP. However the services of higher layers are required for the communication between management entities of remote PSP. If there are not enough ressources (e.g. storage area) at the PSP the transport layer should be realized at least. This layer provides a safe transport servic independend from the communication medium. The transport service can be realized according to the

CCITT recommendations X.214 and X.224, which are based on ISO recommendations (ISO Draft Papers 8072 and 8073). A connectionless transport service is also possible.

Usually a data terminal equipment (DTE) is realized at an end system and a DCE is implemented at the PSP. The connection of the management entities to the data transmission service of the PSP can be realized in a similar way if an internal DTE (part of layer three of the DTE) is implemented in the PSP. The protocol between the internal DTE and the DCE should correspond to the recommendation X.25 (layer three). There is no necessity to use the service of the layers two and one of the PSP, i.e. the protocol data units can be exchanged directly between DCE and internal DTE (see figure 1). The advantage of this solution is, that there are no differences in the communication between PSP and the communication between PSP and user terminals respectively host computers. This can be useful for special applications or the further developement of the network. However a nonprivileged user never should be able to access to management functions. An unauthorized access to management functions can be prevented by forming a closed user group for the internal DTE of all PSP. On this basis it is easy to permit (or forbid) for privileged users the access to management functions .

There are no common recognized rules for the implementation of management functions. This will be understandable if we consider the special problems of network management, which should not exceed a given PSN. The realization of management functions is a local matter of a PSN and there is no necessity for a separate implementation of the layers five to seven. The implemented protocol for the management functions depends on:

- the importance of the transmitted data;
- the quality of the used transport service;
- the special hardware and software ressources at the PSP.

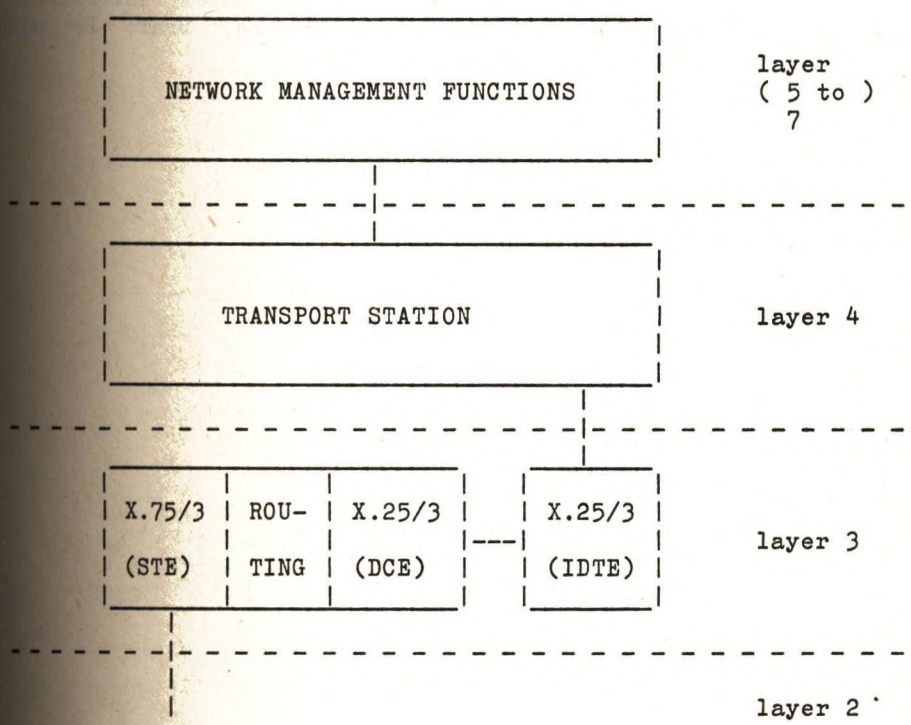
Number and kind of the implemented management functions are determined by the special purpose of the PSN, that means privat, public and experimental networks and the stage of the network developement and extension.

During network establishment management functions are necessary, which support the test of the network, the error detection and correction. Typical functions are:

- the generation of different packet flows;
- local and remote disply of storage areas of all PSP;
- local and remote modification of storage areas of all PSP.

With the further use of the network management functions are necessary for supervision and control of the PSN . Depending on the network purpose, several functions should be available:

- to test links;
- to connect and disconnect links;
- to signal link failures;
- to change user characteristics;
- to handle permanent virtual circuits;
- to collect and transmit statistic data.



STE ... signalling terminal equipment (part of layer three)
DCE ... data circuit-terminating equipment (part of layer three)
IDTE ... internal data terminal equipment (part of layer three)

Figure 1: Management functions within a packet switching processor

REFERENCES:

[1] Adler, H.-M., Management functions of packet switching networks, 5th Int. Symp. on Computer Networks, Frankfurt/Oder G.D.R., ZfR Informationen 83.02.

[2] Adler, H.-M., Experiences in the use of the management functions of the packet switching network KOMET, INFO 84, Dresden G.D.R.

[3] Information Processing Systems - Open Systems Interconnection- Basic Reference Model, International Standard ISO 7498

[4] Information Processing Systems - Open Systems Interconnection - Transport Service Definition, Draft Proposal ISO/DP 8072.

[5] Information Processing Systems - Open Systems Interconnection - Transport Protocol Specification, Draft Proposal ISO/DP 8073.

SESSION 6
FORMAL DESCRIPTION METHODS

THE OSI TRANSPORT SERVICE AND ITS FORMAL DESCRIPTION IN LOTOS

Giuseppe Scollo[1], Giuseppe Pappalardo[1], Luigi Logrippo[2], Ed Brinksma[3]

(1) Istituto di Informatica e Telecomunicazioni
Università di Catania
Italy

(2) Department of Computer Science
University of Ottawa
Canada

(3) Department of Informatics
Twente University of Technology
The Netherlands

An introduction is provided both to LOTOS, the Language fOr Temporal Ordering Specification whose definition is under way within ISO, and to the OSI Transport Service (TS). Concerning LOTOS, the language definition is briefly reviewed. The orthogonal combination in LOTOS of a CCS-derived temporal ordering language with an abstract data type notation is deemed to provide the language user with a large, yet sound, supply of specification facilities. Further, two examples are provided to show the expressiveness of the language, as well as its adaptability to several, quite different specification styles. Both 'global' and 'modular' specifications are shown to be possible. A model of the OSI TS is then formalized in LOTOS, in two variants that show different degrees of modularity. These descriptions set a suitable framework to highlight the discussion of two interpretation problems concerning its current definition. The two, in some way related, problems are: the question of atomicity of T-DATA primitives, and the description of the flow control mechanism. These problems were by no means unknown, but LOTOS enables one to discuss them in a precise way.

1. INTRODUCTION

This paper relates to current activities within ISO and CCITT on the Open System Interconnection (OSI) family of protocols and services [1]. Aim of this effort is to achieve the definition of a standardized family of data communication protocols, to enable cooperation between application processes running on remote computers of different types.

The basic protocol model commonly accepted in the literature, [2], [3], shows a number of entities (in this paper we shall limit ourselves to two) communicating with each other by using the services offered at given access points by a distributed machine called service provider. The entities are known as 'users' of the underlying service (fig. 1).

As usual in state-of-the-art software systems, OSI protocols are structured in a 'layered' fashion, as follows. A (N-1)-layer service provider is used by processes at layer N (called N-entities). The N-entities exchange information according to the N-protocol by using the services offered by the (N-1)-service-provider, thus

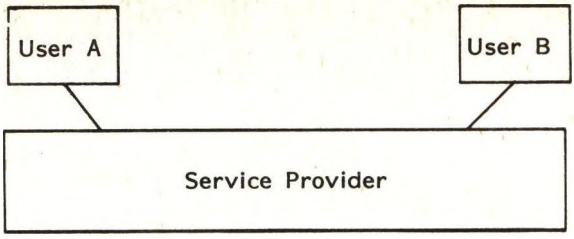


Figure 1. Two entities and a service provider

giving rise to another service provider at layer N (Fig. 2).

In order to add a new layer to this structure, two types of specification are needed: a) the service specification which defines the objective of this new layer; b) the protocol specification which defines the behaviour of the entities in the layer.

Protocol specifications are of course well known in distributed computing; less widely understood is the concept of service specification, and therefore we dedicate a few words to this point. As mentioned above, a service specification is the description of the externally observable behaviour of a service provider. Interestingly enough, it often turns out that, while the internal behaviour of the latter is rather complex, its externally observable behaviour is quite simple. For example, consider a service whose function is to provide a queue-like link between two entities by using a network of computers connected by unreliable communication links. The description of this service is quite simple, i. e. it reduces to the description of a queue; its implementation, however, might be very complex. In fact, keeping service specifications simple is an effective method for preventing the complexity of a system from building up dangerously as the design progresses from lower to higher layers (or viceversa).

Two approaches can be used to structure a service specification. We term the first as 'global'. In this approach, one single specification module shows directly how a sequence of events at one end of the service provider follows from a sequence of events at the other end. In the structure of a global specification, whatever the formalism employed, there appears no concurrent composition of independent, communicating submodules. This is the case in the second approach, the 'modular' one, where the sequence of primitives at either end of the service

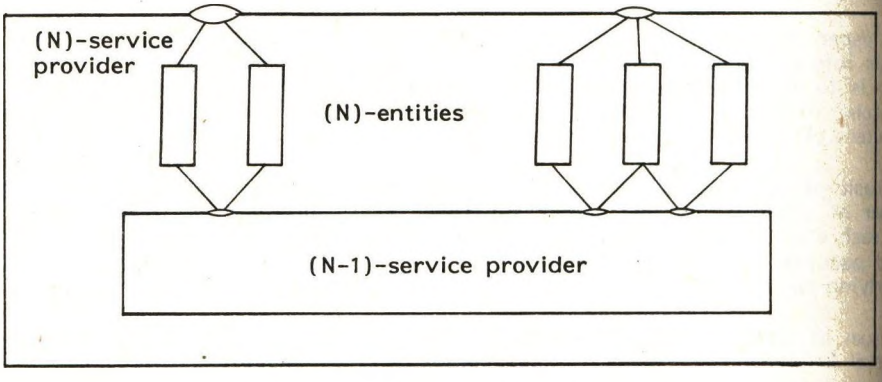


Figure 2. OSI layered architecture

provider results from interactions between submodules. Often a modular service description resembles the description of two communicating protocol entities; however, one must be careful to note that the structure shown in a service specification need not correspond to the actual structure of the underlying protocol.

Whatever approach is followed to structure service specifications, it is important that they be written by using formal specification tools. First of all, formality prevents certain types of ambiguities and wrong interpretations. Moreover, the use of formal tools enables a variety of computer-assisted error-detection methods to be applied to specifications. Such are syntactic and semantic checks, correctness proofs, interactive simulators, etc.

This paper deals with a formalization and related interpretation problems of the OSI Transport Service (TS) [4] (this International Standard is hereafter termed 'the IS' for short).

Section 2 presents a brief review of both informal and formal models of the OSI TS. This section points out the advantages of formality, especially the importance of using appropriate formal tools to avoid unnecessary overspecifications or incomplete specifications. Section 3 is a short review of LOTOS [5], the Language for Temporal Ordering Specification whose definition is under way within the ISO ad-hoc Group on Formal Description Techniques (FDT). In Section 4 a simplified TS model, as suggested in the IS, is formalized in LOTOS; two formalizations are provided, showing a different degree of modularity. In Section 5, some interpretation problems about the IS are proposed. The authors' conclusions are drawn in Section 6.

2. SHORT REVIEW OF PROPOSED MODELS OF THE OSI TRANSPORT SERVICE

The OSI transport service is defined in the IS as the service provided by the first four layers of the OSI hierarchy. In the words of these documents, 'the transport layer provides transparent transfer of data between session entities and relieves them of any concern with the detailed way in which reliable and cost-effective transfer of data is achieved'. In other words, from the point of view of the entities using it, the transport layer provides a service that is:
- end-to-end (i. e. it allows abstraction from the problems connected with routing among intermediate nodes);
- reliable, at least up to a certain specified 'residual error rate'.

This enables the user entities to dispense from the complex error detection and error correction procedures, that are among the most important functions of the first four layers.

The IS introduces the OSI TS by means of a model of a Transport Connection (TC) that is described as consisting of two transport users A and B interacting through TS Access Points (TSAP). Information is passed between TS users and the service provider through service primitives. The internal structure of the TS provider consists of two queues linking the two transport users, a queue for each direction of the information flow. Three different phases of behaviour can exist between two TS users: connection establishment, data transfer and release. The pair of queues is considered to be available for each potential TC; each queue is empty before connection establishment and can be returned by the TS provider to this state at any time. Objects are entered and removed from the queue as a result of interactions at the two TSAPs, subject to the control of the service provider. As we shall see later, these queues are somewhat more complex than FIFO queues. The distinction between service primitives and queue objects suggests a modular structure to describe a single-connection TS consisting of three interacting modules: the TSAP at side A, the TSAP at side B and the pair of queues. This is an example of a modular specification, since the end-to-end behaviour of the service provider is described as the result of the interaction

between several entities, which are, in this case, the queues and the TSAPs. Of course, consistently with what is pointed out in the IS, these entities are purely ideal ones, and need not correspond to identifiable entities in the implementation of the service.

The TS definition in the IS does not give a formal specification of the constraints on the order in which the TS primitives may occur. Rather, several types of informal description techniques are used to specify the various aspects of the service. 'Local' constraints [2], namely the allowed sequences of primitives at each TSAP, are described by means of state transition diagrams of the TSAP entity. 'Global' constraints, relating to the end-to-end behaviour of the service provider, are described in several ways. Time sequence diagrams show temporal relationships between the occurrences of one event at one end and another event at the other end. English statements and precedence diagrams describe the internal behaviour of the queues. Of course, when such dissimilar description methods are used together, it is difficult to ensure consistency and completeness of the description. Furthermore, ambiguities can arise not only because of the informality of these methods, but also because it is not always clearly stated how they relate the one to the other.

Due to this unsatisfactory specification of the TS, other models were proposed in the literature. All of them use a queue mechanism to identify the global sequences of service primitive at the TSAPs. In the global approach, this mechanism is described as one of the data structures of the single process which represents the TS, whereas in the modular approach the queue mechanism is described as one of the processes which give rise to the global behaviour of the service. An example of global approach can be found in [6] where the TS is described as a single process by using (one of the early versions of) LOTOS. The most recent definition of LOTOS is used in [7], where a global specification is provided of the TS data transfer and release phases.

Examples of modular approaches can be found in [8] and [9]; both these TS descriptions use a model directly derived from that proposed in the IS.

In particular, [8] describes the TS using a Numerical Petri Net (NPN) which consists of the interconnection of three subnets: the first subnet describes the allowable sequences at TSAP-A, the second one describes the allowable sequences at TSAP-B and the third one describes the interrelationship between the sequences at A and B by means of signals or messages received by a subnet, stored in an intermediate queue and delivered to the other subnet with a quasi-FIFO discipline (expedited data can take precedence over normal data). While the synchronization between the allowable actions at end-sides A and B is clearly expressed by NPNs, the absence of data structures determines an obscure description of the queue mechanism.

In [9] the queue mechanism is described by a notation based on abstract data type concepts and the TSAP modules at side A and B by state-transition diagrams. This approach handles informally the synchronization betweeen the three interacting modules, whereas it clearly represents the queue mechanism. The modular approach seems to have the advantage to represent separately the two TSAPs, thus permitting a better localization of the local management problems which may arise both in single and in multiconnection models.

The main problems with these approaches seem to be the difficulties to handle formally the queue mechanism (see [8]) and the composition in a single process of the interacting modules (see [9]). In the sequel (Section 4) we propose a LOTOS specification of the TS. This specification is modular, reflecting the IS model of TS, and - in our opinion - presents the following advantages:

a) it formally expresses the behaviour of the two interacting TSAPs and of the intermediate module(s);

b) it describes precisely module composition, by using proper LOTOS operators;
c) it gives the possibility of abstracting from the interactions internal to the composed structure, thus obtaining a formal description of the TS which is (hopefully) equivalent to a global one (the hope could be turned into certainty with the aid of automated tools);
d) it makes it easy to express local management problems of the TSAPs in either single or multiconnection models.

3. A SURVEY OF LOTOS

LOTOS is based on the principles of Milner's Calculus of Communicating Systems [10], to which the representation of process expressions is clearly inspired. For the representation of data and value expressions the abstract data type (ADT) language ACT ONE [11] is employed. In this section, however, we shall be concerned only with the representation of processes, since ADT techniques are well-known: some familiarity is assumed with the initial algebra approach, [12], and with the notation, e.g., of [13] and [14]; under these assumptions, type definitions appearing in the TS specification should be easily understood with the aid of comments.

Informal process model

In an informal model for LOTOS a process can be imagined as a black box capable of communication with its environment; the mechanism inside the box is not part of the model: rather, a process can be identified with the specification of its ability to communicate with its environment. The points of interaction of a process and (a process in) its environment are known as 'event gates', the atomic form of interaction being an 'event', i.e. a unit of synchronized communication between processes.

The representation of events

Communication between processes is synchronized in the sense that both processes must agree upon the execution of the same communication event at a common gate. Therefore, the definition of a single process must use the concept of 'potential' interaction, or 'event offer', instead of interaction or event. In a process definition such an offer is written:

g!E

where g is the event gate where the interaction is offered and E is an expression denoting the value offered. Synchronization takes place on the basis of identical offers between two parties, i.e. if two processes both offer the value of E at gate g then the corresponding event can occur. If a process offers g!v for all values v of a datatype t, then this is written as:

g?x:t

g?x:t may synchronize with g!E (provided E is of type t) and with g?y:t. The variable x is bound to the value on which synchronization occurs. Further, a finite combination of offers at one gate is allowed:

example:	g?x:int !(3+5) !true
matches with	g!3 ?y:int ?z:bool
or with	g?y:int ?z:int !true
but not with	g!3 ?y:bool ?z:bool
nor with	g!3 ?y:int !false

It is often convenient to restrict a '?' offer to a subset of the set of values comprising the datatype. The facility is often used of specifying such a 'restricted

event offer' as an event offer postfixed with a predicate on (some of) the variables appearing in the event notation:

examples: g?x:int [x>0]
g!true ?x:int ?y:bool ?z:bool [x<0 or y]

Although the process model is based on the principle of observability, also the internal events of a process (that by definition are not observable) play some role, as they may have observable consequences (usually surfacing as nondeterministic process behaviour). Internal events are all represented by the same symbol (as they cannot be distinguished):

i

The internal character of these events implies that for a process to execute them no matching offer in the process environment is required.

Process expressions

The behaviour of processes is represented in LOTOS by process expressions, that formally relate the order in which events may occur.
Below we define the construction of process expressions recursively, using the following notation:
- P, Pi (i∈N) represent process expressions,
- E, Ei (i∈N) represent value expressions,
- x,y,z, xi (i∈N) represent variables,
- t, ti (i∈N) represent type identifiers.

Inaction: stop

The completely inactive process is represented by stop. A process is in deadlock if at a given moment in its execution all alternatives for future actions are equivalent to stop.

Termination: exit(E1,...,En)

In LOTOS difference is made between a process that is no longer active as a result of deadlock and a process that has successfully terminated, so that control may be transferred to a subsequent process. The latter situation is represented by the process exit. If exit is parameterized by value expressions, upon termination the values of these expressions will be exported to the subsequent process (to see how this mechanism is used see Enabling).

example: exit(x+y,not(z))

Action-prefix: a ; P

This is a construct for defining the sequential order between events. The operator ';' prefixes an event offer to a process expression, yielding a process that offers the event a and, after the occurrence of a, transforms into P.

examples: g!false ; stop
g?x:int ; exit(x+1)
i ; exit

Choice: P1 [] P2

The process P1 [] P2 behaves either like P1 or like P2. In general, the choice

offered is resolved in the interaction of the process with the environment. However, depending on the form of the alternatives P1 and P2, it may also be resolved by the process autonomously. An alternative is enabled if its initial event offer matches with some offer of the environment or if it is an internal event i. If two or more alternative are enabled, one of them is selected nondeterministically.

example: a?x:int ; b?y:char ; exit(x,y) [] b?y:char ; a?x:int ; exit(x,y)

Summation: for x1:t1,...,xn:tn choice P

This construct is a generalization of '| |', summing over all values for x1,...,xn. It can be easily recognized that the '?' event offer notation is just shorthand of this operator, e.g.:

g?x:t ; P is shorthand of: for x:t choice g!x ;P
g?x1:t1 !y ... ?xn:tn ; P is shorthand of:
for x1:t1,..,xn:tn choice g!x1 !y ... !xn ; P

Parallel composition: P1 || P2

This operator models concurrency as the arbitrary interleaving of the event offers of its argument processes, with the possibility of synchronization between them on matching offers. Such internal synchronizations result in i events of the composite process. P1 || P2 terminates iff both arguments P1, P2 do so.

example: a!5 ; exit(true) || a?x:int ; exit(x)

by the 'expansion' theorem can be proved equivalent to:

a!5 ; a?x:int ; exit(true,x) [] a?x:int ; a!5 ; exit(true,x)
[] i ; exit(true,5)

Two other operators for parallel composition are available in LOTOS; we shall not use them in the specifications of Section 4 and, therefore, we refer the interested reader to the LOTOS definition [5].

Restriction: P\[A]

This operator removes P's event offers at gates in the list A; in the absence of alternatives to A-actions this may introduce deadlock. Together with '||' it is often used to enforce synchronization between concurrent processes.

example: (a!5 ; exit(true) || a?x:int ; exit(x))\[a]
is equivalent to:

i ; exit(true,5)

Conditional: [B] -> P

B is an expression of boolean type; if the value of B is true, then the resulting process expression is equivalent to P, if the value of B is false to stop.

example: a?x:int ; ([x>=0] -> b!x ; exit
[]
[x<0] -> b!(-x) ; exit
)

This process inputs an integer at a and outputs its absolute value at b.

It is easily recognized that the 'restricted event offer' notation is just shorthand

of a suitable combination of summation and conditional, e.g.:

g?x:t [B(x)] ; P is shorthand of: for x:t choice [B(x)] -> g!x ; P

Enabling: P1 >> def x1:t1,..,xn:tn in P2

While ';' defines sequential order by combining an event offer with a process expression, '>>' defines a sequential order by combining two process expressions; the successful termination of P1 'enables' the execution of P2. The variables x1,.., xn are the local names in P2 of the values exported by P1 upon termination (exit).

example: (a?x:int ; exit(x) | | b?y:int ; exit(y)) >> def n:int in c!n ; exit

This process inputs an integer at either a or b, then outputs it at c.

Disabling P1 [> P2

The intuitive meaning of this operator is that process P2 may disrupt or prevent the execution of process P1; once the execution of P1 has been disrupted it will not be resumed again. If P1 has terminated, disruption can no longer take place; this counts as a termination of the whole construct. A termination of P2 (after having disrupted P1) also counts as a termination of the whole construct.

example: a?x:t ; b!x ; exit [> c!break ; exit

is equivalent to:

```
c!break ; exit
[]
a?x:t ; ( c!break ; exit
          []
          b!x ; (c!break; exit [] exit)
        )
```

Local variable declarations: let x1=E1,..,xn=En in P

example: a?x:int ?y:int ;
let sum=x+y in b!sum; c!(1/sum); exit(sum)

Note: the following (descending order) hierarchy among operators is used in order to reduce the necessity of bracketing:

\ ; [B]-> [] || [> >> let

Process instantiation: p [a1,..,an] (E1,..,Em)

A LOTOS specification is essentially a list of recursive process abstractions, one of which represents the final object of specification. A named process abstraction has the following format:

process p [g1,...,gn] (x1:t1,...,xm:tm | t1',...,tk') := P endproc

where p is the name of the abstraction,
g1,...,gn are the formal gate parameters,
x1,...,xm are the formal value parameters, with types t1,...,tm, and
t1',...,tk' are the types of the values exported by P on termination,
P is the defining process expression.

A process instantiation is the occurrence in a process expression of the name of a process abstraction, such as p in the previous example, with actual parameters for formal gate and value parameters. The interpretation of such an occurrence is the expected one, .viz. that a process instantiation p is equivalent to the process P with the proper substitution of actual parameters for the formal ones. Note that an instantiation of p exports k values of the types listed in its declaration.

```
example: process lowcut [g] (n:int ! int) :=
             g?x:int;([x>n] -> exit(x)
                      []
                      [x=<n] -> exit(n)
                     )
         endproc
```

can be used to define a process that inputs one integer at gate a and, if positive, outputs it at gate b, otherwise 0 is output at b, as follows:

```
lowcut [a](0) >> def y:int in b!y ; exit
```

4. LOTOS SPECIFICATIONS OF THE TRANSPORT SERVICE

This section is devoted to the specification in LOTOS of a model of the Transport Service, that we intend to be a formalization of the concepts of the IS. Two alternative specifications are provided in order to highlight the language support to different specification styles.

4.1. MORE ON THE 'IS' MODEL OF TRANSPORT SERVICE

Table 1 lists the primitives available in the Transport Service. In the OSI architecture requests and responses are primitives presented by a user to the service provider for delivery to the other user. Indications and confirms are primitives presented by the service provider to a user as a consequence of a primitive issued by the other user; indications can be also the result of an internal decision of the provider itself.

Of course, the service provider will not necessarily accept a primitive presented to it. For example, a T-CONNECT response is quite inappropriate after the connection establishment phase. The IS defines the allowed sequences of primitives at a Transport Connection End Point (TCEP) by means of a state transition diagram. As we have seen in Section 2, these 'local' constraints do not thoroughly specify the TS; requests and responses appearing in these sequences are issued by the 'local' user, but indications and confirms may be the result of the remote user's actions; therefore, in order to complete this 'local' description, the relationship ought to be described between the events at the two TCEPs associated by a connection.

As mentioned above, the IS suggests that this relationship may be described by means of two queues connecting two remote TCEPs and conveying objects, i.e. the parameters of the primitives; further, the IS remarks that other equivalent ways of describing this relationship exist, e.g. (1) defining 'constraints on the local ability to invoke primitives' or (2) 'service procedures defining particular sequencing constraints'. We find that the queue model is adequate to our specification needs and we formalize it in two different styles.

4.2. OUTLINE OF THE SPECIFICATIONS

In this section we outline the model underlying our TS specifications and we discuss some of their features.

Phase	Service	Primitive	Parameters
TC-establishment	TC-establishment	T-CONNECT request	(Called Address, Calling Address, Expedited Data option, Quality of Service, TS User Data)
		T-CONNECT indication	(Called Address, Calling Address, Expedited Data option, Quality of Service, TS User Data)
		T-CONNECT response	(Quality of Service, Responding Address, Expedited Data Option, TS User Data)
		T-CONNECT confirm	(Quality of Service, Responding Address, Expedited Data Option, TS User Data)
Data transfer	Normal data transfer	T-DATA request	(TS User Data)
		T-DATA indication	(TS User Data)
	Expedited data transfer	T-EXPEDITED-DATA request	(TS User Data)
		T-EXPEDITED-DATA indication	(TS User Data)
TC release	TC release	T-DISCONNECT request	(TS User Data)
		T-DISCONNECT indication	(Disconnect Reason, TS User Data)

Abbreviations: TS = Transport Service
TC = Transport Connection

Table 1. Transport Service Primitives (TSP)

Two variants for the process structure of the model are shown in Figure 3.

Processes CallingTSAP and CalledTSAP specify the local constraints on the user's ability to invoke primitives, just like the IS state transition diagram mentioned above. Process Transfer specifies the global constraints that relate events taking place at the two TSAPs.

In the first specification below, Section 4.3, process Transfer shows no further static structure, and is parameterized with two queues, one per direction of communication. In the variant of Section 4.4 Transfer consists of two submodules,

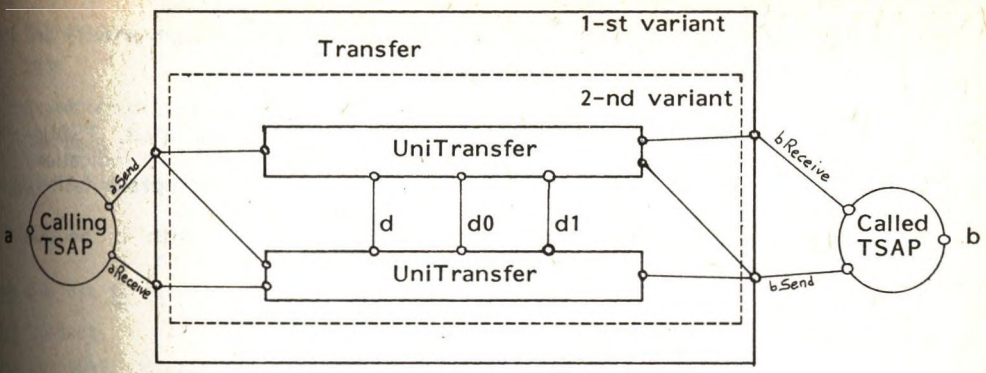


Figure 3. TS model

one per direction of communication, each being an instance of process UniTransfer parameterized with a single queue. These two processes communicate at internal gates of Transfer in such a way that TC termination is properly specified.

The objects that may be placed in the queues are: connect objects (each conveying the parameters of a T-CONNECT primitive), octets of 'normal' data (conveying the parameters of T-DATA primitives, also called TSDUs, for Transport Service Data Units), 'expedited' TSDUs (conveying the parameters of T-EXPEDITED-DATA primitives), and disconnect objects, which can be generated either because of a user's T-DISCONNECT request or because of spontaneous actions by the service provider. The queue has basically a FIFO structure with limited (but not necessarily either fixed or determinable) capacity. The FIFO discipline is enforced among normal data objects; on the other hand, expedited TSDUs, which behave FIFO the one with respect to the others, can bypass normal data objects. In the IS model, the queues also convey another kind of objects: end-of-TSDU indications, which delimit normal TSDUs and can be bypassed by expedited TSDUs.

Table 2 shows the queue objects corresponding to each primitive. Three main restrictions affect this model, that are introduced for simplifying the presentation and the discussion:

at 'user' gate	at 'send' gate
T-CONNECT request	C1 object
T-CONNECT response	C2 object
T-DATA request	N object
T-EXPEDITED-DATA request	X object
T-DISCONNECT request	D object

at 'receive' gate	at 'user' gate
C1 object	T-CONNECT indication
C2 object	T-CONNECT confirm
N object	T-DATA indication
X object	T-EXPEDITED-DATA indication
D object	T-DISCONNECT indication

Table 2. Correspondance between TSPs and queue objects

- a single connection is described;
- quality of service negotiation is not specified;
- possible loss of message or corruption of data (due to residual errors) are not specified.

Two further assumptions affect the specifications below:
- 'normal' objects are assumed to consist of whole, unfragmented, TSDUs: no distinct object is hence defined corresponding to an end-of-TSDU indication;
- the size of each queue is unbounded, which is the simplest representation of a 'not necessarily either fixed or determinable' boundedness.

These potentially very controversial assumptions are dealt with in Section 5, where a way is indicated to alternative formalizations in LOTOS.

4.3. A SPECIFICATION OF THE TS MODEL

```
specification  Transport_Connection
```

The first type definitions are intended to formalize the contents of Tables 1 and 2. We specify only a few of them formally, the others being specifiable in a similar way. Each TSP type definition collects those TSPs which relate to the same service. TSP names (with the standard abbreviations used in [15]) denote constructor functions of the type.

```
type T-CONNECT is C1Object with C2Object with
sort T-CONNECT
opns TCONreq, TCONind: C1Object -> T-CONNECT
     TCONresp, TCONconf: C2Object -> T-CONNECT
endtype (* T-CONNECT *)
...
type T-DISCONNECT is OctetString with DObject with
sort T-DISCONNECT
opns TDISreq: OctetString -> T-DISCONNECT
     TDISind: DObject -> T-DISCONNECT
endtype (* T-DISCONNECT *)
```

The structure of TSP parameters is formalized by the type definition of the queue objects. We use 'ZObj' operators, with Z out of {C1, C2, N, X, D} (see Table 2), to denote the type constructor functions, whereas individual parameter names denote observer functions which access the parameter structure components. Notice however the exception to this policy in the definition of the type DObject below; the two cases of disconnection, either user- or provider-initiated, correspond to distinct constructors; the former applies to an octetstring, viz. the user data in a T-DISCONNECT request, whereas the latter is a nullary function (i.e. a function that has no argument and, therefore, takes a constant value).

```
type C1Object is TSAddress with TSQuality with Boolean with OctetString with
sort C1Object
opns C1Obj: TSAddress, TSAddress, TSQuality, Boolean, Octetstring -> C1Object
     CallingAddress, CalledAddress: C1Object -> TSAddress
     TQOS: C1Object -> Boolean
     ExpeditedOption: C1Object -> Boolean
     UserData: C1Object -> OctetString
```

```
eqns CallingAddress (ClObj(a1, a2, q, x, u)) = a1
     CalledAddress  (ClObj(a1, a2, q, x, u)) = a2
     TQOS           (ClObj(a1, a2, q, x, u)) = q
     ExpeditedOption(ClObj(a1, a2, q, x, u)) = x
     UserData       (ClObj(a1, a2, q, x, u)) = u
endtype (* ClObject *)

...

type DObject is OctetString with
sort DObject
opns UserDObj: OctetString -> DObject
     ProviderDObj: -> DObject
     UserData: DObject -> OctetString
eqns UserData(UserDObj(x)) = x
     UserData(ProviderDObj) = Empty
endtype (* DObject *)
```

The definition below formalizes a FIFO structure where expedited objects may take precedence over normal ones (function 'Forward') and where disconnect objects may destroy preceding objects (function 'Delete'). We assume a parameterized definition of the type Queue(Element) with the usual functions New: -> Queue, Add: Element, Queue -> Queue, Remove: Queue -> Queue, Head: Queue -> Element, IsEmpty: Queue -> Boolean. The sort 'Object' is intended to consist of the disjunct sum of the sorts defined for the various objects in Table 2, such that an observer function Kind: Object -> ObjectKind (with C1, C2, N, X, D: -> ObjectKind) may be used to determine the original sort of the object. Also, a boolean equality operator 'eq' is assumed to be defined on sort ObjectKind. The definition of the type TSQueue is obtained in two steps. In the first step, a type QueueOfObjects is defined which is simply the actualization of Queue(Element) by Object.

```
type QueueOfObjects is Queue(Element) actualized by Object using
sortfunction  TSQueue for Queue,  Object for Element
endtype (* QueueOfObjects *)
```

In the second step, TSQueue is defined as consisting of the enrichment of QueueOfObjects with further constructors, viz. 'Forward' and 'Delete', and with the observers XNin, DC1in, DC2in, DNin, DXin: TSQueue -> Boolean.

```
type TSQueue is QueueOfObjects with Boolean with
opns Forward, Delete: TSQueue -> TSQueue
     XNin, DC1in, DC2in, DNin, DXin: TSQueue -> Boolean
     if.then.else.: Boolean, TSQueue, TSQueue -> TSQueue
eqns
```

Each of the new observers tests whether or not a pair of adjacent objects is present in the queue that are of given kinds, e.g. for any q, XNin(q) is true iff q contains some normal object which is immediately followed by an expedited one.

```
     XNin(New) = false
     XNin(Add(x,New)) = false
     XNin(Add(x,Add(y,q))) = (eq(Kind(x),X) and eq(Kind(y),N)) or XNin(Add(y,q))
```

... analogously for DC1in, DC2in, DNin, and DXin.
The constructor function Forward swaps the last expedited object preceded by a normal object with the latter. If no such pair of adjacent objects exists, the queue remains unchanged.

```
Forward(New) = New

Forward(Add(x,New)) = Add(x,New)

Forward(Add(x,Add(y,q))) =    if eq(Kind(x),X) and eq(Kind(y),N)
                              then Add(y,Add(x,q))
                              else Add(x,Forward(Add(y,q)))
```

The constructor function Delete destroys the object that precedes a disconnect object (again, if no such pair exists, the queue remains unchanged); if the disconnect object is preceded by an object of kind C1, then both are destroyed.

```
Delete(New) = New

Delete(Add(x,New)) = Add(x,New)

Delete(Add(x,Add(y,q))) =     if eq(Kind(x),D)
                              then if eq(Kind(y),C1) then q else Add(x,q)
                              else Add(x,Delete(Add(y,q)))
```

The if.then.else. construction returns the value of either the second or the third argument, depending on the truthvalue of the first argument.

```
if true then p else q = p

if false then p else q = q

endtype (* TSQueue *)
```

The definition below formally describes the structure shown in Fig. 3. CallingTSAP, CalledTSAP and Transfer are described as terminating processes, as a Transport Connection must also be a terminating process. The specification is parameterized with two boolean values, denoting the local availability of the expedited service at each TSAP. Process Transfer starts with two empty queues, one per direction of communication.

```
process TransportConnection [a,b] (aExp, bExp: Boolean) :=

( CallingTSAP[a, aSend, aReceive](aExp)

|| Transfer[aSend, aReceive, bSend, bReceive](New, New)

|| CalledTSAP[b, bSend, bReceive](bExp)

)\[aSend, aReceive, bSend, bReceive]

endproc (* TransportConnection *)
```

The process below describes the sequences of actions at the calling TSAP: after a T-CONNECT request, a T-CONNECT confirm followed by a DataTransfer process will occur.

A T-DISCONNECT primitive may interrupt (operator [>) any such started sequence at any time, thus causing the connection to be released and process CallingTSAP to terminate.

The expedited service may or may not be made available by the service provider, and if it is available, its use must be negotiated by the users. The boolean parameter ExpService is true iff the expedited service is locally available. The boolean value ExpSelected is true iff the option is available at both ends and its use is agreed upon by the users.

```
process CallingTSAP [User, Send, Receive] (ExpService: Boolean) :=
ConnectReq[User, Send](ExpService) >>
( (ConnectConf[User, Receive] >> def ExpSelected: Boolean in
  DataTransfer[User, Send, Receive](ExpSelected)
 ) [> Disconnect[User, Send, Receive]
)
endproc (* CallingTSAP *)
```

The next process looks quite similar to CallingTSAP above but, unlike ConnectReq above, ConnectInd exports the two boolean values Terminated and ExpProposed. The former tells whether or not early TC termination occurred; the latter tells whether or not the expedited option was proposed in the T-CONNECT indication.

```
process CalledTSAP [User, Send, Receive] (ExpService: Boolean) :=
ConnectInd[User, Receive](ExpService) >> def Terminated, ExpProposed: Boolean in
( [Terminated] -> exit
[] [not Terminated] ->
  ( (ConnectResp[User, Send](ExpProposed) >> def ExpSelected: Boolean in
      DataTransfer[User, Send, Receive](ExpSelected)
    ) [> Disconnect[User, Send, Receive]
  )
)
endproc (* CalledTSAP *)
```

Process DataTransfer below handles the local transfer of TSDUs. While normal TSDUs are always accepted, the acceptance of expedited TSDUs depends on the boolean parameter ExpSelected.

```
process DataTransfer [User, Send, Receive] (ExpSelected: Boolean) :=
(Data[User, Send, Receive] (ExpSelected)
 [] [ExpSelected] -> ExpeditedData[User, Send, Receive]
) >> DataTransfer[User, Send, Receive](ExpSelected)
endproc (* DataTransfer *)
```

The definition of CallingTSAP and CalledTSAP is completed by the definitions below, where each process, in the usual case, accepts a given primitive from the TS user (at gate 'User') and passes the corresponding object to Transfer (at gate 'Send') or vice versa (with 'Receive' replacing 'Send'). A few exceptions need to be also specified, so that proper behaviour results with respect to expedited and disconnect objects. The connection establishment phase is described, in the 'local' process CallingTSAP, by ConnectReq followed by ConnectConf, whilst, in CalledTSAP, by ConnectInd followed by ConnectResp. In a global view, these processes are activated in the same order as they are presented in the sequel.

```
process ConnectReq [User, Send] (ExpService: Boolean) :=
for x:ClObject choice [Length(UserData(x)) =< 32] -> User!TCONreq(x) ;
( [ExpService or not ExpeditedOption(x)] -> Send!x ; exit
[] i ; Send!(CallingAddress(x),CalledAddress(x),TQOS(x),false,UserData(x)) ; exit
)
endproc (* ConnectReq *)
```

```
process ConnectInd [User, Receive] (ExpService: Boolean | Boolean,Boolean) :=
Receive?x:C1Object ;
   ( [ExpService or not ExpeditedOption(x)] ->
     User!TCONind(x) ; exit(false, ExpeditedOption(x))
   []
     i ;
     User!TCONind(CallingAddress(x),CalledAddress(x),TQOS(x),false,UserData(x)) ;
     exit(false,false)
   )
[]
Receive!quit ; exit(true,false)
endproc (* ConnectInd *)

process ConnectResp [User, Send] (ExpProposed: Boolean | Boolean) :=
for x:C2Object choice
  [(ExpProposed or not ExpeditedOption(x)) and Length(UserData(x)) =< 32] ->
  User!TCONresp(x) ; Send!x ; exit(ExpeditedOption(x))
endproc (* ConnectResp *)

process ConnectConf [User, Receive] (|Boolean) :=
Receive?x:C2Object ; User!TCONconf(x) ; exit(ExpeditedOption(x))
endproc (* ConnectConf *)
```

The 'quit' signal is used in ConnectInd to deal with early termination (a C1 object destroyed by a D object in the queue) so that all the constituent processes of TransportConnection can terminate.

Notice how ConnectReq and ConnectInd specify: 1) the necessity of not providing the expedited data option whenever either this is locally not available at any of the two ends or it is not requested by the calling user, and 2) the possibility that successful selection of this option might not take place even when it is available at both ends and it is requested by the calling user. The definition of ConnectResp prevents the called user from selecting this option if the option was not proposed in the T-CONNECT indication.

Negotiation of the expedited data option is ruled differently in the IS, insofar as the service provider is not allowed to take an active role; in other words, if the calling user requests the expedited data option in the T-CONNECT request, the option must appear as proposed in the T-CONNECT indication, and it is solely the called user that may accept or refuse selection of the option. This service requirement seems to be somewhat 'too strong' with regard to the current transport protocol definition [15], where the protocol entities of class 0 never implement the expedited data service. The authors' formalization enhances consistency between service and protocol definitions, but implies a change in the TS IS.

The data transfer phase is locally described by the recursive definition of the DataTransfer process given above, whose constituents are the Data and ExpeditedData processes below. Each of these processes handles just one TSP interaction in its lifetime, but for DataReq and DataInd (the constituents of process Data), as explained in the sequel. Whenever the expedited data option is selected, two

possibilities are specified after input of a normal TSDU: either the latter is output or an expedited TSDU is further input and then output before the output of the normal TSDU previously accepted. The expedited TSDU is aimed at soliciting the receiving TS user to allocate resources when the normal flow is blocked.

```
process Data [User, Send, Receive] (ExpSelected:Boolean) :=
DataReq[User, Send](ExpSelected) [] DataInd[User, Receive](ExpSelected)
endproc (* Data *)

process DataReq [User, Send] (ExpSelected:Boolean) :=
for x:NObject choice [Length(x) > 0] -> User!TDATreq(x) ;
(Send!x ; exit
[] [ExpSelected] -> ExpReq[User, Send] >> Send!x ; exit
)
endproc (* DataReq *)

process DataInd [User, Receive] (ExpSelected:Boolean) :=
Receive?x:NObject ;
(User!TDATind(x) ; exit
[] [ExpSelected] -> ExpInd[User, Receive] >> User!TDATind(x) ; exit
)
endproc (* DataInd *)

process ExpeditedData [User, Send, Receive] :=
ExpReq[User, Send] [] ExpInd[User, Receive]
endproc (* ExpeditedData *)

process ExpReq [User, Send] :=
for x:XObject choice [Length(x) > 0 and Length(x) =< 16] ->
  User!TEXreq(x) ; Send!x ; exit
endproc (* ExpReq *)

process ExpInd[User, Receive] :=
Receive?x:XObject ; User!TEXind(x) ; exit
endproc (* ExpInd *)
```

The release phase is locally described by process Disconnect below. Notice that DiscReq not only passes a disconnect object at Send, but also accepts possible disconnect objects incoming at Receive; similarly, at gate User, DiscInd not only gives a T-DISCONNECT indication, but also accepts a T-DISCONNECT request. This avoids deadlocks in the release phase.

```
process Disconnect [User, Send, Receive] :=
DiscReq[User, Send, Receive][] DiscInd[User, Receive]
endproc (* Disconnect *)
```

```
process  DiscReq[ User, Send, Receive ] :=
for x: OctetString choice [ Length(x) =< 64 ] -> User!TDISreq(x) ;
(Send!UserDisc(x) ; exit
[] Receive?x:DObject ; exit
)
endproc (* DiscReq *)

process  DiscInd[User, Receive] :=
Receive?x:DObject ;
(User!TDISind(x) ; exit
[] for x:OctetString choice [ Length(x) =< 64 ] -> User!TDISreq(x) ; exit
)
endproc (* DiscInd *)
```

The definition of process Transfer consists of three main cases:
- i/o of an object of kind other than disconnect (as specified in the definition of process CNX); the interested queue is updated accordingly by the related instance of CNX, which exports this new value of the queue; then Transfer is recursively invoked;
- user-initiated release: a disconnect object is accepted at either 'aSend' or 'bSend' gate and added to the interested queue; the latter is the parameter of the proper instance of process Release which is invoked thereafter;
- provider-initiated release: a provider disconnect object is internally generated and added to both queues; in this case TC release is specified as consisting of the parallel composition of two similar processes Release, one per TSAP end.

```
process Transfer [aSend, aReceive, bSend, bReceive] (qab, qba: TSQueue) :=
     CNX[aSend, bReceive](qab) >> def qab in
        Transfer[aSend, aReceive, bSend, bReceive](qab, qba)
  [] CNX[bSend, aReceive](qba) >> def qba in
        Transfer[aSend, aReceive, bSend, bReceive](qab, qba)
  [] aSend?x:DObject ; Release[bSend, bReceive](Add(x,qab))
  [] bSend?x:DObject ; Release[aSend, aReceive](Add(x,qba))
  [] i ; ( Release[bSend, bReceive](Add(ProviderDisc,qab))
         || Release[aSend, aReceive](Add(ProviderDisc,qba))
         )
endproc (* Transfer *)
```

The definition below consists of three cases:
- addition to the queue of an object of kind other than disconnect;
- removal from the (non empty) queue of its heading object;
- an expedited object overtaking a preceding normal one.

In all cases the modified value of the queue is exported.

```
process CNX [Send, Receive ] (q: TSQueue | TSQueue) :=
      Send?x:Object [not eq(Kind(x),D)] ; exit(Add(x,q))
   [] [not IsEmpty(q)] -> Receive!Head(q) ; exit(Remove(q))
   [] [XNin(q)] -> exit(Forward(q))
endproc (* CNX *)
```

The definition of process Release describes the interactions which may take place until local termination of the TC. In the order specified below, they are:
- output of a disconnect object: this event leads to termination; notice that the disconnect object may be provider-initiated even though a user-initiated disconnect object is removed from the queue;
- removal from the queue of its heading object, of kind other than disconnect;
- input of an object of kind other than disconnect: the object is discarded;
- input of a disconnect object: this event leads to termination;
- an expedited object overtaking a preceding normal one;
- deletion of a connect-2 or normal or expedited object by a disconnect one.
- deletion of a connect-1 object by a disconnect one, in which case the latter is also deleted, and the process terminates after synchronization at the Receive gate on the 'quit' signal.

```
process Release [Send, Receive] (q: TSQueue) :=

  [eq(Kind(Head(q)),D)] ->
    (Receive!Head(q) ; exit [] Receive!ProviderDObj ; exit)

[] [not eq(Kind(Head(q)),D)] ->
    Receive!Head(q)) ; Release[Send, Receive](Remove(q))

[] Send?x:Object [not eq(Kind(x),D)] ; Release[Send, Receive](q)

[] Send?x:DObject ; exit

[] [XNin(q)] -> i ; Release[Send, Receive](Forward(q))

[] [DC2in(q) or DNin(q) or DXin(q)] -> i ; Release[Send, Receive](Delete(q))

[] [DC1in(q)] -> i ; Receive!quit ; exit

endproc (* Release *)

endspec (* Transport Connection *)
```

4.4. A VARIANT OF THE SPECIFICATION

The second variant of our specification inherits both the type definitions given above and the process definitions of CallingTSAP, CalledTSAP, of their subprocesses, and of TransportConnection (but the instantiation of Transfer shows no value parameters, in agreement with the definition of Transfer below). According to Fig. 3, the variant affects the definition of Transfer and of its subprocesses. Transfer now consists of the concurrent composition of two instances of a process UniTransfer, one per direction of transmission. The third gate of each instance is only used in the release phase. Moreover, the two processes are interconnected at three additional gates, d, d0, d1, where they synchronize at the beginning of the release phase.

```
process Transfer [aSend, aReceive, bSend, bReceive ] :=

( UniTransfer [aSend, bReceive, bSend, d, d0, d1 ] (New, true, false)
|| UniTransfer [bSend, aReceive, aSend, d, d1, d0 ] (New, false, false)
)\[d, d0, d1]
endproc (* Transfer *)
```

UniTransfer is parameterized with the value of the queue (initially empty) and with two booleans, viz. 'init', which is true iff the process at gate Send is CallingTSAP (i.e. the TS user is the TC initiator), and 'start', which must be false initially and becomes true as soon as the first object has been entered.
Processes CNX and Release are defined exactly as in the former specification.
The definition of UniTransfer is quite obvious, but for the beginning of the release phase, that can be so understood:

- UniTransfer is compelled to terminate by synchronization at dR on signal U, meaning that a user-initiated disconnect object has been accepted by the other process. This signal is generated by UniTransfer at dS only after such an input (remember that in each instance of UniTransfer gate dS is actualized by the gate actualizing dR in the other instance).

- If a user-initiated disconnect object has been accepted by both processes, mutual termination is enforced by synchronization at gate d on signal U.

- UniTransfer may also synchronize at gate d on signal P for the generation of a provider-initiated disconnect object, which is added to the queue. If the process at the input gate is CallingTSAP, this generation may only take place after a connect object has been entered.

```
process UniTransfer [Send, Receive, Ignore, d, dS, dR ]
                        (q:TSQueue, init, start: Boolean) :=
    (CNX[Send, Receive](q) >> def q:TSQueue in
        UniTransfer[Send, Receive, Ignore, d, dS, dR](q, init, start) )
  [] dR!U ; exit
  [] Send?x:DObject ; (  dS!U ; Release[Ignore, Receive](Add(x,q))
                       [] d!U ; exit
                      )
  [] [start or not init] -> d!P ; Release[Ignore, Receive](Add(ProviderDObj,q))
endproc (* UniTransfer *)
```

One may wonder why this variant, somewhat more complicated than the former, is worthy of any consideration. We have introduced it not only to illustrate the adaptability of the language to different specification styles, but also because it will result in a simpler specification of the TS features related to flow control, as it is going to be shown in the next Section.

5. INTERPRETING THE 'IS' MODEL

How to formalize the fact, expressed in the IS model, that a TS queue shows a bounded capacity which, however, is neither fixed nor known? It seems that what should be formalized is a nondeterministic upperbound that, at any time in the data transfer phase, may prevent a T-DATA request from being accepted by the TS provider. This is to be related to the notion of end-to-end flow control, which can be either explicit or implicit (so called 'backpressure' flow control). The IS indeed states that 'the queue model is used to introduce the flow control feature. The ability of a TS user to add objects to a queue will be determined by the behaviour of the TS user removing objects from that queue and the state of the queue'. The topic of properly specifying an unboundedly finite capacity arises also with respect to another interpretation problem of the IS, related to the size of normal TSDUs exchanged at service boundaries. Although session protocol entities do negotiate a maximum TSDU size in the session connection establishment phase [16], this fact is not relevant from the point of view of the TS definition. These considerations are further developed in the sequel.

5.1. UNLIMITED BOUNDEDNESS OF SERVICE QUEUES

A bounded queue allows the receiving side to exert control over the rate at which data units are transmitted by the other side without any special mechanism; this is what is known as implicit end-to-end flow control, or flow control via backpressure. For some of its classes, the OSI transport protocol defines the possib-

ility of negotiating, in the connection establishment phase, the so-called 'explicit flow control option'. Under this option, a credit allocation mechanism is defined in such a way that the receiver side may communicate to the other end the availability of new resources, either because of the removal of data units from the queue or because of some local allocation of new resources. Only if the explicit flow control option is agreed upon, the 'expedited option' may also be mutually agreed, in which case the expedited service provides the transmitting side with a means of soliciting the allocation of additional credits by the other side.

In this framework one can better understand not only the purpose of a service like the expedited, but also the reason why the IS require that the addition of an expedited TSDU or disconnect object to the queue must never be prevented by the addition of normal data or of an end-of-TSDU indication. The implicit flow control via a nondeterministic upperbound for the queue capacity can be formalized, in the variant of Section 4.4 above, by:

```
process UniTransfer [Send, Receive, ... ] (q:TSQueue, ...) :=

( (for full:Boolean choice
    (i ; CNX[Send, Receive](full, q) >> def q in
            UniTransfer[Send, Receive, ...](q, ...)
    )
  )
...
endproc (* UniTransfer *)


process CNX [Send, Receive ] (full:Boolean, q:TSQueue | TSQueue) :=
  [IsEmpty(q) or not full] -> Send?x:Object [not eq(Kind(x),D)] ; exit(Add(x,q))
[] [not IsEmpty(q)] -> Receive!Head(q) ; exit(Remove(q))
[] [XNin(q)] -> exit(Forward(q))
[] [full and not XNin(q)] -> Send?x:XObject ; exit(Add(x,q))
endproc (* CNX *)
```

It is important to stress the necessity of the internal event i, so that the choice specified in Transfer excludes the environment from cooperating in the determination of the value of 'full'.

5.2 UNBOUNDED FINITE SIZE OF SERVICE DATA UNITS

As we have recalled, the IS states that no upperbound is to be specified on the TSDU size (it is only stated that the TSDU size is greater than zero). This topic should however be related to the atomicity of TS primitives, assumed in the specifications of this paper consistently with the view of clauses 11-14 of the IS, as well as with the general statement here quoted from the OSI Service Conventions [17]: 'The occurrence of a primitive is a logically instantaneous and indivisible event. The event occurs at a logically separate instant, which cannot be interrupted by another event'.

This statement carries major implications on the definition of T-DATA primitives just because of the finite but unbounded size of their parameter (the TSDU). These implications become apparent in the IS model, which admits a different conceptual view when dealing with a service-oriented description of flow control: 'The description of flow control requires a less abstract description than that used for describing sequences of primitives in clauses 11-14. The T-DATA primitives, which are there treated as indivisible, are here subdivided conceptually

into a sequence of octets of data followed by an End-of-TSDU indication. The T-DATA request primitive occurs when the End-of-TSDU indication is entered into the queue. The T-DATA indication primitive occurs when the End-of-TSDU indication is removed from the queue. This does not imply any particular subdivision in any real interface'.

The 'conceptual subdivision' of a T-DATA primitive implies, in LOTOS formal terms, the description of the execution of a T-DATA primitive as a finite process, where the 'occurrence' of the primitive corresponds to the successful termination of the process. Giving up the assumption of T-DATA atomicity means allowing the user to interrupt the execution of a T-DATA primitive in favour of the execution of an atomic one, i.e. for requesting the transfer (or accepting the delivery) of expedited data or even for disconnecting (thus disrupting the former T-DATA primitive execution).

This view of the IS model is formalized in the specification of [9] by introducing two distinct (atomic) primitives, i.e. T-DATA-ITEM and T-DATA-ETSDU. It is to be realized, however, that such descriptions correspond to a 'de facto' formally different definition of the service, insofar as the service interface consists of a different set of primitives. To be more precise, two distinct and, in some sense, hardly comparable TS definitions are given by the IS. The distance between the two may perhaps be better appreciated by trying to solve the following problem.

In the service definition with atomic T-DATA, a T-DATA indication may not take place before the corresponding T-DATA request at the other side (safety requirement). This has several 'counterpart' statements in the other service model, e. g. 'the beginning of a T-DATA indication (process) may not occur before the beginning of the corresponding T-DATA request', or also 'an End-of-TSDU indication may not occur before the corresponding End-of-TSDU request'; the same applies to any of the T-DATA-ITEM primitive in the middle. An important open question, however, is whether, in this model, the beginning of a T-DATA indication is or is not allowed to occur before the end of the corresponding T-DATA request. The IS states nothing on this concern.

A positive answer would seem to allow a by far larger class of devices to be able to cooperate for providing transport services: we guess that, ultimately, even a start-stop terminal might be arranged with little addition of machinery to implement the simplest class of the transport protocol.

A negative answer, on the other hand, seems to specify unbounded memory capacity (to enable saving whole TSDUs at either end), which can hardly be required of any 'real' implementation.

We leave the question open, but with the concluding remark that, at least in LOTOS, changing the set of primitives entails changing the process definition (due to the observational nature of the underlying model), and no observation equivalence can be proved to hold between two such definitions.

6. CONCLUSIONS

The purpose of this paper was twofold: we intended to introduce the specification language LOTOS, while at the same time providing a formal specification of the OSI Transport service.

Concerning LOTOS, we showed the expressiveness of this language and its adaptability to several quite different specification styles. LOTOS shows a very appealing flexibility in that 1) it allows both modular and global approaches to systems specifications and 2) it offers an orthogonal set of abstraction facilities, which encompass both process and data type abstraction.

Modular specifications are similar in structure to official OSI informal specifications, and therefore may be easier to understand for a reader familiar with the latter. Global specifications exhibit more clearly some global properties of the service, and therefore may be preferable for some uses, such as some types of correctness proofs. Since both styles of specification seem to have a role, the problem of being able to prove a specification in one style equivalent to a specification in another is practically quite important.

A similar problem exists relating to the use of Abstract Data Types. This formalism is in itself powerful enough to specify most aspects of a service specification, with the sole exception of temporal sequencing aspects. How much of a formal specification should be done in ADT, and how much in a CCS-like temporal ordering language, is a matter of specification style.

Beside being amenable to proof techniques, LOTOS specifications are 'constructive' and therefore executable as programs. This yields several advantages, among which their usefulness in testing environments [9].

As it was to be expected, the exercise of formulating the Transport Service specifications in LOTOS has led us to the discussion of two problems of interpretation of the IS. These problems were by no means unknown, but LOTOS has allowed us to discuss them in a precise way. The two, somehow related, problems are: the question of atomicity of T-DATA primitives, and the description of the flow control mechanism. In the first case, the IS seems to contain an ambiguity, due to an attempt to combine two levels of abstraction that are really not reconcilable. In the second case, on the contrary, the model given in the IS has been formalized quite easily, simply by specifying that the queue mechanism can, for its own internal reasons, suspend acceptance of any item to be enqueued.

ACKNOWLEDGEMENTS

The authors wish to thank all the participants to Subgroup C of the Ad-Hoc Group on FDT of ISO/TC97/SC21/WG1 for the fruitful discussions on the definition and application of LOTOS. They like to acknowledge also the contribution of Alberto Faro to an earlier version of this paper.

This work was partially supported by the European CEC Project ESPRIT/SEDOS (Software Environment for the Design of Distributed Open Systems) and by the National Science and Engineering Research Council of Canada.

REFERENCES

[1] ISO and CCITT, Basic Reference Model for Open Systems Interconnection, IS 7498 and CCITT X.200 (1984).

[2] Bochmann, G. v., A general transition model for protocols and communication services, IEEE Trans. Communic., 28 (1980) 643-650.

[3] Schindler, S., The distributed abstract machine. A model for service specifications in distributed systems, Comput. Communic., 3 (1980) 208-220.

[4] ISO and CCITT, Transport Service Definition, ISO 8072 and CCITT X.214 (1984).

[5] ISO, Information Processing Systems, Open Systems Interconnection, LOTOS - A Formal Description Technique Based on the Temporal Ordering of Observational Behavior, DP 8807 (March 1985).

[6] Brinksma, H., Karjoth, G., A specification of the OSI transport service

in LOTOS, in: Yemini, Y., Strom, R. and Yemini, S. (eds.), Protocol Specification, Testing, and Verification, IV, (North-Holland, Amsterdam, 1985).

[7] Tocher, A.J., OSI Transport Service: Specification in LOTOS of Data Transfer with Disconnect, ESPRIT/SEDOS/11 (November 1984).

[8] Billington, J., Specification of the Transport Service using Numerical Petri Nets, in: Sunshine, C. (ed.), Protocol Specification, Testing, and Verification, II, (North-Holland, Amsterdam, 1982).

[9] Logrippo, L., 'Constructive' and 'executable' specifications of protocols and services by using abstract data types and finite states transducers, in: Rudin, H. and West, C.H. (eds.), Protocol Specification, Testing, and Verification, III (North-Holland, Amsterdam, 1983).

[10] Milner, R., A Calculus of Communicating Systems, LNCS 92 (Springer-Verlag, Berlin, 1980).

[11] Ehrig, H., Fey, W., and Hansen, H., ACT ONE: an algebraic specification language with two levels of semantics, Ber. 83-03, Techn. Univ. Berlin (February 1983).

[12] Goguen, J.A., Thatcher, J.W., and Wagner, E., An Initial Algebra Approach to the Specification, Correctness, and Implementation of Abstract Data Types, in: Yeh, R. (ed.), Current Trends in Programming Methodology, (Prentice-Hall, Englewood Cliffs, 1978).

[13] Musser, D.R., Abstract Data Type specification in the AFFIRM system, IEEE Trans. Softw. Eng., 6 (1980) 24-32.

[14] Burstall, R.M. and Goguen, J.A., An informal introduction to specifications using Clear, in: Boyer, R. and Moore, J. (eds.), The correctness problem in computer science (Academic Press, New York, 1981).

[15] ISO and CCITT, Connection Oriented Transport Protocol Specification, ISO 8073 and CCITT X.224 (1984).

[16] ISO and CCITT, Connection-mode Session Protocol Specification, ISO 8327 and CCITT X.225 (1984).

[17] ISO and CCITT, OSI Service Conventions, ISO Revised DP 8509 and CCITT Draft Recommendation X.210 (1984).

SDL AND ESTELLE:

STATE TRANSITION ORIENTED SPECIFICATION TECHNIQUES IN CCITT AND ISO

Michael Steinacker

Corporate Research and Development
Siemens AG
Munich, FRG

Specification Techniques using a state transition model are developed with various claim to the formal definition of the technique. In the area of standardization CCITT (International Telegraph and Telephone Consultative Committee) and ISO (International Organization for Standardization) developed the Specification and Description Language (SDL) resp. the Formal Description Technique Estelle. The concept of both specification techniques are the same. To describe a system a tree structure of subunits can be used to decompose the system in manageable units. The unit of the lowest level are communicating processes which are specified by extended finite state machines. SDL provides two representation methods one being graphical supported and one using only a textual representation. Estelle provides only a textual representation. In this paper both specification techniques are characterized and their main capacities and attributes are discussed. A subsequent comparison also discusses the possibility of a translation from one specification technique into the other. Finally some obscurities are brought up.

1 Introduction

For the purpose of specification, software systems are usually decomposed into small units that are commonly referred to as modules or processes. Each process is described by its input-output behavior, for this and this alone characterizes a process. The description of a process therefore requires a transition-oriented (or state-oriented) method of specification that defines the possible sequences of inputs and outputs. State transition techniques, sometimes also called state-event techniques, which are frequently used for this application, serve to define explicit states and transitions. The transitions contain the inputs and outputs that represent essentially the primary information. Such description techniques are often misnamed "implementation description" techniques on the assumption that the states from the (abstract) specification must necessarily also exist in an implementation. Although this is one possible application of transition techniques, the states serve in reality solely as an aid to describe the next output for an input as a function of past history. Other description techniques referred to as transition-oriented description techniques in below, rely exclusively on inputs and outputs of the description without any explicit naming of states. It should, however, be noted that even with such description techniques states still exist since an input does not always result in the same output. A state must exist in addition to the input and they together characterize an output.

For describing the sequence of transitions, two alternative trends are distinguishable in specification practice. Either the transitions that can directly follow a state or transition are described, or, more generally, i.e. more problem-oriented, transitions are described which can or must at some time follow, or must not be allowed to follow, a state or transition. Initial approaches to the latter alternative, in which

of course directly consecutive transitions or states do not occupy the foreground of interst, use, for example, temporal logic /15,16,17,18/.

For the first alternative there are a few approaches to transition-oriented description techniques, i.e. techniques in which the description is mainly confined to the inputs and outputs of a process /3,4,5,6,7/.

Various known and very similar description techniques such as state-event diagrams /8,9/, state-event tables, Specification and description Language (SDL) /2/, and Estelle (A Formal Description Technique Based on Extended State Transition Model) /1/ are based on the concept of the state oriented description technique. Petri networks /e.g. 10,11/, in which the terms input und output are not generally used, are closely related to this concept.

Formal specification techniques are important tools for the design, analysis, and implementation of information processing systems. It is by means of formal techniques that system specification can be produced that are complete, consistent, precise, consice, and unambiguous. This is only possible if a formal specification technique is self contained, so that the description given in a formal specification technqiue need not refer to any informal knowledge of the system that is described. An important aspect of a formal specification technique is that it allows analysis by mathematical methods. A formal specification technique has such a formal, mathematical basis and can be used to simulate a specification or prove the correctness of a specification. It also can be a basis for the implementation and for conformance testing. A formal specification technique usually does not replace an implementation language.

In the area of standardization of the Reference Model for Open Systems Interconnection (OSI) two specification techniques are discussed. In CCITT the definition of the Specification and Description Language (SDL) is reorganized and extended in a new recommendation /2/ and in ISO the Formal Description Technique Estelle /1/ is developed. The formal specification technique LOTOS /3/ which is also developed within ISO pursues another approach and is not considered in the following. In ISO both techniques have the stage of a Draft Proposal (DP). All three specification techniques can be used to specify distributed systems and are not restricted for the use in the OSI context.

2 Concept of Estelle and SDL

Both specification techniques SDL and Estelle use the same model - an automaton - as a semantic basis. They differ somewhat in regard to the higher language elements derived but their essential differences resides in the representation method (syntax).

Two forms of representation have been defined for SDL, one having graphic support (SDL/GR, SDL Graphical Representation) and the other being a purely textual form of representation (SDL/PR, SDL Phrase Representation). For Estelle only a textual form of representation is defined. The syntax of Estelle is adapted closely to the syntax of PASCAL so that the software does not have to be described completely new for syntax checking or for the simulation of a specification. In Estelle all the declaration part in a module and the description of a state transition is essentially appropriated from PASCAL.

Although, for example, with PASCAL, each statement is an individual step from one PASCAL state to the next, with Estelle the PASCAL elements within a transition describe constructively or algorithmically only the final (i.e. next) state and the output, if any, without intermediate states appearing in the transition; i.e. the characteristics of a specification technique will not be lost if a programming language is used to formulate the state transitions.

The specification techniques provide various kinds of structuring. At first they have a concept for modularisation of the specification of a system to structure the whole system in separate communicating units which can interchange information via communication pathes (channels). These units should be easy understandable and manageable. Although both techniques have different kinds of structures they use the same idea. The unit of the most interest is the unit called in the following Process (with a capital letter). The idea of a Process is to specify a part of the system which itself will run sequentially but which will run in parallel regarding to the other Processes. Furthermore a Process defines the dynamic behavior of the system. Essentially Processes interchange information via inputs and outputs. These informations are called interactions in Estelle and signals in SDL. This communication of the Processes is based on a rendezvous technique which is a basic communication technique /3,12,. 13,14/. Nevertheless the rendezvous communication is not available at the user level in Estelle and SDL. The interchange of interactions resp. signals between Processes will always use a queue. This will be discussed later.

A Process itself is described by the possible sequences of the inputs and outputs of the Process from resp. the other Processes. In both techniques these sequences are described with use of states and transitions of an automaton (extended finite state machine).

Thus a Process is characterized by states, inputs (events), outputs and state transitions. A transition always has an initial state and a final state; it may further have an input and an output. The final state and the output, if any, depend solely on the initial state and the input, if any.

The states, inputs and outputs describe three classes of information:
- information inherent in the Process is called the state;
- information put into the Process is called the input;
- information put out by the Process is called the output.

It is said that a Process is only recognizable in a state. State transitions show the possivle sequences of states, inputs and outputs, thereby describing the dynamic response of the Process.

Another principle of structuring is used by the description of a Process. Both the states and the inputs and outputs are structured as known from the area of programming languages: a state is structured by means of (so called) variables whereas the interactions are structured by means of (so called) parameters. The description of transitions are also structured, e.g. by means of procedure. In addition the concept of types and instances are used for Processes, channels etc., too.

The various means of structuring enable the user of the specification techniques to describe their problem (system) in an appropriate manner and to produce clear and understandable specifications. To support this aim each Process will include implicitely (one or) more input queues as (an) additional extended finite state machines, as mentioned above. An input queue decouples the transition with the output and the corresponding input and gives the possibility of other strategies than FIFO.

A further aspect is the use of formal or informal text (or symbols). A formal text is represented using a well defined syntax and semantic. An informal text may have any form suitable to convey information. In this case the writer assumes some common basis of understanding between himself and the reader. In this sense informal text must be sufficient formal in the reader's understanding in order to be unequivocally understood.

It should be mentioned that there is a very clear distinction between informal text and comments. Comments are introduced to help readers in understanding the representation, their absence or presence does not alter the representation. The informal text is part of the representation itself, so that it is not understood without considering the associated text.

3 SDL

For the definition of the state structure of a system SDL has three levels of abstraction for defining the modularity. The upper level is the system. A system is composed of a number of blocks connected together by (SDL-) channels which are used to show the communication path between blocks. Each block is independent from each other block. A block is composed of a number of (independent) communicating Processes, called (SDL-) processes.

A process is defined by an extended finite state machine and communicates with other processes by sending and receiving (so called) signals. Furthermore processes within the same block can communicate with another by shared variables. A process also can create dynamically a new process within the same block. Thus blocks provide a convenient mechnaism for grouping processes and in addition a block forms a boundary for the visibility of data. The block definition contains the list of processes belonging to the block.

Channels are the (only) means of communication between blocks and between the system and its environment (more precisely between blocks and the environment of the system). A channel has two ends and can carry signals in one direction. The mechanism of carrying a signal through a channel from one process of one block to another process of another block is the same as the mechanism of carrying a signal from one process to another within a block. This mechanism includes an "unspecified" delay. Channel definitions contain a signal list that lists all the signals that can be carried by the channel. That signal list is provided as a means of ensuring that every signal sent by a process at one end of the channel can be received by the process in the block of the other end of the channel. (Fig. 1a)

Conceptually blocks and channels on the one hand and processes and the in SDL unnamed transfer of signals within a block on the other hand are similar methods of structuring and represent only a different level of abstraction. But in addition blocks and channels can be further subdivided into (sub)channels and (sub)blocks and processes can be subdivided into (sub)processes. In this case the definition of a channel resp. a block consists of a list of (sub)blocks and (sub)channels connecting the (sub)block. Thus SDL provides a multi-level (recursive) method for grouping processes and (sub)blocks of a system.

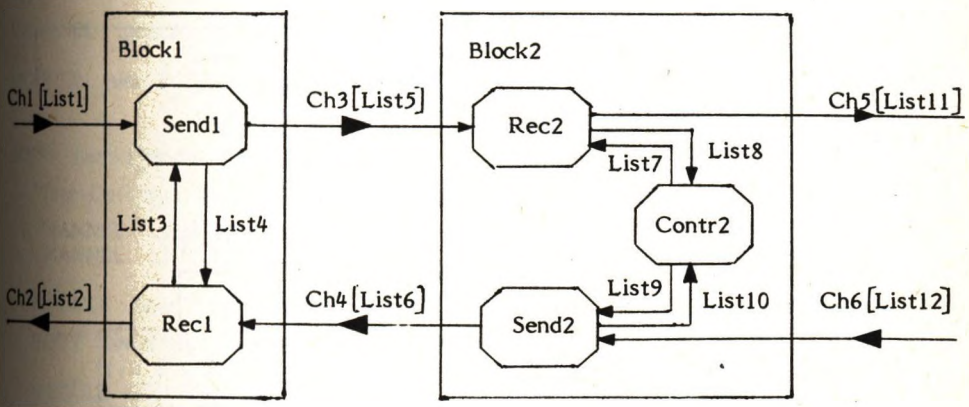
As mentioned above a process is defined by an extended finite state machine and defines the dynamic behaviour of a system. Furthermore a process includes one input queue. The states, input signals and output signals are characterized by special graphical symbols (SDL/GR) or by special keywords (SDL/PR). The same is valid for special parts of the transition, e.g. the decisions, descriptions of the variables of the new state (task), comments, procedures, and connectors. In both representations in SDL/GR and in SDL/PR the declaration parts for the states (and variables) and the signals (and parameters) and the description of the states, signals and transition parts are given in a textual representation. For these textual representation formal or informal text can be used. A formal text is represented using the SDL/PR syntax. An informal text associated with SDL/GR symbols or SDL/PR statements may only be informally interpreted. (Fig. 1b)

For sturcturing the representation of transition SDL introduced additional elements (SDL, Z.103), the procedure and the makro. If a part of a transition or many subsequent transitions occur at different places of a specification in a similar style they can be replaced by a procedure or a makro. A procedure definition contains formal parameters and can therefore be used for recursion and for the replacement of transitions with different actual names for states, variables etc. A makro is a simplified form of a procedure. It allows only a direct replacement of the text.

Data in SDL are defined as an "abstract data type" (SDL, Z.104). "Data" does not only consits of the set of possible values but also of the set of possible operations on the values and the effect of this operations.

"DATA DEFINITIONS"

```
SIGNAL  Dat(Vc_Type,CHARACTERSTRING,Seq_Type,Seq_Type),
        Rr(Vc_Type,Seq_Type),
        Rnr(Vc_Type,Seq_Type),
        Dis(Vc_Type),
        ....
```



List1 = [.... ....]

....

List6 = [Dat Rr Rnr Dis ....]

....

List12 = [.... ....]

Fig 1a: Example of a block structure (SDL/GR)

To use the input queue of a process with another strategy than FIFO there is the language element SAVE. A SAVE lists input signals and is attached to a state symbol. The input signals used for the next transition are than the first input signal in the input queue not listed in SAVE.

As a short hand notation for often used compley signal exchange SDL provides composite operations, which also introduce some slight extensions. These composite operations are the import and export of data values, the enabling condition, and the continuous signals.

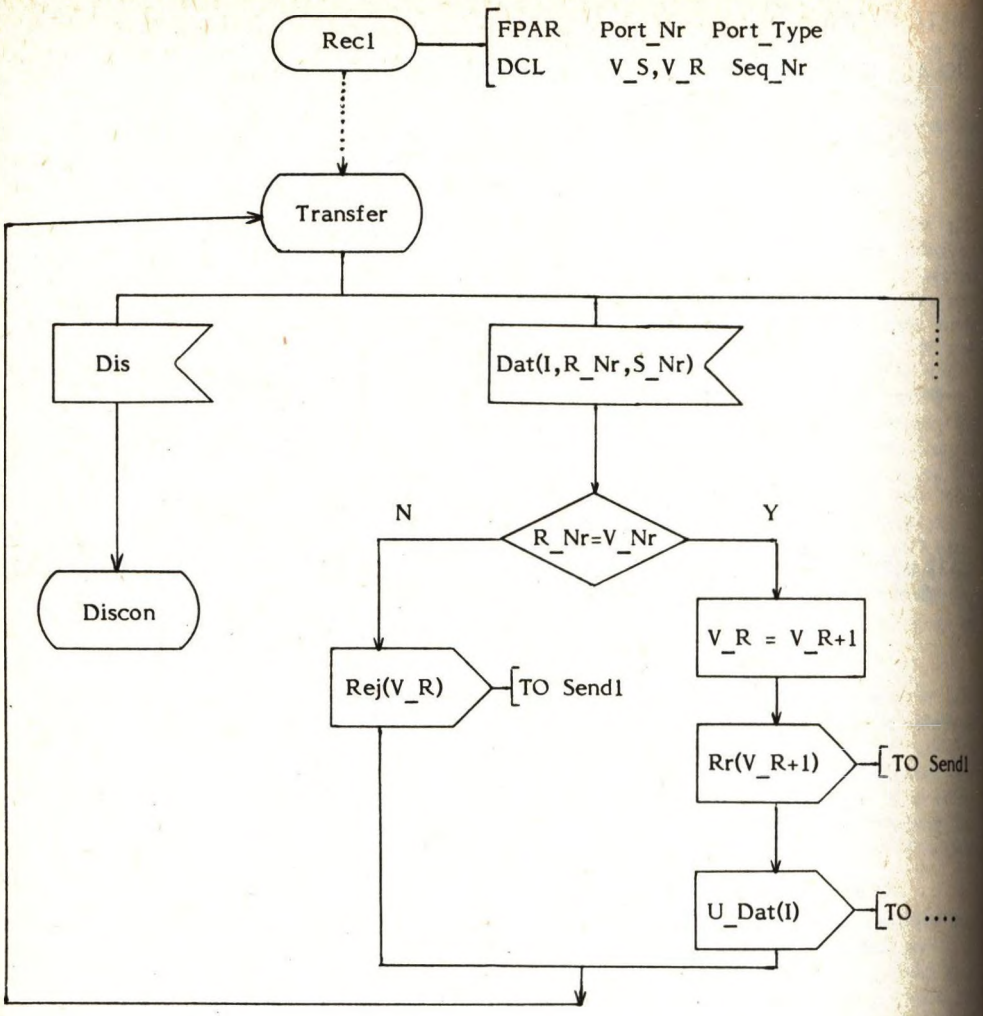


Fig 1b: Example of a part of a process (SDL/GR)

The import and export of data values is modelled by a signal interchange containing additional states and transitions between the processes reffered to. The enablings conditions are a shorthand notation for the capacity either to accept a signal as an input or to save the signal depending on the condition, which may contain imported values. This is modelled as having several states in which the signal is accepted as input and another state in which the signal is saved. The implied state is chosen after evaluating of the condition. The operation may also imply signal interchange. A continuous signal is a shorthand notation for leaving a state and performing a transition when a condition, which may use (local) variables and imported values, becomes true. A continuous signal has lower priority than "normal" signals, and may used to stimulate a transition when external data values change. The operation implies a number of state and signal interworking. The operation is represented by an enabling condition and a priority clause to prevent a non-deterministic decision. Using only local variables in the enabling condition the continuous signal defines transition without input signal.

```
"Type_Definition_Part"

CHANNEL  List1_Id ....
.....
CHANNEL  List6_Id(Sender,Receiver)
  BY Sender: Dat(Vc: Vc_Type; I: ARRAY Num OF CHAR; R_Nr,S_Nr: Seq_Type);
             Rr(Vc: Vc_Type; S_Nr: Seq_Type);
             Rnr(Vc: Vc_Type; S_Nr: Seq_Type);
             Dis(Vc: Vc_Type);
             ....
....
CHANNEL  List12_Id ....

MODULE  Block1_Type PROCESS (List1 .... List2 .... List5 ....;
                             List6: List6_Id(Receiver) INDIVIDUAL QUEUE;
                             PARAM Init_Port_Nr: Port_Type);
BODY  Block1_Body  FOR  Block1_Type;
  "Type-Definition-Part"
  CHANNEL  List3_Id ....
  CHANNEL  List4_Id ....

  MODULE  Send1_Type  PROCESS (List1 .... List3 .... List4 .... List5 ....);
  BODY  Send1_Body  FOR  Send1_Type; ....

  MODULE Rec1_Type PROCESS (List2 .... List3 .... List4 .... List6 ....;
                            PARAM Port_Nr: Port_Type);
  BODY  Rec1_Body  FOR  Rec1_Type;
        ... (see Fig. 2b) ...
  VAR  Send1: Send1_Type;
       Rec1:  Rec1_Type;

  INITIALIZE BEGIN INIT Send1 WITH Send1_Body (Init_Port_Nr);
                   INIT Rec1  WITH Rec1_Body (Init_Port_Nr);
                   ATTACH List6 TO Rec1.List6;
                   ....
             END

MODULE  Block2_Type  PROCESS ....
BODY  Block2  FOR  Block2_Type ....

MODULE  Ch1_Type  PROCESS ....
BODY  Ch1_Body  FOR  Ch1_Type;
EXTERNAL;
....
MODULE  Ch4_Type  PROCESS (List6a: List6_Id(Sender);
                           List6b: List6_Id(Receiver));
BODY  Ch1_Body  FOR  Ch4_Type;
EXTERNAL;
....
MODULE  Ch6_Type  PROCESS ....
BODY  Ch6_Body  FOR  Ch6_Type;
EXTERNAL;
```

Fig 2a: Example of a module defined by a substructure (Estelle)

4 Estelle

In Estelle there is the concept of the (so called) modules. A module can be defined by a substructure of modules and thier communication pathes, by an extended finite state machine, or by a combination of both. There is no difference between the static and dynamic structuring of a module. Furthermore the system as well as a Process is a module. This structuring principle leads to a tree structure as in SDL, where the root is the system and each leaf is an extended finite state machine. A process without substructure corresponds to a Process.

A module also has the property called class: a module is a (Estelle-) process or an activity. All direct submodules (called the siblings) of a module (called the parent) have to be of the same class. If the siblings are processes they can run in parallel, and can be substructured. If the siblings are activities they will never run in parallel and they also can have shared variables. (At present a parent and its children cannot run in parallel, too.) Therefore a subdivision of a process into activities is a kind of structuring a process. Regarding to the tree structure above the father of the activities is the leaf of the tree. The activities should not be regarded as a part of the modularity of the system rather than a possibility to structure a Process.

```
MODULE Rec1_Type PROCESS (List2 .... List3 .... List4 ....;
                          List6: List6_Id(Receiver) INDIVIDUAL QUEUE;
                          PARAM Port_Nr.: Port_Type);

BODY  Rec1_Body  FOR  Rec1_Type;
   ....
   VAR  V_S,V_R: Seg_Type;
   ....
   TRANS
      FROM  Transfer  TO  SAME
         WHEN List6.Dat
            PROVIDED R_Nr ≠ V_R + 1
               BEGIN  OUTPUT List3.Rej(V_R);  END
            PROVIDED R_Nr = V_R + 1
               BEGIN  V_R:= V_R + 1;
                      OUTPUT List3.Rr(V_R + 1, S_Nr);
                      OUTPUT List2.U_Dat(I);
               END

      FROM  Transfer  TO  Discon
         WHEN List6.Dis
               BEGIN  END
   ....
```

Fig 2b: Example of a part of a module without substructure (Estelle)

A module communicates with other modules by sending and receiving (so called) interactions. Interactions are grouped by means of interaction points. The interaction points do not introduce a new property they are only a means of structuring. The communication pathes between modules are defined by (Estelle-) channels. A channel definition consist of the definition of two "rules" and a list of interactions to each rule. A channel identifies or binds the two interaction points having the same list of interactions respecitvely. In contrast to a SDL-channel a Estelle-channel does not have any property as "delay", "transition error", etc. (Fig. 2a)

The creation and life time of a module and the binding of its interaction points is controlled by the father. A father can bind the interaction points of two sons, an interaction point of a son to an own (internal) interaction point which he uses itself, or an interaction point of a son to an own (external) interaction point which is used of a module of the environment of the father. This also implies that for all bindings of a (possible dynamic created) son to a module of the environment of the father the father need an interaction point.

As mentioned above a Process, i.e. a process without submodules, is defined by an extended finite state machine. The representation has some similarities to SDL/PR. The states, input interactions, output interactions, transitions etc. are characterized by keywords. (Fig. 2b) In contrast to SDL informal text can be used only very restrictively in a body of a so called primitive function to define implementation dependent parts. This functions must not have a side effect. In Estelle there do not exist language elements like SAVE in SDL and at present there do not exist abstract data types as defined in SDL-Z.104. On the other hand there are some extensions of an extended finite state machine compared with SDL. A process may have more than one input queue if there are more than one interaction point. Each interaction point can be assigned to a common queue or can use an own queue. Furthermore, a transition may not have an input interaction. These both features and a special statement introduce the non-deterministic choice of a transition. This is also possible by specifiying in a state two transitions with the same input interaction. The non-deterministic choice can be restricted or prevented by use of the priority of a transition, another additional feature.

To structure the representation of transitions starting from the same major state or triggered by the interactions having the same name Estelle allows an enabling condition which is different from the one used in SDL. In this enabling condition the parameter of the input interaction and the variables of the current state can be used to restrict a transition. Furthermore some additional language elements and many language elements taken from PASCAL increase the expressive power and flexibility and can also be used to structure the transition (e.g. with, all, any). (Fig. 3)

It should be stressed once more that in contrast to SDL the definition of a process, i.e. a module of the class process, can consist of a definition of both an extended finite state machine and a substructure of modules (Fig. 4).

It should also be stressed that the properties of a Process are also valid according to that part of a module of the class process, which is defined by an extended finite state machine itself.

5 Comparison of Estelle and SDL

As mentioned in chapter 2 both techniques use the concept of communicating Processes whereas a Process is described by an extended finite state machine. The information interchanged between Processes are called interactions in Estelle resp. signals in SDL. The differences in the style of the syntax are also explained in chapter 2.

```
MODULE Send_Type PROCESS  (List3: ARRAY Vc_Type  OF List3_Id (Receiver);
                           List1 .... List4 ....;
                           List5: List5_Id(Receiver) INDIVIDUAL QUEUE;
                           PARAM Port_Nr: Port_Type);
BODY  Send_Body  FOR  Rec_Type
  ....
  VAR  Con: ARRAY Vc_Type  OF RECORD State: (...., Transfer,....);
                                          Send_Ok,Conf_Ok: BOOLEAN;
                                          V_Conf,V_S,V_R: Seq_Type END;
  ....
  TRANS
    WHEN  List3 Vc .Rr
      WITH  Con Vc  DO
        FROM  Transfer  TO  SAME
          PROVIDED  S_Nr = V_S AND Send_Ok
            BEGIN  V_Conf := S_Nr;
                   V_R    := R_Nr;
                   V_S    := V_S + 1;
                   OUTPUT List5.Dat(Vc,I,V_S,V_R);
            END
          PROVIDED  S_Nr = V_S AND Conf_Ok
            BEGIN  V_Conf := S_Nr;
                   V_R    := R_Nr;
                   V_S    := V_S + 1;
                   OUTPUT List5.Rr(Vc, V_R);
            END
      END
    ALL  Vc : Vc_Type  DO
      WITH  Con Vc  DO
        PROVIDED  Conf_Ok
          FROM  Transfer  TO  SAME
            BEGIN  OUTPUT List5.Rr(Vc,V_R);  END
      END
    ....
```

Fig 3: Some additional features of Estelle

```
(* Module Header *)

MODULE  H_Type_Id  "class"   ("Interaction_point_decl"
                              "Parameter_List");
                              "Exported_Vars"


(* Module Body *)

BODY  B_Id  FOR  H_Type_Id;

  "Declaration Part"
     (* Const_Declaration_Part
        Type_Declaration_Part
        Channel_Type_Definition
        Module_Type_Definition
        Variable_Declaration_Part
        State_Set_Definition_Part
        Use_Clause
        Procedure_And_Function_Definition_Part *)

  "Initialization_Part"

  "Transition_Declaration_Part"

  "Termination_Part"
```

Fig 4: Structure of a module type definition in Estelle

To support the modularity of a specification both techniques provides a tree structure of specification units - system and modules in Estelle resp. system, blocks and processes in SDL. In Estelle a Process corresponds to a module of the class process without a substructure whereas in SDL a Process corresponds to a (SDL-)process. Although both methods have very much similarities they have a different kind of construction. Furthermore the representation of the relationship between output and input is different. Estelle uses the interaction point and (Estelle-)channel which only bind the interaction points. In SDL the destination process is described togehter with the output signal in the transition. Additionally conveying a signal from one SDL-process to another seems to consume "time". Thus an automatic translation from one technique into the other will be very difficult or impossible, if the special features of one of these techniques are extensively used. Therefore in the following some restrictions of Estelle and SDL are discussed.

The translation of a static structure of SDL into Estelle can easily be done. A block in SDL can be mapped on a module in Estelle, whereas this module is of the class process and has to be defined only by a substructure. A SDL-process can be mapped on a module in Estelle whereas the module is of the class process and has to be defined only by an extended finite state machine. The "delay" of the conveyance of signals between SDL-processes can be modelled in Estelle by (unspecified) additional modules between the "true" modules. (Fig. 1a and 2a) The translation of the dynamically created processes is more difficult. In SDL the new process is in the same hierarchical level whereas in Estelle the new module is a son of the creator and therefore in the next lower level. Thus the creation of a new module has to be done by the "father" of the regarded module.

SDL(GR)	Estelle
block	module defined only by a substructure
channel	additional module
signal	interaction
process	module defined only by an extended finite state machine
input symbol	when
state symbol, next state symbol	from, to, next state
ouptput symbol	output
decision symbol	if statement, case statement
task symbol	PASCAL statements

Fig. 5: Translation of concepts and terms

The translation of the structure of an Estelle specification into SDL cannot be done if in a module of Estelle the combination of the definition by an extended finite state machine and by substructuring is used. This style of substructuring cannot be modelled by SDL. The following restriction in Estelle will prevent this problem: all modules of the same father have to use the same kind of definition, either by an extended finite state machine or by a substructure. Note, the (supposed) "delay" of the conveyance of signals between SDL-processes is a new property of the translated SDL specification and there is no corresponding property in the original Estelle specification.

In contrast to a SDL specification in an Estelle specification is the restriction that each transition of a module (e.g. each creation of a new module) forces all sons of that module to wait. This difference holds for both both translations.

For specifying a Process both techniques use an extended finite state machine with states, inputs and outputs (interactions resp. signals), and transitions which can all be structured (see also Figure 5). Important differences are the different additional language elements some of them introducing an extension of the basic concept of an extended finite state machine and a contrary definition in the absence of an input specification. By the latter one SDL provides a "null" transition to the same state consuming the input whereas Estelle does not consider this input if there are more than one input queues and defines this as a specification error if there is only one input queue. Note, in Estelle the presence of more than one input queue prohibits the solution of SDL.

In SDL the procedures, makros, and the composite operations are defined to support a clear understandable representation. One composite operation, the continuous signal also introduces transition without input signal. However, this transition may only occur in absence of an input signal in the input queue. A non-deterministic choice as in Estelle is also forbidden by the rule to use the priority clause. The shared variables, SAVE, and abstract data types are extensions of SDL and are not defined in Estelle (at present).

In Estelle there are some additional language elements (e.g. enabling conditions, any, all, forone) and many language elements taken from PASCAL (e.g. procedures) to increase the expressive power and flexibility to the description of transitions. To prevent misleading specifications in contrast to SDL procedures are allowed only within a transition and must not contain states.

The enabling condition of Estelle is more powerfull than the enabling condition of SDL (exept the use of imported variables not defined in Estelle). At first the conditions in relation to one interaction name may not exclude another. These conditions can overlap to specify a non-deterministic choice between different transitions. Secondly there is not a difference between an input interaction with an explicitely specified name but with a condition not explicitely specified and an input interaction with not explicitely specified name: in Estelle both interactions are not considered as an input for this state whereas in SDL the first one is saved and the second one is consumed with a "null" transition.

Thirdly in Estelle the parameters of the input interaction can be used in the enabling condition. This is motivated by the meaning of input names and parameters. From the view of the semantic of the extended finite state machine there is no difference between a name of an interaction and a parameter (resp. between a state and a variable). An interaction is exactly identified only if the name and a value for each parameter is defined. The destinction is only for the human reader to structure the information. Thus the restriction of SDL not to use an enabling conditions as a "normal" part of the selection of the representation of transitions but only to use the state and the interaction name can be by-passed using a decision symbol after the input symbol or using more states and names for the interaction. In the first case the readability of the specification may decrease. In the second case this new structure of states and interaction names must consider the requirement of all combinations of states and input interaction of the process. This frequently will lead to a large number of states and interaction names and to a small number of variables and parameters. This increase of the number of states and interactions will increase the lenght of the specification and decrease the readability of the specification so that the advantages of the introduction of variables and parameters (regarding to the old fashioned state machines) take not really effect.

As extensions of the concept of the (basic) extended finite state machine and in contrast to SDL Estelle knows more than one input queue in a Process, a non-deterministic choice of transitions, and an unrestricted use of transitions without input interactions.

Summarizing the following features are difficult to translate

- creation of a new module in Estelle resp. process in SDL
- enabling conditions in Estelle
- implicite transitions in SDL

and the following features can not (efficiently) be translated

o from Estelle to SDL

- Estelle channels
- modules defined by a substructure and an extended finite state machine
- father and son do not run in parallel
- more than one input queue
- non-deterministic choice of transition
- transitions without inputs
- various language elements for describing the behaviour of a module

o from SDL to Estelle

- graphic support of the representation
- shared data
- abstract data types
- SAVE

Finally it should be mentioned that SDL is recommended to be used with informal text as well as with formal text. Estelle is intended to be used mainly as a formal description technique. The power of the formal text of Estelle seems to be much better than the power of the formal text of SDL.

6 Remarks on Estelle and SDL

In SDL it is recommended to use only three level for a specification, the system which is structured in blocks, the blocks which are structured in processes, and the processes itself. The use of subblocks and subchannels is recommended only for the so-called refinement of the representation of a system in different levels of abstraction.

It should be stressed that semantically the most important part of a specification in SDL are the processes. The (sub)blocks and (sub)channels are "only" an additional information to improve the readability but do not have a real effect on the system. A system consisting of all these processes (without blocks and channels) would have the same behaviour. That means a system specification containing blocks without processes is incomplete. It is misleading to call such a specification a "specification of a certain level". Note that this also is valid for an Estelle specification containing modules at the lowest hierarchical level which are not exclusively defined by an estended finite state machine.

It seems to be very useful to structure a specification in SDL with the aid of subblocks and subchannels in a way appropriate for the probelm considered and without the restriction of three levels. Also the number of levels of different parts may be distinct.

In this context the difference between a channel as a communication path between blocks and the unnamed communication path between processes in the same block seems to be wise: a channel can be partitioned. On the other hand the partitioning of a channel in blocks and channels involves that each channel must have a "delay" as supposed before. But this perception implies that a specification in SDL will never be a formal specification because all channels are unspecified. Additionally it is not clear whether or not the communication between processes of the same block do have the delay as a channel.

Another problem is the semantic of a subprocess. Processes are defined by a sequential behaviour and two processes are independent and concurrent. A process can be partitioned in subprocesses. Different subprocesses of a process can be in different blocks. Thus are subprocesses independent and concurrent? If so, it seems that a process can consist of independent or concurrent parts and does not have a sequential input/output behaviour. What is the solution?

In this context the semantic of shared variables is not clearly defined. In the known definitions of communicating state machines the state machines does not have common states. The communication only works via signals. If the state machines do have common states, what is the independence of concurrency of these state machines?

Last but not least it should be mentioned that the coherence between the abstract data type and the basic definition of SDL is not totally clear.

As mentioned above in Estelle the semantic of the behaviour of a father of modules is very unusual and it seems that this semantic can not be found in the reality. Each transition of a father, e.g. a creation of a new son, forces all sons and their descendant to stop and wait until all descendants have stopped and the father has performed its transition. This also implies that it is not very useful to specify a module by a large extended finite state machine when it also is defined by a substructure.

Another remarkable point in Estelle is the following: On the one hand a module once created has a fixed number of interaction points. On the other hand a module may dynamically create new modules (sons). If these sons want to communicate to other modules outside the father they need an interaction point of the father for each connection. Thus there is a static property of the father to solve a dynamic requirement.

In both techniques the support of the specification of time variations is very poor until now. This includes the language elements defined until now as well as the completeness of the definition of these language elements. Furthermore the description of multible connections between more than two Processes are not supported.

7 Conclusion

The specification techniques Estelle and SDL are very similar. SDL provides a graphic oriented and a textual representation whereas Estelle provides only a textual representation in PASCAL style. Both techniques are based on the idea of partitioning a system in a hierarchical style into smaller specification units. The lowest level is defined by communicating concurrent processes described by extended finite state machines.

Although both techniques differ in the way of constructing a system they can be compared if the methods are used in a restricted way. Nevertheless some differences in the cooperation of the processes based on unusual definitions of the techniques remain.

The processes are described by the possible sequence of inputs and outputs. These sequences are described by means of states and transitions of the extended finite state machine. Also for the description of the structure of the states, inputs and outputs and transitions both techniques use the same ideas essentially.

In addition SDL provides mainly some further operations for the communication with other processes. Some of these operations are extensions and some operations are derived from other operations. Furthermore abstract data types are defined. The use of shared variables and the use of abstract data types seems not to be completely defined.

