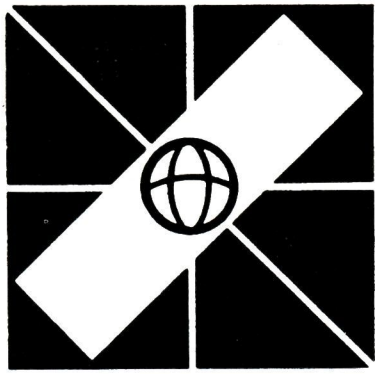
ICCC–IFIP–ITU–UNESCO



COMNET'90

FORWARD INTO THE SECOND QUARTER–CENTURY IN NETWORKING

DELEGATES COPY

JOHN VON NEUMANN SOCIETY FOR COMPUTING SCIENCES
BUDAPEST, HUNGARY

ICCC — IFIP — ITU — UNESCO

COMNET'90

"Forward into the second quarter-century in networking"

8—10 May 1990

Budapest, Hungary

Organized by:

John von Neumann Society for Computing Sciences

Sponsored by:

International Council for Computer Communication
International Federation for Information Processing
International Telecommunication Union
United Nations Educational Scientific and Cultural Organization

Publication Committee

Csaba, L.
Szentiványi, T.
Tarnay, K. (Mrs.)

PREFACE

It is my great pleasure to launch this volume with papers submitted to the COMNET '90 conference. We gladly took the opportunity to organize in Budapest the fourth event of an excellent series, within fifteen years, reflecting the results in the development and applications of computer networks.

We are moreover proud that several outstanding achievements came out for the first time on each occasion and of course this occurence offered the possibility for experts coming from all over the world to meet each other and to have fruitful discussions. Thus the COMNET performed the function of bringing specialists together from East and West too establishing new personal contacts or strengthening the existing ones between them. These relations are important factors of research and development.

On this occasion special importance has been attached to the fact that the original idea of computer communications emerged just a quarter-century ago. The leading slogan of the conference reflects adequately this milestone which marks at the same time the start of the next 25 years in networking. This renowned anniversary aimed not only at the demonstration of the present status in networking but to some extent at giving a brief account of the past phase and at showing the new prospects pointing ahead.

The moment of the conference coincides with the circumstance that Hungary, the host country of COMNET, underwent just in this period significant political changes, becoming herewith a real democratic state which ensures the return to Western-European institutions joining into the circulation of that part of the world.

The Conference Organizers made an effort in the selection of topic areas of COMNET '90 to give place to some of the timely questions such as the protocolspecification and implementation, similarly the high-speed-local-networks the application of which reached the turning-point. Another important thing which the organizers kept to the fore was that leading companies in this field should explain their views about the further development of computer networks.

The topic area of networking is very broad and because of the limited length of the conference every subject could not have been presented and discussed during this period in spite of every effort the organizers made.

Finally our thanks should go to all institutions supporting and helping the realization of the conference in the first place the John von Neumann Society for Computer Sciences /Budapest/

which took the lion's share in preparing the event but also above all to IFIP whose TC-6 /Data Communication/ Committee acted as principal initiator of the COMNET meetings. And last but not least we expand special thanks to the authors of this volume who, with their contributions, ensured the high scientific level of the conference.

T. Szentiványi
Conference Governor

CONFERENCE COMMITTEES AND OFFICIALS

Honorary Chairman:	Valter, F. (H)
Conference Governor:	Szentiványi, T. (H)

PROGRAMCOMMITTEE

Tarnay, K. (H), Chair
Bailes, P. (UK)
Bakonyi, P. (H)
Bochmann, G.v. (CDN)
Danthine, A. (B)
Ercsényi, A. (H)
Gabler,H. (FRG)
Gagliardi, D., ETSI (I)
Handler, G.J. (USA)
Harangozó, J. (H)
Horváth, P. (H)
Ivanyos, L. (H)
Kappel, K.W. (USA)
Khakhar, D. (S)
Könyves, Tóth, P. (H)
Lajta, Gy. (H)
Makino, Y. (J)
Mazgon, S. (H)
Meandzija, B. (USA)
Pouzin, L. (F)
Radford, P. (UK)
Rudin, H. (CH)
Schicker, P. (CH)
Spaniol, O. (FRG)
Szabó, Cs. (H)
Telbisz, F. (H)
Tóth, P. (H)
Uhlig, R. (USA)
Ullmann, K. (FRG)
Westcott, J. (USA)

INTERNATIONAL ADVISORY BOARD

Basu, D. (IND)
Bazewicz, M. (PL)
Boutmy, E.J. (NL)
Boyanov, K. (BG)
Casaca, A. (P)
Ekberg, J. (SF)
Eliassen, F. (N)
Housley, T. (AUS)
Königshofer, F. (A)
Ono, K. (J)
Owolabi, A. (UNESCO)
Puzman, J. (CS)
Riera, J. (E)
Samoylenko, S. (SU)
Sattler, S. (GDR)
Sobczak, W. (PL)
Twyver, D.A. (CDN)

ORGANIZING COMMITTEE

Chair:	Gergely, Cs.
Co-chair:	Bagonyi, L.
Public relations:	Brückner, H.
Proceedings:	Csaba, L.
Secretary:	Hetthéssy, G. (Ms.)
Treasurer:	Tóth, M. (Ms.)
Industry liaison:	Vadász, P.

TABLE OF CONTENTS

PLENARY PRESENTATIONS

Integrated and Managed Telecommunications Networks - the Next Decade
D.M. LEAKEY P-1

High Speed Local Area Networks - What, Why, When and How: Planning, Installation and First Experiences of a HSLAN in a Heterogeneous Environment
O. SPANIOL P-10

Telecommunication Standardization in Western Europe
H. GABLER P-20

SESSION 1 - LANs, MANs and ISDN

A Concept Becomes Reality First Experiences with Commercial ISDN Services
P. SCHNEIDER 1-1

A Routing Scheme for a Non-Strict Hierarchical Networking Environment
S. BAHK, M. EL ZARKI 1-15

ARQ Error Control for Multipoint Satellite Communication
S. JACOBS, U. QUERNHEIM 1-25

An Approximate Analysis of Multi-Hop S-Aloha Networks with or without Immadiate Acknowledgements
J. WOZNIAK 1-35

Towards High Speed Networking in Local and Metropolitan Areas
P. MARTINI 1-45

Bandwidth Allocation and Flow Control in ATM Networks
J. FILIPIAK 1-56

Performance Evaluation of Token Passing Bus LANs
G. PAPP, J. HARANGOZÓ 1-65

Performance models of Access Integration Policies in ISDN Channels
O. EL DESSOUKI, H. BARAKA 1-75

A Survey of Fault - Tolerant LANs
W. MOLISZ 1-85

Error Statistics in Data Networks
J. PRIBYL, I. VAJDA 1-95

An Algorithm for the Design of Multilevel Concentrator Networks
G.M. SCHNEIDER, M.N. ZASTROW 1-104

Software Reliability Optimization in Distributed Information Control System
S.G. TRIFONOVA, V.A. TZANOV, R.T. TRIFONOV 1-128

O & M Experience Gained in the Introduction of ISDN
F. BLOME, A. HAMICH 1-137

SESSION 2 - MESSAGE HANDLING SYSTEM AND DIRECTORIES

Standards for Message Handling and their Impact
W. TIETZ 2-1

Extensions and Evaluation of Method for MHS Interwork
Y. SAITO, M. TERAMOTO, K. UEGAKI 2-21

Approaches to the MHS Message Store Structures
V. SABEV, F. FOERSTERLING, N. KLEHN 2-31

A Simple Multi-Purpose Internet Directory
N. MEULEMANS 2-41

Directory Names and Schema - An Evaluation
K. JAKOBS 2-53

Using Directory Services in Mobile Communication
K. JAKOBS, F. REICHERT 2-63

SESSION 3 - MANAGEMENT AND MONITORING OF NETWORKS

A Traffic Generating and Monitoring Technique for Standard and Integrated Voice/Data Ethernet LANs
GY. CSOPAKI, T. MOLNÁR, CS. SZABÓ, L. SZANDI 3-1

Network Management in a Heterogeneous Computer Centre
D. GOLLNICK, H. LAMPRECHT 3-9

An Experimental OSI Network Management Implementation for a PC Network
K. GARBE 3-20

Central System Support for Digital Switching Systems (CSS)
K. HOLZINGER 3-28

SESSION 4 - PROTOCOL SPECIFICATION AND VERIFICATION

Formal Specifications for Protocols: Issues and Experiences
G.v. BOCHMANN, L. LOGRIPPO, B. SARIKAYA 4-1

Protocol Validation Using Process Algebraic Recursive Equations
J. DE MEER, R. AHOOJA 4-20

A Synthetic Localized Approach to Protocol Validation
K. SALEH, R. PROBERT 4-30

Liveness Verification as a Tool for Protocol Validation
L.ZS. VARGA 4-41

Experience in Using Estelle for the Specification and Verification of a Fieldbus Protocol: FIP
M.L. BARETTO, J.-P. COURTIAT, P. DE SAQUIS-SANNES 4-51

SESSION 5 - PROTOCOL IMPLEMENTATION AND TESTING

Experience with the Astride Testing Approach
O. RAFIQ, R. CASTANET 5-1

Experience in Conformance Testing with the Test System Tekos
U. BUHRANDT, V. HEYMER, J. HINKELMANN, P. TOLLKÜHN 5-11

Sequence Generation from EFSMs for Protocol Testing
A. FARO, A. PETRENKO 5-21

Conformance Testing System for the KASK Network: Some Implementation Problems
A. DUBIELEWICZ, I. DUBIELEWICZ,
W. KOMOROWSKI, J. RATAJCZAK 5-33

Comments on TTCN Implementation in the Kask Network
I. DUBIELEWICZ, K. KOLESNIK 5-42

Protocol Software Engineering Tools for Implementation of a General Purpose OSI Stack
A.P. AHTIAINEN, J. KESKINEN, J. SIMOLIN,
K. TARPILA, I. TURUNEN 5-48

The Pre-Implementation and Verification of LOTOS
E. NAJM, J. QUEIROZ, A. SERHROUCHNI 5-59

Functional and Performance Prototyping Concepts Based on SDL
D. CARL 5-76

Experience in Computer-Aided Protocol Implementation
H. KOENIG 5-90

Network LOPOS - Performance Analysis
J. PISTELAK, M. SAREK 5-100

SESSION 6 - KNOWLEDGE BASED SYSTEMS IN COMPUTER COMMUNICATION

Meta-Management of Communications Networks
B. MEANDZIJA, W.P.-C. HO 6-1

Distributed/Cooperative Office
L. CSABA, E. CSUHAJ-VARJÚ 6-12

SBS: A Multimedia Service Base System Based on Distributed Objects
Q. HE, H. TANAKA 6-22

Frames for Protocol Representation
K. TARNAY, S. WÁGNER-DIBUZ, P.A. WÁGNER 6-32

SESSION 7 - DATA PROTECTION AND SECURITY

Computer Networks, Privacy Protection and Security
H.P. GASSMANN 7-1

Issues in Computing Security and Telecommunications Network Integrity
J. FREESE 7-7

Aspects of Security, Integrity and Privacy in the Telecommunication Network
O. AMBROIS 7-15

SESSION 8 - MAJOR CORPORATION IN NETWORKING

SNA Usability Enhancementss for the 90's
L.M. RAFALOW 8-1

Foundation for Future Networking
D.A. TWYVER 8-11

Fifth Generation Private Wide Area Networking
V.H.C.M. EVERS 8-23

Author Index
List of Reviewers

PLENARY PRESENTATIONS

INTEGRATED AND MANAGED TELECOMMUNICATIONS NETWORKS - THE NEXT DECADE

D M Leakey
Chief Scientist, British Telecom plc, UK

The major forces continuing to drive the rapid changes in telecommunications include the advances in technology, the changing regulatory scenarios, and the service demands of customers, particularly the large international business customers. Due primarily to the inertia inherent with large systems, this service demand is likely to outstrip the ability of public service operators to update networks in a uniform manner. It is therefore appropriate to plan for a multi-service network environment with many specialist networks to meet specialist needs. Accepting the pressures for a more competitive environment in the provision of enhanced services increasingly on an international basis, will lead to a need for closer co-ordination between national regulatory regimes to provide an adequate international framework. In this environment coordinated network management will be vital to provide an acceptably flexible and fault free end to end customer service.

1 INTRODUCTION

Three major forces are continuing to drive the changes in telecommunications experienced over the previous decade. First there is the driver of technological innovation. This driver is well publicised, reasonably well understood and likely to continue unabated, but possibly in forms difficult to predict.

The second major force is associated with the regulatory structure in which the telecommunications service industry operates. In this area there is more confusion and uncertainity although there appears to be an increasing trend to take much of the service provision out of the control of public monopolies and place it elsewhere on a competitive basis. This trend is accompanied by considerable re-regulation (not de-regulation as is normally assumed) initially on a country basis, although ultimately the more international issues must be tackled.

The third major driver is that associated with market pull. Dividing the market up into large, usually international, business customers, medium to small business customers and residential customers, it is the large business customers who are exerting the maximum pressure for new services to support their need for improved international coordination.

However services introduced initially for the large business market then filter down to the medium/small business and residential sectors, where solutions originally based largely on private circuit networks become less acceptable economically. Thus the public switched networks need to be modernised to meet this spread of service application. However this is only likely to be possible to timescales unacceptable to meet many of the continuing business needs and there will continue to be a gap between the demands of the large business customers and the capability of the basic switched public networks, a gap which could even widen with time particularly on an international basis. To fill this gap it is likely that there will to be a growth of international special service networks, encouraged or discouraged by the prevailing regulatory regime.

Other network trends are also evident. For example there is the emergence of separate networks to meet the needs of mobile communications where the basic public networks are deficient in control and signalling power.

To counter the trend of separately operated advanced service networks, several public operators are responding by introducing their own overlay special service networks. Thus the acceptance of the need for a multi-service network environment appears to be achieving general acceptance. The purists might argue that this is unnecessary since a single integrated services network is theoretically possible. However it is relevant to note that the foundation for this integration is now moving from current ISDN proposals to the considerably more advanced Broadband ISDN approach. In the meantime customer service demands must be met almost certainly to timescales where this fully integrated approach is simply not practical.

2 THE EVOLVING SCENARIO

Given that whatever the nature of the technological advances and the service demands of customers, there will remain a limiting inertia associated with large universal service public networks, then what will be the areas where the major incompatibilities continue to be evident. Examples include:

* The basic incompatibility between interactive speech handling and the requirements to deal efficiently with transaction communications.

* The need for enhanced centralised control on a national and even an international basis coupled with the need for advanced signalling.

* The growing incompatability between high speed local area networks and the available public network facilities for efficient interconnection.

* The rapid financial payback for improvements designed to meet specific needs given the availability of only a common basic transmission infrastructure.

* The benefits of size outweighed by the benefits of easier and more rapid change.

* The extended use of efficient communications increasingly as a vital element of business.

The first three issues are essentially technical in nature. Transaction communication involves connections of usually very short duration, but with the number of required connections possibly already well in excess of those associated with speech. Examples include reservation systems in variety, creditcard verification, remote cash dispensing, and a large number of other essentially financial or stock control transactions. Normal public exchanges are not dimensioned to cater for this type of traffic and could be subject to severe overload if it were transferred from private circuits to the main switched network.

The need for enhanced centralised control and enhanced network signalling capability is particularly evident with cellular mobile systems. Not only does the network need to maintain records of the locations of all terminals in a standby mode, but connections to active terminals must be transferred from cell to cell in accordance with customer movements. To achieve such centralised control not only requires the associated extra equipment, but suitable ports and application software must also be added to the local exchanges. At present this can be a difficult in service upgrade.

Local area networks now operate at speeds in excess of ten megabits per second. The method of operating by which one terminal has full availability of the complete bandwidth for a very short time, means that LAN interconnection is much simplified if the associated interconnection network functions in a similar mode. Existing public networks are likely eventually to follow a similar approach using some form of asynchronous transfer mode (ATM), but the widespread adoption of ATM must be many years away. However the demand for more efficient LAN interconnection

is already arising which will have to be met using separate networks operating over very high speed private circuits. The second three issues are more commercial in nature and combine to emphasise that the rapid response to specialist customer demand can be very cost effective even if a specialist network is required. Commonality with other networks need only be limited to the use of an existing basic transmission infrastructure.

With this background it would appear that the next decade will be characterised by a growing multi-network scenario where the various service networks will:

* share common transmission media.
* be interconnected as required to meet customer service demands.
* need to conform to appropriate interface standards.
* be provided increasingly competitively to encourage adequate response to customer demand.

Probably equally significant will be the likely need to accept the increasing dominance of selected international carriers. At the present time the telecommunications scene can be characterised as a series of national networks interconnected by lower functionally international links. However this scenario could be incompatible with the rapid internationalisation of business. To meet this challenge the telecommunications scenario could in future be characterised by a limited number of international networks on to which the national networks are connected as local adjuncts. Such a scenario would require a more internationally co-ordinated regulatory regime and improved uniformity regarding the definitions of regulated basic services and competitive enhanced services particularly in regard to data. These requirements might be difficult to meet, but customer pressure must ultimately force adequate agreement.

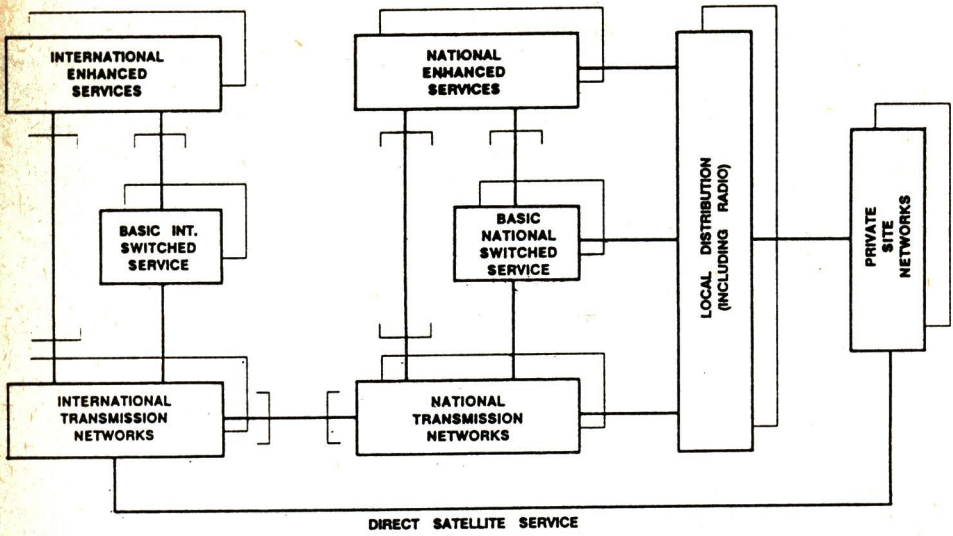


Figure 1: Global Network Structure

Given this scenario, the global telecommunication complex would appear as illustrated on figure 1. To the left the regulated international networks with commodity transmission capabilities and basic switching facilities form the core on to which competitively supplied value added/enhanced services can be mounted. National provision would also comprise regulated commodity transmission capabilities and basic switching facilities on to which national enhanced services could also be mounted. The local distribution network will be a rapidly evolving combination of copper cables with more modern optical fibre and radio methods taking an increasing share of the traffic. For narrow band speech and data communications, radio connection could gain rapidly in popularity as the full potential of personalised communication becomes increasingly apparent. The private site systems, ranging from simple multi-telephone residential systems to complex multi-service business systems, would be connected via the national local networks or by satellite links effectively bypassing the national operators. Such bypass might be resisted strongly by national interests, but the reality of internationalisation must ultimately overcome such objections.

3 CUSTOMER SERVICES MANAGEMENT

Overall customer services management is a vital but often poorly provided feature of telecommunications service provision. End customers expect services to be easy to

access and once established to be reliable. When failure or congestion does occur, understandable explanations should be provided and service restoration should occur rapidly.

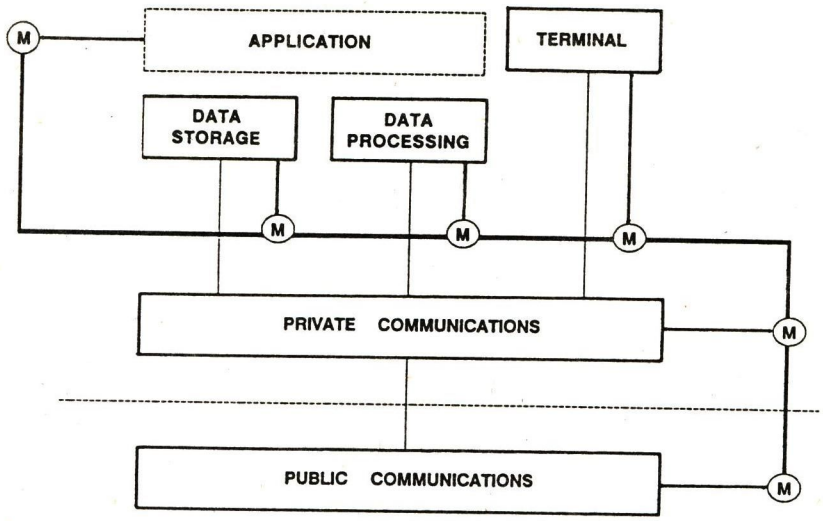


Figure 2: Total Services Management

Unless such overall management is provided in the public domain, large customers will continue to make use of only the minimum level of public service, that is private circuits.

The end customer service management divides into several constituent elements. As illustrated in Figure 2 the elements comprise:

* Customer terminals.
* The applications.
* Data Processing and Storage equipments.
* Local private communications.
* Public Communications.

In the case of direct speech communications, the service management task is simpler, but the need remains to co-ordinate terminal, local communications and public communications management.

The public telecommunications management task can then be sub-divided into:

* Transmission and basic switching management.
* Special Network Features (eg 800 service) management.
* Enhanced services management.

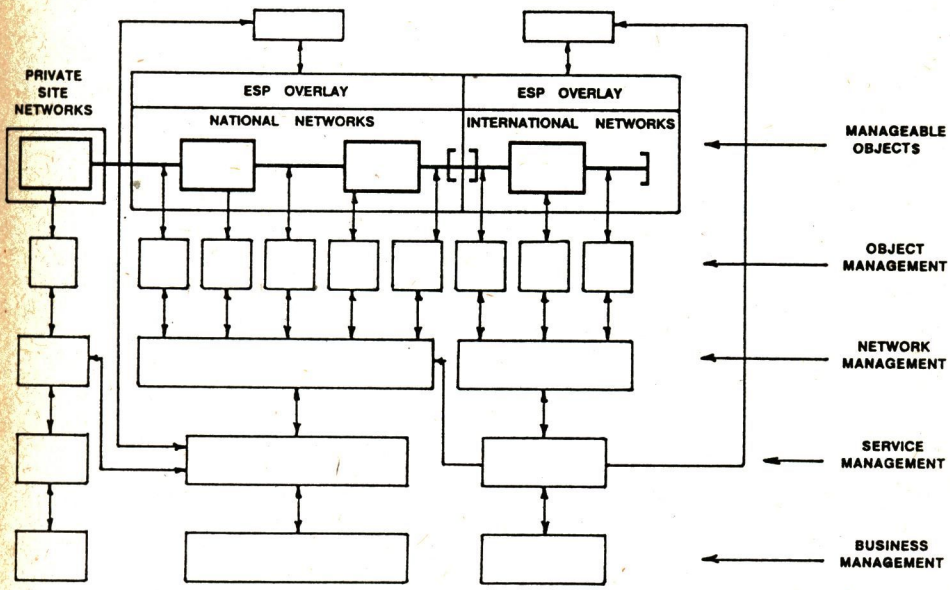


Figure 3: Basic Services Management

In turn the transmission and basic switching services management subdivides into a structural architecture as illustrated in Figure 3.

The basic network is considered to comprise a series of manageable objects which can be physical (eg transmission links) or functional (eg customer data). The first management layer is concerned with managing these objects either individually or in groups (eg modems). The next

layer is concerned with managing the network performance, fault and congestion management etc.

Customers of the network should not have to become involved with the internal performance of the network. Their sole interest should be the performance as seen from the peripheral interfaces. Management interconnection to customer systems is therefore via a service management layer. Whether management interconnection should be solely at this service level, stepped between the operator service level and the customer network level as illustrated, or at other levels, remains a continuing debate. Issues of reasonable interface definition suggest the simplest of interconnection, wheras arguments concerning speed of response would suggest otherwise. The need to establish international agreement on such issues, together with the need for appropriate standards, is now well appreciated.

The top level involves management of the business which needs to be tightly coupled into the network and service levels and use the same data structures.

4 REGULATION AND STANDARDS

The more adequate the services provided by the public carriers, then the more likely will the large international customers make use of public services above the commodity tranmission level. To achieve the necessary improvement, operational and interface standards must be agreed at an international level so that variety is minimised to a an adequate level. Complete standardisation is probably unnecessary even if it were practicable, but where different standards do exist, adequate gateways must be provided. Issues that must be tackled include:

* A more international approach to Open Network Architecture/Open Network Provision

* Interface Definitions:

  - Basic Services to Enhanced Services
  - Enhanced Services to Local Networks
  - Local Networks to Private Site Networks
  - Network Service Management to Enhanced Service Management
  - Enhanced Service Management to Customer Systems Management

* Defined service support facilities including:

  - numbering and directory provision
  - charging methods

* Defined network performance including:

  - transmission
  - congestion
  - service reliability and restoration times.