Estelle mainly provides some extensions of the concept of the extended finite state machine and many powerful language elements, partly from PASCAL, to describe the transitions. However, abstract data types are not yet added to the language. Nevertheless, Estelle seems to be a more powerful specification technique than SDL when a formal description is required.

Altogether SDL and Estelle are specification techniques with many features and both specification techniques can complement one another. Further on it seems to be possible and efficient to bring both specification techniques together particulary since Estelle and SDL are used also in the same area of application, the communication protocols and services.

References

/1/ Estelle - A Formal Description Technique Based on an Extended State Transition Model, ISO/TC97/SC21(DP 8806), Febr. 1985

/2/ Functional Specification and Description Language (SDL), CCITT, Red Book, Fascicle VI.11, Recommendations Z.100 to Z.104, 1985

/3/ LOTOS - A Formal Description Technique, ISO/TC97/SC21(DP 8807), Feb.1985

/4/ Milner, R.: A Calculus of Communicating System, Lecture Notes on Computer Science 92, Springer Verlag, 1980

/5/ Milner, R.: Calculi for Synchrony and Asynchrony, Theoretical Computer Science 25, 1983, 267 - 310

/6/ Lauer, P.E., Shields, M.W.: COSY. An Environment for Development and Analysis of Concurrent and Distributed Systems, Software Engineering Environments, 1981, 119 - 156

/7/ Schindler, S., Steinacker, M.: A Formal Specification of an X.25 Protocol Machine, Trends and Applications 1979: Advantages in Systems Technology, May 1979

/8/ Bochmann, G. v.: Finite State Description of Communicating Protocols, Computer Networks, Vol 2, Oct. 1978, 361 - 372

/9/ Keller, R.M.; Formal Verification of Parallel Programs, Communication of ACM, 7, 1976, 371 - 384

/10/ Diaz, M.: Modelling and Analysis of Communicating and Cooperation Protocols Using Petri Net Based Models, Computer Networks 6, 1982, 419 - 441

/11/ Eckert, H., Prinoth, R.: A Computation-Systems Based Method for Automated Proving of Protocols Against Services, Protocol Specification, Testing, and Verification III, H. Rudin and C.H. West (eds), North Holland, 1983, 213-228

/12/ U.S. Dept. of Defense, Reference Manual for the ADA Progamming Language, Jan. 1983

/13/ Budkowski, S., Najm, E.: Structured Finite State Automaton, Protocol Specification, Testing, and Verification III, H. Rudin and C.H. West (eds), North Holland, 1983, 95 - 110

/14/ Steinacker, M.: A Common Semantic Model for State Transition-Oriented Specification Techniques, Siemens Research and Development Reports, Bd. 13, 1984, Nr. 6, 260 - 265

/15/ Pnueli, A.: The Temporal Semantics of Concurrent Programs, Theoretical Computer Science 13, 1981, 45 - 60

/16/ Hailpern, B.: Verifying Concurrent Processes Using Temporal Logic, Techn. Report 195, Computer System Laboratory, Stanfort University, August 1980

/17/ Schwartz, R.L., Melliar-Smith, P.M.: Temporal Logic Specification of Distributed Systems, Proc. of the IEEE Conf. on Distributed Systems, April 1981

/18/ Schwartz, R.L., Melliar-Smith, P.M., Vogt, F.H.: Interval Logic: A Higher-Level Temporal Logic for Protocol Specification, Protocol Specification, Testing, and Verification III, H. Rudin and C.H. West (eds), North Holland, 1983, 3 - 18

PROTOCOL VERIFICATION SYSTEM BASED ON MODULA-2 LANGUAGE

Laszlo Kovacs
Computer Network Department
Computer and Automation Inst.
Budapest,Victor Hugo u 18-22
1132 HUNGARY

Jozsef Harangozo
Department of Process Control
Technical University
Budapest,Müegyetem rkp.9.
1111 HUNGARY

ABSTRACT

The paper presents an automated protocol verification system based on the improved reachability analysis. The protocol is described by an ESTELLE-like specification language. The protocol specification is transformed into MODULA-2 modules. On the basis of these modules a transition system model is generated by the verification system. By the help of this model the verification system checks the dynamic properties of the protocol. The implementation structure and the usage of the verification system is described.

1. INTRODUCTION

Several specification and verification methods and systems have so for elaborated for computer network protocols. Some of them are well-known and widely used even if there is no one to meet all the requirements and able to solve all the problems of protocol specification and verification [Bochmann 82],[Harangozo 83].

A working group of the ISO/TC97/SC16 developed a hybrid specification language named ESTELLE that provides tools for specifying a layer of the network architecture together with its entities, services and protocols [ESTELLE 84]. A protocol is modelled with an extended finite state automaton. The entities, the services and the protocol model are described by a Pascal-like language. Our method is based also on the same idea, it is also a hybrid specification approach, but our description tool is built on the MODULA-2 programming language [Wirth 80]. This language was chosen, because it seemed to be more advantageous than the Pascal for describing, specifying and verifying a system consisting of modules of parallel processes.

The next sections present our verification system based on an improved reachability analysis. We deal with the structure and the usage of the system, with the representation of the reachability graph. The abstract data type library used and the MODULA-2 representation of the protocol specification are also presented. Finally the operation of the verification system and some experiences of its operation and usage are given.

2. IMPROVED REACHABILITY ANALYSIS

The improved reachability analysis verification method checks the invariance of the predicates of the protocol properties in which the human designer is interested. It is based on the reconstruction of /Q,R,N,P/ transition system from the /K,L/ protocol language model. The /Q,R,N,P/ system is a modified version of the Keller model of parallel computation [Keller 76, Kovacs 84]. The /Q,R,N,P/ partitional transition system is a quadruple where Q is a set of states, R is a binary relation on Q, N is a set of actions and P is a partition of actions. The most important general and individually formulated specific protocol properties in the form of predicates /deadlock freeness, liveness, functional correctness/ could be defined in this system and could be verified by this verification method. The high-level language implementation of the /K,L/ model is an ESTELLE-like protocol specification [Kovacs 83]. The reconstruction is equivalent to the definition of the semantics of the ESTELLE-like language in the /Q,R,N,P/ transition system.The essence of the verification method consists of building a G graph of /Q,R,N,P/ system of the ESTELLE-like language representation. This transformation is done by the verification system.

The G graph of /Q,R,N,P/ system is an oriented graph whose nodes correspond to the Q states and whose arrows correspond to the state transitions labelled by actions. In other words this means that there is an arrow from the node indicated by q to the node q' labelled by p.n if and only if there exists a $q \xrightarrow{p.n} q'$ relation in the /Q,R,N,P/ system /$q,q' \in Q, p \in P, n \in N$/. The correspondence between the structural /geometrical/ properties of the G graph and the protocol properties defined in [Kovacs 84] makes it possible to map the protocol vertification to the structural analysis of the G graph.

The node in the G graph from which no arrows descend represents a protocol deadlock state. Protocol has liveness property if the G graph is strongly connected. Several algorithms were published to examine the strong connectivity of an oriented graph /e.g.[Aho-Hopcroft-Ullman 75]. The protocol cycles correspond to the loops of G graph. Examination of protocol cycles to determine the useless loops /progress property/ concerning the human designers demands the knowledge of a concrete algorithm. Other protocol properties formulated in the form of predicate invariance can be examined with the evaluation of the predicates in some nodes of G graph.

In order to put this method into practice the system state has to be defined in the case of concrete language representation of /K,L/ system. The language representation of /K,L/ system uses entities communicating with each other to describe the protocol. In that case the system state was defined a q= e<[1], e[2],... e[n]> where q is equal to a vector /n-tuple/ of the states of entities. The state of the entities is defined by the values of /entity/ local variables and the state of the queues used in the communication between entities.

3. VERIFICATION SYSTEM

The protocol vertification system consists of MODULA-2 modules represented by compiled files under RT-11 operating system. Having preprocessed and compiled the protocol specification the user can link it to these files. Thus a form suitable to be run can be

obtained by including both the actual specification and its verification system. The usage of the system is shown in Figure 3.1.

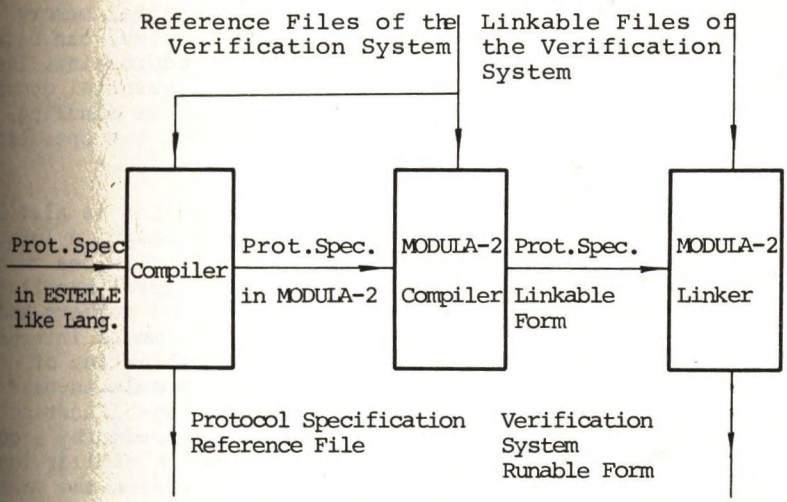


Figure 3.1 Scheme of the Usage of the Verification System

3.1 System Structure and Usage

The structure of the verification system with its most important modules and the access rights can be seen in Figure 3.2.

The Protocol Specification module has access rights to the modules of the Abstract Data Type Library denoted by solid lines in Figure 3.2. The MODULA-2 language requires static export and import lists which should be known at the compile time, i.e. the main module of the protocol specification and also the names of the variables and procedures imported from it. This contradicts one of the general requirements which prescribes that constant modules of a verification system are not to be changed in case of verification of another protocol. To solve this problem the simplest way was chosen: the name of the module containing the MODULA-2 representation of protocol specification and its interface to the modules of the verification system were fixed in advance, i.e. fixed module and procedure names can be seen from the main module.

In the Abstract Data Type Library there are separate modules for each data type. One of them is used for the system, too, so it is fixed /Queue/.The module with dotted lines represents the expandability of the Library. The SysTypeLibrary module provides a uniform interface for the verification system.

The largest and most important module of the system is the VerificationSystem module. It provides an algorithm for generating the reachability graph. The construction of the the reachability graph can easily be made with TreeStreamHandling module, which is

connected to the Files module of the RT-11 operating system through the VirtualStorage module. The Files module opens, closes, creates and deletes the files under the RT-11. Its basic service allows access to an assigned block of an assigned file. Based on this service the VirtualStorage module generates a virtual memory, which is homogenous and placed in one file. Words/or bytes/ can be written into or read from this memory according to its addressing. The Stream standard module generates the strictly sequential accessable word /or byte/ sequences. The VFile module provides handling, registering, opening and closing files needed for the operation of the verification system.

Similarly to file handling the input-output handling is also layered. The modules of the hierarchical system use services of a lower layer and provide more sophisticated and more abstract services to the upper layer. The TtIO module provides character level handling of a terminal, the MultIO module inputs and outputs the numbers and the strings while the VInOut module is the central command interpreter of the verification system which provides initialization of different domains of the screen. The VTextView module is used for the forward and backward paging in the protocol specification when the reachability analysis is interrupted and suspended by a command coming from the user of the verification system. A similar function is given by the VVarView module sending to the screen the values of variables in the specification. The VidFile module handles a service file containing reference information for the identifiers of the protocol specification.

3.2 Representation of the Reachability Graph

The choose of the inner memory representation of the reachability graph is one of the most important decision of the implementation of the verification system because it determines to the greatest extent the algorithms of the system.

In order to estimate the size of the reachability graph the expected average state numbers of the graph should be taken into account. Even in the case of medium size protocol the global state number is in the looo to lo.ooo interval if a finite state machine protocol representation is used. Using the ESTELLE-like specification language the dimension of the global state space is reduced because it increases the complexity of the operation of a transition and the granularity of the system observation. Our implementation is ready to deal with thousands of global states. It is expected that the memory size needed for a global system state covers about 5o - loo words in the case of our MODULA-2 language implementation. This memory need can be changed because in a protocol specification there are dynamic data structures /e.g. queues/ whose size /e.g. length/ can be modified at the time of operation.

The reachability graph is a non-binary tree because the system is nondeterministic and so it possible that from a global state there are some enabled state transitions. The nodes of the reachability graph are saved on the disk. Special pointers /disk address in two memory words/ are used to describe the arrows of the tree. The non-binary tree is represented on the disk by a binary tree in order to fix the number of special pointers of a tree node /global state/. To reduce the complexity of the tree tour algorithm the so-called "back to father" pointers and special field /tset field referring the enabled but not walked state transitions/ were used in the

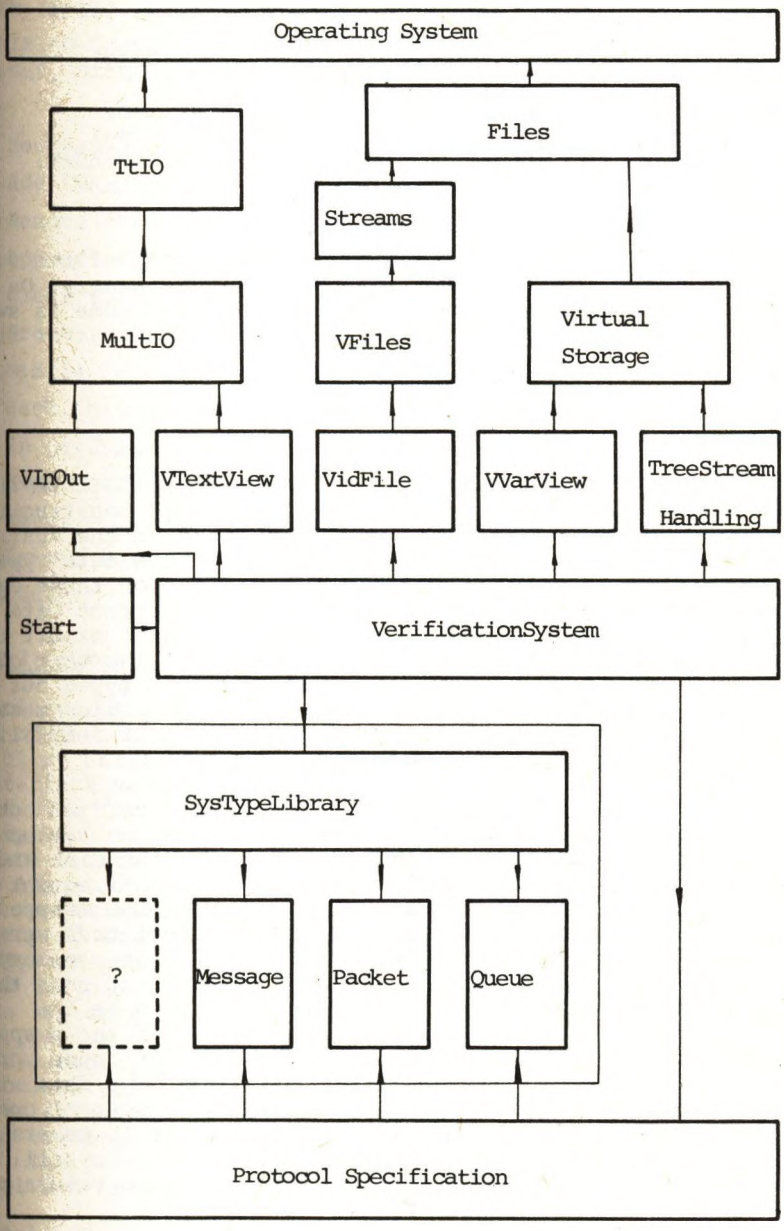


Figure 3.2 Implementation Structure of the Verification System

implementation. The last two fields of the Node record refer to the state transition executed finally.

```
TYPE
  ErrorType        =(good, deadlock, beforedeadlock, wait,
                     invarianterror, cyclefound);
  TransitionSet    = ARRAY [0..ENTITYNUM-1] OF BITSET;
  Node = RECORD
           father, sublist, nextnode       : Position;
           error                           : ErrorType;
           tset                            : TransitionSet;
           entitynumber, transitionnumber  : CARDINAL;
         END;
```

The VirtualStrorage module carries out the swapping of the subtrees of the reachability tree between the disk and the memory. On the base of a pointer /disk address/ the corresponding tree node is swapped in and out making possible to modify the state /e.g. error field/ of the node.

3.3 Abstract Data Type Library

The protocol designer describes the protocol in an ESTELLE-like specification language. Besides the basic language constructions he can use predefined abstract data types contained in the Abstract Data Type Library of the verification system. These data types consist of data structures and operations defined on these data structures. The MODULA-2 implementation of the Abstract Data Type Library is an expandable collection of modules which export the data types in the form of hidden type which means that the user of the data types can access to only the name of the data types but not to the inner implementation structure. The library contains message, packet and queue types handling procedures and special auxiliary algorithms.

The implementation of the abstract data types was carried out with utmost care. During the operation of the verification system the swapping of the global state information between the disk and the memory performed by the VirtualStrorage module is a frequent operation therefore it is worth to minimizing the amount of the swapped data. In this order the different message and packet instances were collected in some sublibraries implemented in the heap memory. Instead of swapping the whole message and packet instances the system swaps the indexes of the instances contained in the sublibraries. This solution decreased the amount of the swapped data, the limited size, however of these sublibraries can cause some problems. Theoretically, a large number of instances of a data structure consisting of n components can exist. During the protocol operation practically only a small subset of these instances is generally used but if this preliminary assumption does not hold it is still possible to eliminate this sublibrary concept and to return the swapping of the whole data structures.

The following example illustrates one of the data-type definition of the data-type library. The universal QueueModule does all queue handling within the system.

```
DEFINITION MODULE QueueModule;
FROM SYSTEM IMPORT WORD, ADDRESS;
EXPORT QUAIFIED Queue, NewQueue, Add, Remove, Append,
 Clear, Front, Middle, Back, In, Equal, Length;
TYPE Queue;
PROCEDURE NewQueue() : Queue;
PROCEDURE Add (VAR q:Queue; l:ADDRESS) : Queue;
PROCEDURE Remove (VAR q:Queue) : Queue;
PROCEDURE Append (VAR q1,q2:Queue) : Queue;
PROCEDURE Clear (VAR q:Queue) : Queue;
PROCEDURE Front (q:Queue) : ADDRESS;
PROCEDURE Middle (q:Queue; seq:INTEGER) : ADDRESS;
PROCEDURE Back (q:Queue) : ADDRESS;
PROCEDURE In (q:Queue; i:ADDRESS) : BOOLEAN;
PROCEDURE Empty (q:Queue) : BOOLEAN;
PROCEDURE Equal (q1,q2:Queue; slice:INTEGER) : BOOLEAN;
PROCEDURE Length (g:Queue) : INTEGER;
END QueueModule.
```

3.4 MODULA-2 Representation of the Protocol Specification

The protocol entities of a protocol specification were implemented in modules of the MODULA-2 language. The state and interface variables of an entity are locally declared variables of the module but the interface variables having been used for the communication between entities are exported by the module. The state transitions of an entity were transformed into procedures operating on the state and interface variables. The ActivePreCond procedure is used to inform the system about the enabled transitions at a given moment. The system can achieve the execution of a given state transition with the help of the RaiseActionWell procedure. The SaveSystem and the UnSaveSystem procedures save and reload the state of an entity to and from the reachability tree. The communication between the entities is implemented by assignments referring to the interface variables. Besides the built-in predicates, the user-defined predicate describing the individual protocol properties is implemented in a Boolean value function which is called if a new protocol state was established. In this way the system can check the invariance not only of the predefined but also that of the user-defined predicates. The procedures representing the fixed interface between the protocol and the verification system are the following.

```
PROCEDURE ActivePreCond (VAR t:TransitionSet);
PROCEDURE RaiseActionWell (entity, transition:CARDINAL):BOOLEAN;
PROCEDURE SaveSystem(s:VSTORAGE);
PROCEDURE UnSaveSystem;
```

3.5 System Operation

Having been loaded the linked, runable form of the verification

system can be started. During the initialization the state and interface variables get their initial values in the executable section of the protocol specification. Then the system saves the state of the protocol. This state means the root of the reachability tree. In the next step system examines whether it has executable transitions. If it reaches one or more, it choses one of them, executes it and starts to analyse the new state got after the transition. This state is considered as a deadlock if the system has no executable transition and the function Wait results in a false value. If the function Wait is true the new state is given a "wait" qualification. It will be given an "invarianterror" qualification if an invariant error is found. In order to find the appropriate protocol cycles the system compares the new state with the previously saved states. During saving a new state the identifier of the starting state is also recorded in a protocol cycle. If the examination of the new state does not result in a deadlock or a wait situation, a cycle or invariant error, the new state is given a "good" qualification and the operation is continued in a normal way as mentioned above. When the qualification of the new state is not "good" this is the end of a leaf of the reachability tree. In this case the system steps back, reloads the previously saved state of the protocol and examines whether the reachability tree has other subtrees originating from the same node that have been examined yet. If there is one, the new subtree is examined. If there is not, another operation is continued with another backstep. The verification process comes to an end when the whole reachability graph has been constructed.

The user sitting in front of the screen can follow the operation of the system. A task of the user is to pick out the unlimited branches of the reachability tree if there is any. During the operation he can ignore certain unimportant sections of the reachability graph with operator commands. The user also can get information about the values of different variables and about the text of the protocol specification if he interrupts the construction of the reachability tree with an appropriate command.

4. CONCLUSIONS

The verification system presented here is considered an experimental system. Our first few experiences about the operation of the system are the following:

/1/ The system works quite well, it is capable to check all the static and dynamic characteristics of the protocol that was included in the specification.

/2/ If a complicated protocol was chosen, the large number of protocol states caused a general problem. A great amount of memory space and processing time was needed to generate the reachability tree.

/3/ The system operation was much slower than expected. One of the reasons is the ineffective organization of the input and output. It can be improved by reducing the size of memory necessary for storing the reachability tree. These mesures may result in a less frequent access to the disks, so they may speed up the operation of the system.

/4/ The applicability of the system is limited as it is implemented on a relatively small computer under the RT-11 operating system. Larger configuration may provide wider applicability.

REFERENCES

[Aho-Hopcroft-Ullman 75] V.A.Aho,J.E.Hopcroft,J.D.Ullman: "The Design and Analysis of Computer Algorithms", Addison-Wesley Publ. Comp. 1975.

[Bochmann 82] G.V.Bochmann, E.Cerny, M.Gagne,C.Jard, A.Leville,C.LaCaill,M.Maksud,K.S.Raghunathan, B.Sarikaya: "Experience with Formal Specifications Using an Exteded State Transition Model". IEEE Trans. on Comm. Vol. COM-3o. No. 12. Dec. 1982.

[ESTELLE 84] "A Formal Description Technique Based on an Extended State Transition Model" ISO TC97/SC21/WG1 - FDT, Subgroup B Working Document 1984 Dec. 1o.

[Harangozo 83] J.Harangozo: "State of the Art of Protocol Design in the Early Eighties", New Results on Computer Communication II. Protocol Models, Technical Report KFKI-1983-98

[Keller 76] R.M.Keller: "Formal Verification of Parallel Programs", Comm of the ACM Vol. 19. No. 7. Jul. 1976.

[Kovacs 82] L.Kovacs: "Formal Specification and Verification of Computer Network Protocols", Technical Report, MTA-SZTAKI Tanulmányok 138/1982. /In Hungarian/

[Kovacs 83] L.Kovacs: "Formal Description of Protocols with the Help of a Specification Language", Informacio Elektronika 18. 1983/4. /In Hungarian/

[Kovacs 84] L.Kovacs: "Automated Protocol Verification", Third International Meeting of Young Computer Scientists, Smolencie 1984.

[Wirth 8o] N.Wirth: "MODULA-2", ETH-IFI, No. 36 Zurich 198o.

VEDA: a software simulator for the validation of protocol specifications

Claude Jard, Roland Groz, Jean-François Monin

Département Evaluation et Validation de Protocoles
CNET LANNION A, Route de Trégastel, BP40,
F-22301 LANNION CEDEX FRANCE

ABSTRACT

This paper presents an overview of the functions of a protocol simulator, called Veda. Veda is a software tool addressing the problem of protocol specification, design and validation. It allows rapid prototyping of distributed algorithms. Veda is based on the ISO-CCITT Formal Description Techniques (FDT) for protocols.

Two particular aspects of validation using a simulation method are presented in a little more detail: the possibility of programming test strategies and on-line verification by observing the system simulated.

I Introduction

An increasing number of distributed algorithms (also called protocols) are being designed and implemented. They provide such services as the transmission of information between sites (communication protocols), resource allocation, or consistency control in data bases, etc ...

Proving that these algorithms work correctly is difficult because of their distributed nature; moreover, they usually communicate through an imperfect medium.

For some time, verification methods have been developped (see for example |IFIP 81,82,83,84|). Proving a protocol is still quite a difficult problem, and until now, proof systems cannot take into account real-size protocols (a validation experience using different methods on a real protocol is described in |Groz 84|).

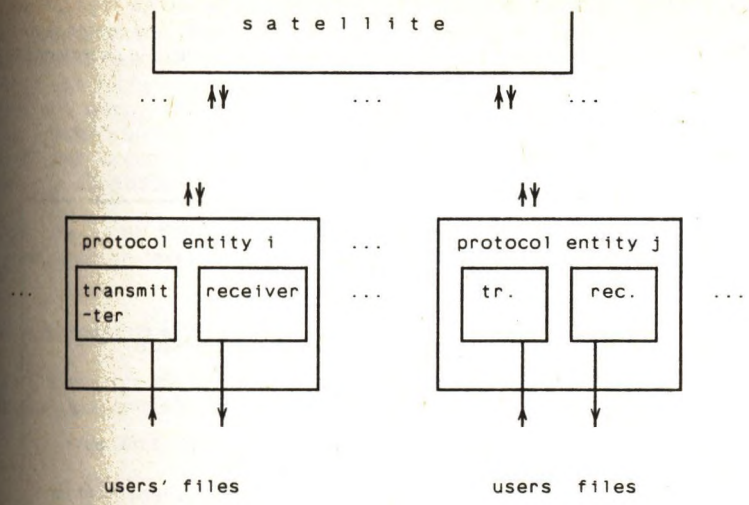
There are many techniques for proving the correctness of such systems. They are based on exhaustive exploration of states, proof by induction on the structure of programs, or algebraic computations.

The first approach is severely limited by the explosion of the number of states, even for simple programs. Methods using induction techniques usually work only on restricted classes of algorithms, and most of the time they are not fully automatic, and ask for hints from the user. Lastly, algebraic methods require a significant modelization effort, and a proof based on such a model only gives results on a part of the initial problem.

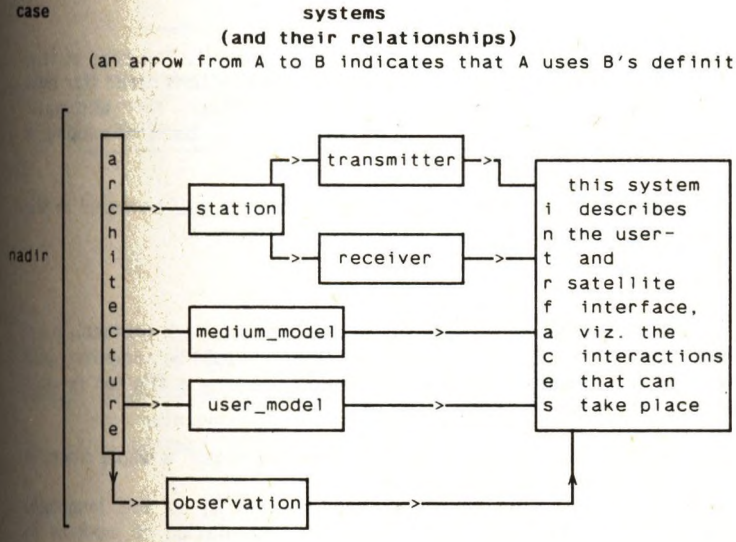
Being asked by designers of distributed systems in our research center to validate their real-size protocols, we have prefered to limit our objectives to obtaining an acceptable degree of confidence. Therefore, we have taken the view that a given protocol works tolerably well if it is proved to work correctly under various conditions, particulary when "dangerous" events happen.

Preliminary studies have shown that "specification testing" allows the discovery of non-trivial errors.

This paper presents a simulation system called "Veda", which could be used for debugging, validating and evaluating distributed algorithms. Veda was designed and implemented in our team of the French National Research Center for Telecommunications (CNET).



Example of a satellite file-transfer protocol



one possible veda representation (structure only)

Figure 2

Section II describes the general organization of the system . The specification language used for the described protocols is presented in section III, and methods used in Veda for validation are discussed in section IV. Section V mentions our experience in the use of Veda.

II System organisation.

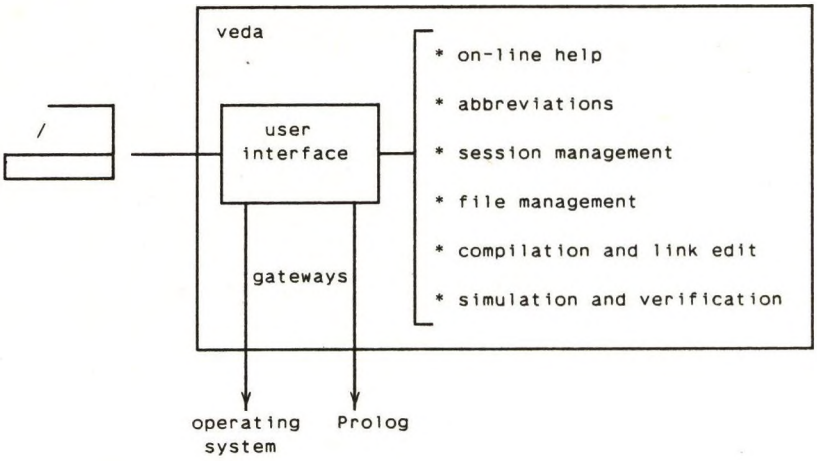


Figure 1

Veda offers a programming environment summarized in figure 1. The core of it is the compiler-simulator described in figure 3. Other facilities are considered (for next year and further) including a syntactic editor, facilities for automatic implementation ... A parallel project called Atlas (see V.2) has been developing a database of (modelized) network packages.

We shall now briefly outline the system organization. Details will be found in the Veda reference manual [Jard 85a].

II.1 Structuring: an example

Consider the example of a satellite file-transfer protocol (one such protocol, from NADIR, has been simulated on Veda). The global architecture would be something like figure 2. Any station (protocol entity) can be either transmitting a file (on request from users connected to this station) or receiving one.

Suppose we wish to validate this protocol. In Veda, we call "case" a whole example (it is a case study). Let us call this case "nadir".

The language chosen for describing protocols (see section III on this language) allows a modular specification. We have added separate compilation of modules in Veda. In that way, all the different descriptions can be put into different files. Each description is called a "system". More precisely, a "system" in Veda corresponds to one compilation unit.

A simulation of this protocol involves a description of the transmitting protocol, a description of the receiving protocol (separately, or both of them in the same description). So, we could have, within case Nadir, two systems: transmitter and receiver.

But we simulate only closed architectures in Veda, so we would also provide models for the behaviour of the users and that of the satellite medium. That could add two other systems: medium_model and user_model (possibly more if several behaviours are

considered for users, or if the medium is further refined into subsystems).

Finally, in Veda, one or more compilation units could also include observers: these are special modules used for verifying properties of the protocol (see IV.3).

The whole architecture would also be described within a system. It does not have to be compiled in a system of its own: it could have been put into the same file as, say, the medium specification, and (or) the protocol specification. The same applies to all other specifications: they can be described together or separately: on figure 2, we have only shown a "natural" way to split the example.

In short, the basic object in Veda is a "case". It consists of one or several specifications in a high-level protocol-oriented language. Each specification is called a "system".

II.2 Flow of information within Veda

Figure 3 illustrates the path from a formal specification to its simulation. Each system appears as a compilation unit.

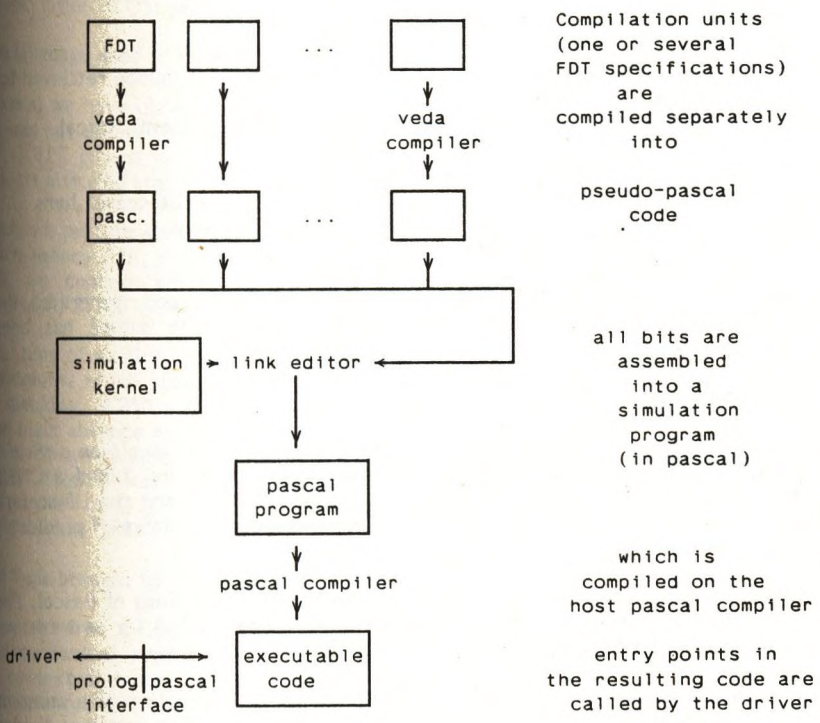


Figure 3

Given a compilation unit (a source file), the "compiler" produces an intermediate code (in a temporary file), used by the "generator" which in turn generates a file containing a pseudo-Pascal text. So we perform basically a two-pass compilation. The first pass performs lexical, syntactic and semantic analysis; the second pass generates (Pascal-)object code. The compiler is written in Prolog, except lexical analysis directly written in Pascal (here we use the Pascal-Prolog interface of Prolog-CNET). The compiler performs parsing and semantic analysis. Semantic errors are recovered from in the present version. At the time of writing, recovery on syntactic errors is very limited. The intermediate code is a set of Prolog clauses including all the information necessary for generating the pseudo-Pascal text.

The different pseudo-Pascal files, when assembled together during a link editing stage, produce a set of Pascal data structures and procedures. After that, they are compiled together with a "simulation kernel", so as to produce an executable code. The kernel has to manage parallelism (scheduling), and communications. Its code does not depend on the protocol under study. Link editing is performed simply with a stream editor.

The different Pascal procedures (configuration, guard evaluation, transition firing) are executed from the Prolog interpreter (we again use the Pascal-Prolog interface). The Prolog module driving the simulation is called "executeur" (French for driver or executioner !). It is very flexible, and could deal with evolutions of the execution semantics of the language. It can be used for programming special coverage algorithms. See section IV.2 on this driver.

The information maintained by Veda for each component is the following.
NB: filenames depend on the operating system; those given below correspond to the Multics implementation of Veda.

For each system:

* its source description in a file: <case>.<system>.desc
* its compiling environment in file <case>.<system>.env
  this is used for implementing an authentic separate (not indebe retrieved for types internal to a subsystem)
* the object (pseudo-Pascal) code generated in <case>.<system>.Pascal

For each case:

* a graph of the separate compilation relation "use", in a file <case>.liens
* the Pascal code resulting from link editing: <case>.Pascal
* the executable code, in <case>

Technical information about the compiling process, the Pascal code generated, the simulation kernel and the semantics simulated for this Estelle subset has been published in |Jard 85b|.

II.3 Size and performance

Veda is a high-level package written in Prolog, simulating protocols on a virtual Pascal machine. We use standard ISO Pascal, and Prolog/P, version 2.00 (a CNET product, which is available through the CRIL company in France and the US so far). This Prolog is efficient and offers an interface with Pascal for "hardwiring" predicates (in Pascal).

Veda consists of approximately 6700 lines of Prolog and 2300 lines of Pascal. The Prolog part is structured in 13 modules (a feature of this Prolog). As far as we know, this is the first real-size compiler ever written in Prolog.

Users of Veda currently compile systems from 2 (separate definition of a constant or type ...) to 1200 lines; some cases are made of sevaral thousand source lines divided into up to 30-odd systems.

III. Language used for the description of protocols

This section summarizes what has been done so far in ISO and CCITT study groups. Veda uses a version of a language currently in the process of standardization. We indicate how Veda stands regarding this language, and what work has been done in Veda to complete the specification of the language.

III.1 ISO and CCITT languages

ISO (International Organization for Standardization) and CCITT (Comité

Consultatif International Télégraphique et Téléphonique) have been studying for some years formal description techniques (FDT) for the specification of protocols. This work was motivated by the fact that informal specifications, as those currently given in most protocol standards, cannot be trusted to eliminate all ambiguities. Unfortunately, ambiguities in standards often lead to incompatible implementations, a result opposite to what was intended by standardization.

On the ISO side, working group 1 of ISO/TC97/SC21 (formerly SC16) has designed two kinds of techniques. One is named LOTOS; it is based on temporal ordering of interactions (using a CCS-like model). The other is called Estelle (for Extended State Transition Language); it is based on finite automata, extended with Pascal. Since February 1985, these two techniques are now ISO Draft Proposals.

On the CCITT side, question 39 of subgroup VII had worked on SDL (Specification & Description Language, defined in recommendations Z.101 to Z.104) and harmonization with Estelle. Recommendation X.250 (of 1984) defines a usable subset of SDL and a language similar to Estelle (in fact a subset of Estelle).

The CCITT group and the Estelle group of ISO merged in 1984 Following the latest ISO meeting (February 1985), three languages are henceforward recognized for the formal description of OSI protocols: SDL, Estelle and Lotos.

III.2 The language used in Veda

For Veda, we have chosen to describe protocols in a standardized (-to-be) language. Our aim was not to develop a new language. The state of the language defined in the ISO and CCITT groups was close enough to that of a workable language for us to choose it. Its intended standardization was an asset.

In fact, we participated in CCITT's effort, and, together with the RHIN project of the French agency ADI, and the LAAS laboratory of the French National Center for Research, we contributed to its definition. Work on the formal semantics was conducted with LITP (University of Paris VII) and LGI (University of Grenoble).

At the time of writing, the language used by Veda is the one defined in X.250, plus some extensions, which situate it half-way between X.250 and (the current) Estelle. Roughly speaking, X.250 was incomplete because it included some points for further study; the main absence was that of language constructs to describe the architecture of a system. So we completed X.250 by adding some kind of static configuration. Since that time, Estelle has been completed with both static and fully-dynamic configuration. We plan moving to the full Estelle (as defined in |ISO 85|) during the course of this year (1985).

III.3 What Veda brought to the definition of this language

The context-free syntax is taken from X.250. However, almost nothing is said about the contextual syntax, neither in X.250, nor in |ISO 85|. So we defined a complete contextual syntax for this language; for the Pascal part of the language, we refered to extended attribute grammar methods from |Watt 79|.

As the pool of users of Veda increased, and as they had bigger and bigger Estelle descriptions (of one or several thousand lines), it soon became necessary to add a mechanism for separate compilation to the language. In fact, the language is very modular itself, making it natural to define separate entities as different compilation units.

As far as the semantics is concerned, Veda is oriented towards simulation. To that purpose, we defined a centralised implementation simulating the parallel execution of several Estelle modules. X.250 and Estelle have rendez-vous communication for further extension. We have given a semantics for this extension, and it is implemented in Veda. So, a user can use either of two communication modes: the rendez-vous mode

* between processes (to take account of difference of speed);
* on the next temporal event to be chosen (this comes from the temporal construct "delay (d1,d2)" in the language); a uniform choice is made between compatible delays, viz. when their intervals (d1,d2) (which represent lower and upper bound) overlap.

- Interactive mode: when a choice has to be made, a question is asked to the user. The different possibilities to be chosen from are shown on the screen.

- Trace mode: choices are made randomly, but they are shown to the user.

Programming efficient strategies is still research work. By efficient strategies, we mean those which would provide a better detection of errors in a specification. The driver of Veda has been designed to make the implementation of such strategies possible, and relatively easy.

IV.3. Observing a simulation

IV.3.a Rationale

In order to validate a protocol, it is necessary to have means of expressing some properties of its behaviour, in the form of relations between events or logical formulas for instance. Simulation by itself does not provide such means. Nor does it give a precise account of what happens: even an interactive simulation displays only internal events at the lowest level, it does not illustrate higher level events like those relevant to the service. In Veda, we use a notion of observer to placate these needs.

We may distinguish two levels of use: trace and verification. Tracing the behaviour of a protocol consists in writing events occuring in the course of a simulation. Tracing is of great help when debugging a "first draft" specification of a protocol. Verification is a further step: it means ensuring that events occur as expected (according to, for instance, a service specification). One way of verifying is to record traces on a file and input this file to an off-line trace-checker. Another possibility is on-line verification.

What we have been looking for, in Veda, is a method for tracing and verifying the behaviour of a protocol. And, from the start, we have put the emphasis on offering the highest expressive power Veda was able to provide. The crucial point, in that view, is not so much the formalism for expressing service properties, as the basic notion of event. Events are the primitive conditions on the behaviour of a protocol which an observer can take into account. Then, we have maximum power (for this class of events) simply by allowing the observer to compute any computable function on these events. For this, we just have to make the observer a program. Our choice for the class of events is discussed below.

IV.3.b Shortcomings of some simple methods and choice for Veda

Consider for instance the simplest tracing method: inserting display commands (writeln ...) inside the specification of a protocol. This was the method used in Veda when observation was not yet implemented. It has numerous shortcomings. First, the specification is polluted with irrelevant procedure calls. It turns out that they represent about as much text as the specification itself. Second, its expressive power is limited by the fact that it can trace only local events (those involving only one module), and only when the module is active. Lastly, it is well-suited to tracing, but no on-line verification can be performed.

Another course, used in implementations, is to add extra modules whose function is to observe the system. In that case, observer modules are limited by the delays in the transmission of information, which implies that they cannot have a coherent view. They are also limited by their own speed. Worse, there is a risk that the behaviour of

the system within the compound system+observers is different from the behaviour of the system alone.

In Veda, observers are spies watching the execution of a protocol.They have the following characteristics:

- they are described apart from the protocol; the specification of the protocol is thus in no way altered;
- they do not interfere with the semantics of the execution of the protocol under observation;
- they have immediate access to what they observe (see semantics below);
- the events observed are described in high-level terms, in fact, in terms of the specification itself;
- the language in which observers are described is an extension of Estelle; in this way, there is no extra language to learn when using Veda.

IV.3.c Basic observation structure

A system under study is made up of an assembly of modules. It can be observed by one or several observers. This is shown in figure 4.

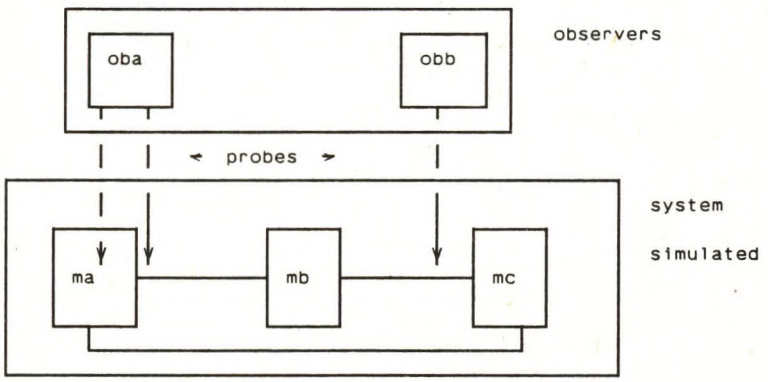


Figure 4

An observer is a module. What makes it different from other (i.e. protocol) modules is that it has "probes". Probes differ from interaction points (an Estelle concept) in that they do not interfere with whatever point they are connected to. They simply enable the observer to know immediately what has happened at this point.

IV.3.d Semantics for observers and observable events

In Veda, an observer has a perfectly coherent view of what happens in the system. It can be seen as a kind of daemon which knows instantly the state of all the modules it observes. This is important, because most service specifications for protocols presuppose such a global view. Since our main intention was to validate protocols against their service specification, we had to be able to express this service. All this daemon stuff is no trick; it is possible simply because we simulate a distributed system, so that we have complete control over its execution, which would not be the case if we were in the distributed system itself.

An observer observes through its "probes". A probe can observe

- either a module interaction point (in Estelle, this represents a "gateway" through which a module can interact with other modules); in this case, the probe communicates all interactions observed to the observer (in the figure, observer oba can observe all interactions between ma and mb).
- or a module itself; in this case, the probe gives the observer access to all the internal state variables of the module. This can be used, for instance, to prove that some predicate on the variables of one or several processes is always true (i.e. is an invariant). An observer can verify a predicate bearing on several modules, because through its probes it can have access simultaneously to all these modules.

The observer itself is an extended automaton. Its transitions are fired by the events observed. The syntax is basically that of Estelle, but with extra-facilities to express any computable function on the events.

IV.3.e A small example

To illustrate the use of observers, we shall describe a very simple protocol for resynchronizing logical clocks between two stations. This algorithm was designed by G. Roucairol and O. Carvalho (private communication).

Consider two stations, each with a logical clock h. The algorithm ensures a bound on the drift: |h1 - h2| < or = delta . Logical clocks and delta are assumed to be integer. Each station maintains two other variables: u and l (upper and lower bound for the clock). At any time, a station can increment its clock provided it is less than u. From time to time, it communicates the value of its clock to the other station. And when it receives a clock value, it updates its u bound. Here is a short description of the algorithm (NB in this version, l is not used).

```
system Carvalho_Roucairol ;
   const  delta = 4 ;
          d1 = 0.6 ; d2 = 8 ;
          inc1 = 1 ; inc2 = 13.4 ;
   type   logical_time = integer ;
   block global ;
   refinement rg for global ;
      channel Resynchronizing ;
         My_time (x: logical_time) ;
      block Station (R: Resynchronizing) ;
      process CR_resync for Station ;
         queue R ;
         state (dontcare) ;
         var u, l, h: logical_time ;
         initialize begin h:=0; l:=0; u:= delta end ;
         trans
            delay (inc1,inc2) provided h<u begin h:=h+1; l:=h end ;
            delay (d1,d2) begin output R.My_time (l) end ;
         trans
            input R.My_time(x) begin if u<x+delta then u:= x+delta end ;

      subblock a,b : Station ;
      configure   begin
                        initialise a with CR_resync ;
                        initialise b with CR_resync ;
                        connect a.R = b.R
                  end ;
```

Although this algorithm can cope with an unreliable communication network (loss and reordering of messages), we have not modelized this network here, for brevity's sake: the two stations communicate directly.

Suppose we wish to observe the behaviour of these two stations. We define a simulation architecture which contains one instance S of the system, and an observer O which we shall set on observation of S. This is done in the following manner:

```
simulation O_C_R ;
   use Carvalho_Roucairol ; (* uses a separately compiled system *)
   observe
   block obs observe (   process p1,p2: rg.CR_resync
                         channel c: rg.Resynchronizing ) ;
(* this declares an observer module with two probes for observing processes of type
CR_resync defined within refinement rg, and one probe for observing
communication between the two, which is of type Resynchronizing *)

   observer obsr for obs ;
      ... (* details later *)
   subblock S: global ; O: obs ;
   configure begin
                  initialize S with rg ;
                  initialize O with obsr ;
                  observe O.p1 : (S,rg).a ;
                  observe O.p2 : (S,rg).b ;
                  observe O.c : (S,rg).a.R
             end .
(* We have set probe p1 to observe submodule a, i.e. the first station, and probe p2
on the second station; and probe c on the channel that links a and b *)
```

Suppose we simply wish to display on the console the evolution of the logical clocks on the two Stations, and their communications. Then we write observer obsr as:

```
observer obsr for obs ;
   state (dontcare) ;
   initialize begin end ; (* syntactic rules of X.250 makes initialize compulsory *)
   trans
      delay (0) begin writeln ('p1:',p1.h:3,' ; p2:', p2.h:3) end ;
      delay (0) begin exrepprobesmess end ;
```

The first transition states that the observer will trace the two logical clocks, the second that it will trace all messages seen by probes (here there is only one probe observing messages: c); exrepprobesmess is a predefined procedure that traces interactions (messages). Note that the reference to the two logical clocks is very simple: p1.h is variable h of the module observed by probe p1.

Now, if we were interested in verifying the service property |h1-h2|<=delta for this protocol, the observer would simply have a transition:

```
   trans delay (0) provided abs (p1.h-p2.h) > delta
      begin writeln ('error') end ;
```

We see that the verification of a logical property is very easily translated into an observer program.

Instead of having an error message forwarded to the user, we could program the observer as an acceptor automaton:

```
observer accept_protocol for obs ;
   state (OK,NOK) ;
   initialize begin nextstate OK end ; (* this is the initial state *)
   trans delay (0) provided abs (p1.h-p2.h) > delta
      from OK,NOK to NOK begin end ;
```

In the case of more sophisticated service specification, we could thus program observers to accept behaviours defined by regular expressions or other methods. Since

observers are fully programmable, they are not limited to regular expressions: the could recognize any language that a Turing machine could recognize (ignoring memory limitations).

IV.3.f A few words about linguistic constructs

The requirement that observation should be specified apart from the system under study was motivated by the fact that it is desirable to have a system specified, and then, observe such or such feature in its behaviour. This requirement, coupled with the fact that elements observed are refered to in high-level terms, i.e. using identifiers and objects from their source (Estelle) description, implies that observers must have access to the "compilation symbol tables" (this is implementation talk) of the system being observed. This constraint is made harder by the fact that the configuration is always done dynamically (even in our version of Estelle). The way all this was implemented is not detailed in this paper.

We shall just explain what appears on the example described in IV.3.e. First, the construct rg.CR_resync (or rg.Resynchronizing) is an instance of the pattern:

<access-path> . <observed-type>

where the access path is needed to reference a type that has been declared within a sub-construct. If there were several nested levels of declaration, the path would have to go down all these levels. For instance, if process CR_resync had been defined within refinement Ry, itself declared within Rx, itself declared within rg, we would have:

rg.Rx.Ry.CR_resync.

Note that rg is made visible in the simulation by the use clause.

Second, to set a probe on a given module instance, or an interaction point, we must again give an access-path to this instance. However, in this case, the path does not correspond to the nesting of declaration scopes, but to the actual nesting of instances. Since the initialisation of module instances is done dynamically, we must specify at compile-time the type (of behaviour) of each instance in the access-path. This is what is done in the following construct:

(S,rg).a.R

to observe interaction point R of instance a within instance S which has been initialized with rg.

Other constructs exist for refering to interaction parameters, interaction types etc... But since they do not appear in our simple example, we shall ignore them in this presentation paper.

IV.3.g Versatility and use of observers

With observers, we have seen that it is possible to:

- trace some events, and then analyze them either "manually" or with an automatic trace-checker; this tracing is fully programmable.

- make an on-line verification of whichever predicates or service properties are of interest.

For verification, programming observers is of the same level as the protocol under study, whereas some higher level of specification could be better suited for the service. However, observers programmed in Estelle have all the expressive power needed to represent all sorts of formalisms. For instance, rational expression translation is straightforward (since automata are a restriction of observers). We plan introducing the verification of services expressed in such formalisms as rational expressions or temporal logic formulas or communicating local observers as in |Banâtre 84|, with automatic generation of observer programs to verify them.

But observers can be put to other uses. For instance, they are very convenient for experimenting with various real observation schemes. Veda's observers are almighty

daemons. But they can mimick the physical limitations that reality imposes on real observers. For instance, local observers as in |Banâtre 84| can be implemented on a real auction system, or any kind of distributed application. With R. Dssouli, from the University of Montréal, we have also experimented schemes for testing Transport protocols |Dssouli, Bochmann 85|. In that view, Veda can be a valuable tool for rapid-prototyping testing schemes or devices.

V Applications

Most simulation features of Veda had been tested first on a preliminary version of this tool. Several real-size protocols were tested, among which a fault-tolerant mutual exclusion algorithm (| Groz 84|), and the first version of the Architel session protocol (|Jard 84|).

Veda is a basic tool for the validation service we provide to protocol designers. It is currently being used on an experimental basis, for the Enchère system (Banâtre 84|), for a duplicated data-base algorithm, and telecommuniction applications (modelling switch interfaces and functions for an integrated digital service).

Veda has also been chosen as a basic component of the ATLAS project (see section V.2 below).

V.1. The stages of a protocol validation using Veda

The first step will be the specification of the algorithm in the ISO-CCITT language. This step is quite productive in pointing out all remaining ambiguities in an informal specification.

The second step is spent debugging the model, using the interactive and automatic simulation modes. Gross errors are detected at this step.

Once a protocol has been debugged, it should be validated. The service properties to be verified must be defined, and some test strategy too (at least values for all parameters and the length and number of simulation runs should be chosen). This step will detect anomalies due to unforeseen situations, which could hardly have been found without automatic tools.

A last stage could be some sort of performance evaluation of the algorithm, measuring such characteristics as required.

When simulation no longer yields errors or anomalies, sub-models of the whole protocol can be extracted to attempt a formal proof, by means of other tools. Otherwise, the algorithm may undergo a number of shuttle steps between the design board and the validation step.

Our experience with protocols already scrutinized with this method shows that simulation can detect real non-trivial anomalies which had gone unnoticed, even after months of execution of the protocol in a real distributed system.

V.2. The ATLAS project

This project was started to set-up a permanent structure for creating and updating an "atlas" of distributed algorithms, in a standardized way. Algorithms in this ATLAS are expected to be small, didactic algorithms, representative of a key-idea; classical, published algorithms would find their way in the ATLAS, and algorithms resulting from new research as well.

To be more than a mere survey or review of algorithms, it is stated that all algorithms will be described and validated using the same systematic approach. All algorithms will be presented using the same language, and their requirements

concerning the underlying environment (as well as the way they should be used) will be stated in a standard way.

The emphasis is on formalism, verification and performance evaluation.

ATLAS is a nation-wide project, with contributions from universities from all over France. It is one of the project in the C^3 action launched by the CNRS (National Centre for Scientific Research).

Veda is used for compiling algorithms and testing them systematically.

VI Conclusion

This paper does not present a research work, but a prototyping tool for protocols and all kinds of distributed algorithms. The method is based on simulation. Tools of this kind are, as yet, scarce.

Veda does not address all problems in the validation field. We think that such a tool is a useful complement to validation tools based on exhaustive proof. As a matter of fact, we validate protocols for protocol designers in the French Telecommunication Research Centre or elsewhere using Veda and other tools (currently Petri-net tools, especially OGIVE/D, and next CESAR |Queille, Schwartz, Sifakis 83|).

Another interesting feature is that the language used for describing protocols is a version of ISO-CCITT standards. It is one of the first "compilers" of this language which will be used to describe the next protocols to be standardized (the session level).

VII References

|Banâtre 84| Michel Banâtre, "Le système Enchère, une expérience dans la conception et la réalisation d'un système réparti", Thèse de doctorat d'état, Université de Rennes I, 23/3/84.

|Dijkstra 83| EW. Dijkstra, WHJ. Feijen, AJM. Gasteren, "Derivation of a Termination Detection Algorithm for Distributed Computations", Information Processing Letters 16 (1983), 217-219.

|Dssouli, Bochmann 85| R. Dssouli, G. v. Bochmann, "Error detection with multiple observers", Proceedings from the 5th IFIP WG6.1 workshop on Protocol Specification, Verification and Testing, M. Diaz ed., North-Holland publ.

|Groz 84| R. Groz, C. Jard, C. Lassudrie, "Attacking a complex distributed algorithm from different sides: an experience with complementary validation tools", IV IFIP workshop, Skytop, june 1984. To be published in Computer Networks.

|IFIP 81,82,83,84| Proceedings of the annual workshops on Protocol Specification, Verification and Testing organised by IFIP/WG6-1.

|ISO 85| ISO/TC97/SC21/WG16-1 N422. Draft Proposal. Estelle: a FDT based on an Extended State Transition Model.

|Jard 84| C. Jard, "Protocoles et services: test des spécifications", Thèse DDI, Université de Rennes, n° 173, mai 1984.

|Jard 85a| C. Jard, J.F. Monin, R. Groz, Véda: Présentation générale et manuel d'utilisation, CNET Internal Report: NT LAA/SLC/203, May 1985.

|Jard 85b| C. Jard, J.F. Monin, R. Groz, "Experience in implementing X.250 (a subset of Estelle) in Veda" V IFIP WG6.1 workshop, Moissac June 1985. Proceedings published by North-Holland.

|Queille, Scwartz, Sifakis 83| J.P. Queille, J.P. Schwartz, J. Sifakis, "Description, Specification and Analysis in CESAR : four examples", Proceedings workshop on the analysis of concurrent systems; published in Lecture Notes of Computer Science.

|Richier 84| JL. Richier, Distributed termination in CSP: symmetric solutions with minimal storage", Rapport interne LITP, septembre 84, 84-49.

|Watt 79| D.A. Watt, "An Extended Attribute Grammar for Pascal", SIGPLAN Notices 14 (2), 60-74 (1979).

FORMAL DESCRIPTION,VERIFICATION AND IMPLEMENTATION OF A LOCAL AREA COMPUTER NETWORK TRANSPORT PROTOCOL

J.Szentner
Research Inst.
VIDEOTON
Budapest, Pf.65
1525 HUNGARY

L.Kovács
Comp.Netw.Dept.
Comp. and Aut.Inst.
Budapest, Pf. 63.
1502 HUNGARY

J.Harangozó
Dept. of Proc.Control
Techn. University
Budapest
1521 HUNGARY

ABSTRACT

The paper presents a complete design process of a local computer network transport protocol. The transport protocol is well-suited for the special demands of a local area computer network architecture. The design process consits of the informal and formal specification, verification and implementation of the protocol. The protocol is described by a high level protocol specification language which makes it possible to use an automated protocol verification system. The structure of the implementation corresponds to the modules of the protocol specification.

1. INTRODUCTION

The paper describes a complete design process of a local area computer network transport protocol. Our attention was focused on the problems how to join the formal specification technique, the verification method and the implementation of the communication protocols. Using these "steps" as part of a unified design process yielded a transport protocol well-suited for the special demands in an experimental local area network EXLOC of the VIDEOTON factory.

The second section deals with the requirements of the transport layer which come from the characteristics of the architecture of VIDEOTON EXLOC experimental LAN. The third section describes the main functions of the designed protocol.

An FDT dialect formal description method was used combining the state transitions of finite state machines with the power of a high-level programing language. Our FDT dialect is based on the MODULA-2 language instead of PASCAL which had been proposed by the Subgroup B of the ISO TC97/SC16/WG1 in [FDT 81], [Boch 83].

Two approaches to protocol validation were used. One of them is the "major state abstraction" of the system ignoring the additional state variables and the interaction parameters of the protocol model. The second one is the extended reachability analysis involving the state variables and the interaction parameters.

The operating system environment and the structure of the implementation is dealt with in the last section. The implementation was

checked with the help of a protocol environment simulator.

2. ARCHITECTURAL REQUIREMENTS FOR THE TRANSPORT LAYER

The EXLOC architecture is based on a 10 Mbits/s coaxial cable bus system using CSMA/CD medium access method. In such networks the main characteristics of the data transfer /important for the higher layers/ are the following: /1/ the maximum packet lifetime is small and can be bounded; /2/ there is no acknowledging mechanism; /3/ the possibility of error-free arrivals of packets is very high but lost, duplicated or missequenced packets may exist.

The architecture should support communications among many application entities /not necessarily placed within a single LAN/. The reliability feature mentioned in the previous paragraph may usually be proved insufficient for an application process. As it is known the reliability generally can be improved at two levels: the data-link or the transport levels. Here the transport level was chosen because it simplifies the data-link layer protocol and it has an end-to-end significance through the concatenated LANs.

In our system the different applications require mainly transaction type communication because the transport service data units /SDUs/ are selfcontained and not longer than the network SDU and the destinations of the SDUs may be different during the life of an application process. Sometimes, however, a few applications may need long SDUs, too. To be flexible the transport service should provide transmissions of short and long SDUs by a single service call.

Having considered the most important factors the main requirements for the transport layer are as follows: /1/ reliable two-way communication; /2/ connectionless transport service; /3/ segmentation; /4/ end-to-end flow control; /5/ efficient use of the lower layer service.

3. DESCRIPTION OF THE TRANSPORT PROTOCOL

The protocol is described in two forms. First, the most typical characteristics of the protocol mechanisms are discussed, then the protocol description is given by using a high-level protocol specification language.

Protocol data units /PDU/, sent from a single transport service accesss point /SAP/ are continuously and independently sequenced from the service data units /SDU/. The sequence-number field is 32 bits long to guarantee the individual identification of PDUs. The length of this field is determined by the following condition:

$$2^n = W * T$$

where "n" is the number of bits in the sequence-number field, W is a time interval during which a PDU and its acknowledgement can occur in the network, and T is the maximum transmission rate on a SAP. Having fixed the length of the sequence number field values of W and T can be varied in a wide range. For the proper operation of the protocol the value of W has to be bounded as discussed below.

Since the lower layer SDU can attain a length of maximum 546 bytes segmentation is used. As the PDU numbering is continuous the first

PDU of a SDU must be marked. It is indicated by a"start of SDU" flag in the PDU header. An"end of SDU" flag is also used instead of a SDU length field.

The flow control is based on the window mechanism and relates only to a single transport SDU. The sending part sends a PDU containing user data and waits for acknowledgement and credit /in number of PDUs/. Having received them the sending entity sends PDUs permitted by the received credit number. The credit can be modified during the communication.

The protocol provides a two-way communication between any pair of transport SAPs. Acknowledgement and credit information can be sent either in system PDUs having user data or in normal PDUs containing user data. If the PDU contains acknowledgement and credit information it is indicated by the "acknowledgement valid" flag in the header field.

There are special timer mechanisms to guarantee reliable transmission and to avoid deadlock situations. When the receiving entity receives a PDU that needs an acknowledgement, the entity must send it within an acknowledgement time interval /A/. If the sending entity does not receive the necessary acknowledgement in a the retransmission timeout /R = 2*t + A/ determined by the physical transmission delay /t/ and A, it will retransmit the PDU. This retransmission procedure can generate duplicated PDUs. For that very reason the time interval of all retransmissions must be bounded. This value is the maximum retransmission time /MR/. The time period during which the network can generate duplicates is the maximum packet lifetime /MPL/. If A, MR and MPL are bounded the timer values /see below/ can be determined exactly.

Timers are also used for detecting the inactivity of the partner entity. They are called as the receiving and sending inactivity timers /RI,SI/ /see Fig.1 and Fig.2/.

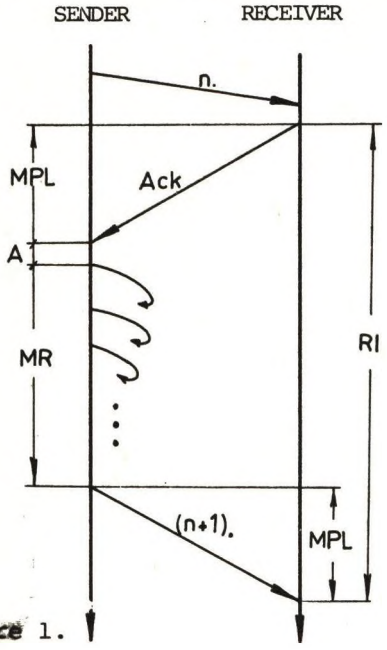


Figure 1.

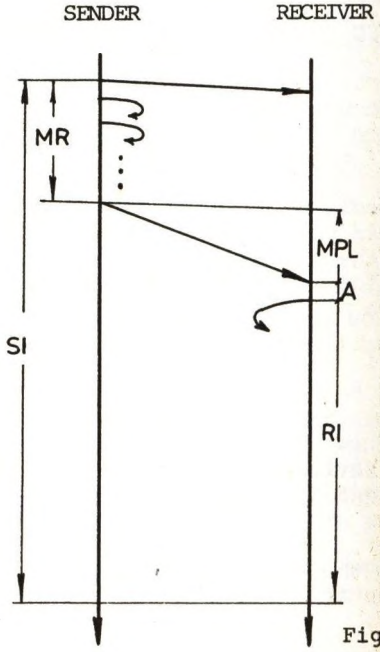


Figure 2.

The RI timer is started at any time when the receiver waits for the rest of SDU and is cancelled when a correct PDU arrives. Fig.1. shows the worst-case situation of the latest arrival of a PDU. The receiving entity sends an acknowledgement after the receipt of a PDU. This acknowledgement will arrive at the sending entity not later than MPL. The sending part starts the transmission of the next PDU within A. If all these transmissions but the last one are lost the PDU will arrive at the receiving part within /MR+MPL/. The bound for RI timer is:

$$RI = 2 * MPL + MR + A.$$

The SI timer is started by the first PDU sent and is cancelled when the acknowledgement arrives. The receiver should get this PDU not later than /MP+MPL/ and send a PDU within A. If this acknowledgement is lost both of the inactivity timers will expire. The SI timer, however, has to be expired later than the RI. In another case it would be possible that the sender is already sending a new SDU while the receiver is still trying to receive the rest of the earlier SDU. The equation for the SI timer is:

$$SI = MR + MPL + A + RI.$$

After the expiration of the SI timer the sender can use any initial sequence number. However, the problem of the initial sequence number choice still exists on recovery from a crash with a loss of memory. At the sending entity a worst-case situation occurs when the host crashes just after sending a PDU /Fig.3/.

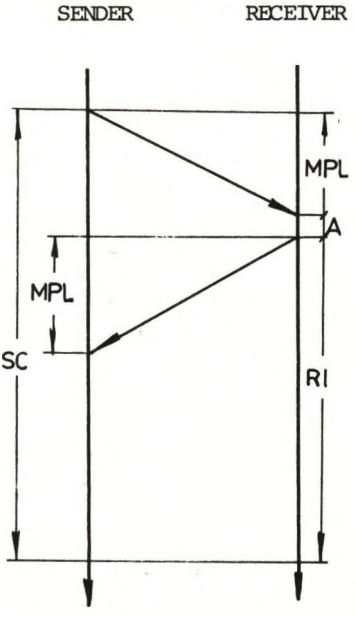


Figure 3.

In this case a new sending procedure can not be started till the PDU and its acknowledgement are in the network. This time interval is /2 * MPL + A/. The deadlock situation can be avoided /as it was mentioned at the explanation of the SI timer/ if a new transmission starts after the expiration of the RI timer. The conditions for the crash down timer /SC/ are:

$$SC > = RI + MPL + A.$$

If the host of the receiving entity crashes a PDU from the aborted SDU cannot arrive later than RI /as it was illustrated at the RI timer/. The value determined by SC is used as a unique crash down timeout. This timer is necessary because after the crash of the host it is impossible to determine whether sending or receiving procedure was in progress, and because the receiving crash down time is smaller than the sending one.

The receiving entity saves the highest sequence number of a correctly received SDU for a "sequence number saving" /SNS/ time Fig.4/.

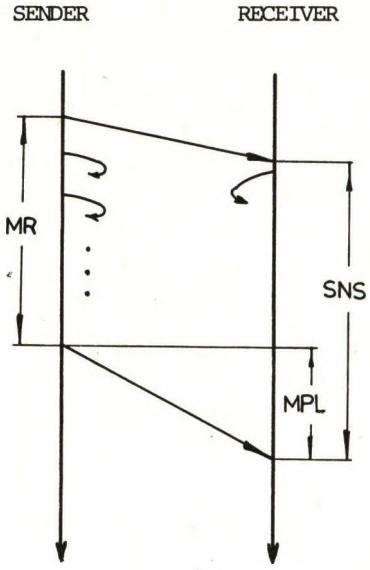


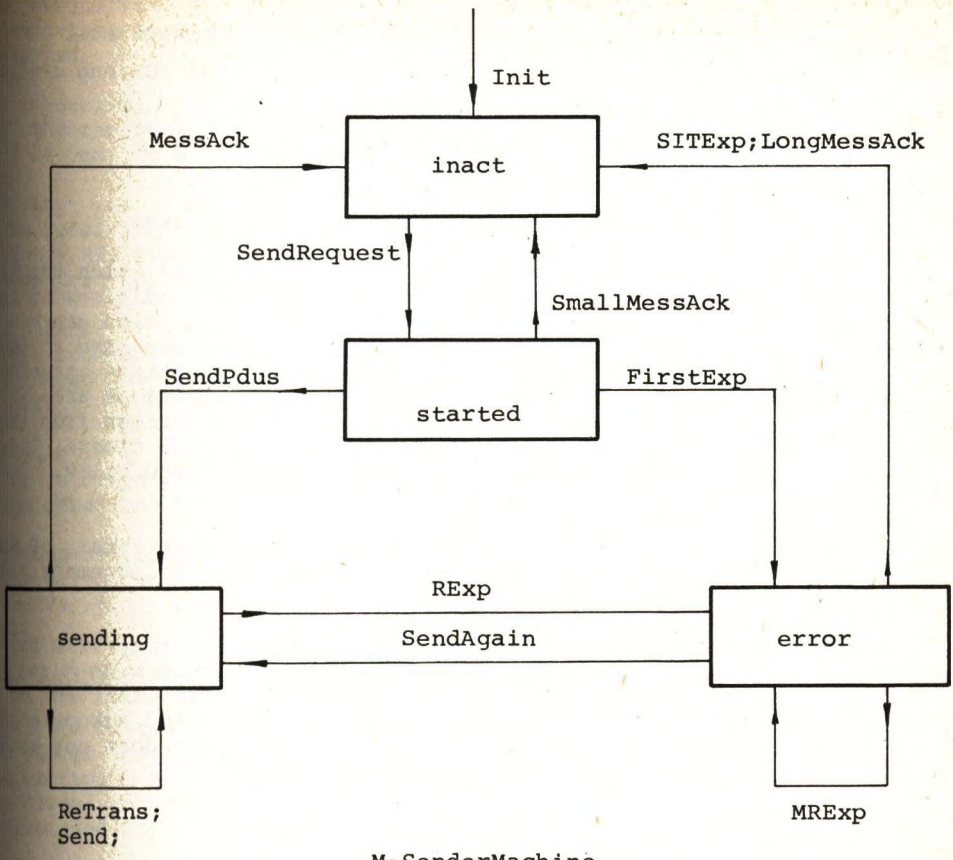
Figure 4.

The need for this timer arises from the closing problem. After receiving a complete SDU the receiving part sends an acknowledgement and gives an SDU to the user. If this acknowledgement is lost the sending part will feel that its final PDU does not arrive at the destination. In this case the sender will send system PDUs to ask for an acknowledgement. Its final request will arrive at the receiving entity not later than:

$$SNS = MR + MPL.$$

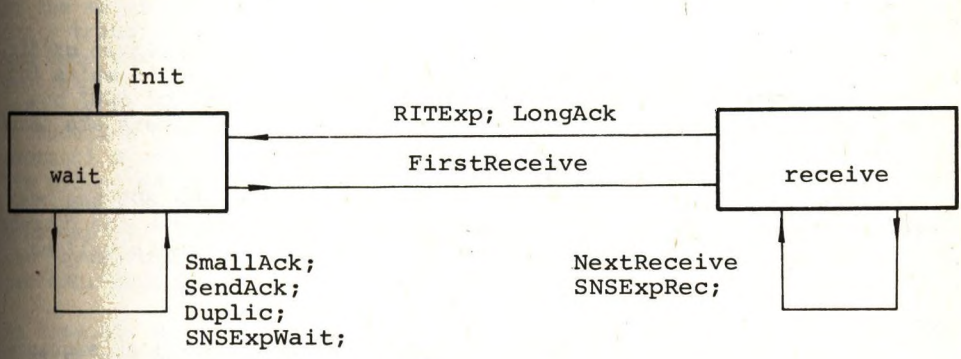
The receiving entity can send the required acknowledgement if it saves the highest received sequence number for SNS.

A more detailed and more formal description of the protocol is given in Appendix A by using a protocol description language. It was possible to demonstrate the specification in diagram form because the protocol is simple. Separate diagrams are given for the sending and receiving procedures /Fig.5 and Fig.6/.
These state diagrams compose an entity providing services for a single SAP. The names of the states and transitions of the diagrams correspont to the names used in the Appendix A.



M-SenderMachine

Figure 5.



M-ReceiverMachine

Figure 6.

The states in the diagrams illustrate the following situations:

inact: there is no outstanding send-request service call

started: the sending entity waits for the initial ACK and credit

error: retransmission state

sending: a normal sending procedure is in progress

wait: the entity does not receive any SDU

receive: a normal receiving procedure is in progress.

The transitions of the automaton are driven by events which usually defines the values of several parameters. Certain conditions of the parameters may have to be satisfied before the transition can be activated. During the transition new values may be assigned to the variables of the automaton and the automaton may initiate interface events [Blum 82] . The events which may cause transitions are: /1/ the user requests the sending of a SDU; /2/ a PDU arrives from the lower layer; /3/ the expiration of R, MR, SI, RI, SNS timers. The main actions of the transitions are shortly summarized below.

SendRequest:
The protocol entity sends the first PDU of a SDU with "start of SDU" and "send acknowledgement" flags on. Waits for acknowledgement and credit. Starts R,MR,SI timers.
MRExp,RExp,FirstExp:
If the MR timer has not expired yet, the protocol entity asks for acknowledgement and credit by sending a system packet with "send acknowledgement" flag on. The acknowledgement of the first PDU of a SDU must be asked by retransmitting the PDU with "send acknowledgement" flag on. After the expiration of MR the sender does not send any PDU, only it waits for acknowledgement and credit.

SendPdus,SendAgain:
A transmission of the rest of the SDU is started.

SITExp:
The entity gives up sending the SDU because the partner is inactive. The user is notified of this situation. The sequence number can take any value in the next PDU.

Send:
The protocol entity sends the PSUs allowed by the received credit, asks for acknowledgement and credit in the last PDU and waits for them. The last PDU of a SDU is marked with the "end of SDU" and "send acknowledgement" flags on. The SI, R and MR timers are started.

SmallMessAck,MessAck,LongMessAck:
A sending confirmation is given to indicate that the SDU has been sent correctly. The first PDU of the next SDU must get the current sequence number plus one. The possible remaining credit will be lost.

ReTrans:
The requested PDUs are retransmitted by setting the next sequence number to the value defined by the received acknowledgement.

SendAck:
An acknowledgement is sent for an earlier received SDU.

FirstReceive:
Acknowledgement and credit of the received PDU is sent. RI timer is started.

NextReceive:
If the received PDU has a correct sequence number it will be saved. If it has not retransmission is asked. The RI timer is started.

RITExp:
The receiving procedure of a SDU aborted, because of the expiration of the RI. From this time a PDU with any sequence number arriving from this source SAP is acceptable.

SmallAck:
An acknowledgement is sent and the received highest sequence number is saved for SNS time. A receipt indication will be given to the user.

Duplic:
The received PDU must be destroyed because it is a duplicated one. The receiving procedure will be continued in a normal way.

SNSExpWait,SNSExpRec:
The saved sequence number corresponding to a given destination is destroyed.

4. VERIFICATION OF THE TRANSPORT PROTOCOL

The aim of the verification of a protocol is to make sure that the protocol algorithm is correct, ie. there are no unforeseen protocol operations. We do this by analyzing the interactions between the protocol entities. The correct definition of verification was discussed in the technical report [Kov 82].

Our FDT dialect formal description language allowed us to verify the protocol with the help of a semi-automated protocol verification system. Our verification effort had two directions.

At first the "major state abstraction" method was used. On the basis of the major states of protocol entities and the potential valid state transitions, the possible global state transitions /Sender and Receiver machines/ were determined by the use of a computer program. The invalid global states and transitions were selected manually by investigating the FDT specification of the transport protocol. A computer program determined the "cycle atoms" of the protocol system in the knowledge of the global state transitions. The cycles of the protocol consist of these "cycle atoms" linked together. Manual examination of "cycle atoms" shed light on some problems of inconsistent interpretations. Appendix B contains the "cycle atoms" and the valid global state transitions of the protocol designed.

As Appendix B shows, there are one invalid global state and three invalid global state transitions. The invalid global state /inact, receive/ means that the sender /M-SenderMachine/ is already in the "inact" state while the receiver /M-ReceiverMachine/is still in the "receive" state /see Fig.5 and Fig.6/. The impossibility of this situation can be proved below by examining the mechanisms and the timer values of the protocol. The sender can reach the "inact" state in two ways: either by receiving an ACK for an SDU or by the

expiration of the inactivity timer /SI/. In the first case sending an ACK causes a "receive" →"wait" transition in the M-ReceiverMachine automaton. When the sender machine receives this ACK, the receiver will be allready in the "wait" state.
In the second case the receiver must also be already in the "wait" state at the expiration of SI, because the receiver's inactivity timer /RI/ is shorter than SI /see Fig.1 and Fig.2/. The impossibility of invalid global state transitions can also be proved in the same way. None of the global state cycles listed in Appendix B is useless loop and the termination of these cycles is guaranteed by the protocol mechanisms.

For the total verification of the transport protocol a complete protocol model is being built consisting of the description of protocol entities and the specification of the most important features of the lower protocol layers. Lost, duplicated and missequenced packets is permitted in the lower layers. The FDT description of the complete model will be transformed into the input language of the semi-automated protocol verification system developed by one of the authors [Kov 82]. Finally, the dynamical operation of the transport protocol model will be examined by considering the actions of the upper layer.

The verification system is based on an improved reachability analysis [Kov 84]. This method checks the invariance of certain predicates in the reachability graph. There are predicates built into the verification system which determine the deadlock situations, completeness and liveness properties of the protocol. The verification system determines the protocol cycles. The examination of these cycles in order to pick up the useless loops /progress property/ requires the knowledge of the concrete algorithm. Other user-defined protocol properties formulated in the form of predicate invariance can also be examined with the verification system. By the help of this interactive software tool /verification system/ the progress of the protocol operation was easily traced.

5. IMPLEMENTATION AND TESTING

The requirements imposed by the implementation method of the architecture strongly influenced the implementation of the protocol: /1/ the protocol had to be implemented as part of a user process of a UNIX-like operating system; /2/ the service implementation had to be based on the function calls and the UNIX signaling mechanism could be used for synchronization.

The largest part of the architecture has been implemented in a so-called socket process. There is a socket process for each socket in use. The consequences of the socket process conception for the transport protocol are the following: /1/ the transport entity will receive a SDU for the user if he requests it; /2/ the communication of the SAPs are independently handled; /3/ there can be only a half-duplex communication between any two SAPs.

The implementation minimizes the user data transfer in the main memory. A correctly received PDU is directly placed into a buffer allocated by the user.

The implementation model is shown in Fig.7. It illustrates the main functional code modules and the flow of information.

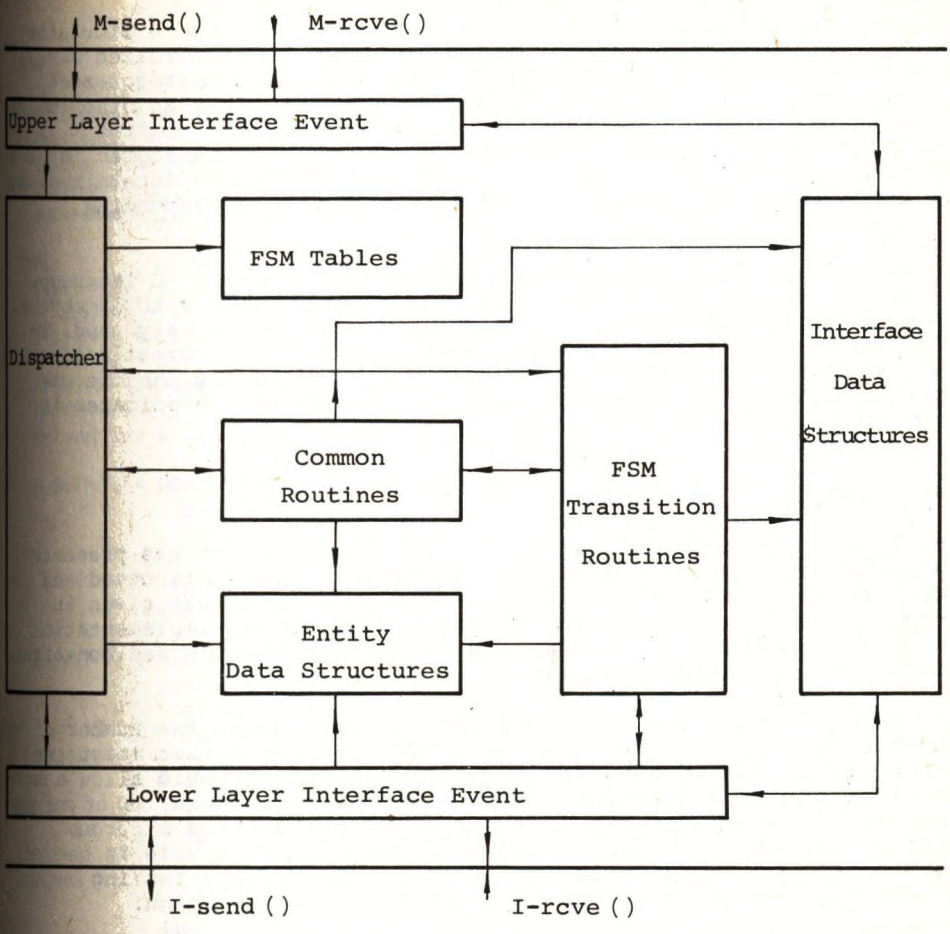


Figure 7.

A module is a set of functions completing the same logical task. The transport interface primitives are received and controlled by the Upper-Layer Interface Event module. The necessary primitive parameters are saved into the interface event data structures, then the Dispatcher module is called. In this module there is a non-terminating cycle operation. The Dispatcher module controls both the sending and receiving procedures. The module uses the FSM tables for defining the correct transition routine. A transition routine can modify the entity data structures, can send PDUs or can ask the dispatcher to receive a PDU from the lower layer.

The Lower Layer Interface Event routine composes the parameters to send or receive PDUs. If a PDU arrives at the lower layer and no request is received, a signal will be sent. The handling of this signal will cause a receive request for the lower layer. This signal mechanism is necessary to implement the acknowledgement sending functions of the protocol.

The Common Routine module contains functions used frequently by all other modules /eg. address controlling, buffer manipulation etc./.

The testing of the implementation is based on a Network Simulator module. The simulator causes network errors in a controlled way. It is able to lose PDUs or to generate duplicated and missequenced packets. The operation of the protocol was traced by an other module which saves all the necessary state, transition and control variables as well as time information. By using this stored information the cause of malfunction was easily and quickly determined, and the programming errors or the functions of the corresponding routines were corrected.

The size of the implementation is about 4ooo lines in C language. The object code together with data structures amounts to 15 kbytes. The retransmission capability of the protocol was rarely used. In every recovery situation the protocol carried out correct actions. The program in the Dispatcher module was much langer than it was expected because the determination of the enabling predicates is very complex.

6. CONCLUSION

A transport protocol operating in a LAN architecture was presented. The architecture environment of the protocol were discussed and widely considered. The description of the protocol was given in formal and informal ways. The chosen verification, implementation and testing methods are based on a formal description and constitute a consistent planning procedure.

The value of the protocol header credit field gives the number of the sendable PDUs. Our implementation experience shows that measuring the credit in bit or byte units instead of frames would allow a more efficient buffer management. Despite the correct operation of our implementation in a long period of time a more exact and rigorous verification based on the extended reachability analysis is needed. Another step of our work is to develop a conformance testing method for checking the correct operation of the implementation.

REFERENCES

[Blum 82] Th.P.Blumer, R.L.Tenney:
"A Formal Specification Technique and Implementation Method for Protocols"
Computer Networks 6 /1982/ 2ol-217.

[Boch 83] G.V.Bochmann: "Formal Description Techniques for OSI: an Example"
Department d'informatique et de recherche operationelle
Universite de Montreal, publication 492, November 1983

[FDT 81] "An FDT based on an extended state transition model"
Subgroup B of ad hoc group on FDT /ISO TC/97/SC16/WG1/
Working Draft,Boston Dec. 1981.

[Kov 82] L.Kovacs: "Formal Specification and Verification of Computer Network Protocols", Technical Report,
MTA-SZTAKI,Tanulmányok 138/1982. /In Hungarian/

[Kov 84] L.Kovacs:"Automated Protocol Verification" Third International Meeting of Young Computer Scientist, Smolenice 1984.

APPENDIX A

```
CONST
  retranstime = ... ;
  maxretranstime = ... ;
  recinacttime = ... ;
  sendinactime = ... ;
  snstime = ... ;
  maxseqnum = ... ;

TYPE
  DataType = ... ;
  M-DataType = ... ;
  M-QosType = ... ;
  I-QosType = ... ;
  ResultType = (ok,dest-inactiv);
  AddrType = ... ;
  M-TimerType = (R,MR,SIT,RIT,SNS);

  M-SduType = RECORD
                da, sa : AddrType;
                res : ResultType;
                qos : M-QosType;
                data : M-DataType;
              END;

  M-PduType = RECORD
                syspdu : BOOLEAN;
                sendack : BOOLEAN;
                firstpdu : BOOLEAN;
                lastpdu : BOOLEAN;
                ackvalid : BOOLEAN;
                typ : M-QosType;
                seqnum : LONGCARDINAL;
                ack : LONGCARDINAL;
                credit : CARDINAL;
                data : DataType;
              END;

  I-SduType = RECORD
                da, sa : AddrType;
                qos : I-QosType;
                data : M-PduType;
              END;

CHANNEL M-AccessPoint(User,Provider);
  BY User:
      UnitDataRequest(msdu:M-SduType);
  BY Provider:
      UnitDataIndication(msdu:M-SduType);
      UnitDataConfirmation(msdu:M-SduType);

CHANNEL I-AccessPoint(User,Provider);
  BY User:
      UnitDataRequest(isdu:I-SduType);
  BY Provider:
```

```
        UnitDataIndication(isdu:I-SduType);

CHANNEL S-AccessPoint(User,Provider);
  BY User:
       TimerRequest(name:M-TimerType; time:CARDINAL);
       TimerCancel(name:M-TimerType);
  BY Provider:
       TimerIndication(name:M-TimerType);

MODULE M-SenderMachine( M : M-AccessPoint(Provider) commonqueue;
                        I : I-AccessPoint(User) commonqueue;
                        S : S-AccessPoint(User) individualqueue);

VAR  state:(inact, started, sending, error);
     nextseqnum, firstseqnum : LONGCARDINAL;
     actcredit, messlength, pduseqnum : CARDINAL;

PROCEDURE Init;
BEGIN
  state:=inact;
  nextseqnum:=0;
  pduseqnum:=1;
END Init;

PROCEDURE ModuloIncrement(VAR l:LONGCARDINAL);
PRIMITIVE;

PROCEDURE FirstSendPdu;
  VAR msdu:M-SduType; pdu:M-PduType; isdu:I-SduType;
BEGIN
  GetBuffer(msdu);
  messlength := MessLength(msdu.data);
  pduseqnum := 1;
  WITH pdu DO
    syspdu := FALSE;
    sendack := TRUE;
    firstpdu := TRUE;
    lastpdu := (messlength=1);
    ackvalid := FALSE;
    typ := msdu.qos;
    seqnum := nextseqnum;
    data := Extract(msdu.data,pduseqnum);
  END;
  firstseqnum:=nextseqnum;
  ModuloIncrement(nextseqnum);
  Increment(pduseqnum);
  WITH isdu DO
    da := msdu.da;
    sa := msdu.sa;
    data := pdu;
  END;
  OUT I.UnitDataRequest(isdu);
END FirstSendPdu;

PROCEDURE BufferSufficient(msdu:M-SduType):BOOLEAN;
PRIMITIVE;

PROCEDURE PutBuffer(msdu:M-SduType);
PRIMITIVE;

PROCEDURE GetBuffer(VAR msdu:M-SduType);
PRIMITIVE;
```

```
PROCEDURE ReleaseBuffer;
PRIMITIVE;

FUNCTION CorrectCall(msdu:M-SduType):BOOLEAN;
PRIMITIVE;

FUNCTION CorrectSdu(isdu:I-SduType):BOOLEAN;
PRIMITIVE;

FUNCTION Extract(data:M-DataType; pduix:CARDINAL):DataType;
PRIMITIVE;

FUNCTION MessLength(data:M-DataType):CARDINAL;
PRIMITIVE;

PROCEDURE MessageAck(yes:BOOLEAN);
  VAR msdu:M-SduType;
BEGIN
  GetBuffer(msdu);
  WITH msdu DO
    IF yes THEN
      res := ok;
    ELSE
      res := dest-inactiv;
    END;
  END;
  firstseqnum := nextseqnum;
  pduseqnum := 1;
  messlength := 0;
  OUT M.UnitDataConfirmation(msdu);
  OUT S.TimerCancel(R);
  OUT S.TimerCancel(MR);
  OUT S.TimerCancel(SIT);
  ReleaseBuffer;
END MessageAck;

PROCEDURE AskAckAndCredit;
  VAR msdu:M-SduType; isdu:I-SduType;
BEGIN
  GetBuffer(msdu);
  WITH isdu DO
    da := msdu.da;
    sa := msdu.sa;
    data.syspdu := TRUE;
    data.sendack := TRUE;
  END;
  OUT I.UnitDataRequest(isdu);
END AskAckAndCredit;

SendRequest IS TRANSITION FROM inact TO started;
WHEN M.UnitDataRequest;
PROVIDED CorrectCall(msdu) AND BufferSufficient(msdu);
BEGIN
  PutBuffer(msdu);
  FirstSendPdu;
  OUT S.TimerRequest(R,retranstime);
  OUT S.TimerRequest(MR,maxretranstime);
  OUT S.TimerRequest(SIT,sendinactime);
END SendRequest;

SmallMessAck IS TRANSITION FROM started TO inact;
WHEN I.UnitDataIndication;
PROVIDED CorrectSdu(isdu) AND (isdu.data.ackvalid) AND
```

```
  (isdu.data.ack = nextseqnum) AND (messlength = 1);
BEGIN
  MessageAck(TRUE);
END SmallMessAck;

SendPdus IS TRANSITION FROM started TO sending;
WHEN  I.UnitDataIndication;
PROVIDED CorrectSdu(isdu) AND (isdu.data.credit > 0) AND
 (messlength > 1) AND (isdu.data.ackvalid) AND (isdu.data.ack = nextseqnum);
BEGIN
  actcredit := isdu.data.credit;
END SendPdus;

ReTrans IS TRANSITION FROM sending TO sending;
WHEN I.UnitDataIndication;
PROVIDED CorrectSdu(isdu) AND (isdu.data.ackvalid) AND (isdu.data.credit)0);
VAR a, saveack:CARDINAL;
BEGIN
  a := isdu.data.ack;
  saveack := isdu.data.ack;
  IF a <> nextseqnum THEN
    WHILE a <> nextseqnum DO
      Decrement(pduseqnum);
      ModuloIncrement(a);
    END;
    nextseqnum := saveack;
  END;
  actcredit := isdu.data.credit;
END ReTrans;

Send IS TRANSITION FROM sending TO sending;
NO DELAYS;
PROVIDED (actcredit > 0) AND (messlength >= pduseqnum);
  VAR msdu:M-SduType; pdu:M-PduType; i:I-SduType;
BEGIN
  GetBuffer(msdu);
  WITH i DO
    da := msdu.da;
    sa := msdu.sa;
  END;
  WITH pdu DO
    syspdu := FALSE;
    firstpdu := FALSE;,
    lastpdu := (pduseqnum = messlength);
    typ := msdu.qos;
    seqnum := nextseqnum;
      ModuloIncrement(nextseqnum);
    data := Extract(msdu.data,pduseqnum);
    IF (actcredit=1) OR (messlength=pduseqnum) THEN
      sendack := TRUE;
      i.data := pdu;
      OUT S.TimerCancel(R);
      OUT S.TimerCancel(MR);
      OUT S.TimerCancel(SIT);
      OUT S.TimerRequest(R,retranstime);
      OUT S.TimerRequest(MR,maxretranstime);
      OUT S.TimerRequest(SIT,sendinactime);
    ELSE
      sendack := FALSE;
      i.data := pdu;
    END;
    Decrement(actcredit);
    Increment(pduseqnum);
```

```
   OUT I.UnitDataRequest(i);
  END; (* with *)
END Send;

MessAck IS TRANSITION  FROM sending TO inact;
WHEN I.UnitDataIndication;
PROVIDED CorrectSdu(isdu) AND (isdu.data.ackvalid) AND
  (isdu.data.ack = nextseqnum) AND (pduseqnum > messlength);
BEGIN
  MessageAck(TRUE);
END MessAck;

RExp IS TRANSITION FROM sending TO error;
WHEN S.TimerIndication;
BEGIN
  IF name = R THEN AskAckAndCredit; END;
END RExp;

MRExp IS TRANSITION FROM error TO error;
WHEN S.TimerIndication;
PROVIDED (name <> SIT);
BEGIN
  OUT S.TimerCancel(R);
  IF name = R THEN
    AskAckAndCredit;
    OUT S.TimerRequest(R,retranstime);
  END;
END MRExp;

SITExp IS TRANSITION FROM error TO inact;
WHEN S.TimerIndication;
PROVIDED (name = SIT);
BEGIN
  MessageAck(FALSE);
END SITExp;

LongMessAck IS TRANSITION FROM error TO inact;
WHEN I.UnitDataIndication;
PROVIDED CorrectSdu(isdu) AND (isdu.data.ackvalid) AND
  (isdu.data.ack = nextseqnum) AND (pduseqnum > messlength);
BEGIN
  MessageAck(TRUE);
END LongMessAck;

FirstRExp IS TRANSITION FROM started TO error;
WHEN S.TimerIndication;
BEGIN
  IF name = R THEN AskAckAndCredit; END;
END FirstRExp;

SendAgain IS TRANSITION FROM error TO sending;
WHEN I.UnitDataIndication;
PROVIDED CorrectSdu(isdu) AND (isdu.data.ackvalid) AND (isdu.data.credit)0);
VAR a, saveack:CARDINAL;
BEGIN
  a := isdu.data.ack;
  saveack := isdu.data.ack;
  IF a <> nextseqnum THEN
    WHILE a <> nextseqnum DO
      Decrement(pduseqnum);
      ModuloIncrement(a);
    END;
    nextseqnum := saveack;
```

```
    END;
    actcredit := isdu.data.credit;
  END SendAgain;

  BEGIN
    Init;
  END M-SenderMachine;


  MODULE M-ReceiverMachine( M : M-AccessPoint(Provider) commonqueue;
                            I : I-AccessPoint(User) commonqueue;
                            S : S-AccessPoint(User) individualqueue);

  VAR  state : (wait, receiving);
       nextseqnum, firstseqnum: LONGCARDINAL;
       pduseqnum : CARDINAL;

  PROCEDURE Init;
  BEGIN
    state := wait;
    nextseqnum := 0;
    pduseqnum := 1;
  END Init;

  PROCEDURE FirstReceivePdu(isdu:I-SduType);
    VAR msdu:M-SduType; pdu:M-PduType; i:I-SduType;
  BEGIN
    WITH msdu DO
      da := isdu.da;
      sa := isdu.sa;
      qos := isdu.data.typ;
    END;
    InitBuffer(msdu);
    pduseqnum := 1;
    Fill(isdu.data.data,pduseqnum);
    Increment(pduseqnum);
    nextseqnum := isdu.data.seqnum;
    firstseqnum := isdu.data.seqnum;
    ModuloIncrement(nextseqnum);
    WITH pdu DO
      syspdu := TRUE;
      sendack := FALSE;
      ackvalid := TRUE;'
      ack := nextseqnum;
      credit := GetActCredit();
    END;
    WITH i DO
      da := isdu.sa;
      sa := isdu.da;
      data := pdu;
    END;
    OUT I.UnitDataRequest(i);
    OUT S.TimerRequest(RIT,recinacttime);
  END FirstReceivePdu;

  PROCEDURE SendAckAndCredit;
    VAR msdu:M-SduType; pdu:M-PduType; isdu:I-SduType;
  BEGIN
    WITH pdu DO
      syspdu := TRUE;
      sendack := FALSE;
      ackvalid := TRUE;
      ack := nextseqnum;
```

```
    credit := GetActCredit();
  END;
  GetBuffer(msdu);
  WITH isdu DO
    da := msdu.sa;
    sa := msdu.da;
    data := pdu;
  END;
  OUT I.UnitDataRequest(isdu);
END SendAckAndCredit;

PROCEDURE ModuloIncrement(VAR l:LONGCARDINAL);
PRIMITIVE;

PROCEDURE InitBuffer(msdu:M-SduType);
PRIMITIVE;

PROCEDURE GetBuffer(VAR msdu:M-SduType)
PRIMITIVE;

FUNCTION GetActCredit():CARDINAL;
PRIMITIVE;

PROCEDURE Fill(content:DataType; pduix:CARDINAL);
PRIMITIVE;

FUNICTION GoodSdu(isdu:I-SduType):BOOLEAN;
PRIMITIVE;

PROCEDURE SaveAddrAndSeqnum(a:AddrType; seq:LONGCARDINAL);
PRIMITIVE;

PROCEDURE UnSave;
PRIMITIVE;

FUNCTION GetSavedSeqnum(a:AddrType):LONGCARDINAL;
PRIMITIVE;

FUNCTION Locked(a:AddrType; seq:LONGCARDINAL):BOOLEAN;
PRIMITIVE;

FUNCTION IsSaved(a:AddrType):BOOLEAN;
PRIMITIVE;

PROCEDURE SendOldAck(saaddr,daaddr:AddType; seq:LONGCARDINAL);
  VAR pdu:M-PduType; isdu:I-SduType;
BEGIN
  WITH pdu DO
    syspdu := TRUE;
    sendack := FALSE;
    ackvalid := seq;
    credit := 0;
  END;
  WITH isdu DO
    da := saaddr;
    sa := daaddr;
    data := pdu;
  END;
  OUT I.UnitDataRequest(isdu);
END SendOldAck;


SmallAck IS TRANSITION FROM wait TO wait;
```

```
WHEN I.UnitDataIndication;
PROVIDED GoodSdu(isdu) AND (NOT isdu.data.syspdu) AND
  (isdu.data.sendack) AND (isdu.data.firstpdu) AND (isdu.data.lastpdu);
VAR msdu:M-SduType;
BEGIN
  SaveAddrAndSeqnum(isdu.sa,nextseqnum);
  FirstReceivePdu(isdu);
  GetBuffer(msdu);
  WITH msdu DO
    res := ok;
    qos := isdu.data.typ;
  END;
  OUT M.UnitDataIndication((msdu);
  OUT S.TimerRequest(SNS,snstime);
  ReleaseBuffer;
END SmallAck;

SendAck IS TRANSITION FROM wait TO wait;
WHEN I.UnitDataRequest;
PROVIDED GoodSdu(isdu) AND (isdu.data.syspdu) AND
  (isdu.data.sendack) AND IsSaved(isdu.sa);
VAR s:LONGCARDINAL;
BEGIN
  s := GetSavedSeqnum(isdu.sa);
  SendOldAck(isdu.sa,isdu.da,s);
END SendAck;

FirstReceive IS TRANSITION FROM wait TO receiving;
WHEN I.UnitDataIndication;
PROVIDED GoodSdu(isdu) AND (NOT isdu.data.syspdu) AND (isdu.data.sendack) AND
  (isdu.data.firstpdu) AND (NOT isdu.data.lastpdu);
BEGIN
  FirstReceivePdu(isdu);
  S.TimerRequest(RIT,recinacttime);
END FirstReceive;

RITExp IS TRANSITION FROM receiving TO wait;
WHEN S.TimerRequest;
PROVIDED (name = RIT);
BEGIN
  ReleaseBuffer;
END RITExp;

NextReceive IS TRANSITION FROM receiving TO receiving;
WHEN I.UnitDataIndication;
PROVIDED GoodSdu(isdu) AND ((isdu.data.syspdu) OR
   (NOT ((isdu.data.seqnum = nextseqnum) AND (isdu.data.lastpdu))));
VAR duplic:BOOLEAN; c:CARDINAL;
BEGIN
  IF (isdu.data.syspdu) AND (isdu.data.sendack) THEN
    SendAckAndCredit;
  ELSIF (NOT isdu.data.syspdu) THEN
    IF (isdu.data.seqnum = nextseqnum) THEN
      Fill(isdu.data.data,pduseqnum);
      Increment(pduseqnum);
      ModuloIncrement(nextseqnum);
      OUT S.TimerRequest(RIT,recinacttime);
      IF (isdu.data.sendack) THEN SendAckAndCredit; END;
    ELSE
      duplic := FALSE;
      c := firstseqnum;
      WHILE (c<>nextseqnum) AND (NOT duplic) DO
        duplic := (c=isdu.data.seqnum);
```

```
      ModuloIncrement(c);
    END;
    OUT S.TimerRequest(RIT,recinacttime);
    IF NOT duplic THEN SendAckAndCredit; END;
  END;
 END;
END NextReceive;

LongAck IS TRANSITION FROM receiving TO wait;
WHEN I.UnitDataIndication;
PROVIDED GoodSdu(isdu) AND (NOT isdu.data.syspdu) AND
 (isdu.data.seqnum = nextseqnum) AND  (isdu.data.lastpdu);
VAR msdu:M-SduType;
BEGIN
 SaveAddrAndSeqnum(isdu.data,nextseqnum);
 Fill(isdu.data.data.pduseqnum);
 Increment(pduseqnum);
 ModuloIncrement(nextseqnum);
 SendAckAndCredit;
 GetBuffer(msdu);
 WITH msdu DO
  res := ok;
  qos := idsu.data.typ;
 END;
 OUT M.UnitDataIndication(msdu);
 OUT S.TimerCancel(RIT);
 OUT S.TimerRequest(SNS,snstime);
 ReleaseBuffer;
END LongAck;

Duplic IS TRANSITION FORM wait TO wait;
WHEN I.UnitDataIndication;
PROVIDED GoodSdu(isdu) AND (NOT isdu.data.syspdu) AND
 (NOT Locked(isdu.sa,isdu.data,seqnum));
BEGIN (* empty *)
END Duplic;

SNSExpWait IS TRANSITION FROM wait TO wait;
WHEN S.TimerIndication;
PROVIDED (name = SNS);
BEGIN
 UnSave;
END SNSExpWait;

SNSExpRec IS TRANSITION FROM receiving TO receiving;
WHEN S.TimerIndication;
PROVIDED (name = SNS);
BEGIN
 UnSave;
END SNSExpRec;

BEGIN
 Init;
END M-ReceiverMachine;
```

The aim of these investigations should be practicable methods which have a chance for high user acceptance.