5 SOME CONCLUSIONS

The telecommunications challenges of the next decade are most likely to be based on the probability that the demand for modern communications will to outstrip the practical ability to update universal public networks in a uniform manner. The result will be a scenario of multiple service networks often sharing only a common transmission facility. Although at least in principle the vision of Broadband ISDN based on ATM techniques could avoid the need for multiple networks, customer demand will force the introduction of interim and intercept solutions which once introduced will have a considerable influence on further evolution. The increasing importance of international communications is likely to result in the emergence of dominant international partnerships more able to meet the needs of the large business customers than is currently possible. Pressures to provide advanced services to short timescales will result in the growth of international enhanced service providers. There will be a need to introduce co-ordinated international regulation in regard to interface and service definitions. In the local distribution networks the decade will see increasingly mixed solutions being adopted with copper gradually being replaced by optical fibre and radio.

Most importantly however standards and procedures must be developed so that end-to-end customer service can be managed in an adequate manner even though a connection may be carried over a series of several quasi-independent networks.

High Speed Local Area Networks - What, why, when and how: Planning, installation and first experiences of a HSLAN in a heterogeneous environment

Otto Spaniol
Computer Science Department
Technical University of Aachen
D - 5100 Aachen
(Fed. Rep. of Germany)

Abstract:
The paper begins with a description of characteristics of High Speed Local Area Networks. Following that the pro's and con's of the main standardisation candidates (FDDI, DQDB) are presented together with a comparison of their relative merits. A major part of the manuscript is devoted to the detailed description of lessons learned from a FDDI installation within a heterogeneous environment (the network is to be used as a backbone for the interconnection of university institutes which have different applications as well as different computing facilities). The paper ends with some perspectives and a few speculations for the future of HSLAN development, market potential and acceptance.

1. High Speed Local Area Networks - what is it?

High Speed Local Area Networks (HSLAN) come as a direct consequence of classical and well established 'low speed' local area networks like Ethernet or Token Ring. Much of the situation for HSLAN in the beginning of the nineties is very similar or identical to the situation for LAN at the beginning of the eighties. This includes:

- the high price for the first prototypes (indeed, the cost situation may well be considered to be prohibitive)
- the difficulty to provide acceptance for HSLAN
- the difficulty to answer questions about 'need' as well as about 'return on investment'
- missing or incomplete standards
- the confusion of customers by different HSLAN standardisation activities (how to guarantee that a choosen solution is 'correct' and will have a promising future?).

In addition to that, HSLAN have to cope with problems which are of significant higher complexity than their correponding LAN counterparts:

	LAN	HSLAN
- HSLAN cover a larger geographical distance. Many HSLAN might even be considered as Metropolitan Area Networks (MAN)	2 kilometers	20-100 kilometers
- HSLAN have a substantially higher data rate.	10 Megabit/sec.	100-1000 Megabit/sec.
- Whereas LAN are very often tailored to satisfy the needs of a homogeneous user group, HSLAN are mainly used to inter-connect networks of different type and of very different traffic structure.	homogeneous traffic scenario	heterogeneous traffic scenario

- Moreover, the installation of HSLAN goes together with the need for a solution of tedious and sometimes very hard problems like regulations, tariffs,...; this holds at least for some European countries.

HSLAN come in three different structures:

A. Backend networks	for interconnection between mainframes or between mainframes and high capacity peripherals
B. Backbone networks	interconnection of local area subnetworks (and of some mainframes)
C. Frontend networks	connection of (mainly diskless) workstations with common servers and with a few larger mainframes.

Backend networks are known since many years; the HYPERchannel is a well known candidate of that type.

Frontend-HSLAN networks are still to come. They will surely evolve but in order to be successful the price of an adapting unit must become significantly lower, i.e. it should be a fraction rather than a multiple of the price for the station which is to be interconnected by it.

By far the majority of HSLAN which are installed or which are going to be installed during the near future are backbone networks which interconnect a number of local area networks (mostly of Ethernet or of Token Ring/Bus type) as well as a few larger mainframes. The backbone will be mainly used for file transfer between terminals, workstations, mainframes and specialized servers. In the coming years we may predict that a larger capacity proportion will be used for graphics, voice and even images. However, actually available HSLAN products (which offer a raw data capacity of 100 Megabit/sec and adapters of much lower performance) are not well suited for traffic types which require an extremely high capacity range. It is questionable whether HSLAN should be used (or abused?) for digital voice transmission; this holds at least for most available HSLAN products of today. In the HSLAN project of Technical University of Aachen, we were lucky enough that we didn't have to answer this particular question since voice transmission on HSLAN was forbidden anyway.

The rationale for HSLAN installations will become more obvious during the next few years. Some of the reasons for HSLAN are:

- the ever increasing rate of decentralisation and of specialisation
- the ever increasing complexity of problems and solutions which require the cooperation between different units
- the need for resource sharing (in order to get reasonable investment cost per institution).

All of these reasons are well known from the introduction strategy of LAN. They reappear with higher urgency.

Another fact which (unfortunately) reappears is that HSLAN become rarely possible by giving good arguments for them - and then getting the money from the government or from the headquarters of the institution. The reality is that 'mainframe zealots' are still dominant or at least of big importance. That is to say that the need as well as the cost for new mainframes is accepted and well understood. On the other hand, networks are still considered to be the successor of a few telephone copper cables and nobody is willing to support the idea that maintenance costs or suspicious things like PTT tariffs have to be paid for them.

Experience shows that HSLAN are most efficiently installed as a by-product of a new mainframe. This is the easiest and in most cases the only possible introduction strategy. Nevertheless, you shouldn't expect that the relative amount which will be 'wasted' for the network will correspond to your dreams or to your requirements. In our project (which followed exactly the strategy of being a by-product of a mainframe replacement) the budget for HSLAN was less than 10% of mainframe cost.

2. HSLAN: the standardisation activities

2.1. What can we learn from LAN standards?

LAN's exist in three major types (with respect to media and media access):

a. Ethernet, random access, CSMA/CD:

It is obvious that the random access principle is no suitable candidate for HSLAN. Indeed:

- The principle doesn't make sense for larger distances (since the collision window would be much longer).
- It may not be translated to higher data capacities (the number of data lost after a conflict and, therefore, the minimum packet size would be much too high).
- Random access doesnot really fit to regular traffic structure which may not be excluded or which is, on the contrary, rather typical for HSLAN (especially if graphical services or transfer of large files is envisaged as a major application).

Thus, the random access principle would violate each of the three distinguishing characteristics of HSLAN.

b. Token access.

The token access principle comes in different versions. The best known representatives of token systems for variable packet size are the token ring and the token bus respectively. It is easy to see that the IEEE 802.4 token bus principle is not a very good candidate due to the excessive amount of overhead (indeed, the overhead of this access method is already considerable in the LAN environment!).

On the other hand, the token ring principle is well suited. However, we have to modify the access principle from 'single token' (which makes sense for small and/or slow rings) towards 'multiple token' or 'early token release' (which means that a new free token is generated as soon as possible after the previous message has been transmitted; thus, within a very long and/or very fast ring there may be many messages on the ring at the same time).

One of the two major HSLAN standardisation activitites (namely FDDI-I) is based on the token ring access principle. Due to a lot of additional subtleties (which are - to the opinion of the author - not always very useful but which are rather disturbing and costly and which may even lead to performance degradation) the standard is voluminous and it took much time to implement it; nevertheless, it is based on the 'multiple token' access rule.

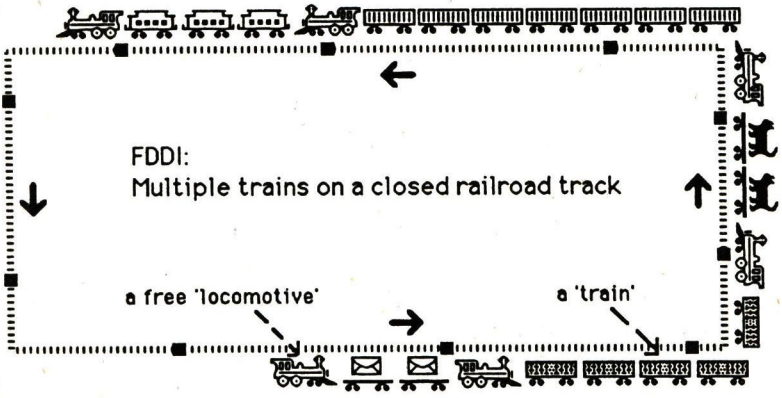


Figure 1: The FDDI access principle

c. Slotted systems.

These systems are special versions of the token access rules. Each station sees a continuous stream of fixed size slots which may be empty (i.e. having a free token) or occupied (i.e. with a busy token bit).

Slotted systems are well suited for regular traffic scenarios. They are best for a perfect interplay with slotted wide area systems of today (ISDN) as well as of the future (ATM,...).

The second HSLAN standardisation activity (DQDB) is based on a slotted token bus scheme.

2.2. Some comments about FDDI, FDDI-II and DQDB.

FDDI (or FDDI-I) is the only standardised version of a HSLAN which already has a larger number of installations (even if some components of the standard, especially questions about station management have not yet been agreed upon). The principle of FDDI is to some extent analogous to a closed railroad track ('a ring') where a fast locomotive is available for the transport of trains of variable size. Several trains (but at most a single free locomotive) may coexist on the ring (see figure 1).

FDDI is very convenient for packet transmission and especially for mixed traffic structures. On the other hand, the concept has drawbacks for stations which require a more or less constant transmission delay (e.g. digital voice).

Another problem with FDDI is that the time during which the 'locomotive' is running from one station to the next station may not be used for new transmissions (and is, therefore, lost from a troughput point of view). The amount of such times may be considered as of minor impact but its influence increases with ring length (i.e. in a MAN situation) as well as with data capacity (i.e. in a HSLAN context).

In particular the first of those problems has resulted in the creation of an activity for an extended standard FDDI-II which essentially adds a slotted structure to the variable length structure of FDDI-I. Thus phases of non-slotted and of slotted access follow each other and the customer should select between both access modes according to his requirements.

The major problems for FDDI-II are:
- it comes too late (at least three years later than FDDI-I, probably it will never come!)
- it is largely incompatible to FDDI-I
- it reduces the acceptance of FDDI-I (since many users wait for the winner of the game)
- it reduces its own acceptance (since reasonable market prices may only be obtained if sufficient FDDI-I installations have been installed and if other potential institutions hear of positive experiences)
- its market segment is annihilated by both FDDI-I and DQDB.

As a consequence, it is more than likely that FDDI-II is dead!

DQDB (Distributed Queue Dual Bus) uses a slotted access scheme which may be compared with two assembly lines going in opposite directions. New containers of constant size are continuously inserted by a slot generating unit at the beginning of each assembly-line. A sender has to learn whether the destination is upstream or downstream and it has to use one or more free containers which arrive on the appropriate assembly-line (see figure 2).

A possible monopolisation by stations which are in the vicinity of the slot generating unit (and which, therefore, see the slots when they are still empty) is avoided by requesting that a reservation has to be made on the opposite assembly-line. If, for example, the reservation counter for the downstream bus has an actual value of M=2 then a station knows that two transmission requests have already been made by downstream stations and that the next two empty slots should be left free for them. By putting a new reservation on the upstream bus the reservation counter of upstream stations will be increased. The transmission will then be effected in free slot number M+1, i.e. in the third free downstream slot.

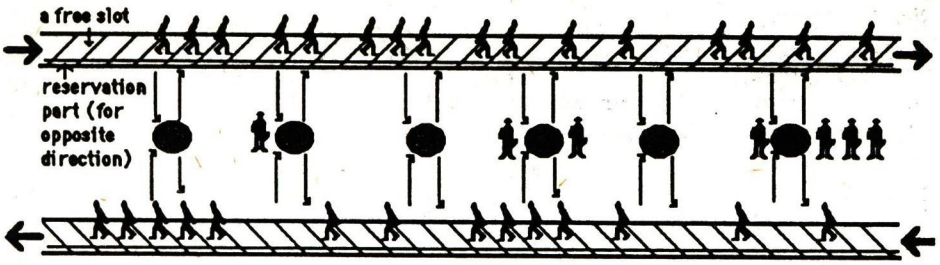


Figure 2: The DQDB access principle

Problems with DQDB are:

- the standard is not yet stable, there are some design flaws to remove (for example, in spite of the seemingly fair access rule there may be situations where some of the stations are favoured against other stations)
- the management of the system seems to be rather complex (continuous monitoring of several counters, two bus access points per station,...); this might result in rather high cost at least for the first installations
- there are no products available! According to the experiences made with FDDI-I we will probably have to wait three more years before mature systems will be on the market.