Today, it is evident that the protocol development process has to be aided by computers. For this purpose an unified approach of protocol specification, verification and implementation seems to be useful.

In this paper, a possible solution is presented with the PDL-system. In the following chapter the design criteria and the structure of the system are explained. The other chapters deal with the main components of the system - the specification language, the verification approaches and the graphical means proposed.

2. AIM AND STRUCTURE OF THE PDL-SYSTEM

The PDL-system combines several approaches to utilize their advantages during the different development phases.

The kernel of the system is the special-purpose language PDL (Protocol Description Language)/Koenig 85a/. The choice of a language-oriented specification approach bases on the following considerations:

- Protocols are a special form of algorithms which can be exactly described by high-level programming languages /Bochmann 80a/.
- Many situations can suitably described by linguistic means, only. (This fact led to the hybrid methods.)
- The implementation can be logically prepared. This is important with regard to computer-aided protocol development.
- The numerous results of systematic software design can be exploited.

The danger of the application of programming languages lies in a possible overload of the specification with language-dependent details or a too much implementation-oriented representation, respectively. However, common high-level programming languages are not primarily designed for protocol specification and therefore can not satisfy all its demands.

With PDL the attempt was made to create suitable language features for important notions of the OSI-protocol mechanism such as layer, entity, protocol, service, and others.

PDL supports a protocol development by stepwise refinement introducing two language levels - PDL/D and PDL/I. PDL/D (D = Design) is to specify protocols for first design, while PDL/I (I = Implementation) is applied in the implementation-preparing phase.

PDL/D concentrates on the description of the virtual communications between peer entities. The realization of the communication by underlying layers is partly considered, only.

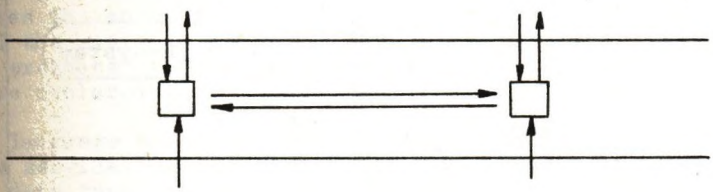


Figure 1: PDL/D

PDL/I describes the communications between peer entities including the services of the underlying layer.

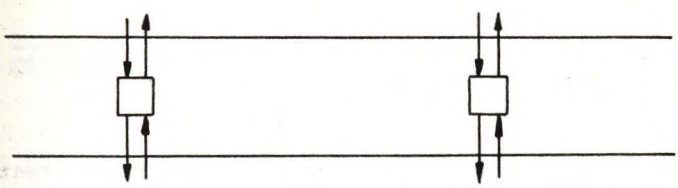


Figure 2: PDL/I

For the transformation from the PDL/D- into the PDL/I-specification a syntax-based editor is supported.

The verification approach of the PDL-system uses the analysing possibilities offered by the petri net theory. For this purpose, both language specifications are automatically transformed into a petri net which is analysed and simulated. Thus, errors in the logical design can be detected.

Then possible implementation frames can be generated from the PDL/I-specification. This implementation proposal must be supplemented, if necessary. The extent to do this depends on the completeness of the PDL/I-specificaton and the implementation variant chosen.

As a supplement to the language description the PDL-system supports graphical means (e.g. state transition diagrams) for users prefering these during the design phase and for documentation.

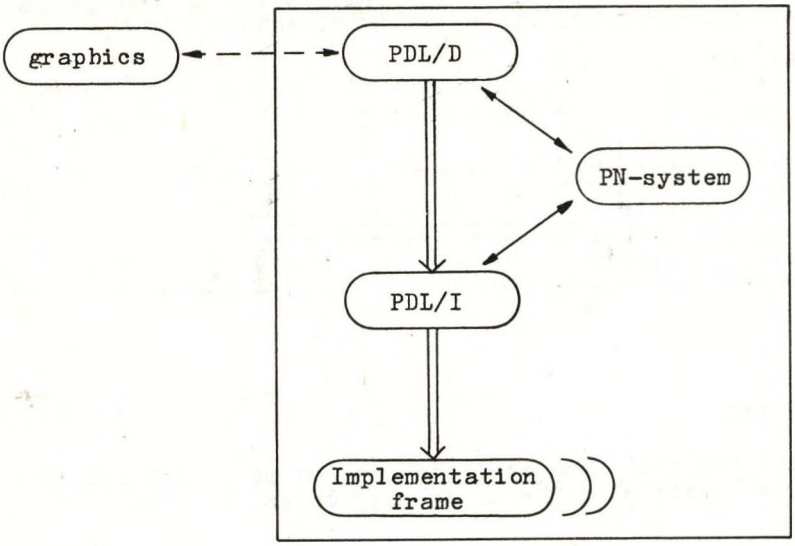


Figure 3: A survey of the PDL-system

3. THE SPECIFICATION LANGUAGE PDL

PDL proposes special linguistic means for some important notions of the OSI-reference model using modern concepts originating from high-level programming languages. Simultanuously, a language-oriented model of the OSI-protocol mechanism is presented by PDL.

Protocol specification in PDL is a protocol flow and not a state transition description. The latter (see e.g. /Bolt 80/) generally describes the behaviour of entities by means of finite state automata. The description obtained does not always reflect the sequence of communication. (There are some similarities to the so-called goto-programming style).

On the contrary, PDL enforces a sequential (goto-free) description of the interaction between peer entities. We believe that this protocol flow description approach is a more natural and adequate one. (It is possible to derive automatically the state transition diagram from the PDL-specification (see chapter 5), but not vice versa.)

Another design aim of PDL is the creation of simple and familiar linguistic means which reflect the nature of protocols without overloading them with implementation-dependent details and complicated language rules. Furthermore, a complete, unambiguous, and readable protocol description should be guaranteed.

3.1. Basic Elements of PDL

The PDL-specification consists of the layer-specification followed by so-called protocol parts. The layer describes a layer or a sublayer in the sense of the OSI-reference model. It includes the specification of the services presented by the layer and the entities which belong to it. Besides, the messages exchanged between the entities and the possible protocols are declared (see Example 1).

From the language point of view the layer is an abstract data type. The services correspond to the functions presented by an abstract data type, while their "implementation" is realized by the communicating entities /Koenig 85b/.

The service-specification distinguishes between service primitivs called from the user of the layer (service asked) and those initiated within the layer to inform the user about certain events (service responsed).The parameters of the service primitives are specified by their data types.

Example 1: [1]

```
layer AB;
  service asked AB-SEND request(data: data block);
          responsed AB-RECEIVE indication;

                  . . .

  message PDU: record(data: data block;control bit: bit);

                  . . .

  protocol Alternating Bit Protocol: send<=>receive;
  entity AB-E;

      . . .

  #entity {AB-E}
#layer {AB}
```

The entity-specification, which can be also interpreted as a special form of an abstract data type, describes the structure and the behaviour of an entity. (Note, there is only one specification for each type of entity in the layer.) In the interface specification of the entity the services, messages and protocol parts connected with the entity are specified. Besides, there is also the possibility to declare (shared) variables and timers which can be used in all protocol parts of the entity. The be-

1) The examples presented in this chapter are extracted from a description of the Alternating Bit Protocol in the appendix.

haviour of the entity is described by the par event-statement. This statement specifies the reaction of the entity to certain events. Such events could be a service primitive call, the reception of a message (<--), an information from the communication channel (signal), a timeout, and an exception state (error) The ocurence of such an event evokes the assigned actions. Usually, this is the call of a protocol part, but local actions (e.g. assignments to variables) are allowed, too. The entity actions are handled as parallel processes, if the events occur simultaneously or during the reaction to another event. If there is no event, the entity waits until one will occur.

Example 2:

```
entity AB-E;
   service AB-SEND request;
   message PDU;
   protocol part send,receive;
   par event
     AB-SEND request: send |
     PDU <-- AB-E: receive
   #par
#entity {AB-E}
```

The protocol parts which are separated from the layer-specification for better readability and modularization describe the communication of the entity from its point of view. The protocol parts which form a protocol are assigned to each other in the protocol-specification part of the layer. As communication partner a fictive entity is introduced in the protocol part header. The actual entity is assigned in the protocol part call. (It can be omitted, if there is only one entity-specification in the layer.)

Similar to procedures in some programming languages, a protocol part begins with an interface specification and a declaration part which fulfil the same function like in the entity-specification. The following action part describes the communication of the entity by means of protocol statements (see Example 3).

The most important protocol statements are the send- (-->) and the wait event-statement.

The wait event-statement like the par event-statement is a modified form of the guarded regions. It describes the waiting for and the reaction to certain events. In opposite to the par event statement the wait event-statement reacts only to one event. This means a choice between events and their sequential processing, if necessary.

Furthermore, there are statements for starting and resetting of timers, for assignments to data structures, a statement for the call of a responsed service primitive (response), and others.

For the description of the protocol flow one can use the well-known control structures of high-level programming languages like the if-, case-, and the loop-statement.

There is also the possibility to describe certain protocol actions informally by means of verbal constructions.

Example 3:

```
protocol part send(entity receiver);
service AB-SEND indication;
message PDU to receiver,
        ACK from receiver;
timer t: 0..60;
begin
   .  .  .

   PDU --> receiver;
   start t;
   wait event
      ACK <-- receiver: .  .  .

                        response AB-SEND indication;
                        .  .  .                    |
      timeout t: skip
   #wait;
   .  .  .

end {send}
```

3.2. PDL/D

In PDL/D the virtual communication is described by means of the send-statement, the receiving- and signal-event already presented above.

In order to support the design aspect of the specification process and to free the user in a wide range from language and implementation dependent details, PDL/D contains some descriptive features, e.g. there are no language means to form arithmetical and logical expressions. For this purpose verbal constructions have to be used. Moreover, an axiomatic introduction of arbitrary elementary data types is permitted. The indices of data structures are optionally.

3.3 PDL/I

In PDL/I the virtual communication is substituted by adaption to the lower layer. Therefore the send-statement, the receiving and

the signal-event must be traced back to lower layer services. For this reason an interface description to the underlying layer must be included in the entity-specification.

For preparing the implementation without prescribing a concrete one the descriptive features of PDL/D must be refined by introducing expressions, indices, and definitions of data types.

PDL/D and PDL/I differ only in those features which result on the mentioned differences in the specification. The other language elements, about two-thirds, are identical.

4. PETRI NET-BASED VERIFICATION

The necessity of verification is well-established in literature (see e.g. /Bochmann 80a/). Debugging of computer network protocols only by means of systematic run-time tests is not possible because some erroneous behaviour, such as synchronization errors, are controlled by the current progress of the processes which is time-dependent and non-reproducable, in general.

In our opinion, the notion verification should comprise all methods for a design-accompanying analysis of behaviour at first. What it actually means, should be decided later according to the criterion of practicability.

4.1. Classification and Delimitation

First of all, two types of verification have strongly to be distinguished:

(G) GENERAL verification of properties which must be followed independent of the special semantics of the system being designed, such as liveness, deadlock freedom, boundedness, reinitialization, and others. These tests of whether a specification is free of contradictions are based on context considerations by means of mathematical models.

(S) SPECIAL verification of properties determined by the semantics of the system being designed. This verification, in his classical meaning, is founded in a comparison of a specification with an (further) external description.

For type (G) some mathematical approaches are known (reachability graph of a state transition or petri net model, structure theoremes on petri nets) leading to unique decisions by algorithms. In the contrary, only assistant decisions means are available for type (S), in general, which have to be brought into line with the external description in an interactive manner.

For this reason, the PDL-system concentrates only upon type (G) at first. An evaluation of S- and T-invariants obtained by demand for type (S) by users is possible, but not supported any further.

As a next step a more rigorous verification of the protocol will be taken into consideration to prove to what extent a protocol fulfils the requirements of services. For this purpose the PDL-service specification must be extended by a context specification of services describing the possible relations between them. We believe that language-oriented means must be supplemented for this by more suitable ones. In this connection we see interesting approaches in the method proposed in /Bochmann 80b/ and in temporal logics.

According to the two levels of protocol specification introduced a further classification is given:

(H) HORIZONTAL verification investigates communications between entities within one layer.

(V) VERTICAL verification proves communication between entities of adjacent layers (interface protocols /Danthine 80/)

Because considerations between layers are evidently based upon assertions on the behaviour of one layer, first (H) using PDL/D, then (V) on the base of PDL/I is to be considered (cp. Fig. 5).

Petri nets are a suitable model for both directions. Especially important is the fact that parts of system components can be already analysed when the essential behaviour of the environment has been modeled.

4.2. Preconditions

The procedure proposed here is an application of experiences which originally arose from the field of high-level programming languages for parallel processing /Heiner 80/ to computer network protocols.

The language design aspects important for the generation of and analysis by petri nets are the following:

- The scanning of the relevant synchronization primitives requires special syntactical units which are treated as basic elements like message, service, --> (sending), and <-- (receiving).

- Compact control structures (like par event and wait event) support simple petri net structures which can decrease the analysing amount substantially.

- An offered or enforced modularization supports a decomposition and reduction of the analysis (computer resources!).

In this connection it is worth mentioning that the costs of analysing a system of parallel processes for possible synchronization errors and the range of results are dependent on the synchronization concepts of the considered language to a high degree.

C.A.R. Hoare generalized this statement emphasizing that one of the most important aspects of high-level programming are simplification of program checking and improvement of error detection /Hoare 73/.

4.3. The Petri Net Generator, Analyser, and Interpreter

For verification the synchronization skeleton of the specified protocol is automatically mapped on a petri net by the so-called petri net generator. Such a generator needs for all synchronization primitives and control structures contained in the language a corresponding petri net module which maps the relevant control flow. These petri net modules are composed according to the result of the parser - the syntax diagram. This means, within a general scheme of a compiler structure the petri net generator appears as a particular semantic synthesis.

This automatically generated petri net is examinated by means of a general program system for petri net analysis /Christowa 85/. Thus, it is possible to draw conclusions about the absence of well-defined synchronization situations in the protocol according to (G).

Furthermore, the petri net model allows a simulation of the PDL-specification. For this purpose, the token flow through the synchronization skeleton which is possibly enriched by sequential protocol pieces is driven by a petri net interpreter. Thus the protocol specification becomes an experimental base, simultaneously.

Starting from this realistic point of view more rigorous verifications of the protocol should be taken into consideration including times and predicates.

4.4 Summary

We hope having shown by means of this short survey that the opinion "language-based specification is no suitable basis for verification", as sometimes pointed out in literature, is not completely justified.

5. GRAPHICAL DESIGN SUPPORT

In addition to the two main components of the PDL-system discussed above two further components will be included which can optionally be used for graphical design:

- a petri net translator, and
- a state transition diagram extractor.

By means of these features the user can transform graphical representations into a PDL-notation, and vice versa. The following figure gives a survey of the possible transformations between the allowed representations in the PDL-system.

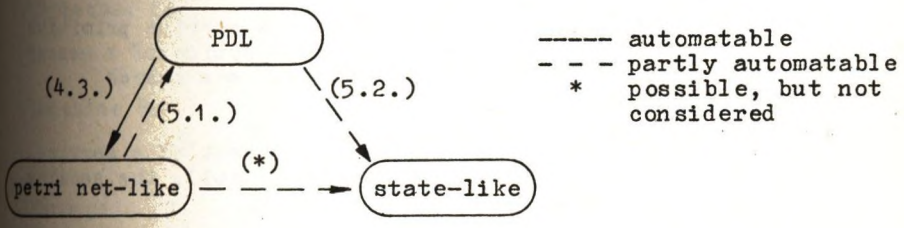


Figure 4: Transformation possibilities between several types of description

5.1. The Petri Net Translator

For users prefering a petri net-like design a procedure, the so-called petri net translator, is provided to support the transformation into a PDL-notation by means of a computer-aided upward translation.

In the context of communicating entities, e.g. parallel processes, such a translation bases on the recognizability of the synchronization primitives and the sequential control structures in the description.

Using this information the petri net translator produces a PDL-notation of the synchronization skeleton in which the positions of all sequential protocol pieces are exactly determined. If there are any sequences still omitted, they have to be included by the user.

Petri nets which are live and without deadlocks, yield protocol skeletons with just these properties.

It is well-known that upward translations are not always possible and generally not unique but always finite. Therefore this component needs the dialogue with the user who on prompting has to formulate his intention more detailed.

5.2. The State Transition Diagram Extractor

This extractor derives a state transition diagram from the PDL-notation which can be used for documentation purposes or for a comparison with possible state transition diagrams of first design.

The following mapping rules characterize the degree of abstraction of the state transition diagram automatically extracted:

- Its evident that meaningful identifiers for the states must be added by the user, because they are not contained in the PDL-description.

- Possible input parameters are the call of a service primitive by the upper layer (service asked), the reception of a message (<--), and a logical condition, if at least one branch containes a synchronization action.

- Possible output parameters are the announcement of a service primitive to the upper layer (service responsed), the sending of a message (-->), and an empty action.

It should be emphasized once more that the result of the extractor can only serve as a decision means which must be interpreted by the user.

6. FINAL REMARKS

The development of the PDL-system presented in this paper is part of the research work in the field of local area networks at the Technical University of Dresden.

The essential components of the system are implemented. For the language transformations (PDL/D-->PN, PDL/D-->PDL/I, PDL/I-->PN, PDL/I-->implementation frame) the DEPOT-metasystem (Lehmann 77) has been used. At the moment, an experimental transformation of PDL into Modula-2 as an implementation frame is being realized.

PDL was successfully applied to specify the protocols of the LAN LOTUNET /Loeffler 85/. Furthermore, the ISO-Draft Proposal for the transport protocol /Oniszeit 85/ and the ECMA-90 Bus Token Technique /Kolodziejczyk 85/ were specified and are verified and analysed now.

ACKNOWLEDGEMENT

The authors wish to thank Prof. H. Loeffler for his support and helpful discussions.

REFERENCES

/Bochmann 80a/ Bochmann,G.v. and Sunshine,C.A., Formal Methods in Communication Protocol Design, IEEE Trans. on Com., Vol. 28, No. 4, (1980) 624-631

/Bochmann 80b/ Bochmann,G.v., A formal Transition Model for Protocols and Communication Services, IEEE Trans. on Com., Vol. 28, No. 4, (1980) 643-650

/Bolt 80/ Bolt Beranek and Newman Inc. (eds.), Formal Description Techniques for Network Protocols, INWG General Notes 257, (Dec. 1980)

/Christowa 85/ Christowa,V., Ein Beitrag zur Realisierung und Anwendung der Netzanalyse, Diss. A, TU Dresden Sektion Informationsverarbeitung, (1985)

/Danthine 80/ Danthine,A.A.S., Protocol Representation with Finite-State Models, IEEE Trans. on Com., Vol. 28, No. 4, (1980) 632-642

/Harangozo 83/ Harangozo,J., State of the Art of Protocol Design in the early Eighties. Hungarian Academy of Sciences. Central Research Institute for Physics. KFKI-1983-98, Budapest, (1983)

/Heiner 80/ Heiner,M., Ein Beitrag zur Deadlockanalyse auf der Grundlage einer sprachlich gefuehrten Programmiermethodik, Diss. A, TU Dresden, Sektion Informationsverarbeitung, (1980)

/Hoare 73/ Hoare,C.A.R., High Level Programming Languages - the Way behind, Proc. of the British Computing Society Conference (1973)

/Kolodziejczyk 85/ Kolodziejczyk,T., Realisierung eines deterministischen Buszugriffsverfahrens mit dezentralisierter Steuerung in einem lokalen Netz Diplomarbeit, TU Dresden, Sektion Informationsverarbeitung, (1985)

/Koenig 85a/ Koenig,H. and Heiner,M., Definierender Bericht der Protokollbeschreibungssprache PDL, TU Dresden, Sektion Informationsverarbeitung, Forschungsbericht TU 08 RS-L/LN-K5/015, (1985) (available from the authors)

/Koenig 85b/ Koenig,H. and Heiner,M., PDL - ein sprachlich orientierter Ansatz zur Spezifikation von Rechnernetz-Protokollen, Tagungsberichte bit '85, Berlin (1985)

/Lehmann 77/ Lehmann,N.J., Problem-oriented languages and DEPOT implementing system, Lecture Notes on Computer Science 47 (1977) 1-20, Springer-Verlag Berlin-Heidelberg-New York

/Loeffler 85/ Loeffler,H. et al., LOTUNET - a LAN for Process Control, in this volume

/Oniszeit 85/ Oniszeit,J. and Koenig,H., PDL-Spezifikation des ISO-Standardvorschlags fuer das Transport-Protokoll, TU Dresden, Sektion Informationsverarbeitung, Forschungsbericht TU 08 RS-L/LN-K5/017, (1985)

/Rudin 83/ Rudin,H. and West,C.H. (eds.), Protocol Specification, Testing, and Verification III, North Holland, Amsterdam, (1983)

/Sunshine 82/ Sunshine,C. (ed.), Protocol Specification, Testing, and Verification, North Holland, Amsterdam, (1982)

APPENDIX

The following example, a description of a duplex variant of the Alternating Bit Protocol, should give a more complex survey about PDL. For methodical reasons, the protocol was modified introducing services to request and to indicate the sending and receiving of messages.

Figures 5 and 6 want to give an impression of automatically derived models.

```
layer AB;

   service asked AB-SEND request(data: data block);
           responsed AB-SEND indication;
                     AB-RECEIVE indication(data: data block);

   message PDU: record(data: data block, control bit: bit);
           ACK: bit;

   protocol Alternating Bit Protocol: send<=>receive;

   entity AB-E;

      service AB-SEND request;

      message PDU;
      protocol part send, receive;

      par event
         AB-SEND request: send |
         PDU <-- AB-E: receive
      #par

   #entity {AB-E}

#layer (AB}
```

```
protocol part send(entity receiver);

service AB-SEND indication;

message PDU to receiver;
        ACK from receiver;

timer t: 0..60;

begin
   PDU<:=data;
   loop
      PDU --> receiver;
      start t;
      wait event
         ACK <-- receiver: if ACK correct
                           then PDU.controlbit:=ACK;
                                response AB-SEND indication;
                                exit
                     #if |
         timeout t: skip
      #wait
   #loop
end {send}



protocol part receive(entity sender);

service AB-RECEIVE indication;

message ACK to sender;

var awaited bit: bit;

begin
   if controlbit correct
      then response AB-RECEIVE indication(PDU.data);
           awaited bit:=new bit;
      else ignore PDU
   #if;
   ACK:=awaited bit;
   ACK --> sender
end {receive}
```

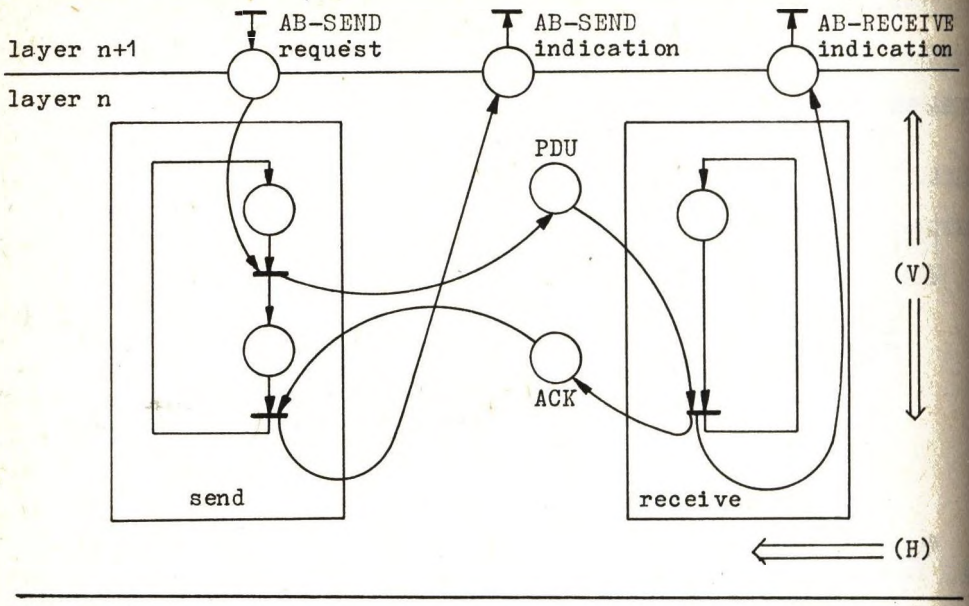


Figure 5: Reduced model of a petri net protocol

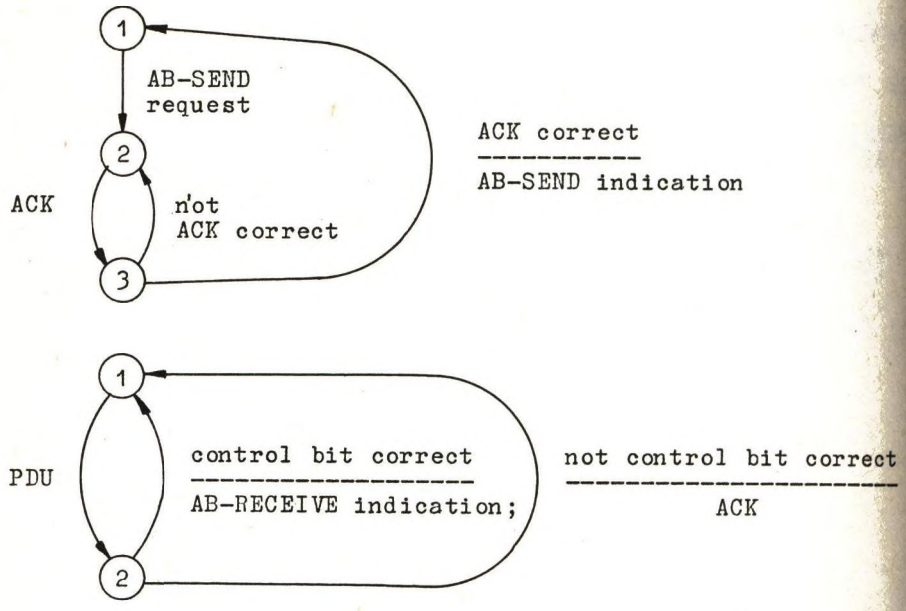


Figure 6: Derived state transition diagram

PNPUO - A Petri-net Based Software Package for Protocol Validation

Xiaoling Qiu * and To-Yat Cheung +

Abstract An interactive software package called PNPUO has been developed at the University of Ottawa. It can be used for specifying systems in terms of token-coloured or predicate/action Petri nets and for validating some of the system's logical properties, such as the presence or absence of deadlocks, livelocks and home-states, detection of place and transition invariants, etc. This paper describes our application of PNPUO to the validation of the Alternating Bit Protocol for both perfect and imperfect channels and the Connection and Disconnection phases of Class 1 and Class 2 of the Transport Protocol.

Key Words Petri nets, software tool, protocol validation, specification, alternating bit protocol, transport protocol.

+ Distributed Computing Research Group, Department of Computer Science, University of Ottawa, Ottawa, Ont. Canada.

* Beijing General Research Institute of Mining and Metallurgy, Peijing, People's Republic of China, on leave to University of Ottawa when writing this paper.

- Research supported by Natural Sciences and Engineering Research Council of Canada Grant No. A8963.

1. INTRODUCTION

For many real-life systems, it is hard and expensive to correct errors once they have been implemented. It is important to develop methods for describing and analysing these systems and detecting their errors at an early stage. Automation of these methods are even more useful.

There exist many approaches for system modeling and analysis [DANT80] [DIAZ82]. Petri nets (PN) are efficient and powerful abstract formal models for studying both static and dynamic properties of concurrent systems [MURA77] [RAMA82]. Their simple diagrams for description and powerful techniques for analysis are widely used in the modeling of software and hardware systems [NELS83] [PETE77]. In particular, Petri nets have been applied in the specification, verification and testing of communication network protocols [BERT82] [COUR84] [JURG84] [TANE81].

As the size and complexity of a Petri net grows, analysis by manual processes becomes extremely long and tedious, it is useful to have a software tool for automating the processes. As far as we know, there exists only one such tool, called OVIDE [MONT8^]. Publications about this commercially available system are very limited, especially about the system's implementational and operational details. It is coded in FORTRAN and can handle basic Petri nets only (e.g., no predicates and actions). OGIVE [COUR84], an earlier version of

OVIDE but with fewer facilities, has been used in several research institutes. Recently Jurgensen and Vuong [JURG84] applied it for the specification and validation of Class 4 ISO Transport Protocol. It was reported that OGIVE was not able to produce a basis of the vector space containing the net's place invariants within a reasonable amount of time (10 sec. of CPU time). Only by including additional places and excluding other results could place invariants be determined. Also, no transition invariants have been obtained.

Independent of the works of OGIVE and OVIDE (we do not have these packages), an interactive software package called PNPUO has been developed at the University of Ottawa for our distributed computing research. PNPUO is coded in PASCAL and can handle most of the major functions for analysis as OGIVE and OVIDE, except the graphical and property-preserving net reduction capabilities of OVIDE. However, unlike OGIVE and OVIDE which accept only basic Petri nets, PNPUO can specify and validate systems using token-coloured and predicate/action Petri nets. As a consequence, PNPUO can handle problems using Petri nets of comparatively smaller sizes, since some of the states, transitions and conditions are accounted for by the predicates/actions or token colours.

In Section 2, we briefly explain the theory and processes of using Petri nets for system specification and validation. Section 3 summarizes the main features of PNPUO. Section 4

describes in greater details the application of PNPUO to two widely known protocols -- the Alternating Bit Protocol and the Class 1 and Class 2 of ECMA Transport Protocol. Some conclusive remarks follow in Section 5.

2. SYSTEM ANALYSIS BY PETRI NETS

This section presents some basic properties and fundamentals of Petri nets (PN), as well as the techniques for system analysis based on PN.

Definition A Basic Petri net (BPN) is a 5-tuple (P,T,PRE,POST,M0) , where

P is a set of places p1,p2,... pm , m >= 0 .

T is a set of transitions t1,t2,... tn , n >= 0 ,
where P ∩ T = null .

PRE : P X T --> {0,1} is the forward incidence function defined on the set of arcs leading from places to transitions .

POST : P X T --> {0,1} is the backward incidence function defined on the set of arcs leading from transitions to places.

M0 : P --> {0,1} is the initial marking (See definition below).

Definition A marking M is an m-vector denoting a distribution of tokens in the places. Notationally, M = (M(p1),M(p2),... M(pm)), where M(pi) is the number of tokens

residing in place pi . An initial marking M0 is the marking of the Petri net at an initial moment.

Definition The incidence matrix of a PN is an mxn matrix

$$C = (cij)$$

where $cij = POST(pi,tj) - PRE(pi,tj)$.

A transition t is said to be firable if $M(p) >= PRE(p,t)$ for all input places p of t. The firing of transition t at a marking M leads to a new marking M' . The operation is mathematically expressed as

$$M(t >_M '$$

where $M'(p) = M(p) - PRE(p,t) + POST(p,t)$, $\forall p \in P$. (1)

Equation (1) is called the firing equation for transition t.

Suppose a sequence σ of firing transitions lead a PN from marking M to M'. The new marking M' is given by the following matrix equation:

$$M' = M + C\sigma$$

where C is the incidence matrix of the PN ,

$\sigma = (\sigma 1, \sigma 2, \ldots \sigma n)$ is the vector of the firing transition sequence $\bar{\sigma} = (tj1, tj2, \ldots tjk)$, with $1 <= ji <= n$, $1 <= i <= k$, i.e., σi is the number of occurrences of ti in sequense $\bar{\sigma}$.

The reachability set of a marking M , denoted by [M], is the set of markings each obtained from marking M by firing a transition sequence σ .

$$[M] = \{ M' \mid M(\sigma > M' , \sigma \in T^* \}$$

where T^* is the Kleene closure of T .

The reachability set of the initial marking M0 , denoted as [M0] , is called the reachability tree of the PN.

In this paper, we consider two extensions of BPN:

a. Token-coloured PN.

b. Predicate/Action PN.

In a token-coloured PN, tokens are allowed to have different nature (represented in terms of colours) and values. The marking M of a token-coloured PN has become a matrix : M = (M(p,k)) , where M(p,k) is the number of tokens with the kth colour in place p.

In a predicate/action PN [DIAZ82], predicates are associated with a transition and its input places as additional conditions for the firability of the transitions, and actions are associated with a transition as operations to be done after having fired the transition.

2.1 Problems for Analysis

In analysing a system, one of the objectives is to determine the presence or absence of desirable or undesirable properties. In the following, we classify the properties of a PN according to three aspects: token-related, execution-related and structural.

(1) Token-related properties include safeness, boundedness, and token conservation [PETE81] . They are defined below .

Definition A PN is safe if the number of tokens in each place never exceeds 1. That is,

$$M(p) <= 1 \ , \ \forall p \in P \ , \ \forall M \in [M0] \ .$$

Definition A PN is bounded if there exists an integer K > 0 which the number of tokens in each place never exceeds. That is,

$$M(p) <= K \ , \ \forall p \in P \ , \ \forall M \in [M0] \ .$$

Definition A PN is conservative if the total number of tokens in the net remains constant, that is

$$\sum_{p \in P} M(p) = \sum_{p \in P} M0(p) \quad , \ \forall M \in [M0] \ .$$

(2) Execution-related properties concern the reachability of certain markings, such as the following:

Deadlock - In this status, no transition can be fired.

Livelock - In this status, marking M is included in an execution loop without any transition leaving

the loop.

Live-looping - In this status, marking M is included in an execution loop which has at least one possible transition leaving the loop.

Home-state - In this status, Marking M is the same as the initial marking, i.e., M = M0, but is not the root of the reachability tree.

(3) Structural Properties

The structure of a PN may be investigated through place invariants and transition invariants of the net. A place invariant states that, during execution of a PN, the total number of tokens within a specific subset of places remains constant. It thus indicates the co-existence of certain states. A transition invariant represents a sequence of fired transitions which have no effects on a marking M [BERT82].

2.2 Techniques for Analysis

Two major techniques are often used to derive the above properties: enumerative analysis and structural analysis [MONT83] [JURG84]. The first technique is based on executing the PN and generating its reachability tree. The second explores the structure of the net by deriving its place and transition invariants.

Enumerative Analysis

Enumerative analysis begins with generating the reachability tree of a PN. The tree has its root at the initial marking M0. Its nodes represent the reachable markings and its arcs are labelled with the fired transitions of the PN. The generated markings (i.e., nodes of the tree) are checked for the status of deadlock, or livelock, or live looping, or home-state.

Structural Analysis

In structural analysis, we get two sets of invariants (if they exist) - place invariants and transition invariants.

(1) Place invariants are the solutions of the matrix equation

$$\phi C = 0 ,$$

where ϕ is an m-vector (m is the total number of places of the PN) and C is the incidence matrix of the PN. A place invariant corresponds to a set of places the total number of tokens of which is constant.

(2) Transition invariants are the solutions of the equation $Cf = 0$. A transition invariant represents a sequence of transitions which have no effects on the marking.

3. THE PACKAGE PNPUO

To automate the aforementioned analysis techniques, we have developed a software package called PNPUO at the University of Ottawa. It was coded in PASCAL 8000 and implemented in an AMDAHL computer. It operates under the Conversational Monitor System. The program requires a memory space of about 10K words (as compared to OVIDE which is coded in FORTRAN and requires a memory space of 50K words).

3.1 Capabilities of PNPUO

PNPUO provides the user with interactive facilities for analysing properties of systems which use basic, or token-coloured, or predicate/action Petri nets as design models. This is done mainly in two phases :

Phase I Creation and Storage of Data

A user can interactively input, display, modify, store and get hard copies of the following input data:

a. The numbers of places, transitions, and token colours of the Petri net.

b. Labels (numbers and colours) of arcs from places to transitions and from transitions to places.

c. User variables (integer, real, Boolean) used in the predicates and actions of Petri nets.

d. Predicates and actions at places and transitions.

e. Initial marking and initial values of variables.

Phase II System Analysis

After data have been correctly input, the system enters the analysis phase, during which the user can investigate properties of the system by the following two processes:

(i) In the process of enumerative analysis, it checks:

a. Token-related properties: safeness, boundedness and conservation of tokens.

b. Execution-related properties : deadlock , livelock , live-loop and home-state.

(ii) In the process of structural analysis, it determines:

a. All place invariants .

b. All transition invariants .

3.2 Limitation on Applications

Though the framework of PNPUO is designed in such a way that many types of Petri nets can be analysed, the current version has only the software for token-coloured and predicate/action PN, including basic PN as a special case.

The following limitations are mainly due to implementational convenience and restrictions and can be changed quite easily.

a. The maximum numbers of places, transitions, and token colours are limited to 40, 40, and 9, respectively.

b. The maximum number of each type of user variables, namely integer, real and Boolean, is limited to 8.

c. The symbols for integer, real and Boolean variables are restricted to A, B,... H; A1, B1,... H1; and A2, B2, ... H2, respectively.

d. A predicate must be one of the following two logical expressions:

i) X = R

ii) X = R and Y = S

where X, Y are program variables of any type (integer, real, or Boolean), and R, S are constants of the same type as X and Y, respectively.

e. An action must be one of the following two logical expressions :

i) X := X' + R

ii) X := X' + R and Y := Y' + S

where X, Y, R, S are the same as above and X' (Y') is either X (Y) or null.

f. The maximum number of levels of the reachability tree is 15 .

4. PROTOCOL VALIDATION BY PNPUO

Using BPN and predicate/action PN models, we apply PNPUO to validate two well-known protocol systems, the alternating bit protocol in the link layer and the Connection and Disconnection Phases of Class 1 and Class 2 of the Transport Protocol.

4.1 Analysis of the Alternating Bit Protocol by PNPUO

We assume a noisy channel (i.e., messages may be damaged or lost in the channel) for the alternating bit protocol. In this protocol, every message has a one-bit sequence number as an identifier . During transmission, the sequence number alternates between the values 0 and 1. A protocol entity does not send the next message until the one sent previously has been positively acknowledged. Recovery from transmission anomalies (errors or loss) is achieved by a time-out mechanism and retransmission of messages.

A Predicate/Action PN Model

As shown in Figure 1, our predicate/action PN model for the alternating bit protocol is obtained by adding to the basic Petri net model of Tanenbaum [TANE81] two variables, A and B, to specify the functions 'message loss' and 'time-out'. It attaches predicate [A=1] and actions 'A:=0 and B:=1' to the transitions 'loss' (t5, t6 and t7) and predicate [B=1] and action 'B:=0' to the transitions 'timeout' (t3 and t4).

In the initial state M0 (only places p1, p3 and p7 each have a token), Sender is waiting for ACK0 (acknowledgemnet for the message with sequence number 0), while Receiver has sent ACK1 and is expecting for message 0. Suppose Receiver receives message 0 (by firing t10) and sends ACK0 (also by firing t10).

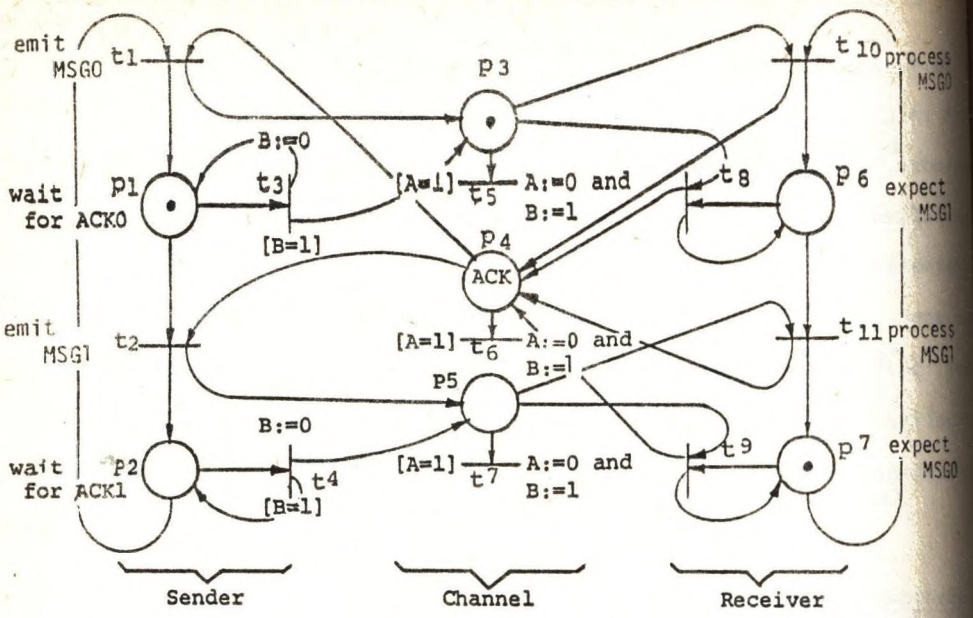


Figure 1: A Predicate/Action PN Model for the Alternating Bit Protocol

The state of the PN changes to marking M1 (only places p1, p4 and p6 each have a token). On receiving ACK0, Sender sends message 1 (by firing t2) and then waits for ACK1. Once Receiver gets message 1 (t11), it sends ACK1 (p4 gets a token). If Sender receives ACK1 (by firing t1), it transmits the next message 0. The system returns to the initial state of the next cycle.

When one of the transitions t5, t6 and t7 are fired and a message is lost, i.e., when Predicate [A=1] is true, Action 'A:=0 and B:=1' is executed. Thus, the Predicate [B=1] associated with transitions t3 and t4 ('time-out') becomes true

and one of them can be fired (i.e., time runs out), allowing the sender to retransmit the message.

Note: Throughout this paper, the notation M(r,q) (a, b, ..., k) used in a reachability tree indicates the qth marking at level r of the tree, where each of the places p_a, p_b, ..., p_k has one token.

Results of Analysis

(1) Enumerative Analysis

Reachability Trees Figure 2 shows the reachability tree for a channel without message loss and timeout, (i.e., A=B=0 all the time). Starting from M0, it has only one path reaching the home-state, also M0. If a message is lost (i.e., variable A is set to 1), the reachability tree will depend on the state (marking) in which the message is lost. Figures 3, for example, shows the reachability tree where a message is lost in the second state of Figure 2.

Token-related properties (output of PNPUO):

```
THE RESULTS OF REACHABILITY ANALYSIS:
<1>  THE NET IS SAFE .
<2>  THE MAX NUMBER OF TOKENS IN EACH PLACE IS :
                          ( FOR COLOUR1 )        1
<3>  THE NET IS NOT CONSERVED .
```

Execution-related Properties (output of PNPUO):

```
     <4>  EXECUTION-RELATED PROPERTIES:
```

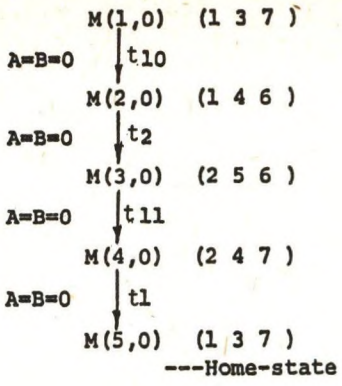


Figure 2: Reachability Tree for A Perfect Channel

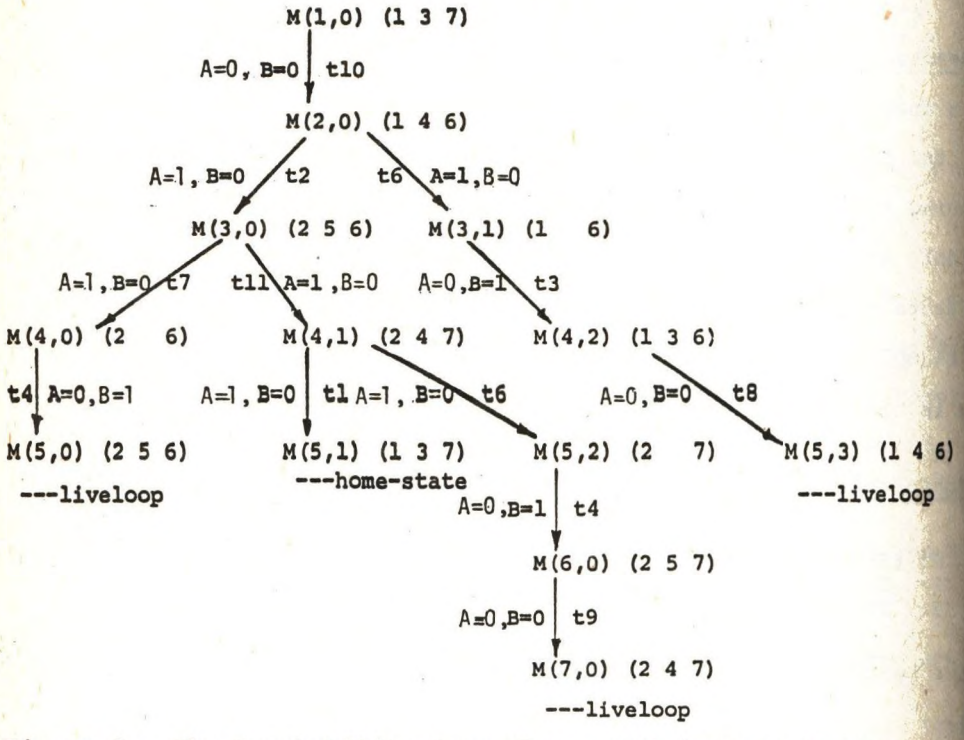


Figure 3: The Reachability Tree When a MSG is Lost in the 2nd state of Execution

```
NO DEADLOCKS
NO LIVE-LOCKS
NO LIVE-LOOPING
M0 IS A HOME-STATE
```

The above output information shows that the protocol is free from these anomalies. Since M0 is both the initial state and a home-state, the protocol terminates correctly. There is no message overflow, as the PN (including the channel as a subset) is bounded.

(2) Structural Analysis

Place Invariants (output of PNPUO):

```
PLACE INVARIANT 1:      M(1) + M(2) = 1
PLACE INVARIANT 2:      M(6) + M(7) = 1
PLACE INVARIANT 3:      M(1) + M(2) + M(6) + M(7) = 2
```

These place invariants show, for example, that, during the execution of this PN, there is always a token in either p1 or p2, but not in both. The same situation is also true for p6 and p7. This means that in the alternating bit protocol, both entities (Sender and Receiver) have to stay in a single state, i.e., there is no state ambiguity of the protocol entity.

Transition Invariants (output of PNPUO):

```
TRANSITION INVARIANT 1 :      T3 T5
                            M -----> M

TRANSITION INVARIANT 2 :      T4 T7
                            M -----> M

TRANSITION INVARIANT 3 :      T3 T4 T5 T7
                            M ----------> M

TRANSITION INVARIANT 4 :      T3 T6 T8
```

```
                                    M --------> M

TRANSITION INVARIANT 5 :              T3 T4 T5 T7 T8
                                    M ------------> M

TRANSITION INVARIANT 6 :              T4 T6 T9
                                    M --------> M

TRANSITION INVARIANT 7 :              T3 T4 T5 T6 T9
                                    M ------------> M

TRANSITION INVARIANT 8 :              T1 T2 T8 T9
                                    M ----------> M

TRANSITION INVARIANT 9 :              T1 T2 T3 T5 T8 T9
                                    M ----------------> M

TRANSITION INVARIANT 10:              T1 T2 T4 T7 T8 T9
                                    M ----------------> M

TRANSITION INVARIANT 11:              T1 T2 T3 T4 T7 T8 T9
                                    M ------------------> M

TRANSITION INVARIANT 12:              T1 T2 T10 T11
                                    M ------------> M

TRANSITION INVARIANT 13:              T1 T2 T3 T5 T10 T11
                                    M ------------------> M

TRANSITION INVARIANT 14:              T1 T2 T4 T4 T7 T10 T11
                                    M ----------------------> M

TRANSITION INVARIANT 15:              T1 T2 T3 T4 T5 T7 T10 T11
                                    M ------------------------> M

TRANSITION INVARIANT 16:              T1 T2 T6 T8 T10 T11
                                    M ------------------> M

TRANSITION INVARIANT 17:              T1 T2 T3 T4 T6 T7 T8 T10 T11
                                    M --------------------------> M

TRANSITION INVARIANT 18:              T1 T2 T4 T6 T9 T10 T11
                                    M ----------------------> M

TRANSITION INVARIANT 19:              T1 T2 T3 T4 T5 T6 T9 T10 T11
                                    M --------------------------> M
```

These transition invariants indicate that there are 19 loops during the execution of this PN. They may be life-cycles

(correct termination) or live loops, or even livelocks. The structural analysis process cannot distinguish them. It is necessary to check them in the reachability tree.

Transition invariant 1, for example, implies that firing the transition sequence (t3, t5) leads a marking (here it is the initial marking M0) to itself again. This is obviously true for this PN. Correctness of the other invariants can also be checked in a similar way.

4.2 Analysing the Connection and Disconnection Phases of Class 1 and Class 2 of Transport Protocol

This section describes our application of PNPUO to a portion of the ECMA Transport Protocol. We first use a basic PN model proposed by Berthelot [BERT82] and compare his manually obtained results with ours. We then apply PNPUO again, using a predicate/action PN.

The Transport Protocol (TP), as proposed by both ISO (International Organization for Standardization) and ECMA (European Computer Manufacturers Association), has five different classes which handle errors of increasing severity [ECMA81] [ISO 83]. Each class provides services in three phases :

(1) Connection Establishment Phase

(2) Data Transfer Phase

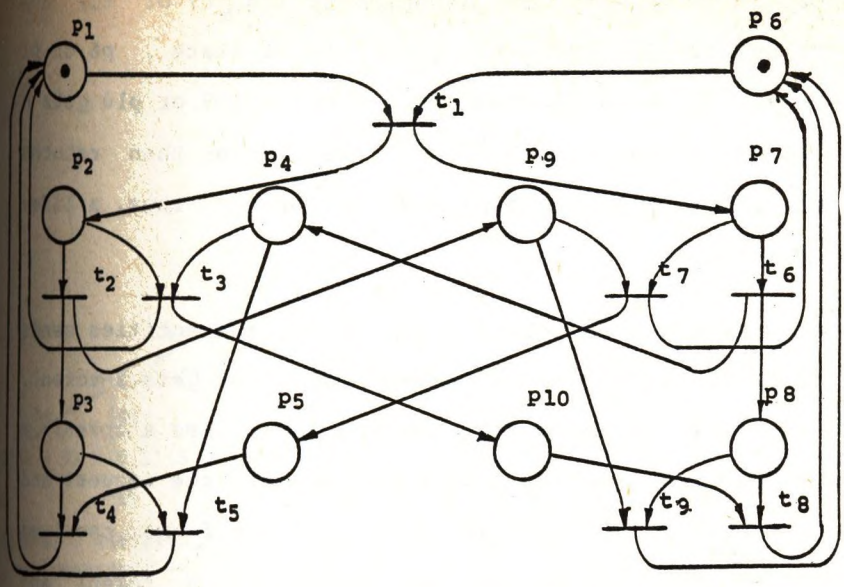
(3) Disconnection Phase

The function of the Connection Establishment Phase is to provide the TP primitives: connect request, connect indication, connect response and connect confirmation for the establishment of an end-to-end transport connection between two transport entities. The Data Transfer Phase serves the exchange of data messages via TP primitives: data request, data indication, expedited-data request and expedited-data indication. The Disconnection Phase uses the TP primitives disconnect request, disconnect indication, disconnect response and disconnect confirmation to terminate a transport connection.

In the Class 1 protocol, not only are the functions of these three phases provided, the ability of error detection and reporting and of recovery from failures is also included. Class 2 provides also flow control for merging multiple Transport connections on a single Network connection.

4.2.1 Results of Analysis Based on A BNP Model

In Berthelot's model (Figure 4), transition t1 is shared by the two transport protocol entities and is used to indicate the reception of a connection request either from its Session Layer or from its peer Transport Protocol entity.



PLACES

State of Entities
p1,p6 : idle
p2,p7 : data transfer
p3,p8 : waiting p5 ,p10

Control Frames
p4,p9 : disconnection request
p5,p10: disconnection confirm

TRANSITIONS

Event of Entities
t1 : connection accepted
t2,t6 : request for disconnection
t3,t7 : disconnection accepted
t4,t8 : end of disconnection
t5,t9 : disconnection accepted

Figure 4: A BPN Model for the Connection and Disconnection Phases of Transport Protocol [BERT82]

When t1 is fired, each entity enters the Data Transfer State (p2 and p7 each get a token). Suppose, having transmitted data, one of the two entities sends (firing t2 or t6) a Disconnect Request (p9 or p4 gets a token). It then waits (p3 or p8) for Disconnection Confirm (waiting until p5 or p10 has got a token). If the other entity is still in a Data Transfer State

(p7 or p2 has a token), it accepts (firing t7 or t2) the Disconnect Request and returns to its initial state , p6 or p1 . Meanwhile, it sends Disconnection Confirm (p5 or p10 gets a token) to the requesting entity. The latter then returns (firing t4 or t8) to its initial state p1 or p6. Thus, a life-cycle is completed.

Another possible case occurs when one of the entities sends (firing t2 or t6) a Disconnect Request (p9 or p4 gets a token). The other entity, while waiting for (p8 or p3 has a token) a Disconnect Confirm, accepts (firing t9 or t5) the request and returns to the initial state (p6 or p1). The former receives (firing t5 or t9) the request (p4 or p9 has a token) and returns to the initial state p1 or p6.

Following is the analysis results of applying PNPUO to the model discussed above.

(1) Enumerative Analysis

Reachability Tree (See Figure 5)

Token-related properties (output of PNPUO):

```
THE RESULTS OF REACHABILITY ANALYSIS:
<1>  THE NET IS SAFE .
<2>  THE MAX NUMBER OF TOKENS IN EACH PLACE IS:
                          ( FOR COLOUR1 )     1 .
<3>  THE NET IS NOT CONSERVED .
```

Execution-related Properties (output of PNPUO):

```
<4>  EXECUTION-RELATED PROPERTIES:
          NO DEADLOCKS
          NO LIVE-LOCKS
```

NO LIVE-LOOPING
M0 IS A HOME-STATE

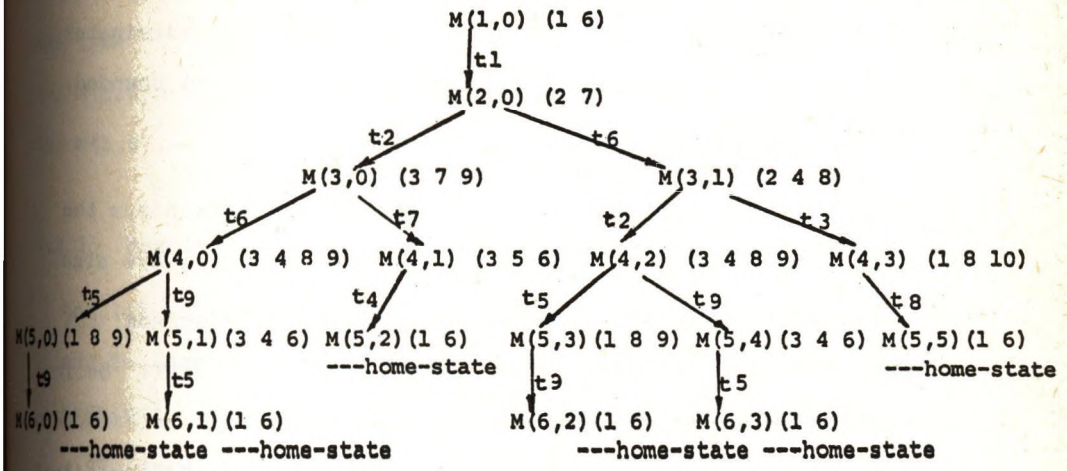


Figure 5: Reachability Tree of the PN in Figure 4

(2) Structural Analysis

Place Invariants (output of PNPUO):

```
PLACE INVARIANT 1:  M(1) + M(2) + M(3)                          =  1
PLACE INVARIANT 2:  M(1) + M(4) + M(5) + M(7)                   =  1
PLACE INVARIANT 3:  M(6) + M(7) + M(8)                          =  1
PLACE INVARIANT 4:  M(1) + M(2) + M(3) + M(6)
                    + M(7) + M(8)                               =  2
PLACE INVARIANT 5:  M(2) + M(6) + M(9) + M(10)                  =  1
PLACE INVARIANT 6:  M(1) + M(2) + M(4) + M(5)
                    + M(6) + M(7) + M(9) + M(10)                =  2
```

Transition Invariants (output of PNPUO):

```
TRANSITION INVARIANT 1 :      T1 T2 T4 T7
                          M ------------> M

TRANSITION INVARIANT 2 :      T1 T3 T6 T8
                          M ------------> M

TRANSITION INVARIANT 3 :      T1 T2 T5 T6 T9
```

M ----------> M

From these results, it can be concluded that the protocol is free from deadlocks and livelocks and that it terminates correctly at the initial state. As the PN is safe and bounded, there is no channel overflow.

The place invariants (1), (3), (2) and (5) we obtain are the same as those of [BERT82]. The other two, (4) and (6), are also correct but do not appear in [BERT82]. Here, invariants (1) and (3) show that each entity stays in one and only one of the three phases at any moment. Invariants (2) and (5) indicate that, if one entity is in the Disconnection Phase (p1 or p6), the other, while being in the Data Transfer Phase (p7 or p2), cannot issue a Disconnect Request (p4 or p9) or Disconnect Confirm (p5 or p10). Place invariant (4) is implied by invariants (1) and (3), and place invarint (6) by invariants (2) and (5).

All our transition invariants are the same as those obtained by Berthelot [BERT82] manually. The first one indicates that firing the sequence of transitions t1, t2, t4, t7 leads a marking M (here M0) to itself (M0). The same situations occur for the sequence t1, t3, t6, t8 and sequence t1, t2, t5, t6, t9. These invariants are reflected in the reachability tree (Figure 5) as those paths starting from the root M0 and terminating at the leaves which are home-states. Note that

occasionally different paths in the tree may correspond to the same sequence of firing transitions. For example, the third transition invariant corresponds to the two leftmost paths in the tree :

(a) M(1,0) -> M(2,0) -> M(3,0) -> M(4,0) -> M(5,0) -> M(6,0)

(b) M(1,0) -> M(2,0) -> M(3,0) -> M(4,0) -> M(5,1) -> M(6,1) .

4.2.2 Results of Analysis Based on A Predicate/action PN

For analysis involving with more details, we have also applied PNPUO to the above Transport Protocol using a predicate/action PN (Figure 6). We associate a predicate, say [A=1], with transition t1 to indicate that the demanded quality of the connection is met. The initial value of A is 0. Two predicates, [B=1] and [C=1], attached to t2 and t6 respectively, indicate that no more data is to be transferred in each entity. The corresponding actions, A:=0, B:=0 and C:=0, indicate the opposite situations, repectively.

We got the following analysis results for this model: If we assign the value 1 to the variables A, B and C at the beginning of the reachability analysis, we get the same reachability tree as Figure 5. This means that, if the entities have no more data to transfer, the system terminates properly.

If we keep A=B=C=0 during the generation of the reachability tree, we get the outputs as shown in Figure 7. This result indicates that if the system cannot fulfill the connection

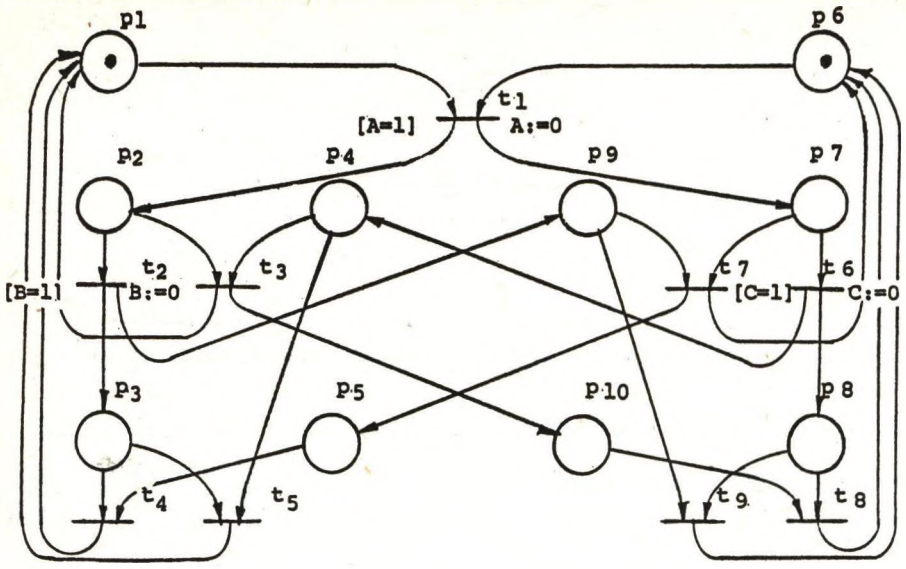


Figure 6: A Predicate/Action PN Model for the Transport Protocol

quality requested, the Transport Protocol will not enter the Connection Phase.

Figure 8 shows the reachability tree in the case of assigning A=1 at the beginning of executing the PN and keeping B=C=0 during the execution. This is the situation when both entities stay in a deadlock with respect to the two phases described in this figure. Operationally, it means that the system stays in (i.e., locked within) the Data Transfer Phase forever.

If we set A=1 and B=1 but keep C=0 during the execution, we get the reachability tree as shown in Figure 9. This figure

shows that one of the entities is forced back to the Disconnection Phase before it has finished transferring its data.

As for structural analysis, we get the same results as for the BPN model (Subsection 4.2.1).

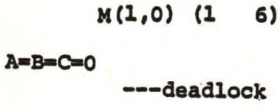


Figure 7: The Reachability Tree for the Enhanced Model with A=B=C=0

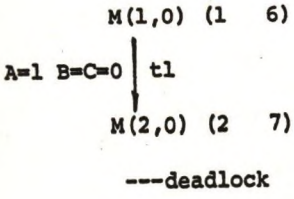


Figure 8: The Reachability Tree for the Enhanced Model with A=1 and B=C=0

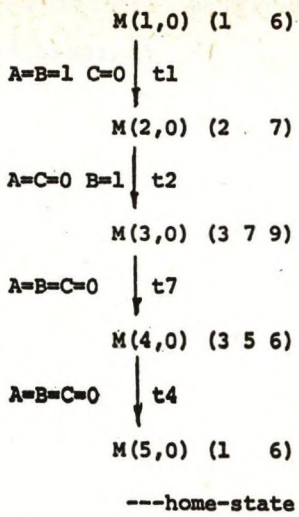


Figure 9: The Reachability Tree for the Enhanced Model with A=B=1 and C=0

5. CONCLUSION

Independent of OGIVE and OVIDE, we have developed a software package called PNPUO for specifying and analysing logical properties of systems modelled in terms of Petri nets. Our package can handle more types of Petri nets, namely token-coloured and predicate/action Petri nets.

Through enumerative and structural analysis, PNPUO can determine such properties as boundedness, token-conservation, deadlocks, livelocks, live-loops, home-states, place invariants and transition invariants.

PNPUO has been applied to the validation of the ALternating Bit Protocol and Class 1 and Class 2 of the Transport Protocol.

New properties about these protocols not reported in the literature have been found.

We find PNPUO a very useful and powerful design tool for research. It can easily be extended to include more types and techniques of Petri nets. Its main drawback (in fact a drawback of the current Petri net technology itself) is that it depends heavily on reachability analysis to determine some of the system's properties. Computationally, this is sometimes quite inefficient, especially for large systems.

REFERENCES

BERT82 G. Berthelot and R. Terrat, 'Petri Nets for the Correctness of Protocols', IEEE Trans. on Communications, Vol.COM-30, No.12, December 1982, pp. 2497 - 2505.

COUR84 J. P. Courtiat, J. M. Ayache and B. Algayres, 'Petri Nets Are Good for Protocols', Proc. SIGCOMM'84, (the same as Communcation Review, Vol. 14, No. 2), June 1984, pp. 66 - 74.

DANT80 A. S. Danthine, 'Protocol Representation with Finite-State Models', IEEE Trans. on Communications, Vol.COM-28, No.4, April 1980, pp. 632 - 642.

DIAZ82a M. Diaz, 'Modeling and Analysis of Communication and Cooperation Protocols Using Petri Net Based Models', Computer Networks, Vol.6, 1982, pp. 419 - 441.

DIAZ82b M. Diaz, 'Modeling and Analysis of Communication and Cooperation Protocols Using Petri net Based Models', Protocol Specification, Testing, and Verification, C. Sunshine (ed.), North-Holland Co. 1982, pp. 465 - 510.

ECMA81 'Standard ECMA-72 Transport Protocol', ECMA (European Computer Manufacturers Association), Jan. 1981.

JURG84 W. Jurgensen and S. T. Vuong, 'Formal Specification and Validation of ISO Protocol Components Using Petri Nets', Proc. SIGCOMM 1984, (the same as Computer Communication Review, Vol.14, No.2), June 1984, pp. 75 - 82.

MONT83 B. Montel, et. al., 'OVIDE, a Software package for the Validation of Systems Represented by Petri Nets Based Models', 4th European Workshop on Applications and Theory of Petri Nets, Sept. 1983, Toulouse, France.

MURA77 T. Murata, 'State Equation, Controllability and Maximal Matchings of Petri Nets', IEEE Trans. on Automatic Control, Vol.AC-22, No.6, June 1977, pp. 412 - 416.

NELS83 R. A. Nelson, L. M. Haibt and P. B. Sheridan, 'Casting Petri Nets into Programs', IEEE Trans. on Software Engineering, Vol.SE-9, No.5, Sept. 1983, pp. 590 - 602.

PETE77 J. L. Peterson, 'Petri Nets', ACM Computing Surveys, Vol.9, No.3, Sept. 1977, pp. 223 - 252.

PETE81 J. L. Peterson, Petri Nets Theory and the Modeling of Systems, Prentice - Hall, 1981.

TANE81 A. S. Tanenbaum, Computer Networks, Prentice - Hall, 1981.

OBTAINING NORMAL FORM SPECIFICATIONS FOR PROTOCOLS

Behcet Sarikaya
Universite de Sherbrooke
Departement de Math-Info
Sherbrooke, Quebec, Canada
J1K 2R1

G. v. Bochmann
Universite de Montreal
Departement d'I.R.O.
C.P. 6128 Succ "A"
Montreal, Quebec, Canada
H3C 3J7

ABSTRACT

We take Estelle as an example protocol specification formalism. We explain transformations on Estelle constructs and develop an algorithm to combine modules. The resulting specification is called a normal form specification. After defining the graph models, we discuss various uses of the normal form specifications concentrating on protocol design validation.

1. INTRODUCTION

Software development is the process of obtaining a solution to a problem thought to be solvable by a computer system. There are two main approaches to software development: conventional and operational. In the conventional approach, the solution progresses in the form of requirements specification, design specification and finally implementation. The operational approach is based on separation of problem-oriented from implementation-oriented concerns. It develops an operational specification then a transformed specification and finally an implementation [13].

Operational specifications describe what is desired not how it is to be realized. They are process-oriented, tolerant of incompleteness and augmentable. Operational specifications can act as a generator of possible implementation behaviors [6].

Formal specifications are considered important tools for the reliable design and implementation of computer network protocols. In particular, we consider the specification language Estelle developed by Subgroup B of the ISO TC97/SC16/WG1 [15]. Since this paper limits its scope to the protocols based on ISO Reference Model [2], problem definition is assumed to be available in the form of English description

of the standard protocol/ Service. The software development continues with the elaboration of a reference specification in Estelle. In the next step, protocol specification is validated. The validated specification is further elaborated with the addition of implementation decisions obtaining a transformed specification from which an implementation is obtained ([16], [3], and [7]). Note that these steps come close to the operational approach to software development.

In this paper we describe a method for validating Estelle specifications. In particular we limit the validation to syntactic/ semantic analysis of the protocol/ service specification. Estelle is introduced in Section 2 where an example is given. This example is used to show the transformations on the constructs of Estelle and combining modules which are discussed in Section 3. The specification obtained after these operations is called a normal form specification. In Section 3, we also discuss the flow graphs of the normal form specification. Design validation is discussed in Section 4. In particular, syntactic and semantic errors including design errors are detected during the development of the normal form specification and then using the flow graphs. Errors found in three Estelle specifications are enumerated. Finally, Section 5 concludes the paper.

2. ESTELLE

Estelle is based on an extended finite state transition model and the Pascal programming language. A protocol/ service is specified as a set of communicating modules. The state space of a module is defined by a set of variables including a variable called "STATE" which represents the major state of the module. The interactions of a module with its environment are specified by transition types.

A transition type is characterized by an enabling condition (pre-condition) and an action defining post-condition. The enabling condition is expressed using special Estelle constructs called PROVIDED

and FROM clauses and (possibly) the specification of an input using the WHEN clause. A transition type without input is called a spontaneous transition.

The action of a transition type is to be executed as part of the transition. All Pascal executable language constructs (assignment statement, conditional statements (IF, CASE), loop statements (FOR, WHILE)) are allowed. The action may also specify the initiation of output interactions with the environment using OUT clause, as well as a next major state, using the TO clause. Consideration of the special Estelle construct called ANY clause is left outside the scope of this paper.

2.1. Sample Transition Types

The transformations of Section 3 will be explained using an example protocol specification. In accordance with the method suggested in [5], this protocol has two modules (mapping and logical protocol modules). The substructure of the protocol is shown in Figure 2.1. The protocol is connected to its environment with the channels called NS and N_1S.

Two example transition types of this specification are given in Figure 2.2. CONNECT_req over NS is an input primitive, A, B and C are elementary expressions involving module variables and the input primitive parameters, S1 and S2 are blocks of assignment statements, and mapping event1 is an output statement to mapping module. The second transition type has event1 which can be received from the logical protocol module as an input with parameters data=unit and length. S3 represents a block of statements and CONNECT_pdu is an output to an external entity over N_1S.

2.2. Normal Form Specification

In an effort to decompose a complex protocol into its functions using the formal specification of the protocol we found it useful to

simplify a given specification. Major complications in an Estelle specification include conditional and loop statements in the actions and intermodule communication. Eliminating these complications results in a simpler specification. This specification can easily be modelled using graphs, one representing the flow of control and the other representing the flow of data. Functions of the protocol are identified using the flow graphs [14]. The transformed single-module specification containing single path transitions is called normal form specification and its transitions are called normal form transition (NFT).

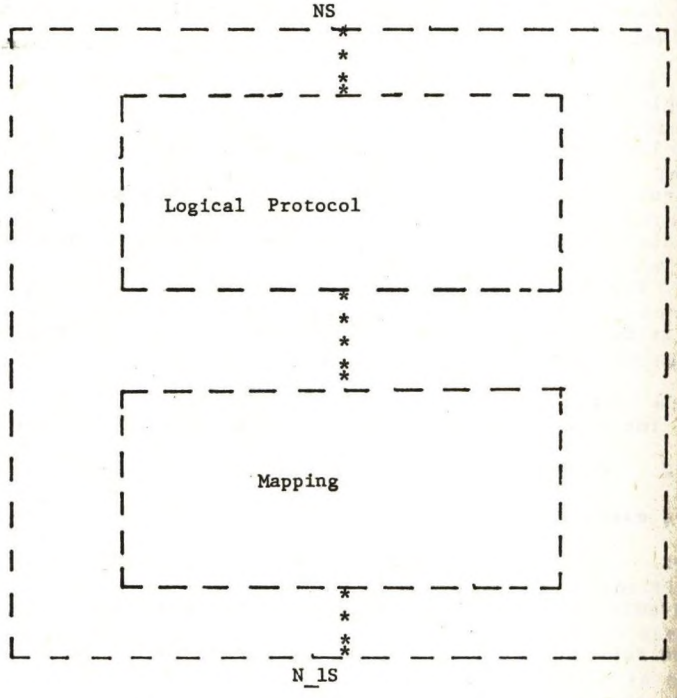


Figure 2.1. A Protocol Entity with Two Modules

```
(in Logical Protocol)
WHEN    NS.CONNECT_req
FROM idle
TO    connecting
PROVIDED    A and  (B    or   C)
BEGIN
    if  D  then  S1
         else         S2;
    out mapping.  event1 (CONNECT_req.  X+2)
END;

(in  Mapping)
WHEN  Logical Protocol.event1 (data_unit, length)
PROVIDED data_unit = CONNECT_req and length <= 10
BEGIN
    S3;
    out   N_1S.CONNECT_pdu
END;
```

Figure 2.2. Two Sample Transition Types in Estelle.

3. TRANSFORMATIONS AND COMBINING MODULES

We apply certain transformations on various Estelle constructs. State expressions on FROM/ TO clauses are eliminated and conditional and loop statements in the BEGIN block are removed. These transformations are done separately on all transition type of all the modules. Modules are combined by making textual substitutions. All these are explained in the following subsections and finally the flow graphs are defined.

3.1. Transformations

State expressions such as

state <> closed

can be used in FROM clause and (seldomly) TO clause. They are eliminated by generating more than one normal from transition corresponding to one transition type for every possible state value (state values other than "closed" in the above example).

To remove conditional and loop statements and local procedure calls we adapt the techniques from symbolic execution of sequential programs [9]. The idea is to create a new transition for every distinct

path in the BEGIN block and to modify the PROVIDED clauses to reflect t conditions imposed from taking these paths.

Local procedure calls in the BEGIN block are translated b symbolically executing the local procedure body. IF and CASE statement are removed by generating a NFT for each path they define. As an example we remove the IF statement in the first transition type of Figure 2.2:

```
WHEN  NS.  CONNECT_req
FROM  idle
TO    connecting
PROVIDED  A  and  (B or C) and  D
BEGIN
          S1;
...

WHEN  NS.  CONNECT_req
FROM  idle
TO    connecting
PROVIDED  A  and  (B or C) and  not D
BEGIN
          S2,
...
```

The WITH clause in the BEGIN block (as in Pascal) is eliminate by concatenating the record name with the proper variable names. The loo statements are eliminated by repeating the body of the loop for ever index variable value. We assume that the local procedures are no recursive and the specification does not contain any loops with variabl bounds. This assumption holds true in all the specifications w considered (see Section 4).

3.2. Combining Modules

Modules in an Estelle specification communicate by generatin internal interactions in output statements of their BEGIN blocks. It i assumed that in the referred modules, transitions are define corresponding to the internal interactions. Such a transition type i called an internal transition type. Also, we assume "rendez-vous" typ communication where the first module ready to execute the interaction mus

wait for the second to be ready. To obtain single-module specifications internal transition types should be eliminated. This elimination can be done by body replacements of the internal transition types in all other transitions containing an output statement referring the internal interaction. Symbolic replacements for parameter values of the input interaction are also done. Since the internal transition type can have its enabling predicate, this predicate is added to the PROVIDED clauses of the transitions where body replacement is done as a conjuction. After all the replacements are done, the internal interactions can be eliminated from the resulting specification.

Two facts complicate the elimination of the internal transitions: More than one transition may be specified or generated after the transformations of Section 3.1 for a given internal input and internal transitions may in turn contain output statements to other modules. We develop an algorithm to combine modules of an Estelle specification in two steps:

Step 1.

Eliminate intermodule interactions in the internal transitions.

Step 2.

Eliminate intermodule interactions in WHEN and spontaneous transitions.

In order to facilitate the processing in Step 1 we link together the internal transitions corresponding to a given internal input. Figure 3.1a shows an example where there are three internal inputs (A,B,C) and three internal transitions for A (also for B) and two internal transitions for C are linked. Internal output statements in some of these transitions are indicated by single letters (A,B,C) on the body of the transitions. The end of each list of transitions is indicated by the symbol "^".

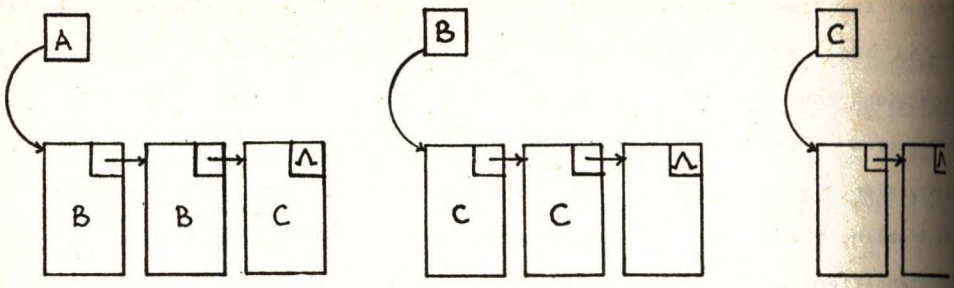
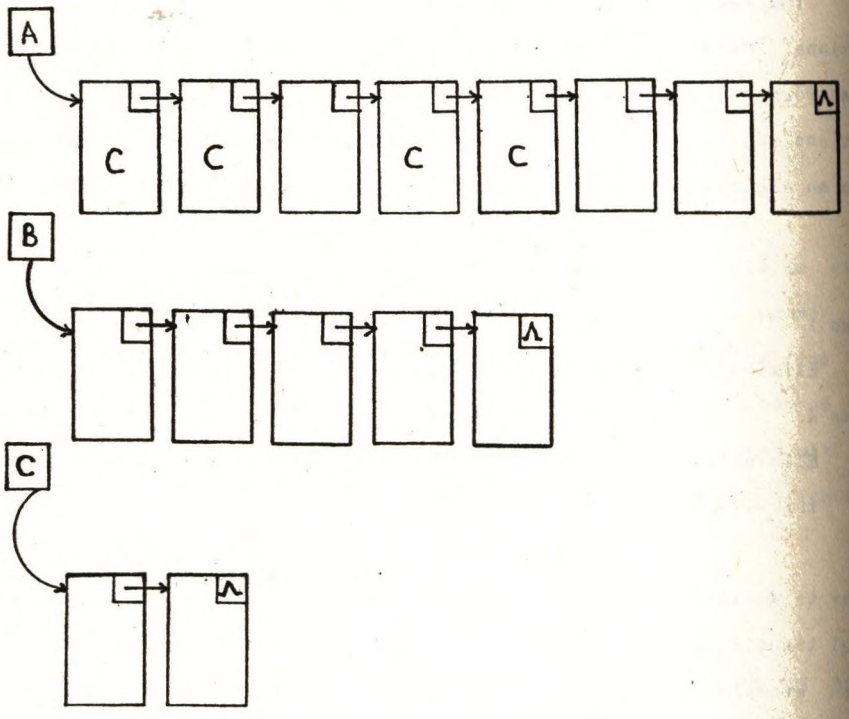
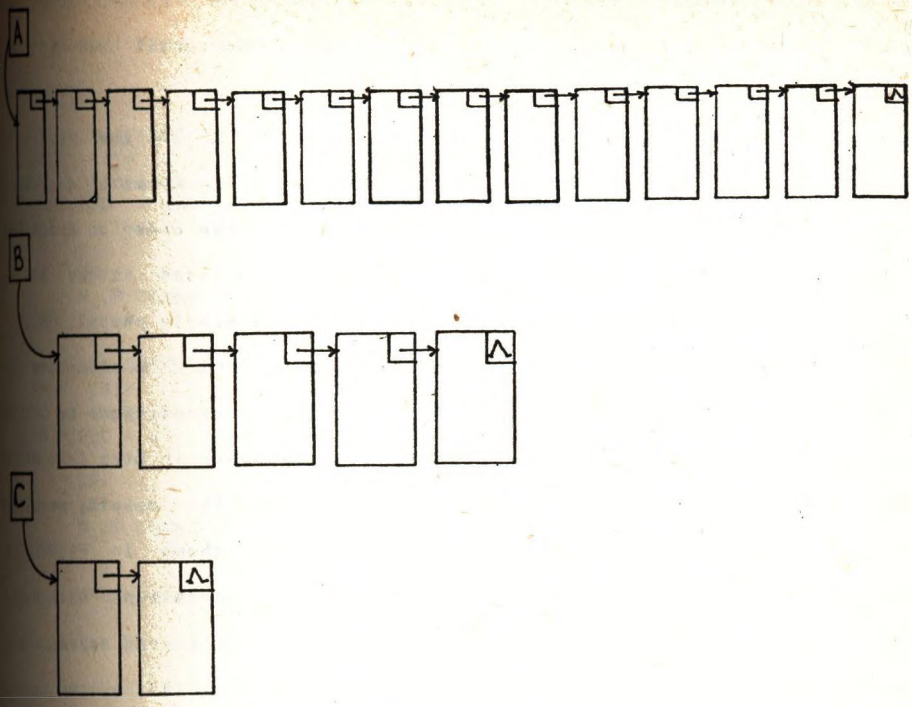


Figure 3.1a Linked-List Representation of Internal Transitions



After Phase 1

Figure 3.1b Resulting Linked-List Representation from Step 1



After Phase 2

Figure 3.1b Resulting Linked-List Represention from Step 1 (contd.)

The processing in Step 1 can be sketched as follows:

Each list is processed in order to eliminate internal communication. If there exists none, the list is passed to the next phase. The internal output statements are eliminated by finding the first corresponding transition and making necessary replacements as explained above. This way we create a new transition to be passed to the next phase. The other corresponding transitions (if any) are found using the list created for the internal input and the replacement process is repeated in the same transition but using a different corresponding transition. This process continues until all the transitions of the first

list are considered. Then other lists are processed. In the second (a possibly other) phases we continue in the same manner until all inter communication is eliminated.

As an example, we apply Step 1 to Figure 3.1a, the result shown in Figure 3.1b. Note that the efficiency (in terms of decreasi the number of phases) can be increased by changing the order of processi the linked-lists (for example, if B's were processed before A's Fig. 3.1a, the result would have been obtained in a single phase).

Step 2 uses the transitions obtained in Step 1 to eliminate t internal communication in the WHEN and spontaneous transitions by textu replacements of the corresponding transitions obtained in Step 1. As example, we eliminate the output statement to the mapping module Fig. 2.1, the resulting normal form transitions are shown in Fig. 3. During the replacement, the first elementary expression "data_unit CONNECT_req" is removed since it is satisfied and the second expression added to the PROVIDED clauses of the WHEN transitions as a conjunct after parameter replacements.

After Steps 1 and 2 we assign labels to all the NFTs obtained be used in the flow graphs as discussed in Section 3.4.

3.3. Conservation of the Semantics

It should be established that the syntactic transformations Section 3.1 and 3.2 preserve the semantics of the original transit types. In order to proceed with a formalism, the semantics of Esti should be well defined. More work can be done in this area possibly us the formalism of [4].

3.4. Flow Graphs

There are two types of flow in a normal form specification:

-Flow of <u>control</u> as the value(s) of the major state variable(s) (one each module of the protocol entity) changes, and

flow of data as the input primitive parameters change the values of the protocol variables and they in turn determine the output primitive parameters.

```
  WHEN  NS.CONNECT_req
  FROM  idle
  TO    connecting
  PROVIDED  A and  (B or C) and  D  and (X <= 8)
P1:BEGIN
      S1;
      S3;
      N_1S.CONNECT_pdu
  END;

  WHEN  NS.CONNECT_req
  FROM  idle
  TO    connecting
  PROVIDED A and  (B or C) and ~D and (X<=8)
P2:BEGIN
      S2;
      S3;
      N_1S.CONNECT_pdu
  END;
```

Figure 3.2 Normal Form Transitions of Figure 2.1.

3.4.1. Control Graph

The control graph (CG) is easily constructed from the FROM/ TO clauses of the NFTs by drawing an arc from the initial state to the next state. The arcs are identified by the corresponding labels of the NFTs. These labels are short-hand notations for

input/ouput

as used in the finite-state machines. The sequence of NFTs in correct state order may be obtained from the CG. These sequences starting and ending in the initial state are called subtours.

3.4.2. Data Flow Graph

A data flow graph (DFG) models the flow of information in a normal form specification, excluding major state changes. A DFG contains four types of nodes: I-nodes to represent input primitives, D-nodes to represent protocol variables and constants, O-nodes to represent output primitives, and F-nodes to represent data operations (functions). Arcs

are used to represent the flow as derived from the BEGIN block of the NFTs: A DFG for the assignment statement "remote_T_addr := determine_T_addr(remote_N_addr, calling_addr)" which appears in [11] is given in Fig. 3.3.

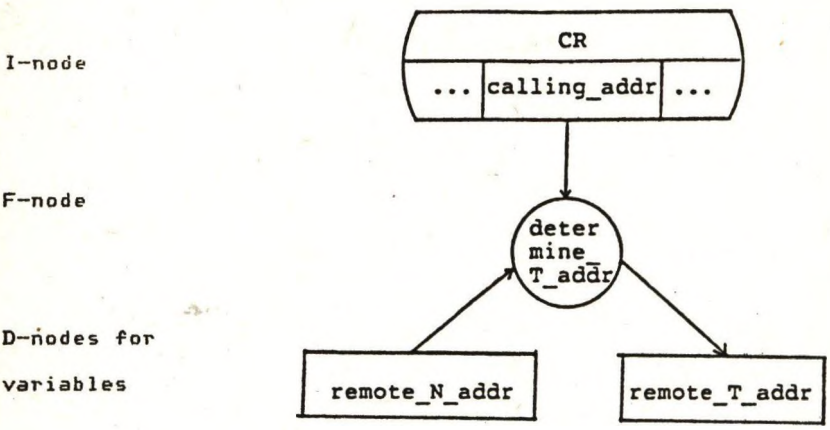


Figure 3.3. DFG for an Assignment Statement

4. Protocol Design Validation

In this section we will classify different validation aspects that involve the use of the flow graphs of normal form specification (see Section 3) and point out the errors found in the protocol specifications of the Class 0, Class 2 and Class 4 transport protocol (TP) ([10], [11] and [12]). Note that the work on the Class 4 TP is continuing so we will mention only partial results in this section.

We classify the errors that can be detected as syntactic and semantic, and discuss them separately in what follows. Finally design errors will be discussed.

4.1. Syntactic Errors

During the transformations phase (Section 3.1) syntactic errors of the type:

-undefined local procedures/functions,

-undefined variables,

-spelling errors in variable names

-missing parameters in procedure/ function calls

-missing parameters in output statements

-missing keywords such as BEGIN, END in the transition types,

-etc.

can be detected. These errors are easily detected by a compiler of the Estelle [7]. We have detected many errors of the above types in the class 4 TP specification.

The control graph (CG) can be used to detect missing assigments to the major state variables in normal form transitions (NFT). For example, in the Class 2 TP specification [11], the major state variable "NC_state" of the Mapping module is assigned to the value "closed" in none of the NFTs although some NFTs do close the network connection.

The data flow graph (DFG) can be used for detecting missing assignment statements to the protocol variables. After a DFG is constructed, some of the D_nodes may have no incoming arcs, these D_nodes represent missing assignments. The Class 0 TP specification [10] contained such a D_node, namely "TCEP"

Since variables used in a NFT need not be assigned in the same NFT, detecting missing assignments may not always be straight forward. Both CG and DFG should be used to detect such errors. The flow over each variable of a DFG is checked using the subtours: a variable assigned in a NFT of a subtour may be used in a NFT which follows the assigning NFT. Class 2 TP specification [11] has a missing assignment to the variable "local_T_addr" for user-initiated connections.

4.2. Semantic Errors

Errors that require knowledge of the protocol for their detection are called semantic errors. We briefly discuss two errors

detected in the specifications.

In the Class 2 TP specification, "credit_value" parameter of the CR input primitive is assigned to the variable "R_credit" in one of the NFTs. Unfortunately, however, "R_credit" is used for credits (number of data packets that can be sent before acknowledged) given to the peer entity, thus its value should not be determined by the peer entity.

In the Class 4 TP specification, the local procedure "set_max_size" is used for the dual purpose of deciding maximum values of "TSDU_fragment" and "TPDU_size" while "TSDU_fragment" size should depend on the user interface and "TPDU_size" should be decided by negociation during the transport connection establishment.

4.3. Design Errors

Sometimes, a specification may allow an implementation to indefinitely send a PDU creating channel overflow. This is the result of spontaneous transitions that can be executed as many times as desired provided that the protocol stays in the same major state. Such transitions are easily identified in a normal form specification. We call such a transition an indefinitely executable NFT (IENFT in short). In the remaining part of this section we classify IENFTs and discuss their consequences.

IENFTs can be divided into two classes: those that produce an output and those that do not produce any output. The former class allows the protocol to produce more than one consecutive PDUs carrying exactly the same information, i.e., parameter values. Similarly, the latter case allows the protocol to execute the BEGIN block of the spontaneous transition more than once assigning the same values to the same variables. If it can be assumed that the spontaneuos transitions of the above types will be implemented so that they will be executed only when they produce different outputs (the former case above) or when they assign different values to the variables in the BEGIN block (the latter case above) then

there is no design errors; otherwise the existence of such spontaneous transitions may represent a design error.

There are two spontaneous NFTs in the Class 2 TP specification that are IENFTs : one that produces an output (AK PDU) and the other that does not produce any output. The latter NFT assigns a value (representing available space in the "receive-buffer") to the variable called "R_credit" and the former that outputs an AK PDU with "R_credit" and "TR" (both R_credit and TR are modified in a WHEN transition receiving a DT from the peer entity). These NFTs can repeatedly assign R_credit with the same value and send AK PDUs carrying the same parameter values. Channel congestion due to consecutive AK PDUs is avoided by the flow control mechanism of class 2 TP.

5. Conclusions

A method was developed for simplifying protocol/ service specifications in Estelle. The simplified specification called normal form specification is obtained in two phases: transformations are applied on various language constructs to eliminate state expressions, conditional and loop statements, and modules are combined by textual replacements in order to eliminate internal communication. Normal form specification are modeled with flow graphs. Normal form specifications can be used to analyze the protocol design syntactically and semantically.

The transformation phase helps reveal various syntactic errors in a protocal specification. The flow graphs are used to detect syntatic errors such as missing assignments to the major state variable and other protocol variables. Semantic errors such as wrong assignments to the variables or misuse of the local procedures/functions are also detected. After these errors are corrected, the design validation can be completed by validating its communication content for problems such as deadlock, unspecified transitions, ect. using a method such as the one described in [8].

The flow graphs of the protocol specification help identify protocol functions there by simplifying protocol test design [14].

Transformations and combining modules need to be defined formally, leading to the proof of the conservation of the semantics in the normal form specification. More research is needed in this area. There is also a need for implementing the process of obtaining normal form specifications. The use of a very high level language tool as Prolog may result in a fast implementation. Eventually, the protocol implementers will be able to base their implementation to validated formal specifications and these specifications will become "reference" specifications.

Acknowledgements. This research was supported in part by the National Science and Research Council of Canada.

6. References.

[1] Fourth International Workshop on Protocol Specification, Testing and Verification, Y. Yemini (Ed.), June 11-14, 1984, Skytop Pennsylvania.

[2] Several papers in Special Issue on Open Systems Interworking, Proc. of the IEEE, Dec. 1983.