These problems are in contrast with the potential benefits of the access scheme which is particularly suitable when a combination with other fixed size access schemes (like ATM) is envisaged. This fact was probably a major argument for some PTT's towards expressing a strong preference for DQDB as being their HSLAN concept of the future.

2.3. FDDI or DQDB - what is the better solution?

A first comment: This question should not be asked that way. It is similar to and as useless as the question whether Ethernet or Token Ring is the best (and only?) local area network structure. The answer - for both questions - is that each of the concepts has its own advantages as well as its drawbacks. Each one is suited for a particular situation and probably both will survive and share a significant part of the market.

FDDI comes earlier (this is a trivial but nevertheless very important fact) and has clear advantages for variable traffic patterns. It seems to be the best scheme for 'medium distance' ranges, i.e. for backbone HSLAN serving a large institution whose buildings are dispersed over a city.

DQDB becomes interesting for large distances. Another trivial (but again very important) aspect is that accounting for fixed size slots is easier than for variable message length - this profane fact has been another argument for PTT's in favour of DQDB.

The following strategy is proposed by the author:

- Accept FDDI as being the convenient solution inside a city.
- Create a global HSLAN which interconnects the local FDDI-rings by means of DQDB.

A prerequesite for the feasability of that strategy is that gateways between both concepts become available as soon as possible. The need for that has already been recognised and several manufacturers are already starting the development of such gateways.

3. Planning and installation of a HSLAN in a university environment.

The realisation of a 'HSLAN dream' became possible for Technical University of Aachen at the moment when the discussion of new mainframe computers (a large universal computer as well as a supercomputer) was started in 1986. At that time our 'network' was starshaped and consisted of telephone links of very low capacity for data transmission. It was envisaged already in 1986 to install a FDDI network as soon as possible. FDDI was the only HSLAN which was known that time and it was hoped that (including all the usual delays) reasonably priced products would be available at the time when the new mainframes are installed.

In the following we describe some of the more important directions and redirections of our FDDI planning steps.

a. Topology.
Which of the institutes are 'of sufficient importance', which of the buildings should contain an FDDI-node? The answer of these questions depends on:
- the geographic location of the building
- the networks which are already available or planned for the near future
- the 'degree of importance and readiness' (i.e. does the institute in question really need remote data and/or computation? Does it need a node immediately or - say - in three years from now? Should we adjust the topology by 'speculative' requirements?).

The resulting topology was a (double) ring which:
- has a cable length of approx. ten kilometers
- has a logical ring length of approx. 16 kilometers.
[The cable is shorter than the logical ring since by cost reasons the cabling has been done (over a significant part of the global distance) in form of a 'flat ring': Both sides of the ring are in the same cable which consists of 2x2 = 4 fibres (see figure 3)].

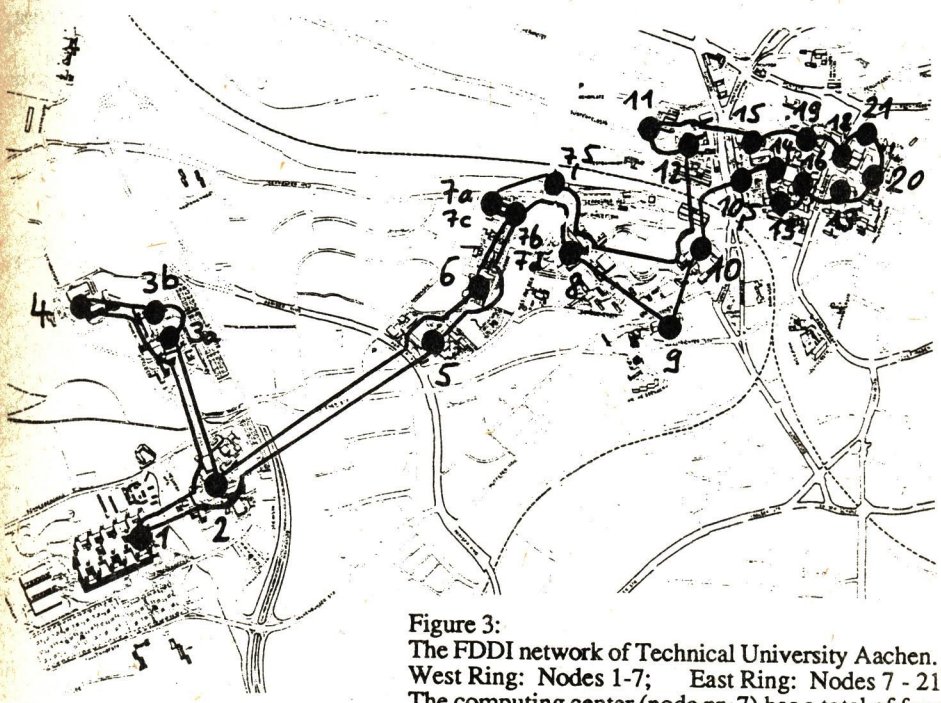


Figure 3:
The FDDI network of Technical University Aachen.
West Ring: Nodes 1-7; East Ring: Nodes 7 - 21.
The computing center (node nr. 7) has a total of four FDDI stations.

b. Tariffs.
At first, this problem seemed of insurmountable complexity: The regular tariffs by German PTT applied to links with a capacity of 100 Megabit per second were completely out of our financial possibilities.
Fortunately, we could reach a very positive agreement with the PTT which became rather famous in Germany and now serves as a basis for similar contracts.

The basis of contract is that the PTT provides the cabling (also for those parts of the network which do not fall under the PTT regulations) and that the university
- pays the cost of cabling
- pays relatively limited tariffs for using the cables.

Both parties agreed that the contract is a common project. This statement was necessary in order to get a special and unusual tariff agreement.

c. Modifications.
From the very beginning the planning had to be modified almost continuously due to changing situations. This was caused by removal of institutes, by the installation of new computers or networks in some buildings and by many other reasons.

The number of nodes increased from the beginning of planning until the beginning of the installation from 21 to (actually) 27.

A special problem (especially for our budget) was inhouse cabling since the PTT contract was only valid for the cabling between the outside walls of the buildings.

d. Installation strategy.
Cabling by means of a 'flat ring' (i.e. 2x2 fibres within the cable) due to economy aspects has a severe drawback: Indeed, in case of a damaged cable it is likely that the ring will be divided into two halfs and that, therefore, a substantial part of the communication facilities will be annihilated; the 'wrap around' mechanism (deviation via the security ring) will not be possible any more since it is very probable that all the four fibres will be damaged.

On the other hand, the cabling economy had an unexpected advantage: it allowed to open the ring in several phases. This was very convenient due to the following reasons:
- the budget for FDDI nodes came in two parts for two years (1989 and 1990)
- the 'cold start' with an installation of all 27 nodes simultaneously would have been very difficult to manage.

For these reasons it was decided to install the ring as two 'half-rings'. The so-called west-ring covers the region west of the computer center and contains 9 nodes whereas the east-ring consists of the remaining 18 nodes. Both rings became operational in October 1989 and in March 1990 respectively.

It was particularly convenient that the interface between both rings, namely the computer center, is situated approximately in the geographical center of the university buildings and that by far the majority of the traffic (at least in the first years) will be directed to and from this node which has connections to both half-rings.

The strategical decision in favour of half-rings had another two positive consequences:

1. The raw data capacity is doubled since full capacity is available for both half-rings.
2. It is easier to install products of different vendors in separate parts of the global FDDI system. Thus **interoperability** is facilitated. In our network we intended to install components from different vendors within the east-ring and within the west-ring. Unfortunately, this was not possible due to ever growing lateness for product delivery time of one of the vendors. Nevertheless, the positive impact was that we got an interoperability guarantee (thus the risk of overpricing for additional nodes was considerably reduced) and that we could specify the nodes which are still to be delivered in such a way that their performance is 'the union of promises made by different vendors'. We will have to check whether this may become reality since it is expected that some of the required performance parameters will be hard to achieve!

e. Actual status and perspectives.

As of today (March 1990) all but five nodes are installed in a preliminary version which only allows for an interconnection of Ethernet subnetworks or of a channel coupling for IBM compatible mainframes.

The facilities for token ring interconnection (4 Megabit/sec. as well as 16 Megabit/sec.) are scheduled for May 1990. By the end of 1990 we should reach the (preliminary) end of the FDDI installation. At that time, the node structure will be as follows (provided that the requirements of the contract will be fully satisfied!):

- X.25 interfaces for some of our FDDI nodes.
- Increase of the number of interconnected Ethernets (up to 8 per node).
- Replacement of all nodes which will eventually need more than two subnetwork interfaces by a 'modular' version of higher performance. According to our contract a modular FDDI node will have a maximum of 8 interfaces which may be (almost) freely mixed between different subnetworks. At the moment, however, the vendor tries to convince us that this requirement is neither reasonable nor satisfiable and that a node with eight interfaces should be replaced by two nodes with four interfaces.
- In the first phase all the nodes are operated as bridges (i.e. interconnection at layer 2). The bridging priciple is easy and relatively performant but excludes interoperability (if different frame encapsulation schemes are used) and complicates communication between different subnetwork types (since different layer 2 addressing schemes are used for example in Ethernet and in token ring). As a consequence, we intend to replace the FDDI bridges by routers (which operate on layer 3 addresses). This will be in effect by the end of 1990.
  Routing will be available for a variety of protocols (including TCP/IP, DECnet and ISO). In order to facilitate the interconnection of different vendor components, the routing process itself will be based on different protocols such as RIP or EGP.

f. First experiences.

Only a minor part of users were prepared to use the new network intensively from the beginning. These users are quite satisfied: no bottleneck (which might be attributed to the network) has become visible until now. Of course, this is due to several reasons:

- the limited amount of users which are <u>really</u> active
- the concentration towards non-graphical traffic streams.

The network became operational in a surprisingly short time. Apart from some tedious but unavoidable problems (such as incompatibility of new units with already existing units and corresponding waiting times until the remedy of this situation) the ring is stable.

The cooperation with the vendor is not quite in accordance with the enthusiastic promises made beforehand. We are not too much disappointed about that since we knew from the beginning about the different situation before and after the signing of the contract.

4. High Speed LAN - the need, the trends and some speculations

Local communication systems come in a variety of alternatives. A classification by their geographical coverage and their raw data capacity as parameters shows that at least four types of networks may be distinguished from each other:

- 'Capillary' networks with very low speed
- 'Classical' LAN such as Ethernet or token ring
- High Speed LAN such as FDDI or DQDB
- Very High Speed LAN.

In this context, the distance-independent TELECOM activities such as narrowband ISDN, wideband ISDN or (later) ATM should not be forgotten. Most of the high speed or very high speed LAN are still in a project stage or at best in a prototype phase. It is to be expected that only a very small part of the proposals will survive and become available as a product.

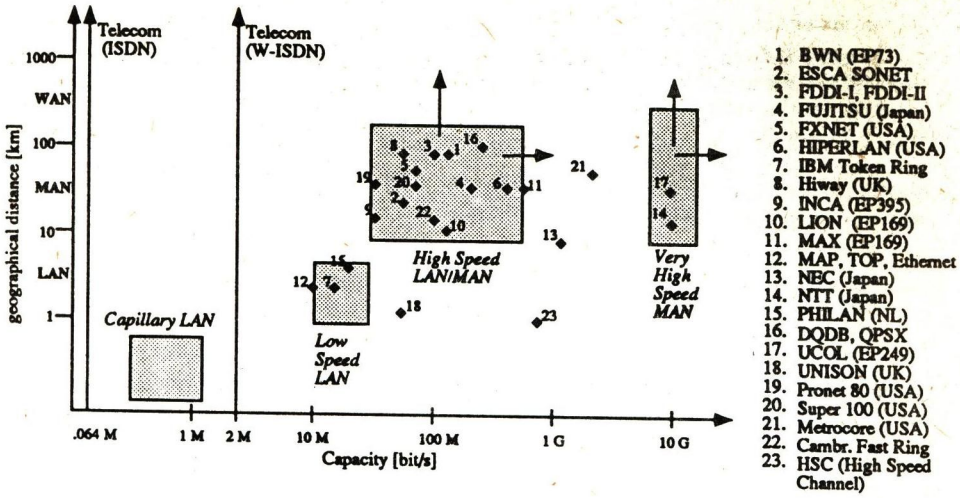


Figure 4: High Speed Networks - Projects and Products

EP = Esprit Project

For High Speed LAN we expect that:

- in the short-term (until the end of 1990) they will be used mainly or exclusively for data transfer; the majority of those systems will be of FDDI type
- in the medium-term (until 1993) users will become more and more familiar with service integration (i.e. voice and graphics will be added to the data transfer). It is to be expected that new products and standards will appear (DQDB, HSC, Metrocore,...) and that the network capacity - at least for some prototypes - will eventually reach or overcome the Gigabit/sec. threshold.