[3] Bochmann, G. v., et al., Use of Formal specifications for Protocol design, Implementation and Testing, in [1].

[4] Budkowski, S., Generation of a Global System Description From the Descriptions of Cooperating Subsystems, in [1].

[5] Bochmann, G. v., Concepts for Distributed Systems Design, Springer Verlag, 1983.

[6] Balzer, R., Goldman, N., Principles of Good software Specification and Their Implications for Specification language, in Specifications of Reliable Software Conference, Cambridge, MA, April 1979, 58-67.

[7] Bochmann, G. v., Gerber, G., Serre, J. M., Semi-automatic Implementation of communication protocols, Universite de Montreal, Doc. de Travail # 518, Dec. 1984. 20p.

[8] Jard, C., Bochmann, G. v., An Approach to Testing Specifications, Proc. of ACM SIGSOFI/SIGPLAN Software Eng. Symposium an Hign-Level Debugging, March 1983, pp. 53-59.

[9] Clarke, L.A., Richardson, D.J., Symbolic Evaluation Methods for Program Analysis, in Program Flow Analysis, Muchnick, S.S, Jones, N.D. (Eds.), Prentice Halls, 1981.

[10] Bochmann, G.v., Examples of Transport Protocol Specifications, ISO TC97/SC16/WG1 Subgroup B, April 1982.

[11] Bochmann, G.v., Example of a Transport Protocol Specification, ISO TC97/SC16/WG1 Subgroup B, Nov. 1982.

[12] NBS, Formal Description of the DIS 8073 Transport Protocol, ISO TC97/SC16/WG1 Subgroup B, 1984.

[13] Zave, P., The Operational Versus The Conventional Approach to Software Development, CACM, Vol. 27, No. 2, pp. 104-118, 1984.

[14] Sarikaya, B., Bochmann, G.v., Cerny, E., A Test Design Methodology for Protocol Testing, Proc. of 18th Hawaii Int. Conf. on System Sciences, Vol. 2, pp. 710-720, Jan. 1985.

[15] A FDT based on an Extended State Transition Model, ISO TC97/ SC16/WG1 Subgroup B, March 1984.

[16] Bochmann, G.v., et al., Experience with Formal Specifications Using an Extended State Transition Model, IEEE Trans. on Comm., Vol. Com-30, No.12, 1982, pp. 2506-2513.

NOTES

NOTES

SESSION 7
PROTOCOL TESTING EVALUATION AND MEASUREMENTS AND CONFORMANCE TESTING

STANDARDIZING CONFORMANCE TESTING FOR OSI

Dr. D. Rayner

Leader of the Protocol Standards Group
Division of Information Technology and Computing
National Physical Laboratory
Teddington, Middx. TW11 0LW, England

International standards for Open Systems Interconnection (OSI) services and protocols are well advanced. Complementary standardization work has begun for testing conformance of products to OSI protocol standards. This is drawing on about 6 years R & D work on techniques and tools for testing protocol implementations. This paper presents the major aspects of the testing methodology and framework being standardized by ISO. It relates some of the ideas to the work done in the R & D community and identifies several topics on which further work is needed.

0. Introduction

Open Systems Interconnection will not be completely achieved until systems can be tested in order to determine whether they conform to the relevant OSI protocol standards.

Standard test suites need to be developed for each OSI protocol standard, for use by the supplier or implementor in self-testing, by the user, or by the carrier or other third party tester. This should lead to comparability and wide acceptance of test results produced by different testers, and thereby minimise repeated conformance testing of the same system.

The standardization of test suites requires international definition and acceptance of a common testing methodology and appropriate testing methods. Thus, the International Organisation for Standardization (ISO) has begun work to produce a standard to define that methodology and to provide a framework for specifying conformance test suites. Although the project only received formal approval in February 1985, the current working draft [1] has been developed over 4 meetings by an enegetic and very representative group of people, many of whom have significant expertise in protocol testing. As a result, a large measure of stability and agreement has already been achieved. Moreover, it is being used and referenced by the ISO work on standardizing a test suite for X.25 DTEs.

This paper presents the major aspects of the methodology and framework and relates the abstract concepts to different ways of realising them in practice. It also identifies some major topics on which further work is needed. Sections 2 to 6 record the current agreed position in ISO, with the exception only of sections 2.5 and 6.2, although the selection and ordering of the material are the author's own. Sections 1, 2.5, 6.2 and 8 give the author's analysis of other aspects of the current state of ISO work on this topic. Section 7 is the author's own attempt to relate the ISO abstract test methods to the known techniques for realizing them.

1. Conformance Requirements

A prerequisite of ISO's work on OSI conformance testing was an understanding of the meaning of conformance in the context of OSI. Firstly it was decided that the term should be restricted to the conformance of an implementation or system to one or more protocol standards. It is incorrect to claim either conformance to the reference model or to service standards. Thus any conformance requirements related to the service must be stated in the relevant protocol standard(s).

The study of conformance revealed that there was much misunderstanding of the matter by protocol definers. In particular, it is important to distinguish two different types of conformance requirement: static and dynamic conformance requirements. Dynamic ones define the allowed behaviour in instances of communication, i.e. what a protocol implementation does; they constitute the bulk of a protocol specification. Static ones, on the other hand, define the allowed minimum capabilities of an implementation, i.e. what a protocol implementation contains; they are often only found in special conformance clauses.

Another important distinction is between mandatory requirements, conditional requirements, options and prohibitions. Each of these can apply to either static or dynamic conformance so it is important that protocol standards should state which is which. For example, it is possible to have a protocol procedure whose use in a particular instance of communication is optional but whose support by an implementation is mandatory, or indeed vice versa.

There may also be requirements on what a supplier or implementor should state about an implementation. In any case, for the purposes of conformance testing, a statement is needed of the capabilities and options which have been implemented. Such a statement is called a protocol implementation conformance statement (PICS). It should be consistent with the static conformance requirements in that it should include all mandatory capabilities plus all additional capabilities required as a result of the options which are supported.

All these concepts are covered in a draft answer to the question of conformance [2] which is being voted on as an approved interpretation of the OSI reference model. In addition, a checklist has been produced [3] to assist protocol definers to get a clear unambiguous statement of the desired conformance requirements. This was based largely on the UK input of a published conference paper [4]. It is expected to be progressed as an annex to the conformance testing methodology and framework standard [1].

2. Types of Testing

In principle, the objective of conformance testing is to establish whether the implementation being tested conforms to the specification in the relevant standard. Practical limitations make it impossible to be exhaustive, and economic considerations may restrict testing still further.

Four types of conformance testing have been identified, according to the extent to which they provide an indication of conformance:

- basic interconnection tests, which provide prima facie evidence that an implementation under test (IUT) conforms;

- functional range tests, which determine the capabilities of an IUT, i.e. which features and options are supported;

- conformance tests, which endeavour to provide as comprehensive testing

as possible over the full range of requirements specified by the standard;

- conformance resolution tests, which provide a definite yes/no answer in the context of specific conformance issues.

2.1 Basic Interconnection Tests

These provide limited testing of the main features in a standard, to establish that there is sufficient conformance for interconnection to be possible, without trying to perform thorough testing.

Basic interconnection tests are appropriate:

- for potentially detecting severe cases of non-conformance;
- as a first filter before undertaking more costly tests;
- to give a prima facie indication that an implementation which has passed full conformance tests in one environment still conforms in a new environment (e.g. in a multi-layer implementation, to check that a tested (N-1)-implementation has not undergone any severe change when linked to the (N)-implementation);
- for use by users of implementations, to determine whether the implementations are usable for communication at all with other conforming implementations, e.g. as a preliminary to data interchange.

Basic interconnection tests are inappropriate:

- as a basis for claims of conformance by the supplier of an implementation;
- as a means of arbitration to determine causes for communications failure.

Basic interconnection tests should be standardized as a very small subset of a full conformance test suite.

2.2 Functional Range Tests

Functional range tests determine which options and features are supported by an IUT at various levels of detail, from which major subsets or classes are supported to the range of values supported for a particular parameter. They can include not only tests to determine which protocol data units (PDUs) are supported, but also ones to determine which service primitives are supported.

Functional range tests can be used to:

- check the validity of the protocol implementation conformance statement made for an IUT;
- check that the static conformance requirements are met;
- enable an efficient selection of other conformance tests to be made for a particular IUT.

Functional range tests should be standardized as a subset of a full conformance test suite.

2.3 Conformance Tests

Conformance tests are intended to provide as thorough testing of an implementation as is practical, over the full range of requirements specified in a standard. Since the number of possible combinations of events and timing of events is infinite, such testing cannot be exhaustive. There is a further limitation, namely that these tests are designed to be run collectively in a single test environment, so that any faults which are difficult or impossible to detect in that environment can be missed. Therefore, it is possible that a non-conforming implementation passes the conformance test suite; one aim of the test suite design is to minimise the number of times that this occurs.

It is reasonable to regard an implementation as conforming if it satisfies the conformance test suite, so long as there is no evidence to the contrary.

Conformance tests are appropriate:

- as a basis for claims of conformance, so long as other tests or live usage have not revealed contrary evidence;

- as a basis for procurement.

Conformance tests are inappropriate:

- for resolution of problems experienced during live usage or where other tests indicate possible non-conformance even though the conformance test suite has been satisfied.

Conformance tests should be standardized.

2.4 Conformance Resolution Tests

These provide diagnostic answers, as near to definitive as possible, to the resolution of whether an implementation satisfies particular requirements. Because of the problems of exhaustiveness noted above, the definitive answers are gained at the expense of confining tests to a narrow field.

The test architecture and test method will normally be chosen specifically for the requirements to be tested, and need not be ones that are generally useful for other requirements; they may even be ones that are regarded as being unacceptable for generally specified conformance tests, e.g. involving implementation-specific methods using, say, the diagnostic and debugging facilities of the specific operating system.

The distinction between conformance tests and conformance resolution tests may be illustrated by the case of an event such as a Reset. The conformance tests may include only a representative selection of conditions under which a Reset might occur, and may fail to detect incorrect behaviour in other circumstances. The conformance resolution tests would be confined to conditions under which incorrect behaviour was already suspected to occur, and would confirm whether or not the suspicions were correct.

Conformance resolution tests are appropriate:

- for providing a yes/no answer in a strictly confined and previously identified situation (e.g. during implementation development, to check whether a particular feature has been correctly implemented, or during operational use, to investigate the cause of problems);

- as a means for identifying and offering resolutions for deficiencies in a

current conformance test suite.

Conformance resolution tests are inappropriate:

- as a basis for judging whether or not an implementation conforms overall;

- as a condition for procurement.

Conformance resolution tests need not be standardized.

2.5 Other Types of Testing

In addition to conformance testing, other types of testing have been proposed [5] including:

- performance tests to measure the performance characteristics of an IUT, such as its throughput and responsiveness under various conditions;

- robustness tests to determine how well an IUT recovers from various error conditions.

There may be some overlap between these types of test and conformance tests and where there is then such tests could be extracted as a subset of a full conformance test suite. However, where the test purpose is not concerned with conformance, then such tests fall outside the current scope of standardization.

3. Test Suite Design

The ISO framework for conformance testing includes guidance on the design of conformance test suites. A test suite will be structured into test groups, possibly substructured into subgroups. These will consist of a set of related abstract test specifications, each with its own test purpose, which should be precise and detailed so that it is clear what failure would mean. There will also be appropriate instructions for determining how to perform test selection and decide on the order in which the tests should be run.

Each abstract test specification will consist of a tree of test events (e.g. the sending or receipt of a service primitive or PDU) structured into test steps. A test step may be shared by many abstract test specifications. In order to run a test, the abstract test specification has first to be converted into an executable test definition appropriate for some specific testing system; for the tests are standardized at an abstract level, independent of any particular way of executing them.

A full conformance test suite, for a particular protocol, should be capable of testing all mandatory and optional features over the maximum range of parameters and variations.

The design of a test suite must take a number of factors into consideration, and should result in the optimum combination of them. The factors relevant for single-layer conformance test suites are itemized below.

3.1 Features

A full test suite should be capable of testing all the mandatory and optional features of the protocol, selected in all the feasible combinations allowed by the static conformance requirements of the protocol specification. If the specification allows for selecting a partial set of features, such as 'receive only', the test suite must be adaptable to this situation.

3.2 Protocol Phases

A full test suite will allow the selection, for each function of the protocol, of specific tests of the major protocol phases, in various combinations:

- connection establishment, when a connection is being set up;
- data transfer phase, while a connection is current (or at all times for a connectionless protocol);
- connection release, when a connection is being broken.

These phases are those used in OSI service and protocol descriptions. There may be subphases (half-open connection, data transfer awaiting acknowledgement, non-connection requests in the idle state, etc.).

3.3 Variations

The full test suite must include a range of variations in the following domains:

- sequence variations (the order in which PDUs occur);
- timing/timer variations;
- PDU encoding variations;
- parameter variations in PDUs.

Possible choices are suggested below:

- support of all PDU types and all structural variations of each type;
- 'normal' or default values for each parameter on each PDU;
- boundary values plus at least one mid-range value for each integer parameter;
- for bitwise parameters, as many values as is practical, but not less than all of the 'normal' or common values;
- for interdependent pairs of PDU parameters, 'critical' value pairs (representing multi-dimensional boundaries) and one 'normal' value pair;
- all sequences of PDUs where one PDU is 'out of sequence' with respect to the defined protocol;
- at least one invalid PDU type;
- at least one invalid value for each PDU parameter, where such invalid values exist;

- all defined timers should be exercised, i.e. allowed to expire at least once.

3.4 Valid/Invalid Behaviour

A test suite will check that the behaviour of the IUT is valid under all the above variations. Invalid and unexpected outputs from the IUT must be detected and given a specific branch in the abstract test specification to end the test and be identified.

Consistent reaction of the IUT upon receipt of invalid PDUs from the tester must also be exercised. Single erroneous PDUs should be mapped into correct (N-1)-service primitives.

Invalid PDUs are obtained by extending the variations (3.3) beyond the valid ranges.

3.5 Interdependence

The design of a test suite must consider test interdependence. Some tests must be successfully completed before others are attempted. For example, before testing detailed functional behaviour related to data transfer, the tests of connection establishment and transition to the data transfer phase should be successfully carried out.

Another type of dependency is the inter-functional dependency. Before selection of tests of data transfer, the capability to negotiate data length must be tested and established.

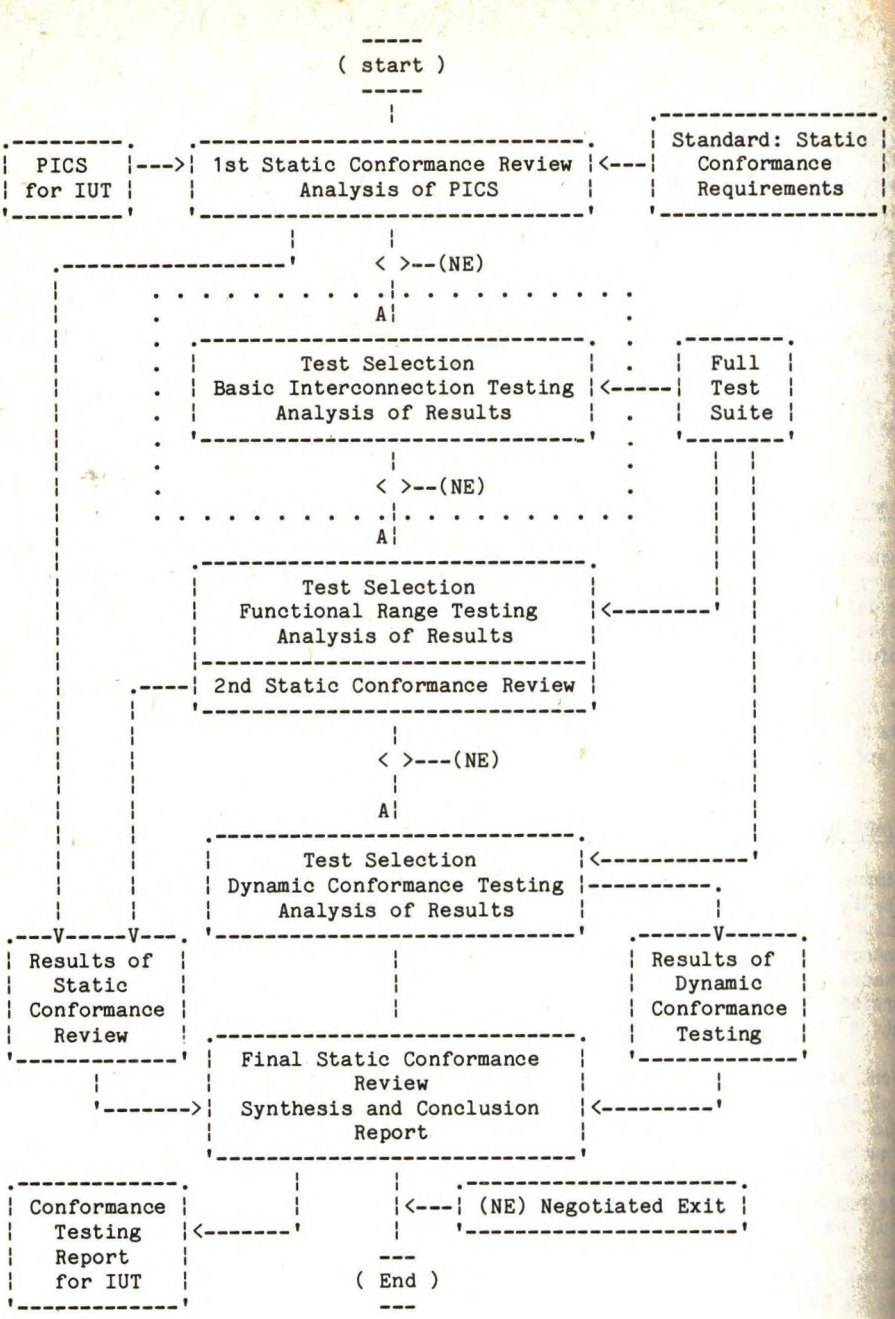
4. Typical Use of a Conformance Test Suite

It is important to understand how conformance testing should relate to static and dynamic conformance requirements and to the protocol implementation conformance statement. There are many possible ways of interleaving dynamic conformance testing with reviews of the information obtained about static conformance which can then affect subsequent test selection. However, ISO has illustrated the typical use of a conformance test suite in the form of a flow-chart, to clarify the main interrelationships. This is shown in Figure 1.

The first step, static conformance review, is a paper analysis, in which the protocol implementation conformance statement accompanying the IUT will be analysed for its own consistency, and its consistency with the static conformance requirements specified in the protocol standard(s) to which the IUT is claimed to conform. This paper analysis can be followed by one or two steps of live testing. The first is basic interconnection testing which is optional, and would be used detect severe cases of non-conformance and to determine whether more extensive testing is going to be worthwhile. The second is functional range testing, which will ascertain the validity of the PICS with respect to the actual, observable capabilities of the IUT.

A second static conformance review will combine the results of the functional range tests with the results of the first review.

The fourth step, dynamic conformance testing, will concentrate on live testing, checking the correct behaviour of the protocol implementation. State transition control, syntactic checking of the protocol elements, and behaviour of the implementation are in the scope of this phase. The behaviour will be tested in various instances of communication, both simple and complex, independent and



NOTE: A: Acceptable Results

Figure 1. Typical use of a conformance test suite

dependent on the environment, normal and erroneous.

It cannot be proved by testing that an implementation conforms dynamically in all instances of communication. However, it can be shown by testing that an implementation consistently conforms dynamically in representative instances of communication.

Two phases of 'paper analysis' will take place, one before and the other after the live testing phase. The first will be a phase of preparation/selection of the test suite(s) to be used, based on the results of the second static conformance review. The second will be a phase of analysis of the results of the live testing from the dynamic conformance point of view.

The final static review involves a synthesis of the results of the dynamic conformance tests with those of the second static conformance review. A conclusion on the conformance of the IUT to the requirements of the standard(s) can then be reached. This conclusion is recorded in a Conformance Testing Report detailing the tests run, their results, and the environmental conditions.

Provisions for "negotiated exits" can be seen in Figure 1. They are points where the concerned parties can decide that the results of the previous step are not good enough to justify continuing the tests.

5. Abstract Testing Methodology

5.1 Control and Observation and Abstraction

The ISO abstract testing methodology is based upon the OSI reference model. Abstract test methods are described in terms of what outputs from the entity under test are observed and what inputs to it can be controlled. More specifically, an abstract test method is described by identifying the points closest to the entity under test at which control and observation are to be exercised.

The OSI protocol standards define allowed behaviour of a protocol entity (i.e. the dynamic conformance requirements) in terms of the PDUs and both the abstract service primitives (ASPs) above and below that entity. Thus the behaviour of an (N)-entity is defined in terms of the (N)-ASPs and (N-1)-ASPs (the latter including the (N)-PDUs). Each of these two sets of interactions could be observed and controlled from several different points, directly or remotely, as identified in Figure 2.

Note that the possible points of observation and control are identified by three factors:-

(a) whether it is the ASPs or PDUs which are observed and controlled;
(b) the layer identity of the ASPs or PDUs concerned;
(c) whether they are controlled and observed within the system under test or in a system remote from the system under test - if the latter then the ASPs or PDUs are distinguished by the addition of a double-quote character (").

Complete control and observation of the (N-1)-ASPs (or (N-1)-ASP"s) will include control and observation of the (N)-PDUs (or (N)-PDU"s), but not vice versa.

It is possible that the (N)-ASP activity of the entity under test might not be controllable nor observable, directly or indirectly, in which case this activity is said to be hidden. It is, however, assumed that the (N-1)-ASP activity will at least be indirectly observable and controllable, via some real means of communication (e.g. via (N-1)-ASP"s). Thus when the (N-1)-ASPs are not

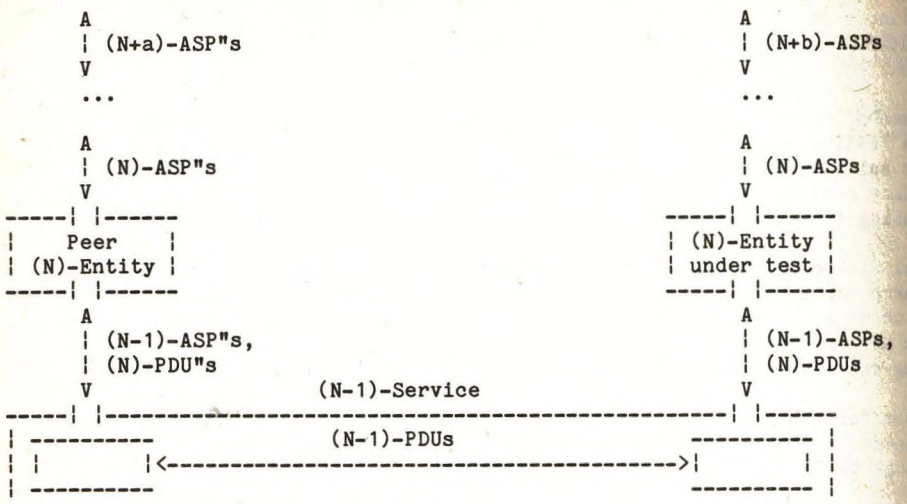


Figure 2. Possible points of control and observation

controllable nor observable directly, conformance testing can only be carried out if the (N-1)-service is provided sufficiently reliably for control and observation to take place remotely.

It is important to make the distinction between the interactions which are controlled and observed, and the subset of those observations on which judgement is passed. Judgement can only be passed on whether or not the conformance requirements are being met. This will usually apply to the PDUs exchanged rather than the particular realisations of ASPs. However, the reason for wishing to control ASPs is to try to determine more precisely which PDUs should be exchanged in a particular test.

The virtue of expressing control in terms of ASPs is that this is an abstract form of specification which does not unduly limit the freedom of testers to implement the tests in different ways using whatever interfaces are accessible externally, provided that the required degree of control and observation is available.

Associated with this abstract view of control and observation, there needs to be a correspondingly abstract view of the testers which perform the control and observation. They are referred to as the lower and upper testers. The lower tester is the means for providing control and observation of events which approximate to what the IUT sees at its lower SAP (i.e. the (N-1)-ASPs, (N-1)-ASP"s, (N)-PDU"s, etc). The upper tester is the means for providing control and observation of events which approximate to what the IUT sees at its upper SAP (i.e. the (N)-ASPs, (N+1)-ASPs, etc). In addition, there is the concept of test coordination procedures which are the rules for cooperation between upper and lower testers during testing.

5.2 Test Method Overview

Conformance test suites for a given OSI protocol or set of protocols should be defined for each of a limited number of abstract test methods. Testers would then be expected to select the most appropriate suite for the IUT or system under test (SUT). The chosen abstract test methods fall into three main categories:-

(1) Local test methods: which use control and observation of the ASPs directly above and below the IUT;

(2) Distributed test methods: which use control and observation of the (N-1)-ASP"s together with control and observation of the ASPs directly above the IUT;

(3) Remote test methods: which use control and observation of the (N-1)-ASP"s, with the ASP activity directly above the IUT being hidden.

The abbreviated name for an individual abstract test method begins with an 'L', 'D' or 'R', to denote Local, Distributed or Remote respectively, according to the main category to which it belongs. These categories can be applied to testing a single-layer at a time, in which case the second letter of the abbreviated test method name is an 'S', or they can be applied to testing all layers of a multi-layer IUT, in which case the second letter of the abbreviated name is an 'M'. Finally, single-layer test methods are qualified by the adjective 'embedded' when they apply to a single-layer within a multi-layer IUT, such that the control and observation is less direct than it would be for the corresponding method of testing a single-layer IUT; for such methods an 'E' is added as a third letter in the abbreviated name.

With this naming scheme for abstract test methods, the following methods are proposed:-

LS - Local single-layer test method
LM - Local multi-layer test method
DS - Distributed single-layer test method
DSE - Distributed single-layer embedded test method
DM - Distributed multi-layer test method
RS - Remote single-layer test method
RM - Remote multi-layer test method

5.3 The Local Single-layer Test Method

The LS test method is illustrated in Figure 3.

The tests defined for this method would provide the base from which corresponding tests for any other abstract test method could be derived. They also provide a basis for deriving multi-layer tests.

They are useful to implementors for "unit testing", that is the testing of an implementation of a protocol entity in isolation from the rest of the system.

However, since this method involves no communication over an (N-1)-service-provider, the protocol entity may require further testing using another method which uses an (N-1)-service-provider. It may, however, be satisfactory to carry out a conformance test suite using the LS test method and then follow it up with a confidence check by running a basic interconnection test suite using the DS, DSE or RS test method.

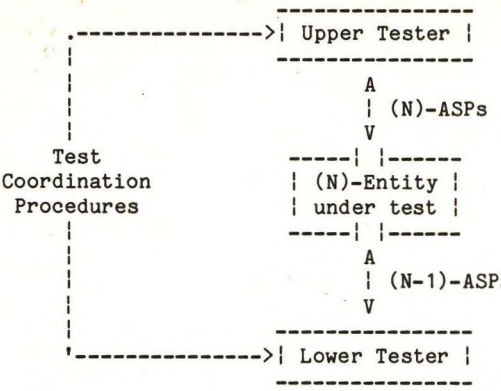


Figure 3. The local single-layer test method

5.4 The Distributed Single-layer Test Method

The DS test method is illustrated in Figure 4.

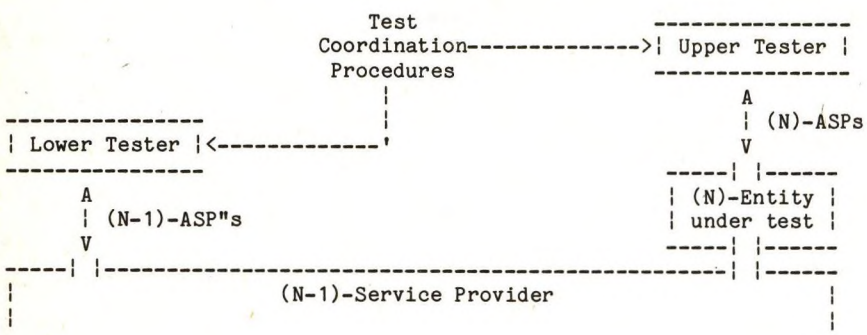


Figure 4. The distributed single-layer test method

When applicable, this method provides the most complete form of single-layer testing over the (N-1)-service-provider. It permits a large number of (N)-PDU errors and unusual but valid (N-1)-ASP"s to be included in the tests.

The tests defined for this method could be realised in practice in many different ways, provided that the system under test can be made to provide some realisation of the (N)-ASPs. However, they would not constrain the form of implementation of the upper and lower testers.

The upper tester will not be purely passive since it needs to exercise some control over which (N)-ASPs are generated, but it will not have the same scope for introducing errors as the lower tester.

Higher-layer forms of describing the tests (e.g. in terms of (N)-ASP"s) should

be used where this will reduce the size of the test specification without loss of required precision. This will enable identification to be made of those tests which are equivalent to ones for a method involving control and observation of (N)-ASP"s instead of (N-1)-ASP"s. Those who wish to use this subset of the test suite should be warned that it is not adequate as a complete conformance test suite; although it could be used as a set of basic interconnection tests. The problem is that not all (N-1)-ASP"s can be invoked merely by control of (N)-ASP"s; e.g. Network Reset cannot be invoked by control of Transport Service Primitives.

5.5 The Remote Single-layer Test Method

The RS test method is illustrated in Figure 5.

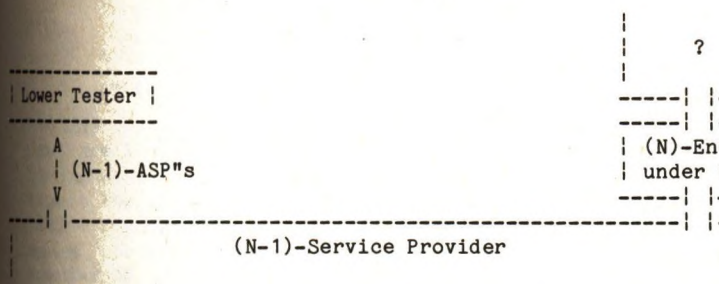


Figure 5. The remote single-layer test method

Tests for this method should be possible to use with all systems under test. It places no additional requirements for the sake of testing on the implementation other than those placed on it by the protocol standard. (N-1)-ASP"s are used, rather than just (N)-PDU"s, in order to allow tests to specify when the underlying connection should be set up or released.

This method is appropriate for those systems in which control and observation of (N)-ASPs is not possible.

5.6 Multi-layer Test Methods

Multi-layer testing consists of testing all the layers of a multi-layer IUT as a whole, without accessing the inter-layer interfaces within the IUT. This type of method could be used when these inter-layer interfaces are not accessible, and also when the combined allowed behaviour of the multi-layer implementation must be known.

The LM test method is similar to the LS test method but the IUT goes from (N-1)-ASPs up to (N+n)-ASPs and therefore the upper tester controls and observes (N+n)-ASPs. The same transformation turns the DS test method into the DM test method. In terms of control and observation, the RM test method is the same as the RS test method. However, in all three cases, the lower tester will need to deal with (N)-PDU"s up to (N+n)-PDU"s.

5.7 Incremental Embedded Single-layer Testing

Incremental embedded single-layer testing permits testing of multi-layer implementations layer by layer, from (N) up to (N+n), without requiring the access to interfaces for each layer within the implementation. This is applicable to both distributed and remote test methods. The distributed approach is illustrated in Figure 6. For a general layer (N+i) the DSE method uses control and observation of (N+i-1)-ASP"s and (N+n)-ASPs. Note that for the top layer in the multi-layer IUT, (N+n), the ordinary DS test method is used.

In the same way, the RS test method can be used incrementally for each layer of the IUT.

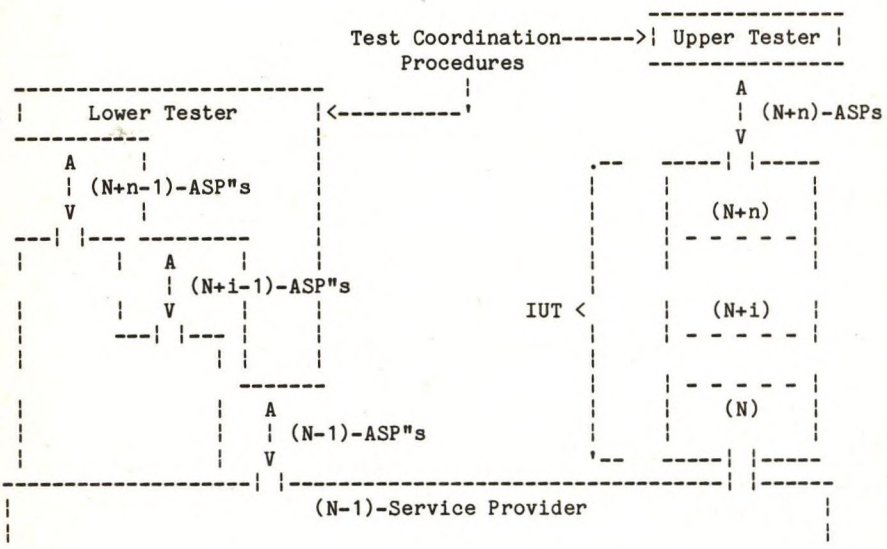


Figure 6. Incremental use of the DSE test method

Use of the DSE method on successive layers of a multi-layer IUT seems to be a more practical approach than using the DM test method to test the same IUT. Similarly, use of the RS method on successive layers of a multi-layer IUT would seem to be better than using the RM method. The main advantage of incremental embedded single-layer testing is that it requires less complicated test specifications.

5.8 Test Methods for Network Relay Systems

The abstract test methods discussed above all apply to testing protocol implementations in end-systems. Different methods are required for testing Network Relay Systems. Two have been proposed by SC21 and accepted by SC6/WG5. These are abstractions of the methods which have been used in practice at NPL [6]. They are as follows:-

(a) The case of testing a relay system from one subnetwork (loop-back testing).

This test configuration is shown in Figure 7.

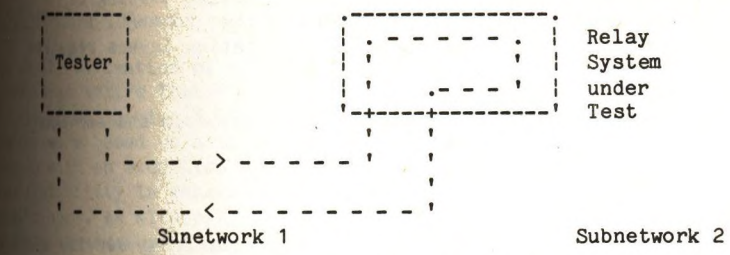


Figure 7. Loop-back Testing Configuration

For this testing configuration, one tester controls the test connections which are looped within the relay system. Therefore, it does not need two testing systems on different subnetworks and the problem of synchronization between the two testers is avoided. This testing configuration involves control and observation of PDUs on two connections on one side of the relay system, assuming that proper arrangements can be made for looping them together on the other side.

(b) The case of testing a relay system from two subnetworks (transverse testing).

This testing configuration is shown in Figure 8.

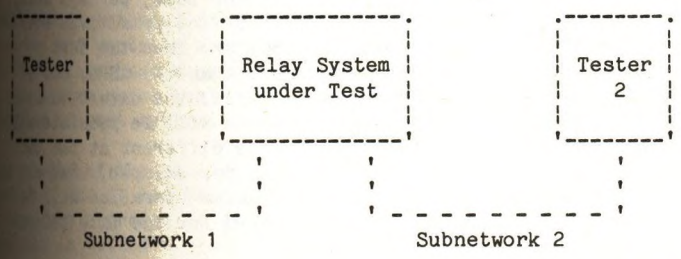


Figure 8. Transverse Testing Configuration

This testing configuration involves control and observation of PDUs on both sides of the relay system. If this test method is applied to testing the routing and relaying functions of the relay system, each of the testers may be required to simulate two or more end-systems.

Since Services below the Network Service are generally either unavailable or not well-defined, due to the fact that most protocols for real subnetworks were defined before Physical or Link Services were thought of, it is generally felt that the control and observation for Network layer testing needs to be interpreted in terms of PDUs rather than ASPs.

6. Applicability of the Test Methods

These abstract test methods are all applicable to the active testing of Transport, Session and Presentation layers without further comment. Their applicability to the Physical, Link, Network and Application layers requires further comment.

Further development may be needed for connectionless protocols and services.

6.1 Lower Layers

In the lower layers (Physical, Link and Network), the underlying service will not be end-to-end and so testing using the distributed and remote methods will need to be carried out over a single link rather than between end-systems. Furthermore, as with testing Network relay systems, the idea of specifying tests in terms of underlying service primitives is not likely to be of practical value; instead the tests will need to be defined in terms of PDUs. This is partly because the Link and Physical Services are not directly related to the protocols in those layers; partly because there is no service under the Physical layer; and partly because the protocols are technology dependent and in some cases do not correspond to the OSI layers.

The fact that these layers tend to be implemented in hardware or firmware makes it likely that the emphasis will be on testing multi-layer IUTs for these layers. The exception is the use of the LS test method to test the design of the algorithm for a protocol before it is put onto a chip.

6.2 Application Layer

With the application layer, there is no layer service above it which can be controlled or observed via (N)-ASPs at a SAP. However, for some, perhaps most, application protocols there is a defined service with service primitives some or all of which may be controllable or observable. If they are then the test methods can be applied by using these primitives rather than (N)-ASPs, but if they are not then only the remote test methods can be used. The extent to which control and observation of application service primitives will be possible will vary from one application protocol to another and may be different at the two sides because of inherent asymmetry in the application. For example, what can be controlled and observed for an initiator of a file transfer operation will be rather different from what can be controlled and observed for the file transfer responder (i.e. the filestore end).

Another issue occurs in the application layer, that of testing conformance requirements which go beyond the pure protocol exchanges. It is not only necessary to know that PDUs are exchanged in a valid manner but also that the exchanges have the desired effects in terms of the service provided to the user. This may require the definition of special application protocol dependent test methods to complement those described above. For example, it may be necessary to inspect the contents of a filestore by local means before and after certain file transfer tests.

7. Realization of Abstract Test Methods

In order to run the tests that will in due course be standardized it will be necessary to realize at least one of the abstract test methods. This will involve making mappings between the abstract concepts and real events and real systems. This section identifies some of the possible ways of realising lower testers, upper testers and test coordination procedures by reference to

approaches proposed and used by various research and development groups.

7.1 Lower Testers

The usual realisation of a lower tester for remote and distributed methods is known as an Active Tester. There are several different possible designs [7,8] not all of which give full control and observation of (N-1)-ASP"s. Some early designs were based on using a reference implementation of an (N)-entity, perhaps coupled with an exception generator [9] or configuration module [10] to give some flexibility in generating desired patterns of behaviour, particularly invalid ones. An alternative is to use an encoder/decoder module [10,11,12] to generate any desired sequence of PDUs, whether valid or invalid. However, to overcome the deficiencies of these approaches, a second generation design has been proposed [8,13] which will enable control and observation to be specified in detailed (N-1)-ASP" terms when necessary, or in higher level terms (e.g. (N)-ASP"s) when that is more appropriate. In addition, it may be necessary to use a portable testing unit, such as Cerbere [10,14], as part of the realisation of the lower tester when the underlying service is not end-to-end.

Whatever means is chosen for generating and recognising the (N-1)-ASP"s, an Active Tester will also need some form of Test Driver to control the operation of the test according to the executable test definition. Some Test Drivers are manual, some take commands from a file, and some use finite state based techniques [15]. Future ones may well be based on standard formal description techniques. Whatever approach is used, it is important that the executable test definition should be able to cover the whole tree of possible behaviours specified in the abstract test specification, not just a few of the main paths through that tree.

7.2 Upper Testers

Local and distributed test methods require the use of an upper tester. This observes and controls service primitives within the system under test. There must therefore be a mapping between abstract service primitives and real events in the system under test, but these need only be known to the implementor or supplier of that system.

The details of the realization of the upper tester can be entirely system specific, but it has been found advantageous to have a portable system independent approach to implementing upper tester functionality. The usual approach is called a Test Responder, but there are many possible designs [7] which vary in the demands they make on the system under test and in the amount of control and observation that they can perform. Another approach which has been recently proposed [16] is to implement a portable module called a Ferry in the system under test, the purpose of which is to relay all the (N)-ASP events over the network to a Test Driver in the same system as the Active Tester. This avoids having to design a special Test Responder with its inevitable compromise between portability and flexibility, but does require the system under test to support a module (the Ferry) which uses two connections simultaneously.

It has been proposed that there should be standardization of a single design of Test Responder but this has not yet been accepted. Instead the current focus of attention is on the possibility of standardizing the general functionality required by an upper tester.

7.3 Test Coordination Procedures

The major problem to be solved by the test coordination procedures is the synchronization of activities by the upper and lower testers. Research work has shown that this is a serious problem with the distributed test method. Loose synchronization by means of remote terminal access, use of the telephone and the setting to appropriate timer values has been shown to be inadequate, as might have been expected.

A better approach is to use a Test Driver-Responder Protocol (or Test Control Protocol) designed in conjunction with a Test Responder. There are two main design choices to be made for such a protocol. The first is whether to operate it on the same connection as the test traffic, or on a separate parallel connection going through the IUT, or on a separate parallel connection which interfaces to the Test Responder without going through the IUT. The second choice is whether to make each PDU of this protocol relate directly to one test event or to have PDUs which set up a test step involving several events up to the next synchronization point. All these approaches have some technical inadequacies and some add considerably to the complexity of the Test Responder.

An alternative approach is available if the Ferry idea is used. Since the control of the upper tester is then realised by a Test Driver in the same system as the Active Tester, synchronization between upper and lower tester becomes merely a matter of inter-process communication within a single system. The weakness is that there is a delay between activity by the Test Driver and corresponding activity by the Ferry.

Synchronization is, of course, much easier to solve with the local test method, since the test coordination procedures only need to operate within a single system. It may also be thought that the remote test method avoids the problem entirely. This is not so because, although there is no defined upper tester activity to be coordinated with the lower tester activity, experience shows that some undefined, system specific control of the system under test will be needed; and this will have to be synchronized with the lower tester activity, but the option of using a protocol to do it is almost certainly not available. So it is back to remote terminal access and use of the telephone.

8. Topics Requiring Further Work

The main topics which require further work for the standardization of OSI conformance testing are:-

(a) test notation;
(b) relationship between testing and FDTs;
(c) timing considerations;
(d) analysis and comparability of results;
(e) principles for use and design of multi-layer test suites;
(f) production of test suites.

It is desirable to have a single preferred test notation in which abstract test specifications can be written. Currently four candidates are under study in SC21 and a fifth has been used for the X.25 test suite by SC6. The SC6 notation is an informal one which is only applicable to simple tests using the RS test method. The ones being studied by SC21 are supposed to be able to handle complex tests for any test method. However, the tabular method (which is also proposed for Teletex tests by CCITT) is better at concentrating on the main paths rather than specifying the whole tree of behaviour for a complex test. The other three are an informal tree notation and the two ISO FDTs, Estelle and LOTOS. A comparison of the four is being made by using them to define some complex example tests [17] for the Session and Transport protocols.

Other aspects of the relationship between FDTs and testing are seen as having lower priority. They concern the suitability of each of the ISO FDTs for derivation of tests from formal descriptions of protocols and services. Current evidence [18] suggests that LOTOS is better suited for this than Estelle.

There are two main aspects of timing considerations: conformance requirements concerning timing and timers and how these should be tested; and the use of timers in testing for such things as the detection of inactivity. There are currently a great many questions on these aspects and very few answers [19].

Very little has been contributed so far on analysis and comparability of results. Yet a major reason for standardizing tests is to ensure comparability of results between different organisations which carry out the testing.

There is general agreement that the testing of multi-layer implementations is of major practical importance, but most of the work to date has gone into developing the single-layer testing ideas. It is currently unclear to what extend multi-layer test methods will be used in practice or how multi-layer test suites might be defined. It may well be that incremental embedded single-layer testing will be the dominent technique.

As part of the work of evaluating the conformance testing methodology and framework, it is proposed that trial test suites should be developed, probably for Transport, Session and an application protocol. These could then form the basis of the full test suites for these protocols. However, much effort is required to produce such test suites and it remains to be seen how quickly this will be forthcoming. At present the only test suite which is being developed by ISO is the one for X.25. CCITT is also working on a test suite for Teletex.

9. Conclusion

The ISO work on OSI conformance testing is now well advanced and has successfully generalised the work done by various research groups. A lot still remains to be done, but the energy and enthusiasm of the experts engaged in this work enables one to be optimistic about the outcome.

Acknowledgements

The author gratefully acknowledges the efforts of the ISO rapporteur group on OSI conformance testing, which he leads, without which this paper would not have been possible. He also wishes to acknowledge the support given to him my many members of the NPL Protocol Standards Group and the National Computing Centre COMMS-AID team. His own work at NPL has been supported by the Electronics and Avionics Requirements Board of the Department of Trade and Industry.

References

[1] ISO/TC97/SC21/WG16-1, Working draft for OSI conformance testing methodology and framework, ISO/TC97/SC21 N 410, Paris, February 1985.

[2] ISO/TC97/SC21/WG16-1, Draft answer to the question on conformance to OSI standards (WG1 Q 16 A), ISO/TC97/SC21 N407, Paris, February 1985.

[3] ISO/TC97/SC16/WG1, Checklist for protocol standards to permit conformance testing (revised), ISO/TC97/SC21 N 88, Copenhagen, June 1984.

[4] D. Rayner, Towards an objective understanding of conformance, in: Protocol Specification, Testing and Verification III, proc. 3rd international

workshop on this subject, held in Zurich on 31 May - 2 June 1983, edited by H. Rudin and C.H. West, North Holland, 1983, 477-492.

[5] R.F.L. Henley and D. Rayner, Implementation assessment of Transport and Network Services: an informal description of tests, NPL Technical Memorandum DNACS TM 5/81, July 1981.

[6] H.X. Zeng and D. Rayner, Gateway testing techniques, in: Protocol Specification, Testing and Verification IV, proc. 4th international workshop on this subject, held at Skytop, Pennsylvania, on 12-14 June 1984, edited by Y. Yemini, R. Strom and S. Yemini, North Holland, 1985, 637-655.

[7] G.W. Cowin, R.W.S. Hale and D. Rayner, Protocol product testing - some comparisons and lessons, in: Protocol Specification, Testing and Verification III, edited by H. Rudin and C.H. West, North Holland, 477-492.

[8] J.R. Pavel and D.J. Dwyer, Some experiences of testing protocol implementations, in: Protocol Specification, Testing and Verification IV, edited by Y. Yemini, R. Strom and S. Yemini, North Holland, 657-677.

[9] R.J. Linn and J.S. Nightingale, Some experience with testing tools for OSI protocol implementations, in: Protocol, Specification, Testing and Verification III, edited by H. Rudin and C.H. West, North Holland, 521-531.

[10] J-P Ansart, CERBERE, GENEPI, STQ: Three tools for protocol implementation testing, proc. 1st international workshop on Introduction to High Level Protocol Standards for Open Systems Interconnection, held in Paris, on 27-29 June 1983.

[11] D. Rayner, A system for testing protocol implementations, Computer Networks, 6, December 1982, 383-395.

[12] J-P. Ansart, GENEPI/A - a protocol independent system for testing protocol implementation, in: Protocol Specification , Testing and Verification II, proc. 2nd international workshop on this subject, held in Idyllwild, California, on 17-20 May 1982, edited by C. Sunshine, North Holland, 1982, 539-554.

[13] J.R. Pavel, A new approach to the design and construction of protocol testers, NPL Report DITC 54/85, April 1985.

[14] J-P. Ansart and J. Damidau, CERBERE, a tool to keep an eye on high level protocols, in Protocol Specification, Testing and Verification II, edited by C. Sunshine, North Holland, 529-538.

[15] P.W. Hobson (ed.), Implementation assessment of the OSI Network Service: the test definition language, NPL Report DITC 52/84, December 1984.

[16] H.X. Zeng and D. Rayner, The impact of the ferry concept on protocol testing, to be published in proc. 5th international workshop on Protocol Specification, Testing and Verification, held in Moissac, Toulouse, France, on 10-13 June 1985.

[17] ISO/TC97/SC21/WG16-1, Test examples, ISO/TC97/SC21 N 412, Paris, February 1985.

[18] ISO/TC97/SC21/WG16-1, Conformance testing and FDTs, ISO/TC97/SC21 N 411, Paris, February 1985.

[19] ISO/TC97/SC21/WG16-1, Time dependencies and timers, ISO/TC97/SC21 N 413, Paris, February 1985.

THE USE OF LOTOS FOR THE SPECIFICATION AND THE AUTOMATIC GENERATION OF TEST SEQUENCES

Vincenza Carchiolo[1], Sergio Palazzo, Giuseppe Pappalardo[1]

CREI - Politecnico di Milano
P.za L. da Vinci, 32
20133 Milano - ITALY

In parallel with the development of the ISO standards for the OSI framework, interest is growing for testing methods and tools to assess implementations conformance with the standards and interworking capability. In order to give confidence in the results, testing should be objective and authoritative. A basic condition for this is that test sequences be automatically derived from ISO standards formalizations. To this aim, since conformance requirements must specify about the implementations to be tested nothing more than what can be externally observed, the ISO language LOTOS, which allows for describing temporal ordering of events, has been chosen. The paper deals with the use of LOTOS for specifying and generating test sequences in a protocol testing machine developed at CREI. It also discusses the problem of translating the mathematical model employed (viz. LOTOS expressions) into the data structures available in a programming environment. LOTOS is shown to be adequate also from this point of view, since it admits a natural mapping onto easy-to-implement trees. Finally, some topics are singled out for further study.

NTRODUCTION

he last years the development of computer networks has stimulated a clear nd for tools for debugging and testing communication software. This trend consequence of the peculiarity of communication software, that is obviously cted by the relevant testing tools and techniques; but, above all, it is to be ibed to the impact of standardization in the field. The International nization for Standardization (ISO) has undertaken a massive effort for ing standards for the Open Systems Interconnection (OSI). ISO standards ut conformance requirements with which an implementation must comply and quently are to be taken into account also in the design of a testing tool. A t account of OSI architecture, i.e. OSI view of computer networks environ- seems therefore to be necessary.

sual in up-to-date software engineering, OSI architecture is layered [1]; layer provides a service to the upper one by exploiting the service offered he lower one; a layer is distributed in nature, as it consists of remote ies that cooperate through the lower layer service according to a set of known as the layer protocol.

nteresting debate (see for example [2], [3], [4], [5], [6], [7]) has taken place e literature and within the ISO working groups about 'where' a protocol

---

V. Carchiolo and G. Pappalardo are currently with the Istituto di Informatica e Telecomunicazioni - Università di Catania - ITALY

entity implementation should be tested. Common to all approaches is the conclusion that the testing 'machine' must be distributed, to enable performance tests from a remote site. This is both for the sake of convenience (avoiding moving software and/or hardware) and because, generally, the implementation under test (IUT) has no accessible lower test points and therefore can be accessed below only through the link represented by the underlying layers which are assumed to be reliable.

In Italy CREI (Centro Rete Europea Informatica) in cooperation with CSELT (Centro Studi e Laboratori Telecomunicazioni) started at the beginning of 1983 a project aimed at realizing a software testing machine. In the approach chosen for the early version of the machine [8], testing operations on the IUT, a 4th layer (Transport layer) entity, are performed at the Transport layer upper interface viz. the Transport service [9]; the configuration of the testing system can be observed in Fig. 1.

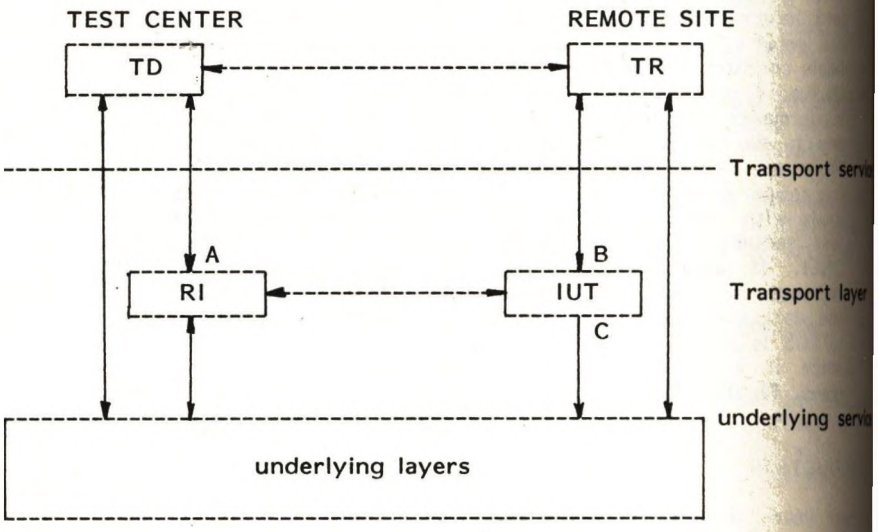


Figure 1. Architecture of the testing machine.

A Reference Implementation (RI) is provided at the test centre, so that a "live" test can be performed on the IUT, while it is interacting with a peer entity trough an underlying service. If the IUT were accessible from the layers above and below, at B and C, a test would obviously consist in inputting given stimuli at B and C and checking the responses; this can be done indeed at B, but since it is reasonable not to rely on access at C, it has been decided to input the relevant stimuli and observe the responses from A. Test sequences will be therefore convenient sequences of events at A and B, i.e. Transport service primitives. They are prepared by the Test Driver (TD); TD also directly controls test point A, instructs the Test Responder (TR) on how to control test point B, records the results at A and receives the results at B from the TR. Thus, also TD and TR carry out a protocol (indicated by a dashed line); this communication is supported by the service underlying the Transport layer, which is supposed to be reliable; the same assumption has to be made about the RI, so that errors and failures (if any) can be ascribed to the IUT. Otherwise, all the machine could do would be testing the Transport layer as a whole (which could also be per se an application of the machine). One might certainly wonder why RI has been assessed and indeed this is a frequent criticism to this test

architecture. A formal specifications compiler could generate correct RIs, provided its own correcteness is ensured. We refer to [6] for a comprehensive treatment of testing with a RI.

The next obvious step in developing the machine is the definition of the TD structure; this was done according to the principle of layer-independence; TD components are either layer-independent or easily adaptable to different layers. In that light, the choice of Transport layer appears only as a case study to which the testing machine is not tied, whereas other testing tools presented in the literature are dedicated to a specific protocol.

Section 2 deals with the problem of choosing a formal language suited to specify-Transport service subsets, from which the TD can extract test sequences. It is indeed our strong belief that testing techniques need to meet formal description techniques to gain in accuracy and reliability. Moreover, an automated tool needs a precise (if not formal) source specification to operate.

Since conformance to OSI standards is required of the external behaviour of a protocol entity, the authors' attention was drawn on LOTOS [10], the ISO-sponsored Language Of Temporal Ordering Specification, to which the concept of observation is central. LOTOS is taken as a starting point to define LOTOS-T, a language for use in our testing machine. The section is concluded by a tree-based semantics for LOTOS-T.

Section 3 describes the machine representation adopted for LOTOS-T specifications, the LOTOS-T compiler that produces such representations and the test sequence generator that extracts test sequences from them.

Finally in Section 4 the authors' conclusions are drawn and possible future extensions discussed.

2. LANGUAGE ELEMENTS

In tackling the problem of choosing a suitable specification language for a service, one is forced to consider what a service really is. According to ISO-OSI, the layers up to the Nth are viewed by the (N+1)th layer protocol entities as a 'Service Provider' (SP); correspondingly the (N+1)-entities are the 'users' of the SP. The SP and its users share a number of idealized interaction points, called Service Access Points (SAPs), where they can execute interactions termed 'abstract service primitives', characterized by a direction (from the SP to the users or viceversa) and by the data values exchanged. Since primitives are interactions, their execution requires the agreement of two parties. Both the SP and a user may offer to execute primitives; when their offers synchronize (i.e. match), a primitive can actually be executed. Thus, specifying the service offered by a SP amounts to specifying the temporal order of its offers to execute a primitive. It can be expected that the inherent nondeterminism of distributed enviroments will surface in a service specification, so that the primitives temporal order is likely to be partial: at each given instant more alternatives will in general be possible.

Amazingly, such a view of a service closely resembles the basic concepts of CCS, Milner's Calculus of Communicating Systems [11]: the SP and CCS 'agents' appear both as a 'black box' to their enviroment, with which they can both establish the very same kind of 'synchronized' communication; primitives and CCS 'actions', SAPs and CCS 'labels' seem to be different names for similar concepts. Besides, CCS provides a mathematical framework to represent temporal partial order of actions, including possible non-deterministic behaviours.

ISO ad-hoc group on Formal Description Techniques has combined CCS (plus some useful extensions) with a type definition language to obtain the

specification language LOTOS. Although LOTOS is an effective tool for specifying communication protocols and services, it was far too complex for our scope, as the first prototype of the testing machine was intended to be applied to the Transport Service (TS) only. Thus, a suitable subset of it, called LOTOS-T (LOTOS for Testing), has been selected for writing TS specifications. Basically, LOTOS-T enables one to combine its atomic components (primitives) by its operators, so as to obtain process expressions that specify the order in which primitives occur (more correctly: in which their occurrence is offered).

Below, the syntax and semantics of (the different forms of) LOTOS-T process expressions are presented informally. In the following P, Pi stand consistently for LOTOS-T process expression.

Primitives: <primitive identifier>(<gate>,x1,x2..xN)

Obviously this construct is intended to represent TS primitives. <primitive identifier> can be one of the following: conreq, conind, conres, concnf, disreq, disind, datreq, datind, expreq, expind. Since LOTOS-T is intended for use in specifying what the testing machine must do, suffixes 'req' 'res' ('ind' 'cnf') denote an input to (output from) the TS provider under test. <gate> can be either a or b, and is needed to distinguish the two SAPs at which the TS Provider is tested by the machine. x1,x2..xN (N>=0) are arbitrary strings but should represent the values conveyed by the primitive (see later for a discussion of this point).

Termination

Termination of a process expression is not denoted explicitly. However, when an expression can offer no more primitives for execution, it is said to have terminated. E.g., conreq(a) being a primitive is also an expression; when the offer it represents has been accepted and 'consumed', it has nothing more to offer and has therefore terminated.

Sequential composition operator: P1;P2

This construct means that when P1 has terminated, P2 is enabled.

Choice operator: P1+P2

P1+P2 behaves either like P1 or like P2. As a consequence of the introduction of this operator, a process expression can offer, at a given moment in time, several primitives for execution; if two or more of these offers match with those of the environment one of them is selected nondeterministically.

Behaviour: behaviour a(x1,..,xn):= P

Names (possibly) carrying formal parameters can be assigned to process expressions and referenced everywhere in the formation of process expressions, with a proper instantiation of the formal parameters. Both self- and mutual recursion are possible.

Session session <session identifier>

The specification of a testing session is made up of a (possibly empty) behaviour declarations list, and a main LOTOS-T process expression.

Example: session successful_opening_phase

behaviour opening(x,y):= conreq(x);conind(y);conres(y);concnf(x)

main opening(a,b) + opening(b,a)

2.1. Formal syntax

LOTOS-T syntax is presented below in extended BNF notation ($\{\ \}^*$ means 0 or more repetitions).

<session> ::= session <session identifier>
{<behaviour definition>}*
<main expression>

<behaviour definition> ::= behaviour <behaviour> := <expr>

<behaviour> ::= <behaviour ident> <formal parameter list>

<main expression> ::= main <expr>

<expr> ::= <term> | <term> + <expr>

<term> ::= <factor> | <factor>;<term>

<factor> ::= <primitive> | <behaviour reference> | (<expr>)

<primitive> ::= <primitive identifier> <primitive parameters list>

<primitive identifier> ::= conreq | conind | conres | concnf | disreq |
disind | datreq | datind | expreq | expind

<behaviour reference> ::= <behaviour ident> <actual parameters list>

<formal parameters list> ::= ({<parameter>}*)

<primitive parameters list> ::= ({<parameter>}*)

<actual parameters list> ::= ({<parameter>}*)

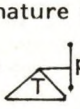
<parameter> <behaviour ident> <spec ident> ::= <character string>

2.2. A tree-based semantics for LOTOS-T

A way of giving a semantics to a formal language is to map it onto a convenient class of objects whose meaning is supposed to be known. The semantics of LOTOS, for instance, is defined in [10] by translating LOTOS into CCS+ (CCS+ is an extension to CCS). In the case of LOTOS-T a mapping in terms of trees has been preferred, both for the wide choice of machine representations available for trees and for their obvious, intuitive meaning.

First, a particular set of trees $\mathcal{T}$ is defined recursively (both a graphical and an equivalent (=) linear notation are employed):

1. b = b (where the nature of label b will be precised later) is a branch.

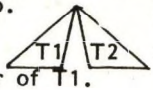
2. If T is a tree, then 

par = (T) is a branch. 'par' is a reserved label.

t3. A branch is a tree.

t4. If b is a branch and T is a tree, then

= b.T is a tree.
T is said to be a son of b.

t5 If T1, T2 are trees then 

= T1 + T2 is a tree.
T2 is said to be a brother of T1.

Below, the interpretation function F, having the set $\mathcal{A}$ of LOTOS-T process expressions as its domain and the set $\mathcal{C}$ as its range, is defined by induction over the structure of $\mathcal{A}$.

m1. $F(term + expr) \stackrel{def}{=} F(term) + F(expr)$

m2. $F(factor;term) \stackrel{def}{=} F(factor).F(term)$

m3. $F((expr)) \stackrel{def}{=} (F(expr))$

m4. The F-image of a primitive (reference) is a branch labelled with the primitive (reference), i.e. with its identifiers and its parameters.

Example 1

Let (conreq(a) + conreq(b));conind(a) be a LOTOS-T expression, then its interpretation is a tree obtained through F, by applying the m-rules.

F((conreq(a) + conreq(b));conind(a)) =
by m2
F((conreq(a) + conreq(b)).F(conind(a)) =
by m3
(F(conreq(a) + conreq(b)).F(conind(a)) =
by m1
(F(conreq(a)) + F(conreq(b)).F(conind(a))

The tree resulting has the following graphical representation:

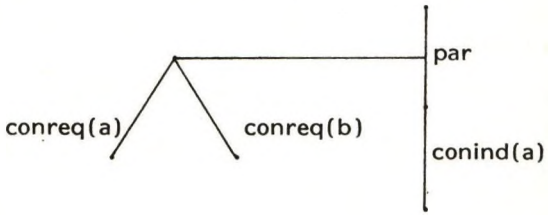


Figure 2.

Note that the mapping defined indeed interprets only the temporal order of primitives, as far as trees define ordering of events. Primitive and reference parameters are simply transferred to branch labels, and this gives no clue as to what they mean. Of course, they represent the data values exchanged by the TS provider, but this fact is transparent to the mapping. Accordingly, both the module that builds (representations of) trees from a LOTOS-T session and the one that selects test sequences from a tree view parameters as bare identifiers; neither do they interpret parameters nor do they check their correctness. This task will instead be up to the module that uses a test sequence to actually perform a test.

Another point deserving attention is a comparison between LOTOS-T and LOTOS. A noticeable restriction of LOTOS-T is its lack of type definition facilities, to be related to the above discussion about parameters and data values. A second main restriction is that, while LOTOS enables one to define arbitrary communication events (in terms of gates and values) and internal events, LOTOS-T simply provides a fixed number of predefined communication events, viz. primitives. Last but not least is the absence in LOTOS-T of disable, parallel composition and restriction operators. Disable operator could indeed come handy in practical tests, to deal with IUT erroneous behaviour. Parallel composition and restriction operator, combined together, model systems interconnection, so their importance could hardly be emphasized.

3. IMPLEMENTATION ISSUES

The modules (fig.3.) presented in this section are intended to be part of the TD.

The first one requires a LOTOS-T specification contained in an external text file as a source, and produces a PASCAL data structure (called tree in the figure, with slight abuse) as an object. The object is actually a concrete representation of the tree that, according to LOTOS-T semantics, conveys the meaning of the LOTOS-T source specification. In the sequel this module is named 'compiler' for obvious reasons.

The second module selects a path, i.e a test sequence, from the tree. In the sequel this module is called 'generator'.

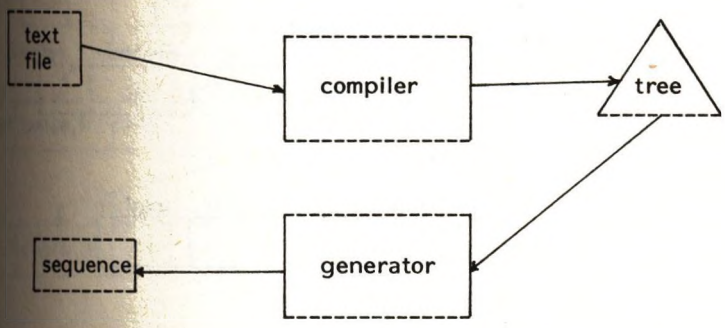


Figure 3. Program structure

3.1. The Compiler

In Section 2 a mapping from LOTOS-T specifications onto a particular class of trees $\mathcal{T}$ has been defined as the semantics of the specifications. Any concrete representation of such trees could be chosen as object code. PASCAL data structures (mainly pointers and records) provide a natural and simple tool for this and have been therefore preferred. Hence, the entire test sequence generator has been written in PASCAL. Although the object code has been especially devised for the needs of a testing machine, it may be useful for a generic application, as a bridge between a LOTOS specification and the output desired.

Below the data structure that represents tree branches is introduced.

```
information = record ident:string;
                     param:parameter
              end;
```

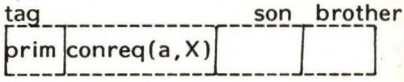
```
pointer = ^ branch;

branch = record son, brother:pointer;
              case tag:kind of
                   prim:(primitive:information);
                   ref:(reference:information;
                        refjunc : behavpointer);
                   par:(junction:pointer)
         end;
```

Let F stand again for the interpretation function. When the compiler processes a LOTOS-T factor, a data value of type branch is created with fields son, brother pointing to the son and the brother of F(factor) (remember that by rules m3, m4 F(factor) is a branch). The field tag is set to prim, ref or par depending on the kind of the factor. If the factor is a primitive, the field primitive will contain the identifier and the parameters of the primitive. The same applies to behaviour references, except for the field refjunc that points to the (representation of the) behaviour referenced. If the factor is an expression enclosed in brackets, say '(expr)', field junction points to the (representation of) the tree F(expr).

Figure 4 illustrates the internal structure of a branch in the three possible cases.

conreq(a,X) is mapped onto



behav(x,y) is mapped onto

(expr) is mapped onto

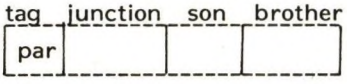


Figure 4. Branches internal structure.

Finally behaviour representation is considered. When a behaviour declaration like p(x) := expr is encountered, a record with three fields is created: the first contains the identifier and the parameters of the behaviour (namely p(x)); the second points to (the representation of) the tree F(expr); the third points to a data value of the same type, so that behaviour representations are connected in a list.

Example 1

```
behaviour con(x,y) := conreq(x);conind(y)
behaviour dis(x,y) := disind(x);disind(y)
```

The representation of these behaviours is depicted in fig.5.

Example 2

Assume tnat con(x,y) and dis(x,y) are defined like in Example 1. The representation of:

main (con(a,b) + con(b,a));(dis(a,b) + dis(b,a))

is shown in fig. 6.

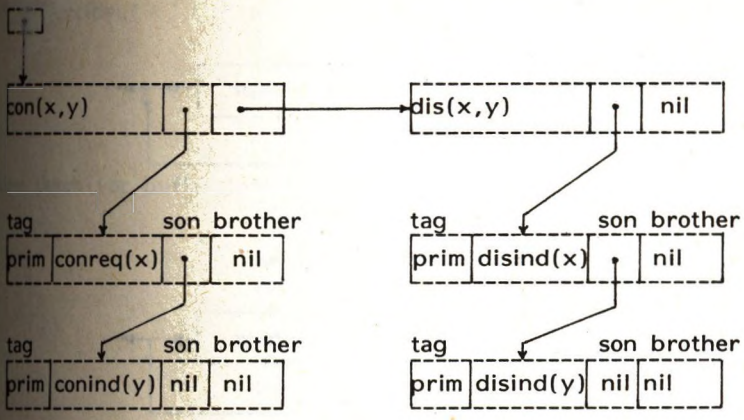


Figure 5.

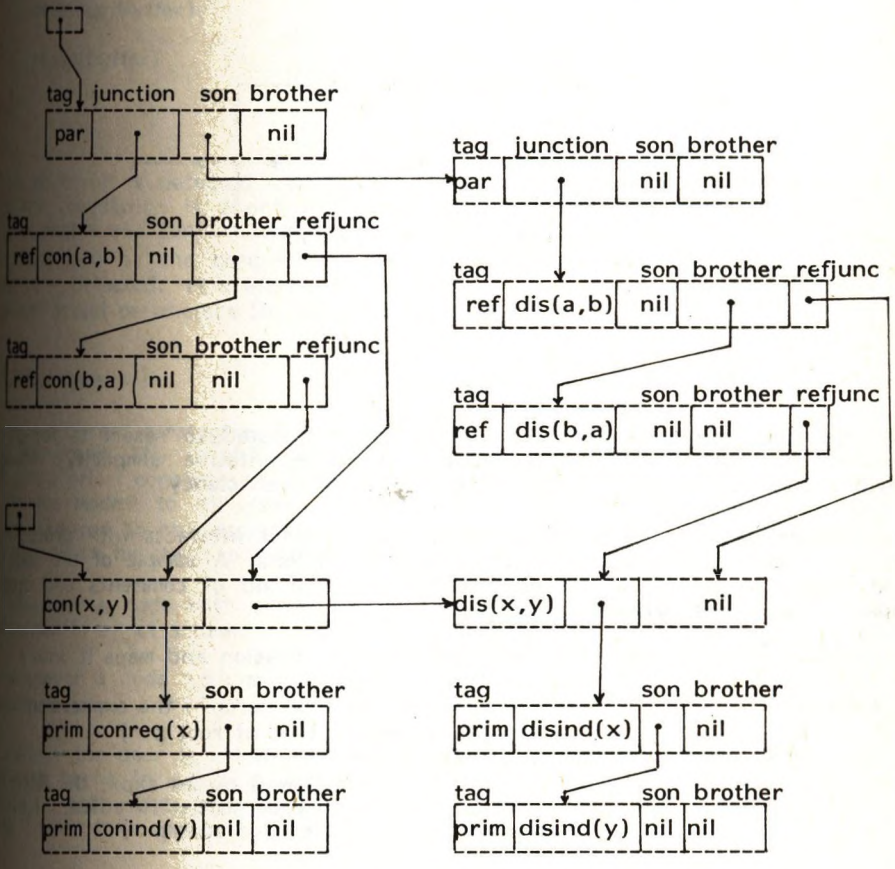
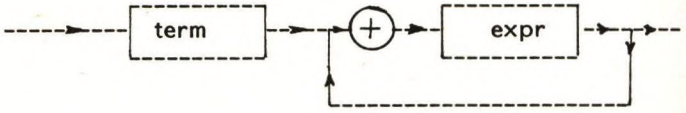


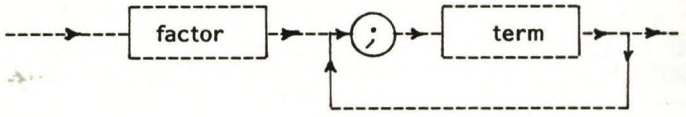
Figure 6.

The technique used by the compiler to analyze LOTOS-T specification and to create the object code is quite well-known. LOTOS-T expressions are made up of terms and factors. This fact can be depicted with the syntactic chart in fig. 7.

LOTOS-expression



term



factor

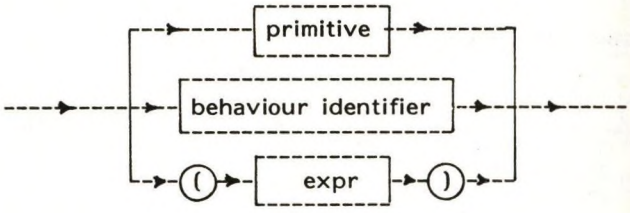


Figure 7.

The recursive structure of these charts has suggested to resort to top-down recursive parsing for its conceptual and implementative simplicity, though probably a price has to be paid for this in terms of efficiency.

The compiler is basically a procedure called expr; it interacts with procedures term and factor to analyze and recognize expressions. A sample of the code is listed below; it can be easily understood with the aid of comments and cross-comparison with the syntactic chart.

```
procedure expr(p:pointer);      (* recognizes an expression and maps it onto a
                                   tree of root p *)
begin term(p);                  (* recognizes the first term in the expression
                                   and maps it onto a tree of root p *)
      if buffer.ident = '+' then expr(p^.brother)
                                (* if the term is followed by an expr, the latter
                                   is mapped onto a tree of root p^.brother, i.e. a
                                   tree brother of the one of root p *)
end;
```

```
procedure term(p:pointer);      (* recognizes a term and maps it onto a tree of
                                   root p *)
begin factor(p);                (* recognizes the first factor in the term and
                                   maps it onto a tree of root p *)
    if buffer.ident = ';' then term(p^.son)
                                (* if the first factor is followed by a term, the
                                   latter is mapped onto a tree of root p^.son, i.e. a
                                   tree son of the one of root p *)
end;


procedure readt(var buffer:information)
                                (* This is the lexical analyzer that performs the
                                   task of scanning the file to recognize tokens, i.e.
                                   keywords, terminal symbols, identifiers. There-
                                   fore, through this procedure, the text file
                                   containing the specification can be viewed as a file
                                   of tokens; procedure readt assigns the current
                                   token to variable buffer and advances to the next
                                   one. *)


procedure factor(p:pointer);    (* maps a factor onto a branch of root p*)
begin readt(buffer);
    create(p,buffer);           (* creates the appropriate instance of type
                                   branch *)
    readt(buffer)               (* moves buffer to the terminal symbol (viz.
                                   '+' or ';') following the factor *)
end;
```

When an error is detected by the compiler an appropriate diagnostic is issued and the compilation is aborted. Syntax errors can be divided in context free syntax errors and context dependent syntax errors. Errors of the first class are all recognized by the compiler, but only the following members of the second class are detected: references to undeclared behaviours and length mismatch between actual parameters list and the formal one.

3.2. The Generator

Several different strategies have been proposed to solve the problem of choosing test sequences [12]. In the authors' view, the test sequences generator should not put a priori constraints on the test sequences but should leave a wide range of 'choice modes' to its user. Basically, the generator implemented visits the trees produced by the compiler, making available three choice modes selected by the user when he starts a testing session:

1) whenever a node with more than one son is reached, the user is prompted and asked to select a branch.

2) whenever a node with more than one son is reached, the generator makes a random choice.

3) initially the user is asked for the primitives he wishes and for those he does not wish to be included in the path. Since this may not suffice to uniquely identify a path, the generator must be directed to act like in either mode 1 or mode 2 when it is confronted with multiple alternatives.

4. CONCLUSIONS

The main extension toward a full LOTOS compiler would be the introduction of disable, parallel composition and restriction operators. This can be easily arranged by extending CCS expansion theorem [111] to LOTOS. Such extended theorem would state that an arbitrarily complex LOTOS expression can be put in a canonical form in which only operators + and ; appear, therefore ensuring that the machine representation of any LOTOS source specification can ultimately show the same structure devised for the simpler LOTOS-T. The advantage in this is twofold: 1) the already existing test sequence generator need not be changed, 2) even if it is changed, choosing a test sequence would still amount in principle to selecting a path in a binary tree. Two possible alternative implementation strategies can be envisaged. 1) the expansion theorem could be 'included' in the compiler, in order to directly obtain as output ordinary trees of the class $\mathcal{T}$ defined in Section 2. 2) the compiler could be (slightly) refined, so as to yield trees of a more complex form, capable of representing process expressions containing the new operators. A module implementing the expansion theorem could then process these complex trees to obtain again regular trees of $\mathcal{T}$. In other words, in this second case the compilation would be executed in two separate passes.

Semantically LOTOS-T conveys little more meaning than ordinary trees do. Yet LOTOS conveys much more and since extending LOTOS-T compiler to LOTOS seems feasible, its construction has an obvious justification.

Even though LOTOS is generally judged an effective formal tool for several applications in the field of communication protocols, it has been employed up to now only for specification and verification (see for example [13] [14]). This paper gives an account of the first attempt to introduce its use in the new and challenging realm of protocol testing. The adoption of LOTOS has led to encouraging results, since two modules of a testing machine could be designed and developed quite in a straightforward way. However, much has still to be done, both concerning testing methods and, on the other hand, concerning the extension of the kernel language understood by the machine.

5. REFERENCES

[1] International Standard Organization - Information Processing Systems - Open Systems Interconnection - Basic Reference Model - ISO/TC97/IS 7498

[2] International Standard Organization - Information Processing Systems - Open Systems Interconnection - Working draft answer to question Q 16 B on conformance testing - ISO/TC97/SC 21/N. 90 - June 1984

[3] D. Rayner, A system for testing protocol implementations, Computer Networks, Vol.6, No.6, pp.383-395, December 1982

[4] J.P. Ansart, STQ, CERBERE and GENEPI: three tools for implementation testing, EUTECO-European Teleinformatics Conference, Varese, 3-6 October 1983, North-Holland 1983

[5] U. Faltin et al., TESDI manual: testing and diagnosis aid for higher level protocols, IFV-IK-RZ,GMD, Darmstadt 1983

[6] J.S.Nightingale, Protocol testing using a reference implementation, Proc. 2nd International Workshop on Protocol Specification, Testing and Verification, Idyllwild (California), May 17-20 1982, North-Holland 1982

[7] R.Hale, D.Rayner, Protocol product testing: Some comparisons, EUTECO - European Teleinformatics Conference, Varese, 3-6 October 1983, North-Holland 1983

[8] S.Palazzo, P.Fogliata, G.Le Moli, A layer independent architecture for a testing system of protocol implementations, Proc. 3rd International Workshop on Protocol Specification, Testing and Verification, Zurich, 31 May - 2 June 1983, North-Holland, 1983

[9] International Standard Organization - Information Processing Systems - Open Systems Interconnection - Transport Service - ISO/TC97/DIS 8072

[10] International Standard Organization - Information Processing Systems - Open Systems Interconnection - Draft tutorial document on Temporal Ordering Specification Language - ISO/TC97/SC 16/N. 1731 - 1984

[11] R.Milner, A Calculus of Communicating Systems, LNCS 92 Springer- Verlag, Berlin 1980

[12] B.Sarikaya, G.v.Bochmann, Some experiences with test sequence generation for protocols, Proc. 2nd International Workshop on Protocol Specification, Testing and Verification, Idyllwild (California), May 17-20 1982, North-Holland 1982

[13] V.Carchiolo, G.Le Moli, S.Palazzo, G.Pappalardo, Modelling and Specifying a Presentation Protocol by Temporal Ordering, Proc. 4th International Workshop on Protocol Specification, Testing and Verification, Skytop Lodge, (Pennsylvania) June 11-14 1984, North-Holland 1984

[14] V.Carchiolo, A.Faro, G. Scollo, A Temporal Ordering Specification of some Session Services, ACM SIGCOMM 1984 Symposium on Communication Architectures and Protocols, Montreal 6-8 June 1984

Description of Protocol Testing Scenarios

O. Rafiq, J. Haddad

Groupe Bull
Direction Technique
Bull Transac BP 92
91301 Massy, France

STP is a distributed System for Testing OSI Protocol entities, defined and developed by Bull. It is in keeping with the classical architecture of distributed testing : a responder system over the entity under test and a testing center accessing to the entity under test via an adequate network. Moreover, STP is characterized by a good methodology for constructing and describing the testing scenarios. This paper presents the principles of this methodology through a significant example : class 2 of the OSI transport protocol.

1. Introduction

The OSI model /ISO DIS 7498/ and the associated protocols, defined by the ISO (International Organization for Standardization), are already in their realization stage at several companies and manufacturers (see the agreement concluded on the 15th of March 1984 between the twelve European manufacturers, immediately followed by another agreement between fifteen American companies on the National Bureau of Standards initiative).

At the same time, on one hand, several testing systems are designed throughout the world : Canada/Bochmann 83/, France/Ansart 83/, UK/Rayner 82/, USA/Linn 83/, West Germany/Giessler 83/, etc., to verify the correct behavior of the OSI layers implementations in the communicating systems as well as their conformance to the specifications and standards. On the other hand, studies are in progress in different working groups : ISO, NBS(USA), DSTC(Bull), etc., to define a methodology to construct and describe the scenarios which will be executed in these verifications.

In this paper, we present the principles we are using in STP(System for Testing Protocols) of Bull/Rafiq 85a/, for constructing and describing the testing scenarios of an OSI protocol entity. In particular, we adopt two description techniques : the testing graph and the SDL language/SDL 81/ defined by the CCITT(Comité Consultatif International Télégraphique et Téléphonique). The testing graph is a concept which is simple, analytical and adapted to the scenarios construction. The use of SDL is inspired by the classical techniques of protocol specification based on the finite state models. It is particularly useful for programming the testing scenarios in STP.

In section 2, we will present the STP architecture and the main principles of its behavior. Section 3 gives the definitions and the principles used for the testing scenarios construction and description. Section 4 shows some examples of scenarios from the ISO transport protocol/ISO DIS 8073/. Section 5 deals with the problems of the testing refinement and chaining. Finally, section 6 comments the use and the development of these techniques.

2. Testing Approach

2.1 The OSI Protocol Entity

In the OSI model/ISO DIS 7498/, the relationships between the open systems are defined in terms of seven hierarchical levels of protocol and service : 1-Physical, 2-Data-Link, 3-Network, 4-Transport, 5-Session, 6-Presentation and 7-Application. In each open system, the functions associated to an (N)level are achieved by one or several (N)protocol entities. Two (N)entities belonging to two different systems are called peer entities. They exchange (N)messages called (N)PDU(Protocol Data Unit), by obeying a set of rules called (N)protocol.

In order to exchange information between two peer entities, each (N)entity delivers the data and control information to the lower level ((N-1)entity) by using the (N-1)service primitives ((N-1)SDU : Service Data Unit) and so forth until the physical medium is reached. After that, in the other system, information goes up towards the related (N)entity. This link between two peer entities is called a connection and designated by (X)connection in each (X)level. An entity can manage several connections in parallel. Each one of these connections is identified in a unique manner.

In a system, an (N)entity is linked to each one of the adjacent entities , ((N-1)entity, (N+1)entity), only by a service interface ((N-1)service, (N)service) which defines the primitives for use.

The testing of an (N)protocol entity consists of executing one or several sets of testing sequences (scenarios) in an adequate environment (e.g. STP), to verify whether the (N)entity correctly uses the (N-1)service for implementing the functions of (N)protocol, in order to provide a specific (N)service. Given that the considered protocols are connection oriented, the usual functions implemented are : connection establishment, connection release and data transfer : segmenting, reassembling, concatenation, separation, flow control, expedited data, sequencing, errors detection, etc.

2.2 The General Architecture of STP

STP is in keeping with the classical distributed architecture for testing the OSI protocol entities/Cowin 83/. This architecture is based on a testing system (or simply TS) which is resident in a testing center (TC) and able to have access to the (N)entity under test ((N)EUT) via an appropriate communication service ((N-1)service). The (N)EUT is implemented over the (N-1)service in its real environment (system under test : SUT). A system usually called a responder system (RS), is implemented over the access interface of the (N)service provided by the (N)EUT, in order to simulate a user. The design of such an RS, poses number of problems /Cowin 83, Rafiq 84, Rafiq 85b/ so great is the influence of its functions on the execution and the value of the testing. The testing itself, in this arrangement, consists mainly of "observing" the behavior of the (N)EUT seen as a "black box", in the course of messages exchanges (scenarios) between the TS and the RS.

In STP, in order to design a responder, we have taken account of an important characteristic of the OSI implementations made by Bull (ISO/DSA WorkStation, Questar 400, SPS7, ...) and others, that we call a staircase implementation. This arrangement allows a user to access to the communication levels of his choice and not necessarily to the highest implemented level. Thus, as an example, a user may indifferently access to the network level, transport level, session level, etc. Furthermore, this carrying out of the OSI model, permits the design of a responder system which is particularly handy and efficient. We call it the astride responder (AR).

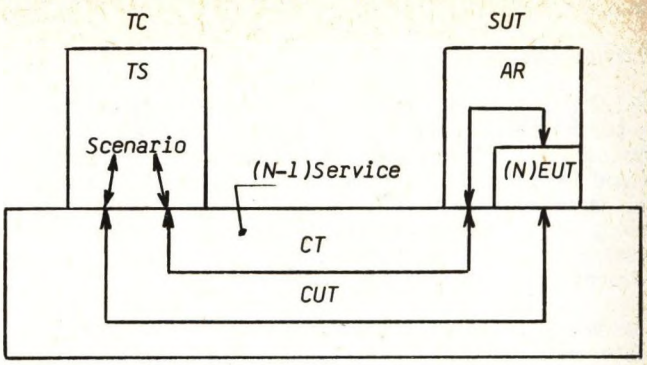


Figure 1. General Architecture of STP

The AR is a software component in the SUT, implemented over two communication levels : the (N)level under test ((N)EUT) and a lower level already tested ((N-1)level in this case). The fundamental interest of such a disposition is to be able to accomplish exchanges between the TS and the AR via a reliable connection without passing through the (N)EUT. This (N-1)connection is called the connection for testing (CT). In this way, we arrive at the general architecture of STP, presented in figure 1.

The AR is able to establish the CT initiated by the TS. The (N)service provided by the (N)EUT, can then be used by the TS through the (N-1)data exchanged with the AR. These (N-1)data convey the (N)service primitives allowing this kind of use. At the receiving of (N-1)data, the AR extracts the (N)request to deliver to the (N)EUT. On receiving an (N)indication, the AR packs it into (N-1)data to be sent to the TS. To begin testing, the (N)EUT can be reached from TS, either by a connection under test (CUT in figure 1) or by the CT through the AR. If the (N)EUT is not reachable by either means, it is not testable.

The features of testing in STP are presented in /Rafiq 85a, Rafiq 85b/. Herein, we will particularly focus on the construction and description of the protocol testing scenarios. Since the AR has no "intelligence", the ordering of the testing messages is defined in the TS. Consequently, it is possible to follow a clear and methodological approach to design scenarios and supervise the testing.

3. Scenarios Construction and Description

This section presents the elements of the methodology we are using for constructing and describing the testing scenarios in the TS. Our approach is mainly based on the ordering of the sendings of messages and the associated responses on CUT and CT in the TS.

3.1 Definitions and Principles

A messages exchange in the TS, is defined as a sending of message on CUT or CT and the associated response. The response can be either a receiving of message on CUT or CT, or a receiving of message on each of CUT and CT.

In principle, after each sending of message on CUT or CT, any other sending of message must not be executed without receiving either a response on CUT and/or CT, or on time-out. However, a sending (respectively receiving) of message may require successive sendings (respectively receivings) either when the segmenting mechanism is used or when the data transfer is based on the credit mechanism.

A scenario is defined as a finite chaining of messages exchanges which executes a precise and specific function of the (N)EUT : e.g. connection establishment, data transfer with flow control, etc.

Two types of scenarios are considered : basic scenario and generalized scenario. The basic scenario corresponds to the normal achievement of a function. However, during any protocol phase, number of exceptions and errors may occur. In general, these cases and the associated treatments are specified in the given protocol. As examples, we quote the following cases : error in message parameter, refusal, unknown message, time-out, etc. Therefrom, a generalized scenario is constructed from the basic scenario by associating to each concerned step (sending or receiving) the error and exception cases that one wants to study.

3.2 The testing graph

In order to construct the testing scenarios, we use first of all the notion of testing graph. It only presents the sendings and receivings on CUT and CT as well as the time-out. It particularly gives a synthetic and analitycal view of the testing carrying out in the TS.

The testing graph is an oriented graph /Berge 70/ : a finite set of nodes linked by arcs (oriented edge). Each node specifies either a sending or a receiving of message on CUT or CT, or time-out. Each arc stems from a node and ends at another node which describes a possible progress of the testing. The graph is developed from a given initial node.

Schematically, a node is represented either by the extremities of CUT and CT (in TS) accompanied by an arrow corresponding to the sending or receiving of message, or by the symbol ⏲ . The node ⏲ specifies that any message (response) has not been received by the TS within a time delay which was fixed in the predecessor node. Each node is labeled by a number. The messages in brackets mean that they are packed into (N-1)data transfer primitives. Figure 2 shows a general example of testing graph.

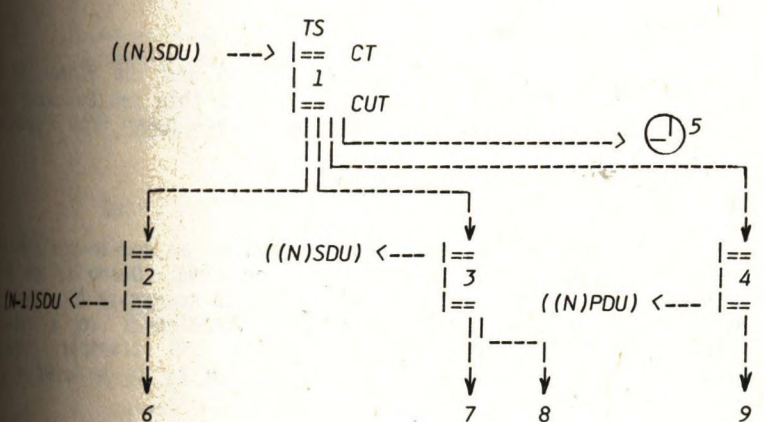


Figure 2. Testing Graph

3.3 The Use of SDL

The testing graph specifies only sendings, receivings and time-out. In order to

make clear and elegant descriptions of the testing scenarios in a high level programming language, one must be able to express explicitly the notion of waiting state. Indeed during the testing, the TS can be within one of both following situations :
- waiting for a message (or time-out)
- execution of the treatment associated with the receiving of a message (or time-out).

The finite state models /Danthine 80/ are particularly adapted to describe this behavior. They are defined in terms of states (waiting for messages), transitions (execution of treatment), inputs (incoming messages) and outputs (outgoing messages). A finite state model is a finite set of states linked by transitions. An input causes the model to leave a state and travel along a transition, executing tasks, generating outputs and branching on decisions until another state is reached.

Several high level languages based on these models are proposed /Rudin 83/. Two are defined by both international standardization bodies : ISO /FDT 84/ and CCITT /SDL 81/. The ISO language is in progress and will describe the OSI protocols. The SDL (Functional Specification and Description Language) is a formalized presentation method, recommended by the CCITT (Z101 to Z104). It is quite general, easy to learn, known and used throughout the world. Furthermore, several examples of compiler exist already /SDL 85/. We have choosen this language as the basis of LTP (Language for Testing Protocols) we are designing for describing and programming the testing scenarios in STP.

SDL can be presented in both forms : graphical (SDL/GR) and linear (SDL/PR). the graphical form is mainly adapted to description and the linear form is programming oriented, but these forms are closely connected. Through lack of space, only the linear form is used in the examples of scenario presented in the following section. /Haddad 84/ presents several graphical form examples.

The linear SDL structures used are : state(STATE, ENDSTATE), internal input(INPUT INTERNAL), external input(INPUT), task(TASK), internal output(OUTPUT INTERNAL), external output(OUTPUT) and next state(NEXTSTATE).

4. Examples from the Transport Layer

In order to illustrate the use of the testing graph and the linear SDL to construct and to describe the testing scenarios in STP, this section shows some examples from the ISO' transport layer. These examples are taken from a detailed document /Haddad 84/.

4.1 The Transport Layer

Briefly, the transport layer in the OSI model, provides an end-to-end transfer of data between users (session entities) through networks. Owing to the fact that the transport service is connection oriented, it is necessary to establish, use and terminate a transport connection. It corresponds to a two-way simultaneous path between a pair of users. The ISO transport service /ISO DIS 8072/ defines the service elements (see Table 1) to be used by the session entities for these three phases.

In order to provide this service between two users, the transport protocol has to establish, maintain and terminate the transport connection between both associated transport entities by exchanging the appropriate TPDUs. As to that, the transport protocol uses the network service (see Table 2). Furthermore, a primary objective of the transport protocol is to provide a quality of service better than delivered by the used network. Thus, the ISO

Phase	Service Elements (TSDU)		Protocol Elements (TPDU)	
	Name	Primitive	Name	Primitive
Establishment	T_CONNECT			
	Request	TCONreq	Connect Request	CR
	Indication	TCONind		
	Response	TCONrsp	Connect Confirm	CC
	Confirm	TCONcnf		
	T_DATA		Data	DT
	Request	TDTreq		
Data	Indication	TDTind	Data Acknowledge	AK
Transfer	T_EXPEDITED DATA		Expedited Data	ED
	Request	TEDreq		
	Indication	TEDind	Expedited Acknowledge	EA
	T_DISCONNECT		Disconnect Request	DR
Release	Request	TDISreq		
	Indication	TDISind	Disconnect Confirm	DC
All phases			Error	ER

Table 1 - Service and Protocol Elements of the ISO Transport (Class 2)

Phase	Service Elements (NSDU)	
	Name	Primitive
Establishment	N_CONNECT	
	request, indication	NCONreq, NCONind,
	response, confirm	NCONrsp, NCONcnf
	N_DATA	
	request, indication	NDTreq, NDTind
Data	N_DATA ACKNOWLEDGE	
	request, indication	NACKreq, NACKind
	N_EXPEDITED DATA	
	request, indication	NEDreq, NEDind
Transfer	N_RESET	
	request, indication	NRESreq, NRESind,
	response, confirm	NRESrsp, NREScnf
Release	N_DISCONNECT	
	request, indication	NDISreq, NDISind

Table 2 - Network Service Elements

defines five transport protocol classes /ISO DIS 8073/ : Class 0-Simple Class, Class 1-Basic Class Error Recovery Class, Class 2-Multiplexing Class, Class 3-Error Recovery and Multiplexing Class, Class 4-Error Detection and Recovery Class, associated to different qualities of network service.

The examples we develop below, are from the class 2 of /ISO DIS 8073/. Table 1 presents the used protocols elements (TPDUs).

4.2 Connection Establishment

The transport connection establishment between two users, is achieved by the exchange of the transport PDUs (or TPDUs) : CR and CC, between both associated transport entities. This exchange is started on one of the users initiative and then they use the transport service primitives : TCONreq, TCONind, TCONrsp and TCONcnf. However, the TPDUs : CR and CC, must be conveyed via the network service. Consequently, if the multiplexing is not possible, the initiator transport entity has to establish a network connection with the other transport entity by using the network service primitives : NCONreq, NCONind, NCONrsp and NCONcnf. Thus, the information exchanged by the transport (TPDUs) is conveyed by the data transfer primitives of network (NDTreq, NDTind).

If a transport connection can not be accepted by an entity (e.g. invalid CR TPDU, etc.), this one sends DR or ER TPDU and shall regard the connection as released. The DR TPDU is confirmed by a DC TPDU only when it is sent after a CC TPDU.

Figure 3 shows a testing graph associated to a generalized scenario of a transport connection establishment initiated by the SUT and preceded by a network connection establishment. The node 1 is the initial node. The basic scenario is specified by the nodes 1, 2, 3, 4, 5 and 6. The nodes 23, 24 and 26 specify the time-out associated to the basic scenario. The nodes 7, 8, 13, 15, 17 and 19 show the testing developments taking into account errors and exceptions :

- node 7 : refusal of the EUT to initiate the establishment of the transport connection,
- node 8 : refusal of the TS to establish the network connection,
- node 13 : refusal of the TS to establish the transport connection,
- node 15 : error of CR TPDU parameter,
- node 17 and node 19 : refusal of the EUT to establish the transport connection (error of CC TPDU parameter, etc.).

4.3 Connection release

The release function is executed by a transport entity in order to terminate a transport connection either during the establishment phase (case 1) or when the connection is established (case 2). Case 1 is considered in the previous example. In this paragraph, we show a testing graph of case 2 (figure 4) and an associated linear description based on SDL.

The release function is achieved by the exchange of DR and DC TPDU between both transport entities. The transport service primitives used are TDISreq and TDISind.

The following linear description is made up of two parts : DECLARATION_PART and PROCESSING_PART. DECLARATION_PART declares the used states, (N)protocol elements, (N)service primitives and timers. PROCESSING_PART is developed state by state using the linear SDL structures. The statement TASK Print() prints comments. The message Start_Disc initiates the testing. The timers are

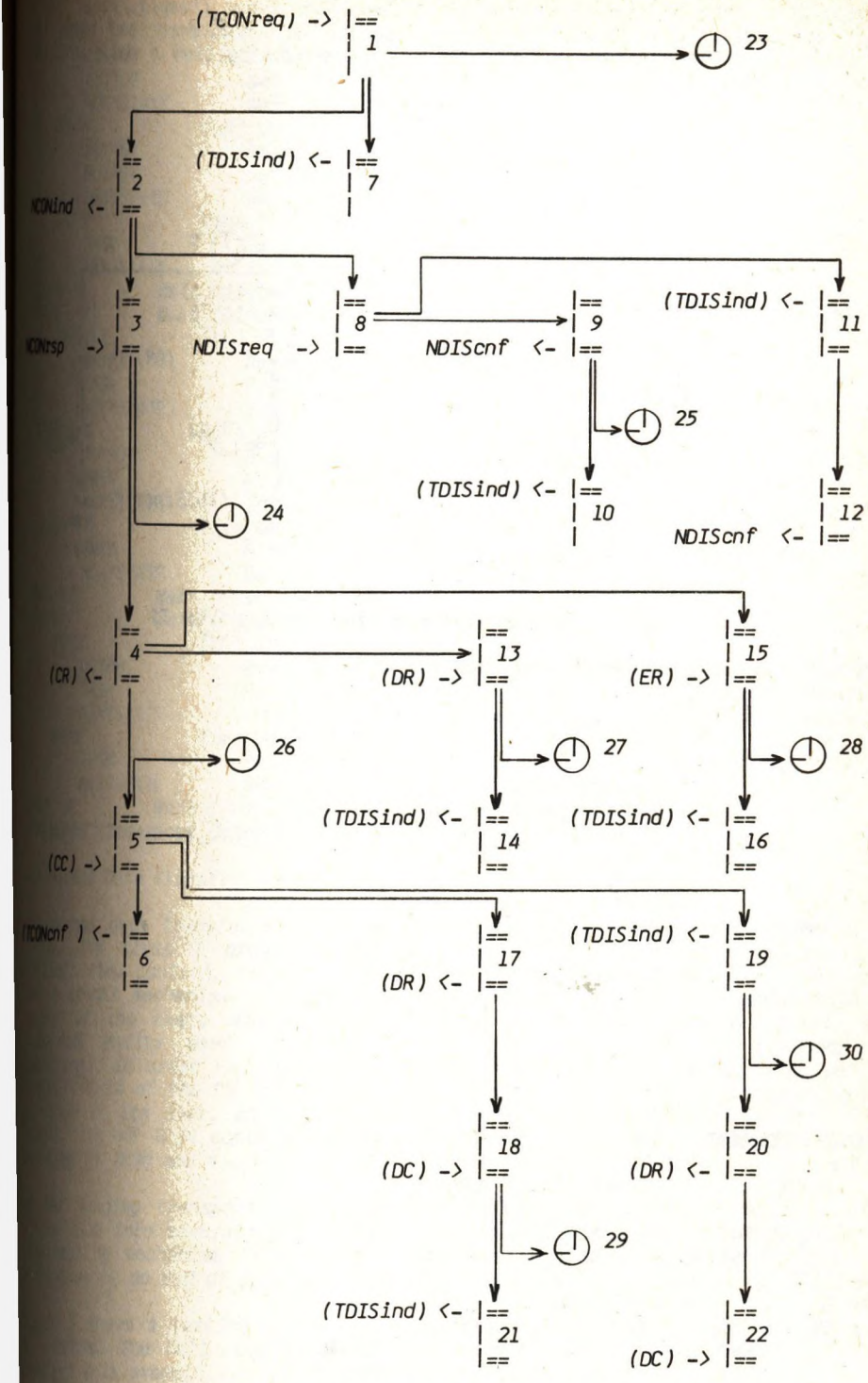


Figure 3. Transport Connection Establishment from SUT

```
PROCESS Normal_Data_Transfer_from_SUT;
PROCESSING_PART;
STATE           Open;
    INPUT           Start_Transfer INTERNAL;
        OUTPUT          (TDtReq)_On_CT;
        OUTPUT          Set_Timer(T1) INTERNAL;
        NEXTSTATE       Wait_(DT0)_On_CUT;
ENDSTATE        Open;
STATE           Wait_(DT0)_On_CUT;
    INPUT           (DT0)_On_CUT;
        OUTPUT          Reset_Timer(T1) INTERNAL;
        OUTPUT          Set_Timer(T2) INTERNAL;
        NEXTSTATE       Wait_(DT1)_On_CUT;
    INPUT           Time_Out(T1) INTERNAL;
        TASK            Print(Exception1);
        NEXTSTATE       Refine1;
ENDSTATE        Wait_(DT0)_On_CUT;
STATE           Wait_(DT1)_On_CUT;
    INPUT           (DT1)_On_CUT;
        OUTPUT          Reset_Timer(T2) INTERNAL;
        OUTPUT          Set_Timer(T3) INTERNAL;
        NEXTSTATE       Wait_(DT2)_On_CUT;
    INPUT           Time_Out(T2) INTERNAL;
        TASK            Print(Exception2);
        NEXTSTATE       Refine2;
ENDSTATE        Wait_(DT1)_On_CUT;
STATE           Wait_(DT2)_On_CUT;
    INPUT           (DT2,EOT)_On_CUT;
        OUTPUT          Reset_Timer(T3) INTERNAL;
        TASK            Print(Comment1);
        OUTPUT          (TDtReq)_On_CT;
        OUTPUT          Set_Timer(T4) INTERNAL;
        NEXTSTATE       Wait_(DT3)_On_CUT;
    INPUT           Time_Out(T3) INTERNAL;
        TASK            Print(Exception3);
        NEXTSTATE       Refine3;
ENDSTATE        Wait_(DT2)_On_CUT;
STATE           Wait_(DT3)_On_CUT;
    INPUT           (DT3)_On_CUT;
        OUTPUT          Reset_Timer(T4) INTERNAL;
        OUTPUT          Set_Timer(T5) INTERNAL;
        NEXTSTATE       Wait_(DT4)_On_CUT;
    INPUT           Time_Out(T4) INTERNAL;
        TASK            Print(Exception4);
        NEXTSTATE       Refine4;
ENDSTATE        Wait_(DT3)_On_CUT;
STATE           Wait_(DT4)_On_CUT;
    INPUT           (DT4)_On_CUT;
        OUTPUT          Reset_Timer(T5) INTERNAL;
        OUTPUT          AK(5,1)_On_CUT;
        OUTPUT          Set_Timer(T6) INTERNAL;
        NEXTSTATE       Wait_(DT5)_On_CUT;
    INPUT           Time_Out(T5) INTERNAL;
        TASK            Print(Exception5);
        NEXTSTATE       Refine5;
ENDSTATE        Wait_(DT4)_On_CUT;
STATE           Wait_(DT5)_On_CUT;
    INPUT           (DT5,EOT)_On_CUT;
        OUTPUT          Reset_Timer(T6) INTERNAL;
```

```
    TASK            Print(Comment2);
    NEXTSTATE       Current_Transfer;
  INPUT          Time_Out(T6) INTERNAL;
    TASK            Print(Exception6);
    NEXTSTATE       Refine6;
ENDSTATE       Wait_(DT5)_On_CUT;
ENDPROCESS Normal_Data_Transfer_from_SUT;
```

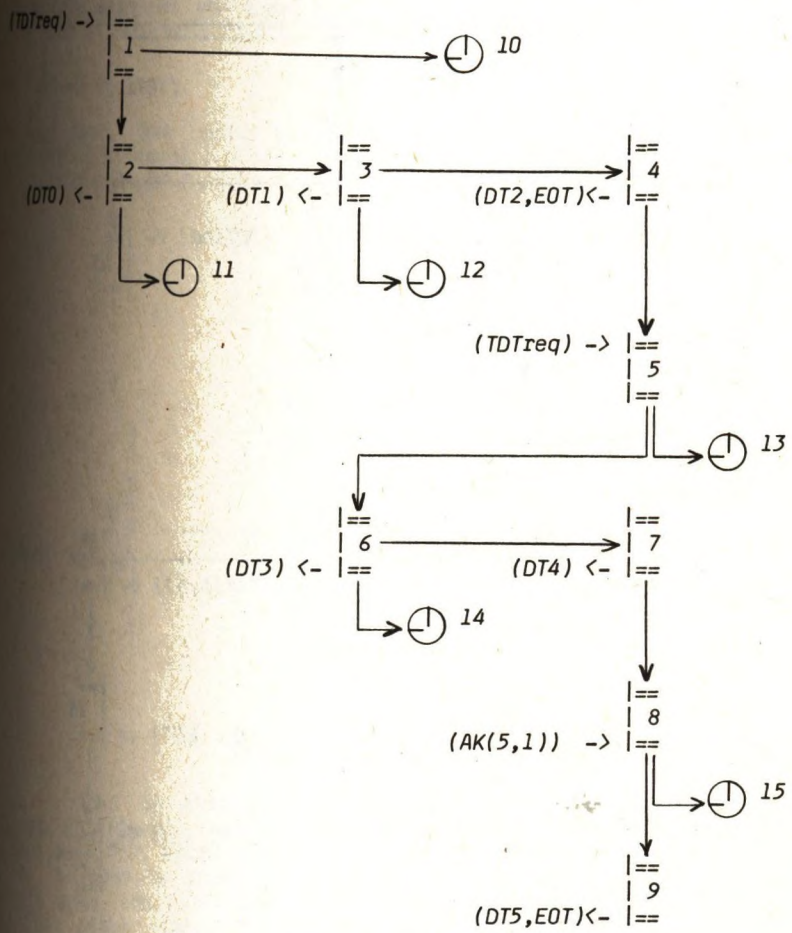


Figure 5. Normal Data Transfer from SUT

4.5 Expedited Data Transfer

The expedited service provides a data transfer under a separate flow control. The expedited flow ensure that TEDreq is delivered before any TDTreq sent after. Each TEDreq is conveyed as the data field of an ED TPDU. An ED TPDU can only be sent, if the previous one has been acknowledged by an EA TPDU.

Figure 5 shows a testing graph of normal and expedited data transfer. The initial credit is equal to 5 TPDUs. The time-out are omitted to simplify the graph. The nodes 13 and 14 correspond to both cases where TDTind is delivered before TEDind.

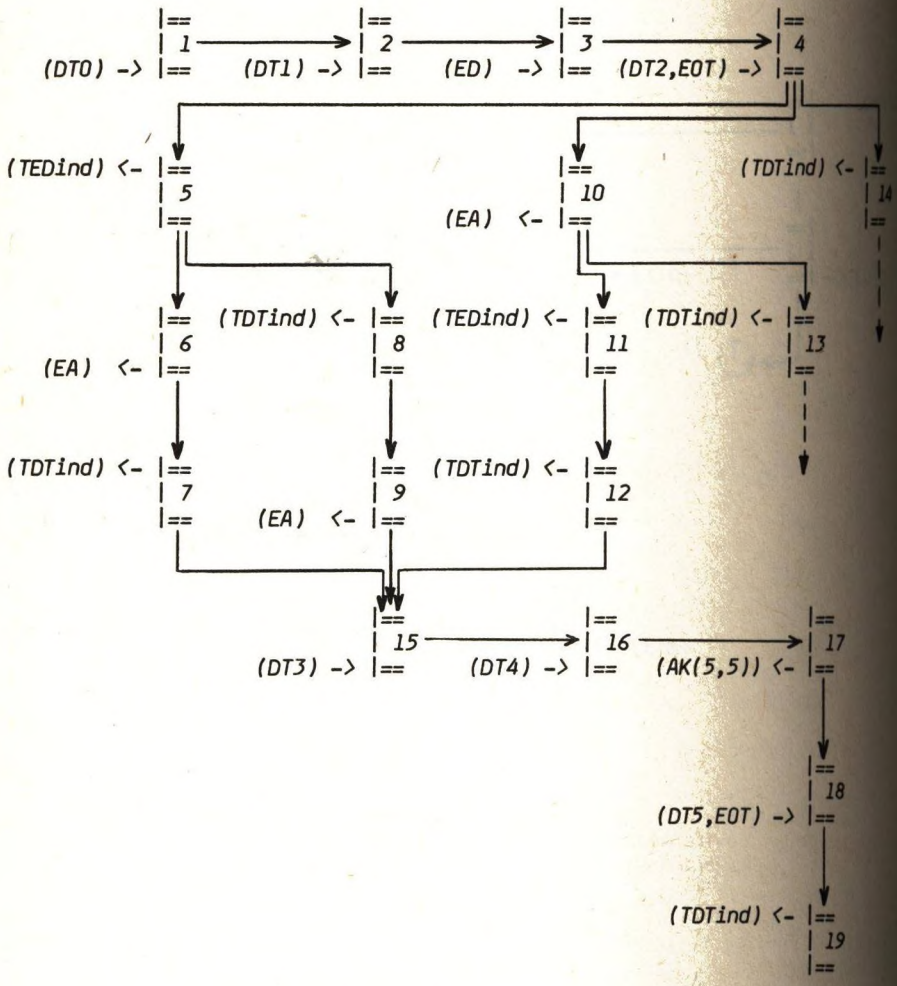


Figure 6. Expedited Data Transfer from TS

5. Refinement and Chaining

5.1 Refinement

In order to wait for messages from the EUT, the TS uses delays through timers. Besides the integration of the treatments of error cases in a scenario, it is necessary to adopt solutions to face time-out in order to continue the testing

as far as possible : refinement. The used delays are generally based on the time values defined in the protocol under test and the environment. Two main cases can be considered : the delay is precisely fixed (maximum value, case 1) or it is specified by a minimum value (case 2). Let us examine what attitude we may adopt to face time-out associated to both cases.

(a) Case 1

Owing to the fact that the value of the delay is known, this case is an error and the testing may be stopped. Otherwise, in order to continue the testing, it is required to lead the EUT to a known state (e.g. Idle) by exchanging appropriate protocol and service messages. If this is possible, the testing can be carried on from this state through another path of the scenario.

(b) Case 2

In this case, the necessary delay is not precisely known. As previously, exchanges of protocol and service messages are executed to determine whether the EUT is in the desired state.

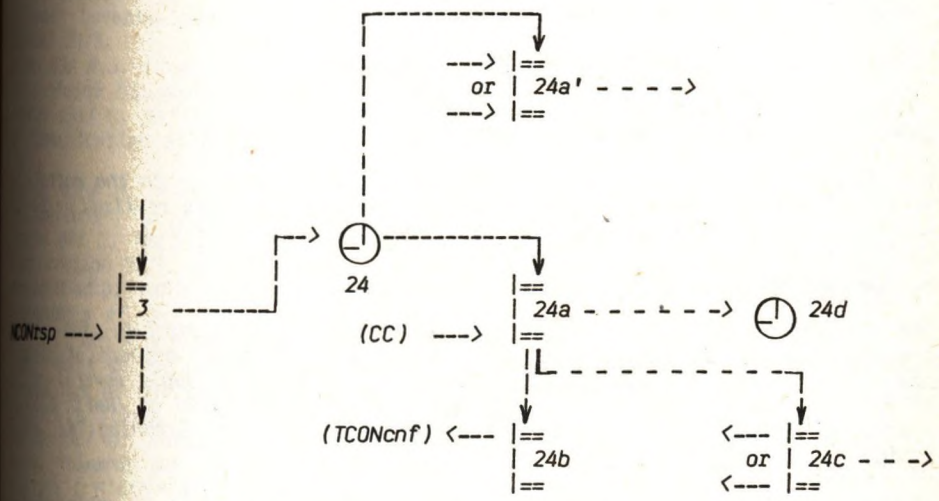


Figure 7. Example of Refinement

Figure 7 shows an example of refinement extracted from the testing graph presented in figure 3. Let be the node 24 and let us try to determine if the EUT has sent a CR TPDU after the time-out. The TS sends a CC TPDU on CUT (node 24a). If the TS receives TCONcnf on CT (node 24b) within a certain time, it is probable that the EUT has sent a CR TPDU, the testing may be finished. Otherwise, either specific exchanges of protocol and service messages may be defined and executed again or the tester decides to stop the scenario.

This approach is very useful in many cases but needs intuition and a good knowledge of the protocol under test. /Haddad 84/ presents several examples concerning the class 2 of the ISO transport protocol.

5.2 Chaining

The integration of error cases, the possibility of refinement and the leading of the EUT to a supposed known state, allow one to construct scenarios which can be

executed without control of a human operator. A scenario can then be executed many times with different values of parameters, in order to cover various aspects of the function under test. It can also be chained with a scenario of a function reachable from the function which has been tested.
Example of a scenarios chaining (scenarios set):

```
BLOCK Testing_of_Transport_Entity;
  PROCESS Transport_Connection_Establishment_from_SUT;
  STATE     Idle ;
        INPUT   Start_Test INTERNAL ;
  ..............
            OUTPUT  Start_Transfer INTERNAL ;
            NEXTSTATE Open ;
  ENDSTATE  ..... ;
  ENDPROCESS Transport_Connection_Establishment_from_SUT;

  PROCESS Normal_Data_Transfer_from_SUT;
  STATE     Open ;
        INPUT   Start_Transfer INTERNAL ;
  ..............
  ENDPROCESS Normal_Data_Transfer_from_SUT;

  PROCESS Transport_Connection_Release_from_TS;
  ..............
  ENDPROCESS Transport_Connection_Release_from_TS;
ENDBLOCK Testing_of_Transport_Entity;
```

The construction and execution of a scenarios set are based on the correlation of the chained functions. The chaining between processes, is realized by sending and receiving of internal messages (e.g., Start_Transfer, etc.).

6. Conclusion

STP allows us to use a technique to construct and describe the testing scenarios, which is particularly related to the classical methods of protocol specification /Rafiq 83a/. This approach will allow us to apply several results of verification and validation techniques. A validator based on /Rafiq 83b/ and a proving technique inspired by the assertions technique /Stenning 76/, are in design.

A first version of an LTP analyser is implemented in UNIX* /C environment by using LEX and YACC tools /Chraïbi 85/.

Finally, several testing scenarios of the ISO transport protocol (class 2), are constructed as well as the ISO session protocol (BCS). These scenarios are used for an interactive testing in an STP model /Rafiq 85c/. This model allows us to carry out several experiments in order to have a good knowledge of the astrice technique usage and its consequences before the actual assembly of TS.

7. References

/Ansart 83/ J.P. Ansart, "Présentation des Techniques et Outils de test du Projet RHIN", Actes des Journées de Présentation du Projet RHIN, Agence de l'Informatique, Paris la Défense, France, Avril 1983.
/Berge 70/ C. Berge, "Graphes et Hypergraphes", Dunod, Paris, France 1970.

* UNIX is a trademark of AT&T Bell Laboratories

Testing Communication Protocols Using Random Legal Inputs

Jerrold Rubin

IBM Architecture and Communications
Research Triangle Park, N.C. 27709
U.S.A

This paper describes a system for testing an implementation of Systems Network Architecture (SNA) by randomly generating legal inputs (verbs) at the architected application interface. This testing system has proven highly successful at disclosing errors in the implementation. Two machines, both using the implementation, maintain the user's view of each conversation state, as well as the control operator's view of the machine state. The conversation state depends upon the last input issued, and upon the result returned from that input. The control operator's view includes the parallelism of conversations and sessions, and the physical and logical connections of the machine.

Each machine randomly selects new inputs from the variety of legal ones at a given instant, and issues the selected inputs via the architected interface. Legal inputs include operator control inputs as well as conversation verbs. The machines continue indefinitely in this manner. Such a system has run for 5 days, generating over a million verbs at each machine, and has uncovered several dozen errors, including some rather subtle timing conditions and deadlocks.

1. Introduction

A number of papers [1,2,4] have been written to describe methods for assuring the correctness of a protocol specification. Fewer [7,8] deal with the correctness of an implementation of the protocol. Perhaps this is because

an implementation may be written in a low-level programming language which makes difficult or entirely prohibits the use of theorems of correctness.

Methods such as state exploration [3,5,6] which try to explore each reachable state by executing each event possible in every state yet reached, will prove impractical with many real implementations, due to the difficulty of re-establishing a given former state with its procedure calling tree, its automatic storage, various hardware flags and the instruction pointer. Checkpointing mechanisms to accomplish re-establishment of a machine state face the difficulty that many states are logically the same, even though the actual storage differs (for example, a buffer may be allocated in a different part of memory). This difficulty is compounded by the fact that some items of data in a given machine state are irrelevant - their value will not affect future operation of the protocol.

The most common method of testing actual implementations is to create a fixed bucket of test cases, designed to execute every feature of the protocol. Often an attempt is made to cover every line of code written, but of course not in every state. The protocol is run with a connection to another machine which implements the same protocol. In early runs, the second machine often executes the same implementation. However, since this allows both machines to misinterpret the protocol in the same way, it is normally considered wise to test against a different implementation - indeed, to be safe, against every other implementation with which the protocol will run.

Attempts have been made to establish a single implementation which will be the standard against which all other implementations of a protocol

must run [8]. This standard implementation may be written in a high level executable specification language, and may have undergone some form of testing, formal or otherwise, to assure its correctness. This approach, while useful, tends to ignore the unique details of each implementation which must also be tested, which the standard implementation cannot know of. For example, the user interface (as opposed to the communication line interface) may be quite different from implementation to implementation.

The method of testing described below cannot be regarded as a replacement for more standard test methods. It, too, has its limitations. It should be considered as an additional testing methodology, preferably to be used after more standard methods have been applied. It is capable of exhibiting errors which simpler test methods will miss - errors which show up not the first or second time a feature is tested, but the tenth or 100th time; errors which depend on the execution of an event with storage in a particular state, or on subtle timing and race conditions; errors which are not reproducible. For much as the automobile that never exhibits its symptoms in the shop, but only on the road, a communications protocol is peculiarly subject to the rare glitch that cannot be captured or repeated.

2. Methodology

Assume a protocol with a clearly defined user interface, which starts in a particular state (for example, a reset state). A set of commands may validly be issued in that state and, depending upon the returned values, the

protocol may transit to a different state, in which a different set of commands is now valid.

The states of the protocol are visible to the user - indeed, he must fully understand the transitions in order to continue to issue valid commands. Typically a single named command will have a number of options and will in fact represent several slightly different commands. In order to fully service the command, several lower levels of protocol will likely be brought into play - for example, there may be a level controlling the data flow (to avoid congestion), a level to perform routing through a network, a level to initiate transactions, a level to manage the immediate link, etc.

Imagine that a communications link exists between two machines, each implementing the protocol - for simplicity, assume that we are dealing with the same implementation on the same type of machines. Whatever the complexity of the underlying protocol implementation, every state in which the protocol may find itself can be generated by executing some set of commands on each machine, with some set of delays between commands, and some set of link generated signals such as polling interrupts (or lack of them). For complete generality, all other forms of interrupts - clock interrupts, interrupts from other ports, etc. should be considered, but this may prove very difficult and will be ignored here.

Interrupts from the link will depend on factors such as line speed and the details of the link control level. It is therefore advisable to use the method with varying line speeds, at least one run per different possible line speed.

In each machine a program simulating a user lists the commands which may validly be issued in the current state of the protocol, and randomly selects and issues one. If several options exist for a command, each valid combination is considered to be a separate command, and is listed separately. For options that are more or less continuous in nature, such as the amount of data to be sent, a single random choice within a fixed interval is made. Data content can be arbitrarily chosen, unless it has an effect on the state of the receiver.

The command may not return control immediately to the user; the protocol may have to wait to receive data or some signal from the opposite side. When control is returned, the returned values are examined and the proper state transition is executed. It may be convenient to maintain a finite state machine for this purpose. Error tests are made on the returned values in accordance with predetermined knowledge of the valid possibilities and, if no error has occured, the valid commands in the new state are listed and a new choice made. This process continues as long as machine time limitations allow - days or weeks for a thorough test.

If an error does occur, the machine is stopped for human examination of machine memory, and of an audit trail of commands and messages on the line. A hardware debugger to record the last N instructions executed can also be most useful, since exact reproduction of the error is doubtful. Timing differences between runs will quickly cause a second run to diverge from the sequence of events executed in the first run. Nevertheless, experience with the technique seems to clearly indicate that a given error can be regenerated

eventually, although perhaps not in the precise circumstances in which it first occured.

A randomly chosen time delay, probably measured in milliseconds, can also be selected. The author has not used this feature, primarily because the errors discovered so often seem to depend on messages appearing too quickly on the line, not the reverse. Also, one of the main purposes for a randomly chosen delay is to assure that commands and interrupts do not synchronize with each other. In a long run, commands and interrupts tend to occur asynchronously without any need for random delays.

A sufficiently long run should guarantee that any error in the implementation will eventually occur, other than

a) those which would be exposed by hardware error

b) those which would be exposed only by a different partner implementation

or c) those which would be exposed by invalidly issuing the commands.

However, an error which does occur may not appear as a condition which will cause the run to stop. (An obvious example of this is the failure of the protocol to deliver all the user data.) This is one weakness of the technique. Still, many errors do exhibit themselves in terms of

1) An invalid returned value which can be tested for

2) A deadlock situation in which both sides are awaiting signals

3) A loop or a stray branch causing an abnormal machine condition

or 4) A recognizable abnormal end (such as running out of storage or detecting an invalid finite state machine transition)

This inability to test if the protocol did indeed execute the intended function is one reason why more normal testing methods should still be applied. The requirement to test error conditions is another. Still, as a stress tester for the valid sequences, the method seems to be very valuable.

The length of run necessary to gain real confidence in the implementation is very much a matter of judgment. In practice, one examines the increasing times of the runs, and if a sufficiently large jump occurs - say a jump from one day to five days without a recognizable error - one proceeds to celebrate success. Added confidence can perhaps be gained by re-running with a different random seed, though this is debatable - the continuation of a run with an unbiased seed should be just as good as using a new seed.

3. Advanced Program to Program Communication

The methodology described above has been applied to an implementation of the part of IBM Systems Network Architecture (SNA) known as Advanced Program to Program Communication (APPC) [9,10,11]. This is a general protocol designed to permit communications between any two programs in any machines in an SNA network. APPC is basically a half-duplex protocol in which one partner has the right to send, until he relinquishes that right to the

other partner. A brief description of the implemented user interface commands follows:

ALLOCATE	establish a conversation with a named partner
SEND_DATA	cause data to be moved into a buffer, to be sent to the partner either when the buffer fills, or as a result of another command (e.g., FLUSH or CONFIRM)
FLUSH	flush out the send buffers - send the data to the partner
CONFIRM	wait for an indication that the partner has received and processed all previous data without error - i.e., synchronize with him
PREPARE_TO_RECEIVE	give the partner the right to send. There are CONFIRM and FLUSH flavors for this command.
POST_ON_RECEIPT	post the application when sufficient data has arrived
WAIT	delay until the application has been posted (used in conjunction with POST_ON_RECEIPT)
RECEIVE_AND_WAIT	wait until a logical block of data or signal is received from the partner, and receive any data into a user buffer
REQUEST_TO_SEND	request permission from the partner to enter send state
SEND_ERROR	indicates to the partner that error data must flow, and guarantees that send state will be entered so that the error data may be sent
CONFIRMED	sent in answer to a received CONFIRM (SEND_ERROR may be issued instead)
DEALLOCATE	end the conversation (issued in send state)
DEALLOCATE_ABEND	end the conversation abnormally (issued in any state)

In addition, there are several commands to examine the current state, but which do not cause changes of state.

The commands have returned values - for example, most commands may return "REQUEST_TO_SEND_RECEIVED" to indicate that the partner has requested

permission to send. (Permission would be granted by a PREPARE_TO_RECEIVE command). Many commands may have multiple returned values - data, a return code for errors, and a WHAT_RECEIVED code to indicate whether data or some signal (DEALLOCATE, CONFIRM, REQUEST_TO_SEND, etc.) has been received.

There are a number of options for each of these commands, and different conversation states can result from a combination of the command, the options specified, and the returned values.

Conversations are allocated to "sessions", which are logical connections between partners, presumably maintained over a longer period of time than a conversation. Various error recognition procedures are associated with sessions. Only one conversation is permitted on a session at a time. The number of active or potential sessions can be controlled with a CHANGE_NUMBER_OF_SESSIONS command.

The implementation involves numerous features which are transparent to a user of the conversation commands. These features include pacing of messages to avoid congestion, buffering, flow control, establishment and disestablishment of sessions, assignment of conversations to sessions, network address assignment, network routing, management of hardware errors, storage management, and link management, to name some of the more important.

It is beyond the scope of this paper to present more than a brief summary of APPC, but a fuller description can be found in [11]. The finite state machine below describes the applicable state transitions for a

conversation. The rows are the commands (with returned values). The columns are the states.

A '/' represents a combination which should not occur, either because the command may not be issued in the state, or because the returned value is not possible. In the former case, the random tester will not issue the command; in the latter case, receiving that returned value is regarded as an error.

A '-' indicates that no state change occurs.

A number indicates the state which will result after the state transition.

C = Confirm
F = Flush
A = Abend

Return Codes
ok = Successful completion of the command
dt = data received
se = Send (partner issued some form of receive)
da = DEALLOCATE_ABEND issued by partner
co = CONFIRM issued by partner
cs = Confirm Send (partner issued PREPARE_TO_RECEIVE using Confirm)
cd = Confirm Deallocate (partner issued DEALLOCATE using Confirm)
ep = Partner issued SEND_ERROR, data purged
er = Error on link or when allocating a conversation to a session

COMMANDS	STATES					
	RESET	SEND	RECEIVE	CONFIRM	CONFIRM SEND	CONFIRM DEALLOC
	1	2	3	4	5	6
Conversation initiated by partner	3	/	/	/	/	/
ALLOCATE(ok)	2	/	/	/	/	/
ALLOCATE(er)	-	/	/	/	/	/
CONFIRM(ok)	/	-	/	/	/	/
CONFIRM(ep)	/	3	/	/	/	/
CONFIRM(er)	/	1	/	/	/	/
CONFIRMED	/	/	/	3	2	1
DEALLOCATE(F or C)(ok)	/	1	/	/	/	/
DEALLOCATE(A)(ok)	/	1	1	1	1	1
DEALLOCATE(C)(er)	/	1	/	/	/	/
FLUSH	/	-	/	/	/	/
POST_ON_RECEIPT	/	/	-	/	/	/
PREPARE_TO_RCV(F or C)(ok)	/	3	/	/	/	/
PREPARE_TO_RCV(C)(ep)	/	3	/	/	/	/
PREPARE_TO_RCV(C)(er)	/	1	/	/	/	/
RCV_AND_WAIT(ok.dt)	/	3	-	/	/	/
RCV_AND_WAIT(ok,se)	/	-	2	/	/	/
RCV_AND_WAIT(ok,co)	/	4	4	/	/	/
RCV_AND_WAIT(ok,cs)	/	5	5	/	/	/
RCV_AND_WAIT(ok,cd)	/	6	6	/	/	/
RCV_AND_WAIT(ep)	/	3	-	/	/	/
RCV_AND_WAIT(er)	/	1	1	/	/	/
REQUEST_TO_SEND	/	/	-	-	/	/
SEND_DATA(ok)	/	-	/	/	/	/
SEND_DATA(ep)	/	3	/	/	/	/
SEND_DATA(er)	/	1	/	/	/	/
SEND_ERROR(ok)	/	-	2	2	2	2
SEND_ERROR(ep)	/	3	/	/	/	/
SEND_ERROR(er)	/	1	1	1	1	1
WAIT	/	/	-	/	/	/

4. Extensions to Incorporate Parallelism

Advanced Program to Program Communication permits parallel programs within one machine to issue commands separately, under the control of a multi-tasking operating system. In order to extend the concepts of random testing to this situation, one can set up an environment in which several transaction programs each randomly generate APPC commands. In this environment, we would allow the use of two APPC commands not described above:

TP_STARTED start a new transaction program

TP_ENDED end a specified transaction program

These two commands are also generated randomly -i.e., we start and end transactions (executed in parallel) at any time. Among other things, this will simulate a user ending a transaction before deallocating all its conversations (this is a valid use of the commands).

Of course, a limit must be placed on the number of parallel programs in existence at any one time; otherwise machine resources might be quickly exceeded.

The ability to randomly test features of the implementation which address parallellism goes considerably beyond the random testing of single conversations, and would be expected to expose errors not found when running a single task in each machine.

One final extension is still possible. In this implementation there are commands to bring up and take down APPC. These also have options, such as a hard takedown which does not wait for conversations to end, vs. a soft takedown which does wait. Such commands can also be randomly generated, causing conditions such as line outage at the partner (when polling stops at one side), and re-initialization of the system. It is probably sensible to have a single monitoring program, which may be regarded as a simulation of a control operator, issue these kinds of commands randomly.

5. Staged Runs Using The Random Tester

I have found it wise to run using several stages:

Stage 1 would be a single session, with one conversation at a time. The purpose of this stage would be to test the normal conversation commands.

Stage 2 would add a command to control the number of sessions, so that more than one conversation can proceed simultaneously.

Stage 3 adds parallellism so that several transaction programs can execute simultaneously.

Stage 4 adds random bringup and takedown capability.

One purpose of staging the runs in this manner is to assure that the normal conversation verbs get a thorough workout, without interruption from commands which would be rarely executed in a real environment. In principle this is not necessary if one were willing to run long enough, since eventually any desired sequence would occur. But in practice, few long conversations would occur if APPC were continually taken up and down in a random fashion.

Some dozens of implementation errors were found during the stage 1 runs. The most interesting ones involved timing or race errors. The following are some examples of errors which normal debugging methods would almost undoubtedly have failed to uncover:

1) The program invalidly purged a queue which in almost all cases would have been empty anyway. The rare instance in which it was non-empty involved two "back-to-back messages" over the line - i.e., the machine had no time in which to process the first message before receiving the second. This situation occured only once in an approximately two hour run, and resulted in a deadlock.

2) Interrupts from the Link Control layer should have been inhibited over a slightly larger window than was the case - one instruction larger, in fact. The interrupt would have had to occur at exactly this one instruction with the machine in a particular state, for the error to be found. This situation occured only once in a two day run, and resulted in a storage overwrite and wild branch.

3) Exhaustion of allocated storage resources occured under certain race conditions combined with certain machine states. This situation occured once in an approximately five hour run, and resulted in an error message indicating an abnormal end to the run.

A number of errors in the original architectural description were also uncovered as a result of these runs.

The longest error free stage 1 run continued for 5 days, exec over a million commands at each side.

The stage 2 runs also uncovered numbers of invalidly managed conditions involving the CHANGE_NUMBER_OF_SESSIONS command being issu approximately the same moment from both sides.

At the moment, we are planning the stage 3 and 4 runs, involvi high degree of parallelism. We expect the technique to be very successf uncovering errors in this environment also.

5) Conclusions

It is sometimes difficult to convince someone not directly involve the random runs of their great value. The counter argument expressed is t one would like to know exactly what one has tested. But it is precis because the errors that are left in a protocol implementation after nor testing is finished are the unexpected ones, that random testing is valuable. They are errors for which one cannot design specific tests. It these errors that cause protocols to fail in a manner which cannot be repor because it cannot be repeated.

There is probably no single method which can thoroughly test a comp protocol implementation and guarantee it to be free from bugs - especial when the implementation is written in a layered manner in a low-lev programming language. This methodology comes closer than any I know

establishing a large degree of confidence in the accuracy of the implementation.

Acknowledgments

I would like to thank Colin West for suggesting this method.

References

1 G. V. Bochmann and C. Sunshine, "Format Methods in Communication Protocol Design," IEEE Trans. Communication, Vol. Com-28, No. 4, (April 1980), pp. 624-631.

2 P. M. Merlin, "Specification and Validation of Protocols," IEEE Trans. Communication, Vol. Com-27, No. 11, (November 1979), pp. 1671-1680.

3 P. M. Melliar-Smith, "A Validation Technique for Tightly Coupled Protocols," IEEE Transactions on Computers, Vol. C-31, No. 7, (July 1982), pp. 630-636.

4 C. A. Sunshine "Survey of Protocol Definition and Verification Techniques," Proc. Computer Network Protocols Symposium, Liege, Belgium, (February 1978) pp. F1-1 - F1-4.

5 C. H. West, "General Technique for Communications Protocol Validation," IBM J. Res. Develop., Vol 22, No. 1, (July 1978), pp. 393-404.

6 C. H. West and P. Zafiropulo, "Automated Validation of a Communications Prot the CCITT X.21 Recommendation," IBM J. Res. Develop., Vol. 22, No. 1, (Januar 1978), pp. 60-71.

7 R. M. S. Cork, "The Testing of Protocols on SNA Products - av Overview," IF (1983), pp. 455-463.

8 A. W. Kleitsh, R. T. Winston, "Preparation of DTE'S For Attachment to X.25

Public Data Networks," CCITT, pp. 300-307.

[9] G. D. Schultz, D. B. Rose, C. H. West, J. P. Gray, "Executable Description and Validation of SNA," IEEE Trans. Commun., Vol. Com-28, No. 4, (April 1980), pp. 661-677.

[10] IBM Corp., "Systems Network Architecture Format and Protocol Reference Manual: Architecture Logic" IBM Form No. SC30-3112-1, (1978).

[11] J. P. Gray, P. J. Hansen, P. Homan, M. A. Lerner, M. Pozefsky, "Advanced Program to Program Communication in SNA," IBM Systems Journal, Vol. 22, No. 4, (1983), pp. 298.

LOCAL TESTS FOR DISTRIBUTED TESTING

R. Castanet (*), R. Sijelmassi (**)

* U.E.R. de mathématiques et d'informatique
Université de Bordeaux I 33405 Talence France.
** I.U.T. A département informatique
Université de Bordeaux I 33405 Talence France.

ABSTRACT:

Test centers require test scenarios for normalized protocols. This paper introduces methods for producing test sequences from a formal representation of an entity. Those sequences are used in a simulator in order to produce scenarios for distributed testing. Possibilities of partial testing that increase test efficiency in a part of protocol are proposed along with measurings of the test obtained.

1. INTRODUCTION.

Developments in the area of networks have led manufacturers and software houses to use equipments connected to these networks. Some manufacturers design machines according to internal norms. International norms for communication protocols (ISO,CCITT) have appeared because of the necessity of interconnection between heterogeneous equipments and public networks. These norms seem to be accepted by all the manufacturers. Conformity of particular implementations to these norms must be checked. Similar problems exist for norms defined by big manufacturers like IBM, when some other manufacturers have tried to offer compatibility with SNA protocols. The same thing occurs in FRANCE with ARCHITEL norms (ARC 84A). Hence the necessity for test centers. Several organisms have defined methodologies and systems for protocol testing (ANS 83) (BOC 83) (LIN 83) (RAF 84) (RAY 82).

Various solutions have been proposed, from the arbitrary test (not efficient enough) to program proofs (too complex). Our aim is to provide test scenarios, corresponding to one level of test, for protocol layer organisation. We automatically generate test sequences from the protocol model. Techniques used in this generation are relative to finite state machines with predicates. These techniques can easily be implemented, they also make test-measuring possible.

The generated sequences can be used in local test; in this case the implementation is considered as a "black box" where the only observable points are inputs and outputs. We apply the sequences and then compare the outputs produced with the ones produced by the model. This kind of test is particularly relevant for the implementors who can control their work in a like manner.

A distributed test is more relevant in test centers. An architecture test is designed which consists of:

- implementation under test (IUT),
- entity model,
- responder.

It uses the lower level layers. Using generation techniques before a distributed test allows measuring of executed scenarios. We also introduce different levels of test: an elementary test concerns the normal use of a protocol whereas a limit test regards the reaction of IUT to abnormal request from entities which communicate with it. We can also test a part of protocol. These selective tests are designed either to improve test efficiency on a part of protocol or to detect particular faults.

In (2) we present local test and generation techniques, in (3) we compare the different methods used, in (4) and (5) we examine decomposition methods for selective tests and in (6) we consider the use of generated test sequences in distributed test.

2. AUTOMATIC GENERATION OF TEST SEQUENCES:

We have extended the techniques defined in (NAT 81) and (CHOW 78) for finite state machines to machines with predicates. The input sequences are caculated from the model, then for each of them, we calculate the normal response. We assume here that the IUT is equivalent to a finite state machine with the same number of states as the model. We first define finite state machines with predicates, then we define the errors that can occur when implementing and finally describe two generation techniques.

2.1 Finite state machines with predicates

a. definition

A finite state machine with predicates consists of:

- Q : set of states,
- A : set of input messages,
- S : set of output messages,
- VAR : set of protocol variables,
- ACT : set of actions changing the values of the variables,
- P : set of predicates depending on the values of the variables,
- TRANS : list of transitions ; a transition consits of:
  - q1: source state, a: input-message,
  - p : predicate to be tested, q2 : target state,
  - s : output-message, act : action to be executed

We define a context C as particular values for protocol variables.

We also define initial state and initial context (q0, c0).

From TRANS we can define three functions:

- transition function : $\delta(q1,a,c) = q2$
- output function : $\lambda(q1,a,c) = s$
- trace function : trace $(q1,a,c)$ = act

where q1,a,q2,s,act are such that $< q1,a,p,q2,s,act> \in$ TRANS
and p is true in the context c.

The three functions above can be extended to δ^*, λ^* and trace with

$\delta^*(q1,a.m,c) = \delta^*(\delta^*(q1,a,c), m, trace(q1,a,c)(c))$

$\lambda^*(q1,a,c) = \lambda(q1,a,c).\lambda^*(\delta(q1,a,c),m,trace(q1,a,c)(c))$

$trace^*(q1,a,c) = trace(q1,a,c).trace^*(\delta(q1,a,c),m,trace(q1,a,c)(c))$

where act(c)=c' if the execution of act in the context c leads to the context c'.

b. example

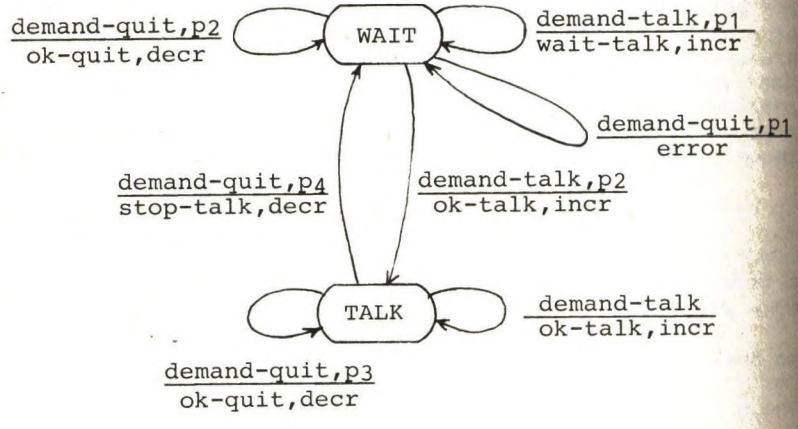
The following machine represents a process which controls the access to a discussion.

Anyone making a request for discussion (by demand talk) must wait if no discussion is on and nobody waiting for discussion. Otherwise he receives an ok-talk. Anybody can leave discussion (by demand-quit). If this occurs when two persons only are discussing, the second one must wait for an interlocutor.

We define:

- Q : wait, talk;
- A : demand-talk, demand-quit;
- S : ok-talk, ok-quit, wait-talk, stop-talk, error;
- VAR : count ;
- P : p1 : (count = 0) ; p2: (count > 0);
  p3 : (count > 2) ; p4: (count = 2);
- ACT : incr (count <- count + 1);
  decr (count <- count - 1).

initial state : wait; initial context : count = 0.



2.2 Different types of errors:

We assume that implementation is equivalent to a finite state machine with predicates whose number of states is equal to that of the model and whose variables take the same set of values as the model's ones. Then three types of errors can occur:

- operation-error: $\lambda(q1,a,c) \neq \lambda'(q1,a,c)$
- transfert-error: $\delta(q1,a,c) \neq \delta'(q1,a,c)$
- action-error: $trace(q1,a,c) \neq trace'(q1,a,c)$

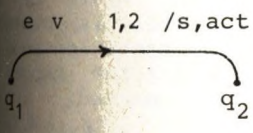
where $a \in A$, $q1 \in Q$, c is an accessible context and $(\lambda, \delta, trace)$ and $(\lambda', \delta', trace')$ are respectively defined for model and implementation. Whereas operation errors can always be detected by testing all states for all inputs, transfer and actions faults can't be detected if they don't induce a wrong output message (ie: a fault in λ').

2.3 Test-sequence generation methods:

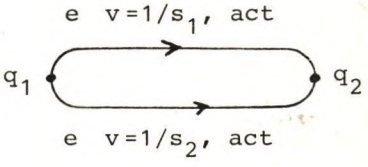
Two methods extended to finite state machines with predicates are discussed here.

a. transition-tour: (TT)

The input sequence to be generated here causes the machine to go trough all its transitions at least once for each possible context that sets the predicate true. We do that to detect the possible error mentioned below.



model



implementation

All predicates are not given in such distinct terms as the example above, thus the problem is to determine the contexts in which the machine can be, when it is at a given state. If we assume that all variables are discrete and bounded, we can construct a finite state machine (without predicate) equivalent to the one we have to test (FASI B4). In the new machine all transitions are simple and TT is practicable. Applying a TT allows to detect operation faults and execution of a wrong action during a transition (ie: an action-error).

A transition-tour can be generated by two techniques:

- random techniques
- graph algorithms.

Random technique: (NAT 81)

Random inputs are applied to the model until the machine has gone through every transition at least once. It is allowed to go through a transition more than once, but cycles with only redundant transitions are removed from TT. Sometimes, processing a TT by this method is too long , especially when the input-set is important, or when there is a large number of cycles in the machine to test.

Graph algorithms:

Many graph algorithms applied to a given machine give a transition-tour for it. Here we present one of it based on a Eulerian-cycle construction after transforming the machine in such a way as it verifies:

$$\forall ei \in Q : deg_int(ei) = deg_ext(ei)$$

where deg_int(ei) = number of incomming edges to ei
deg_ext(ei) = number of outgoing edges from ei

We proceed through three stages:

(1) determine minimal oriented paths between all states (BER 63),
(2) for each pair of states (ei,ej) that verifies:
deg_int(ei) < deg_ext(ej)
deg_ext(ej) > deg_int(ej)
we add a new transition ej --> ei labelled by the minimal path between ej and ei
(3) determine Eulerian-cycle from the initial state by the algorithm given in (BER 63) slightly modified. When it is possible to choose in the algorithm between an added transition and an original transition, we take the latter solution; the algorithm terminates when all original transitions have been marked.

remarks

1) By adding transitions with no labels in (2) one can directly proceed through step (2) and (3) without going through (1). Then we just calculate necessary paths, that may be not minimal.
2) All the techniques above require the machine to be strongly connected. Some transfer and action faults may be not detected.

b. P-W method:

This method is typically designed to detect transfer and action faults. The W set consists of input sequences that can distinguish between states two by two. This distinction is about the generated output sequences.

Two states q1 and q2 are said to be non-equivalent for a common context c if one of the following properties is verified:

(1) $\exists\, m \in A^* / \quad \lambda^*(q1,m,c) \neq \lambda^*(q2,m,c)$
(2) $\exists\, m \in A^* /\ trace^*(q1,m,c) \neq trace^*(q2,m,c)$

W(q1,q2,c) denotes m whenever it exists.

When the following propertie is verified by W, we say that W is total.

(3) $\forall$ q1,q2 and c a common context for q1 and q2 $\exists\ W(q1,q2,c) \in A^*$

otherwise W is said to be partial with a particular cover rate.

- W construction (FASI 84)

W is constructed step-by-step. We first try to separate all the states by input sequences whose length is one. The sequences determined at each step are used to separate new pairs of states in the next step. The algorithm terminates when no new sequence is determined in a given step. This method is an extension of the one given in (CHOW 78) for finite-state machines without predicates.
As for the TT method, it's possible to construct the F.S.M. equivalent to the F.S.M with predicates we have to test before applying the original version of the W-method. But this approach brings about more calculations (FASI 84). On the other hand, using this technique allows to detect also errors that can occur in programmation of actions since they will correspond to a transfer errors in the new F.S.M.

- Use of W-set ; P.W construction:

Let us consider :

t= < q1,e,pred,q2,s,act > $\in$ TRANS
p= acyclic path from initial state and initial context (q0,c0) to (q1,c) , where pred is true for the context c.

If W(q1,q2,c) is defined for each q that can be in context c, let us construct :

$$E = \{ p.t.W(q2,q,c) \quad , q \in Q \}$$

If we assume that all transitions appearing in p are error free, we can detect all transfer ,operation and action faults which can occur in t by applying the sequences in E to the IUT. Note that action faults are treated here like operation faults. We only detect execution of an action whose name does'nt correspond to the one appearing in t.

Thus, for complete testing of the machine, we first construct:

P = the set of all model acyclic paths from (q0,c0)

then p∈ P and W(q1,q2,c) are concatened, iff p is from (q0,c0) to (q1,c) and q2 such that it can be associated with c. The set of all input sequences obtained in a like manner is called P.W and form test sequences to be applied to IUT.

If some of the required W(q1,q2,c) are not found, the test is not complete. In this case, even transition covered by P.W may contain some errors. But an estimation of the efficiency of the test can be carried out. Some extensions as state transition duration can be proposed (TAR 84).

c. Other methods:

There are other methods also based on distinguishing sequences: DS and P.DS; in those cases we have to construct a unique sequence which distinguishes between all states (DS). Obviously this sequence exists less often than the W-set. Those methods are related in (CHDA 83).

3. CONSIDERATIONS ABOUT TESTS AND MEASURES.

It is important, for test centers, to have test measures available. Many kinds of measures are possibles:

- complexity of test sequenc calculations : it is not the most important,since only one sequence-generation is performed for each protocol.
- test sequence lengths,
- test efficiency which will be discussed in § 4 and § 5.

3.1. Complexity of the test sequence calculations:

The two techniques used to construct the TT can be compared. Complexity of the one based on a graph algorithm is proportional to (n +1) where n is the number of states and 1 the number of transitions of 'the model. The probability of succeeding in constructing a TT by the random method in equivalent time is very small.

The complexity of the P.W method is proportional to (n .k) where n is the number of states and k the number of input-messages of the model.

3.2. Test sequence lengths:

The following comparison between test sequence lengths is given in (GUI 84).

length(TT) < length(P.DS) < length(DS).

Note that the length of P.W ,DS and P.DS is the sum of all sequence lengths.

3.3. Choosing a method:

An automatic choice, based on considerations about length, complexity and efficiency of the differents methods, has been implemented. The following schema presents this choice.

DS exists	yes	yes	no	no
W total exists	yes	yes	yes	no
hard reset	yes	no	-	-
automatic choice	P.DS	DS	P.W	TT
detected errors	tranfer, action operation errors			action operation

4. SPLITTING A PROTOCOL.

When we have a complex protocol, with a large number of states it is very difficult to obtain a whole covering by using transfer errors detecting methods (P.W, P.DS, DS). We suggest to split the protocol into independent modules and to test each module on its own. So, it is possible to increase the cover rate for each module. But we have to split the protocol by using heuristic methods (SAR 84). On the other hand, it is not possible to evaluate the power of the union of all the test sequences produced for each independant module. The splitting methods are based on contraction (union of states or transitions) or on projection techniques (LAM 84). Some problems of connexity appear with the projection techniques. For example, it is obvious that it is impossible to have a deconnection without starting with a connection. So it is necessary to make some hypothesis on the step of connection before testing the deconnection phase.

Another way is the design and the implementation of testable protocols. The basic principle of this method is the introduction of transitions that do not disturb the functionalities of the protocol but that increase the cover rate of the tests (PIA 80).

5. SELECTIVE TESTS.

We propose to provide local testing for a protocol particular properties. This proposition induces the notion of selective test. Testing only a subset of protocol is justified by the fact that on the one hand the specifications of some part of the protocol are very clear (with weak risks of errors at implementation level), on the other hand , some other parts are slightly ambiguous. In the

latter case we will speak of ambiguous areas of the protocol. The test center will be particularly interested in these areas.

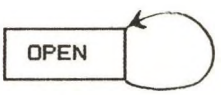
5.1. Ambiguous areas of protocol:

Many examples can be presented to explain the lack of specifications in some protocols. Let us take the case of the OSI transport layer, many solutions can be chosen for the negociation phase according to the class of the transport, or for the choice of timer values that have to be defined by the implementor. A better example for test sequences concerns the implementation of the TCON_REQ. The specification can be understood in two ways:

- previous action of TALLOC_REQ for the allocation of a CEP, followed by a TCON_REQ.
- generation for each TCON_REQ, of a TALLOC_REQ and TCON_REQ.

In the first case, with a sequence of two TCON_REQ we will obtain errors of unspecified receptions after some transitions, because of the allocation of only one CEP.

Another example is the test of data transfer from the state OPEN:



(3): action :"see data transmission procedure"

This specification is clearly too vague. We have to send a number of data corresponding to different limit cases. These are computed with taking into account the window size and the length of the message (segmentation service).

5.2. Software for selective test:

The selective tests deal with a view of a protocol. The tester wants to select some part of the protocol for the test. Analogous technique can be found in external view of data base. To achieve these tests we propose some software which is an extension of a previously developed one (FASI 84) (GUI 84). Let us give a list of the main specifications of selective tests and of the adapted solutions for their fulfilment.

a. Elementary testing:

Elementary testing is used to test a simplified form of a protocol. This form is obtained by projection or contraction techniques and represents the normal run of the protocol.

b. Refinement testing:

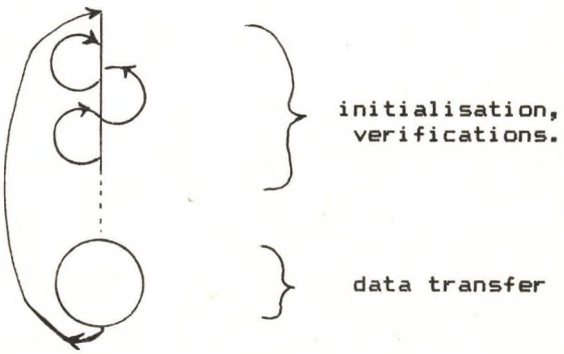
The refinement testing concerns the whole protocol including all

possibilities of errors or rejections. The execution of the test methods (TT and P.W) generates sequences for the refinement testing. Several refinement steps can be proposed :

- local area testing:

In many cases the local area testing is needed:

- for protocols with both complete specified area and ambiguous area.
- for protocols with long preview steps before data transfer



- for testing a particular area of the protocol:connection, disconnection...

The specification of the local area is done in terms of a set of states to be tested.

With the TT method, the test sequences begin with a single path in order to reach the local area, and contain all the transitions allowing to run within the local area.

With the P.W method, only the paths leading to the states of the local area are kept. Only the W(q1,q2,c), with q1 and q2 states of the local area, are built. In this case, we suppose that the paths used to reach the local area are correctly implemented.

- Robustness testing:

Some or all the unspecified receptions have to be tested. These receptions may be unspecified either because the service provided by the entity does not give any information about these situations or because it corresponds to an abnormal run of the entities in communication with the IUT. An algorithm exploring the graph of the entity provides sequences allowing the access to the states containing unspecified receptions and then, fires all (or some) entries. As it is impossible to control protocol behavior in this case, an automatic reset is necessary. Contrarly to the other testing levels, any output from the reference automatom is available to compare the effective outputs with the theoretical ones. The analysis of the results must be achieved by the tester who will use his knowledge.

- Testing without some transitions and/or states:

In some cases it may prove interesting to by pass some irrelevant aspect of protocol behaviour, namely when one has to test an incomplete implementation of the model.

By suppressing the transitions linked with some inputs, other transitions or states may become unreachable. With the TT method those transitions and states are eliminated in order to maitain the strong connexity of the graph. In the PW method the paths including the transitions not to be tested are eliminated from the set P of the partial paths.

The same procedure as above is applied on states. In the TT method, the states to ignore, the transitions linked to those states and the states and transitions disturbing strong connexity are eliminated. In the P.W method, all paths including the suppressed states are eliminated.

- High level test of a given scenario:

A complete scenario made up of input sequence can be built by an user who have a high level knowledge of the protocol to be tested. A simulation language will be proposed to describe the composition of the sequences and to define parametersfor PDU or SDU.

If the sequence is directly applied, only the operation errors will be detected. In order to have a more powerful testing, the W sequences for the transitions of the scenario have to be computed, and then concatened to this scenario.

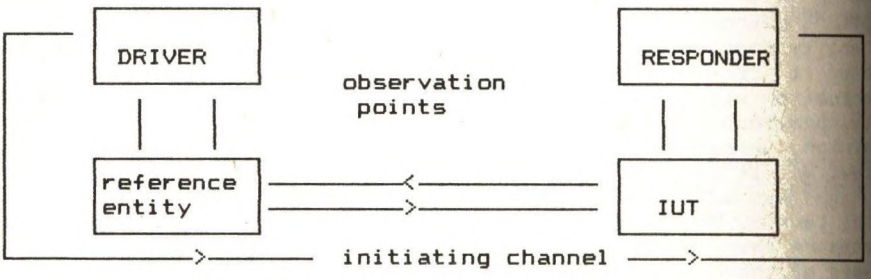
6. TEST SEQUENCE ORDERING.

If the test sequences produced by a local generation are used in a distributed system, some problem may occur: the responder may not have a good understanding of some messages, synchronisation problems between input messages of the IUT may appear. In order to test the produced sequences in a distributed way we propose to use a simulator (SUR 84). Using this simulator, we can either:

- evaluate a given scenario by comparison with generated test-sequences;

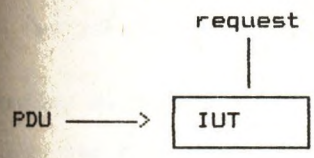
or - try to make those test sequences work in a distributed test.

This simulator is an interactive system, we briefly present it here.



simulated system for distributed testing.

Because of the asynchronous behaviour of the IUT, various sequences can be produced by the execution of a given scenario.



conflict situation.

The figure 3 represents a conflict situation between a request and a PDU. Which one of the two inputs has to be first taken into account?

Sixteen global actions of the system are enumerated: input and processing of the request from the driver to the reference entity, reception of the service indication from the IUT to the responder ... The option taken by the IUT will determine the input sequence and thus the cover rate of the scenario. The user have to use the simulator in order to cover generated test sequences as well as possible.

Another problem is the production of test sequences adapted to the responder. The restriction can be obtained by direct coupling between the IUT and the responder (ANS 85).

7. CONCLUSION:

The different test methods described in this paper can be used either in local test system or in distributed test system. The two ways can be treated either separately at design or implementation steps or complementarily, the local test giving sequences the distributed test. The local test can be used by the implementor. The use of selective test generation allows the following of the development of the implementation. On the contrary, a test center that will give a conformity label to an implementation must use the distributed test. The proposed methodology is made up of three steps:

- generation of test sequences for different levels of test;

- using simulator to design scenarios that produce input sequences for the IUT as near as possible of the generated test sequences. Measuring the obtained cover rate;

- management within a distributed test architecture.

The last two points are linked since the simulator must reproduce the real test architecture which will be used for the distributed test. The proposed software is integrated in global system for description, validation and conformity testing for protocols. It has been achived in cooperation with the french project RHIN. The system will be used for test sequence generation for selective tests on ARCHITEL protocol (ARC 84) in cooperation with the french research center CNET.

8. REFERENCES.

(ANS 83) J.P. ANSART . Presentation du projet RHIN ADI, France. April 83.

(ANS 85) J.P. ANSART . Generation automatique de sequences de test pourle transport classe 0. Rapport technique NOR 7575. RHIN. ADI, France. 1985.

(ARC 84A) Methode generale de test de protocoles ARCHITEL. Note NT/PAA/TIM/1305. CNET, France. 84.

(ARC 84) Protocoles ARCHITEL. Specifications techniques d'utilisation et de raccordement (S.T.U.R.). CNET, France. 84.

(BER 63) C. BERGE. Theorie des graphes et applications. Dunod. 63.

(BOC 83) G.V. BOCHMANN, E. CERNY, M. MAKSUD, B. SARIKAYA. Testing transport protocol implementations. Proceedings of C/PS Conference. Ottawa. Canada. May 83. le transport classe 0. Rapport technique NOR 7575. RHIN. ADI, France. 1985.

(CHDA 83) A. CHAMPEVILLE, V. DAHER. Systeme de generation automatique de sequences de test GAST. Rapport 8310. UER de Mathematiques et Informatique. Universite de Bordeaux I, France. Oct 83.

(CHOW 78) T.S. CHOW. Testing software design modeled by finite state machines. IEEE Trans on software engineering, Vol 4.3. May 78.

(FASI 84) J.P. FAVREAU , R. SIJELMASSI. Systeme d'acquisition, de validation et de test de protocoles. Rapport 8422. UER de Mathematiques et Informatique. Universite de Bordeaux I, France. Oct 84.

(GUI 84) P. GUITTON. Description, validation et test de conformite de protocoles de communication. These de troisieme cycle. Universite de bordeaux I. Janv 84.

(LAM 84) SS. LAM, A. V. SHANKAR. Protocol verification via projections. IEEE Trans on software engineering. Vol 10.4. July 84.

(LIN 83) R.J. LINN, J.S. NIGHTINGALE. Testing OSI protocols at the National Bureau of Standards. Rep. N° ICST/SNA, 83-1.

(NAT 81) S. NAITO, M. TSUNOYAMA. Fault detection for sequential machines by transitions tours. Proc. of IEEE fault tolerant computing conference. 81.

(PIA 80) T.F. PIATOWSKI. On the feasibility of validating and testing ADCCP implementations, NBS Trends and Applications.USA May 80.

(RAF 84) O. RAFIQ. Presentation generale de STP.
Note technique. BULL-TRANSAC, Massy France. 84.

(RAY 82) D. RAYNER. A system for testing protocol implementations.
Computer Networks. Dec 82.

(SAR 84) B. SARIKAYA. Test design for computer network protocols.
Ph.D. Mc Gill University. Montreal. Canada. March 84.

(SUR 84) B. SURRAULT. Simulation de l'execution des scenarii de test et qualification.
Rapport 8425. UER de Mathematiques et Informatique.
Universite de Bordeaux I, France. Oct 84.

(TAR 84) K. TARNAY & ALL. Some experiences with test sequence generation in application layer.
Forth International workshop on protocol specification, testing verification. Skytop, Pennsylvania, USA. June 84.

PERFORMANCE EVALUATION OF ARQ PROTOCOLS IN RING TOPOLOGY NETWORKS

M.E. Anagnostou, S.A. Bacopoulou, E.N. Protonotarios

Division of Computer Science
Department of Electrical Engineering
National Technical University
Athens 157 73, Greece

ARQ protocols are typically used by the Data Link layer of computer communication networks in order to ensure reliable communication over a noisy channel. Two of them, namely the Stop-and-Wait and the Go-Back-N ARQ protocol have been successfully analyzed in the past. Discrete time stochastic models have been employed for this purpose. The effect of the use of the above mentioned protocols on the performance of general topology computer networks (loops excluded) has also been studied. In the present paper we investigate the properties of ring topology networks using the Stop-and-Wait or the Go-Back-N ARQ protocols. Discrete time GI/G/1 queueing models are used and the underlying Markov chains are described. Then the performance measures of the protocols are computed for each network node. The performance of the resulting algorithm is tested for various packet arrival distributions. The analytical results are compared with the results of extended simulation runs, and excellent agreement is observed.

INTRODUCTION

Procedures to cope with transmission errors are essential for the proper performance of data communications networks. Errors in transmission through a datalink are usually dealt with by a retransmission mechanism, such as automatic repeat request[1-3] (ARQ). ARQ schemes employ error-detecting codes that allow the receiving node to check for errors in the transmitted packet. If the receiver finds the packet error-free, an acknowledgement (ACK) message is returned to the transmitter. It should be made clear here that the receiver cannot distinguish between error-free packets and packets with undetectable errors, and that such undetectable errors cannot influence the performance evaluation presented in this paper. The detection, however, of existing errors causes the generation of a negative acknowledgement (NACK) message.

The transmitting node is supposed to retain copies of the transmitted packets, and when an ACK is received, the corresponding copy is released. If either a NACK or no ACK in a suitable period of time is received or an error in the ACK/NACK is found, then retransmission is demanded. Several ARQ protocols have been devised[4-6] which differ mainly in the procedure they use during the retransmission process. Hybrid protocols, which use codes with both error-correcting and detecting capabilities have been proposed[7]. According to these schemes, retransmission is requested only for packets with uncorrectable detected errors.

In an early paper Benice and Frey[8] presented an analysis of the throughput rate and the undetected block error probability for the Stop-and-Wait (SW) and the Go-Back-N (GBN) ARQ protocols. It was not understood until much later, that a theoretical analysis of more sophicticated performance measures of the protocols

e.g. delays and queue lengths, would not become possible unless some details of the protocols were sacrificed for the sake of simplicity and mathematical tractability[5,9]. In the meanwhile, experimental analyses were devised[10,11]. Some tractable problems were also examined[12,13,14].

In 1979 Towsley and Wolf[9] presented a queueing analysis of the SW protocol and the GBN protocol. Also remarkable is the work of A.G. Konheim[5]. Both papers use discrete time models and assume independent block errors. Also in both papers two nodes (a transmitter and a receiver) are considered.

In a previous paper[15] we investigated the properties of a tandem network, in which the SW or the GBN protocol is used. The comparison of the results of that analysis with extended simulation results has shown that it is possible to approximately consider the arrival processes at each node as renewal processes for the sake of mathematical tractability without considerable loss of accuracy. The same method was also applied in the case of an acyclic network[16] with excellent results.

A careful examination of the analysis method used in the previous two papers[15,16] suggests that general topology networks could possibly be analyzed in almost the same way. Moreover the method is general, and can be used for the analysis of general networks of discrete time queues[17]. As we have shown in Ref.17, a queueing network, which can be modeled as an acyclic graph can be analyzed much more easily than a network not possessing this property. In acyclic networks the computation for each node is performed only once, whereas repetitions of the computations are required for non-acyclic networks. The simplest non-acyclic network is a ring.

In the next section we describe the model, which is adopted. Then the analysis is presented. In the final section various examples are given.

THE MODEL

Apart from the links the network consists of N nodes, which form a ring topology network. Each node sends packets to the next node of the ring and accepts (/rejects) packets sent to it by the previous node. Each node serves also as an input/output point of the ring. Therefore there are two of sources of the packets which enter each node : a) The previous node and b) a source external to the ring. Also, a packet arriving at a node is either forwarded to the next node or quits the ring. Therefore each node has two inputs and two outputs. The packet flow is considered unidirectional.

Time is concidered discrete, that is the time axis is divided into fixed length intervals called slots. The slot duration is taken equal to the transmission time of a packet, and the slot duration is considered as the time unit. All packets have the same length.

Packets, which enter the ring at the k´th node follow are characterized by their interarrival times. Packet arrivals occur just after the beginning of each slot (at t+0, where (t,t+1) is a slot). Therefore the packet interarrival times assume only non-negative integer values. Let X_i^k be the interarrival time between the k'th and the (k+1)'th packet, which arrive at the i'th node, either from an external source or from the (i-1)'th node. If the elements of the set $\{X_i^k : i=1,2,..,N\}$ (where N is the number of nodes of the ring) were mutually independent, identically distributed random variables, the node performance could be analyzed as shown in Ref. 15 (see also below). Also, the distribution of the packet interdeparture times could be derived.

If only the packets coming from external sources are considered, let Y_i^k their interarrival times. Similarly, let Z_i^k denote the interarrival times for packets, which have left the (i-1)'th node and arrive at the i'th node. The elements of the set $\{ Y_i^k : k=1,2,... \}$ are assumed to be mutually independent identically

distributed random variables, and the distributions

$$\Pr \{ Y_i^k = m \} = y_i(m) \qquad \begin{matrix} i = 1,2,\ldots,N \\ m = 0,1,2,\ldots \end{matrix} \tag{1}$$

are assumed to be known.

The process Z_i^k does not include all packets, which leave the (k-1)'th node. It is assume that each packet, which is succesfully transmitted from the (i-2)'th node to the (i-1)'th node is forwarded by the (i-i)'th node to an external sink with probability p. Obviously, it is transmitted towards the i'th node with a probability equal to 1-p.

Generally, even if X_i^k (k=1,2,..) were a renewal process, Y_{i+1}^k (k=1,2,...) would not be a renewal process. We approximately assume that the r.v.'s X_i^k (k=1,2,..) describe a renewal process. Moreover, the processes Y_i^k (k=1,2,...) and Z_i^m (m=1,2,...) are easily combined into the composite process X_i^l (l=1,2,...). The way to do it is described in Ref. 16 (and below, in the next section).

Transmission errors are considered to be independent on each data link. Let N_k^m be the number of transmissions required for the successful delivery of the m'th packet from node k to node k+1. No explicit expression for the probability distribution function of this random variable is assumed for the analysis. However, in the examples presented later, the results are given for the case where a constant probability P_k of a successful transmission from node k exists, which means that N_k^m follows a geometric distribution. The crucial assumption is that a sequence of independent random variables R_k^m, which represent the time spent for the successful transmission of the m'th packet from node k, k=1,2...,N,m ≥ 1, can be defined. This assumption implies that the type of ARQ policy employed should be characterized by the property that successful transmission of a packet cannot occur if its previous packet is not successfully transmitted. Therefore, the case of selective ARQ[5] is excluded from the analysis.

Under this assumption, the delay analysis can be handled if a "modified" queueing model is defined, in which each arriving packet carries R_k^m work units. This is equivalent to saying that the m'th packet has a service time of R_k^m slots, at node k ; that is in the "modified" model, the m'th packet is transmitted successfully after R_k^m slots from the moment it entered the head of the queue. It should be noted here that the time m'th packet remains in transit in the buffer of the actual system may be larger than R_k^m .

Two specific ARQ protocols which can be analysed as described above, are the Stop-and-Wait ARQ and the continuous ARQ[1,4,5,9] schemes. In a Stop-and-Wait ARQ scheme, a packet m transmitted for the first time is stored in a buffer and any subsequent transmission stops until either an ACK message for packet m is received, a NACK is received, an error is detected in the ACK/NACK message or no ACK/NACK is received during the next s slots. If an ACK is received, the packet transmission is successful and the transmission of the next packet waiting in the queue starts. If a NACK is received, or if no ACK/NACK is received, or an error is detected, packet m is retransmitted. Obviously, in this case, the random variables R_k^m and N_k^m are related as follows

$$R_k^m = (s+1)\, N_k^m \tag{2}$$

since s+1 slots are required for each transmission and reception of an ACK/NACK.

A GBN ARQ system behaves like the Stop-and-Wait ARQ scheme, except that after the transmission of the m'th packet, the (m+1)'th packet is transmitted, and so on (if there are packets in the queue). A copy of all transmitted packets is retained. If for a packet m', the time-out of s slots expires without reception of an

ACK, then, in the next slot beginning with packet m', all subsequent packets are transmitted again, whether their transmissions were error free or not. That is, ACK/NACK messages for packets subsequent to m' are discarded. Apparently, R_k^m and N_k^m are now related as follows

$$R_k^m = (s+1)(N_k^m - 1) + 1 = (s+1)\, N_k^m - s \qquad (3)$$

Of course, the m'th packet remains stored in transit for $(s+1)\, N_k^m$ slots until it is successfully transmitted after $R_k^m < (s+1)\, N_k^m$ slots. Nevertheless, knowledge of the statistical behaviour of R_k^m is enough for the computation of delays according to the proposed model.

The delay analysis described below uses only the assumption of existence of independent random variables R_k^m. This means that the following analysis is not restricted to the two cases of ARQ schemes presented above, and can be applied to more general situations. For example, the hybrid ARQ protocol described in Reference 12 could be used, instead of the simple Stop-and-Wait or continuous ARQ schemes.

DELAY ANALYSIS

As described in Ref. 15, the random variable W_k^m, which describes the time that packet m has to spend in node k until it is successfully transmitted can be related to the node waiting time of the previous packet, W_k^{m-1}, the interarrival time X_{k-1}^m and the service time R_k^m by the following expression

$$W_k^m = (W_k^{m-1} - X_{k-1}^m)^+ + R_k^m \qquad (4)$$

where the notation A^+ means

$$A^+ = \max(A,0) \qquad (5)$$

A relationship connecting interarrival times X_{k-1}^m at node k to interarrival times at node k+1 is also needed. Since X_k^m can be viewed as the interdeparture time from node k, it holds that

$$X_k^m = t_k^m - t_k^{m-1} = X_{k-1}^m + W_k^m - W_k^{m-1} \qquad (6)$$

and using equation (4) the following expression is obtained

$$X_k^m = (X_{k-1}^m - W_k^{m-1})^+ + R_k^m \qquad (7)$$

In the steady state the variables W_k^m $(m=m_o,m_o+1,\ldots$ where $m_o \to \infty)$ are identically distributed. By the approximate assumption stated in the previous section the random variables X_k^m, for different values of m, are identically distributed too. Finally, the variables R_k^m are identically distributed by definition. Therefore it is plausible to set:

$$W_k(n) = \Pr\{ W_k^m = n \} \qquad (8)$$

$$X_k(n) = \Pr\{ X_k^m = n \} \qquad (9)$$

$$R_k(n) = \Pr\{ R_k^m = n \} \qquad (10)$$

Then Eqs. (4) and (7) yield respectively

$$W_k(n) = R_k(n) \sum_{i=1}^{\infty} X_k(i) \sum_{j=1}^{i} W_k(j) + \sum_{j=1}^{n-1} R_k(j) \sum_{i=0}^{\infty} X_k(i)\, W_k(i+n-j) \qquad (11)$$

and

$$X_{k+1}(n) = R_k(n) \sum_{i=1}^{\infty} W_k(i) \sum_{j=0}^{i} X_k(j) + \sum_{j=1}^{n-1} R_k(j) \sum_{i=1}^{\infty} W_k(i)\, X_k(n-j+i) \qquad (12)$$

By letting

$$X_k(n) = 0, \quad \text{for} \quad n > N_x \qquad (13)$$

and $$W_k(n) = 0, \quad \text{for} \quad n > N_w \qquad (14)$$

where N_x, N_w are sufficiently large integers, Eqs (11) and (12) can be used for the iterative computation of the delay distribution. The algorithm to do it is obviously the following :

1. Set

$$\hat{W}_k(i) = R_k(i) \quad , \qquad i = 1,2,\ldots, N_w - 1$$

$$\hat{W}_k(N_w) = 1 - \sum_{i=1}^{N_w-1} \hat{W}_k(i)$$

2. Obtain a new estimate $\hat{W}_k(i)$ for $W_k(i)$ from equation (11)

3. Repeat step 3 until a convergence criterion for $\hat{W}_k(i)$ is satisfied. Then $W_k(i) = \hat{W}_k(i)$

4. Compute $X_k(n)$ from equation (12).

What remains is to find a method to compose and decompose traffic flows. Let X_1 be the interarrival time between two packets of the process 1 and X_2 be the interarrival time of the process 2. It can be proved (see Ref. 16) that the interarrival times of the composite process are given by

$$Pr\{X=k\} = E\{X\} \left[Pr\{ R=n \} - Pr\{ R=n+1 \}\right] \qquad (15)$$

where R is the residual life corresponding to X . R is given in terms of the residual lifes R_i (i=1,2) of X_i by

$$Pr\{ R_i=n \} = \frac{Pr\{ X_i \geq n+1 \}}{E\{X_i\}} \qquad (16)$$

and $$Pr\{R=k\} = Pr\{R_1=k\}\, Pr\{R_2 > k\} + Pr\{R_2=k\}\, Pr\{R_1 > k\} +$$

$$+ Pr\{R_1=k\}\, Pr\{R_2=k\} \qquad (17)$$

The decomposition of an arrival process characterized by interarrival times X into two process of interarrival times X_1 and X_2 under the assumption that a packet of the initial process becomes a packet of the first process with prob. p, is performed as follows[16] :

$$\Pr\{X_1=k\} = \sum_{n=1}^{k} A_n(k)(1-p)^{n-1} p \qquad (k \geq 1) \tag{18}$$

where

$$A_n(k) = \sum_{i=n-1}^{k-1} \Pr\{X=k-i\}\, A_{n-1}(i) \tag{19}$$

Now the analysis method should be obvious :

1. Assume an arbitrary interarrival time distribution for node #1.

2. Set k = 1.

3. Compute the delay and the interdeparture time distribution for node #k.

4. Decompose the flow of packets leaving node #k in two streams, one quitting the ring (with prob. p), and one entering the node #(k+1).

5. Compose the stream coming from the node #k with the stream of packets entering the ring at the node #(k+1) for the first time.

6. Repeat steps 3-6, until k = N.

7. Repeat steps 2-6, until a convergence criterion for the distributions involved is satisfied.

RESULTS

The results we present here have been organized into two examples, as follows :

Example 1 : The ring consists of four nodes, each receiving packets from sources external to the ring with mean interarrival time equal to 4.5 slots. The block error probability e equals 0.05.The round trip delay s equals 5 slots. A packet received by the i'th node quits the ring with prob. 0.4 if i=2,3,4. If i=1 the prob. varies between 0.5 and 0.9. The interarrival time distributions used were the geometric, the uniform, and the binomial one.

Example 2: Same as in example 1, except for the interarrival times of packets, which enter the ring. Their mean interarrival time at node 1 varies between 2.0 and 5.0 slots, and at the nodes 2, 3, 4 it equals 4 slots. Also, the probability that a packet leaves the ring equals 0.5 and the block error probability e equals 0.01 everywhere.

Input distr.	Variable PROB	Node	Mean delay Analytical	Mean delay Simulation	Relative Error %	Standard deviation of delay Analytical	Standard deviation of delay Simulation	Relative Error %
GE	0.50	1	4.035	4.114±0.040	1.91	4.615	4.708±0.085	1.97
GE	0.40	2	4.455	4.577±0.046	2.66	5.064	5.216±0.090	2.92
GE	0.40	3	4.744	4.844±0.059	2.07	5.368	5.471±0.083	1.88
GE	0.40	4	4.935	5.075±0.053	2.75	5.566	5.770±0.130	3.53
GE	0.60	1	3.318	3.349±0.018	0.93	3.827	3.838±0.041	0.29
GE	0.40	2	3.896	3.946±0.029	1.27	4.464	4.501±0.049	0.81
GE	0.40	3	4.345	4.410±0.033	1.49	4.947	5.044±0.061	1.94
GE	0.40	4	4.667	4.788±0.056	2.53	5.287	5.427±0.069	2.58
GE	0.70	1	2.851	2.857±0.028	0.22	3.294	3.281±0.047	0.41
GE	0.40	2	3.486	3.523±0.048	1.07	4.014	4.065±0.081	1.26
GE	0.40	3	4.024	4.088±0.050	1.57	4.603	4.681±0.093	1.67
GE	0.40	4	4.438	4.506±0.027	1.50	5.046	5.110±0.041	1.25
GE	0.80	1	2.534	2.519±0.016	0.60	2.919	2.895±0.023	0.85
GE	0.40	2	3.180	3.200±0.020	0.62	3.672	3.709±0.019	0.99
GE	0.40	3	3.768	3.841±0.014	1.91	4.325	4.431±0.048	2.39
GE	0.40	4	4.245	4.292±0.025	1.09	4.841	4.901±0.083	1.23
GE	0.90	1	2.314	2.286±0.009	1.22	2.649	2.618±0.026	1.20
GE	0.40	2	2.948	2.939±0.031	0.31	3.407	3.385±0.056	0.64
GE	0.40	3	3.561	3.610±0.031	1.36	4.098	4.155±0.056	1.37
GE	0.40	4	4.083	4.147±0.033	1.54	4.670	4.742±0.057	1.51

Table I : Mean delay for geometrically distributed interarrival times in example 1

Input distr.	Variable MINT	Node	Mean delay Analytical	Mean delay Simulation	Relative Error %	Standard deviation of delay Analytical	Standard deviation of delay Simulation	Relative Error %
GE	2.00	1	4.894	5.118±0.125	4.37	4.816	5.078±0.231	5.17
GE	4.00	2	2.351	2.432±0.025	3.32	2.279	2.401±0.043	5.11
GE	4.00	3	2.077	2.127±0.013	2.34	1.953	2.027±0.029	3.66
GE	4.00	4	1.976	2.028±0.011	2.54	1.829	1.924±0.032	4.95
GE	2.50	1	3.015	3.110±0.027	3.06	2.903	3.010±0.042	3.55
GE	4.00	2	2.130	2.196±0.010	2.96	2.018	2.119±0.023	4.80
GE	4.00	3	1.996	2.036±0.011	1.95	1.854	1.914±0.022	3.14
GE	4.00	4	1.941	1.979±0.008	1.95	1.785	1.838±0.023	2.91
GE	3.00	1	2.390	2.439±0.008	2.03	2.254	2.314±0.015	2.60
GE	4.00	2	2.013	2.057±0.014	2.14	1.874	1.933±0.026	3.01
GE	4.00	3	1.947	1.983±0.013	1.81	1.793	1.844±0.018	2.80
GE	4.00	4	1.918	1.947±0.007	1.48	1.756	1.795±0.016	2.16
GE	3.50	1	2.078	2.122±0.010	2.04	1.923	1.983±0.017	3.03
GE	4.00	2	1.940	1.972±0.008	1.60	1.784	1.824±0.014	2.19
GE	4.00	3	1.914	1.948±0.007	1.74	1.752	1.804±0.020	2.92
GE	4.00	4	1.902	1.933±0.009	1.59	1.736	1.782±0.018	2.55
GE	4.00	1	1.891	1.918±0.006	1.40	1.722	1.758±0.014	2.07
GE	4.00	2	1.891	1.922±0.011	1.62	1.722	1.768±0.021	2.62
GE	4.00	3	1.891	1.928±0.010	1.94	1.722	1.774±0.024	2.95
GE	4.00	4	1.891	1.921±0.006	1.59	1.722	1.756±0.018	1.96
GE	5.00	1	1.677	1.694±0.006	1.02	1.487	1.514±0.015	1.82
GE	4.00	2	1.828	1.854±0.006	1.41	1.641	1.685±0.009	2.59
GE	4.00	3	1.859	1.886±0.007	1.39	1.682	1.714±0.012	1.90
GE	4.00	4	1.875	1.902±0.006	1.44	1.702	1.743±0.011	2.39

Table II : Mean delay for geometrically distributed interarrival times in example 2.

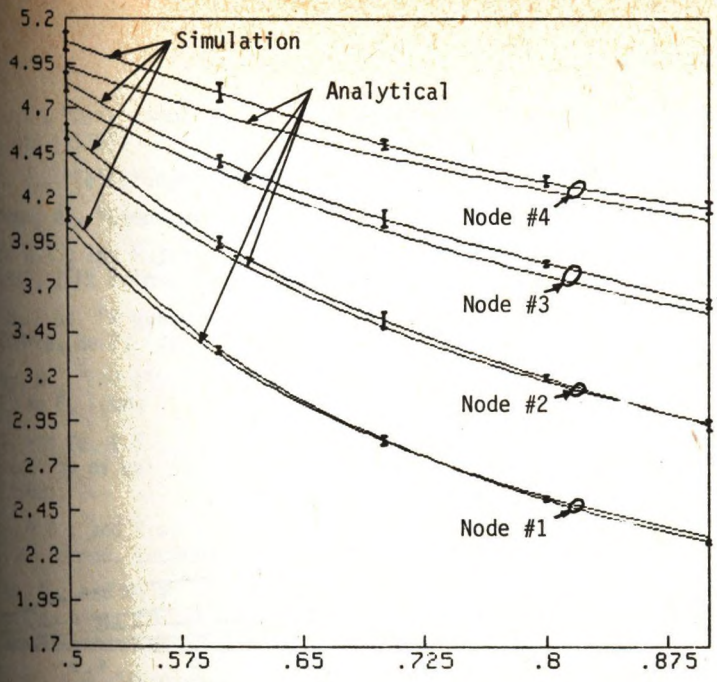


Figure 1
Mean delay versus prob. p in example 1, for geometric arrivals.

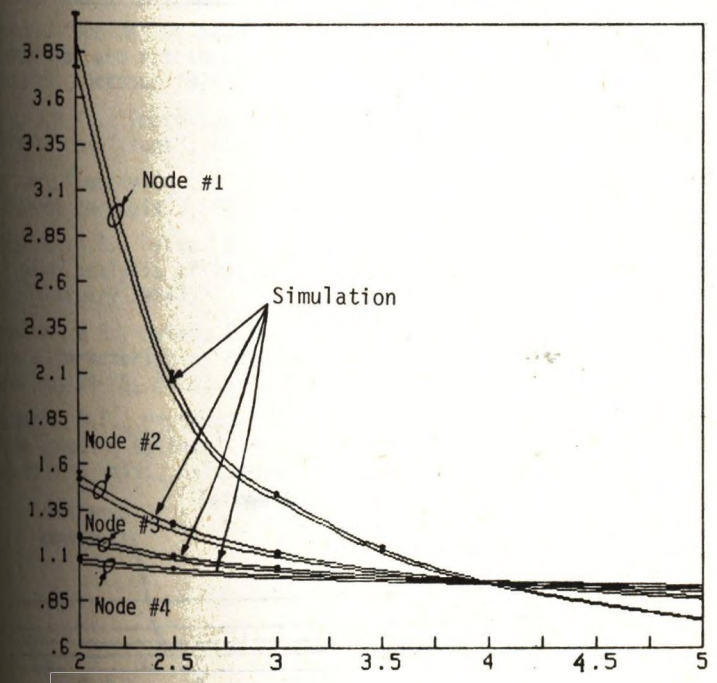


Figure 2
Mean queue length versus mean interarrival time at the first node. Geometrically distributed arrivals.

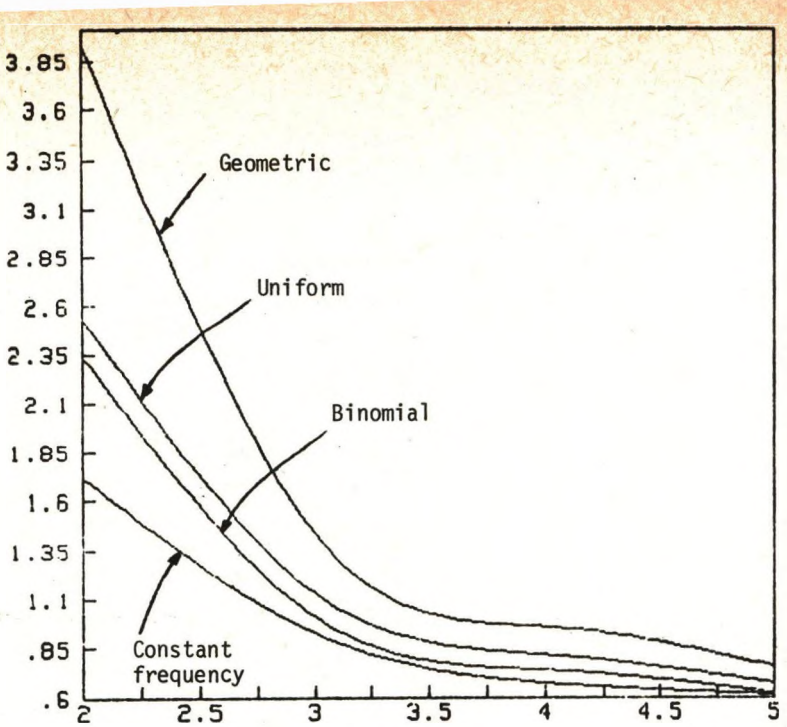


Figure 3
Mean queue length versus mean interarrival time, which varies only at the first node (example 2).

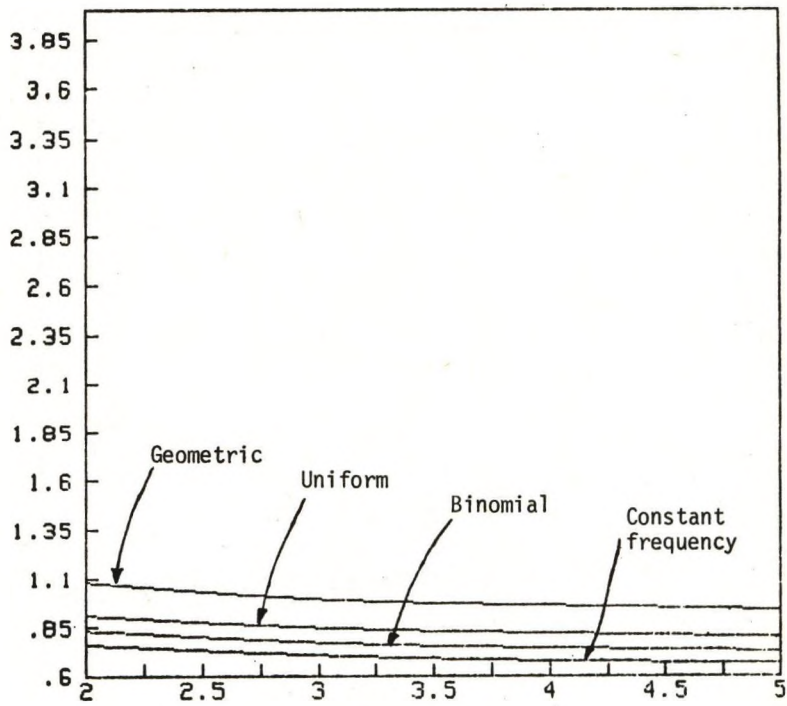


Figure 4
Mean queue length at the fourth node versus mean interarrival time at the first node (example 2).

REFERENCES :

[1] Doll, D.R., "Data communications : facilities, networks and systems design", Wiley, USA (1978).

[2] Martin, J., "Teleprocessing Network Organization", Prentice Hall, USA (1970).

[3] Tanenbaum, A., "Computer Networks", Prentice Hall, USA (1981).

[4] Irland, M.I., and Pujolle, G., "Comparison of two packet-retransmission techniques", IEEE Trans. Info. Theory, Vol. IT-26, No 1 (January 1980)

[5] Konheim, A.G., "A queueing analysis of two ARQ protocols", IEEE Trans. Commun. Vol. COM-28, No 7, (July 1980).

[6] Towsley, D., "The stutter go back-N ARQ protocol", IEEE Trans. Commun., Vol. COM-27, No 6, (June 1979).

[7] Lin, S. and Yu, P.S., "A hybrid ARQ scheme with parity retransmission for error control of satellite channels", IEEE Trans. Commun., Vol. COM-30, No 7, (July 1982).

[8] Benice, R.J., and Frey, A.H., "An analysis of retransmission systems", IEEE Trans. on Commun. Technology, (December 1964).

[9] Towsley, D., and Wolf, J.K., "On the statistical analysis of queue lengths and waiting times for statistical multiplexers with ARQ retransmission schemes", IEEE Trans. Commun., Vol. COM-27, No 4, (April 1979).

[10] McGruther, W.G.,"Throughput of high speed data transmission systems using block retransmission error control schemes over voicebandwidth channels", Proc. of ICC, (June 19-21, 1972).

[11] Burton, H.O., and Sullivan. D.D., "Errors and error control", Proc. of the IEEE, Vol. 60, No 11, (November 1972).

[12] Chu, W.W., "Optimal message block size for computer communications with error detection and retransmission strategies", IEEE Trans. Commun., Vol. COM-22, No. 10, (October 1974).

[13] Sastry, A.R.K., "Improving ARQ Performance on satellite channels under high error rate conditions", IEEE Trans. Commun., (April 1975).

[14] Arazi, B., "Improving the throughput of an ARQ Stop-and-Wait scheme for burst error channels", IEEE Trans. on Commun., (June 1976).

[15] Anagnostou, M.E., Sykas, E.D., Protonotarios, E.N., "Steady state and transient delay analysis of ARQ protocols", Computer Communications, Vol. 7, No. 1, (February 1984).

[16] Anagnostou, M.E., Sykas, E.D., "Influence of the network topology on the performance characteristics of the Go-Back-N ARQ protocol", Proc. Melecon '83, Athens (May 24-26, 1983).

[17] Anagnostou, M.E., Protonotarios, E.N., "Delay analysis of networks of discrete time queueing systems", Proc. of 1985 Conf. on Info. Sciences and Systems, Johns Hopkins Univ., Baltimore, USA (March 27-29, 1985).

PERFORMANCE OF UNSLOTTED ALOHA WITH PARTIAL ACK OF RANDOM LENGTH PACKETS

Józef Woźniak

Department of Electronics
Technical University of Gdańsk
Gdańsk
POLAND

An unslotted ALOHA channel with geometrically distributed packet lengths and possible partial positive acknowlegement of packets is discused. The analysis of the system with respect to the information throughput as a measure of performance quality is presented. Variations of this information throughput are compared with the results obtained for the pure ALOHA protocol. The conditions under which the proposed protocol becomes preferable are given.

INTRODUCTION

A classical example of a computer network, employing packet-switching on a random-access radio channel is the ALOHA system, [1] , [2] , [3] . The simplest protocol used in this system - the pure ALOHA protocol - permits a user to transmit any time it desires. It causes mutual collisions of packets. This fact indeed limits the value of the average channel throughput S . The first analysis, carried out by Abramson [1] , showed that; for fixed packet lengths, the maximum channel throughput attains 1/2e. Next, Ferguson [4] has proved that usage of random lengths packets always leads to values of the channel throughput smaller than 1/2e. All this analysis were carried out on the assumption that all overlapping packets destroy each other and must be repeated. The existance of positive acknowledgment scheme permits a transmitter to determine whether its transmission is successful or not.However, in many cases when two packets conflict, they will overlap partially. On the other hand users very often generate random length messages [5] . In the paper we are mainly interested in the improvement of the channel utalization for the case when packets lengths are geometrically distributed and we allow partial positive acknowledgment of one of interfering packets.
As a performance parameter we choose the information throughput S_i, i.e, the ratio of successful transmissions of users data only. This parameter seems to be more resonable - from the users point of view - than the channel throughput. The results obtained for this modified protocol are compared with two other cases when the whole interfering packets have to be repeated. In the first of them packets lengths are geometrically distributed. In the second one packets have fixed lengths.

GENERAL ASSUMPTIONS

We consider the unslotted random access channel in three cases:

A - There is a possibility of partial acknowledgment of the first of interfering packets. The lengths of information parts in packets are geometrically distributed.

B - The whole packets have to be repeated. The lengths of information parts in packets are geometrically distributed (the pure ALOHA protocol is used).

C - The whole packets have to be repeated. Packets have fixed lengths (the pure ALOHA protocol is used).

According to the principle of packet (message) switching, each message generated by a user is transmitted with an appropriate header, forming together a packet of the length of

$$L = H + I \tag{1}$$

bits, where H - the length of the header, I - the length of the information part (message).

In cases A and B the lengths of the information parts $I = I^A = I^B$ are geometrically distributed, i.e.,

$$P(I = kL_1) = q^{k-1}(1 - q) \; ; \quad k = 1,2,\ldots \; q \in \langle 0,1) \tag{2}$$

where L_1 denotes the length of one information segment. For these cases the average message length (i.e., the length of the information part) is equal to

$$\bar{I} = \frac{L_1}{1 - q} \tag{3}$$

Further we will assume $L_1 = 1$.

But, in case C the length of information part is fixed and equal to the average length obtained for case A or B.

$$\bar{I}^C = \bar{I} \tag{4}$$

The above given length distributions concern all packets (new and retransmitted) sent over the random access channel.

The principle of the system operation in case A is explained in Fig. 1.

In all considered cases we assume that:

(i) - The number of users tends to infinity.

(ii) - Users present their packets (new or retransmitted) to the channel in such a way that their beginning moments create a Poisson process with the mean input rate of λ [packets/second]. Thus the intervals between beginning moments are exponentially distributed with the density function

$$p(t) = e^{-\lambda t} \; ; \quad t \geqslant 0 \tag{5}$$

(iii) - The random access channel is noiseless.

(iv) - The offered traffic is the same for all three cases and equal to:

$$G = \lambda \bar{L} \tag{6}$$

where $\bar{L}$ denotes the average packet length.