There are some favourable conditions for a rapid increase of HSLAN installations:

- The number of 'classical' LAN becomes higher and higher (according to some market reports, the increase is in excess of 20% per year, i.e. it is exponential!)
- An ever increasing percentage of LAN installations will not stay in isolation but become interconneted to other networks and/or mainframes via (high speed) backbone networks.
- New workstations with higher performance and with higher communication requirements (e.g. for CAD/CAM use) will be affordable.
- New HSLAN installations will come almost automatically by reason of:
  - positive experiences reported by the owners of first HSLAN systems
  - competition between different institutions (e.g. between universities) in order to occupy or maintain a leading role in new technology
  - easier operation and simplified management
  - new hardware/software components, new applications, new services
  - reduced prices due to higher number of installations.

All these arguments are exactly in accordance with the observations made during LAN introduction in the decade 1980-1990.

Finally, my (very subjective and hopefully too pessimistic) view about the only major risk for a successful introduction of HSLAN is the uncertainty about the political situation. When political systems come nearer to each other then there will be an obvious and justified need for a better balance of different infrastructures. This might result in other priority settings than just maximising the number of HSLAN installations - at least for a period of transition which might last a couple of years.

5. Summary

The balance sheet of our FDDI installation leaves very mixed impressions:

Frustrating experiences:

- The effort:
  a lot of hours has been invested for planning (and - after all - the resulting configuration looks very simple or even trivial).
- The negative comments by some users :
  - who didn't accept that it took so long to install the network
  - who critisized some decisions about products or introduction strategy (for example, it was not difficult to predict that users which belong to the East-ring had some complain about the decision that the West-ring came earlier)
- The restrictions:
  The most innovative concepts could not be realized. We had to accept that first priority is given to stable operation of the network. Dreaming of new, beautiful, but risky systems is dangerous and, therefore, forbidden in the context of a computing center.

Positive experiences:

- The effort (once again!):
  After all, a configuration has been obtained which momentarily belongs to the largest FDDI installations worldwide. All of the planning and of the realisation was done by a very small group of people, i.e. at extremely low cost.
- The positive comments:
  - mainly coming from other institutions who are in a similar situation and who know about the numerous difficulties.
- The satisfaction:
  - we learned a lot of - mostly nontechnical but nevertheless very important - things; most important for us was that we learned to distinguish between 'science' and 'science fiction'
  - the realisation of a 'dream' was possible despite a lot of very gloomy predictions.

6. References

HSLAN is a rather new topic for research and development. In order to become familiar with the most recent results it is highly recommended to begin with some monographs which are entirely devoted to the topic and to make 'backwards referencing'. Such monographs published as a result of IFIP WG 6.4 conferences in the period 1987-1989 are:

Spaniol, O., Danthine, A.:
High Speed Local Area Networks. Proceedings of First IFIP WG 6.4 workshop on HSLAN, Aachen 1987. North-Holland 1988.

Danthine, A., Spaniol, O.:
High Speed Local Area Networks II. Proceedings of Second IFIP WG 6.4 workshop on HSLAN, Liège 1988. North-Holland 1989.

Rudin, H.:
Protocols for High Speed Networks. Proceedings of First IFIP WG 6.1/6.4 workshop on HSLAN protocols, Zürich 1989. North-Holland 1989.

Moreover, there are numerous contributions in journals such as 'IEEE Transactions on Communications', 'IEEE Selected Areas on Communication', 'Computer Networks and ISDN Systems' etc.

Finally, there is abundant material of standardisation documents which is available from the respective standardisation bodies.

TELECOMMUNICATION STANDARDIZATION IN WESTERN EUROPE

Hermann Gabler, Deutsche Bundespost Telekom

INTRODUCTION

Transfer of the European standardization work from CEPT to ETSI, the European Telecommunications Standards Institute, in spring 1988 has considerably changed the European as well the national standardization scenes in the individual countries. Common standards are vital for the completion of the internal market foreseen for the end of 1992 and for the creation of a Europe-wide telecommunications system. This new situation compels the Europeans to discontinue purely national standardization work. The old way of making international standards through national standardization bodies is superseded.This procedure is a detour which causes loss of time and leads enherently to additional confrontation in the international bodies. In future every new telecommunication standardization project has to be regarded in a European perspective. The European standard has to become the "national" standard; bluntly stated, with the setting up of ETSI the German DIN will publish no national telecommunication standard anymore. Future DIN-ETS (European Telecommunication Standards) are standards which will have no national predecessor.

A consequence of having common telecommunication standards in Europe is the new policy of the Commission of the European Community (EC) of mutual recognition of type approval of IT/T products with the aim to come to a common approval system within the countries of the EC. We have already reported about the Conformance Testing Programme in a previous COMNET. This paper summarizes briefly the achievements of this project and gives a survey of the ongoing activites in the newly founded European Open Systems Testing Consortium OSTC. Europe closes up between Madrid and Moscou from the economic and political points of view. Germany is the most important transit country and gateway to Eastern Europe. I deduct from this fact the justification to invite you to join the activities going on in Western Europe.

BACKGROUND TO THE FORMATION OF ETSI

In the area of telecommunication and information technology, worldwide standardization takes part in three organizations; these are ISO (International Standardizations Organization), IEC (International Electrotechnical Commission) and CCITT/ CCIR. As a link between the national standardization bodies and the aforementioned international bodies in Europe but as well in Asia and Northamerica regional standardization bodies have been established. On the one hand, they aim at

making special, only regionally needed standards and on the other hand they care for the transposition of world-wide standards into regional standards possibly eliminating options. In Europe those regional organizations are CEN (Comité Européen de Normalisation) for the area of ISO and CENELEC (Comité Européen de Normalisation Electro-technique) for the area of IEC. CEN and CENELEC are recognized by the Commission of the EC and they have contractually fixed responsibilities. For the area of CCITT/CCIR hitherto no such regional organization existed. It is true that in the field of telecommunication CEPT had accomplished some coordination and published "recommendations", which had about the status of a European standard like those published by CEN/CENELC. But work was nearly exclusively carried out by the telecommunication administrations. The wide public had no chance to actively participate. CEPT working procedures were slow. Achieving a consensus was a time-consuming process as no precise voting procedures existed. Recommendations therefore contained very often many options which reduced the value of the standard in practical work.

Standardization bodies have to comply with certain prerequisites to become acknowledged as an official European standards organization. Those conditions are

- transparency. Standardization work is open to all parties concerned.
- public inquiry. The public must get a chance to comment draft standards.
- standstill period. Parallel national standardization on on the same subject must quit.
- transposition. The National Standardization Organizations (NSO) have to transpose the European Standard in their country and to withdraw possible opposing national standards.

As CEPT complied with none of these prerequisites they could not as recognized standardization organization become acknowledged by the Commission.

At the beginning of the 1980's it became clear within the CEPT that changes were needed for the production of desired technical specifications for the telecommunications area. An increasing pressure was felt because of the accelerating development and it became obvious that a closer and more balanced co-operation with other interested parties, i.e. industry and users, was needed. Some progress was made within the framework of CEPT, but it became clear that more drastic changes were needed. A further inspiration came in 1987 with the "Green Paper" issued by the Commission of the European Communities. In the Green Paper, the Commission stated that if the European Single Market is to be a reality by the end of 1992, it was necessary for telecommunications products to flow freely across the borders in the same way as foreseen for other goods. To this end a standardization must be established.

The background for this statement is twofold:
- Firstly the telecommunications equipment is a substantial part of the goods purchased in the European countries. Thus the idea of free movement of goods would make little sense if such a substantial group of goods were not included.
- Secondly the sector of telecommunications is of vital importance to Europe in the sense that no modern society can function satisfactorily without a competent telecommunications sector and hence a strong and competitive telecommunications industry.

Against the above background the proposal of the Commission was in favour of an European institute with a flexible and modern organization facilitating the elaboration of telecommunications standards. Three leading principles should form the basis of this institute:

- Firstly the standardization should be made more quickly than hitherto in order to get substantial results before 1992.
- Secondly the principle of consensus, which had hitherto been governing, should be replaced by the principle of weighted voting corresponding to that of the Treaty of Rome Article 148.
- Thirdly the participants should not only be the PTTs but also the industry, the public network operators, the users including private service providers and also some research bodies.

The governing bodies of CEPT responded amazingly quickly. Thus after a deep and complete analysis of the standardization situation in CEPT, within a very short time CEPT decided to create an autonomous institute for the European standards (ETSI) and to establish drafts of the Statutes and Rules of Procedure for ETSI. Very shortly afterwards it was agreed that the site of ETSI should be the International Science Park of Sophia Antipolis near Nice.

The inaugural General Assembly meeting took place in Sophia Antipolis on the 29 and 30 March 1988.

ETSI-STRUCTURE AND ORGANIZATION OF WORK

ETSI's area of competence is listed in § 14 of the rules of procedure.
ETS shall be elaborated in fields which include:
- telecommunications terminal equipment including interfaces to networks,
- areas common to telecommunication and information technology according to agreements made with CEN/CENELEC or common to telecommunications and broadcasting, or
- areas which embrace the telecommunications networks themselves and other areas such as radio communications.

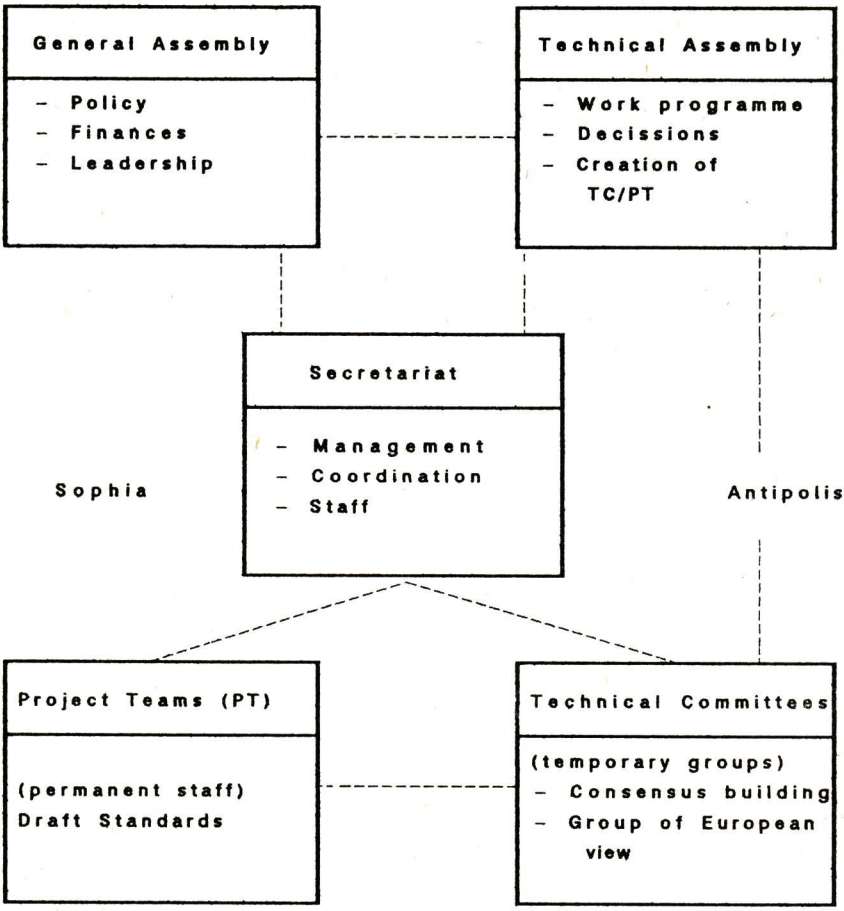


Figure 1
ETSI organization

Fig. 1 shows the organizational structure of ETSI. Executive body is the General Assembly (GA) with the final responsibility for the budget. Standardization work is controlled and directed by the Technical Assembly (TA). The Technical Committees (TC) - presently 14 - perform the factual standardization work. Meetings of TCs or Sub-TCs are normally hosted by ETSI members in their relevant countries. TCs submit the outcome of their work as draft ETS to the National Standardization Organization (NSO) for public inquiry. The final authority for acceptance of an ETS after public inquiry is TA through weighted voting. Every member country has one vote (with weights between 1 to 10). This implies that national members have to agree on their national position prior to voting in the TA through their head of delegation. 71 % affirmative votes are needed to carry an ETS.

To accelerate standardization work, Project Teams (PT) may be established. They work in small groups of up to 6 experts. Terms of reference and duration of work, normally a few months, are succinctly given by the TC concerned. Experts in PTs work full time or part time, they are provided by ETSI members and their salaries and expenses are financed by ETSI.

Members of the institute are generally European organizations, however, everybody concerned as an individual, company or organization within EC or EFTA may become member. Observers from Europe may participate with the right to speak but without the right to vote in the GA and the TA. With the agreement of the TA the observers may also participate in the TCs. Also invited observers from outside Europe participate in TA and TC meetings (with reciprocal invitations for ETSI members to attend meetings in their organizations).

In Febr. 1990 ETSI consisted of 194 members and about 30 observers. The participation of the different European countries is given in Figure 2.

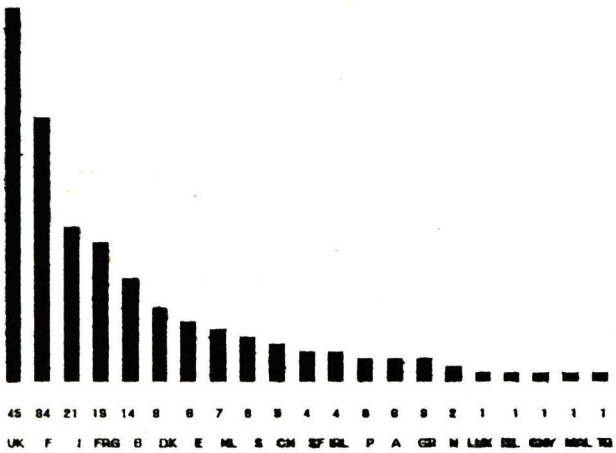


Figure 2
ETSI members per country (as of Febr. 1990)

Figure 3 (next page) shows the ETSI-TCs and their responsibilities. More than 1000 experts are currently listed as active participants.

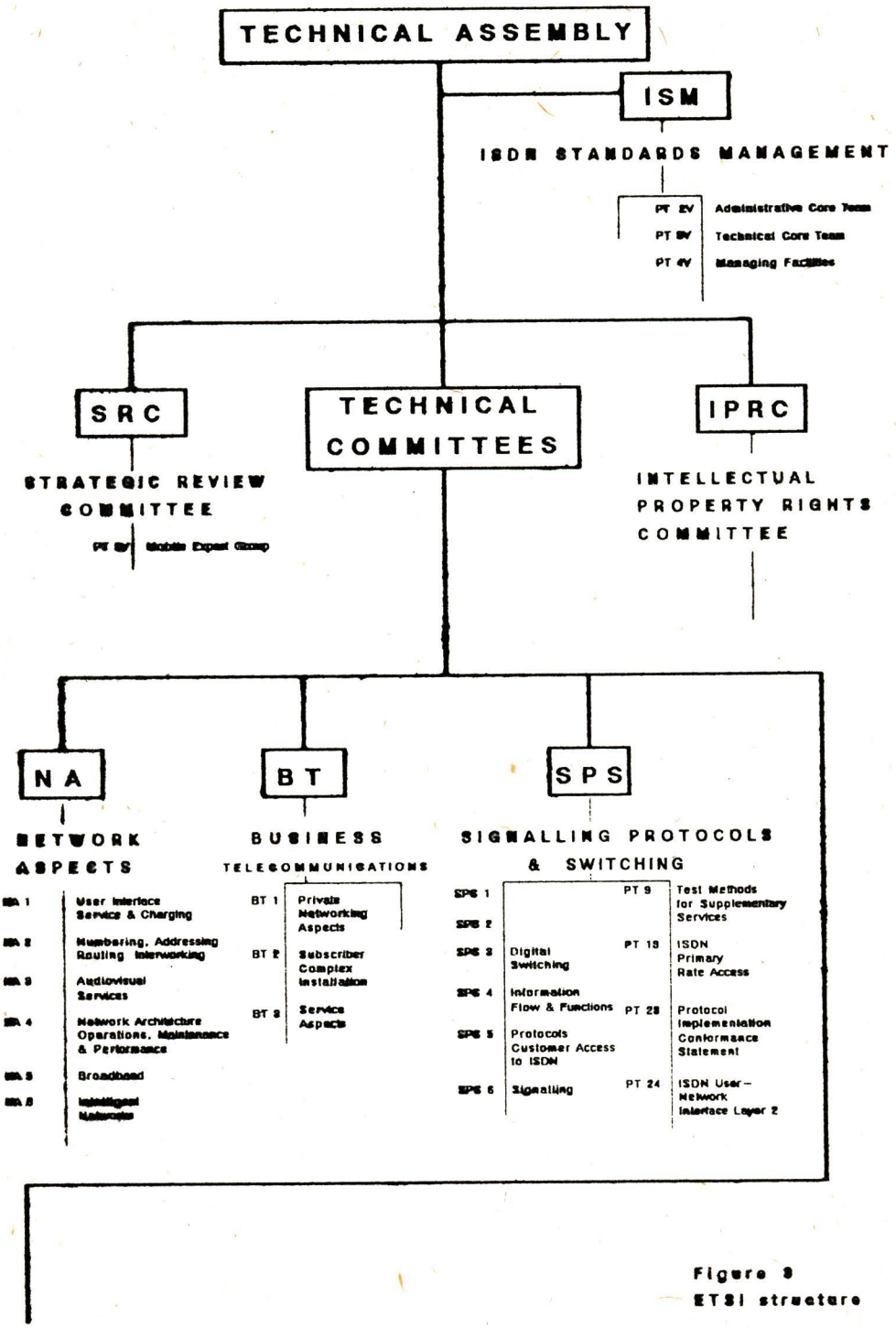


Figure 9
ETSI structure

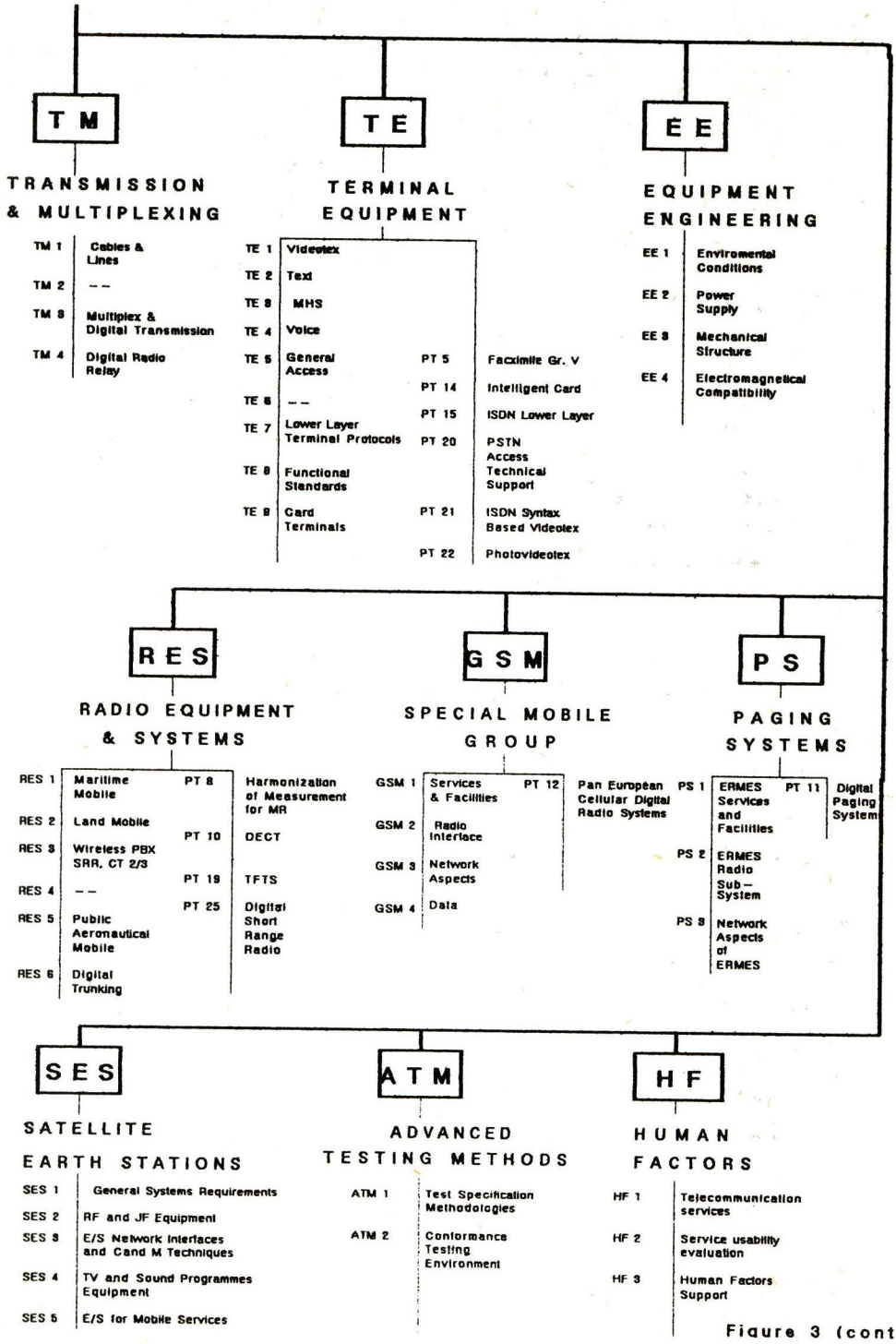


Figure 3 (cont.)

Table 1 demonstrates the working areas of the project teams and the working capacity invested. The about 700 man-months foreseen with about 15.000 ECU per manmonth underline the emphasis which Western Europe places into telecommunuication standardization. The output from these efforts is considerable.

			Manpower in Man-months		
			1989-92	1989	1990
PT	1	ISDN Basic access (NET 3 candidate)	4.5	1.3	
	2	PSTN access (NET 4 candidate)	23.5	18.1	
	3	ISDN Terminal adapter (NET 7 candidate)	13.5	13.3	
	5	Facsimicle G 4 (NET 20-24 candidates)	21	20	
	6	X.32 Access	25	0	
	7	ISPBX interconnection matters	38	29.5	
	8	Harmonization of measurement for equipment for private mobile radio services	18	14	2
	9	Test methods for supplementary services	12	10	
	10	Digital European cordless telephone (DECT)	120	20	70
	11	Digital paging systems	23	9	14
	12	Pan European cellular digital radio systems	96	25.5	72
	13	ISDN Primary Rate Access (NET 5 candidate)	2	2	
	14	Intelligent cards	16	3	13
	15	ISDN lower layer protocols	6	6	
	19	Terrestrial Flight Telephone Systems (TFTS)	100	6	24
	20	PSTN Access Technical Support	2.5	1.5	1
	21	ISDN Syntax-based Videotex	20	12	8
	22	Photovideotex	21		21
	23	PSDN PICS layer 2	3		3
	24	ISDN PICS layer 3	12		12
	25	Digital Short Range Radio (DSRR)	9		9
			586	191.2	249

Table 1
Project Teams and working capacity

		Manpower in Man-months 1989-92	1989	1990
Sum carried forward		586	191.2	249
IV	(PT 4) Voice band modems (NET 20-24 candidates)	24.5	0	
2V	(PT 16) ISM Administrative core team	25	24	
3V	(PT 17) ISM Technical core team	25	27	
4V	(PT 18) ISM Managing facilities	9	8	
5V	Mobile Expert Group (MEG)	25	21	
		694.5	271.2	249

Table 1 (continued)
Project Teams and working capacity

Komitee	ETS		I-ETS		Technische Reports	
	1989	1990	1989	1990	1989	1990
NA	-	55	-	-	17	34
BT	-	-	-	-	1	-
SPS	7	109	-	-	-	-
TM	-	8	-	-	-	-
TE	16	15	-	-	-	-
SES	-	5	-	-	-	-
RES	6	2	1	3	-	-
PS	-	16	-	-	-	-
GSM	-	-	-	-	15	15
	29	210	1	3	33	49

More than 200 of the 239 Draft-ETSs are related to the EURO-ISDN.

Table 2
Number of ETS under production

NET AND ETS

Due to the fact that a large demand exists in Europe for common European telecommunication standards a Memorandum of Understanding (MoU) was elaborated in 1985 on the initiative of CEPT. In this MoU, the signatories agreed to apply common European telecommunication standards (so-called NETs) for terminal approval and equipment procurement, to perform uniform conformance tests and to mutually recognize the results. The telecommunications administrations of most Western European countries signed this MoU.

The EC Directive 86/361 EWG obliges the EC member states to apply the common conformance specifications annexed to the NETs for approval and procurement.

Whenever a public network is changed or nearly installed, this must occur in conformity with the NETs. However, not all network operators are able to apply the new NETs within a short time. For these cases, NETs contain rules for the transitional period.