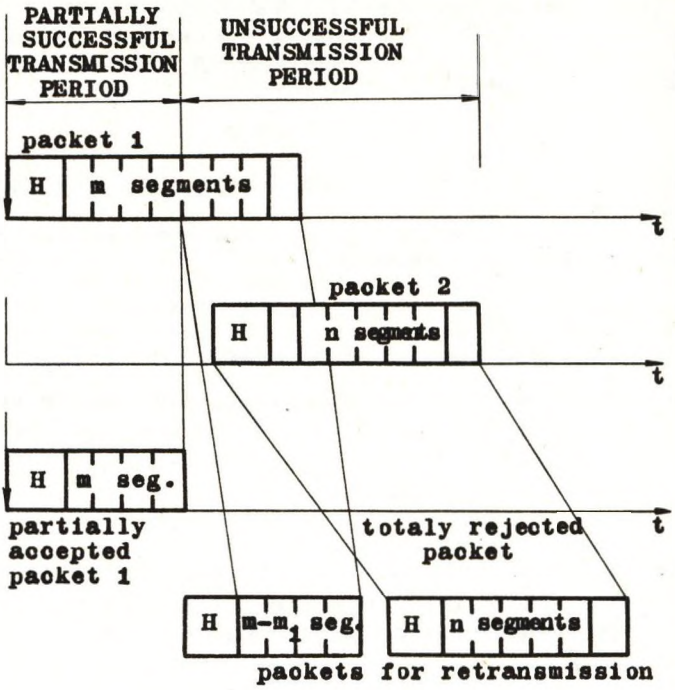


Figure 1

The unslotted ALOHA protocol with partial positive acknowledgment of packets

ANALYSIS OF THE INFORMATION THROUGHPUT

For the evaluation of the information throughput we should consecutively analyse ([6] , [7]) :

(a) The probability P_{us} of unsuccessful transmission of a packet.

(b) The probabilities: P_s - that a packet is successfully transmitted (in case C) or $P_s(k)$ - that a packet with the length H + k appears in the output stream (it may be fully or partially accepted).

(c) The modified distribution probability P_m $(I = k)$; for cases and B, describing lengths of the packets leaving the system we omit unsuccessful transmissions.

(d) The output rate λ_{out} of the packet stream leaving the system

(e) The average length $\bar{I}_s$ of the information parts of the packets leaving the system.

The information throughput, i.e., the rate of successful transmissions of data users only is given by the expression:

$$S_i = \lambda_{out} I_s \qquad (7)$$

For the cases discused in this paper we have, respectively:

CASE A - with partial ACK

$$(a)\quad P^A_{us} = \sum_{k=1}^{\infty} P(I=k)\ P(T_2 < k+H) + \sum_{l=1}^{\infty} P(I=l)\ P(T_1 < 1+H)$$

$$\sum_{m=1}^{\infty} P(I=m)\ P(T_2 \geqslant m+H) = 1 - \frac{(1-q)e^{-2(H+1)\lambda}}{1-qe^{-\lambda}} \qquad (8)$$

$$(b)\quad P^A_s(k) = P(I=k)\ P(T_1 \geqslant k+H) \sum_{i=1}^{\infty} P(I=i)\ P(T_2 \geqslant i+H) +$$

$$+ \sum_{j=k+1}^{\infty} P(I=j)\ P(k+H \leqslant T_1 < k+H+1) \sum_{i=1}^{\infty} P(I=i)\ P(T_2 \geqslant i+H) =$$

$$= (1-q)\ q^{k-1}\ \exp\left[-(2H+k+1)\lambda\right] \qquad (9)$$

In both above equations the random variables T_1 and T_2 describe appropriate intervals, befpre and after the beginning moment of a choosen packet transmission, with or without collision. According to the assumption (ii) both of these random variables are statistically independent.

$$(c)\quad P^A_m(I=k) = \frac{P^A_s(k)}{1 - P^A_{us}} = (1 - qe^{-\lambda})\ q^{k-1}\ e^{-(k-1)\lambda} \qquad (10)$$

$$(d)\quad \lambda^A_{out} = \lambda(1 - P^A_{us}) = \lambda \frac{(1-q)\ e^{-2(H+1)\lambda}}{1 - qe^{-\lambda}} \qquad (11)$$

$$(e)\quad \bar{I}^A_s = \sum_{k=1} P^A_m(I=k) = \frac{1}{1 - qe^{-\lambda}} \qquad (12)$$

Finally we have

$$S_i^A = G\,(1-q)\,d\left[\frac{e^{-(H+1)\,dG}}{1-qe^{-dG}}\right]^2 \qquad (13)$$

where

$$d = \frac{1-q}{1+H\,(1-q)} \qquad (14)$$

CASE B

In this case we obtain the following mathematical expressions:

$$(a)\quad P_{us}^B = 1 - \left[\frac{(1-q)\,e^{-(H+1)\lambda}}{1-qe^{-\lambda}}\right]^2 \qquad (15)$$

$$(b)\quad P_s^B(k) = (1-q)\,q^{k-1}\,e^{-(k+H)\lambda}\,\frac{1-q\,e^{-(H+1)\lambda}}{1-qe^{-\lambda}} \qquad (16)$$

$$(c)\quad P_m^B(I=k) = (1-qe^{-\lambda})\;q^{k-1}\,e^{-(k-1)\lambda} \qquad (17)$$

$$(d)\quad \lambda_{out}^B = \lambda\left[\frac{(1-q)\,e^{-\lambda(H+1)}}{1-qe^{-\lambda}}\right]^2 \qquad (18)$$

$$(e)\quad \bar{I}_s^B = \frac{1}{1-qe^{-\lambda}} \qquad (19)$$

$$S_i^B = Gd\left[\frac{(1-q)\;e^{-(H+1)\;dG}}{1-qe^{-dG}}\right]^2 \frac{1}{1-qe^{-dG}} \qquad (20)$$

CASE C

In this case we modify the length of the information part according to changes of the average length of the geometrically distributed packets lengths, i.e. :

$$\bar{I}^C = \frac{1}{1-q} = \bar{I}$$

After some calculations we have:

(a) $$P^{C}_{us} = 1 - e^{-2\lambda (H + \bar{I})} \quad (21)$$

(b) $$P^{C}_{s} = e^{-2\lambda (H + \bar{I})} \quad (22)$$

(c) $$\lambda^{C}_{out} = \lambda e^{-2\lambda (H + \bar{I})} \quad (23)$$

(e) $$\bar{I}^{C}_{s} = \bar{I} \quad (24)$$

$$S^{C}_{i} = \frac{Gd}{1 - q} e^{-2G} \quad (25)$$

RESULTS

The variations of the information throughput are shown in Figures 3, 4 and 5. From these figures one can see that the information throughput very strongly depends on the operational scheme as well as the distribution of the packets lengths. As it could be expected the best results are obtained for case A, i.e., for the system with partial positive ACK of packets. In Fig. 2 the information throughputs S_i versus the offered traffic G are plotted. In Fig. 2a and 2b, in fact, the channel throughput variations are shown - the header lengths H are equal to 0. For $q \longrightarrow 0$, i.e., for the mean message length tending to one segment all the systems are identical. On the other hand, for $q \longrightarrow 1$, i.e., if geometrically distributed message lengths are longer and longer (at the mean) the system A becomes significantly better than, two others. Simultaneously for this system we observe bigger values of G (in comparision with systems A and B), for which we get the maximum of S_i. For $q \longrightarrow 1$ we obtain the maximum value of S^{A}_{i} approximately equal to 0.25 at $G = 1$.

Variations of the optimal total traffic G_0, i.e., the value of the offered traffic for which S_i reaches its maximum, are plotted in Figure 3. For case A one can observe the permanent increase of G_0 versus q - from $G_0 = 0.5$, for $q = 0$, till $G_0 = 1$, for $q \longrightarrow 1$. But, for the pure Aloha protocol, and geometrically distributed packet lengths (case B) the optimal value G_0 varies, depending on the values of the parameter q as well as the header length H. For $H \gg 1$, the optimal traffic takes values smaller or equal to 0.5. For any $H \geq 1$ the optimal traffic reaches its minimum value (less than 0.5) inside the interval $q \in (0, 1)$. Additionally, for H tending to infinity this minimum value attaines approximately 0.43. But, investigating changes of the maximum information throughput $S_{i_{max}}$, wher $S_{i_{max}} = S_i (G_0)$ one can observe (see Fig. 4) that in case A, for any value of $H \geq 0$ the value of $S_{i_{max}}$ increases with the growth of q. It means that the bigger values of the informa-

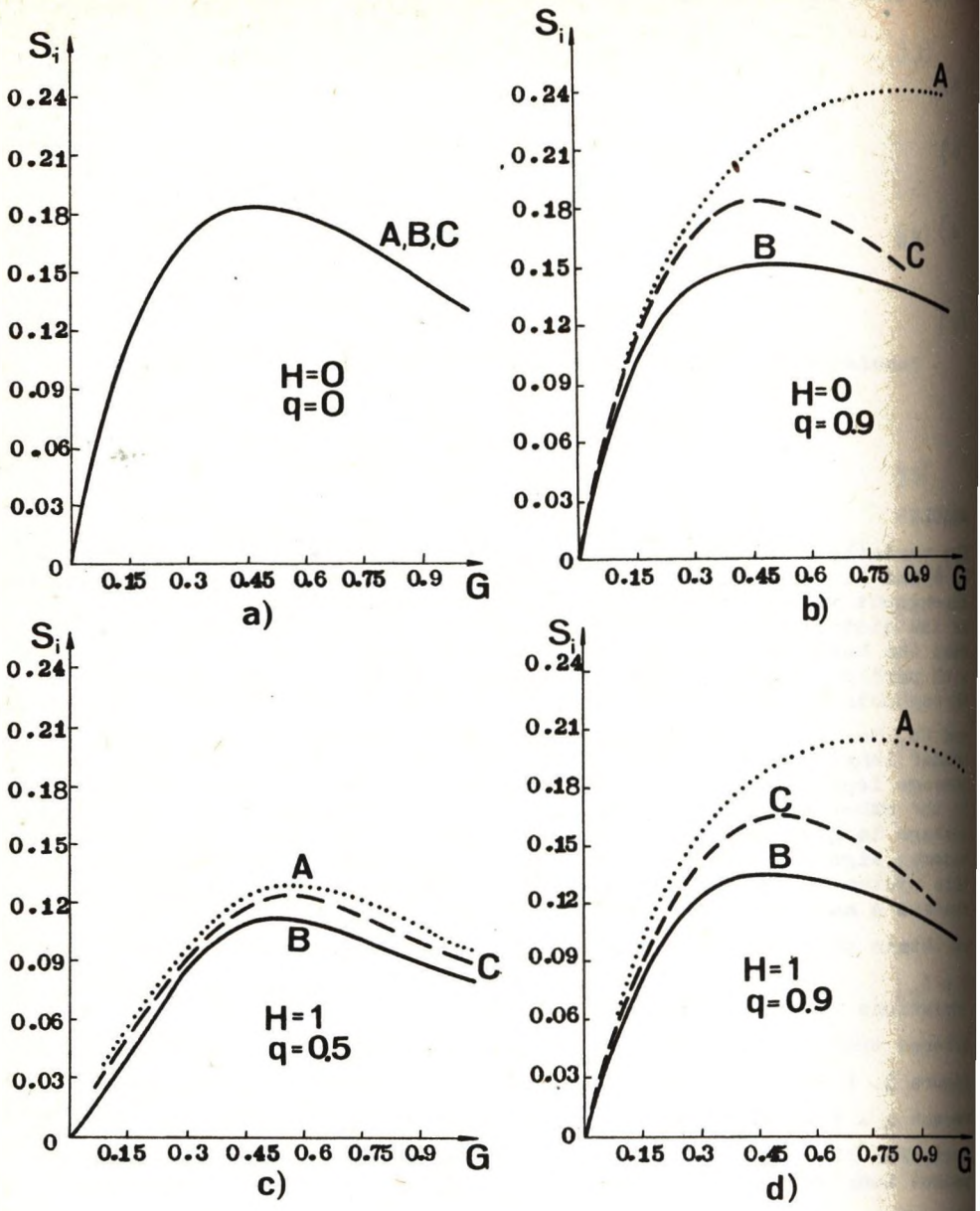


Figure 2

The information throughput versus

the offered traffic

tion throughput as well as the total throughput (for H = 0) are obtained for longer packets, even though such packets have larger dispersion. Influence of partial acknowledgments is very significant.

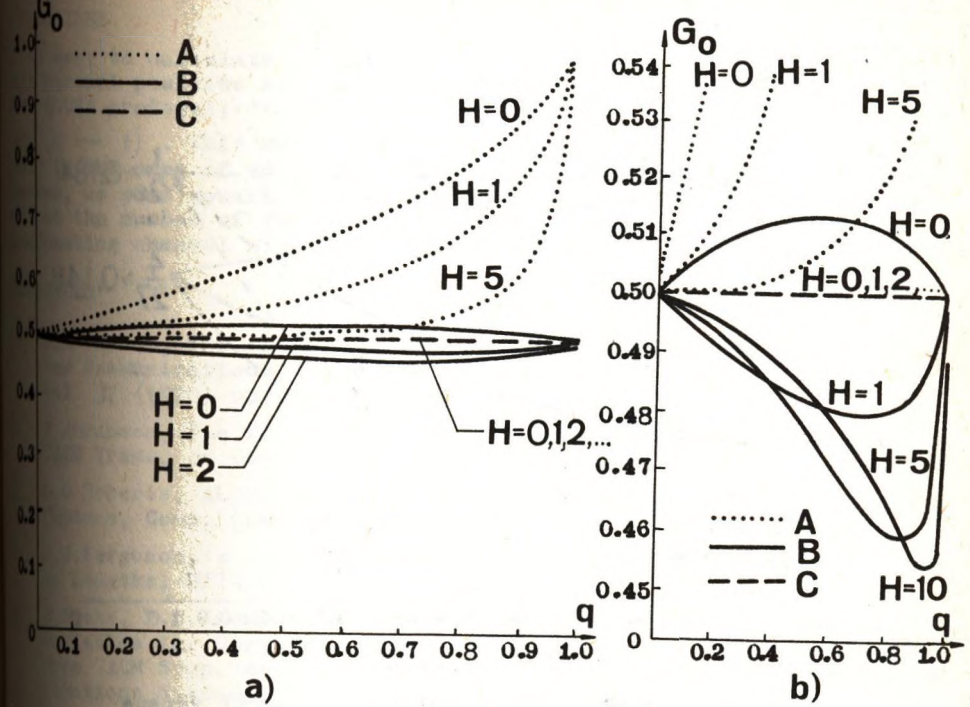


Figure 3

The optimal total traffic versus the parameter q

In case C an increase of $S_{i_{max}}$ versus q is observed for $H \gg 1$, (for $H = 0$ we have $S_{i_{max}} = S = 1/2e$). This increase is caused only by the growth of the packet information part length. Next, in case C, one can see that the total throughput (for $H = 0$) decreases versus q. This is caused by the increase of the packet length dispersion. Simultanously, for $H \gg 1$ one can observe an increase of the $S_{i_{max}}$ up to $4/27$, due to the increase of the length of information parts in packets. In this paper we are mainly interested in the information throughput. However, it has been shown [7] that for geometrically distributed packet lengths the maximum values of S and S_i are reached for different values of G (except $H = 0$).

Next, in Figures 5a - 5c the variations of S_i are plotted versus parameter q. Depending on the value of the input rate λ one can observe changes of the S_i shape. For small values of λ the maximum values of S_i concern very long message (and vice versa).

In all presented cases the worst performance parameters are obtained for case B.

From Figures 2 - 5 one can find conditions under which the usage

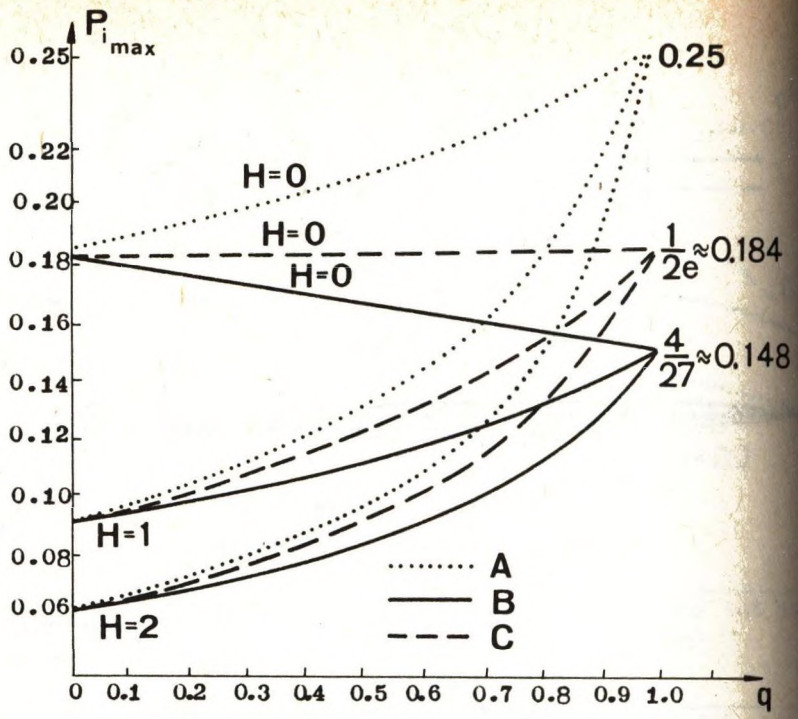


Figure 4

The maximum information throughput versus the parameter q

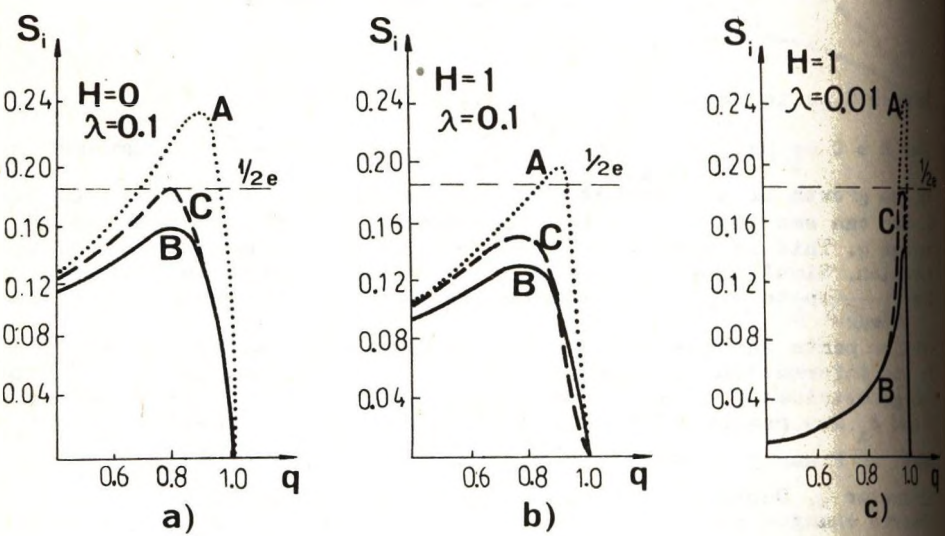


Figure 5

The information throughput versus q

of the system A is particularly profitable. Generally it is always related to very long messages (at the mean).

CONCLUSIONS

It is easy to calculate that usage of the system A, i.e., the system with partial positive acknowledgment gives; in comparison with the pure ALOHA protocol, the increase of the value of S_i in about 30% for $(q \rightarrow 1)$. This means that such a modified protocol is preferable in the case of very long at the mean random packets lengths. However, we must remark that in case A both systems complexity as well as the number of the positive acknowledgments sent over the broadcasting channel significantly increase.

REFERENCES

[1] N.Abramson, The ALOHA System - Another Alternative for Computer Communications, Proc. of the AFIPS Fall Joint Comp. Conf., vol. 37 (1970) pp. 281 - 285

[2] N.Abramson, The Throughput of Packets Broadcasting Channels, IEEE Trans. on Comm., vol. 25 (1977) pp. 117 - 128

[3] L.G.Roberts, ALOHA Packet System with and without Slots and Capture, Comp. Comm. Review, vol. 5 (1975) pp. 28 - 42

[4] M.J.Ferguson, A Study of Unslotted ALOHA with Arbitrary Message Lengths, DATA COMM. Qubec Canada (1975) pp. 5.20 - 5.25

[5] E.Fuchs, P.E.Jackson, Estimates of Distributions of Random Variables for Certain Computer Communications Traffic Models, Proc. ACM Symp. on Problems in the Optimization of Data Communications Systems, Pine Mountain (1969) pp. 202 - 225

[6] J. Cichorek, Optimization of the Packet Length in a Random Access System, M.Sc. Thesis, Dept. of Electronics, Techn.Univ. of Gdańsk (Nov. 1984) /in Polish/

[7] J. Woźniak, Variations of the Channel Throughput fo Geometrically Distributed Packets Lengths in a System with Random Access. Arch. Autom. Telemech. vol. 29 (1984) pp. 377 - 388

PERFORMANCE MEASUREMENTS OF THE COMMUNICATION NETWORK NODE

Iwona Dubielewicz, Zbigniew Fryźlewicz,
Krystyna Koleśnik, Witold Komorowski

Computer Center,
Technical University of Wrocław
Wrocław, Poland

A method of controlling traffic intensity during experiments carried out in the communication subnetwork and its application to the examination of the MSK computer network node behavior are presented in this paper. The SYZYF system of packet traffic simulation is characterized and an example of its command parameters evaluation is given. SYZYF was succesfully used to measure the node's characteristics of the throughput, transit delay and power as functions of packet traffic intensity. The results obtained during these investigation of the node are presented and discussed.

INTRODUCTION

The MSK computer network is a WAN-type network in which the services are implemented using the X.25/3 protocol with PVC and SVC type connections [3]. The node operating system is implemented on a PDP11-compatible minicomputer SM3. The configuration of the MSK network is presented in Fig. 1.

The knowledge of some measures characterizing this network and its components are important to users as well as to network managers. The most popular measures concerning a node are throughput, power and packet transit delay [1,6,7,8,9]; concerning the whole network operation - round-trip delay and the time of standard file transfer [2,6,7].

Throughput is the number of packets emitted from the node, and is expressed in packets/s or bits/s. To avoid the influence of line speed on the results, throughput is often normalized by dividing it by the value of the total capacity of lines connected to the node.

Transit delay (for packets) is the length of time a packet spends in the node; i.e., it is the difference between the time of a packet's arrival and departure. Transit delay can be subdivided into several distinct components, e.g.: delays caused by the input queue, delays caused by processing and delays caused by the output queue. To avoid the influence of line speed on the results, transit delay is often normalized by dividing it by the value of packet transmission time.

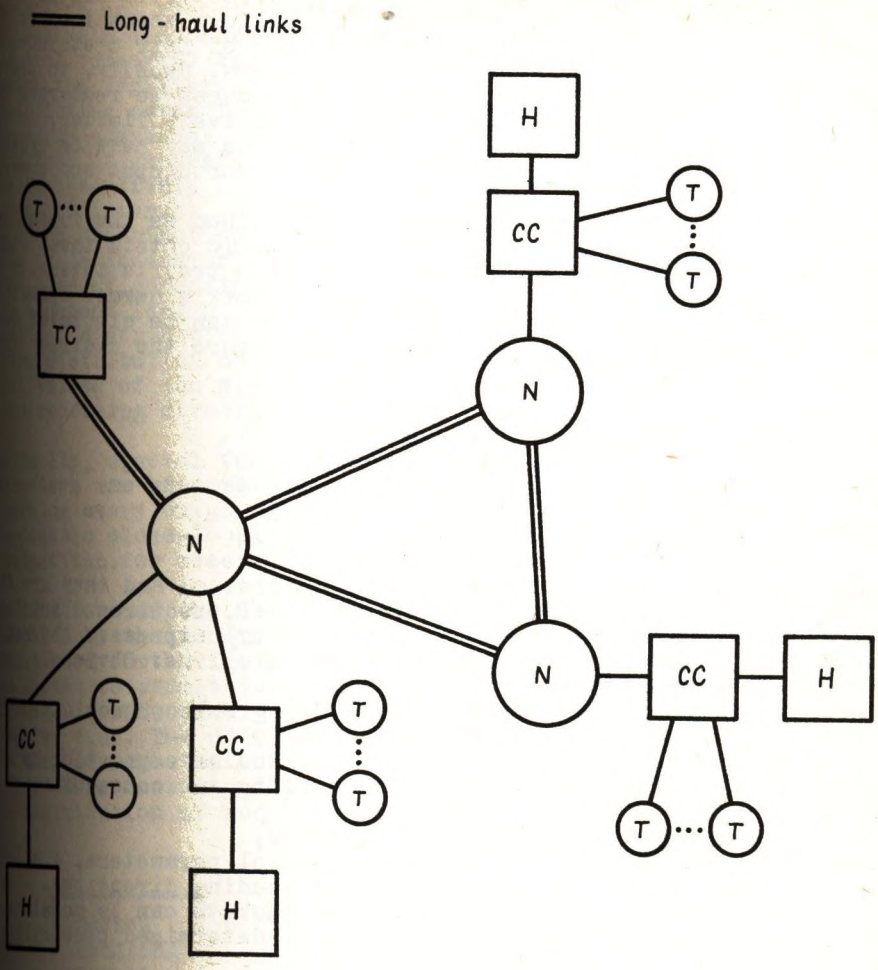


Fig. 1. The MSK computer network configuration (H - host, CC - communication computer, TC - terminal concentrator, N - node, T - terminal).

In general, for a given communication system, the throughput and delay are inversely related, in that higher throughput can be achieved only at the expense of higher delay. Consequently, it has been recommended [4,5] to use a combined performance measure, called power, defined as

$$\text{power} = \frac{\text{normalized throughput}}{\text{normalized delay}} .$$

It is assumed that node throughput determines the node's quality in connection with "file transfer" type of traffic, while transit delay is treated as a quality measure for "interactive" traffic

[1,4,9]. From the user's point of view, the node's features are reflected in its power [4,5], so this characteristic can be used as a global characteristic of the node. Factors which improve the node's power often act contrary to the rule of fair distribution of the node's resources among computing users. Usually, to prevent such distribution problems, the amount of access to resources that each user (or user group) may have is selectively limited, in spite of the negative effect that such restrictions may have on dynamic resource sharing, and, therefore, on total throughput and power.

The characteristics listed above are determined as the functions of the node or network offered load [9,10]. To obtain these functions it is necessary to control the packets traffic intensity but, it is impossible to do this during the network's normal operation. To overcome this difficulty, packet traffic can be simulated in the existing network in such a way as to ensure the desired intensity.

CONTROLLING PACKET TRAFFIC IN THE NODE

The number of incoming packets to the node depends on:
- the number of lines connected to the node,
- the line capacity,
- the offered load on every line.

The first and second factors are well understood and they are fixed in a given configuration. The third, however, requires clarification. The offered load (or traffic intensity) expresses the relative line load according to capacity of this line. Obviously, the offered load can be changed by:
- varying the number and type of acting logical connections on the line (the number of SVC/PVC connection),
- varying the traffic intensity in a particular connection by changing the time interval between sending two consecutive packets,
- varying the length of generated packets,
- changing some of the packet-level protocol parameters, i.e., the window size in both the receiving and sending directions.

The time interval and the length of the packets can be constant or randomly variable, generated according to determined probability distribution.

Obtaining such a wide spectrum of changeable parameters is possible only if a programming simulation of packet traffic is used. The method proposed in this paper was applied the software-hardware system called SYZYF to generate packets in every measured connection, with length and interval adequate to the given parameters. To ensure easy and flexible modification of the parameters, they are introduced to the system as operator command parameters initializing the operation of the measurement system.

THE SYZYF MEASUREMENT SYSTEM

The SYZYF measurement system enables two-way data transmission to a fixed point in the network. These data transmissions take place in the connections on the transport level implemented in SYZYF. One can use physical I/O devices (i.e., paper tape reader, keyboard, disc memory) as well as "logical" devices (i.e., generators of data packets), especially for measurement purposes.

The operator commands are used to:
- fix the initial state of the system,
- establish and accept transport connections,
- selectively disconnect transport connections,
- control temporary traffic intensity,
- print the registered data.

The following parameters can be defined in the set-up/confirmation connection commands:
- the type of generator used for establishing the interval between two consecutively sent packets (a constant interval, or an assigned interval, according to exponential distribution [1,10], uniform distribution, or nonuniform distribution - determined by the table generator),
- the type of generator used for establishing the length of packets (a constant length [10], or an assigned one, according to exponential distribution [2,5,8], uniform distribution, or nonuniform distribution determined by the table generator),
- the values of the window sizes (separately defined for sending and receiving directions).

Optionally, special functions used for measuring the subnetwork parameters can be pointed in commands:
- marking every outgoing packet with its departure time,
- automatic close-looping for every packet sent in the measured connection, for measuring round-trip delay.

The functions of the SYZYF components presented in Fig. 2 are aptly described by their names; only in the case of the Data Registration Module need further details be given. This module was included in the system to permit the registration of data important for measurement in DTE, such as round-trip delay and the standard file transfer time. The data structures on which this module operates are used to register the number of determined events, the sum of values characterizing these events and the histogram containing the distribution of required values.

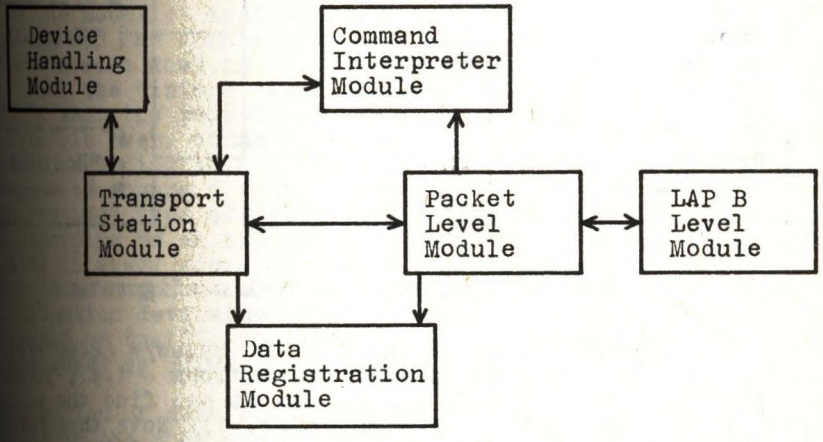


Fig. 2. The SYZYF system structure.

To obtain the characteristic of the round-trip delay, every outgoing packet is marked by current time and is sent to its destination. The destination DTE immediately sends this packet back to its source. After receiving this same packet back, the source DTE updates the number of transmitted packets, the sum of their delays and the relevant histogram item.

To obtain the characteristic of the standard file transfer time, the automatic packet confirmation option is established. During the measurement, the transfer time of files of different sizes is registered.

AN EXAMPLE

Suppose that we want to know how the node throughput varies as a function of incoming packet intensity. In our experimental configuration (Fig. 3), we have four duplex lines connected to the node. If there exist L connections on every line, the line capacity is V bytes/s, the packet length is D bytes and the interval between two succesively sent packets is T seconds, then the node load is equal to S, according to the formula:

$$S = \frac{D \times L}{T \times V} \times 100\%$$

Suppose now that we are interested in the node behaviour during "file transfer" type of traffic in the network. For this type of traffic, it can be assumed that almost all transfered packets have a constant length equal to the maximum length acceptable by the implemented protocol.

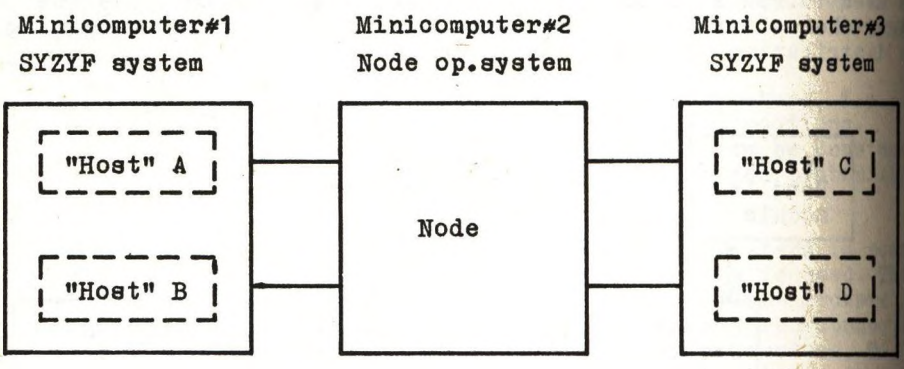


Fig. 3. Basic experimental configuration.

In our case, Dmax = 128 bytes and V = 300 bytes/s (2400 b/s). If, for instance, there are 12 two-way connections - i.e., 3 connections on each line - we have L = 3. Now we can find the suitable interval T for any given load S (see Tab. 1). Note that for calculating the offered load, the real length of the data packet is used, i.e., data field length (128 bytes) augmented by 2 bytes for A and C fields of the frame, 4 bytes packet's header, 2 bytes FCS and trailling flag.

Table 1.
Values of interval T for sampled offered load S.

S (%)	20	40	60	80	100
T (s)	6.8	3.4	2.3	1.7	1.4

The values of parameters T and D are entered into the set-up commands. As soon as a single connection is established, SYZYF begins to generate the flow of packets. Data concerning the node's behaviour are gathered in the special "registration module" attached to the basic node's software. The time of data registration is so short that it does not influence the node's operation.

Now we will turn to an examination of the application of the SYZYF system to the MSK node investigation, and a discussion of the results obtained.

MEASUREMENT OF NODE THROUGHPUT, TRANSIT DELAY AND POWER

The two main goals of our investigation of the node were (1) to determine three basic characteristics - throughput, transit delay and power as functions of the load of input node lines, and (2) to check the fairness of the node resource distribution in response to dynamic, unbalanced host demands.

All the tests were made using standard node software, with three or four hosts connected to the node by transmission lines of equal capacity. The role of the hosts was played by the dedicated system SYZYF which easily allowed us to obtain DTE - DCE line packet flow, with determined characteristic and with a mean intensity equal to λ.

The required value of λ was obtained in our measurements by a selection of length and time interval in the packet flow - appropriate to the modelled traffic, while the number of connections was constant and the window size was maximal in both directions. The traffic models used in the experiments correspond to the "file transfer" and the "interactive" types of traffic. In the former case, hosts sent the packets of the permanent maximal length equal to 128 bytes in every connection, and the interval of generation was choosen according to a uniform distribution with such characteristics as to ensure a total intensity of λ in every line. In the "interactive" type of traffic, hosts sent packets of a length choosen according to an exponential distribution with a mean value of 50 bytes in every connection. The time interval was choosen according to an exponential distribution, and the parameter of this distribution determined the global mean flow in every line. In both types of traffic the packets at their sinks are processed and acknowledged as quickly as possible.

The registration of data collected during the measurements was carried out in the node and a special data structure was implemented to store the histogram. The duration of an elementary experiment computing one characteristic was fixed at 200 s; this period guaranted that the obtained sample was representative for every value of λ. The characteristics of throughput, transit delay and power are computed from the data gathered during a single experi-

ment carried out in the configuration presented in Fig. 4. The measurements were done both for the "file transfer" type of traffic and "interactive" one.

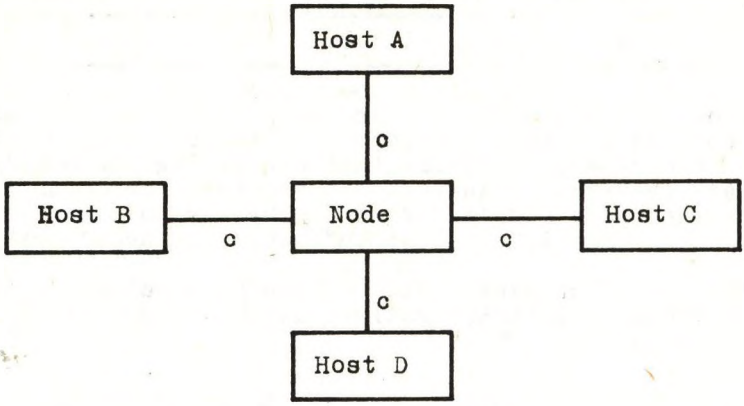


Fig. 4. The configuration for measurements of the node's throughput, transit delay and power(c=2400 b/s).

In this balanced load environment, there was traffic in both directions, so implicit acknowledgment within data packets flowing in opposite directions was possible.

Figures 5, 6 and 7 show the results. The form of the graphs in Figures 5 and 6 coincides with our expectations: the functions are monotonically rising in the entire range of λ . As one can see from Fig. 6, the normalized transit delay for the "interactive" type of traffic is greater than the corresponding curves for the "file transfer" type. This requires a little explanation. To normalize transit delay for "interactive" traffic, we used a value of time equal to the transmission time of a packet of mean length. In this way the normalized transit delay for packets of greater or less length than the mean is underestimated or overestimated, respectively. But transit delay is also a sum of other factors, e.g., processing time and input/output queueing delay. For all data packets, short or long, the processing time is almost equal and this factor adds significantly to the transit delay of the shorter packets. Due to the nature of interactive traffic, there are more packets per time unit than in file transfer traffic, and they are dispersed more irregulary. For this reason, a comparative study of the transit delay in absolute values for both types of traffic is shown in Fig. 7.

The power characteristic (Fig. 8) is the function having maximum value for both traffic types. The existence of the maximum in this case for $\lambda_{op} = 0.8$ suggests that there is an optimal range of node operation for $\lambda \leqslant \lambda_{op}$.

For the fixed intensity λ, node power can vary under influence of other factors. The influence of two factors was investigated: (1) data packet length and (2) windows size on the packet protocol level. Figure 9 shows a plot of power against packet length, which illustrates the unfavourable downward power tendency when the

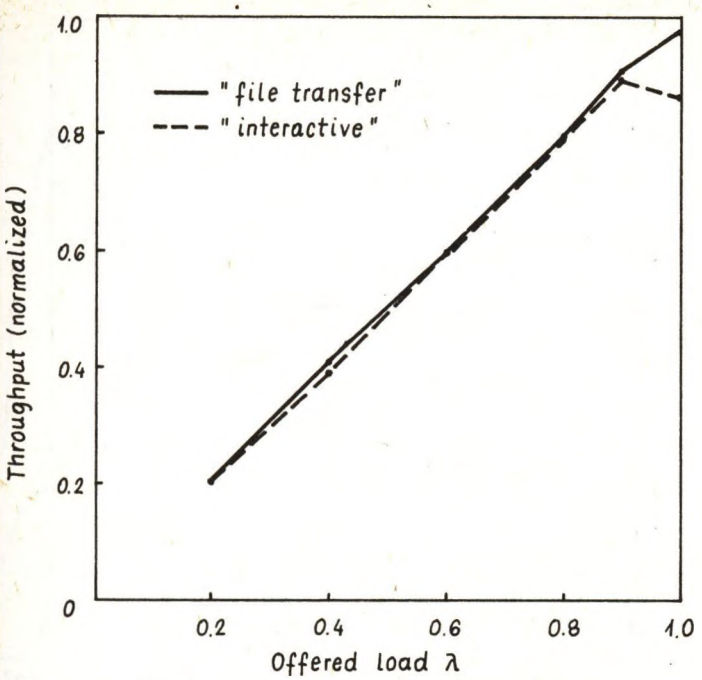


Fig. 5. Effective throughput versus λ.

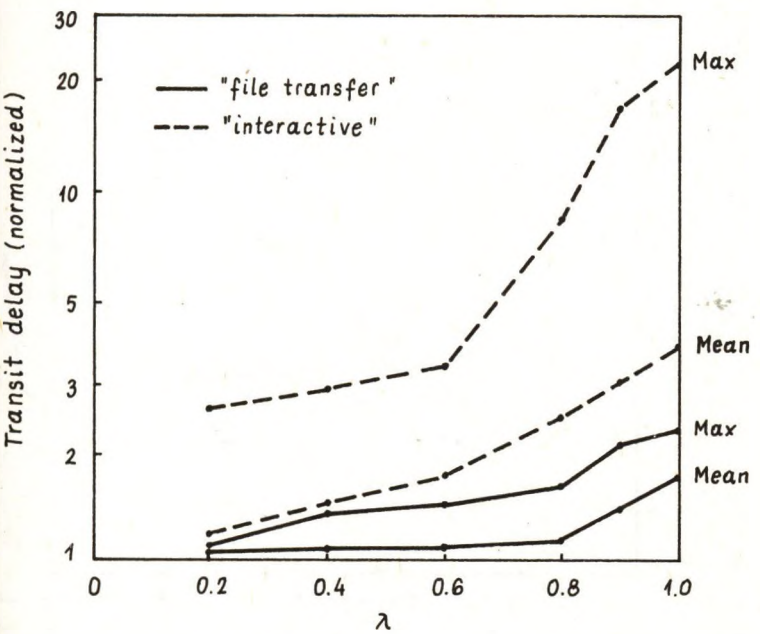


Fig. 6. Packet transit delay versus λ.

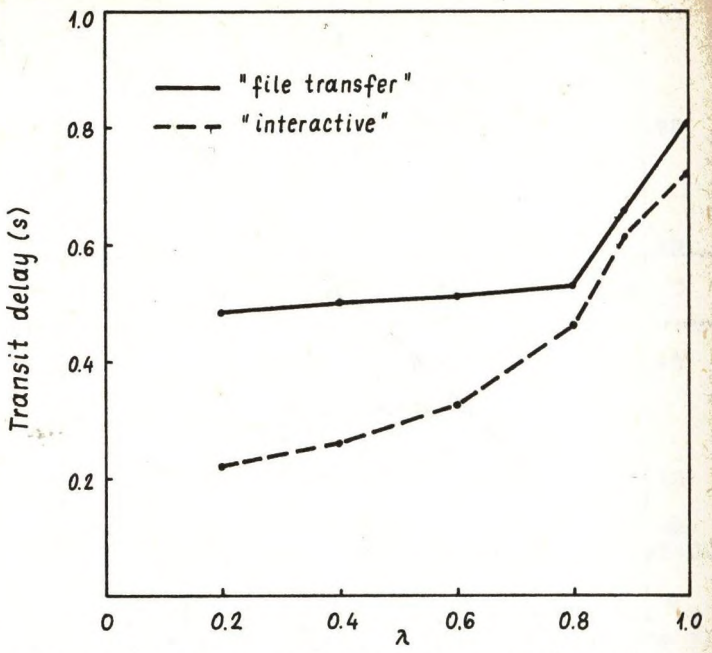


Fig. 7. Packet transit delay vs λ (absolute values).

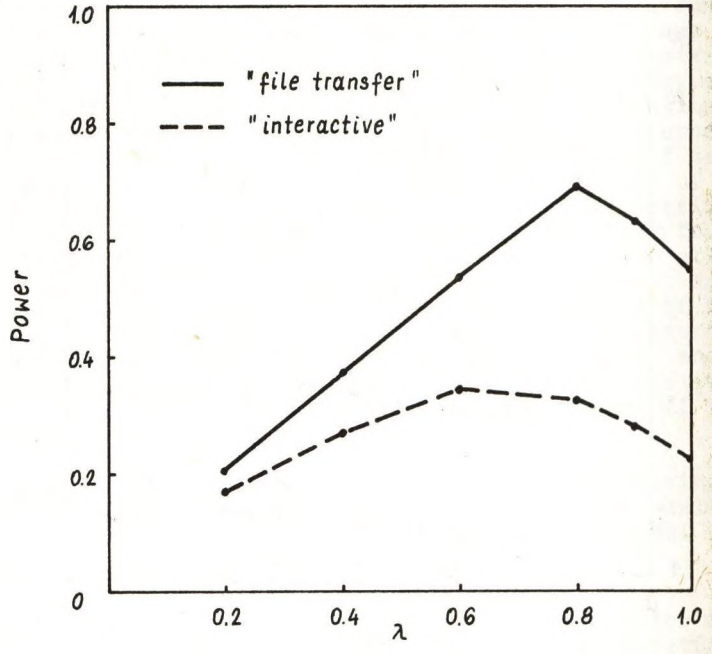


Fig. 8. Node power versus λ .

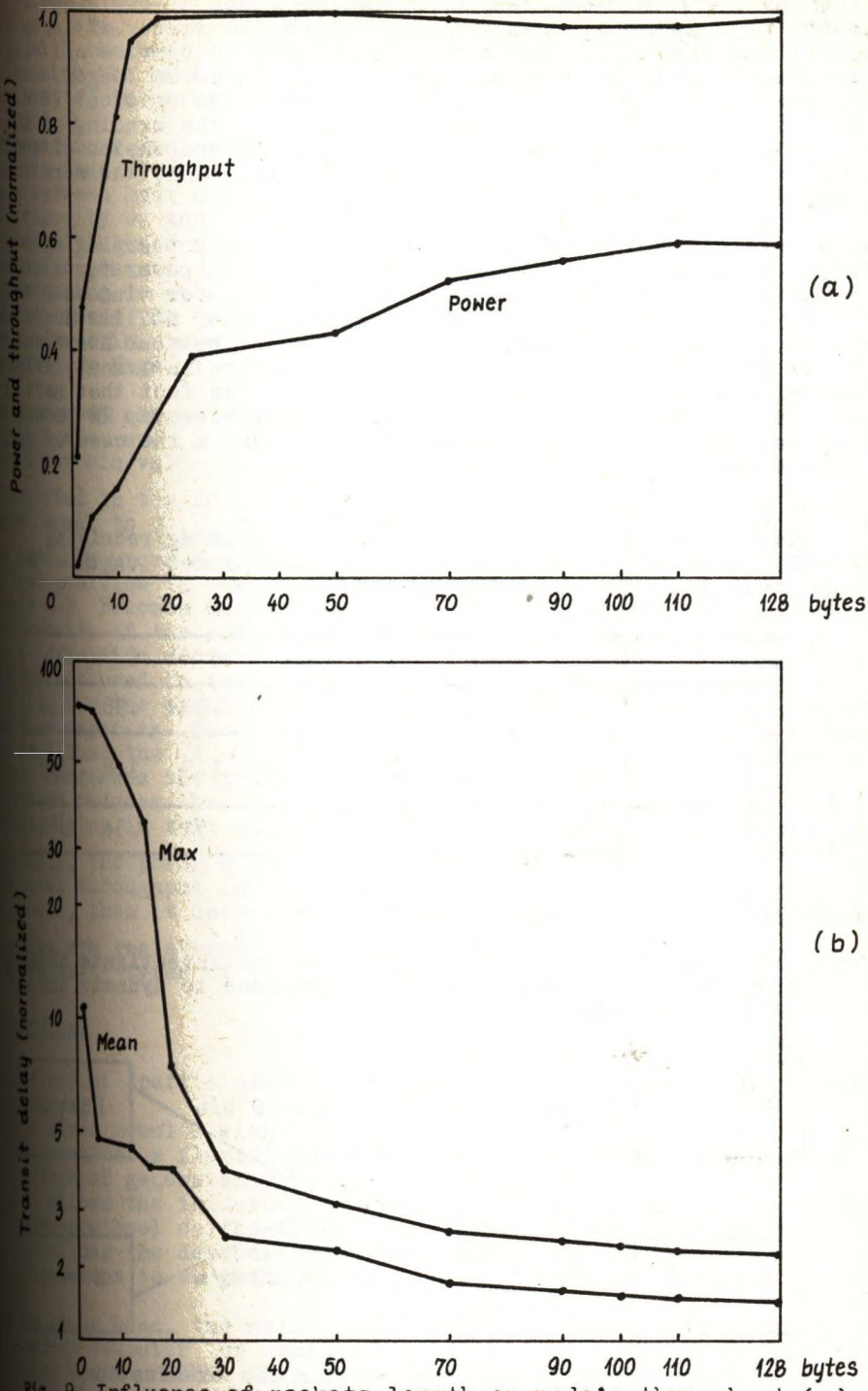


Fig. 9. Influence of packets length on node's throughput (a), transit delay (b) and power (a) for "file transfer" traffic (sending/receiving windows = 7, $\lambda = 1$).

length of packets decreases. This tendency can be explained by the insufficient speed of node software. When the number of incoming packets increases, the software is too slow to service all with acceptable delay, so that the packet queues become longer, and the node resources buffers diminish and reach the critical level. The built-in protective mechanism stimulates the sending of RNR packets and frames, which causes a temporary suspension of sending to the node. The shorter the incoming packets are, the more frequently this mechanism is put to work.

Table 2 lists the values of power obtained under varying window size. This table shows about a 25% decrease in power for a window size of 1 or 2 compared to the power achieved for windows of 6 or 7. The main reason for this inclination is poor utilization of the transmission lines throughput in connections between hosts and node for the smaller values of window size.A change in transit delay was also registered, which can be explained by the fact that part of the node computational power was used to service the RR packets. These packets are generated too often by DTE in the case of small window size.

Table 2.
The change of power as a function of sending/receiving window size ("file transfer" traffic, packet length = 128 bytes, $\lambda = 1$).

Window	1	2	3	4	5	6	7
Throughput	.87	.85	.97	.97	.98	.98	.98
Transition delay	2.2	2.1	2.1	2.0	1.9	1.8	1.8
Power	.40	.40	.46	.49	.52	.54	.54

INVESTIGATION OF FAIRNESS

Figure 10 presents the configuration used to investigate changing node throughput and transit delay in response to dynamic and unbalanced user demand.

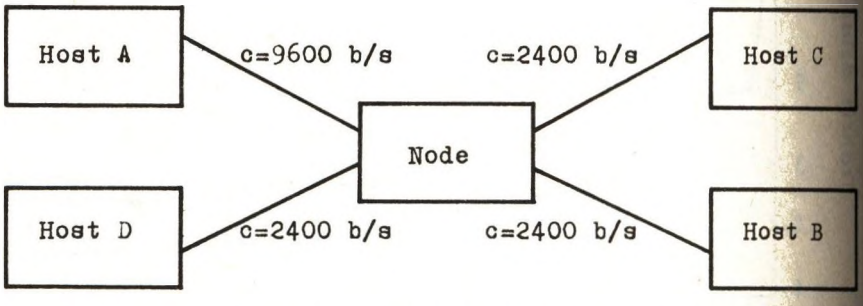


Fig. 10. The configuration for the examination of the node's resource distribution.

pairs of hosts i.e., A with B and C with D communicate through a single node. The capacities of the lines linking these hosts with the node are not equal; specifically, host A is connected with the node by a line whose capacity is four times greater than that of the other lines.

The experiment was carried out in the following way. In the beginning, the node attends only the two-way transmission which takes place between host C and host D. This transmission occupies the full capacity of the lines linking these hosts with the node. Next, host A initializes a one-way transmission to host B. The intensity of traffic generated by host A increases gradually and takes values corresponding to $\lambda = 0.5, 1, 1.5, \ldots, 4$. Under such controlled traffic, there the throughput and transit delay in A - B and C - D directions are measured. The results of the measurements are presented in Figs. 11 - 14.

The presented characteristics show that (1) the node is not "fair" to all hosts, and (2) its total throughput decreases beginning from a certain value of λ_{A-B}. The registered behaviour can be interpreted in the following way. The node handles the hosts demands equal to $\lambda_{A-B} + 1$ correctly when $\lambda_{A-B} \leqslant 1$. If $\lambda_{A-B} > 1$, the node resources are mainly allocated to host A, due to its privileged position in the system. Eventually the throughput of direction A - B becomes constant and remains on a level determined by the capacity of the node - host B line, while the throughput of C - D direction decreases to a level determined by the number of buffers not used in the A - B connection. The C - D intensity of traffic, for the examined node, becomes stablized at about 40% of the line capacity. The flow control mechanism in the node, despite the fact that the A - B throughput is kept on a maximal level, highly increases the transit delay in this direction. The same mechanism reduces the C - D throughput and decreases the mean transit delay in this direction with an increase in the λ_{A-B}; consequently, the power of this direction remains at a constant level. The total throughput in this node achieves the maximum value for $\lambda_{A-B} = 1$, then it decreases and becomes stable. All the described features are registered for both the "file transfer" and the "interactive" types of traffic.

CONCLUSIONS

The method of traffic simulation presented here has the following advantages:

a) It gives results without causing deformations,
b) It provides a flexible and simple way of controlling the parameters of generated packet traffic,
c) It allows the influence of the parameters of the packet- and line-protocol on the node and network operation to be examined,
d) It allows the dependency of node behaviour on the hardware configuration to be tested.

On the other hand, the main disadvantage is the necessity of using a separate computer or computers equipped with the special programming system SYZYF. This special computer should be more powerful than the node itself and equipped with adapters/modems to play the role of a "multi-DTE" and ensure the overloading of the node.

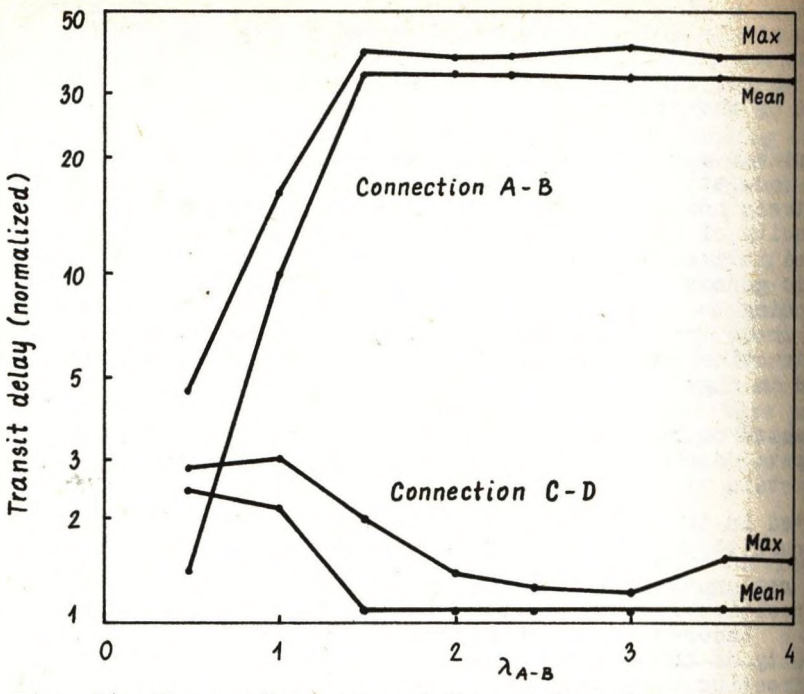


Fig. 11. The performance of the system shown in Fig. 10 - packet transit delay ("file transfer" traffic).

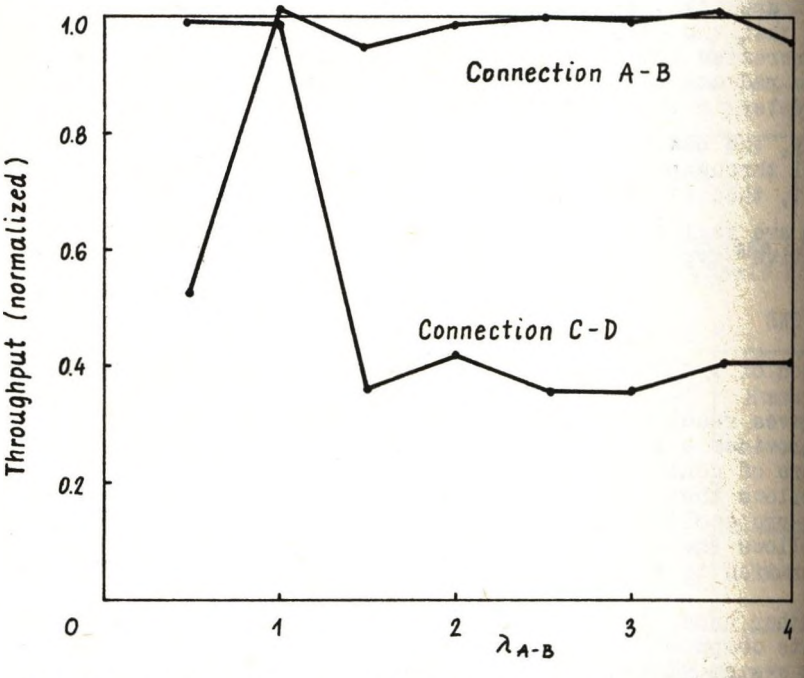


Fig. 12. The performance of the system shown in Fig. 10 - throughput ("file transfer" traffic).

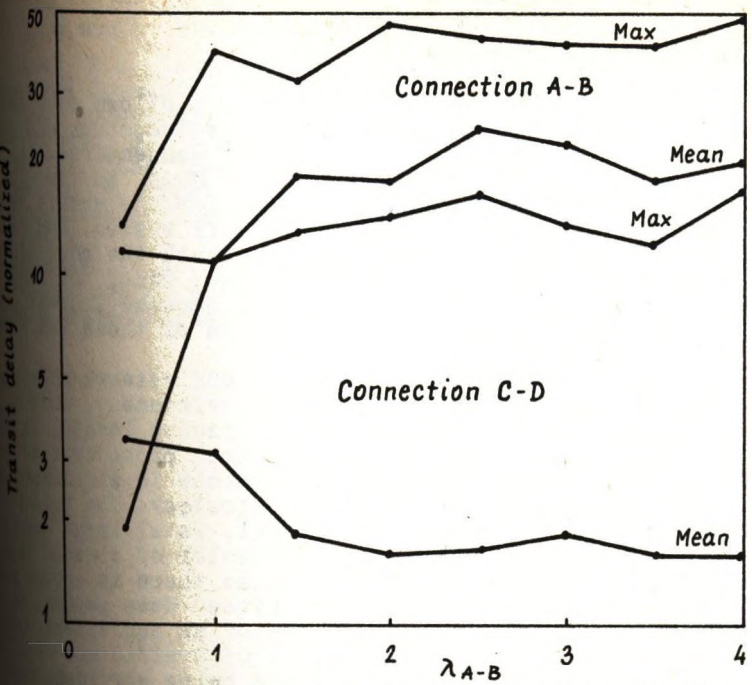


Fig. 13. The performance of the system shown in Fig. 10 - packet transit delay ("interactive" traffic).

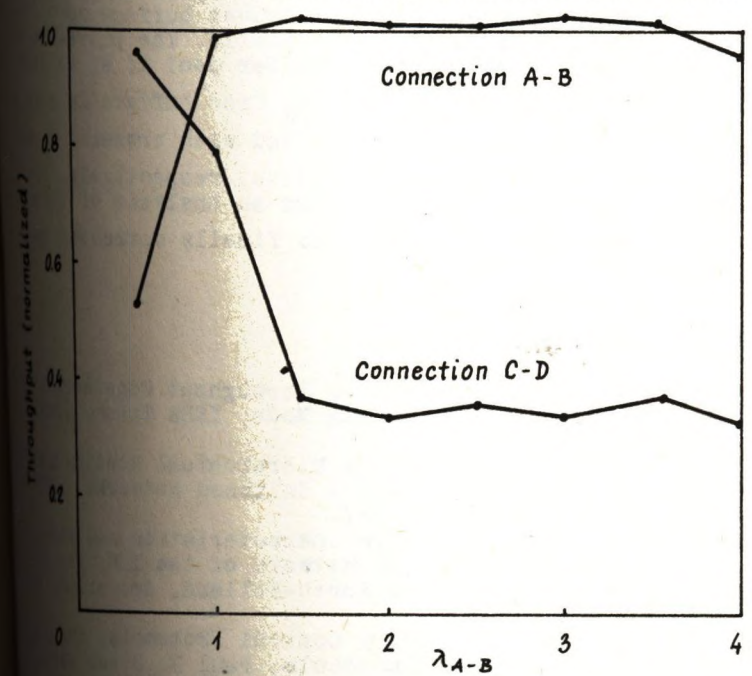


Fig. 14. The performance of the system shown in Fig. 10 - throughput ("interactive" traffic).

It is worth noting that the measuring system described above can be also used to test the operation correctness of new devices (e.g., DTE or node) attached to the network.

A special option incorporated into SYZYF also allows examination of the node's and network's behaviour in the temporary and permanent distortions appearing in the transport connections.
The performance of the MSK computer network node, in terms of throughput, packet transit delay and power for the entire range of input traffic was examined by using the SYZYF system. The study presented in this paper takes into account both the pure "file transfer" traffic and the "interactive" type. Thus, our results clearly characterize the node's behaviour, because in real situations the traffic is always some combination of these two types.

The investigations also revealed that the MSK network node's performance is quite adequate in a balanced environment, with a tendency to change power as a function of packet length. In an unbalanced environment, the node's flow control procedures do not operate acceptably, and the node's performance is significantly worse for some users, due to the buffer strategy. The buffer strategy in the examined node is two-level. Total input traffic is subdivided into classes of traffic. The number of classes is equal to the number of input lines. In each class there is enough storage to accomodate at least four full 128 bytes data packets or higher number of them, but only if they are of an equivalent summed size. The packets from different classes of traffic are serviced in round-robin fashion and while they are being serviced, they compet for a buffer from the common buffer pool, if necessary. That is, if on a certain connection there is a higher intensity of packet transfer then on others, it will request buffers more frequently, and it will get one if any are available. The procedure signals a shortage of buffers in the common buffer pool if n_L buffers are left in it. A return to the level of n_H free buffers is also signalled, where $n_H > n_L$. Signals connected with crossing n_L and n_H limits are used to send RNR and RR frames, respectively. The limits n_L and n_H were established after an analysis of this two-level buffer strategy as a whole, and finally corrected by a "tuning" process.

REFERENCES

[1] Bul, W., Kühn, P., Kümmerle, L., Throughput Considerations in Multi Processor Packet Switching Node, IEEE Trans. Comm. 4 1979 745-750.

[2] Chu, W.W., Yin Shungshen, M., A Hierarchical Routing and Flow Control Policy HRFC for Packet Switched Networks, IEEE Trans. Comput.11 1980 971-977.

[3] Dubielewicz, I., et al., Users characteristics and measurements in the communication subnetwork of the ICN, in: Networks from the users point of view North-Holland, Amsterdam, 1981 425-438.

[4] Gerla, M., Kleinrock, L., Flow Control Protocols, in: Computer Network Architectures and Protocols, Paul E. Green editor, Plenum Press, New York and London, 1982 361-412.

[5] Giesslar, A., et al., Fres buffer allocation - An investigation by simulation, Comput. Networks, vol. 2, 1978 191-208.

[6] Grange, J,L., Flow Control in Computer Network, IEEE Trans. Comm. 4 1979 .
[7] Leiner, B.M., A Simple Model for Computation of Packet Radio Network Communication Performance, IEEE Trans. Comm. 12, 1980 2020-2023.
[8] Matsumoto, J., A New Approach to Performance Oriented Flow Control, IEEE Trans. Comm. 4 1981 .
[9] Pawlita, P., Traffic Measurements in Data Networks. Recent Measurements, Results and Some Implications, IEEE Trans. Comm. 4 1981 .
[10] Weinstein, C.J., Malpass, M.L., Fisher, M.J., Data Traffic Performance of an Integrated Circuit and Packet - Switched Multiplexed Structure, IEEE Trans. Comm. 6 1980 873-877.

ON MEASUREMENTS OF THE PACKET SWITCHING NETWORK - METHODS AND EXPERIMENTS

Ján Pekár

Institute of Applied Cybernetics
Hanulova 5A, 844 16 Bratislava
Czechoslovakia

The extensive measurement system has been created in the Experimental Computer Network at Institute of Applied Cybernetics in Bratislava, Czechoslovakia. This work presents the measurement system /measured variables, measurement tools and measurement center/ and first measurement experiments that were conducted to evaluate various aspects of the network design and to determine some limiting values.

1. INTRODUCTION

The Experimental Computer Network /ECN/ interconnects HOST computer systems at research centers and universities in Bratislava and Prague, Czechoslovakia /Figure 1/. Fundamental reasons for creating this network were:

- to provide users'access to the network resources
- to create experimental environment for testing new protocol implementations and services.

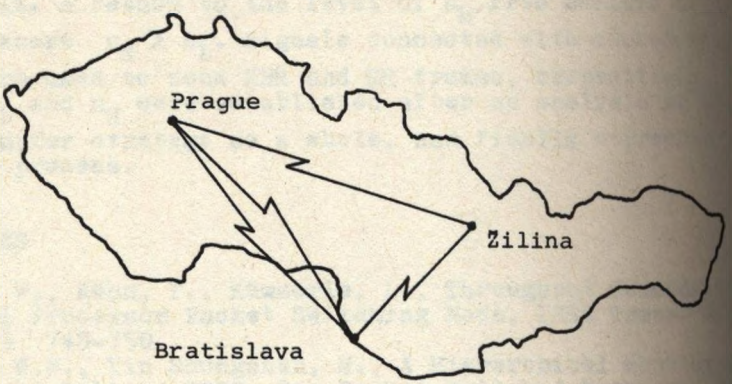


Figure 1

The machine-independent ECN has been in operation since 1983 and number of computer systems linked to the ECN is still growing. The communications subsystem is homogeneous, with nodes built on 16-bit SM minicomputers with only inevitable peripheral equipment running the special - purpose operating system called node program. It is a packet switching network that provides its users with a datagram service. The network design is based on a layered architecture and ISO recommendations.

The extensive measurement system has been created in the ECN, it has been expected to solve followwing problems:

i/ getting the network into the operation - testing its functions, identification of faults and errors in network software and hardware design, verifying of network operation

ii/ network operation - network functional ability and evaluation of the network operations and characteristics, status of network and its elements /HOSTs, nodes, front-ends, etc./ and subsystems, identification of faults in operation, diagnostics, detection of bottlenecks, prevention of breakdowns, network operation control, data for charge calculations, network performance monitoring

iii/ basis for network optimization - basis for network performance growth, evaluation of data flow, network reliability and efficiency growth

iv/ support for network modelling - verifying of network models and collecting data for network modelling

v/ verifying of network protocols and elements implementations - verifying newly introduced protocols and elements, determining their basic characteristics [1], [2], [3].

1. THE MEASUREMENT SYSTEM

The measurement system implemented in the ECN communications subsystem is made up of the following main components:

1.1 The set of measured variables

Measured variables can be divided into categories as showed in Figure 2. The variables are described by formal notation:

a/b/c/d

Letters a, b, c, d characterize four properties of variables:

a - source of variable /where it is to be measured/
- transmission line /TL/
- node computer /N/

b - type of variable /how to measure it/
- time /T/
- event count /EC/
- length /L/
- status /S/

c - form of variable /how to interpret it/
- status flag /F/
- accumulative counter /AC/
- histogram /H/
- immediate value /IV/

d - manner of collecting the variable /when to measure it/
- time interval /TI/
- event driven /ED/
- breakdown /B/
- demand /D/

Table 1 shows the list of measured basic variables with formal notation [4], [5].

The performance variables are derived and computed from the basic variables. Their definitions are various and can be found in [6], [7] and others.

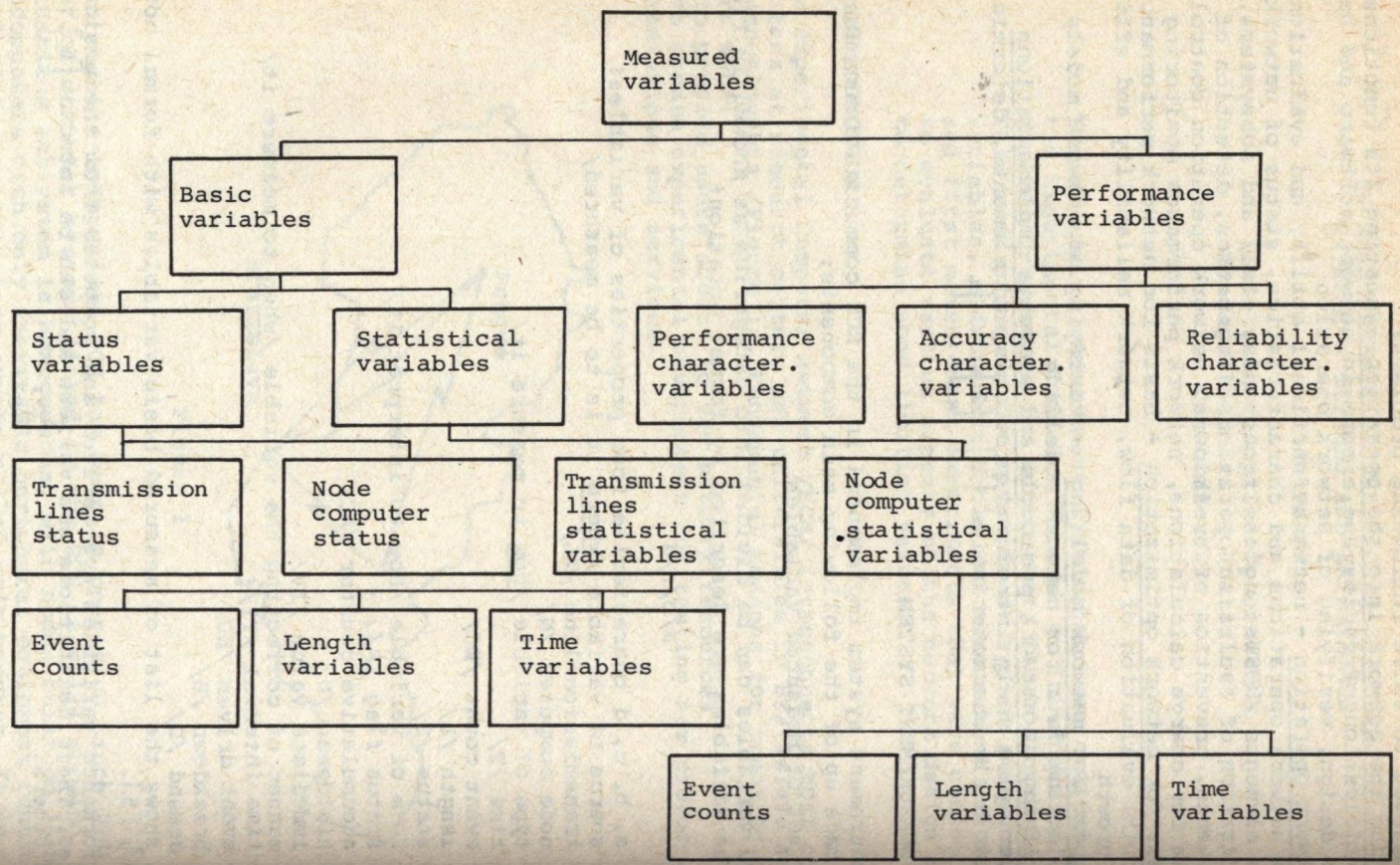

Measured variables
Basic variables
Performance variables
Status variables
Statistical variables
Performance character. variables
Accuracy character. variables
Reliability character. variables
Transmission lines status
Node computer status
Transmission lines statistical variables
Node computer statistical variables
Event counts
Length variables
Time variables
Event counts
Length variables
Time variables

Figure 2

Table 1

MEASURED VARIABLES	FORMAL NOTATION
Transmission line status:	
line on	TL/S/F/TI,B
line off	TL/S/F/TI,B
line up	TL/S/F/TI,ED
line with increased fault rate	TL/S/F/TI,ED
line down	TL/S/F/TI,B
Node computer status:	
up	N/S/F/TI ,ED
down	N/S/F/TI ,ED
Statistical:	
- lines	
- time	
connection set-up delay	TL/T/H/TI,D
disconnect delay	TL/T/H/TI,D
block transmission time	TL/T/H/TI,D
time interval between arrivals of frames	TL/T/H/TI,D
time interval between departures of frames	TL/T/H/TI,D
- event counts	
number of transmitted frames	TL/EC/AC/TI,D
number of received frames	TL/EC/AC/TI,D
number of retransmitted frames	TL/EC/AC/TI,D
number of transmitted bytes	TL/EC/AC/TI,D
number of received bytes	TL/EC/AC/TI,D
number of frames discarded for following reasons:	
- CRS error	TL/EC/AC/TI,D
- incorrect length	TL/EC/AC/TI,D
- incorrect sequence number	TL/EC/AC/TI,D
- overrun by receiving	TL/EC/AC/TI,D
- latency by transmitting	TL/EC/AC/TI,D
- lack of memory buffers	TL/EC/AC/TI,D
- link failures	TL/EC/AC/TI,D
number of transmitted control frames	TL/EC/AC/TI,D
number of received control frames	TL/EC/AC/TI,D
number of retransmitted control frames	TL/EC/AC/TI,D
number of transmitted line test signals	TL/EC/AC/TI,D
number of replies to line test signals	TL/EC/AC/TI,D
number of line test signals without reply	TL/EC/AC/TI,D
number of unsuccesful set-up attempts	TL/EC/AC/TI,D
number of unsuccesful disconnect attempts	TL/EC/AC/TI,D
number of acknowledged frames	TL/EC/IV,H/TI,D
number of frames without acknowledgement	TL/EC/IV,H/TI,D
- length	
length of received frame	TL/L/H/TI,D
length of transmitted frame	TL/L/H/TI,D
length of input queue	TL/L/IV,H/TI,ED
length of output queue	TL/L/IV,H/TI,ED
- node computers	
- time	

Table 1 cont.

round trip delay	N/T/IV,H/TI,D
packet delay in node	N/T/IV,H/TI,D
- event counts	
number of received packets	N/EC/AC/TI,D
number of received control packets	N/EC/AC/TI,D
number of packets discarded for following reasons:	
- nodal lifetime limit overrun	N/EC/AC/TI,D
- network lifetime limit overrun	N/EC/AC/TI,D
- incorrect packet header	N/EC/AC/TI,D
- inaccessible destination node	N/EC/AC/TI,D
number of transmitted packets	N/EC/AC/TI,D
number of transmitted control packets	N/EC/AC/TI,D
- length	
free buffer queue length	N/L/IV,H/TI,D
nodal software module queues length	N/L/IV,H/TI,D

2.2 The measurement tools

The measurement system user can make use of several routines - measurement tools, which can be selectively enabled at appropriate node computers:

i/ Echoer: packets are echoed to its source node

ii/ Statistics: this routine provides a summary report of the activity at each node. Each such report is enabled periodically, status reports with 3-minutes period and statistics reports with 30-minutes period. The data are sent to the Network Measurement Center /NMC/ as a two-packet message containing a subset of measured variables.

iii/ Artificial Traffic Generator: a powerful and necessary tool. Each node can generate artificial traffic consisting of either fixed or random length packets that are periodically sent on either single line or on up to eight lines at either selected or random time intervals. This routine is activated from the NMC by an activation packet containing information about generation start time, duration, frequence, packet length and destination addresses.

iv/ Tracer: this facility allows operator at the NMC to literally follow a packet through the network and to learn its route. Having identified such a packet each node generates a report to the NMC.

v/ Route Predeterminator: it allows to predetermine the route of a packet. Its information field contains the list of nodes representing the packet route. The other optional information in this packet includes the time data. If necessary, the packet is time stamped at each transiting node.

vi/ Diagnostics: if enabled, it makes clear why the packets were discarded.

Some other tools are to be implemented:

i/ Timestamping: the timestamp packet picks up clock values at critical points of nodal software.

ii/ Snapshots: the snapshot packet contains the queue lengts and the routing table information for a particular node and a particular instant of time.

iii/ Dump: the node memory dump is sent to the NMC if required.

iv/ Parameter Change: this routine enables the operator at the NMC to implement parameter changes at the node.

2.3 The network measurement center /NMC/

The last but not least component of the measurement system is the NMC. It is connected to the network via node. Its purpose can be split into three functions [8], [9] :

i/ network surveillance: in this capacity the NMC is processing the status information received from the network and providing failure situation, location and alarms to the NMC personnel. These messages are stored to be included into the daily and monthly network operation reports.

ii/ network administration support: in this role the NMC is collecting and processing statistics data from the network and providing performance characteristics to the personnel. The information is stored to be included into the daily and monthly reports. The NMC is also processing the data for customer usage accounting and storing them to produce customer bills. When the measurement experiments are held, the NMC is a powerful tool for processing and evaluating of measured data. It also plays an important role in verifying new network products.

iii/ network control: there is no special control center in the ECN. The control functions such as distribution of new software, system parameter changes are integrated into the NMC. At the measurement experiment periods the control functions are extended. All measurement tools can be activated only from the NMC and so all experiments are controlled by it as well.

3. MEASUREMENT EXPERIMENTS

Having put the ECN into the operation a few experiments were taken on it. The measurement equipment consisted of discribed measurement system and a data analyser. Measurements were conducted by performing a series of measurement runs.

In the first experiment an attempt was made to determine how various transmission speeds and various node computer models /variable number of memory buffers and processor performance/ influence the throughput /expressed in percentage ratio: number of retransmitted packets vs. number of all transmitted packets - N_R/. Two node computers were interconnected by a local telephone line. Base band modulation modems were used, the speed of transmission /v/ was set to 1200 b/s, 2400 b/s, 4800 b/s, 7200 b/s and 9600 b/s. The line length was about four meters. Artificial traffic generator sent minimum length packets /30 bytes/ and then maximum length packets /150 bytes/ with rate set to a value ensuring that output queue was kept saturated. A is 16-bit SM minicomputer /model SM 3/20/ with 32 KByte memory and B is model SM 4/20 with

128 Kbyte memory. Figure 3 shows the case when generator in one node was active, figure 4 shows the case when generators in both nodes were active.

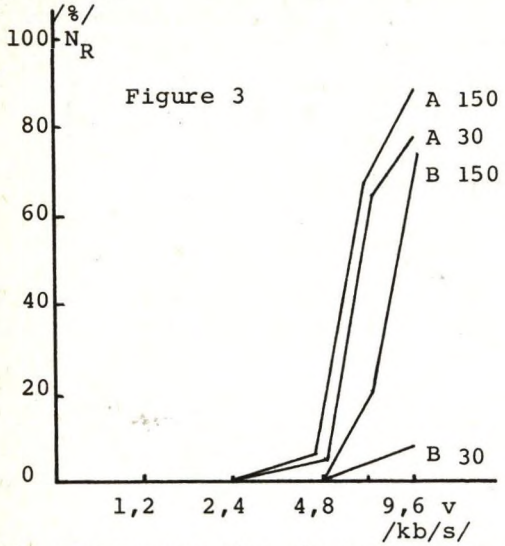


Figure 3

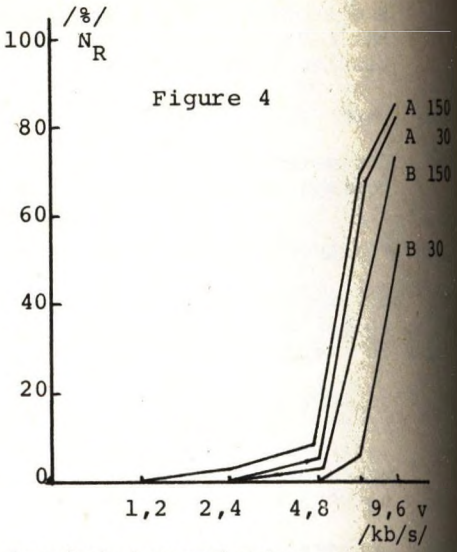


Figure 4

This experiment showed where and up to what transmission speed could be used individual computer models in node function.

The next experiment showed how the throughput of the ECM LAPB line procotol implementation depended on various transmission speeds and packet lengths. The throughput is expressed as percentage ratio of succesfully transmitted user's information /without line protocol headers, flags etc./ vs. maximum theoretical throughput of transmission line in accordance with preset transmission speed. The equipment was the same as in previous experiment with two SM 4/20 computers except the transmission line. The nodes were placed in two different sites, interconnected with leased telephone line /distance approximately 15 km/. Three curves in Figure 5 correspond to 1200 b/s, 2400 b/s and 4800 b/s transmission speeds. Experiments with higher speed did not give reasonable results.

The next experiment tried to show how real transmission lines used in the ECN influence throughput. Three nodes were placed in two different sites at Bratislava /distance approximately 15 km/, one node in Prague /distance approximately 360 km/ interconnected with leased transmission lines with capacity 4800 b/s. N_R in Figure 6 represents percentage ratio of number of retransmitted packets vs. number of all transmitted packets. Two generators generated synchronously in one measurement run.

Figure 7 shows how the amount of succesfully transmitted information /in bit/s/ depends on the packet length and transmission speed. The measurement was carrid out on a Prague - Bratislava transmission line at speed 2400 b/s and 4800 b/s. The curves show that transmitting on this line with smaller speed in case of maximum length packet is more effective then transmitting with higher speed due to increasing number of retransmissions.

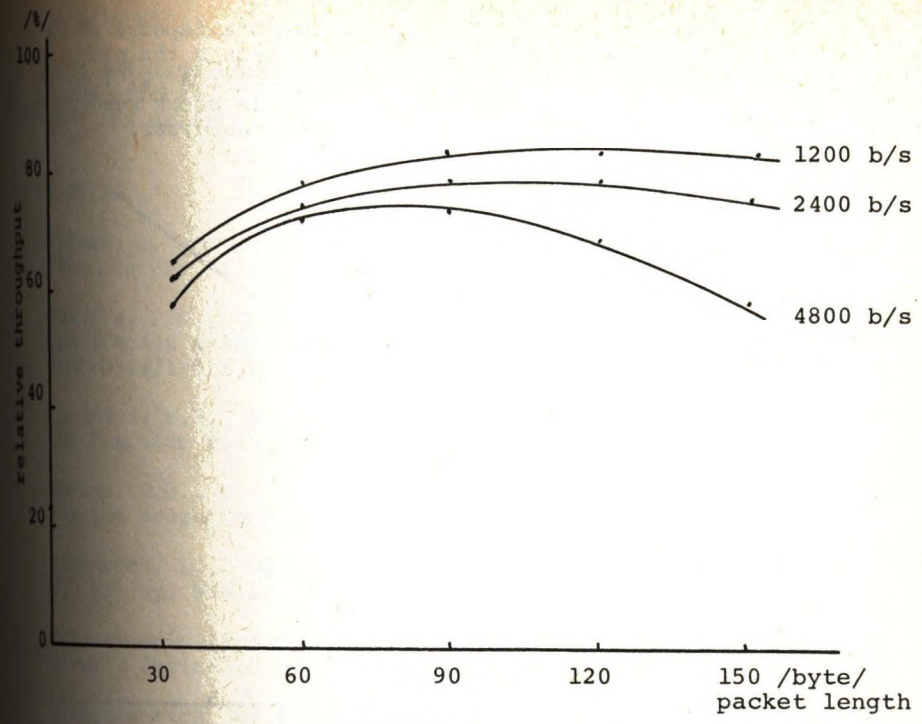


Figure 5

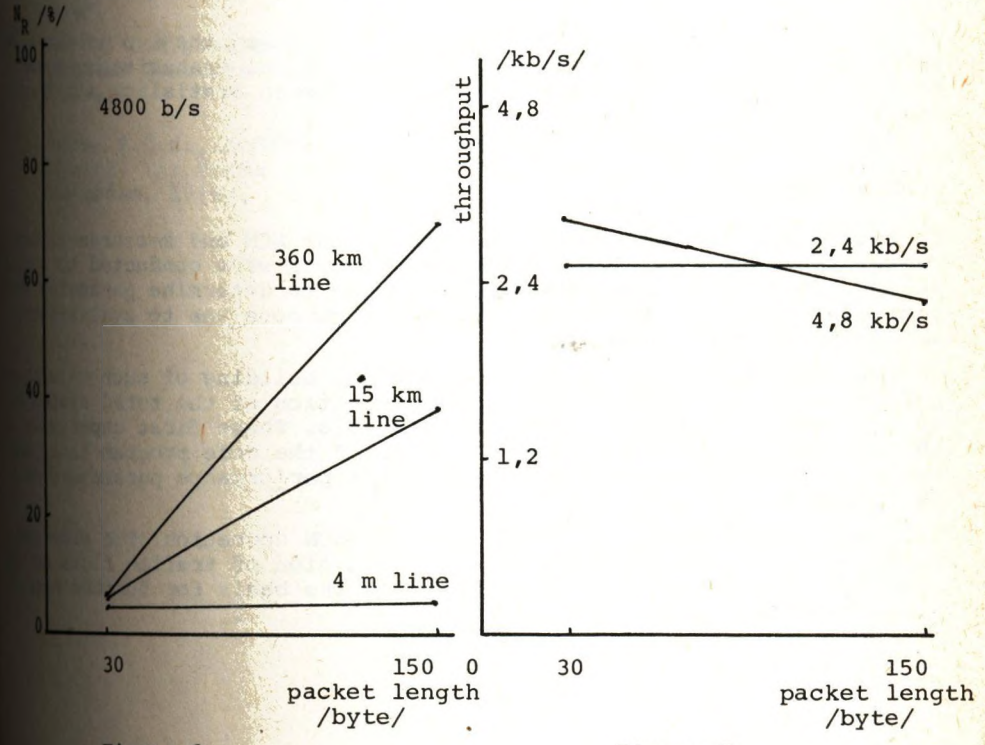


Figure 6

Figure 7

Figure 8 shows how a delay time of packet in node depends on packet lengths. Value 27 ms /packet length - Ø/ represents the packet delay time after subtracting the delay of serial reception /1,67 ms per byte at 4800 b/s/. Due to the fact that limit of node clock resolution was 20 ms, this experiment was supported with data analyser.

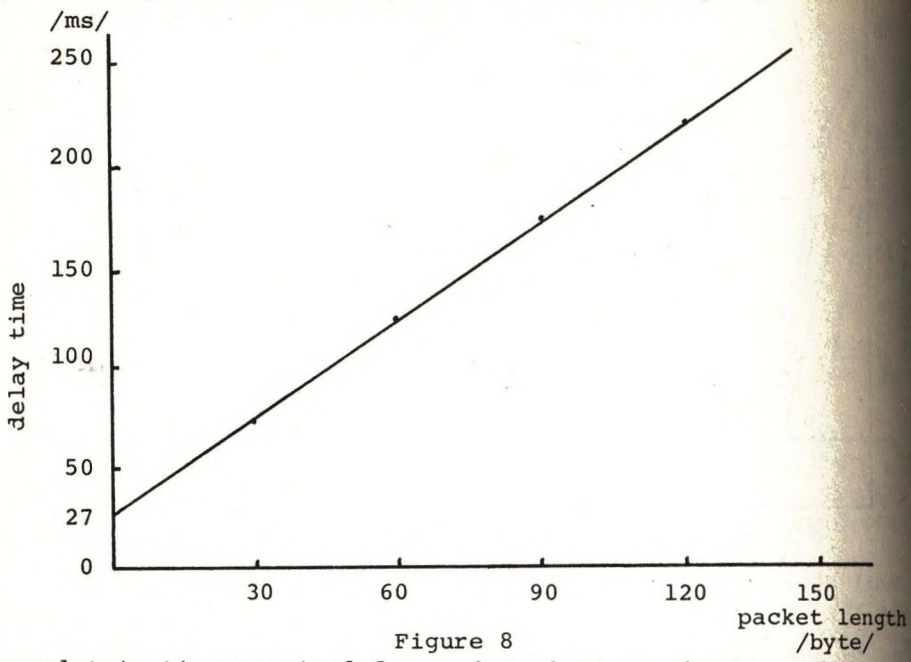


Figure 8

The round-trip time required for packets between the two nodes was measured by packets with fixed length 60 bytes and transmission speed 4800 b/s. The minimum round-trip delay between Bratislava and Prague nodes represents 0,3 s.

4. CONCLUSIONS

In this paper the measurement system of the ECN and measurement experiments have been described. These experiments were conducted to evaluate various aspects of the network design and to determine parameter sensitivities and limiting values. The other purpose was to evaluate the measurement system itself.

The experience showed that the design and building of such network required permanent observation and reevaluation of the total system design by means of measurements and analysis. These first experiments on the ECN led also to a few improvements of the node program that resulted in higher throughput and much better performance parameters /e.g. packet delay time/.

The measurement system is used in daily ECN operation. The experiments will continue with emphasis on the evaluation of traffic flows and user traffic characteristics and determining the basis for further network optimization.

5. ACKOWLEDGEMENTS

The design and development of the measurement system has been the result of the efforts of a number of the staff at IAC and all their contributions are gratefully acknowledged.

6. REFERENCES

[1] Pužman, J. and Pořízek, R., Communication Control in Computer Networks /J. Willey, Chichester, 1980/.

[2] Pawlita, P.F., Traffic measurements in data networks, recent measurement results, and some implications, IEEE Trans. on Comm. COMM-29.4/1981/ 525-535.

[3] Drummond, M.E., Evaluation and Measurement Techniques for Digital Computer Systems /Prentice Hall, Englewood Cliffs, 1973/.

[4] Abrams, M.D., Treu, S., A methodology for interactive computer service measurement, Comm. ACM. 12 /1977/ 936-944.

[5] Kleinrock, L., Queueing Systems. Vol. 2: Computer applications /J. Willey, New York, 1976/.

[6] Telecommunications: Digital Communication Performance Parameters, Interim Federal Standard 1033 /FTSC, Aug. 1979/.

[7] Determination of Performance an Information Path in a Data Communication Systems, British Standard BS 5203 /BSI, London, 1976/.

[8] Davies, D.W., Barber, D.L.A., Price, W.L. and Solomonides, C.M. Computer Networks and Their Protocols /J. Willey, Chichester, 1979/.

[9] Young, S.C.K., McGibbon, C.J., The control system of the Datapac network, in: Verma, P.K. /ed./, Proc. 3rd ICCC /North Holland, Amsterdam, 1976/.

SESSION 8
SPECIAL TECHNICAL AND THEORETICAL PROBLEMS

AN ANALYTICAL MODEL OF A LOCAL NETWORK IN DISCRETE TIME

Boyanov K.,Ruskov P., Yanev K.
Institute of Computing Technique

Dimitrov B.
Institute of Mathematics with Computing Centre
Bulgarian Academy of Science
Sofia,Bulgaria

An analytical model of a Local Area Network (LAN) with Carrier Sense Multiple Access with Collision Detection (CSMA/CD) protocol is considered. The queueing theory for single channel system with repeated requests is used. The LAN is analyzed in discrete moments of time multiple to a time interval Δ chosen before. The state of the system in any moment depends on the state in the previous moment only, i.e. a Markov chain is formed.

A system of recurrent equations for calculating the probabilities of the possible transitions in the network as a function of the input intensities, the service characteristics and the queues' discipline is derived. The corresponding expressions for calculating the system's characteristics and the obtained experimental results are presented.

I.INTRODUCTION

Multiple access procedure are widely used to build up Local Area Networks (LAN) with bus topology. For successfull application their complete study in different conditions is needed. The existing methods used directly because of the specifical feature of LAN.

There are different models of LAN, fuctioning in accordance by Carrier Sense Multiple Access with Collision Detection (CSMA/CD) procedure, described by Kleinrock, Lamm, Metcalfe, Tobagi [3,4,5,10] They are analysing different channel traffic situations, but either are not taking into account the greater request intensity in a busy channel and during a conflict, or they use complicated mathematical expressions that are hard to compute.

A model for LAN as a queueing system with repeated requests, is

presented in [8]. It makes possible the study of the networks in steady state. In the present paper we propose, for the quantitative estimation of LAN, a way for analysing in a discrete moment of time, including a dynamic mode.

II. THE DISCRETISATION METHOD

Let us examine the process $\{\xi_t,\ t \geq 0\}$ in continuous time, choose a time interval and examine the moments

$$t_n = n\Delta,\ n = 0,1,2\ldots$$

Let us define

$$\xi n = \xi t_n,\ n = 0,1,2\ldots \qquad (1)$$

This sequence (1) presents a random process in discrete time, which we shall call a discrete version of the continuous process: $\{\xi t, t \geq 0\}$ We shall use $\{\xi_n\}$ for the approximate determination of the numeric characteristics of $\{\xi t\}$.

The described system in discrete time is characterized in every random moment of time with one or several discrete bounded random variables. The set of values, which these variables may have are called states of the system. The system is characterized with a final number of states, due to the restrictions of the examined random variables. The state of the queue system in any moment depends on the state in the previous moment only, i.e. a Markov chain is formed. Knowing the initial state of the system the aim is to reach the system's state in any momen t_n.

III. DESCRIPTION OF THE SYSTEM

The queue system is described by its three basic elements:

1. Input requests flow. We assume the input flow of primary requests to be random process with Poisson distribution and intensity $\lambda \geq 0$. The arriving requests in the interval $(n-1, n)$ appear for services in the moment n. We consider the requests that have not been immediately served because of a busy channel or collision as forming a secondary flow. It is also a random hyper-poisson distribution with intensity $k\lambda_1 \geq 0$, where k is the number of secondary requests in the queue.

2. Service mechanism. We assume the consequent interval for serving the separated requests to be independent, identically distributed, random variables. Their values are in the interval $(1,2,3,\ldots M)$. M is the maximum time for service a request. Because of the discrete measurement of the time we accept, that if the service of the request begins in the n-th interval. The time needed for an attempt to occupy the channel is Δ. If during the service time of each first interval a new primary or secondary request appears, a conflict occurs and the service aborts. All requests taking part in the conflict become secondary ones.
We shall assume the channel to be absolutely reliable.

3. Queue discipline. If a request finds the channel busy it becomes secondary and the number of secondary requests k increases by one. If a secondary request is successfully served, k decreases by one. After the channel becomes idle, a random order service is carried out. We assume, that for a fixed interval Δ either single request or none can be served. If the channel is idle at the moment it is occupied by the first primary or secondary request that has arrived. If during the first interval a new primary or secondary request has arrived, a conflict occurs. The number of secondary request k increases by the number of primary ones taking part in the conflict. All primary requests become secondary ones. If during the first interval Δ no other request has arrived, the attempt to occupy the channel is successful and the latter is locked for all other requests until the end of the service. If a channel is successfully occupied from a primary request, k does not change, and if it is successfully occupied by a secondary request, k decreases by one.

IV. STATES OF THE SYSTEM

The states of the queue system at the moment tn are determined by the states of the secondary requests, the state of the request being serviced and the state of the channel. The number of secondary requests $k = 1,2,3\ldots$ at the moment tn is denoted by Xn and shall be called state of the source of the secondary requests. The state of the request being serviced shall be called number of units Δ, needed to end the requests service, and denoted by Yn. The state of the channel is denoted by ϵn, where:

$$\epsilon n = \begin{cases} 0\text{-if the channel is idle from the moment } tn\text{-}1 \text{ to the moment } tn; \\ 1\text{-if from the } tn\text{-}1 \text{ to the } tn \text{ moment first } \Delta \text{ interval is being served;} \\ 2\text{-if from the } tn\text{-}1 \text{ to } tn \text{ moment the rest } (m\text{-}1)\Delta \text{ intervals are being served;} \end{cases}$$

The triple $(Xn, Yn, \epsilon n)$ is called state of the system at the moment tn. The change of the system's state depends on the probability of conflict, of arrivals and of the adopted service.

Let us introduce the following symbols for the used variables:

KM-the maximal number of secondary requests in the queue;

M -the maximal number of interval Δ for service of a requst;

$p0$-the probability that a primary request does not arrive in a given interval Δ:

$$P_o = e^{-\lambda\Delta} ;$$

$p1$-the probability that a primary request arrives in the interval Δ:

$$P_1 = \frac{\lambda.\Delta}{1!} \cdot e^{-k\lambda_1\Delta} ;$$

$qk0$-the probability of having k requests in the secondary requests source, and without any of them arriving at the interval Δ:

$qk1$ - - one of them arriving at the interval Δ:

$$q_{ko} = e^{-k\lambda_1\Delta} ;$$

$$q_{k1} = \binom{k}{1} e^{(k-1)\lambda_1\Delta}(1-e^{-\lambda_1\Delta}) ;$$

bm-the probability of needing m intervals necessary for service of a given request:

$$b_m = P\{\eta = m\Delta\} = B_{(m\Delta)} - B[(m-1)\Delta] ;$$

$$m = 1,2,3,\ldots,M$$

$$\eta = b1, b2, \ldots bM;$$

$pn(k, m, i)$-the probability for the system to be at the moment tn, whereas: the secondary requests source has k requests; to the end of the service m intervals Δ is needed and the channel is in

state $\epsilon n = i$; $i = 0,1,2$ i.e.

$$P_n(k,m,i) = P\{X_n = k,\ Y_n = m, \epsilon n = i\};$$

The possible transitions and their probabilities are shown on Fig. 1.