Basis for NETs are ETS elaborated by ETSI. ETS are European standards which may be applied on a volontary basis. From the very beginning of the standardization work certain ETS are designated as NET "candidates" and in contrast to "normal" draft ETS they undergo a worldwide public inquiry phase (consultation).

Final release of a NET is done by TRAC (Technical Recommendation Application Committee) which was founded as a consequence of the MoU. These NETs are of a binding nature for the signatories of the MoU.

Table 3 lists the NETs planned to be established in the near future. NETs 1-3 have already been published. NETs 1 and 2 will become obligatory network access standards from 30 June 1990 onwards.

CONFORMANCE TESTING AND CERTIFICATION

In last COMNET we have reported on the planned Conformance Testing Service (CTS) in the EC countries. Standards alone do not yet guarantee full compatibility between different systems or network components. Purpose of the EC-sponsored CTS programme is therefore to provide means to test and verify whether products conform to standards. The project is a combined effort of several European countries which have set up a consortium and which share technology and experience to achieve this goal. See Figure 4.

The advantages for manufacturers and users by this programme are obvious. Due to the fact, that test results are mutually recognized by the EC countries, cost reduction for equipment approval is certainly the salient improvement.

I Public Network Access

NET 1 X.21 Access
NET 2 X.25 Access
NET 3 ISDN Basic Access
NET 4 PSTN Basic Access
NET 5 ISDN Primary Rate Access
NET 6 Switched Access to PSPDN (X.32)
NET 7 ISDN Terminal Adapter (X.21, X. 25, ...)

II Land Mobile Systems

NET 10 European 900 MHz digital cellular mobile telecommunications network access
NET 11 Telephony characteristic of terminal equipment for the European 900 MHz digital cellular mobile telecommunications network

III Voice Band Modems

NET 20 General Requirements for Voice Band Modems
NET 21 Specific Requirements for Modems V.21
NET 22 Specific Requirements for Modems V.22
NET 23 Specific Requirements for Modems V.22 bis
NET 24 Specific Requirements for Modems V.23
NET 25 Analogue Modem V.32

IV Other Terminals

NET 30 Group 3 Facsimile
NET 31 Group 4 Class 1 Facsimile at 64 kbit/s
NET 32 Teletex
NET 33 Telephony characteristics of terminal equipment for ISDN.

Table 3
Published and planned NETs

Table 4 provides a survey of the test services offered in the member countries. Test centres which provide harmonized test services to test terminal implementations have to cater that their test tools are fully harmonized, otherwise compatibility between the tested products could be lost. For that reason, the consortium has agreed to determine certain centres as Reference Centres. Such test centres provide reference implementations to validate test tools and test services for all contractors and future test providers.

The EC-sponsored CTS programme of the services listed in Table 3 ended in 1989. The consortium was well aware that this programme had to be continued on a voluntary basis, because otherwise lack of cooperation would easily lead to discrepancies in the test services offered. As a consequence, the contractors continued their work on a

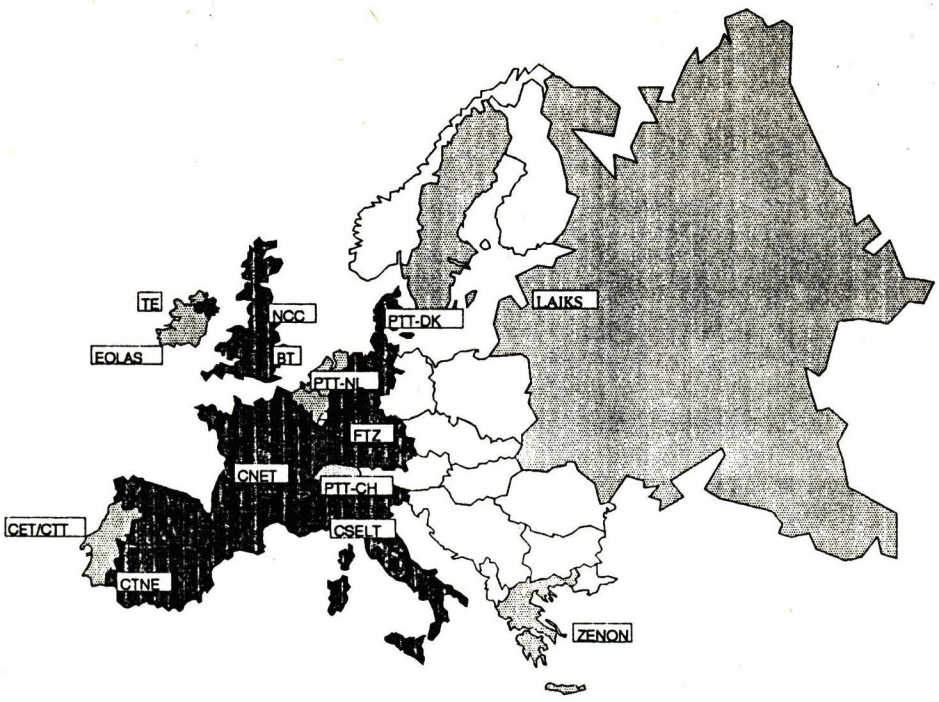


Figure 4
Participation in the CTS programms of the EC

voluntary basis setting up the Open System Testing Consortium (OSTC). OSTC offers the same services as the consortium did hitherto. Membership to OSTC is open to everybody. All members have the right to use the testing services available.

Since the beginning of the Conformance Testing programme consortium members including EC and OSTC signatories have invested some 4000 manmonths and a total investment sum of 50 M ECU. The concept of the OSTC was highly appreciated by the European Committee for Information Technology Testing and Certification (ECITC). ECITC was set up by the signatories of the above-mentioned Memorandum of Understanding (M-IT-03, for mutual recognition of test reports) with the task to promote and coordinate the concept of "Recognition Arrangements" in Europe in the wide areas of information technology and telecommunication. OSTC has been acknowledged by ECITC to comply with the rules for a Recognition Arrangement.

In the recent past many international organizations and companies, between them COS (Cooperative Open Systems - USA), SPAG (Standard Promotion and Application Group - EUROPE), INTAP have declared their strong interest

CTS-WAN					CTS-2				CTS-2bis	
TTX	Net	MHS	FTAM	T&S	PSDN	FTAM	ISDN	MHS	ISDN	DATAM
D	I	GB	GB	F	D	GB	GB	DK	I	I
F	D	F	D	GB	F	I	F	F	E	D
DK	DK	I			I	D	I	G	D	
E	E	D				F	E	NL	DK	
		E				NL	P	EI	NL	
						B	D	GB		
							EI			
							NL			

Table 4
Conformance Test Services provided in the EC

Note: CTS-WAN, CTS-2 and CTS-2b mark the different steps in the development of Conformance Testing Services.

SESSION 1

LANS, MANS AND ISDN

A CONCEPT BECOMES REALITY
FIRST EXPERIENCES WITH COMMERCIAL ISDN SERVICES

Peter Schneider

Deutsche Bundespost TELEKOM
Fernmeldetechnisches Zentralamt, Darmstadt
Federal Republic of Germany

SUMMARY

The transition period from the digitized telephone network to the ISDN started on December 15, 1988. By end of 1989 the most important centres of trade and business in the FRG were supplied by ISDN accesses directly or via concentrators. The paper outlines the technical solution, shows the degree of accordance with international standards and gives an overview on national specifications and their background.

As a matter of special interest the first international ISDN connection between the FRG and the Netherlands will be described.

A description of the ISDN-terminals i.e. telephone, telefax Group 4, Teletex and Videotex with the most important features together with the ability to interwork with terminals connected to dedicated networks based on potential user applications in different market segments will be given.

The tariff concept and the actual tariffs compared with tariffs in other countries e.g. France and Japan will give an impression on the support of the ISDN.

Emphasis will be put on the special concept of the DBP TELEKOM to achieve a nation wide coverage of ISDN for about one million ISDN customers. Finally an overview will be given on the impact of the DBP TELEKOM strategy to setting up the ISDN taking into account the "Memorandum of Understanding of the Implementation of ISDN-services in Europe by 1992" which was signed by about 25 West-European telecommunication administrations and RPOAs in 1989.

1. INTRODUCTION

At the end of 1988 and at the beginning of 1989 several administrations and network operators announced the availability of ISDN in the respective country.

In order to evalute the situation it is necessary to clarify what the characteristics of an ISDN really are.

Furthermore for the presentation of the ISDN in the own country a similar structure should be used to give the audience the opportunity to compare it with the situation in its own country.

1.1 Structure of Rating Telecommunication in ISDN

The Telecommunication in ISDN is based on

- a digitalized telephone network with the CCITT Central Signalling System No. 7 for signalling between the national and international exchanges with the special ISDN User Part (ISUP)

- a digital transmission system on the subscriber line

- the access configuration based on 64kbit/s information transfer channels together with the signalling channels with 16kbit/s or 64kbit/s

- the type of user-network-interfaces from the physical point of view

- the signalling protocol between end systems and the digital exchanges

- the types of bearer and teleservices together with the different supplementary services

- the protocol structure in end-to-end connections in the B-channel

- Voice- and non-voice-terminals.

Finally an overview should be given on the tariffs to show the costs for partizipation in ISDN.

1.2 The Pilot Project

In nearly all countries the commercial use of the ISDN is or was intended to start after a pilot project for the various reasons.

The aim of the Deutsche Bundespost's ISDN pilot project was to test the interworking of all technical components for use in ISDN.

The project was not regarded as a market survey or an acceptance test. Originally the Deutsche Bundespost Telekom assumed that the attractiveness of the ISDN would speak for itself. Today it is clear that ISDN can only be

considered as attractive together with sensible tariffs and usefull applications. The ISDN pilot project in the Federal Republic of Germany was launched in Stuttgart and Mannheim in 1987 with about 400 ISDN users in each area.

All ISDN components and services such as telephone, telefax, teletex and data transmission were based on the CCITT-Redbook recommendations with national completions in cases of lack of standardization. All work on the ISDN pilot project was according to plan and satisfactory results were produced.

2. CURRENT STATUS

2.1 Availability of ISDN at the Beginning of 1990

The transition period from the digitalized telephone network to the ISDN started on the 15th of December 1988 in Stuttgart followed by the other most important centres of trade in the Federal Republic of Germany in Berlin, Düsseldorf, Frankfurt, Hamburg, Hannover, Munich and Nuremberg. Since the middle of February 1989 in each of these cities 1000 accesses for basic access and primary rate access have been available for use by subscribers. Within these cities about 8000 customers in some 80 local networks can be served.

By the end of 1989 the most important centres of trade and business (about 40 cities) are supplied by ISDN accesses directly or via concentrators.

2.2 Technical Solutions

The technical solutions followed by the Deutsche Bundespost Telekom should show the degree of accordance with the international standards. Only in cases without sufficient standardization national specifications have been used.

2.2.1 In the Network

The Deutsche Bundespost Telekom has selected after extensive field trials and an international call for tender two digital exchange systems with ISDN capabilities the Siemens system "EWSD" and the SEL "System 12". The CCITT Signalling System No. 7 is based on the Redbook version of CCITT with the national supplements as a prerequisite for the use of a variety of supplementary services. The national supplements were required since the signalling systems was not specified to convey the information elements necessary to activate and deactivate respectively the different supplementary services.

2.2.2 Access Configuration

In contradiction to many other countries the Deutsche Bundespost Telekom planned the pilot project and of course the commercial service as well in accordance with the relevant CCITT recommendations from the very beginning.

The Basic Access includes two separate B-channels with 64kbit/s each and a signalling channel with 16kbit/s. In the case of the Primary Rate Access the system contains 30 B-channels with 64kbit/s each and a signalling channel with 64kbit/s capacity have been introduced.

2.2.3 Subscriber Line

The Deutsche Bundespost Telekom together with the national telecommunication industry, has made every effort to develop a digital subscriber loop transmission system with the following requirements:

a) use of existing copper pairs in the subscriber loop network;
b) no network rearrangement;
c) support of the two 64kbit/s B channels and the 16kbit/s D-channel of the ISDN basic access;
d) compatibility with existing analogue transmission in the same cable;
e) up to 100% utilization of existing cables;
f) covering the whole range of subscriber loop length, giving preference to systems which reduce the necessity of regeneration in the loop network.

As a result of our studies and after field tests in the local loop network in the city of Berlin, the decision was made in favour of a digital transmission system with echo cancellation, which promises to meet the above-mentioned requirements best. Meanwhile, this transmission system has been extensiveley tested in the ISDN Pilot Project with more than 700 basic access in the cities of Mannheim and Stuttgart.