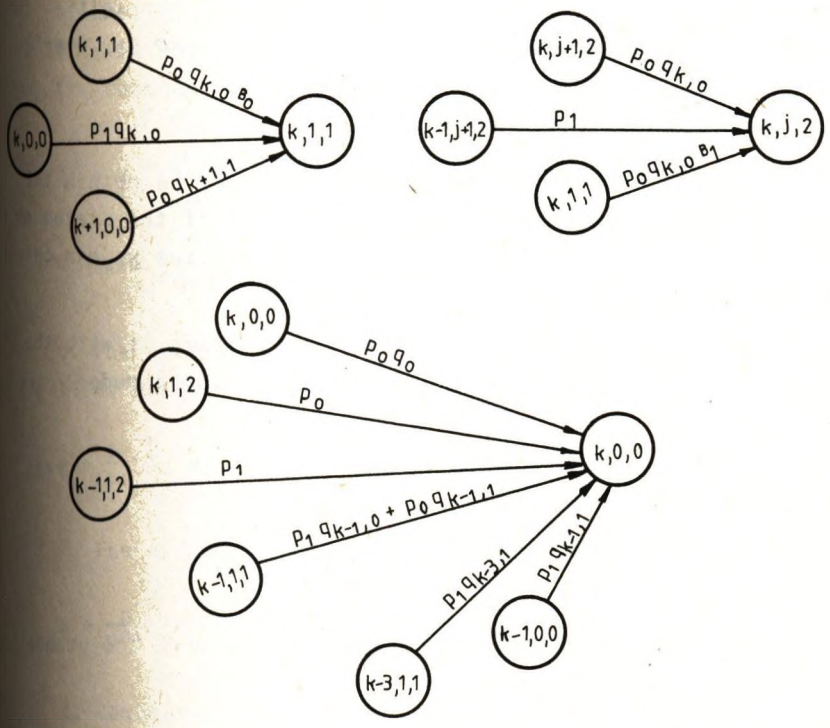


FIG. 1

The triple $(Xn, Yn, \epsilon n)$ presents the state of the system at the moment tn. From the assumption for independence of the input flow the service times and the possible transitions in the system at the two consecutive moments tn, $tn+1$, we come to (2):

$$\begin{aligned} P_{n+1}(k,0,0) &= P_0 q_{k0} P_n(k,0,0) + P_0 P_n(k,1,2) + P_1 P_n(k-1,1,2) + \\ &+ [P_1 q_{k-1,0} + P_0 q_{k-1,1}] P_n(k-1,1,1) + \\ &+ P_1 q_{k-3,1} P_n(k-3,1,1) + P_1 q_{k-1,1} P_n(k-1,0,0); \\ P_{n+1}(k,1,1) &= P_0 q_{k0} b_0 P_n(k,1,1) + P_1 q_{k0} P_n(k,0,0) + \\ &+ P_0 q_{k+1,1} P_n(k+1,0,0); \end{aligned} \qquad (2)$$

$$P_{n+1}(k,j,2) = P_o q_{ko} P_n(k,j+1,2) + P_1 P_n(k-1,\ j+1,2) +$$

$$+ P_o q_{ko} b_1 P_n(k,1,1);$$

$n = 0,1,2,\ldots \quad k = 0,1,2,\ldots k \quad m = 0,1,2,\ldots M \quad \epsilon = 0,1,2;$

For the full description of the process we assume the initial probabilities to be $PO(0,0,0) = 1$ and $PO(k,m,i) = 0$ for $k+m+i>0$

V. CHARACTERISTICS OF THE SYSTEM

After solving the recurrent linear equations (2) we obtain the absolute probabilities for the possible states of the system at the moments t_n - $pn(k,m,i)$. Knowing these probabilities we can calculate the following characteristics of the system:

1. Channel utilization ratio ρ. We identify it with the probability for the channel to be in a transmission mode:

$$\rho_n = P\{\epsilon_n = 2\} = \sum_{k=0}^{\infty} \sum_{m=0}^{\infty} P_n(k,m,2) \qquad (3)$$

$$\rho = \lim_{n\to\infty} \rho_n = \lim_{n\to\infty} \frac{1}{n} \sum_{j=1}^{n} \rho_j ;$$

2. Idle channel ratio h. We identify it with the probability for the channel to be idle:

$$h_n = P\{\epsilon_n = 0\} = \sum_{k=0}^{\infty} \sum_{m=0}^{\infty} P_n(k,m,0)$$

$$h = \lim_{n\to\infty} h_n = \lim_{n\to\infty} \frac{1}{n} \sum_{j=1}^{n} h_j ; \qquad (4)$$

3. Channel occupation 1. We identify it with the probability that the channel is in the state "attempt to occupy channel":

$$1_n = P\{\epsilon_n = 1\} = \sum_{k=0}^{\infty} \sum_{m=0}^{\infty} P_n(k,m,i)$$

$$1 = \lim_{n\to\infty} 1_n = \lim_{n\to\infty} \frac{1}{n} \sum_{j=1}^{n} \qquad (5)$$

4. The average number $\mathcal{H}_n$ of repeated requests for the moment of time tn. It is defined by the secondary requests' source:

$$\mathcal{K}_n = P\{X_n = i\} = \sum_{m=0}^{\infty} \sum_{i=0}^{2} P_n(k,m,i) = EX_n \qquad (6)$$

$$EX_n = \sum_{k=0}^{\infty} kp\{X_n = k\} \qquad \mathcal{K}_n = \lim_{n\to\infty} \frac{1}{n} \sum_{j=1}^{n} EX_j ;$$

5. Collisions probability in the interval $\Delta\Pi$. This is the probability in which the channel has been in the state "attempt to occupy the channel" and a primary or secondary request has arrived:

$$\Pi_n = P_1 \sum_{k=0}^{\infty} q_{k1} P\{X_n = k\} + P_1 + \sum_{k=1}^{\infty} P\{X_n = k\} q_{k1}$$

$$\Pi = \lim \frac{1}{n} \sum_{j=1}^{n} \Pi_j ; \qquad (7)$$

6. The average number g of collisions in the interval Δ. This is the total number of collisions during the n-th interval Δ.

$$g = \frac{1}{n} \sum_{n=1}^{\infty} P^{(k)} ; \qquad (8)$$

7. The average number of collisions for a request z.

$$z = \frac{g}{n\Delta\lambda} = \frac{1(P_1 + q_{k1} - P_1 q_{k1})}{\lambda\Delta} ; \qquad (9)$$

8. Request delay time w. This is the time from the moment of generating the request to the initial moment of its service:

$$W = z(\Delta + \frac{\Delta e^{-\lambda_1 \Delta}}{1 - e^{-\lambda_1 \Delta}}) ; \qquad (10)$$

9. Total request delay time v in the system. This is the sum of the request delay time and the service time:

$$V = w + E_\eta = w + \sum_{m=1}^{M} mb_m \qquad (11)$$

10. The probability for noncollision service of a primary request q:

$$q = \lim_{n\to\infty} p\{\epsilon_n = 1, P_o = 1 \Lambda\ q_{ko} = 1\}$$

$$q = \lim_{n\to\infty} h_n e^{-(\lambda + k\lambda_1)\Delta} ; \qquad (12)$$

VI. RESULTS FROM THE EXPERIMENTS AND CONCLUSION

A FORTRAN programme was developed for the described model. It enable us to obtain a direct quantitative estimation of the characteristics of LAN in different working modes.

The characteristics of local area networks were computed using the 3-12 expressions. A comparison was made to measurement of LAN as given in [9]..The experiments were implemented at a transmission rate of 2.94Mbits/s. The results obtained are shown on Fig. 2,3 and Fig. 2 shows that the coefficient of channel utilization is worse when the packet is smaller. The comparison of the Fig. 2 to the results from [10].proves the overlapping to be statisfactory. The diagram on Fig. 3 helps us conclude that the number of the secondary requests grows with the increase in the intensity of the input flow.

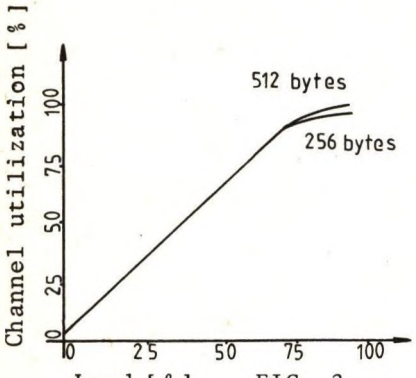


FIG. 2

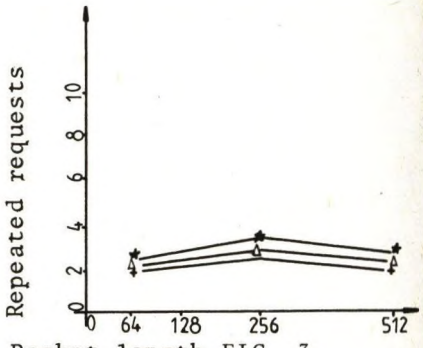


FIG. 3

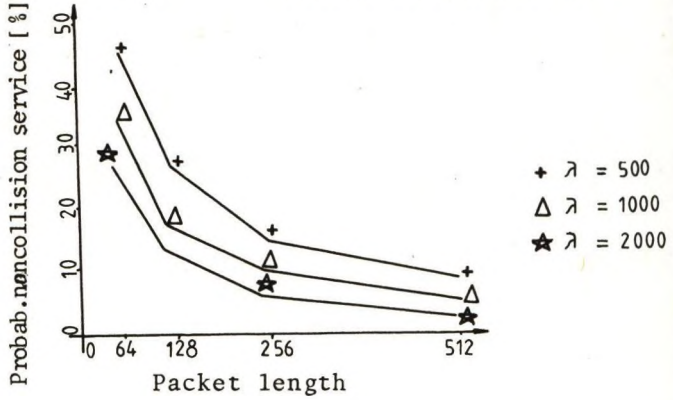


FIG. 4

Fig. 4 shows that the non conflict servicing propability decreases with the lenght of the packet at constant intensity of the input

flow.

The diagrams on Figs. 3 and 4 make it possible to do some additional estimations of the CSMA/CD procedure.

Taking into account the functioning of the procedure only at preliminary chosen moments of time, multiple to the Δ interval, the discrete time model illustrates the dynamics of the process. It permits the investigation of a variable loading of the network through different periods of the day, i.e. to vary the input parameters in the modelled process for each new interval Δ.

The discrete time model is used for more precise evaluations of LAN's performance at her border loading values. The abstract description of the functioning of the CSMA/CD protocol facilitates the expansion of the model and the investigation of different adaptive protocols.

REFRENCES:

[1]. ISO/TS 97/SC 6, IEEE Standart 802.3, Local Area Networks: CSMA/CD Access Method and Phisical Layer Specifications (July 1983).

[2]. Gelenbe E., Mitrani I.,Control Policies in CSMA LCN Ethernet Controls,Proc.of the Conference on Measurement and Modeling of Comp.System (August 30-Sept.1, 1982).

[3]. Kleinrock,L., Queueing Systems.vol.1 and 2(Wiley-Internat. Publ., 1975, 1976).

[4]. Lamm S., A Carrier Sense Multiple Access Protocol for Local Networks, Computer Networks, 1(1980) 21-32.

[5]. Metcalfe R., Boggs D., Ethernet: Distributed Packet Switching for Local Computer Networks,Comm. of the ACM,7(1976)395-404.

[6]. Neuts, M., The Single Server Queue in Discrete Time Numerical Analysis,I,II,III,IV,Neval Reaserch Logistics Quarterly, 2,3,4(1973).

[7]. Ravasio P.,Hopkins G.,Naffah N.(editors) Local Computer Networks (North-Holland, Amsterdam, 1982).

[8]. Ruskov P., Yanev K., Dimitrov B., Boyanov K., Model of Investigation of Local Computer Networks,USiM,5(1984),37-40(in Russian).

[9]. Shoch J., Hupp J., Measured Performance of an Ethernet Local Network, Comm. of the ACM, 12(1980) 711-721.

[10]. Tobagi F.,Hunt U., Performance Anasysis of Carrier Sense Multiple Access with Collision Detection, Computer Networks, 4(1980) 245-259.

OSI COMMUNICATION CONTROLLER FOR HETEROGENEOUS COMPUTER SYSTEMS

Achim Hennecke
Volker Heymer

Academy of Sciences of the GDR
Institute of Informatics and
Computing Technique
DDR-1199 Berlin, Rudower Chaussee 5

Using the OSI Reference Model for interconnecting heterogeneous computer systems a communication controller is under development to provide the functions of the four lower layers of the model. Characteristics of this solution are using the same communication controller for different computer systems, offloading the host computers from a relevant part of the communication software, reducing the overall implementation efforts, providing the possibility of testing an important part of the interconnection functions without participation of the host systems.

1. INTRODUCTION

Today packet switched data networks with X.25 interface become widespread information transfer systems serving as a basis for building up computer networks of different architecture and for different purposes. There is a growing need above all, to interconnect heterogeneous computer systems to computer networks. This ought to be done in accordance with the architectural principles defined in the OSI-Reference Model (OSI-RM)[IS7498] and standards developed for layer services and protocols.

Following the architectural principles of the OSI-RM in all end systems, full hierarchy of the seven defined communication layers has to be provided. The implementation of these layers is not restricted by the OSI-RM and can be different from one end system to the other. It depends, for instance, on the capacity parameters (storage, speed) of the computer systems to be connected and on the existing implementation environment in the form of provided basic software tools as the operating system, programming languages etc.

Starting from our experience in the development and exploitation of the computer network DELTA [CAR81] a communication controller is under development that provides the functions of the four lower layers of the OSI-RM.

2. FUNCTIONALITY OF THE COMMUNICATION CONTROLLER

Efficient implementation and multivalent usage of the outcoming systems is important for the transformation of the architectural model into hardware components. A hard- and software solution meeting these requirements consists in the implementation of the four lower layers of the OSI-RM, where the transmission-oriented services are located, on a special communication controller (CC). This solution has the following characteristics:

- The implemented services are provided by application-invariant protocols and are based on the chosen transmission media, i.e. they are independent of the type of application. This implies that the CCs, from the point of view of their functions, can be used for different computer systems.
- The general-purpose computers to be connected to the data network are offloaded from a relevant part of the communication software.
- The overall implementation efforts are significantly reduced because the CC needs to be developed for different computers only once.
- The test of an important part of the end system software is possible without participating of the hosts and the terminals.
- The application of a specialized communication-oriented hardware architecture is more efficient than that of general-purpose systems.

Figure 1 shows the distribution of the layers of the OSI-RM between the CC and the hosts/terminals.

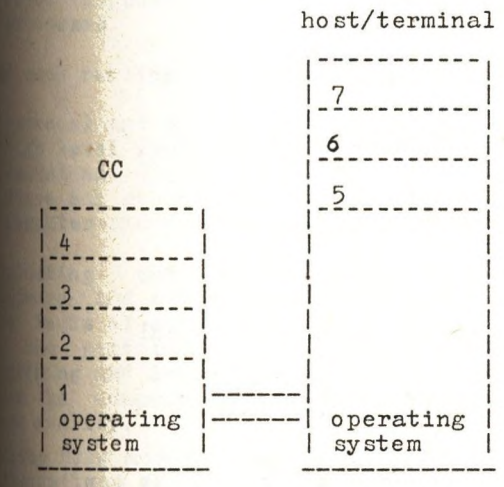


Fig. 1. Distribution of the OSI-RM-layers beetween CC and host/terminal

The CC consists of two logical parts. One provides the communication functions in the sense of the OSI-RM and the other connects the CC to conrete hosts or terminals. The first part is responsible for connecting the CC to the data network and for providing the network and transport service. The CC is connected to the data subnetwork by a X.25-DTE, thus creating the possibility to use the

CC on public data networks which are working on the basis of the CCITT-Recommendation X.25.

The network service corresponds to the ISO-proposal [IS8348], so do the transport service [IS8072] and transport protocol [IS8073]. Therefore we have connection-mode services and protocols. The first release of the transport layer software will support class 0 and class 1 transport protocol.

The connection of the CC to hosts or terminals has to provide the transparent transfer of transport-interface-data-units (TIDU) between the CC and the connected hosts or terminals. (see Fig. 2)

The functions needed to transmit the TIDUs may be different for hosts and terminals. Whereas the host is connected to the CC in a direct way on the channel-to-channel-basis, the terminals can be connected to the CC locally or remotely via lines at transmission speeds usually not higher than 2,4 kbit/s. From these different conditions follows that different control procedures are used for the data transmission between the systems (e.g. character-oriented asynchronous data transmission procedures for terminal connection). Further, in the case of terminal connection it is useful to provide the function of multiplexing several terminal connections to one data network connection.

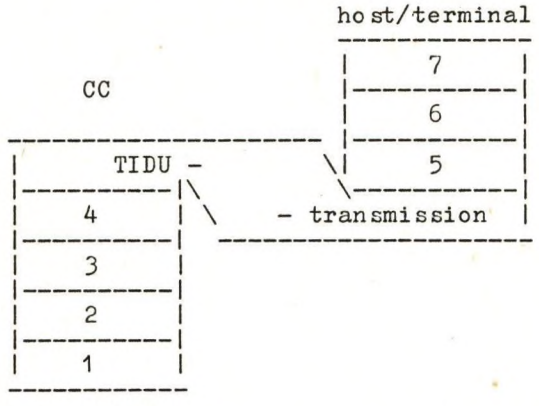


Fig. 2: TIDU-transmission between CC and host/terminal

3. IMPLEMENTATION OF THE COMMUNICATION CONTROLLER

The hardware of the CC is a 8-bit multiprocessor system with connection modules for different computer systems each. The configuration of the system has been estimated by a throughput model. To guarantee the demanded parameters the CC for hosts (HCC) is implemented on a three-processor-system, while the CC for terminals (TCC) is implemented on a two-processor system.
The OSI-RM-functions are located on two processors (see Fig. 3). The network processor (N-P) contains the three layers for the connection to the data network and the functional unit providing the network service. The transport processor (T-P) is responsible for transport functions and controls the TIDU exchange with the host or terminal. In the case of the HCC this control is done via a third processor.

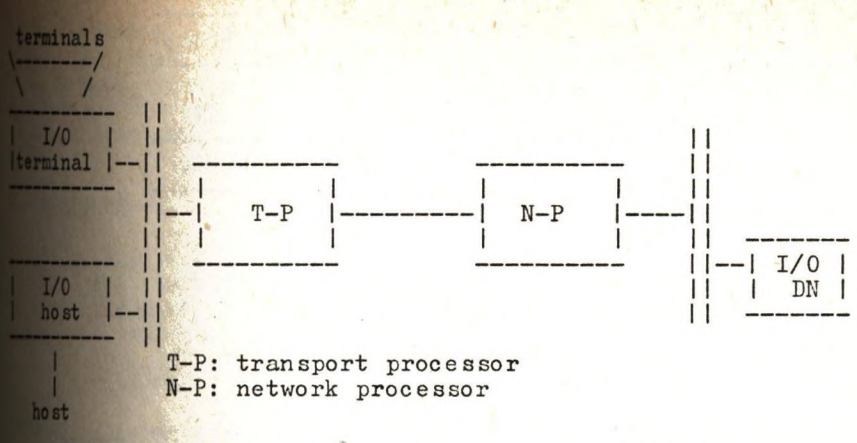


Fig. 3: The general hardware structure of the CC

An example of a computer network composed with the help of CCs is shown in Figure 4.
Our experience in implementing computer communication software led to a standard implementation environment which consists of such basic system components as

- process management including a timer mechanism
- buffer management for different types of buffers
- queue support for the exchange of events and messages between processes

and such runtime and non-runtime tools as

- protocol and service specification method
- high level programming languages (for CC-implementation PASCAL and C),
- trace and statistics features,
- computer-aided documentation system.

Evaluating our past implementations of computer communication software and estimating the efforts taken to implement the CC on the basis of a CPU-time-model, we concluded that an important part of the performance of the hardware would be needed for process switching and local information transfer inside the system. Therefore the components of these functions are to be implemented in the best quality.

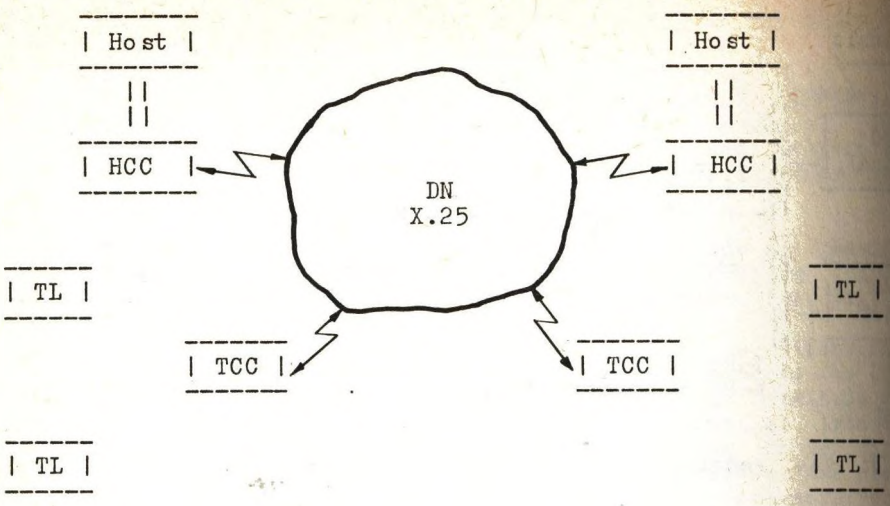


Fig. 4: Computer network with end systems connected by CCs to the data network

4. FIRST APPLICATIONS

First target computer systems connected to a X.25 data network via the CC are ES-computers and personal computers serving as hosts and terminals, respectively. The first application services the computer systems will provide are File Transfer, Remote Job Entry and Operator Correspondence as they are now supplied in the computer network DELTA. On the ES-computers this will be done by a special adapter mapping the connectionless-mode transport service of the computer network DELTA on the connection-mode transport service of the CCs. The application-oriented software remains unchanged.
On personal computers used as a new terminal technique the corresponding application-oriented components are directly based on the connection-mode transport interface.
In order to use the different application services on the terminal in parallel it was necessary to develop a special operating system which provides process control, I/O-handling in parallel to running processes, error-handling and timer organisation.
At a later state of research and development new services corresponding fully to the ISO- and CCITT-standards for the higher layers of the OSI-RM are planned. They will be made on the same transport interface provided by the CC.

5. CONCLUSIONS

The use of communication controllers is a way to reduce the overall efforts necessary for connecting computer systems to data networks. At the same time it shows in which way specialized hard- and firmware components could be developed so that part of the communicational functions is fulfilled more cheaply than it is now done by software.

After beginning to work for wide area networks, it soon became clear that with modifications in the three lower layers the CC can also be used for local area networks. This may furthermore lead to a decrease of the gateway-functions in the case of connecting wide area and local area networks.

6. REFERENCES

[CAR81] Carl, D., et al.
The Architecture of the Communication Services Hierarchy of the Computer Network DELTA and Experiences in its Implementation
Proc. COMNET'81, North Holland Publ. Comp., Amsterdam, 1981

[IS7498] Information Processing Systems - Open Systems Interconnection - Basic Reference Model
ISO/IS7498

[IS8072] Information Processing System - Open Systems Interconnection - Transport Service Definitions
ISO/DIS8072

[IS8073] Information Processing Systems - Open Systems Interconnection - Connection Oriented Transport Protocol Specification
ISO/DIS8073

[IS8348] Information Processing Systems - Data Communications - Network Service Definitions,
ISO/DP8348

CONTROL NETWORK DESIGN USING M A R K MICROPROCESSOR SYSTEM

Petr Popov - Ivo Janoušek
ČKD Semiconductors
Prague
CSSR

The contemporary production processes as rolling mills, chemical plants, compressor stations of gas pipe-lines etc. represent ample complexes of interrelated technological aggregates, the control of which is reflected in demand of autonomous control so as coordination tasks solution. The paper deals with the new modular microprocessor system MARK, product of ČKD Semiconductors Prague, which is suitable for processor nets distributed control. There are described HW and SW means of system and the communication in the nets circuits. The modular structure of MARK gives us the possibility of HW sets specification for various applications.

INTRODUCTION

The development of microprocessor technology since the end of the 1970's in the application field of computer control systems for process and technological complexes has brought about new approaches to control. The previous concept based on the application of a central control computer is being replaced by the development and applications of autonomous processors working in networks on the basis of the distribution /often hierarchically structured/ of control jobs, or the so-called distributed intelligence. The new trend is not only characterized by the production of microprocessor systems /e.g., IBM or HITACHI/ but also by the study of the questions of communication, stability, redundancy, etc. of processor networks.

The present paper describes the MARK modular microprocessor system produced by ČKD Prague, Polovodiče /Semicondustors/Plant designed for control of process and technological aggregates. Thus, attention is focused on the characteristics of the ring network processors and communication facilities.

SYSTEM CHARACTERISTICS

The electronic module of the MARK control system can be conceived as an electronic assembly with its own power supply designed into one 19" rack with a groundplain, comprising a set of electronic

and a 32 kByte RAM.

The MZP-1 unit communicates with the other units via the STD type standard bus.

The unit MKD-1 expands the data processing and control capacity. It consists of an STD bus transformer on the ČKD bus /for I/O unit connection/ and an EPROM type memory of a capacity of up to 64 kByte accessible to microcomputer MZP-1 as a virtual memory of 16 pages, each of a capacity of 4 kByte.

The unit MTV-1 co-operating with unit MZP-1 via the STD bus is used for organizing terminal stations. It allows connection of a black and white or colour RGB monitor as a semigraphic display /24 lines, 80 characters per line, Latin alphabet, Cyrillic alphabet, and 128 selectable semigraphic characters/.

The specialized microprocessor controller of the thyristor-controlled electric drive is also based on the U880 microprocessor.

The RPZ module is provided with a single-card MSP-1 microprocessor. It is a microprocessor compatible with the LSI 11/02 microcomputer. It is built using microprocessor slices of the MH3000 series. The card houses the 16-bit microprocessor proper and control of the A type bus including the control of DMA channels and a four-level interrupt system. In this concept, the microprocessor is completed with a number of RAM type memories and an RAM/EPROM combination.

The single-card logic processor /module RPL/ is a specialized logic control processor. It is a part of the logic processor family produced by ČKD Polovodiče /Semiconductors/ and is identical to the processor of the second-generation SMEP module called SM 54/50 produced by ČKD Polovodiče /Semiconductors/ for SMEP series users in the CMEA countries.

For compatibility with a number of memories and other principal interface units the logic processor cooperates with a Q-type bus. It is built on the basic of microprocessor slices of the MH3000 series.

Communication processor MKP-1 is a specialized processor of an original design built on the basic of MH3000 microprocessor sections. It cooperates with the MCP-1 or RPL-1 processor and shares with it a common working memory.

MARK COMMUNICATION NETWORKS

The mentioned complexes and modules operate in a processor network in both the horizontal and vertical connection by using the so-called ČKD Communication Ring. Connection is realised by using a coaxial cable and communication processors. Fig. 1 shows a typical example of a network /control system for transit gas pipeline compressor station/.

The periodic operation of the communication processor is controlled by a data base stored in the operating memory. The objective of the job is to organize the behaviour of the electronic module in the ČKD ring network. The communication processor control data structure can be divided into three basic parts:

- communication processor status determines the basic modes of the

units pertaining to the assembly. All electronic racks and units are designed in the 6u format of 220 mm in depth. The system consists of a total of 9 basic electronic modules classified into two groups:

1. electronic control modules:

   - RPZ: basic control module
   - RPL: logic control module
   - RMO: small object control module
   - RTP: thyristor drive control module

2. modules for communication with the controlled object, with the operator or with a co-operating SMEP family computer:

   - DIO: data input/output module
   - LIO: logic input/output module
   - TIO: terminal input/output module
   - EIO: input/output expansion module for RMO
   - KIO: module for SMEP family computer hook-up to KRUH ČKD /CKD Ring/

Apart from the EIO expansion module, all other modules are characterized by their capability of co-operation with the other partners in the ring network simultaneously. Thus, the possibility of the general-purpose configuration of hierarchical networks by the MARK control and interface modules has been obtained.

The MARK system of electronic modules is complete with a modular system of electrical equipment including cable interconnect modules, power supply and protection modules, set of system cables, and equipment allowing the connection of standard peripherals, and with a modular system of "coats", i.e., cabinets and air condition racks.

The mentioned system of facilities offers to the system designers the posibility of purposefully configuring control systems both as for their function and the optimal location and protection of the individual parts. Designers can thus form co-called universal technological assemblies, i.e., selected systems of electronic modules and electrical equipment housed in one standard cabinet. Examples of universal assemblies include, e.g.:

- KK: co-ordination assembly comprising for example RPZ+RPL or RPZ+LIO+DIO, etc.; in addition it also comprises external memory floppy disks;
- LK: local assembly, e.g., RPZ+RPL+...
- SDS: discrete signal station /LIO+ /
- SK: interface assembly /LIO+DIO+.../
- TK: terminal assembly /TIO+ /

SYSTEM MARK PROCESSORS

The MARK control system is provided with two universal microprocessors and two specialized processors.

Single-card microcomputer MZP-1 is designed for the control of small objects and of terminal systems /modules RMO, RTP, TIO/. It is based on the single-chip of eight-bit microprocessor Z80 /U880 made in GDR/ and is complete with real time clock, two ports for parallel eight-bit interface and two ports for asynchronous serial channels up to 4.8 kbit/s. The single-card microcomputer also consists of a 48 kByte memory organized as a 16 kByte EPROM

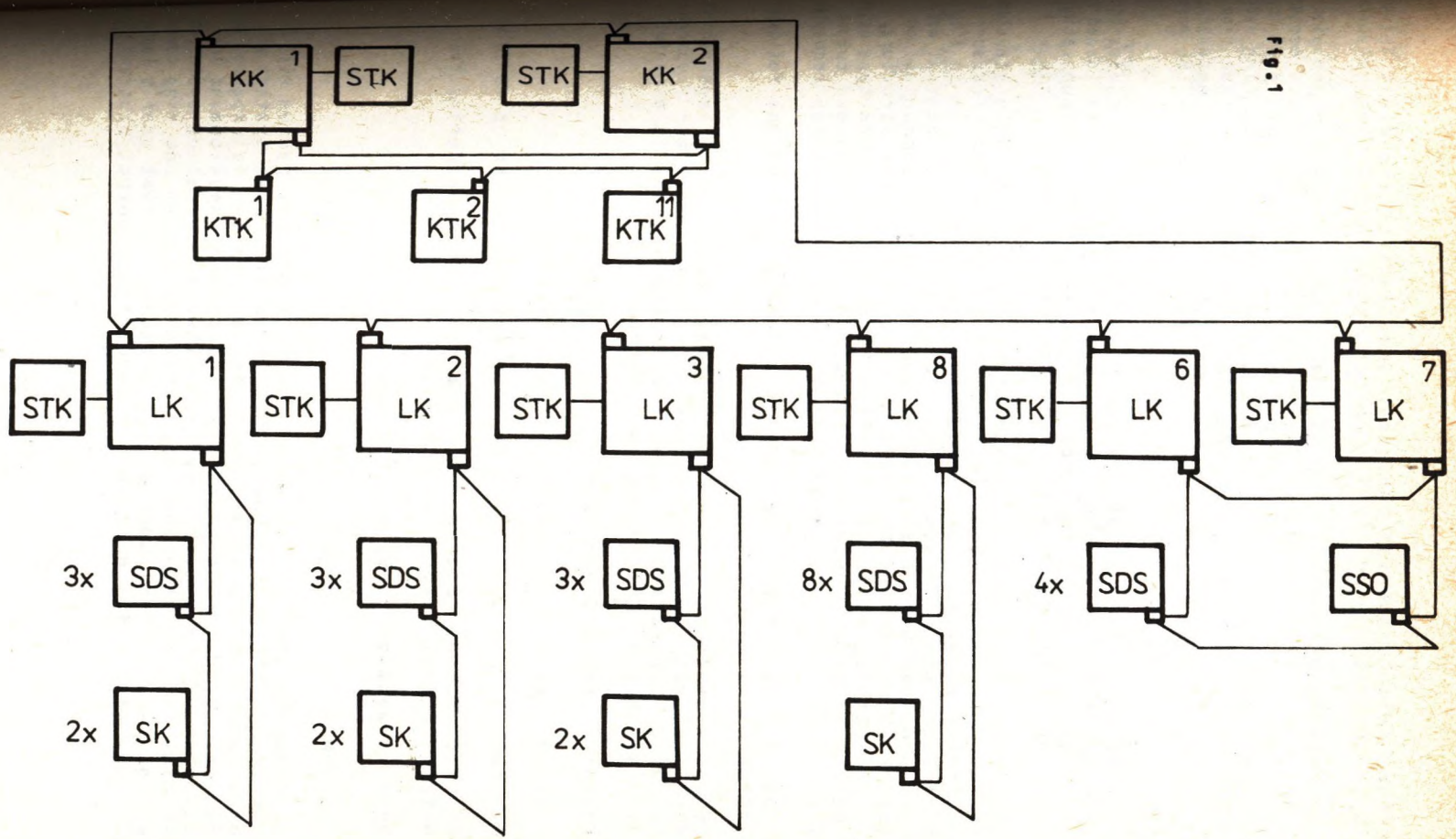

KK 1
STK
STK
KK 2
KTK 1
KTK 2
KTK 11
STK
LK 1
STK
LK 2
STK
LK 3
STK
LK 8
STK
LK 6
STK
LK 7
3x SDS
2x SK
3x SDS
2x SK
3x SDS
2x SK
8x SDS
SK
4x SDS
SSO

Fig.1

communication processor behaviour and contains the basic information on operation and possible error state in the ring network;

- ring network message catalog basically determines the behaviour of the electronic module with respect to the set of all messages which can occur in real time in the ring network. This part together with the universal communication protocol of the ring network make possible that all messages circulating and repeating in the ring can be considered to be potentially circular messages, i.e., any message can either be ignored or received by the module;

- queue of electronic module messages developed for fransmission by communication processor: This FIFO request queue is loaded by the central or logic processor and emptied by the communication processor during message transmission via the network ring.

It is thus obvious that the communication processor data structure determines the dynamic behaviour of the communication processor for periodically repeated closed-circuit message circulation in the ring network. The intergace modules and small object control modules are provided with a simplified version of the communication processor characterized by the feature that the data structure, i.e., the catalog part, is directly written in an EPROM type memory and the interface units operates with a fix programmed output message rather than with a dynamic queue of transmission requests.

The communication along the ring is characterized by the following specifications:

1. Series-type transmission, NZR-I code, formatted by expansion of HDLC protocol, message includes 6 organization bytes /separators, header, protection/; the rest of the message is a useful data message of the transparent type.

2. Transfer rate 0.5 Mbit/sec

3. Delay 5 sec/km of the cable, i.e., 1 bit/400 m

4. Maximum length of a ring 63 km, recommended length up to 20 km

5. Distance between active ring elements 2 km considering two outages /i.e., a total of 6 km/

6. One-way message flow, digital filtering

CONCLUSION

It is supposed that all parts of the MARK control system will be introduced to the market early 1987. However, first systems will be produced in this year and their implementation is expected in 1986. Associated with this is also the plan of development and compilation of the complete system software /operating system and facilities for implementing information subsystem and ring communication/. The system support also includes special user's languages, e.g., for writing logic and sequential control programs [5].

REFERENCES

[1] Brabec, J.; Janoušek, I.; Popov, P.; Řička, J.: Mikroprocesorový řídicí systém MARS /MARS Microprocessor Control System/. Automatizace /1984/ v. 27 /3/

[2] Janoušek, I.; Havelka, J.: Struktura a prostředky řídicích systémů stanic tranzitního plynovodu /Structure and Fecilities of Transit Gas Pipeline Control Systems/. Automatizace /1984/ v. 27 /5/

[3] Cigler, L.; Dobrovolný, I.; Hejna, J.: Systém MARK - komunikační kruh ČKD /Mark System - ČKD Communication Ring/. Technical Report NVZ-R/TZ-303, ČKD Polovodiče /1983/

[4] Popov, P.: Řídicí systém MARK - pokus o jednotné řešení elektronického systému ČKD Polovodiče /MARK Control System - An Attempt of Unified Design of ČKD Polovodiče Electronic System/. Proceedings of Conference 20 Years of ČKD Polovodiče

[5] Systém ČKD Mark - Základní technický popis /CKD MARK System - Basic Specifications/. CKD Polovodiče Progress Report NVZ-R/TZ-351. Prague /1984/

PROBLEMS OF INTER-INTERFERENCE IN A TWO-HOP RANDOM ACCESS RADIO NETWORK WITH BPSK SIGNALS

Aleksander T. Kozlowski

Institute of Telecommunications
Technical University of Gdansk
Gdansk
POLAND

A two-hop random access radio system is considered in the paper. For different spatial distributions of users the problem of inter-interference between different classes of users is analysed and a method of eliminating inter-interference is proposed.

INTRODUCTION

Random access radio computer systems like unslotted or slotted ALOHA networks have many advantages, see e.g. [1] , [2] , [3] , [4] . However, they also have a few disadvantages - as the most important we may mention the relatively high system realization cost, especially in the satellite version, and the low channel utilization. It has been shown, [3] , [4] , that a two-hop random access radio network partly eliminates these two disadvantages. In the two-hop system, users transmit their packets to repeaters which send these packets to the centre. It has been shown that in such a case the utilization of the two-hop system reaches values from 1.6 to 2.0 times greater than the values for star channel utilization. On the other hand, the realization cost of a two-hop system changes from 40% to 60% of that of a star system. Such values of two important system parameters allow the random access radio network to be implemented. However, there are many specific problems of two-hop system design which ought to be considered. One of them is the problem of inter-interference between packets which are sent by users collaborating with different repeaters. This interference generally decreases user-repeater channel utilization, so eliminating it may increase this utilization. In the present paper the problem of inter-interference is analysed and methods of eliminating such interference are given.

MODEL DESCRIPTION

Let us consider a two-hop random access radio system like the two-hop ALOHA network [1] , [2] , [3] . All the system users are divided into K disjointed classes. Users allotted to the same class collaborate with only one repeater in their packet transmissions to the centre. User packets are first transmitted to the repeater over the first-hop channel (FHC) , after which the repeater sends them to the centre via the second-hop channel (SHC) on a frequency other than that used in FHC. It is assumed that Binary Phase Shift Keying signals are used in both FHC and SHC. These signals are received by coherent optimum receivers, which minimize the probability of binary error for a given signal-to-noise ratio. In addition, the powers of user transmitters are chosen in such a way that the probability of

binary error in the packets received by each repeater is not greater than an assumed threshold; all users allotted to the same class use transmitters of the same power. Channel attenuation causes these powers to be a function of the distance between a repeater and the most distant user collaborating with this repeater. Thus, the power of user transmitters may generally be different for different classes of users. It has been shown [4] that correct reception by the k-th repeater of packets from k-th class users is possible even if in FHC these packets overlap packets transmitted by users allotted to other classes. This will be possible if the power of the interfering signals are smaller than the power of a packet received at the receiving point. Assuming that signal power decreases with distance according to $1/r^2$ or $1/r^4$, the relationship between the maximum distance r_k separating a k-th repeater from its user sending packets not disturbing the correct reception of packets transmitted by users allotted to other classes and the minimum distance $\varrho_k = \alpha r_k$; $\alpha \geqslant 1$; separating this user from other repeaters can be obtained. Values of α versus the minimum amplitude of the received signal are illustrated in Fig. 1. Thus, it is assumed that

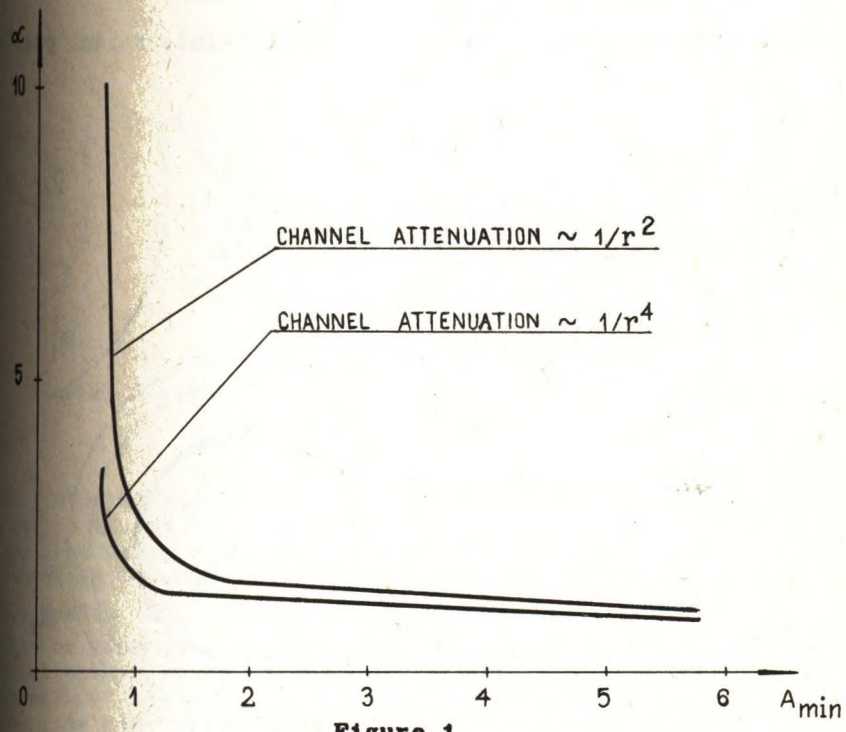


Figure 1
Values of α multipler versus the minimum amplitude of received signals

only the k-th class users within $\mathcal{R}_k^j$ circles of radii ϱ_k around the j-th repeater ; $k \in \mathcal{Y}$, where $\mathcal{Y}$ is a set of repeaters which are in the immediate vicinity of the j-th repeater ; can disturb

the correct reception of a packet sent by j-th class user located inside a $\mathcal{R}_j$ circle of radius r_j around the j-th repeater. Packets sent by other users cannot disturb the correct reception of such a packet.

ANALYSIS

In this section we consider the problem of inter-interference influence on FHC utilization in the two-hop random access system. Three cases of the spatial distribution of users will be analysed:

i/ all system users are concentrated in a number of disjointed areas,

ii/ there are lone users between these areas of user concentration,

iii/ all system users are more or less evenly spread over the system area.

A conception of inter-interference elimination is now presented for each of these three situations, should such interference occur. Let us start our analysis with case i/ when repeaters are allocated in such a way that the distance $r_{i,k}$ between the i-th repeater and the k-th repeater fulfil the following inequality:

$$r_{i,k} \geqslant \varrho_i + \varrho_k \quad ; \; i = k \; ; \; i,k = \overline{1-K} \qquad (1)$$

where $\varrho_i = \alpha r_i$ and $\varrho_k = \alpha r_k$.

In this case, illustrated in Fig. 2, all inter-interfering packets

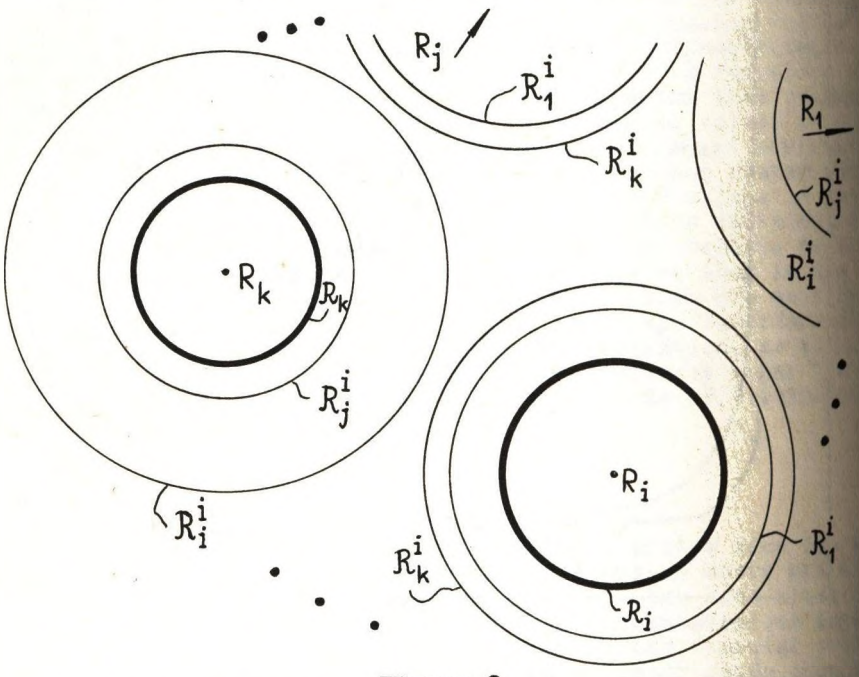


Figure 2
Spatial distribution of users in case i/. R_k – the k-th class repeater, $\mathcal{R}_k$ – the circle of k-th class users allocation, $\mathcal{R}_k^i$ – the circle of the k-th class user interfering range

are sent from a distance greater than ϱ_i or ϱ_k to the i-th or the k-th repeater respectively. Thus, k-th class packets cannot disturb the correct reception of packets sent by the i-th class users to their repeater and vice versa - inter-interference does not decrease the FHC utilization of each class.

We consider next the two-hop random access system in the same way as the system analysed above; however, there are lone users between the areas of user concentration (see Fig. 3) . These lone users have

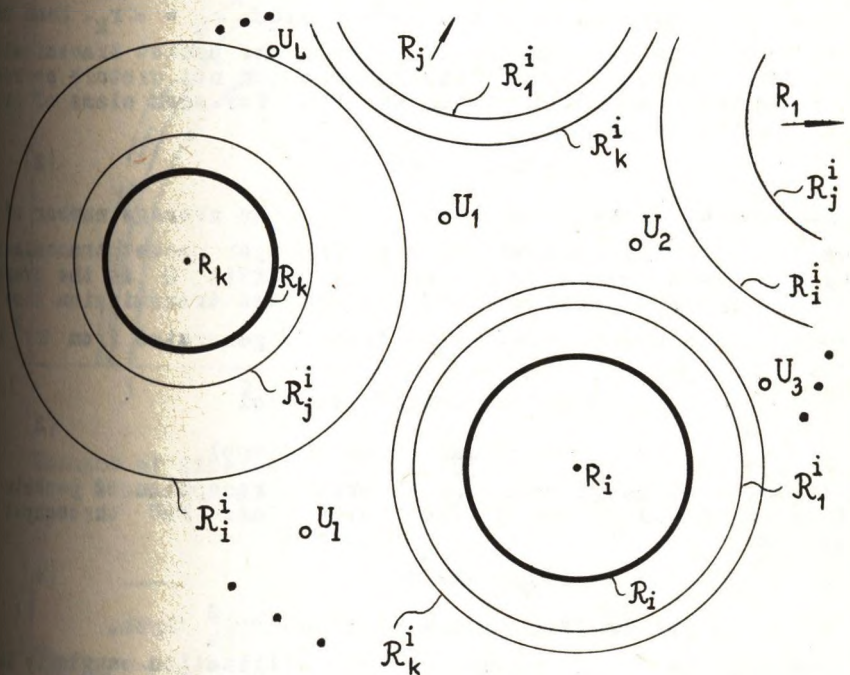


Figure 3
Spatial distribution of users in the case ii/ . U_1 is the k-th lone user.

to be connected with the centre. It is possible to connect them with the centre in the following two ways:

a/ the lone users use transmitters of such power that they are connected through FHC with their nearest repeater. In this case the radius ϱ_k of the R_k^i circle increases and inter-interference generated by these users decreases FHC utilization. Thus, this problem is similar to the problem iii/ which will be considered later,

b/ the lone users are connected with the centre via SHC. Thus, the problem of inter-interference in FHC does not exist - we have a problem like case i/ ; however, the larger number of users using SHC decreases SHC utilization and increases the system realization cost.

In the third case, users are more or less evenly spread over the system area. For such a spatial distribution of users, the R_k

circles with an effective transmission range of radii r_k usually overlap. Thus, there are many users in the system generating additional traffic which disturbs the correct reception of the k-th class user packets. This traffic causes a considerable decrease in FHC utilization.

Let us estimate the changes of FHC utilization which are caused by inter-interference. We assume that all N system users are evenly distributed over the system area. This area is completely covered by $\mathcal{R}_k$ circles. All $\mathcal{R}_k$ circles have the same radius equal to r_k. Thus, all $\mathcal{R}_k^i$ circles have the same radius $\varrho_k = \alpha r_k$. Each user sends to FHC the same number of packets per packet transmission time. In this case, if inter-interference did not disturb correct packet reception, the utilization of FHC for each class of users could be estimated at:

$$S_k = G\exp\left(-aG\right) \qquad (2)$$

where the k-th class throughput S_k is the average number of packets successfully transmitted via FHC per packet transmission time by the k-th class users, and the traffic G is the average number of packets offered to FHC per packet transmission time by the k-th class users, i.e. it is traffic generated from πr_k^2 area. However,

$$a = \begin{cases} 1 & \text{for a slotted protocol} \\ 2 & \text{for an unslotted protocol} \end{cases} \qquad (3)$$

If inter-interference disturbs the correct reception of packets generated by k-th class users, the k-th class FHC throughput S_k' is equal to:

$$S_k' = G\exp\left(-a\alpha^2 G\right) \qquad (4)$$

where $\alpha^2 G$ is the traffic generated from $\pi\varrho_k^2$ area.

Thus, the decrease in first-hop channel utilization caused by inter-interference, which can be measured by the ratio $\gamma = S_k'/S_k$ is equal to:

$$\gamma = \exp\left[a\left(1-\alpha^2\right)G\right] \qquad (5)$$

It is obvious that this ratio ought to be as great as possible, i.e. it ought to be close to 1. On the other hand, this ratio decreases if α increases. Note too, that α increases if the minimum signal power decreases - see Fig. 1. In addition, all the above considerations are made for the assumed probability of binary error caused by external channel noise, i.e. for a given signal-to-noise ratio. Thus if noise power is small, we can use signals of small power. But in this case, the multiplier α is great and inter-interference decreases FHC utilization considerably. Greater channel noise power means that signal power has to be greater, this decreases the value of α and decreases the influence of inter-interference on FHC utilization. All the results can be seen in Fig. 4.

It is obvious that each class traffic G influences the amount of inter-interference which can disturb the correct reception of k-th class packets. This influence is illustrated in Fig. 5. Note that for a great signal power the ratio γ decreases slowly with traffic G, whereas for a smaller signal power this decrease is faster. The results given above show that in many cases inter-interference can

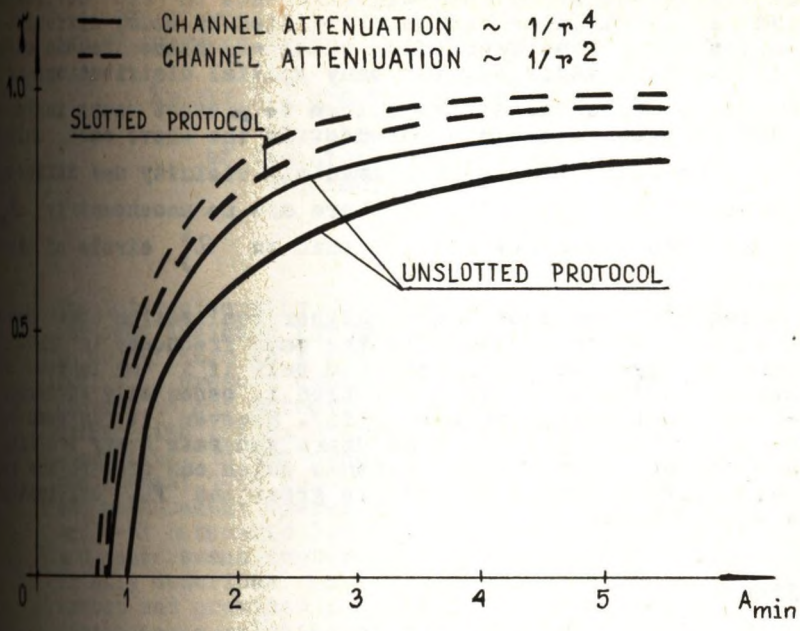


Figure 4
Changes of first-hop channel utilization versus the minimum amplitude of transmitted signals

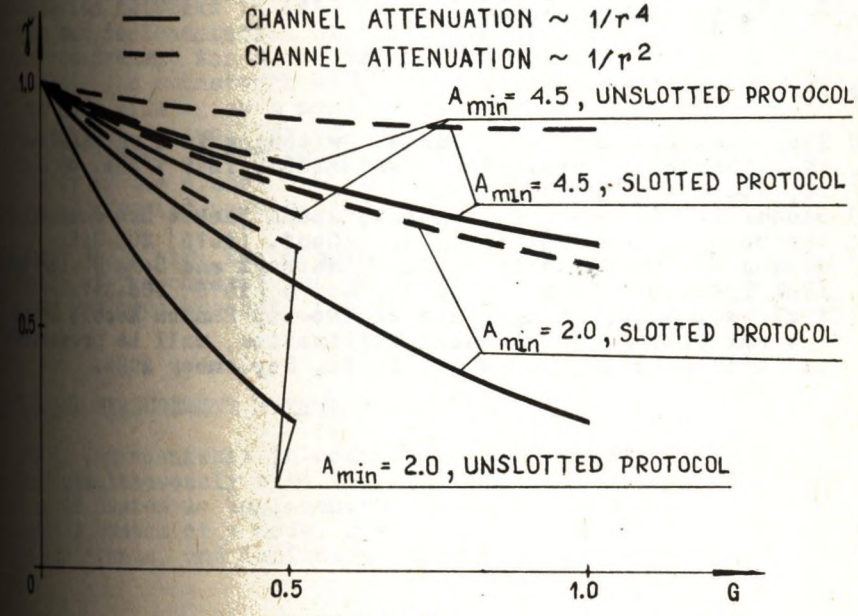


Figure 5
Changes of first-hop channel utilization versus each class traffic

decrease FHC utilization substantially. Thus, it is advisable to decrease the influence of inter-interference on FHC utilization. This kind of interference can be eliminated by using different frequencies for users (and repeaters) which are allocatted in adjacent $\mathcal{R}_k$ circles. In addition, for many spatial distributions of users three different frequencies are enough to protect users inside an $\mathcal{R}_k$ circle from inter-interference. In the above case, only the $\mathcal{R}_k$ circles which are in the immediate vicinity use different frequencies. This is possible if there are no unconcentric $\mathcal{R}_k$ circles with one frequency lying within an $\mathcal{R}_j^i$ circle of the same frequency.

The solution proposed above has a higher realization cost than the two-hop random access system with the same frequency in FHC. So, this solution ought to be implemented only if it can improve FHC utilization considerably. This solution is especially advisable for the spatial distribution in case iii/. However, it can even be implemented in case ii/ if lone users generate heavy traffic. In this case amount of inter-interference which can disturb the correct reception of k-th class packets is great and FHC utilization decreases substantially.

CONCLUSION

In the present paper the problem of inter-interference in a two-hop random access system is considered. It has been shown that when users are more or less evenly spread over the system area this interference can decrease the first-hop channel utilization considerably. In this case it is advisable to eliminate this interference by using different frequencies for the different classes of users which are in the immediate vicinity.

REFERENCES

[1] Kleinrock L., Lam S.S., Packet Switching in a Multiaccess Broadcast Channels: Performance Evaluation, IEEE Trans. on Comm., vol. 23, No. 4 (1975) 410-422.

[2] Binder R., Abramson N., et al., ALOHA Packet Broadcasting - A Retrospect, Proc. of Nat. Comp. Conf. (1975) 203-215.

[3] Gitman I., On Capacity of ALOHA Network and Some Design Problems, IEEE Trans. on Comm., vol. 23, No. 3 (1975) 305-317.

[4] Kozlowski A.T., An Analysis of Two-hop Random Access Radio Systems with Higher Channel Utilization, will be presented on the 11-th Int. Teletraf. Cong., Kyoto, September 1985.

THE STRUCTURE OF THE RELIABLE CONTROL AND MEASUREMENT SYSTEM IN THE COMPUTER NETWORK

Hudyma E., Kosmulska-Bochenek E.

Computer Center
of Technical University
Wroclaw, Poland

The Interuniversity Computer Network in Poland has been designed for the academical community to be utilized as a working and as a research tool. Our present experience with this system brought to our attention two major problems concerned with the maintaince of its operational capability. In the paper these two problems are considered: 1-th reconfiguration of the control and measurement system in faulty environment, and 2-th automatic location of the new control and measurement center in this system. The paper briefly describes the control and measurement system and provides a simplified algorithm for simultaneous solution of both problems.

INTRODUCTION

The problem of control, diagnostic and measurements in computer communication networks is one of the most important elements of creation and implementation of the network. The high reliability is a basic requirement for such system. One way to achieve this goal is to supply the subnetwork with the control and diagnostic system with one node working as a measurement center besides its regular operations. The reliability of this system depends on proper location of the control and measurement center (CMC). Additionally, the reliability of the system can be increased by providing the system with reconfiguration possibility.

This paper deals with two problems, location of the CMC, and reconfiguration of the control and measurement system (CMS). The described system has been proposed for the Interuniversity Network MSK in Poland.

CONTROL AND MEASUREMENT SYSTEM (CMS)

The CMS is indispensable in computer network and it should be designed simultaneously with the network. In MSK a hierarchical three level system is implemented. The lowest level is a local measurement system of a node. A system journal, collection of the states and events, and monitoring them are the basic tools in this system.

The next level of the CMS is the distributed system. Measurement packets are the major features of this level. The packets transfer the measure and diagnostic informations between the nodes. The traf-

fic of that kind of packets is called a diagnostic flow in contradistinction to the rest of the network flow, called an user flo In the distributed system each node has equal rights to send and receive the packets.

The third level of the CMS is characterized by one node functioning as a measurement center. The CMC is entitled to execute special fu ctions which must be done in the network in one node only. Besides this, the CMC realizes its regular functions as any other node. Co munication between the nodes at levels two and three is assured by the packets. The third level uses all features of the lower levels and controls the system by its own packets.

NETWORK MODEL DESCRIPTION

The communication subnetwork is modeled as an undirect graph G(N,L). For each pair s,t (standing for source, and terminal) this graphs is replaced by the directed graph $G^*(N,L^*)$ as follows [3]:

a. replacing each link that ends at S in L by a directed link in L^* from s to the appropriate node
b. replacing each link that ends at t in L by a directed link in L^* from the appropriate node to t
c. replacing each other link (i,j) in L by two directed links (i,j) and (j,i) in L^*.

A link flow from i to j in G^*, f_{ij}, is a nonnegative link weight. Physically, it is a number of packets transmitted between the nodes (i,j) in a time period. Similarly, the capacity of link (i,j), c_{ij} is another nonnegative link weight interpreted as a maximum number of the packets which can be transmitted between the nodes (i,j) in a time period.

A feasible s,t pattern in G^* is a set of link flows that satisfies the following equations for distinct nodes s and t:

$$\sum_{x \in N} f_{ix} - \sum_{y \in N} f_{yi} = \begin{cases} v_{st} & \text{if } i = s \\ 0 & \text{if } i \neq s \text{ and } i \neq t \\ -v_{st} & \text{if } i = t \end{cases}$$

$$c_{ij} \geqslant f_{ij} \geqslant 0, \quad i,j \in N$$

The nonnegative quantity v_{st} is the value of the flow pattern.

Ford-Fulkerson algorithm can be used to find the maximum flow s,t in G^*. The method used is to assign a flow capacity of 1 to all t links in the network. From the maximum flow, minimum cut theorem it is known that the maximum flow from s to t equals the capacity of the minimum cut which equals the number of links in the minimum cut set [4].

THE DESIGN OF THE RELIABLE CMS

The high reliability of the CMS is required because in addition to the measurement function the control and diagnostic functions in the subnetwork are performed. It is assumed that the CMS is operable if the CMC is working properly and if the connections exist between the CMC node and each operable node in the subnetwork. Reliabilit

of this system consists of two factors, reliability of the CMC (hardware and software), and location of the CMC.

The reliability of such system is:

$$R_s(t) = R_c(t) \quad R_k(t) \tag{1}$$

where:
$R_s(t)$ - reliability of CMS
$R_c(t)$ - reliability of CMC
$R_k(t)$ - reliability being the result of the configuration.

R_k reliability, later called configuration reliability which is the result of location CMC in subnetwork is defined as a probability that between CMC and any other node at least one operable link exists. Denote that i-node operates as the CMC, then:

$$R_k(t) = \prod_{j \in N, j \neq i} R_{i,j}(t) \tag{2}$$

where:
$R_{i,j}$ - probability that between i and j nodes at least one operable link exists.

The measure of R_k gives the bottom estimation of the reliability of the link between CMC and others nodes.

The value of $R_{i,j}$ can be easily determined based on the minimal cut set between i-node (CMC) and j-node. We will assume that each link has a probability of failure q, $0 \leqslant q \leqslant 1$ and that all the link and node failures are statistically independant.

Let $Q_{i,j}$ indicates the probability that the link between CMC and the j-node is not operable. Then

$$Q_{i,j}(t) = \prod_{v_{i,j}} q = q^{v_{i,j}} \tag{3}$$

where $v_{i,j}$ - the number of the links of cut set between i (CMC) and j nodes.

Because $R_{i,j}(t) = 1 - Q_{i,j}(t)$ (4)

then finally the configuration reliability of the CMS can be determined by the formula:

$$R_k = (1-q^{v_{i,1}})(1-q^{v_{i,2}})\ldots(1-q^{v_{i,n}}) = \prod_{j} (1-q^{v_{i,j}})$$

$$\text{for } j \in N,\ j \neq i \tag{5}$$

where i-node operates as the CMC.

Proper location of the CMC gives the possibility of increasing the value of this expression. The value of R_c in formula (1) in the CPS is a well important because failure of the CMC results in degradation of the whole system.

The system with a dynamic reliability structure is proposed to improve the R_s reliability. It means that when the CMC fails, its function is taken by any other node in the subnetwork. The choice

is made with regards to the maximum of the configuration reliability R_k. The expression for the reliability function of the system with the reconfiguration is as follows [7]:

$$R_s(t) = R_c(t) \prod_j (1-q^{v_{i,j}}) - \left[\int_0^t R_1(t|x)dR_c(s) \right] \prod_j (1-q^{v_{1,j}}) \quad (6)$$

where

$R_1(t|x)$ - the probability of the normal work of the 1-node, at least to the moment t and under the condition that in the moment x the work conditions of this element were altered because of the switching.

$-dR_c(x)$ - the probability of failing of the CMC near to moment x.

$dR_c(x)$ has a minus sign, because of the derivative of descending value. For this reason the reliability of the system with the reconfiguration is greater than without it.

RECONFIGURATION

The reconfiguration of the CMS under failing conditions as discussed in the previous chapter requires the following steps:

1. fault detection,
2. determination of the new configuration of the defected subnetwork,
3. selection and activation of the new CMC,
4. recovery of the CMS.

The algorithm of the reconfiguration requires the following assumptions:

- only a single CMC is active in the network,
- every node of the subnetwork can function potentially as the CMC,
- defective function is indicated by a lack of communication between the nodes and the CMC over the period longer than τ_k,
- the nature of the lack of communication is insignificant (failed CMC or failed communication lines or dealy on the lines),
- defective function is not permitted during the reconfiguration.

The algorithm of the reconfiguration is realized in every node of the subnetwork. The CMC must have the ability to send the packets to the each node with the frequency not less then time periods τ_k. When the packet from the CMC to i-node has not come in τ_k, the CMS starts to reconfigurate.

The second condition of the proper reconfiguration procedure is the ability of every node to establish the actual donfiguration of the subnetwork. This function can be done by special reconfiguration packets which are emited by each node using flooding strategy. A dynamically created table of the configuration which is initiating from the neighbours of the originating node establishes the content of this packet. The flow of the reconfiguration packets is stopped when any changes have not occured during the information exchanges between the nodes.

The node for the CMC is selected by means of the configuration table existing in each node and it is the subject to the next consideration.

The last step of the reconfiguration procedure is the recovery the new CMS. This recovery can be accomplished in several ways. One of

them is to inform the rest of the nodes about the new CMC. The packets are used for establishing a contact between the new CMC and the other nodes. The new CMC communicates with the nodes of the subnetwork using the standard rules.

LOCATION OF THE CMC

Location of the node serving as the major source of the diagnostic flow is determined by several different factors, depending upon the implementation of the subnetwork. There are three important criteria for the location of the CMC:

- reliability of the CMS
- cost of the CMS
- effectiveness of the subnetwork in terms of flow.

For each of them boundry values are assumed, which will be used for immediate disqualification of the CMC node. Thus, if the reliability of the CMC falls below the predetermined boundry value the node is no more taken into consideration in the next steps of the algorithm. By the same principle if the flow is greater than allowed the continuation of this calculation is not permitted.

The procedure is not time consuming, because each step of it reduces the number of the considered nodes for latter computation. The level of reductions depends on the particular configuration of the defected subnetwork.

The algorithm for the selection of the node to function as CMC consists of 3 steps:

- initial selection (preselection)
- selection according to the reliability parameter
- optimization with regard to the cost and effectiveness.

The algorithm in the form of the Pascal procedures is presented. The number of the all nodes of the network is denoted as n. The type of the variable is defined as bellow:

```
TYPE tab = ARRAY [1..n] OF integer;
```

The network configuration as k matrix is determined and as a global variable is declared:

```
VAR k : ARRAY [1..n,1..n] OF 0..1;
```

The purpose of the preselection is to reduce the number of the nodes which have to be examined in the next steps. Based on the configuration table it is easy to find the number of the input/output lines for each node. Because the minimal cut set between the i-node and any other is not greater than the number of the input/output lines, therefore one can remove from the set of the all nodes in the graph, the nodes for which the number of the lines is less or equal to the assumed value p_{min}.

The procedure SELECT gives the identifications of the nodes (their numbers) which still have a chance to be the CMC, in the table m and their number in the variable d.

```
PROCEDURE select (pmin:integer,VAR m:tab,VAR d:integer);
VAR i,j:integer;
    lin:tab;
BEGIN
    FOR i:=1 TO n DO
        BEGIN
           lin[i]:=0;
           FOR j:=1 TO n DO lin[i]:=lin[i]+k[i,j]
        END;
    d:=0;
    FOR i:= 1 TO n DO
        IF lin[i]>pmin THEN BEGIN
                               d:=d+1;
                               m[d]:=i
                            END
END;
```

The second step of the algorithm determines the reliability parameter. Based on expression (5) in chapter 4 it is sufficient to calculate value v_{ij} which is the number of links in (i,j) minimal cut set, where j denotes the original node, and i denotes the node which is tested for CMC functions.

For this purpose the Ford-Fulkerson algorithm [5] is used. It is given as a function:

```
FUNCTION ff(i,j:integer):integer;
```

As the next the minimum value of v_{ij} is found. This value as a poor reliability parameter is interpreted. The number of its occurance is counted for each node. The nodes with the minimal number of occurences of the poor reliability parameters are taken for the next computation.

The procedure of the selection of the nodes with the best reliability parameters is repeated by iteration until the set of the nodes with identical reliability parameters are identified.

The results of the procedure RELIAB in the table m and in the variable d are given, similarly as it is done in the procedure SELECT.

```
PROCEDURE reliab(VAR m:tab,VAR d:integer);
VAR v:ARRAY[1..d,1..d] OF integer;
    l:ARRAY[1..d] OF integer;
    i,j,r,s,vm,lm:integer;
    b:boolean;
BEGIN
    FOR i:=1 TO d DO
       FOR j:=1 TO d DO
          IF i=j THEN v[i,j]:=maxint
                 ELSE v[i,j]:=ff(m[i],m[j]);
    vm:=maxint;
    b:=true;
    REPEAT
       FOR i:=1 TO d DO
          FOR j:=1 TO d DO
             IF v[i,j]<vm THEN vm:=v[i,j];
          lm:=maxint;
          FOR i:=1 TO d DO
```

```
    BEGIN
       l[i]:=0;
       FOR j:=1 TO d DO
             IF v[i,j] = vm THEN l[i]:=l[i]+1;
          IF l[i]<lm THEN lm:=l[i]
       END;
    s:=0;
    FOR i:=1 TO d DO
       IF l[i] =lm THEN BEGIN
                           s:=s+1;
                           m[s]:=m[i];
                           r:=0;
                           FOR j:=1 TO d DO
                              IF l[j]=lm THEN BEGIN
                                                 r:=r+1;
                                                 v[s,r]:=v[i,j]
                                              END
                        END
                   ELSE b:=false;
    d:=s
  UNTIL b
END;
```

The next step determines the cost and flow parameters. Under assumption that the cost of the hardware and the software in all nodes is the same, the CMS cost can be minimalized by decreasing the cost of the diagnostic transmission between the CMC and the remaining nodes in the subnetwork. Considering that the average distance between the nodes is equal, the cost parameter is determined as the number of intermediate nodes in diagnostic flow between the CMC and each other node of the subnetwork.

The effectiveness criterium for the selection of the CMC protects against overload of the diagnostic flow, taking under consideration the minimal level of this flow between CMC and any other node of the subnetwork. The procedure relies on designation of the diagnostic flow f_d between the CMC and the other nodes under assumption that a minimal feasible link flow f_{dmin} exists and that value of the link flow is not greater than f_{dmax}.

The diagnostic flow should be sent by the shortest path. Hence, the f_d flow can be determined by finding the shortest path between the CMC and the given node, and next by checking the condition $f_{dmin} \leq f_d \leq f_{dmax}$ under assumption of the minimum link flow f_{dmin}. Thus, the cost and the f_d flow can be estimated by the algorithm of designation of the shortest path in the graph G . The problem of the finding of the shortest path in the graph from s to t requires a link weight function. The distance (s,t) in cost determination, and f_{dmin} in calculation of the flow should be regarded as the weight functions.

In the location algorithm the most important are not the absolute values of these two parameters but their relation leading to selection of the node for CMC for which these parameters should be the best in CMS. The advantage from this approach is, that it is sufficient to use f_{dmin} only as the weight function because the f_{dmin} being equal on each link images the cost of the link as well. This simplification reduces the complexity of the algorithm but does not effect the final result.

Finally, the problem relies on the finding of the shortest path

and its length in the graph G^* from s to t. The length of the path is defined as the sum of the links in the path. The sum of the shortest paths determines the diagnostic flow cost.

The length of the shortest path is calculated according to the Dijkstra algorithm [6] and is given as a function:

```
FUNCTION kk(i,j:integer):integer;
```

This last step of algorithm is made by procedure COST. The results are given in the table m and in the variable d.

```
PROCEDURE cost(fm:integer,VAR m:tab,VAR d:integer);
VAR i,j,km:integer;
    c:ARRAY [1..d]OF integer;
    ck:ARRAY[1..d,1..n] OF integer;
    b:boolean;
BEGIN
   FOR i:=1 TO d DO
      BEGIN
         c[i]:=0;
         j:=0;
         b:=false;
         REPEAT
            j:=j+1;
            ck[i,j]:=kk(m[i],j);
            IF cc[i,j]<fm THEN c[i]:=c[i]+ck[i,j]
                          ELSE BEGIN
                                  c[i]:=maxint;
                                  b:=true
                               END
         UNTIL(j=n)OR b;
      END
   km:maxint;
   FOR i:=1 TO d DO
      IF c[i]<km THEN km:=c[i];
   j:=0;
   FOR i:=1 TO d DO
      IF c[i]=km THEN BEGIN
                         j:=j+1;
                         m[j]:=n[i]
                      END;
   d:=j
END;
```

If there is found more than one node with equal reliability values the decision is made on the basis of the flow, and then of the cost. In case if all parameters are equal the selection is arbitrary.

AN EXAMPLE

Given is the subnetwork (Fig. 1). Select the new node performing the CMC function, under condition of failing of the primary CMC.

The subnetwork after failure will have the structure as in Fig. 2.

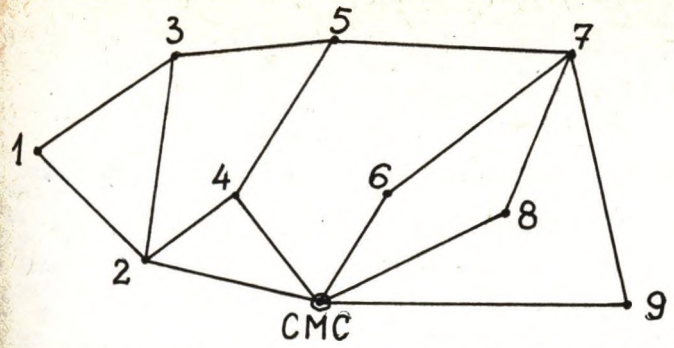


Fig. 1. The subnetwork before the defect of the CMC.

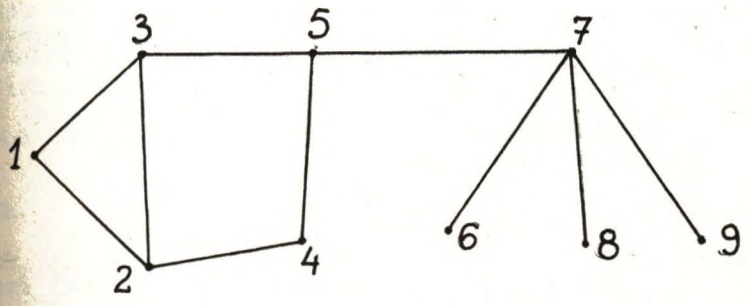


Fig. 2. The subnetwork after the defect of the CMC.

The configuration table of the defected subnetwork will be established according to the Table 1.

k_{ij}	1	2	3	4	5	6	7	8	9	Number of the direct connections
1	0	1	1	0	0	0	0	0	0	2
2	1	0	1	1	0	0	0	0	0	3
3	1	1	0	0	1	0	0	0	0	3
4	0	1	0	0	1	0	0	0	0	2
5	0	0	1	1	0	0	1	0	0	3
6	0	0	0	0	0	0	1	0	0	1
7	0	0	0	0	1	1	0	1	1	4
8	0	0	0	0	0	0	1	0	0	1
9	0	0	0	0	0	0	1	0	0	1

Fig. 3. Configuration table for the defected subnetwork.

The location algorithm for the new CMC.
The assumption:

$$p_{min} = 2$$
$$fd_{min} = 1$$
$$fd_{max} = 3$$

1. From the configuration table we obtain:
   $M = \{2,3,5,7\}$
2. Procedure RELIAB gives intermediate results as follows:

v_{ij}	2	3	5	7
2	max	3	2	1
3	3	max	2	1
5	2	2	max	1
7	1	1	1	max

Fig. 4. The matrix V of the minimal cut sets.

a) after the first iteration:

vm = 1
lm = 1

v_{ij}	2	3	5
2	max	3	2
3	3	max	2
5	2	2	max

Fig. 5. The matrix V after the first iteration

$M = \{2,3,5\}$, d = 3

b) after the second iteration:

vm = 2
lm = 1

v_{ij}	2	3
2	max	3
3	3	max

$m = \{2,3\}$, d = 2

c) after the third iteration:

vm = 3
lm = 1

v_{ij}	2	3
2	max	3
3	3	max

The final results of procedure RELIAB are:

$m = \{2,3\}$, d = 2

3. The intermediate results of the procedure COST are as follows:

d_{ij}	1	2	3	4	5	6	7	8	9
2	1	0	1	1	2	4			
3	1	1	0	2	1	3	2	3	3

Fig. 6. The matrix of the shortest path.

$d_{26} > f_{dmax}$ and for this reason the node 2 is no more taken under consideration.

$M = \{3\}$, $d = 1$

The node 3 becomes the new CMC in the presented example.

SUMMARY

The paper presents the method of dynamic reconfiguration of the hierarchical control and measurement system in the communication subnetwork. Two problem can be distinguished which can exist independently. First, is the recofiguration under failure, and the second is the location of the control and measurement center. The location problems does not need to be a part of the reconfiguration. Instead of the location procedure the next CMC can be established by selecting one of the previously prepared nodes. Thus, the cost of reconfiguration is reduced, but depending on the nature of the failure is does not guarantee the optimal performance of the CMS. The solution of the reconfiguration problem is most important when the failure in the subnetwork is irreparable or if repaires are time consuming.

The location problem can exist without the need for reconfiguration. Presented in the paper algorithm allowes identification of the location of the CMC during designing of the subnetwork.

REFERENCES

[1] Battek, J., Bilski, E., Huzar, Z.: Interuniversity Computer Network (MSK) - technical and user description (in polish), CO PW Reports SPR/20/82 (Wroclaw 1982)

[2] Bieleninik, E., Hudyma, E., Kosmulska-Bochenek, E., and others The idea of the control and measurement system of the communication network MSK (in polish), CO PW Reports SPR/3/83 (Wroclaw 1983)

[3] Cravis, H. Communications Network Analysis (D.C. Heath and Co. Lexington, Massachusetts, Toronto 1981) 35-49

[4] Davis, W., Barber D. Communication Networks for Computers (John Wiley Sons London - N.Y. - Sydney - Toronto 1973) 422-456

[5] Frank, H., Frisch, I. Communication, Transmission and Transportation Networks (Addison-Wesly Publishing Company 1971)

[6] Goodman, S., Hedetniemi, S. Introduction to the Design and Analysis of Algorithms (McGraw-Hill Book Company 1971)

[7] Zamojski, W. Reliability and the system exploitation (University of Wroclaw, 1981)(in polish)

PACKET COMMUNICATION ON A T-USER COLLISION CHANNEL WITHOUT FEEDBACK VIA CYCLOTOMIC PROTOCOL SEQUENCES

Nguyen Quang A, Tamas Zeisel
Institute of Telecommunication and Electronics, Technical University of Budapest
1111 Budapest, Stoczek u.2, Hungary
and Instiute of Computer Science and Cybernetics, Vietnamese Academy of Sciences

In a computer network consisting of T terminals it is assumed that at any given time there are at most M simultaneous active terminals trying to access a common central facility over a common channel. The channel is assumed to be a time-slotted frame-asynchronous multiple-access collision channel without feedback. Each active terminal controls its packet transmission via a periodically repeated binary protocol sequence of length N whose 1's correspond to slots where a packet is sent. By using appropriate protocol sequence the receiver can uniquely identify the active users and reconstruct their packets from the received patterns of idles, successes and collisions at the channel output when the number of active users is at most M. A construction of such sequences is presented based on the theory of cyclotomy.

INTRODUCTION

Consider a computer network consisting of T spatially separated terminals. Terminals are of bursty character so it is reasonable to assume that at any given time only a small percentage of terminals, also called users, wish to transmit information to a central facility over a common channel. The central station sends information to terminals over a separated channel which is not investigated in this paper. The channel is a time-slotted frame-asynchronous collision channel without feedback. Formally the channel can be defined as one having T inputs and one output. If $x_1, x_2, \ldots, x_T$ stand for inputs at a given slot then the output of the channel y is

$$y = \begin{cases} 0 & \text{if } x_j = 0 \text{ for all } j \quad \text{(idle)}; \\ x_i & \text{if } x_j = 0 \text{ for all } j \neq i \quad \text{(succes)}; \\ * & \text{otherwise} \quad \text{(collision)} \end{cases}$$

Each user can send a packet only at the beginning of a slot. A packet may be an arbitrary binary string of fixed length (in one extreme case packet may consist of a single bit). The packet time is assumed to equal to slot-time.
The network is similar to the slotted ALOHA network. A distinctive difference between two networks is that in the ALOHA network each user can hear the channel-output and depending on the state of channel (idle, succes, collision) he makes decision whether he sends/repeats a packet or keeps silent in the very next slot according to some rules; in our network users do not know the state of channel.

Each active user controls his packet transmission via a periodically repeated binary protocol sequence of length N whose 1's correspond to slots where a packet is sent. By using appropriate protocol sequences the central receiver can uniquely identify the set of active users and reconstruct their packets from the received patterns of idles, succeses and collisions at the output of channel when the number of active users is at most M.

For simplicity, suppose that a packet takes values from some finite field GF(Q). A (T,M,s) protocol set is a set of T protocol sequences such that, when at most M of the T users are active, each active user achieves at least s successful transmissions every N slots and the received patterns of idles, successes and collisions in N successive slots uniquely determine the active set of users as well as the subsets of slots where each active user has sent packets, for every possible relative alignment of the protocol sequences. The performance of protocol set is measured by the sum-rate or throughput and delay. Since there are at least s successful packets during a period N of protocol sequence and there are at most M active users, the sum-rate or the throughput of protocol set (T,M,s) is defined to be

$$R = Ms/N \qquad (1)$$

In the sequel we use the length N of protocol sequence as a measure of delay.

Let $X_i = (X_{i,0}, X_{i,1}, \ldots, X_{i,N-1})$ denote the protocol sequence of i-th user with period N. Because of the assumed asynchronism, it will appear to the receiver that the user i has the protocol sequence $S^j X_i$ for some integer j, where any j such that $0 \leq j < N$ is possible. Thus, we assume that $X_1, X_2, \ldots, X_T$ are cyclially distinct, furthermore it is assumed that each X_i is of maximal period, and has weight w, i.e. $S^k X_i \neq X_i$ for all $k \neq 0 \pmod N$ and X_i contains exactly w 1's. The pattern of idles (0), successes (1) and collisions (*) can be written as follows.

$$r = \sum_{k=1}^{m}{}' S^{j_k} X_{i_k} \qquad (2)$$

where $(i_1, i_2, \ldots, i_m)$ is the set of indices of the active users, $0 \leq j_k < N$, and each component of r is replaced by * if it is larger than 1.

Let C be the set consisting of $X_1, X_2, \ldots, X_T$ and all their N-1 distinct cyclic shifts. Let c denote the maximum correlation between X_i and X_j, i.e.

$$c = \max_{\substack{i,j \\ 1 \leq k < N \text{ if } i = j \\ 0 \leq k < N \text{ if } i \neq j}} \langle X_i, S^k X_j \rangle \qquad (3)$$

where $\langle .,. \rangle$ denotes the inner product.

For $X_1, X_2, \ldots, X_T$ to constitute a (T,M,s) protocol set, r in (2) must

first identify uniquely $i_1, i_2, \ldots, i_m$ and $j_1, j_2, \ldots, j_m$ whenever $m \leq M$. This must be the case if every $S_j X_i$ is such that not all its 1's can be covered by 1's in any M of other N-tuples in **C**. But, since $S_j X_i$ has weight w and no other N-tuple in **C** can cover more than c of its 1's, this must be the case whenever

$$w > Mc \qquad (4)$$

The other condition for a (T,M,s) protocol set is that each of the M or fewer users obtain at least s successful packets in slots 1,2,...,N. But, as each other user can cause at most c collisions with the packets of a given user, this condition must hold whenever

$$w \geq (M-1)c + s \qquad (5)$$

Combining (4) and (5) we see that $X_1, X_2, \ldots, X_T$ is a (T,M,s) protocol set for some integer M satisfying

$$M \geq \min [\, (w-1)/c \,;\, (w-s)/c + 1 \,] \qquad (6)$$

Thus, we have established the sufficient condition for a set of cyclically distinct binary vectors $X_1, X_2, \ldots, X_T$ to be a (T,M,s) protocol set.

Procedure of channel access

The procedure of channel access is as follows. Each active terminal encodes s packets **u** (an s-tuple over GF(Q)) into w packets **x** (a w-tuple over GF(Q)) with the aid of a Reed-Solomon code of parameter (w,s), i.e. Reed-Solomon code of length w and s information symbols. Explicitly, if α denotes a primitive element of GF(Q) , $\mathbf{u} = (u_1, u_2, \ldots, u_s)$ and $\mathbf{x} = (x_1, x_2, \ldots, x_w)$ then $\mathbf{x} = \mathbf{uA}$

$$x_1 = u_1 + u_2 + \ldots + u_s$$

$$x_2 = u_1 + \alpha u_2 + \ldots + \alpha^{s-1} u_s$$

$$\cdots\cdots\cdots$$

$$x_j = u_1 + \alpha^{j-1} u_2 + \ldots + \alpha^{(j-1)(s-1)} u_s$$

$$\cdots\cdots\cdots$$

$$x_w = u_1 + \alpha^{w-1} u_2 + \ldots + \alpha^{(w-1)(s-1)} u_s$$

During one period of protocol sequence these w packets **x** are sent at those slots corresponding to 1's in the protocol sequence.

Synchronization and decoding

Let Y be the binary sequence of length 2N formed by the received patterns of idles, successes and collisions such that idle corresponds to 0, success and

collision correspond to 1. Slide the protocol sequence X_i on Y. If there is a segment of length N of Y, say starting at slot j, $0 \leq j < N$, which covers X_i, then decide that the user i is active and $j_i = j$, cf. (2); otherwise decide that the user i is not active during the given period 1,2,...,N. It follows that the synchronization or identification delay is between 0 and N slots. Having determined the set $(i_1, i_2, \ldots, i_m)$ of active users and the set $(j_1, j_2, \ldots, j_m)$ and knowing all X_i the receiver can pick up w packets **x'** (some of them might be destroyed due to collisions and the number of these packets is at most w - s) of each active user and from the patterns of idles, successes and collisions the receiver knows which packets are correct from these w packets **x'**. The number of correct packets is at least s. Choose s correct packets from **x'**, say x_{j1}, $x_{j2}, \ldots, x_{js}$. These s equations from w equations above uniquely determine the information packets **u**, since $w < Q$ by using of Reed-Solomon code and in this case the determinant of any submatrix of size s x s of **A** is Vandermonde determinant.

The region of achievable rates of the collision channel without feedback with accessing via protocol sequences has been determined by Massey [1], Massey and Mathys [2] as well as Tsybakov and Likhanov [3]. They have constructed optimal protocol sequences that achieve these maximum rates. For the case of equal rate users, the maximal sum-rate is given by [3]

$$C_M = (1 - 1/M)^{M-1} \qquad (7)$$

which depends only on the number of active users M and is the same as the maxium throughput of the ALOHA channel [10pp.366-367]. The period of the optimal protocol sequence is, however, given by [3]

$$N = M^T \qquad (8)$$

The total delay is composed of the transmission delay, the synchronization delay and the decoding delay. We can use, for the first approximation, the period of protocol sequence as a rough measure of average delay. Although the optimal protocol sequences achieve the maximum throughput, the delay given by (8) is too large for the size of population of practical interest. In other words the optimal protocol is practically nonrealizable for large population of users. The following results of Bassalygo and Pinsker [4] show that there is trade-off between throughput and delay, i.e. there exist protocol sequences which might not achieve the maximum throughput but may have much shorter length.

Theorem 1. [Bassalygo, Pinsker]
There exists protocol set (T,M,s) of period N satisfying

$$N_l \lesssim N \lesssim N_u \qquad (9)$$

where

$$N_l = \max (sM/C_M ; 0.5\, sM + 0.5\, M \log T) \qquad (10)$$

$$N_u = M/C_M \, [s + M \log T] \, [1 + \alpha + (\alpha^2 + 2\alpha)^{1/2}] \qquad (11)$$

$$\alpha = [\,[M - C_M) \ln T\,]/[s + (M-1)\log T\,] \leq 1.5 \ln 2 \qquad (12)$$

and C_M is given by (7), log denotes logarithm of base two.

It is worthwhile to mention that by (1), (7) and (10) it follows that the optimal protocol sequences achieve the Bassalygo-Pinsker lower bound.

Recently Nguyen, Gyorfi and Messey [5-6] have presented several constructions of protocol sequences based on cyclic sets and cyclic constant weight codes [7]. These protocols are not optimal, i.e. they do not achieve the maximum throughput (6), but have very short period. For example, the protocol (T,2,2) based on optimal cyclic binary code of weight 3 and distance 4 exists for all prime N of form

$$N = 6T + 1 \quad (N \text{ is prime}) \qquad (13)$$

$$M = 2 \; ; \; s = 2$$

The protocol set based on Reed-Solomon codes has the following parameters

$$N = p^2 - p \qquad (14)$$

$$T = p^{k-2} \qquad (15)$$

$$M \geq \min\,[(p-2)/(k-1) \; ; \; (p-1-s)/(k-1) + 1\,] \qquad (16)$$

where $k \geq 3$ is an integer and p is a prime number.

In the next section we present an other construction of protocol sequences based on the theory of cyclotomy [8], so we call them cyclotomic protocol sequences.

CYCLOTOMIC PROTOCOL SEQUENCES

Let $N = wT + 1$ be an odd prime number and let β be a fixed primitive root of N. An integer R is said to belong to the index class i, or to the coset C_i, with respect to β if there exists an integer x such that

$$R = \beta^{xT+i} \pmod N$$

Thus, the coset C_i consists of w distinct numbers

$$\beta^i, \beta^{T+i}, \ldots, \beta^{(w-1)T+i} \pmod N$$

It is clear that

$$C_o = (1, \beta^T, \ldots, \beta^{(w-1)T}) \qquad (17)$$

is a subgroup of the multiplicative group of GF(N). Thus, the index classes C_i, $0 \leq i < T$, are just the cosets of the multiplicative group of GF(N) with respect to the subgroup C_o.

The cyclic number (i,m) counts the number of times $R + 1$ belongs to the index class C_m, when R belongs to C_i [8p.120]. In other words (i,m) is the number of

solutions x, y of the congruence

$$1 + \beta^{xT+i} = \beta^{yT+m} \pmod N \qquad (18)$$

where x and y are chosen from 0, 1, ..., w-1.

Let

$$C_i + k = (k+\beta^i, k+\beta^{T+i}, \ldots, k+\beta^{(w-1)T+i}) \qquad (19)$$

be a shift of C_i with k steps, where addition is taken mod N. The set kC_i is defined in similar ways and is equal to C_{i+t} where $k = \beta^t$. The cyclotomic number can be expressed as follows.

$$(i,m) = |(1+C_i) \cap C_m| \qquad (20)$$

where |A| denotes the number of elements of the set A.

The following elementary results on cyclotomic numbers are well known [8p.121]

Theorem 2.

Let N = wT + 1 be an odd prime number, C_o be defined by (17) and C_i be the cosets of the multiplicative group of GF(N) with respect to subgroup C_o. Let the cyclotomic numbers (i,m) be defined by (20). Then

$$(i,m) = (i',m') \text{ when } i = i' \pmod T,\ m = m' \pmod T \qquad (21)$$

$$(i,m) = (T-i,m-i) = \begin{cases} (m,i) & w \text{ even} \\ (m+T/2,i+T/2) & w \text{ odd} \end{cases} \qquad (22)$$

$$\sum_{m=0}^{T-1}(i,m) = \begin{cases} w-1 & \text{if } i = 0 \quad w \text{ even} \\ w-1 & \text{if } i = T/2 \quad w \text{ odd} \\ w & \text{otherwise} \end{cases} \qquad (23)$$

Let c be the maximum cyclotomic number for a given N = wT + 1, i.e.

$$c = \max_{\substack{0 \le i < T \\ 0 \le m < T}} (i,m) \qquad (24)$$

construction is given by the following theorem.

Theorem 3.

Given an odd prime number N = wT + 1. Let C_i, $0 \le i < T$, be index sets and c be the maximum cyclotomic number defined above. We construct T binary sequences of length N as follows. The k-th component of X_i is 1 if $k \in C_i$

and it is 0 otherwise. Then these T binary sequences constitute a (T,M,s) protocol set with

$$M = \min \{ \lfloor (w-1)/c \rfloor ; \lfloor (w-s)/c \rfloor + 1 \} \qquad (25)$$

where $\lfloor x \rfloor$ denotes the integer part of x.

Proof. By construction we have T sequences and since each C_i has exactly w elements, X_i has weight w. By (3) and (6) it is sufficient to show that

$(X_i, S^k X_j) \leq c$ for all i, j and $k \neq 0$. But, this follows from the fact that

$$(X_i, S^k X_j) = |(C_i \cap C_j + k)| = |(\beta^t C_{i-t} \cap \beta^t C_{j-t} + \beta^t)| = (i-t, j-t) \leq c$$

where t is such that $k = \beta^t$. If $k = 0$ then $(X_i, X_j) = 0$ if $i \neq j$. This completes the proof.

Remark The cyclotomic numbers depend on N, T and β, however, the maximum cyclotomic number c does not depend on particular primitive element of N. To our knowledge there is no closed formula for c as function of N and T, even good bound for c in terms of N and T is not known. There is general procedure to determine cyclotomic numbers for the case of T = 2-6, 8, 10, 12, 14, 16, 18, 20 [8], but c cannot be determined easily in these cases too. Nevertheless it is easy to write a program to determine c for each given N and T. For example, a BASIC program has been written on a personal computer and running time to calculate all cyclotomic numbers and determining c for N = 3943 and T = 27 is less than two minutes.

The following bounds on c are nearly trivials and the proof is omitted.

Lemma For any given prime N and T such that N = wT + 1

$$\max \{1; \lfloor w/T \rfloor \} \leq c \leq w - 1 \qquad (26)$$

By computer experience we believe that the following conjecture holds

Conjecture

Given a prime N = wT + 1. If $w \geq 3$ than

$$\max \{2 ; \lfloor w/T \rfloor \} \leq c$$

NUMERICAL RESULTS AND DISCUSSIONS

A program has been written on a personal computer to calculate c and parameters of cyclotomic protocol sequences. Given N and N = wT + 1 decomposition, the program first determines c and then M and s by

$$M = \lfloor (w - 1)/c \rfloor \quad ; \quad s = w - c(M - 1) \qquad (27)$$

Furthermore the program calculates the lower and upper bounds of Bassalygo and Pinsker from (10) and (11). For all N < 1000 and N = wT + 1 decompositions the average value of N_l/N is 0.177 that of N_u/N is 1.634. Of course, $N \geq N_l$ in all cases and $N < N_u$ in more than 80% of sequences investigated. Obviously, the strategy (27) for choosing parameters of sequence is not optimal if one wants to maximize

the sum-rate. This can be done by choosing M and s satisfying (25) and maximizing the sM product. In the second stage of our investigations the program determines c then M and s such that Ms is maximized. There are protocol sequences of the same parameters T and M but having different length N. In general, in such cases as N increases s, hence R, also increases but no regularity has been observed for all N (1000. In the case of fixed T and M if a given sequence has parameter s less than that of a sequence having the same T and M but with shorter length N then the sequence of longer period is deleted. The results obtained have been tabulated and are given in the Appendix for N (1000.

From these results it is clear that for fixed T and M as s increases the performance of sequences, measured in sum-rate, increases as predicted by Bassalygo and Pinsker (10), (11).

It is worth to mention that the protocol sequences of [5-6] cannot have M) T/2, the cyclotomic protocol sequence can have M) T/2. It is also interesting to note that the protocol set based on optimal binary cyclic code of weight 3 and distance 3 (13) has parameter M = 2, s = 2 for all prime N = 6T + 1. For all N (1000 we observe that for such cases c = 2 and the cyclotomic protocol sequence has parameter M = 2, w = 6, s = 4, i.e. sum-rate of cyclotomic protocol sequence is twice that of protocol set (13) in all cases investigated. We could not prove that c = 2 for all N = 6T + 1 and T $\geq$ 3 (if T = 2 then by (23) c must be 3), we believe that c = 2 ,i.e. we conjecture that cyclotomic protocol set for N = 6T + 1 is always better than the protocol set of (13).

REFERENCES

J.L. Massey "The Capacity of the Collision Channel without Feedback", Abstract of Papers. IEEE Int. Symp. on Inf. Theory, 1982, p. 101.

J.L.Massey, P. Mathys " M-user Collision Channel without Feedback" Reprint ETH Zurich, Switzerland, 1983.

B.S. Tsybakov,N.B. Likhanov "Packet Communication on a Channel without Feedback", Problems of Inf. Transmission, vol.XIX, N 2,pp.69-84,1983

L.A. Bassalygo, M.S. Pinsker " Restricted Multiple-Access of a Nonsynchronous Channel" Problems of Inf.Transmission, vol.XIX, N 4, pp.92-96, 1983

D.A. Nguyen, L, Gyorfi, J.L. Massey " Some Construction of Protocol Sequences for Collision Channel and a Class of Optimal Cyclic Constant Weight Codes" IEEE Int. Symp. on Information Theory, Brighton 1985.

D.A. Nguyen, L. Gyorfi, J.L. Massey "Performances of Protocol Sets for Collision Channel without Feedback" to be published.

D.A. Nguyen, L. Gyorfi, J.L. Massey " Some Construction of Cyclic Sets and Constant Weight Binary Cyclic Codes " Submitted to IEEE Trans.Inf. Theory.

L.D. Baumert "Cyclic Difference Sets", Springer Verlag 1971.

M. Hall Jr. " Combinatorical Theory", Blaisdell Publ. Comp. 1967.

L. Kleinrock " Queueing Systems" vol. II. John Wiley & Sons 1976.

W.H.Kautz, R.C. Singleton "Nonrandom Binary Superimposed Codes", IEEE Trans. Inf. Theory, IT-10, pp. 363-377, 1964.

A.G. Dyachkov, V.V. Rykov " A Survey of Superimposed Code Theory" , Problems of Control and Inform. Theory, vol.12, N 4, pp.229-244, 1983.

APPENDIX

T	M	N	s	R	c	T	M	N	s	R	c
6	2	31	4	.2581	2	8	4	449	29	.2583	9
	3	79	5	.1899	4		5	353	12	.1700	8
		97	8	.2474	4			673	28	.2080	14
		193	16	.2487	8		6	449	11	.1469	9
		223	19	.2561	9	9	2	73	6	.1644	2
	4	223	10	.1794	9		3	73	4	.1644	2
		271	15	.2214	10			163	10	.1840	4
		439	25	.2278	16			379	26	.2060	8
		463	29	.2505	16			487	34	.2095	10
		619	43	.2940	20			523	38	.2180	10
	5	619	23	.1858	20		4	163	6	.1472	4
		769	32	.2080	24			433	24	.2217	8
7	2	43	4	.1860	2			541	30	.2218	10
	3	71	4	.1690	3			739	43	.2327	13
		127	8	.1890	5			811	48	.2370	14
		337	28	.2493	10			883	53	.2400	15
		491	42	.2566	14		5	379	10	.1320	8
		617	56	.2729	16			433	24	.2217	8
		631	58	.2758	16		6	739	17	.1380	13
		953	88	.2770	24			811	20	.1480	14
	4	337	18	.2137	10			883	23	.1563	15
		491	28	.2281	14	10	2	61	4	.1311	2
	5	617	24	.1945	16			71	5	.1408	2
		631	26	.2060	16			151	11	.1457	4
		953	40	.2099	24		3	71	3	.1272	2
8	2	41	3	.1463	2			151	7	.1391	4
		73	6	.1690	3			211	11	.1564	5
	3	97	6	.1856	3			281	18	.1922	5
		137	9	.1971	4			401	26	.1945	7
		253	17	.2189	6			521	34	.1958	9
		241	18	.2241	6			701	46	.1970	12
		353	28	.2380	8		4	211	6	.1137	5
		433	34	.2356	10			461	25	.2169	7
		521	41	.2361	12			821	46	.2241	12
		809	65	.2410	18			991	57	.2300	14
		857	71	.2485	18		5	281	8	.1423	5
	4	137	5	.1460	4			401	12	.1496	7
		233	11	.1888	6			461	18	.1952	7
		241	12	.1992	6		6	461	11	.1432	7
		401	20	.1995	10			821	22	.1608	12

T	M	N	s	R	c	T	M	N	s	R	c
10	7	991	15	.1060	14	14	6	547	9	.0987	6
	2	67	4	.1194	2			743	13	.1050	8
11	3	89	4	.1348	2			953	18	.1133	10
		199	10	.1508	4			967	19	.1180	10
		419	24	.1718	7		7	631	9	.0998	6
	4	199	6	.1206	4	15	3	151	6	.1192	2
		683	32	.1874	10			241	10	.1245	3
		727	36	.1981	10			271	12	.1328	3
	5	419	10	.1193	7		4	151	4	.1060	2
	6	683	12	.1054	10			271	9	.1328	3
		727	16	.1321	10			541	18	.1331	6
12	2	61	3	.0984	2			571	20	.1401	6
	3	97	6	.1237	2			601	22	.1464	6
		193	10	.1554	3			631	24	.1522	6
		277	15	.1625	4		5	241	4	.0830	3
	4	193	7	.1444	3			271	6	.1107	3
		433	18	.1663	6			541	12	.1109	6
		661	31	.1876	8		6	571	8	.0841	6
	5	193	4	.1036	3			601	10	.0998	6
		277	7	.1264	4			631	12	.1141	6
		433	12	.1386	6	16	2	97	4	.0825	2
		577	16	.1387	8			113	5	.0885	2
	6	661	15	.1362	8		3	113	3	.0796	2
13	2	79	4	.1013	2			193	6	.0933	3
	3	131	6	.1374	2			257	8	.0934	4
	4	131	4	.1221	2			337	13	.1157	4
		313	12	.1534	4			353	14	.1190	4
		521	22	.1690	6		4	401	4	.0998	5
		677	31	.1832	7			577	21	.1456	5
	5	313	8	.1278	4		5	337	5	.0742	4
	6	521	10	.1152	6			353	6	.0850	4
	7	677	10	.1034	7		6	577	7	.0728	5
14	2	113	6	.1062	2			641	10	.0936	6
	3	113	4	.1062	2	16	6	881	15	.1022	8
		281	12	.1281	4		7	577	6	.0728	5
		421	18	.1283	6			929	10	.0754	8
		491	23	.1405	6	17	2	103	4	.0777	2
		547	27	.1481	10			239	10	.0837	4
	4	281	8	.1139	4		3	137	4	.0876	2
		421	12	.1140	6			307	10	.0977	4
		547	21	.1536	6			409	16	.1174	4
		631	27	.1712	6		4	307	6	.0782	4
	5	491	11	.1120	6			409	12	.1174	4
		673	24	.1783	6			443	14	.1264	4

T	M	N	s	R	c	T	M	N	s	R	c
17	5	409	8	.0978	4	20	6	541	7	.0776	4
		443	10	.1128	4			701	10	.0856	5
		953	28	.1469	7		7	941	11	.0818	6
	6	443	6	.0813	4	21	2	127	4	.0630	2
	7	953	14	.1628	7		3	211	6	.0853	2
18	2	109	4	.0734	2			421	13	.0926	4
		127	5	.0787	2		4	211	4	.0758	2
		199	8	.0804	3			421	11	.1045	4
		271	11	.0812	4			883	24	.1087	6
	3	127	3	.0709	2			967	28	.1158	6
		163	5	.0920	2		5	463	6	.0648	6
		433	16	.1122	4		6	421	5	.0713	4
		853	47	.1653	12			883	12	.0815	6
	4	163	3	.0736	2		7	967	10	.0724	6
		433	12	.1109	4	22	2	199	7	.0700	2
		487	15	.1230	4		3	199	5	.0754	2
		523	17	.1300	4			353	10	.0850	3
		853	35	.1640	12		4	199	4	.0603	2
	5	433	8	.0924	4			353	7	.0793	3
		853	23	.1350	4			617	16	.1037	4
	6	487	7	.0802	4			683	19	.1113	4
		523	9	.1032	4		5	353	4	.0567	3
	7	523	5	.0669	4			727	17	.1169	3
		919	9	.0686	7		6	617	8	.0778	4
19	3	191	6	.0942	2		7	683	7	.0717	4
		229	8	.1048	2		8	727	5	.0550	4
	4	191	4	.0838	2	23	2	139	4	.0576	2
		229	6	.1048	2		4	277	6	.0860	2
		457	12	.1050	4		5	461	8	.0868	3
		571	18	.1261	4		6	461	5	.0651	3
	5	229	4	.0873	2			691	10	.0868	4
		457	8	.0875	4		7	691	6	.0608	4
	6	571	10	.1051	4	24	2	193	6	.0622	2
	7	571	6	.0736	4		3	193	4	.0622	2
20	2	101	3	.0594	2			241	6	.0747	2
	3	281	8	.0854	3		4	241	4	.0664	2
		421	13	.0926	4			433	9	.0831	3
		461	15	.0976	4			601	13	.0865	4
	4	281	5	.0712	3		5	433	6	.0693	3
		541	15	.1110	4			769	16	.1040	4
		701	20	.1141	5		6	601	5	.0499	4
		941	29	.1233	6		7	769	8	.0728	4
	5	421	5	.0594	4	25	2	151	4	.0530	2
		461	7	.0760	4		3	251	6	.0717	2

M	N	s	R	c	T	M	N	s	R	c
3	401	10	.0748	3	30	3	241	4	.0498	2
4	251	4	.0637	2			421	10	.0712	2
	751	15	.0799	5		4	421	8	.0762	2
5	401	4	.0499	3		5	421	6	.0712	2
	601	12	.0980	3			991	17	.0858	4
7	601	6	.0700	3		6	421	4	.0570	2
2	131	3	.0458	2		8	991	5	.0404	4
	157	4	.0510	2	31	3	373	8	.0643	2
3	313	8	.0763	2		4	373	6	.0643	2
4	313	6	.0767	2		5	373	4	.0536	2
	521	11	.0845	3	32	2	193	4	.0415	2
5	313	4	.0639	2		3	673	13	.0580	4
	937	20	.1067	4		4	449	8	.0713	2
6	521	5	.0576	3			929	17	.0732	4
8	937	8	.0683	4		5	577	6	.0520	3
2	163	4	.0491	2		6	449	4	.0535	2
3	379	8	.0633	3		7	929	5	.0377	4
	433	10	.0693	3	33	2	192	4	.0402	2
	487	12	.0740	3		4	331	4	.0483	2
4	379	5	.0528	3			661	11	.0666	3
	487	9	.0739	3			727	13	.0715	3
	757	16	.0845	4		6	661	5	.0454	3
	811	18	.0888	4		7	727	4	.0385	3
5	433	4	.0462	3			991	6	.0424	4
	487	6	.0616	3	34	4	307	3	.0391	3
6	757	8	.0634	4			443	7	.0632	2
7	811	6	.0518	4		5	443	5	.0564	2
3	281	6	.0641	2		6	443	3	.0406	2
	449	10	.0668	3			919	7	.0457	4
	617	14	.0681	4			953	8	.0504	4
4	281	4	.0569	2	35	2	211	4	.0379	2
	421	6	.0570	3		3	281	4	.0427	2
	757	15	.0793	4		4	701	11	.0628	3
5	449	4	.0445	3		5	721	5	.0428	3
	617	6	.0486	4			911	14	.0768	3
6	757	7	.0555	4		8	911	5	.0446	3
3	233	4	.0515	2	36	3	433	8	.0462	2
	349	8	.0688	2			577	10	.0520	3
4	349	6	.0688	2		4	433	6	.0554	2
5	349	4	.0573	2			757	12	.0634	3
	523	6	.0574	3		5	433	4	.0462	2
2	151	3	.0397	2			937	14	.0747	3
	181	4	.0442	2		6	757	6	.0476	3
	241	6	.0498	2		8	937	5	.0427	3

T	M	N	s	R	c	T	M	N	s	R	c
37	2	223	4	.0359	2	51	5	613	4	.0326	2
	3	593	10	.0506	3		8	919	4	.0348	2
	5	593	4	.0337	3	52	2	313	4	.0256	2
38	2	191	3	.0314	2		4	521	4	.0307	2
		229	4	.0349	2		6	677	3	.0266	2
	4	419	5	.0478	2	54	3	433	4	.0277	2
	5	419	3	.0358	2		4	487	3	.0246	2
		457	4	.0438	2			541	4	.0296	2
39	3	313	4	.0383	2		7	811	3	.0259	2
	5	859	6	.0349	4	55	5	661	4	.0303	2
40	2	241	4	.0332	2		8	991	4	.0323	2
	3	281	3	.0320	2	56	2	281	3	.0214	2
	4	401	4	.0399	2			337	4	.0237	2
		601	9	.0599	2		3	449	4	.0267	2
	5	601	7	.0582	2		4	673	6	.0357	2
	7	601	3	.0349	2		5	673	6	.0297	2
41	5	739	6	.0406	3	57	3	457	4	.0263	2
	6	821	5	.0365	3		4	571	4	.0280	2
42	3	337	4	.0357	2	58	2	349	4	.0229	2
	4	379	3	.0317	2		4	523	3	.0229	2
		421	4	.0380	2		7	929	4	.0301	2
	5	463	3	.0324	2	60	3	421	3	.0214	2
	6	547	3	.0329	2		4	541	3	.0222	2
		673	6	.0536	2			601	4	.0266	2
	7	673	4	.0416	2	61	2	367	4	.0218	2
44	6	617	4	.0390	2	63	2	379	4	.0211	2
	7	661	3	.0318	2		4	631	4	.0254	2
45	2	271	4	.0295	2		5	757	4	.0264	2
	6	631	4	.0507	2	64	3	449	3	.0200	2
46	2	277	4	.0289	2		4	577	3	.0208	2
	5	829	10	.0603	2			641	4	.0250	2
	8	829	4	.0386	2		5	769	4	.0260	2
47	4	659	5	.0303	3	66	3	463	3	.0194	2
	6	941	5	.0319	3		4	661	4	.0242	2
48	3	337	3	.0267	2	66	5	727	3	.0263	2
	4	433	3	.0277	2		7	991	3	.0212	2
	5	577	4	.0346	2	68	2	409	4	.0196	2
	6	673	4	.0357	2		6	953	4	.0252	2
49	4	491	4	.0326	2	69	4	691	4	.0232	2
	8	883	4	.0362	2		6	967	4	.0248	2
50	3	401	4	.0299	2	70	3	491	3	.0183	2
	4	701	5	.0285	3		4	701	4	.0228	2
		751	6	.0320	3		6	911	3	.0198	2
51	3	409	4	.0293	2	71	5	853	4	.0234	2

T	M	N	s	R	c	T	M	N	s	R	c
72	2	433	4	.0185	2	85	5	947	3	.0158	2
	3	577	4	.0208	2	87	2	523	4	.0153	2
	6	937	3	.0192	2	88	3	617	3	.0146	2
73	2	439	4	.0182	2		4	881	4	.0182	2
74	3	593	4	.0202	2	90	3	631	3	.0143	2
75	4	751	4	.0213	2	91	4	911	4	.0176	2
76	4	761	4	.0210	2	92	2	461	3	.0130	2
77	2	463	4	.0173	2		4	829	3	.0145	2
	3	617	4	.0195	2	94	3	659	3	.0137	2
78	3	547	3	.0165	2		4	941	4	.0170	2
	5	859	3	.0175	2	96	2	577	4	.0138	2
		937	4	.0213	2		3	673	3	.0134	2
80	3	641	4	.0187	2	97	4	971	4	.0165	2
	5	881	3	.0170	2	98	4	883	3	.0136	2
81	2	487	4	.0164	2	99	4	991	4	.0161	2
	4	811	4	.0197	2	100	3	701	3	.0128	2
82	4	739	3	.0162	2	101	3	809	4	.0148	2
83	2	499	4	.0160	2	102	4	919	3	.0131	2
	5	997	4	.0200	2	103	2	619	4	.0129	2
84	2	421	3	.0142	2	104	2	521	3	.0115	2
	3	673	4	.0178	2	106	3	743	3	.0121	2
	4	757	3	.0159	2	107	2	643	4	.0124	2
85	2	431	3	.0139	2		3	857	4	.0140	2

A DISTRIBUTED PROCESS CONTROL DATA BASE MANAGEMENT SYSTEM

G. Juhász, K. Kovács, I. Sári

Central Research Institute for Physics
Hungarian Academy of Sciences
H-1525 Budapest, P.O.Box 49, Hungary

Distributed process control systems require uniform access to data in any node of a network. Application of the PCDB system offers powerful data structures and file organization capabilities for local and remote disc resident and core resident data. It simplifies program development and data maintenance. PCDB is implemented on a network of DEC PDP-11 compatible minicomputers and micros.

INTRODUCTION

The Central Research Institute for Physics of the Hungarian Academy of Sciences has been planning and realizing real-time systems for laboratory and industrial data acquisition and control for many years.

As the requirements and specifications of program systems became more complex, and the number of people working on a project increased, it became clear that a method for managing data, in a well defined and uniform way, was needed (Cardenas (1979), Choy (1979), Koller (1979)).

Requirements for a large industrial or laboratory monitoring and control system led to a hierarchically distributed computer network in which three main levels can be distinguished (Horvai (1982)):

Level 1 Overall Plant Control
Level 2 Supervisory Control of a Subsystem

Level 3 Direct Digital Control
(See Fig. 1.)

The 1st and 2nd level may include for example PDP-11 compatible minicomputers supplied with conventional peripherials while the 3rd level may contain CAMAC real-time peripherals controlled by several INTEL 8080 based Intelligent Crate Controllers (ICCs). ICCs can be replaced by micro TPA-11s.

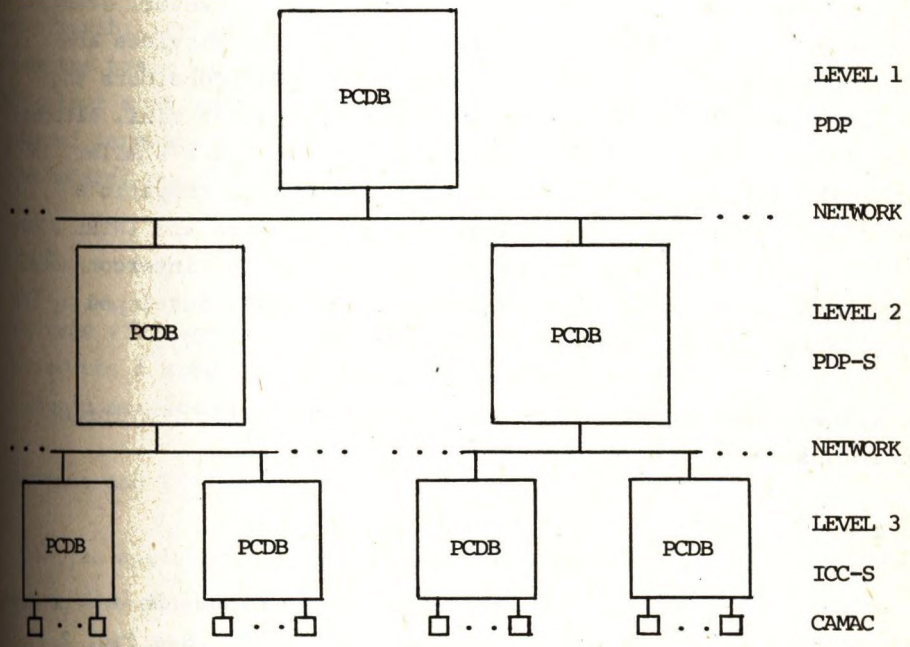


FIG. 1.

It was evident that only a highly developed distributed data base management system could satisfy the special requirements of such a system (Aho (1982), Champine (1980)).

CONCEPTS OF PCDB SYSTEM

Distributed data

PCDB (Process Control Data Base Management System) allows the user to perform various operations in a distributed process control data base, where the data are distributed amongst subsets (called local bases) stored on various sites (mini and micro computers), geographically dispersed but interconnected by a high-speed communication network.

The minis are hierarchically on the same level. However, every micro computer is supervised by a mini. The PCDB services are available for the mini computer users only. PCDB considers the micros as intelligent peripheral devices of a given mini, although the mini-mini and mini-micro network connections are similar. In the present realization mini computers are PDP-11 compatible Hungarian made TPA mini computers, micro computers are INTEL 8080-based Intelligent (CAMAC) Crate Controllers (ICCs) interconnected by the LOCal High-speed NEtwork SyStem (LOCHNESS, developed by the Central Research Institute for Physics, Arató (1983)).

PCDB allows that LOCHNESS be replaced by DECnet for PDP interconnections.

Data Independence

One of the main tasks of the data base is to ensure the independence of the data from the programs, which handle them. (See Fig. 2.)

Physical data independence is the ability to isolate application programs from changes in the physical organization of the data used, e.g. changes in the location of data.

Logical data independence usually arises due to change. Changes often occur in practice due to unforeseen future requirements in industrial systems. In the field of laboratory application this requirement is even more important. Data base technology tries to provide as much data independence as possible. There are many degrees of data independence but complete independence can never really be achived since an application program cannot be defended

from all possible changes.

Data Shareability and Consistency

The goal is to enable independent programs to share an integrated data base containing all the data needed and to eliminate as much as possible the need to store redundant data. Control over intereffects among the user programs must be provided; a given part of the data may only be modified by one program at any one time.

Access Flexibility

Access flexibility is the ability to access any part of the data base on the basis of any access keys and logical qualification, via input/output statements issued from programs written in conventional procedural languages (FORTRAN, BASIC etc.) or via a high-level non-procedural query language.

Recovery

In the case of transient disc, memory or network system errors PCDB executes a special recovery procedure to maintain a non-interrupted service.

Performance and Efficiency

In real-time applications high-speed data access and short response time are vital requirements.

Self-documentation

As a program independent data description is an integral part of the data base, it is possible to get up-to-date information at any time on the current data organization.

Data Model

Data Base technology has introduced powerful logical data base structures or models. They are classified generally into three categories: hierarchic or tree model, network model and relational

model.

The fundamental concepts of PCDB are closest to the relational data model. The relational approach was adopted since it allows data organization in a rather simple way. It conforms to the character of data acquisition systems. Data can be represented in tabular form, such tables are called relations. Rows in tables are called records, columns are referred to as attributes. All elements of a column have the same type and length. Each record is of the same length. Rows are identified by the record sequence number (RSQN), and each column can be referred to by the attribute name. E.q. different records can represent different sensors; a record contains an identifier, the current value, status, dimensioning constants, etc. (See Table 1.)

Record Sequence Number	Status	Value	Add. Const.	Multipl. Const.
5	01101	6.344	41	7.6
6	01100	9.731	81	3.7
7	01101	4.625	81	3.7

TABLE 1.

Records can be located not only by the RSQN - other attributes may be defined as keys too in some cases (e.q. the identifier). By specifying a key attribute name and a value (called the key value) records may be selected from a relation.

Data Description

The structure of the data base is described in a central data descriptor (the so-called schema) file. This file contains one descriptor record for each relation. The schema record contains the name of the relation, the attribute names, attribute types (see Table 2.), the optional key indicator, the position and size of attributes in the record. Moreover it contains the size of the

relation and the organization and specification of the corresponding data file.

Type	Length	Remark
Integer	1,2 or 4 bytes	I/O format: decimal, octal or hexadecimal; vectors may be defined
Floating point	4 or 8 bytes	vectors may be defined
String	1-500 bytes	ASCII characters
Bit-string	1-15 bits	bitgroup in 1,2 or 4 byte field

TABLE 2.

Resource handling

Before making use of a relation the user program has first to reserve (open) it. After reservation the relation is referenced by a logical number. Two reservation modes are available: protected and shared. Protected mode reservation grants program write permission (including read and delete operations) and ensures that the data stay consistent since no simultaneous protected reservations are allowed for a relation.

With a protected reservation the user program can choose one of the following cases:

1. to queue the request automatically by PCDB if the relation has been already reserved in protected mode
2. to get a simple error code immediately

For shared mode (read-only) reservation requests there is no restriction i.e. several programs can reserve a relation in this mode.

To prevent dead-lock conditions a simple rule is followed: a program may specify its reservation request to be queued only if it has no reservations in effect.

PCDB also ensures that at program exit or abort every relation used by the program be released automatically.

Data Access

In process control applications the response time may be critical so PCDB supports not only disc resident data handling but relations can be stored also in the main memory of the TPA-11 or in an ICC memory. (See Fig. 2.)

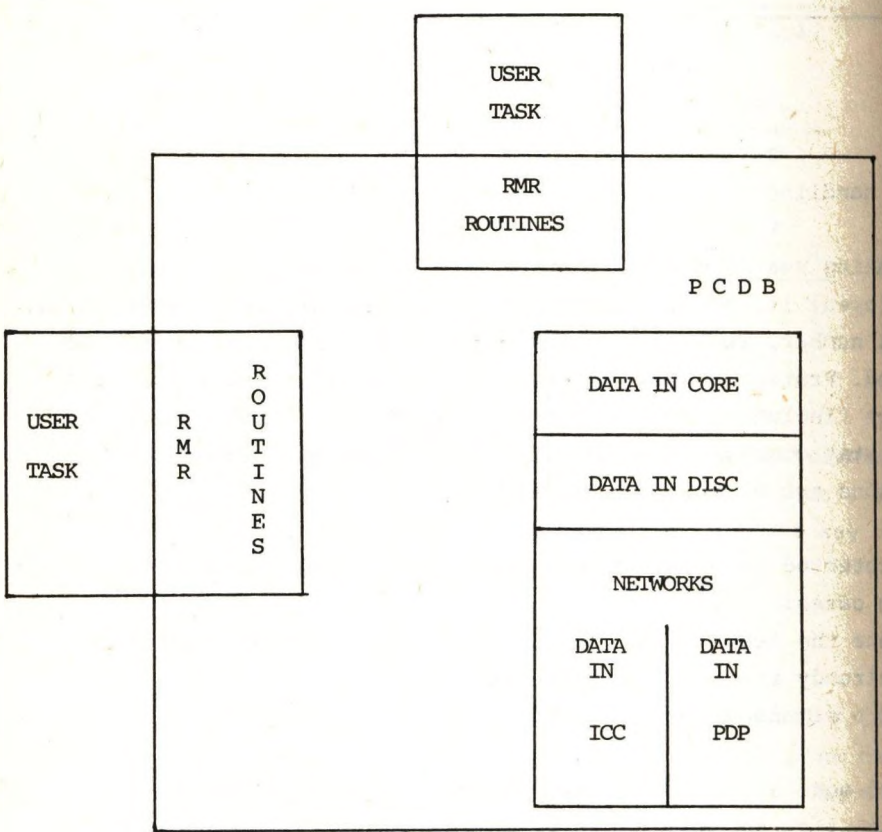


FIG. 2.

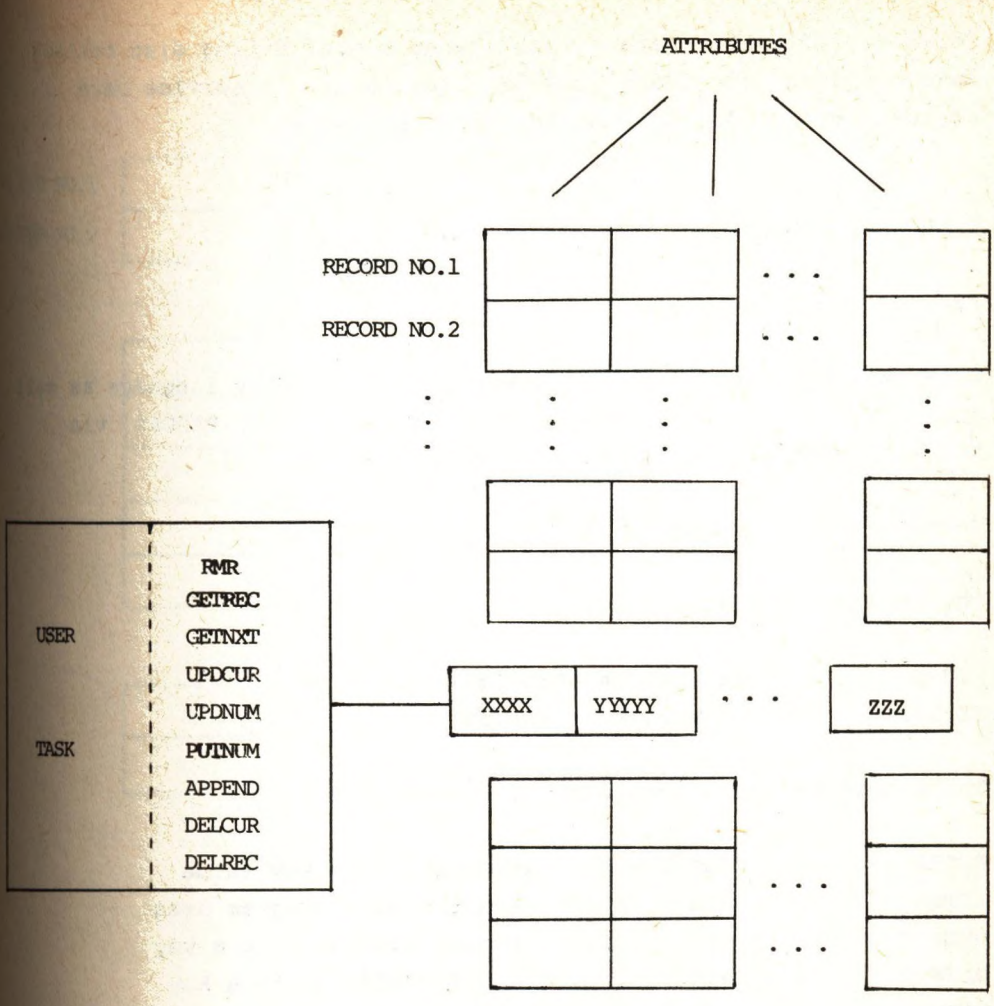


FIG. 3.

The column manipulation routines (See Fig. 4.):

SELECT	select an RSQN array of records according to a key value
GETITV	get attribute values of records by an RSQN interval
GETLST	get attribute values of records by an RSQN array
UPDITV	update attribute values of records by an RSQN interval
UPDLST	update attribute values of records by an RSQN array

There are different data access channels e.g. during disc resident data handling operations PCDB can simultaneously service core resident and remote data manipulation requests.

PCDB USER INTERFACE

Program Level Interface

PCDB can interface to programs written in assembly language as well as in high level languages (FORTRAN, BASIC-PLUS-2, PASCAL) via Relation Manipulation Routines (RMR). (See Fig. 2.)

There are four groups of the RMR-s.

The resource handling routines:

RESREL	reserve relation(s)
RELREL	release relation(s)

The record manipulation routines (see Fig. 3.):

FNDREC	find a record according to a key value
GETREC	read a record into the user program area
GETFIR	read the first record according to a key
GETNXT	read the next record according to a key
UPDCUR	update current record
UPDNUM	update record of a given sequence number
PUTNUM	write a record of a given sequence number
APPEND	extend the relation with a new record
DELCUR	delete current record
DELREC	find a record according to a key value and delete it

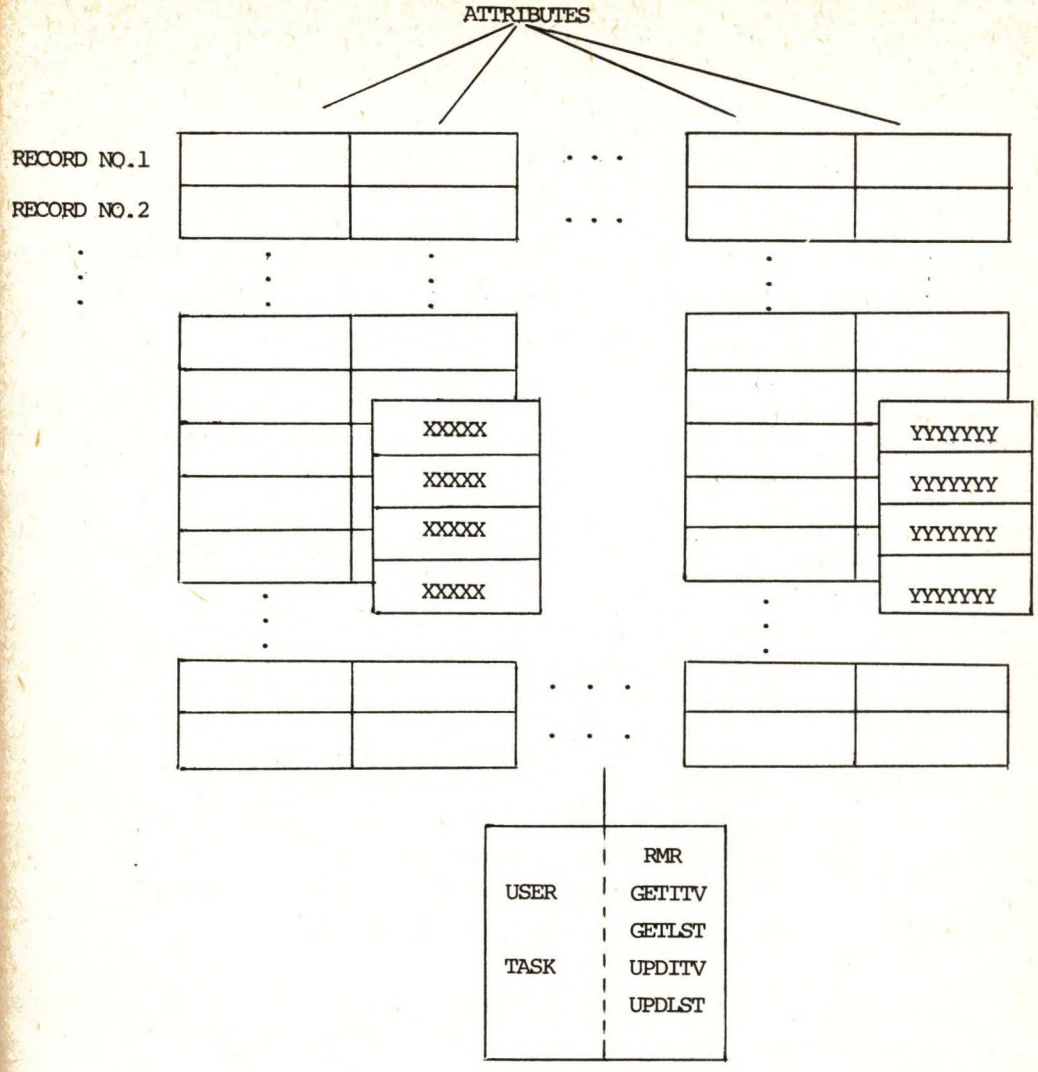


FIG. 4.

The field manipulation routines:

FETCH	move an attribute value from the current record to a user location
STORE	move value from a user location to the current record

Every record and column operation can be issued in either synchronous or asynchronous mode. In asynchronous mode the task can run during the PCDB operation.

Utilities

The schema processor generates the internal form of the schema from the schema's display format as well as the indirect command files needed for the generation of the data files.

The relation editor is an interactive program to store, examine and modify data. Using the editor one can also make a full or selective listing of a given relation.

There are other utilities to support the data saving and loading operation.

HARDWARE, SOFTWARE REQUIREMENTS

In the present realization PCDB can be used on a computer network containing DEC PDP-11 or compatible mini computers with at least 248 Kbytes main memory and INTEL 8080 based micros (ICCs) interconnected by the Local High-Speed Network System (LOCHNESS, developed by the Central Research Institute for Physics).

The PDP-11 part of the PCDB runs under the RSX-11M/M-PLUS operating system and permanently occupies 9-40 Kbytes of core memory. PCDB components occupy about 1 Mbyte of disc storage space.

PCDB allows that LOCHNESS be replaced by DECnet for PDP interconnections.

APPLICATIONS

PCDB has been in use in the following Hungarian industrial systems the National Telemechanic System for Natural Gas Pipeline Control, the Control System for Budapest's Waterworks and the Northern Hungary Power Distribution Control System.

PCDB can be used in the laboratory environment as well: a nuclear fusion test system is to use PCDB.

SUMMARY

In process control applications the response time may be critical so PCDB supports core resident data handling.

PCDB has an efficient resource handling mechanism to ensure data consistency and support continuous operation without dead-lock.

A number of PCDB utilities help the user to create, save, restore, examine and modify data in an interactive way as well as from programs.

PCDB is a component consisting of a set of process control oriented software tools developed for mini computers. E.g. a sophisticated operator communication facility can use PCDB services (DICOM).

PCDB can be easily tailored to meet the requirements of the project and the configuration available.

REFERENCES

[1] Aho, V.Y., J.E. Hopcroft and J.D. Ullmann (1982). Data Structures and Algorithms. Addison-Wesley, Menlo Park.

[2] Arató, A., I. Sarkadi-Nagy, J. Sulyán and F. Telbisz (1983). A Local Area Network Architecture Tailored to Laboratory Environments. Interfaces in Computing, 1 (1983) 255-262.

[3] Cardenas, A.F. (1979). Data Base Management Systems. North-Holland, Amsterdam.

[4] Champine, G.A. (1980). Distributed Computer Systems. Allyn and Bacon, Boston.

[5] Choy, J.H. and J.A. Wade (1979). Data Base Management System for the MFTF. Proceedings Engineering Problems of Fusion Research (IEEE).

[6] Horvai, M. (1982). Realization of Distributed Process Control Systems. Microcomputer Applications, Vol. 1, No. 1, 27-30.

[7] Koller, H. and K. Fruhauf (1979). PRIMO Data Base Management System. Brown Boveri Rev., 3, 204-209.

SERVICEABILITY ANALYSIS OF POLISH UNIVERSITY PILOTE COMPUTER NETWORK - MSK

Marian M. Budka, Józef J. Grzegorczyk

Institute of Computer Science
Silesian Technical University
ul. Pstrowskiego 16
44-100 Gliwice
Poland

The paper focuses on the reliability of computer network and availability of resources from users' point of view. The measure of computer network reliability and availability, defined as a serviceability is the basis of the consideration on how user's jobs are performed in computer network. The model of computer network considered in the paper is the probabilistic graph with the sets of jobs demanded by every graph node (user, terminal, HOST-computer, TIP, switching node) and the sets of resources accessible in every node. The results of the serviceability analysis of MSK - Pilote Computer Network are presented in the paper. All the examples are implementation based.

INTRODUCTION

The main parameters of computer network - reliability, efficiency and cost are strongly related, because the goal of computer networks is always related to fast and reliable access to resources, which are usually shareable on the one hand and a set of resources, connections between host-computers, terminals, switching nodes and other types of redundance imply a cost of computer network, on the other hand. An increase of an efficiency needs highly reliable network components like links, switching nodes, terminals, host-computers etc.

Many approaches to computer network reliability and efficiency have been presented in the papers on the subject. Many of them take into consideration only a problem of connection between given pair of network nodes and others take into account together the problem of connection and availability of resources, switching nodes, links and host-computers.

Therefore, the most frequently used reliability measure - connection reliability defined as a probability of a connection between given pair of network nodes (7) (11) is not adequate for computer network, because it does not take into account an availability of network components like resources or switching nodes and does not take into consideration user requirements on network services.

A serviceability defined by Budka M. and Buczkowska T. (1981) and fault-tolerance measure - FTM defined by Budka M. (1983) are much more adequate for computer network reliability evaluation than measures based on terminal reliability, because they take into account together a reliability and an availability of

resources and other network components like switching nodes or host-computers. Moreover these measures represent user-oriented approach and if the failures of links and nodes are defined as their unavailability due to a technical defect or being temporarily busy, then serviceability and fault-tolerance measure may include some aspects of traffic problems (13).

The approach presented further on may be useful for computer network reliability analysis and as it was shown (9) for the designing of computer networks with required value of reliability and availability.

MSK - PILOTE COMPUTER NETWORK OF POLISH UNIVERSITIES

The reliability analysis presented further on concerns MSK - Pilote Computer Network of Polish Universities (Pilote Project of Ministry of Science, High Education and Technics, president prof. M. Bazewicz).

MSK - computer network is typical WAN (Wide-Area-Network), which architecture and protocols correspond to ISO/OSI requirements. The goal of the MSK - computer network has been to make possible the access to the resources of University Computing Centers to scientists and students of Wrocław, Gliwice, Warszawa, Poznań, Toruń, Kraków etc. Universities.

The basic topology of MSK (1983) is presented in fig. 1.

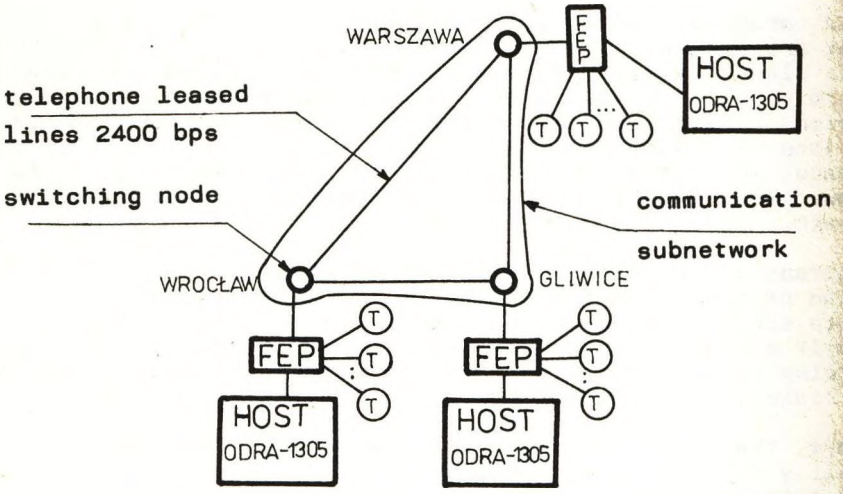


Figure 1
The basic topology of MSK (1983)

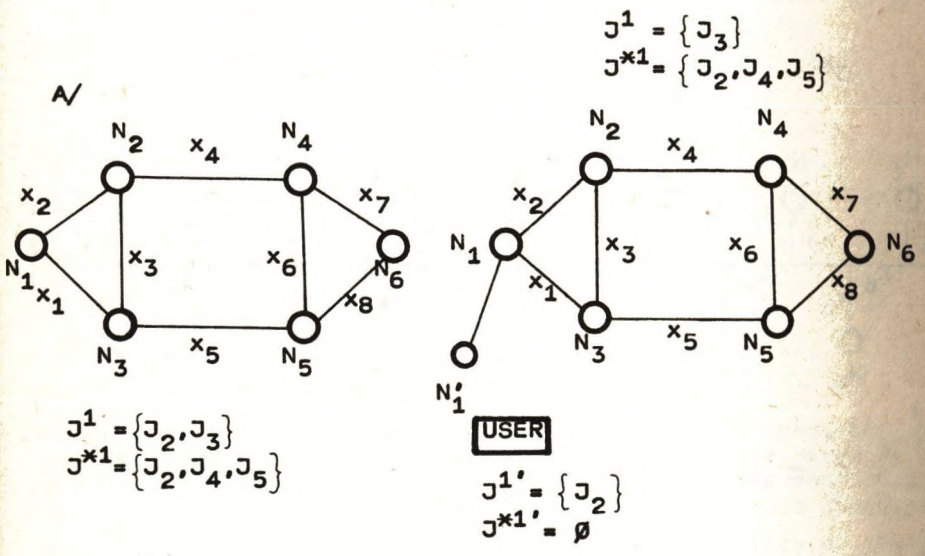
In 1985 other University Computing Centers will be included in MSK - computer network. The serviceability analysis makes possible to choose the optimal topology of new links from users' point of view and taking into account availability of resources and other computer network components.

COMPUTER NETWORK MODEL

Let us consider the graph $G(N, X)$ determined by a network topology, where graph vertices $N = \{N_1, N_2, \ldots, N_n\}$ correspond to nodes of computer network i.e. host-computers, switching-nodes, terminals or resources, and graph arcs correspond to network links $X = \{x_1, x_2, \ldots, x_m\}$.

The graph model for serviceability evaluation contains the sets of resources of every N_i - node denoted by J^{*i} and the sets of jobs demanded by N_j - node to be completed by computer network, denoted by J^j. The sets J^{*i} and J^j for every i and j contain corresponding elements - jobs and resources, which belong to the set of jobs J. Therefore $J^{*i} \subseteq J$ and $J^j \subseteq J$ for every i and j. The model makes possible the analysis of computer network reliability i.e. evaluation of serviceability and fault-tolerance measure. The model is valid for reliability analysis of wide area networks (WAN), local area networks (LAN) as well as multiprocessor supersystem.

Some typical cases (20) of the graph model application are presented in fig. 2, where the general assumption that N_j - node representative of a user may demand the accomplishment of only those jobs, which are not resident in N_j - node, is still valid.



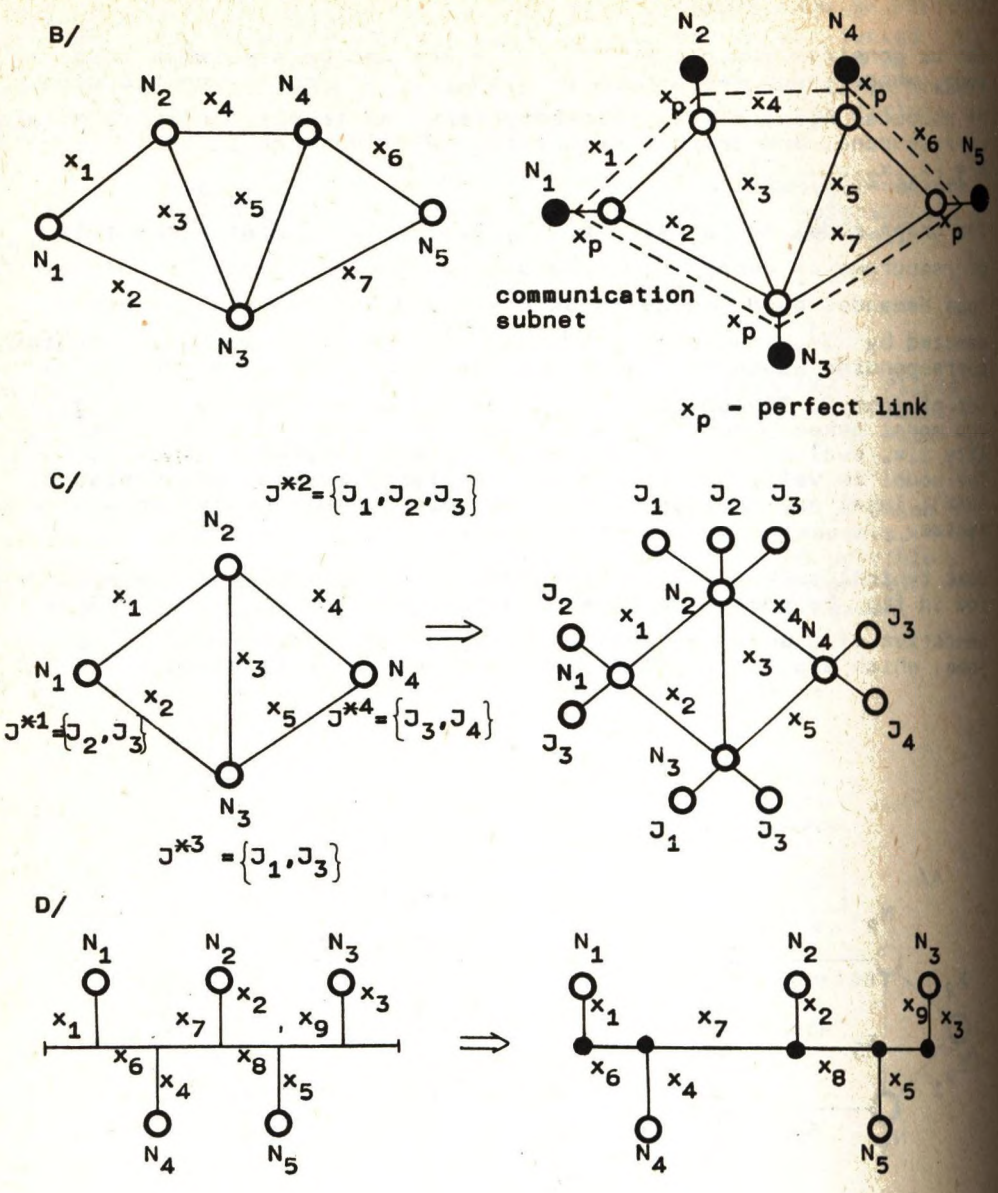


Figure 2
The graph model of a computer network - typical cases
A) N_1 - node (user) demands the J_2 job accomplishment in the same N_1 - node; B) switching and processing functions of network nodes are seperated; C) the access to every resource of N_i - node is independent; D) LAN - bus type

Let us consider Markov model of the computer network node (14) i.e. host-computer, switching-node or a resource. The state graph of such a node contains three states (fig. 3) i.e.

STATE "0" - the state of availability of the node

STATE "1" - the state of temporary unavailability e.g. buffer overflow

STATE "2" - the state of a node failure

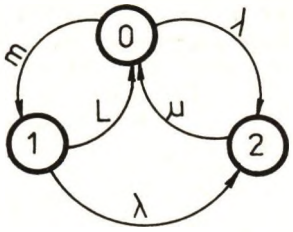


Figure 3
Markov model of the computer network node
λ- failure rate, μ- repair rate, L - buffer (or being busy) rate, m - service rate

The set of differential equations for the above model is presented below, where $P_i(t)$ is the probability of the i-th state (19).

$$\begin{cases} \frac{dP_0(t)}{dt} = -(\lambda + L)P_0(t) + mP_1(t) + \mu P_2(t) \\ \frac{dP_1(t)}{dt} = LP_0(t) - (\lambda + m)P_1(t) \\ \frac{dP_2(t)}{dt} = \lambda P_0(t) + \lambda P_1(t) - \mu P_2(t) \end{cases}$$

RELIABILITY MEASURES

From user's point of view a computer network is operable, if the jobs demanded by the user may be completed in the computer network, but not necessarily by one, given node. As it was mentioned above a serviceability for N_i - node (19) (20) takes into account user's requirements, because it is defined as a capability of job completion for N_i - node. Therefore, a serviceability for N_i - node is the event E_i, that there exist at least one-operational node N_j for every job $J_v \in J^i$ such that $J_v \in J^{*j}$ and at least one operational path Pt_{ij} from N_i node to N_j - node, i.e.

$$(E_i=1)\Leftrightarrow[\forall J_v\in J^i;\ (\exists N_j\in N;\ J_v\in J^{*j},\ N_j=1)\cap(\exists Pt_{ij},\ Pt_{ij}=1)]$$

where Pt_{ij} is the string of links and nodes connecting N_i - node and N_j - node without cycles.

The probability A_i of serviceability for N_i - node is a measure of job completion ability, which takes into account reliability parameters of communication subnetwork, availability of resources and workload parameters, i.e.

$$A_i = P\{E_i = 1\}$$

and because

$$E_i = \bigcap_{v,\ J_v\in J^i} E_i^v, \quad \text{then a serviceability } A$$

of an entire network is defined as follows

$$A = P\left\{\bigcap_i E_i = 1\right\} = P\left\{\bigcap_i \bigcap_{v,J_v\in J^i} E_i^v = 1\right\}$$

The second reliability measure FTM_i - fault-tolerance measure for N_i - node (20) is based on serviceability concept.

FTM_i is defined as the probability of the event $E_{i|F} = 1$, i.e.

$$FTM_i(F) = P\left\{E_{i|F} = 1\right\} = P\left\{\bigcap_v E_i^v|_F = 1\right\}$$

where $E_{i|F}$ is the event E_i in the network with failures $f_k\in F$ of computer network components like links and nodes and $E_i^v|_F$ is the same event for the v-th job.

Analogically, fault-tolerance measure FTM of the entire network is given by the following expression

$$FTM(F) = P\left\{\bigcap_i E_i|_F = 1\right\} = P\left\{\bigcap_i \bigcap_v E_i^v|_F = 1\right\}$$

These two reliability measures of computer networks, namely a serviceability and fault-tolerance measure are the basic ones used further on for the MSK - computer network reliability analysis.

COMPUTER NETWORK NODE ANALYSIS

The analysis presented below concerns one type node of computer network namely - switching node of communication subnetwork. The parameters used futher on correspond to switching node based on microcomputer SM-1633 (21).

According to the model of computer network node presented above one need to know four parameters of the node, namely λ - failure rate, μ - repair rate, L - buffer crush rate and m - service rate.

The first parameter - λ may be determined as a result of node reliability analysis, the second one μ - depends on an organization of repairs. L and m may be estimated after the exploitation of real computer networks.

According to the consideration presented by Grzegorczyk J. (19) L and m may be calculated as follows

$$\begin{cases} L = \alpha \cdot L_{max} \\ m = \gamma \cdot m_{max} \end{cases}$$

where $\alpha \in (0, 1)$; $\gamma \in (0, 1)$

m_{max} - maximal value of service rate (h^{-1})

$$m_{max} = \frac{3600\ f}{N_b(l + t_o f)}$$

f - transmission rate (bps)

l - average length of frames (b)

N_b - minimal number of frames which may be transmitted in one string (window size)

t_o - packet delay in the switching node

$$L_{max} = \frac{3600\ \beta\ n f}{N_b\ l}$$

n - number of communication channels of switching node

β - load factor of any channel (link)

The results of the switching node analysis are presented in fig. 4 where $p_{Ni}(t)$ - is the probability of N_i - node being operational, i.e. not failed and available.

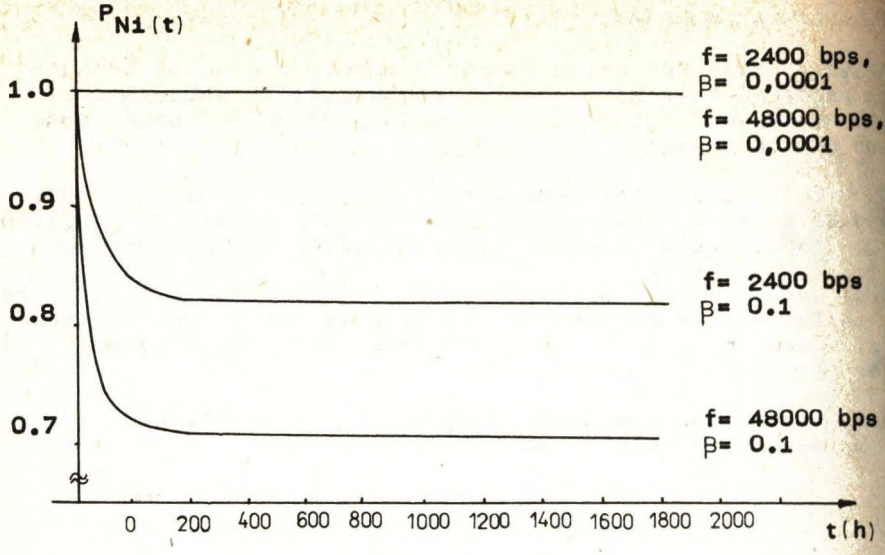


Figure 4
The probability of N_i - node being operational c.d.
$n = 3$, $N_B = 8$, $\ell = 1000b$, $t_o = 0.02$ s, $\alpha = 1$, $\gamma = 1$,
$\lambda = 0.001$ 1/h, $\mu = 1$ 1/h

MSK - SERVICEABILITY ANALYSIS

The consideration and results presented below concern the Pilote Computer Network MSK. In the first item the analysis of three-node network is presented, assuming that the set of jobs-resources is $J = \{J_1, J_2, J_4\}$, where J_1, J_2 are data bases and J_4 represents a computing ability. The topology of the three node MSK (1983) is shown in fig. 5.

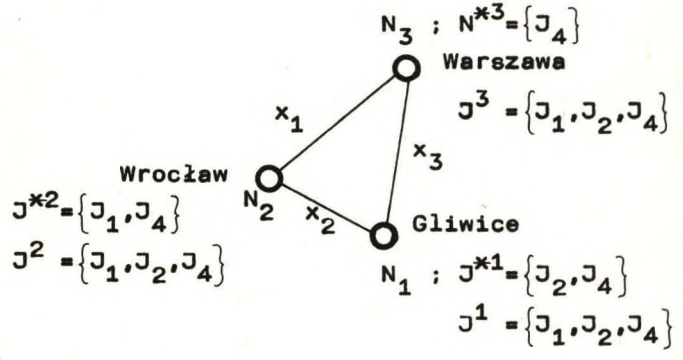


Figure 5
Thres-node topology of MSK

Let us assume that every node-user may demand the job J_1, J_2 or J_4 completion in considered computer network.

The model of computer network considered further on is the model with unreliable links and nodes, taking into account also an availability of switching nodes. The parameters of switching node has been presented in the previous chapter of this paper.

The results of the three-node MSK computer network are presented in fig. 6, as the functions $A_i(t)$ for i = 1,2,3.

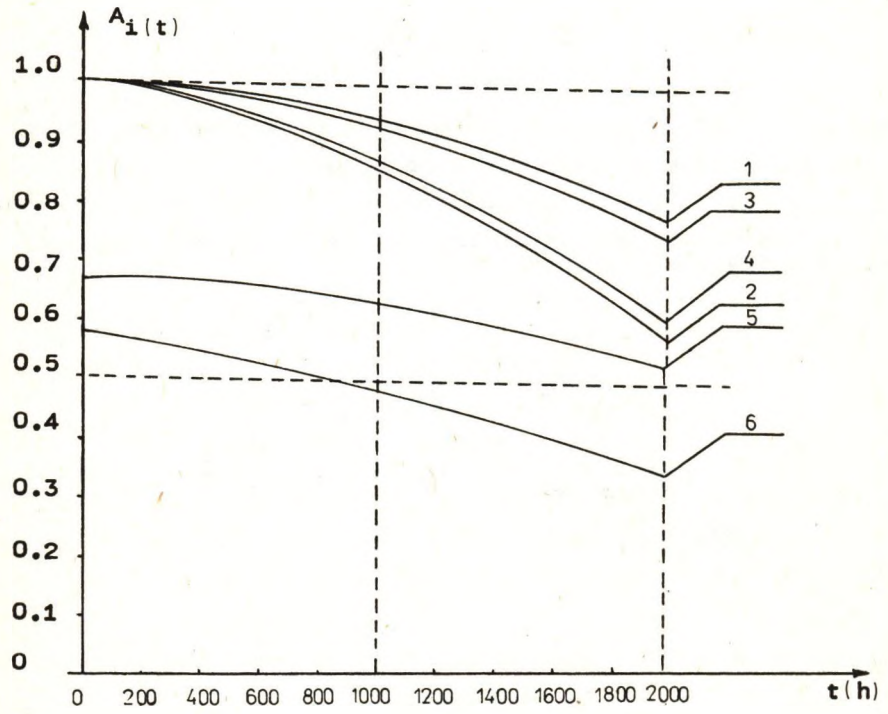


Figure 6
Three-node MSK serviceability analysis results
1 - $A_1(t) = A_2(t)$ and 2 - $A_3(t)$ without taking into account an availalility of switching nodes; 3 - $A_1(t) = A_2(t)$ and 4 - $A_3(t)$ for f = 48000 bps, $\beta = 0.0001$, $\alpha = \gamma = 1$; 5 - $A_1(t) = A_2(t)$ and 6 - $A_3(t)$ for f = 2400 bps, $\beta = 0.1$, $\alpha = \gamma = 1$

In the second item the sample application of serviceability analysis for computer network topology designing is presented. The problem was formulated as follows:

what is the optimal topology of four-node MSK computer network, where the fourth node is Toruń Computing Center, if the reliability of leased lines are the function of distance given by

following expression:

$$p_i = e^{-10^{-4} a \cdot t}, \quad \text{where} \quad a = \frac{\text{length (km)}}{100 \text{ (km)}}$$

For the given sets of resources:

$$J^{*1} = \{J_2, J_4\} \qquad J^{*2} = \{J_1, J_4\}$$

$$J^{*3} = \{J_4\} \qquad J^{*4} = \{J_3, J_4\}$$

where J_1, J_2 and J_3 are data bases and J_4 is computing ability, for the given sets of jobs demanded by every N_i - node $J^1=J^2=J^3=J^4=\{J_1, J_2, J_3, J_4\}$, the optimal topology of four-node MSK computer network from serviceability point of view - maximum value of serviceability, is presented in fig. 7.

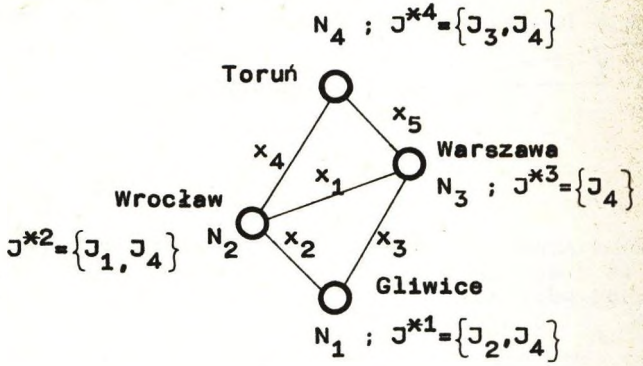


Figure 7
The optimal four-node topology of MSK

The results of the serviceability analysis of optimal four-node MSK are presented in fig. 8.

The application of serviceability analysis for the designing of five-node MSK computer network is presented below. The problem was formulated as follows:

what is the optimal topology of five-node MSK computer network where the fifth node is Poznań Computing Center for given set of resources:

$$J^{*1} = \{J_2, J_4\} \qquad J^{*2} = \{J_1, J_4\}$$

$$J^{*3} = \{J_4\} \qquad J^{*4} = \{J_3, J_4\} \qquad J^{*5} = \{J_4\}$$

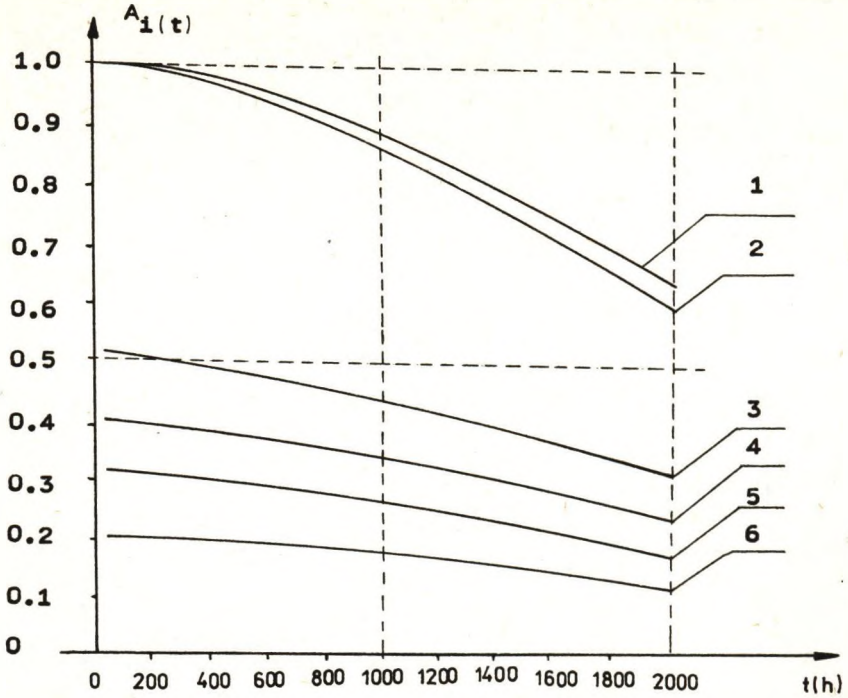


Figure 8
Four-node MSK serviceability analysis results

1 - $A_1(t) = A_2(t) = A_4(t)$, 2 - $A_3(t)$ for f=2400 bps, β=0.0001, $\alpha=\gamma=1$
3 - $A_1(t) = A_2(t) = A_4(t)$, 4 - $A_3(t)$ for f=2400 bps, β= 0.1, $\alpha=\gamma=1$
5 - $A_1(t) = A_2(t) = A_4(t)$, 6 - $A_3(t)$ for f=48000bps, β= 0.1, $\alpha=\gamma=1$

and for given set of jobs demanded by every N_i - node

$$J^1 = J^2 = J^3 = J^4 = J^5 = \{J_1, J_2, J_3, J_4\}.$$

The optimal five-node MSK computer network is shown in fig. 9.

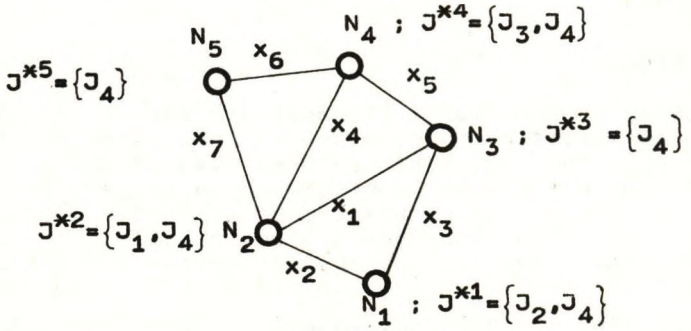


Figure 9
The optimal topology of five-node MSK

The results of the serviceability analysis of optimal five-node MSK are presented in fig. 10.

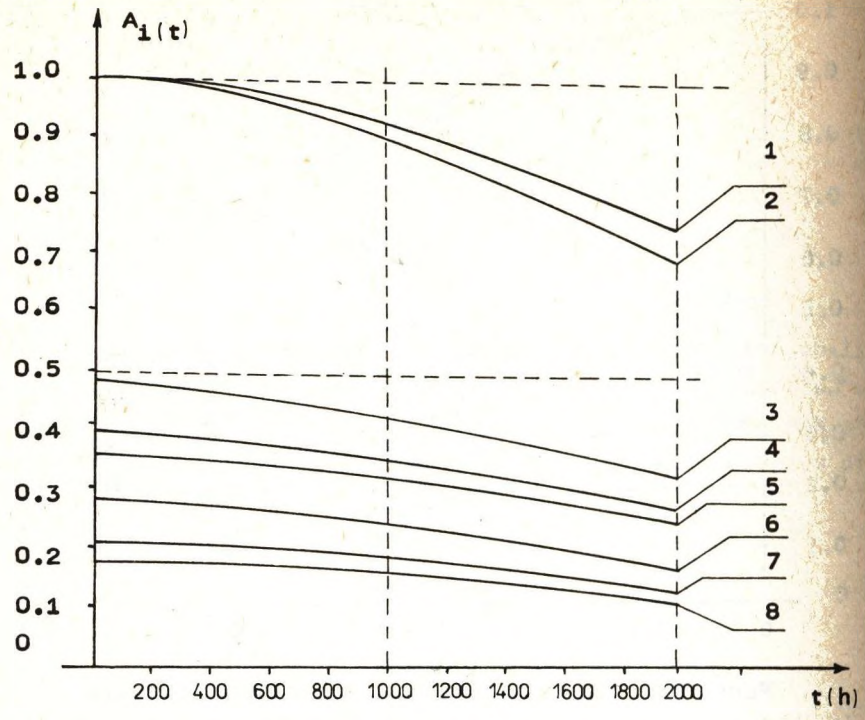


Figure 10
Five-node MSK serviceability analysis results

1 - $A_1(t) = A_2(t) = A_4(t)$ for f = 2400 bps, $\beta = 0.0001$, $\alpha = \gamma = 1$
2 - $A_3(t)$ $A_5(t)$ for f = 2400 bps, $\beta = 0.0001$, $\alpha = \gamma = 1$
3 - $A_1(t) = A_2(t) = A_4(t)$ for f = 2400 bps, $\beta = 0.1$, $\alpha = \gamma = 1$
4 - $A_5(t)$ and 5 - $A_3(t)$ for f = 2400 bps, $\beta = 0.1$, $\alpha = \gamma = 1$
6 - $A_1(t) = A_2(t) = A_4(t)$ for f = 48000 bps, $\beta = 0.1$, $\alpha = \gamma = 1$
7 - $A_5(t)$ and 8 - $A_3(t)$ for f = 48000 bps, $\beta = 0.1$, $\alpha = \gamma = 1$

CONCLUDING REMARKS

The application of serviceability approach for the sample MSK Pilote Computer Network analysis and in some cases for the synthesis of a network topology has been presented in the paper. The results of the analysis presented above concern implemented computer network i.e. the parameters of switching nodes and links, resource allocation and topology are corresponding to MSK-Computer Network, switching node based on microcomputer SM-1633 etc.

The serviceability approach may be also applied for computer network designing i.e. for topology determination and resource allocation (8) (20).

REFERENCES

[1] Beaudry M.D. - Performance - Related Reliability Measures for Computing Systems, IEEE Trans. on Comp. vol. C-27, June, 1978.

[2] Rubin I. - On Reliable Topological Structures for Message--Switching Communication Networks, IEEE Trans. on Commun. vol. Commun. 26, No. 1, January, 1978.

[3] Merlin P.M. - A Failsafe Distributed Routing Protocol, IEEE Trans. on Commun, vol. COM-27, No. 9, September, 1979.

[4] Svobodova L. - Reliability Issues in Distributed Information Processing Systems, FTC-9, 1979.

[5] Mine H., Hatayama K. - Performance Evaluation of a Fault-Tolerant Computing System, FTC-9, 1979.

[6] Chou T.C.K., Abraham J.A. - Performance/Availability Model of Shared Resource Multiprocessors, IEEE Trans. on Reliability vol. R-29, No. 1, April, 1980.

[7] Soi I.M., Aggarwal K.K. - Computer - Communication Network Reliability: Trends and Issues, Microelectronics and Reliability, vol. 21, No. 1, 1981.

[8] Budka M., Buczkowska T. - On the Reliability on Computer Networks, The Aloha System Technical Rep., 1981.

[9] Ignizio J.F., Palmer D.F. - A Multicriteria Approach to Supersystem Architecture Definition, IEEE Trans. on Computers, vol. C-31, No. 5, May, 1982.

[10] Irani K.B., Khabbaz N.G. - A Methodology for the Design of Communication Networks and the Distribution of Data in Distributed Supercomputer Systems, IEEE, Trans. on Computers, vol. C-31, No. 5, May, 1982.

[11] Ihara H., Mori K. - Highly Reliable Loop Computer Network System Based on Autonomous Decentralization Concept, FTC-12, 1982.

[12] Satyanarayana A. - A Unified Formula for Analysis of Some Network Reliability Problems, IEEE Trans. on Reliab., vol. R-31, No. 1, April, 1982.

[13] Anderson T., Lee P.A. - Fault Tolerance Terminology Proposals, FTC-12, 1982.

[14] Budka M., Grzegorczyk J. - Computer Network Model Taking into Account Availability of Resources, for Reliability Analysis, FTSD'83 Conference, Brno, 1983.

[15] Budka M., Buczkowska T. - A Serviceability of Computer Networks as a Reliability Measure, Basis of Control, Polish Academy of Sciences, vol. 13, No. 1, 1983.

[16] Huslende R. - Optimal Cost/Reliability Allocation in Communication Networks, FTC-13, 1983.

[17] Munarin J.A. - Dynamic Workload Model for Performance Reliability Analysis of Gracefully Degrading Systems, FTC-13,1983.

[18] Gunningberg P. - Voting and Redundancy Management Implemented by Protocols in Distributed Systems, FTC-13, 1983.

[19] Grzegorczyk J. - Analiza niezawodności sieci komputerowej z uwzględnieniem obciążeń węzłów podsieci komunikacyjnej,Silesian Technical University Press, Computer Science,No.5, 1983.

[20] Budka M. - Design of reliable computer networks, VII International Conference on Computing Systems Reliability FTSD'84, Sofia, oct., 1984.

[21] Budka M., Czyż A., Grzegorczyk J., Wojtuszek J. - Switching-node based on microcomputer SM-1633 for computer communication networks. International Conference Microelectronics'84, Praha, 1984.

SOLUTION OF THE PROBLEM OF STANDBY FOR DATA CIRCUITS BY MEANS OF A SPECIAL SUPPLEMENTARY HARDWARE AND / OR SOFTWARE

František Dlabola, Jiří Přibyl, Igor Vajda

Prag
Czechoslovakia

The special additional hardware and/or software is designated for data networks, where some circuits are secured by one or several standby circuits.

INTRODUCTION

Data transmission, i. e. the hardware and software for the transfer of messages between sources and processing sites is the relevant component determining the features of data teleprocessing systems. An essential part of any datateleprocessing system design is the protection of the transmitted information against transmission errors.

Standard data teletransmission protocols have been elaborated for this purpose. Most extensive use has been made of the SDLC protocol, representing a part of the IBM System Network Architecture and the HDLC protocol standardized by ISO. The protection is achieved by decision feedback with error correction based upon the repetition of erroneously transmitted frames having a length of several tenths of bytes. It is evident that with growing inherent error rate of the transmission circuit this procedure brings a decrease in data transfer rate. When the error rate exceeds a certain limit depending upon the length of the utilized frame, the data transfer rate will sink to zero. Apart from that, if the system is to be safe also against permanent fault condition of the transmission circuit, it has to be provided by the possibility of switching over to standby. This is being accomplished automatically in packet switched data networks, where bypass routes can be established in parallel with an interrupted section of a transmission circuit. However protection against short-term error-rate increase is very complicated even in this case and the solution affects system problems on a global network level.

This contribution deals with possibilities of solving the given problem. If the standby circuit for securing information transmission in the case of relatively permanent fault conditions of the main circuit is already existing, the possibility is evident to utilize it for data transmission even in situations, when owing to a short-term error-rate increase of the main circuit the data transfer rate sinks under an acceptable limit.

UNIT FOR SWITCHING OVER TO STANDBY CIRCUIT (USSC)

For deciding about switching over to the standby circuit it is necessary to know its instantaneous error-rate condition in both directi-

ons (full-duplex working in both directions, independent of the volume of the information being transmitted is assumed). For some applications it is advantageous to control the switching over to the standby circuit and vice versa by an equipment independent both of the terminal equipment and of the kind of transmitted information. This additional equipment - it may be called "Unit for switching over to standby circuit (USSC)" - should in no way affect the information transmitted between the terminal equipments on both extremities of the protected data circuits. Thus the relevant information about the condition of the circuit transmitting data can be obtained only from the terminal equipment, for each direction on the receiving side. Owing to the character of the utilized data transmission protocol only statistical information (evaluation of error rate for individual frames of variable length) can be obtained without substantial modification in terminal equipment.

The unit can be applied between the terminal equipment on the section, where information is protected according to the SDLC or HDLC protocols. On either extremity of the circuit one unit is inserted. One of them works as "master", the other as "slave". The master unit supervises the data circuits, measures their error rate. Any measuring is performed on a circuit, over which information is just being transmitted. The results relevant for switch-over action on individual circuits are evaluated from data obtained from the terminal equipment. The slave unit only interworks with the master-unit in measuring error rate and switches over the data circuits on its extremity according to instructions from the master unit. The data terminal equipment (DTE) is connected to the unit by a CCITT V.24 interface and behaves towards it in the same way as a synchronous modem (the interchange circuits are in general terminated in the data circuit terminating equipment DCE). The switch-over unit is connected to both DCEs also by means of interchange circuits according to Recommendation CCITT V.24, whereby it behaves towards the DCE in the same way as a DTE.

The DTE has to suply the switch-over unit with relevant information about the mode of operation (length of transmitted frames) and about the error rate (whether the frame was transmitted with or without error). The slave unit has to evaluate these informations and direct the result into the master unit to be used in the decision process. This transmission can be performed only by a circuit which is just not carrying traffic ("cold" circuit) whereas any circuit just transmitting information between DTEs ("hot") is assigned only for this transmission. The "cold" circuit must also simultaneously perform the transmission of a pseudo-random sequence for measuring error rate in both directions. In addition the "cold" circuit must transfer the command (in the direction master to slave) for switching over traffic to standby. For information transmission from slave to master an adequate error-protection code is used which eliminates possible errors and which does not affect simultaneous error measurement. However, for transmitting the switch-over command in the direction from master to slave it is necessary to choose a method which enables to transmit the single command with high security in a precisely defined instant so as to achieve minimum switch-over loss. For this purpose it is advantageous to use sequences, originally suggested by R. H. Barker for synchronization in digital transmission systems (see [1]). These special experimentally assessed sequences have the property of yielding, after passing through a suitable detector (linear sequence filter) a response in an exactly determined instant.

From the point of view of circuit design the switch-over unit consists of two functional blocks: the switch-over block which performs the switching over itself, on the level of the CCITT V.24 interface, and the control block, which performs the decision process. In realistic environment the function of the control block is performed by a relevantly equipped microcomputer (its memory capacity will not exceed 4K) and the switch-over block is built from standard logical TTL circuits. In USCC (the data transfer rate did not exceed 2400 bit/s) of first generation not even a "slow" microcomputer, e. g. with microprocessor I 8080 will be fully utilized in time, even if the function of the linear sequence detector for the mentioned use is performed only by software. Therefore it is expedient to constitute a multicontrol unit, where one computer will control several switch-over blocks.

GROUP UNIT FOR AUTOMATIC SWITCHING OVER TO STANDBY CIRCUITS (GUSSC)

The "Group unit for automatic switching over to standby circuits" is designed as a multimicroprocessor system with one control microcomputer (CM) and several communication microcomputers (CMM). The CM controls the transfer across the system bus, to which the individual CMMs are connected. The system bus is maximally simplified but for the given purpose is fully satisfactory. The operation of the system is distributed among the individual microprocessors in the following way:

The CM at the master station (MS) possesses for each circuit pair four ring memories with a capacity of 16 bytes each, where the transmission statistics of each of the 4 channels is stored. From this information the control computer CM periodically evaluates the condition of both circuits and in case of need it sends a command for circuits switch-over.

For each circuit pair, where GUSSC operates as a slave station (SS) the CM decodes the data arriving from the opposite station and tests them, whether they contain the command for switching over the circuits. In the positive case the CM then sends a command for circuits switch-over to the CMM.

The CMM transmits and receives data on the telecommunication circuit or on all circuits attended by it, which are "cold" at a given instant (i. e. there is no DTE traffic on them). The CMM processes the received data from the point of view of transmission errors and at the same time it logs the part of the data block carrying information from the opposite station into the reserved memory area, from where the CM takes it over. The CMM simultaneously controls the operation of the block for circuits switchover and processes interchange signals from the DTE.

The transfer over the system control bus is controlled by the CM. The transfer is always performed between the operation memory of the addressed CMM and the memory of the CM. The operation of CM and CMMs is not synchronised; the CM, independently of the CMMs reads their status words, from which it must be clear, whether CMM has or has not taken over information for CM.

The CM can serve up to 8 pairs of 9600 bit/s telecommunication circuits; with lower circuit signalling rates proportionally more. The CMM is designated for serving 8 circuit pairs operating with a signalling rate of 1200 bit/s each. For higher circuit signalling rates

the maximum number of attended telecommunication circuit pairs (and thus the maximum number of DTEs) is proportionally decreased.

THEORETICAL PROJECT OF THE GUSSC (USSC)

Physically, the GUSSC consists of a master station (MS) situated at the central node of a star subnet and several (up to 16) slave stations (SS) situated at selected terminal nodes of the subnet. The MS collects all information needed to evaluate instant effective transmission rates (ITR) for both the forward and backward channels of the circuits aiming at the selected terminal nodes as well as the same quantities for the channels of the parallel stand-by circuits. Further, the MS actually evaluates these four quantities for each terminal node separately,and, according to a preprogrammed decision rule, it generates the command switch over the considered "hot" circuit (i. e. that connecting at the given moment the central node with the corresponding terminal node) to the parallel "cold" (stand -by) circuit. In this section we explain what information is used to evaluate the above mentioned ITS's, how the ITR's are defined, and what is the role of SS in collecting and transporting this information, as well as how the command to switch over is transported to the corresponding SS. The main attention is paid to the decision rule - the most decisive element of the unit. Obviously, we can restrict the attention to just one pair of central-terminal nodes.

Hereafter we reserve the symbols $\mathcal{H}$, $\mathcal{C}$ to denote the "hot" and "cold" channels and the symbols $\mathcal{M}$, $\mathcal{S}$ to denote the channels of the "main" and "standby" circuits. The forward and backward channels are distinquished by the additional symbols M→S or S→M respectively. Thus $\mathcal{H}$(M→S) denotes the hot forward channel which is equal either to $\mathcal{M}$(M→S) or $\mathcal{S}$(M→S) respectively etc. We shall write simply $\mathcal{K}$ instead of $\mathcal{K}$(X→Y) for every $\mathcal{K} \in \{\mathcal{H},\mathcal{C},\mathcal{M},\mathcal{S}\}$ and X,Y ∈ {M, S} under consideration.

In a theoretical sense, each channel $\mathcal{K}$ under consideration denotes a binary stationary channel with additive noise. It is described by a consistent system of noise source probabilities $P_n, n \in \mathcal{N}$, $\mathcal{N} = \{1,2,\ldots\}$, defined on n-sequences $\underline{a} \in \{0,1\}^n$. The n-sequence consisting of pure zeros will be denoted by $\underline{0}$. From the point of view of ARQ transmission employing systematic linear (n,k)-codes, the most important operational characteristic of a channel $\mathcal{K}$ is the function $P_n(\underline{0})$ considered on the domain $n \in \mathcal{N}$. (Note that the HDLC protocol specifies exactly such a transmission scheme with $n \in (32, 32.000)$, k = n-32, and with shortened cyklic (n,k)-codes generated by the polynomial $g(\lambda) = 1 + \lambda^5 + \lambda^{12} + \lambda^{16}$). Indeed, it is well known that the effective information transmission rate of a channel $\mathcal{K}$ under the above described transmission scheme is approximately

$$\frac{k}{n} P_n(\underline{0}) \text{ bit/s.} \qquad (1)$$

Since the first term $Q_n = k/n$ is a known function of $n \in \mathcal{N}$ for every given class of codes, the function $P_n(\underline{0})$ yields (through local or absolute maxima of the product (1)) the optimum lengths of frames for the given channel $\mathcal{K}$ or for a class of such channels.

We found that not only for the purpose of this paper but also in the more general problems connected with the design of codes for data transmission, the four state models of channels $\mathcal{K}$ described by six parameters arranged into the Markov matrix

$$\begin{array}{c} \\ B \\ G_1 \\ G_2 \\ G_3 \end{array} \begin{array}{c} \begin{array}{cccc} B & G_1 & G_2 & G_3 \end{array} \\ \begin{bmatrix} 1-w_1-w_2-w_3 & w_1 & w_2 & w_3 \\ 1-\theta_1 & \theta_1 & 0 & 0 \\ 1-\theta_2 & 0 & \theta_2 & 0 \\ 1-\theta_3 & 0 & 0 & \theta_3 \end{bmatrix} \end{array} \qquad \begin{array}{l} \text{where } w_3 > 0 \text{ and } \theta_3 > \theta_i \\ \text{for all } i \in \{1,2\} \text{ with } w_i > 0 \end{array}$$

are satisfactory. The matrix elements describe the probabilities of the corresponding transitions in the state space $\{B, G_1, G_2, G_3\}$, where the noise digit 1 is supposed to be generated in B and 0 in G_i and where G_3 is a "truly good state" (cf. Kanal and Sastry (1974), Vajda (1982,1984a)). The error frequency is here given by

$$p = (1 + \frac{w_1}{1-\theta_1} + \frac{w_2}{1-\theta_2} + \frac{w_3}{1-\theta_3})^{-1} \approx \frac{1-\theta_3}{w_3} \qquad (2)$$

and

$$P_n(\underline{0}) = p(\frac{w_1\theta_1^{n-1}}{1-\theta_1} + \frac{w_2\theta_2^{n-1}}{1-\theta_2} + \frac{w_3\theta_3^{n-1}}{1-\theta_3}) \qquad (\text{for all } n \in \mathcal{N})$$

$$\approx \theta_3^n \qquad (\text{for "large" } n, \text{ say } n \geq 30). \qquad (3)$$

Here we combine the formulas given in Vajda (1982) with the assumption $\theta_3 > \theta_i$ for $i = 1,2$ and with the fact that

$$\frac{w_3}{1-\theta_3} \quad \frac{w_i}{1-\theta_i} \qquad \text{for } i = 1,2$$

for all realistic models of real channels (see Vajda (1984)).
A robust statistical estimation procedure for fitting the four-state models to real channels (represented by the empirical distribution functions of the error-free gaps of sample noise sequences) has been described in Vajda (1984a). In the meantime, simulation studies have been carried out (see Kodera (1984)) where various statistics yielded by models estimated in this manner have been successfully tested against similar statistics evaluated for real telecommunication channels. A maximum likelihood procedure which estimates w_1+w_2, $\theta_1+\theta_2$, w_3 and θ_3 directly from the noise sequences of length n has been proposed in Vajda (1982). This procedure is apparently biased, as indicated by the results of a simulation study presented in the next table and corresponding to $n = 10^5$ and to a "filter size" 10 (three different matrices are considered). Nevertheless the table demonstrates

Matrix	w_1+w_2	$\theta_1+\theta_2$	w_3	θ_3
1	0.1 $0.16(10^{-3})$	0.5 $0.52(5.10^{-3})$	0.3 $0.24(8.10^{-4})$	0.999 $0.9989(10^{-8})$
2	0.3 $0.35(5.10^{-4})$	0.15 $0.32(10^{-3})$	0.2 $0.15(10^{-4})$	0.999 $0.9989(2.10^{-8})$
3	0.3 $0.36(10^{-2})$	0.15 $0.29(5.10^{-3})$	0.2 $0.14(2.10^{-3})$	0.99999 0.999988 (4.10^{-11})

Table 1.: True values and sample means (sample variances) of maximum likelihood estimates of matrix model parameters from 10 realizations of estimates.

that this procedure provides satisfactory statistical approximations to the unknown parameters w_3, θ_3 and parametric functions w_1+w_2, $\theta_1+\theta_2$ as well.

The purpose of the noise model presentation above was to demonstrate the possibility to analyze more deeply the function $P_n(\underline{0})$.

It follows from (1) that, in order to evaluate an instant transmission rate $ITR(\mathcal{K})$ for a real channel $\mathcal{K}$, it is sufficient to evaluate the relative frequency $\hat{P}_n(\underline{0})$ of frames of length n bits transmitted through the channel $\mathcal{K}$ during a time interval $(t-\Delta t, t)$ without the detected error. Then the definition

$$ITR(\mathcal{K}) = \frac{k}{n}\,\hat{P}_n(0) \quad \text{bit/s}$$

for a fixed moving window length Δt applies to the channel at time instant t. There is however one difficulty connected with this definition, namely that in some protocols (like HDLC) the frame length n quite freely fluctuates so, that neither pair of frames transmited during $(t-\Delta t, t)$ has to have the same lengths. In a situation where n fluctuates between 48 (confirmation frames) or 32 ("empty" synchronisation frames) and 2048 (data carying frames) we proceeded as follows.

a) Two signals A FRAME IS BEING RECEIVED and PARITY CHECK OUTCOME are transported from the node telecommunication computer to the corresponding frame computer CM of the unit.

b) The integer interval (32,2048) has been quantized into 5 subintervals with median values 64,128,256,512 and 1024. Using its inner clock and the first of the signals above, the frame computer CM approximates the frame length by one of these median values. Let the corresponding value be n and let $\nu \in \{0,1\}$ be the outcome of the second signal. For each of the four channels $\mathcal{K}(X \to Y)$ under consideration, all values (n,ν) within the time interval $(t-\Delta t, t)$ are stored in the memory of MS (the SS transports the pairs (n,ν) related to $\mathcal{H}(M \to S)$, $\mathcal{C}(M \to S)$ to MS through $\mathcal{C}(S \to M)$ together with a fixed frame while MS feeds (M S) with semi-fixed test frames - details at the end of this section.

c) For each of the four channels, the MS permanently evaluates $\hat{P}_n(\underline{0})$ for every median length n separately, then on the basis of (3) it evaluates the 256-bit modifications

$$\hat{P}_{256}(\underline{0}) = (\hat{P}_n(\underline{0}))^{256/n}$$

and, finally, the MS evaluates the ordinary weighted sum of these modifications (we denote the sum by $P_{256}(\underline{0})$). In other words, the empirical evidence coming from the frames of various lengths is transformed, using the approximately exponential character (3) of the function $P_n(\underline{0})$, into the standardized evidence related to the length 256 bits and $\hat{P}_{256}(\underline{0})$ is nothing but a "mixed relative frequency estimator" of the value $P_{256}(\underline{0})$.

d) Let us denote by $\Pi(\mathcal{K})$ the value of $\tilde{P}_{256}(\underline{0})$ corresponding to a channel $\mathcal{K}$. The decision rule of MS is based on a comparison of the values

$$\|\mathcal{C}\| = \min\{\Pi(\mathcal{C}(M \to S)),\ \Pi(\mathcal{C}(S \to M))\}$$
$$\|\mathcal{H}\| = \min\{\Pi(\mathcal{H}(M \to S)),\ \Pi(\mathcal{H}(S \to M))\}$$

and it also takes into account the fact that

$$\Pi(\mathcal{K}) \pm \frac{1}{3}\sqrt{\Pi(\mathcal{K})(1-\Pi(\mathcal{K}))}$$

is approximately the 90 % -confidence interval for the unknown $P_{256}(\underline{0})$ (this fact has been established using a central limit theorem for Markov chains). MS generates the signal SWITCH as soon as

$$\|\mathcal{C}\| - \tfrac{1}{3}\sqrt{\|\mathcal{C}\|(1-\|\mathcal{C}\|)} \geq \|\mathcal{K}\| + \tfrac{1}{3}\sqrt{\|\mathcal{K}\|(1-\|\mathcal{K}\|)}.$$

If there is a need to prefer the transmission through the main circuit, the last condition can easily be reprogrammed into the following form

$$\|\mathcal{C}\| - a\sqrt{\|\mathcal{C}\|(1-\|\mathcal{C}\|)} \geq \|\mathcal{K}\| + a\sqrt{\|\mathcal{K}\|(1-\|\mathcal{K}\|)},$$

where e. g. a = 4/15 if $\mathcal{C} = \mathcal{M}$ and a = 6/15 if $\mathcal{C} = \mathcal{G}$.

e) The confidence interval in d) has been established under the assumption that 16 frames have been transformed during the time interval Δt seconds. It is to be noted that the unit proceeds in a slightly different manner than described in c), namely, the statistics considered in c) are evaluated after each 4 test frames are transmitted through $\mathcal{C}(S \to M)$ and the data (n,ν) from the last 16 frames transmitted through every channel $\mathcal{K}$ under consideration are used to evaluate the instant value of these statistics. Thus the signal SWITCH can be generated by MS with the period $1024/\varphi$ sec where φ is the transmission rate of $\mathcal{C}$ which is supposed to be automatically adapted to the transmission rate of $\mathcal{K}$.

We conclude this section by an insight into the simultaneous testing of $\mathcal{C}(M \to S)$ and $\mathcal{C}(S \to M)$ by sequences of 256-bit frames and the use of these channels for information interchange between MS and SS. MS transmits through $\mathcal{C}(M \to S)$ either a Barker sequence plus the standard flag bytes or the inverse Barker sequence. The latter case takes place at every moment when the signal SWITCH is generated. Thus, following the signal SWITCH, both MS and SS switch on simultaneously the so far cold circuit. On the other end of the data link, SS transmits through $\mathcal{C}(S \to M)$ the same Barker test sequence with the last 64 bits replaced by two words of a 32-bit code. This code transmits to MS the information about (n,ν) related to $\mathcal{K}(M \to S)$ (32 bits) and to $\mathcal{C}(M \to S)$ (32 bits too). The code is non-systematic extended Hadamard code with the Hamming distance 25. One alternative of the information transmitted by this code is that no new frame has been received at the node of the SS (thus the amount of information transmitted by this code is log(2.5+1/bits).

Let us note that, using the robust $D^{0.2}$-estimator of $P_n(\underline{0})$ (see Vajda (1984b)) instead of the non-robust maximum likelihood relative frequency estimator $\hat{P}_n(\underline{0})$, we can reduce the 16-frame memory of the unit to an 8-frame memory with a practically negligible increase of the frequency of the signal SWITCH (this may be accompanied by one extra retransmission of a frame), with a slight increase of the effective transmission rate and with a substantial increase of responsivity of the unit to abrupt changes of hot circuits. This has been verified by recent simulated computer studies. The robustification of the decision rule can be carried out simply by replacing the statistics $\|\mathcal{C}\|$, $\|\mathcal{K}\|$ by $\|\mathcal{C}\|^*$, $\|\mathcal{K}\|^*$, where

$$\|\mathcal{C}\|^* = \frac{\|\mathcal{C}\|^{5/4}}{(1-\|\mathcal{C}\|)^{5/4}}$$

and where $\|\mathcal{K}\|^*$ is defined analogically.

PURE SOFTWARE SOLUTION OF THE PROBLEM OF STANDBY FOR DATA CIRCUITS

The above described method of standby cannot be regarded from the point of view of effective main and standby circuit utilisation as

ideal, because at a given instant only the main (hot) circuit is utilised for transmission of useful information. The remaining (cold) circuit stays unutilised for the transmission of useful information. However, if we admit simultaneous information transmission over both circuits, there exists the possibility of the following purely software solution which utilises in full extent the software of networks with HDLC procedures. This solution, principle or algorithm, which in the case of two circuits we denote by the term "dichotomic" consists in dividing the useful information by the terminal equipment or the node computer into two equal parts. Each of these parts is then stored in the buffers of the individual circuits. These buffers are parts of DTEs or node computers. The DTE or node computer starts transmitting the buffer content frame by frame into the corresponding circuits simultaneously, irrespective of the actual qualitative condition of the circuits. The remainder of the untransmitted message, which remains in the buffer of the worse quality circuit is again divided into two equal parts. One part stays in the buffer of the instantaneously worse circuit and the other is taken over by the already empty buffer of the better circuit. After the next clearing of one of the buffers the procedure is resumed. The whole process lasts until the whole useful message has been delivered.

The described algorithm, which is an optimum one for circuits, whose quality we have no apriori information about and which is applicable with certain modifications to n circuits (1 main and n-1 standby) was described in Přibyl (1984).

REFERENCES

[1] Barker, R. H., Group synchronization of binary digital systems. Communication theory, W. Jackson (ed.), pp. 273-278 (Butterworth Scientific Publications, London, 1953).

[2] Kanal, L. N., Sastry, A. R. K. (1974), Models for channels with memory and their application to error control, Proc. IEEE, 66(1978), pp. 724-744.

[3] Kodera, J. (1984), M.Sc. thesis, Dept. of Telecomm., Czech Technical University, Prag (June 1984).

[4] Vajda, I. (1982), Estimation of probabilistic noise models based on filtration of sample noise sequences, Kybernetika 19(1982).

[5] Vajda, I. (1984a), Minimum chi-square estimates of multistate communication channel models, Probl. of Control on Inform. Theory 13(1984), pp. 343-356.

[6] Vajda,I. (1984b), In Recent Results in Estimating Theory and Related Topics, E. J. Dudewitz, D. Plachky and P. K. Sen (ed.) (Oldenbourg, Munich, 1984).

[7] Přibyl, J. P. R., Dichotomoc algorithm as a criticism of the unit for switching over to standby circuit (USSC), Internal Lecture Thesis, Dept. of Telecomm., Eltechn. Faculty, Technical University, Prag (November 1984).

AUTHOR INDEX

ANAGNOSTOU,M.E. 7-82
ALFIMENKOV,A.V. 4-48
ALMÁSI,L. 3-128
ARATÓ,A. 4-9
ATANASSOV,Y. 5-55

BACOPOULOU,S.A. 7-82
BAKONYI,P. 2-1
BALÁZS,A. 4-32
BARCHANSKI,J. 3-80
BAZEWICZ,M. 1-8, 1-21
BINNS,S.E. 1-1
BOCHMANN,G.V. 3-80, 6-133
BORZEMSKI,L. 4-38
BOUTMY,B.J. 1-41
BOYANOV,K. 8-1
BRINKSMA,E. 6-1
BUDKA,M.M. 8-67

CARCHIOLO,V. 7-21
CASTANET,R. 7-68
CAUL,D.J. 1-1
CHEUNG,T.Y. 6-103
CSABA,L. 2-1

DESJARDINS,P. 3-80
DIMITROV,B. 8-1
DLABOLA,F. 8-81
DUARTE,O.C. 5-23
DUBILEWICZ,I. 7-102

FILP,A. 3-72
FRYZLEWICZ,Z. 7-102

GABLER,H.G. 3-23
GANOV,J. 5-55
GÁLL,M. 4-32
GECSEI,J. 3-80
GIESE,P.E. 4-48
GIESE,P.H. 4-48
GOTCHEVA,T. 5-55
GROZ,R. 6-50
GRZECH,A. 4-38
GRZEGORCZYK,J. 8-67
GÜTTER,D. 4-52

HACKLER,B. 4-52
HADDAD,J. 7-34
HARANGOZÓ,J. 6-41, 6-66
HEINER,M. 6-87
HENNECKE,A. 8-10
HEYMER,V. 8-10
HORVÁTH,P. 3-128
HUDYMA,E. 8-29

JANOUSEK,I. 8-16
JARD,C. 6-50
JÁKFALVI,R. 4-32
JUNKER,E. 1-50
JUHÁSZ,G. 8-54

KASPRZAK,A. 4-38
KASZUBA,R. 1-36
KELLERMAYR,K.H. 4-23
KERTÉSZ,A. 3-128
KÖNIG,H. 6-87, 4-52
KOLESNIK,K. 7-102
KOMOROWSKI,W. 7-102
KOSZALKA,L. 4-38
KOSMULSKA-BOCHENEK,E. 8-29
KOVÁCS,L. 6-41; 6-66
KOVÁCS,L. 3-118
KOVÁCS,K. 8-54
KOZLOWSKI,A.T. 8-22
KUBAS,CH. 4-52
KUNZE,H. 3-37, 3-95
KUNFT,W. 2-1, 5-12
KWIATKOWSKI,J. 1-36

LÁBADI,A. 3-128
LOGRIPPO,L. 6-1
LÖFFLER,H. 4-52

MAZGON,S.K. 3-61
MAUER,H. P-1
MEIER,I. 4-52
MEIER,H.W. 1-50
MONIN,J-F. 6-50
de LA MOTTE,P. 5-23
MURASZKIEWICZ,M.R. 2-9

NEIDHOLD,G. 4-52
NÉMETH,G. 3-53
NOWICKI,Z.M. 2-9

PALAZZO,S. 7-21
PAPPALARDO,G. 6-1, 7-21
PAUL,M. 2-1, 5-12
PEKAR,J. 7-118
PENZEL,R. 5-63
POPOV,P. 8-16
PORIZEK,R. 5-35

PRIBYL,J. 8-81
PROTONOTARIOS,E.N. 7-82
PUZMAN,J.F. 5-43

QIU,X. 6-103
QUANG,N. 8-40

RAFIQ,D. 7-34
RAYNER,D. 7-1
RUBIN,J. 7-50
RUSKOV,P. 8-1

SAMBOR,K. 3-1
SARIKAYA,B. 6-133
SARKADI-NAGY,I. 4-9
SÁRI,I. 8-54
SCOLLO,G. 6-1
SEBESTYÉN,I. 3-106
SHEPPARD,D.A. 5-1
SIJELMASSI,R. 7-68
SPRATT,E.B. 1-1
STEINACKER,M. 6-25
STOINOV,T. 5-55
STUCHLIK,F. 1-8
SUGÁR,P. 4-32
SULYÁN,J. 4-9
SZENTNER,J. 4-32, 6-66

TASKER,R. 4-1
TELBISZ,F. 4-9

VAJDA,I. 8-81
VÁRSZEGI,L. 1-55

WOZNIAK,J. 7-92

YAKUBAITIS,E.A. P-6
YANEV,K. 8-1

ZEISEL,T. 8-40