The maximum repeater free length has been proved for 8 km with 0,6 mm wires or 4,2 km with 0,4 mm wires. With this capability it is possible to connect about 98% of all existing subscribers to ISDN local exchanges without the use of any regenerators. Under these conditions it was not possible for the Deutsche Bundespost Telekom to follow the decision made e.g. in Japan to use the "Ping Pong" system with its relatetively short range of about 2,5 km.

2.2.4 User-Network-Interface

This section is dealing with the physical requirements only. The implementation of the common interface considers the complete international standards of CCITT, in detail the Recommendation I.430 and I.431 for the Basis Access and for the Primary Rate Access.

In this area also a common European Approval Standard the socalled "Normes Européene de Telecommunication (NET)" has been elaborated and approved by the relevant body. The solution implemented by the Deutsche Bundespost Telekom is in accordance with the CCITT Recommendations as mentioned and with the NET.

In the case of power feeding it was decided to use the option of power feeding for telehpone terminals in the normal mode by a special power supply device connected via the network terminator. This solution is different from the Japanese approach to feed the telephone terminals in normal mode from the exchange.

Unfortunately for the very beginning of the ISDN era the Deutsche Bundespost Telekom is using a national plug system for the "universal telecommunications socket" not in line with the relevant ISO Standard for the passive bus configuration with up to 12 sockets.

Hopefully at the beginning of 1991 the Deutsche Bundespost Telekom will use an installation system strictly in conformity with the international standards. The reason for the hesitation of the DBP to adopt the international agreed plug system was based on national safety regulations. After modifications of these regulations it was possible to use the same system as well as other countries.

2.2.5 D-Channel-Protocol

Whereas the physical layer and the link layer protocol in layer 2 is full compatible with implementations in line with the newest edition of the CCITT recommendations the specification of layer 3 was based on the functional mode of the Redbook version of CCITT recommendation Q.931. Due to a lack of fully standardized supplementary services the DBP together with the German manufacturers enhanced the layer 3 with a great number of supplementary services exceeding all national solutions in other countries. The disadvantage of this proceeding and this is also relevant for all other countries having introduced ISDN is the contradiction to the current Bluebook version of the CCITT. This version contains only a few supplementary services and is incompatible to current implementations.

It is the intention of the Deutsche Bundespost Telekom to compensate this incompatibility without any notice to the subscriber by the extended capability of the exchanges to handle both protocol versions.

2.3 Telecommunication Services

2.3.1 Bearer Services and Teleservices

As in the past, the telephone service is expected to play the dominant role in communications, except that in ISDN both quality of speech and ease operation is significantly enhanced.

But also the Telefax group 4-service might continue the success of the Telefax group 3-service in the analog telephone network. A decisive factor for this assumption could be based on the terminal cost. ISDN Teletex, a complement to the existing Teletex service, is offered together with interworking capabilities to existing Teletex customers connected to the circuit switched public data network, and expected to the middle of 1990 to worldwide Telex customers.

The new ISDN bearer service (circuit mode 64kbit/s unrestricted 8 kHz structured bearer service according to CCITT Recommendation I.230); and services which are currently offered in the telephone network, e.g., Bildschirmtext (the interactive videotex service) and modem-based data transmission, are offered in ISDN via terminal adapters TA a/b. Similarly, data services currently offered in the circuit-switched public data network are also offered in ISDN via terminal adapters TA X.21/X.21bis.

Access to the packet-switched public data network according to CCITT Recommendation on X.31, case A, is available via appropriate terminal adapters TA X.25.

2.3.2 Supplementary Services

As already mentioned in the context of the D-channel protocol a high number of supplementary services are offered.

Unfortunately most of them can only be used on a national basis. Interconnection of international ISDN will lead to a minimum but common used supplementary services.

2.3.3 Protocol Structure in B-channel

Due to the end-to-end relationship of the communication protocols in the B-channel the selection of an appropriate protocol stack causes great impact on the terminal development. This is particulary true for Telematic terminals.

In order to have terminals available at the beginning of ISDN in 1988 the DBP Telekom concluded contracts for the delivery of Telematic terminals already in 1986. At this point in time no international agreement existed. Therefore it was necessary to select a well-known protocol stack already in use over many years for the Teletex service in the CSPDN also for the B-channel in ISDN at least for the common layers 2-5. The application dependent layers for Teletex are the same as for the service in the dedicated network. For the use for Telefax, Group 4 terminals the Redbook versions of CCITT, Study Group VIII have been prescribed. The DBP Telekom is aware of the situation to change or at least to adapt the current protocols in the case of a new terminal generation which will be in line with the protocols definded in CCITT Recommendation T.90 or the European equivalent respectively.

2.4 Overview on terminals

With the introduction of ISDN the participation of the DBP Telekom in the market for all terminal areas could be achieved. With this decision the DBP Telekom could enter into competition with private terminal providers in the areas Teletex and Data communication. This has influenced the procurement of terminals for these areas in such a way to offer in particular powerful and inexpensive terminals.

For Telephone terminals the same might occur but only the other way round. That means the participation of private terminal providers in the Telephone area. This circumstance has been considered by the DBP Telekom and has led to the procurement of digital telephones of different comfort classes to relatively low prices. In order to be prepared for the delivery of terminals in 1988 and 1989 the DBP Telekom concluded contracts for 55.000 telephones, 800 Telefax 4 terminals - an new call for tender for further 1.000 terminals will be published very soon -, 700 Teletex terminals and 11.000 Terminaladapters of different types.

2.5 Tariffs

As far as tariffs are concerned, in the option of Deutsche Bundespost Telekom the ISDN concept with its standardized "telecommunication socket" calls for harmonization of tariff structures and tariff levels for the various ISDN services. The following marginal condition must be kept in mind here:

Since the telephone service constitutes the mass business and the financial cornerstone, it serves as the basis for tariffing all services in the ISDN. In the ISDN, all services are offered via user-network interfaces, i.e., user-network interfaces are provided for dedicated

(leased) circuits just as they are for switched connections. As already mentioned two kinds of interfaces are initially available in the ISDN:

a) The basic access user-network interface with two B-channels of 64kbit/s and a signalling channel (D-channel of 16kbit/s;
b) The primary rate user-network interface with thirty B-channels of 64kbit/s and a signalling channel (D-channel of 64kbit/s.

Within each access any combination of B-channels for switched connections and for semipermanent connections is possible. The relevant tariffs for ISDN were published in the Telekommunikationsordnung (TKO), the Telecommunications Regulation of the Deutsche Bundespost Telekom, in 1988. The cornerstones of the ISDN tariffs are:

a) Above all, the charges for providing ISDN accesses are service-independent:

aa) Basic access (2 B + D)

registration charge:	non-recurrent (77 US $)	130,--	DM
standing charge :	monthly (44 US $)	74,--	DM

ab) Primary rate access (30 B + D)

registration charge:	non-recurrent (120 US $)	200,--	DM
standing charge :	monthly (310 US $)	518,--	DM

Included in these tariffs are all services offered via ISDN and a great number of supplementary services, such as call waiting, call forwarding services, terminal selection, direct dailing in, three party service, calling line identification presentation and restriction, charge in service during a call, call hold, etc. Only those supplementary services which require extraordinary efforts, e.g,, closed user group, conference calling, alarm call service etc. are charged separately.

b) Tariff principles for the provision of demand services are:
- service independence,
- dependence on distance (tariff zones),
- different usage times for a unit fee (at present 0,23 DM/ 0,14 US $) in various tariff zones,
- normal and cheap-rate times.

This list might be continued by information on tariffs on permanent and reserved services.

The tariffing of all services in ISDN on the basis of the

unit fee derived from the Telephone service might occur as a disadvantage for non-voice services. The tariffing of non-voice services operated in dedicated networks is based on the principles to measure the exact time of the connection on a per second basis.

In other words the charge for a Teletex message on one A4 page transmitted e.g. in the same city costs 0.23 DM (0,14 US $) in ISDN and about 0,04 DM (0,02 US $) in the CSPDN. To remove this disadvantage the DBP Telekom intends to introduce for non-voice services a new charging system in 1991 based on the same principles as in the dedicated networks.

3. STRATEGY

3.1 ISDN Penetration 1989-1991

As already mentioned the DBP Telekom installed exchanges capable to provide ISDN features in 8 main cities in the Federal Republic of Germany. All exchanges placed in service in 1989 and the following years will be capable of terminating ISDN customers. Furthermore, those digital exchanges installed between 1985 and 1988 will then be retrofitted with ISDN capabilities. By 1990, approximately 100 trunk exchanges, 200 local exchanges, and more than 500 basic access conzentrator units will have been installed. Since some 50% of all telephone stations representing more than 90% of the business market segment are concentrated in these cities, the majority of the potential ISDN customers can be served without crossing local network boundaries.

3.2 Special Attachment (Remote Connection)

Nationwide coverage of ISDN is planned for 1993. In order to reach this objective, still existing electromechanical switching centres will be equipped with remote controlled basic access concentrator units. According to the actual planning objectives, about one million ISDN customers can be served by the end of 1993.

For the connection between the concentrators and the ISDN exchanges PCM systems with 2Mbit/s are used. Depending on the manufacturer of the concentrator up to 500 remote subscribers can be connected to one concentrator and connected to the ISDN exchange via up to 5 2Mbit/s-systems.

4. PLANNING AND REALIZATION

One of the fundamentel intention of the DBP Telekom is to be in accordance with all their ISDN components with the international standards. To achieve this objective the areas indentified will adopt those standards at the earliest convenience. Furthermore the participation in international ISDN will lead also to an progress in the in the migration to an ISDN.

4.1 Future International Interconnection

Due to bilateral or multilateral interest of European countries the introduction of international ISDN connections appears in outlines.

4.1.1 Deutsche Bundespost Telekom - Rotterdam Area

In February 1989 a bilateral agreement between the PTT Telecom of the Netherlands and the Deutsche Bundespost Telekom was signed to open the worldwide first international ISDN connection between an ISDN installed in the Rotterdam area and the ISDN of the DBP. This project started on the 12th of October 1989. The specification used for the procurement of the relevant ISDN components in the Netherlands are derived from the German specifications to achieve compatibility from the very beginning. The ISDN project is scheduled over 3 years duration and will be replaced by a Pan-European ISDN which is planned by 1992.

4.1.2 Deutsche Bundespost Telekom - France Telecom

This project in planned to start in 1990. The specifications used in both countries have been developed almost independently. The signalling between the international ISDN exchanges will be based on the Signalling System No. 7. However for early interconnections it is only possible to use the "enhanced Telephone User Part" (TUP+) as defined by CEPT and agreed in Europe. As a consequence in the near future a further step must be taken to approach the CCITT No. 7 system with the world-wide standardized and powerful "ISDN User Part" (ISUP). The absorption of this project by the Pan-European ISDN is also scheduled by 1992.

4.1.3 "Memorandum of Understanding on the Implementation of an European ISDN Service by 1992"

The international interconnection of ISDN to a wide extent is only possible in an coordinated way based on international agreed standards. In Europe it is intented to establish a European wide ISDN by 1992. For the commitment of the participating countries a "Memorandum of Understanding on the Implementation of an European ISDN Service by 1992" (MOU) was signed in April 1989 by about 25 West-European telecommunication administrations and RPOAs. The agreed target for the communication of the European ISDN Services in accordance with this MOU, is 1992. All parties, by this MOU, commit themselves to the necessary co-operation activities, in particular the development of the necessary standards by the "European Telecommunication Standardization Institute" (ETSI) by December 1989. Due to some delay in the elaboration of about 200 standards the final date was moved to April 1990. Considering the different importance of services priority has been given to have at least a bearer service, unrestricted 64kbit/s and the teleservices

Telephony (3.1 kHz bandwith), Teletex and Tele fax (Group 4) available from the very beginning of the project.

4.2 Future Services

Already with the introduction of ISDN the DBP Telekom has published its intention to provide additional services depending on the availability of relevant international standards and to support the development of appropriate terminals as well. The DBP Telekom is considering the introduction of Telephony (7 KHz), Data Communication and Videotelephony (2 x 64 kbit/s) and the relevant bearer services by the end of 1990.

There have been a number of opinions opposing the introduction of an "information era", in which virtually unlimited access can be gained to all form of information. A telecommunications administration such as the Deutsche Bundespost Telekom that intends to remain a driving force in the field of telecommunication must face this challenge.

A Routing Scheme for a Non-Strict Hierarchical Networking Environment*

Saewoong Bahk Magda El Zarki

Systems Department Electrical Engineering
University of Pennsylvania
Philadelphia, PA19104, U.S.A.

In this paper we propose a routing protocol which can be used for the communication between two adjacent autonomous networks. The two networks are connected to each other via a high speed hierarchical backbone network and a direct low speed local transmission facility. Our objective is to design a routing protocol for communication between the two networks that makes efficient use of the less costly local facility. The term '*efficient*' here indicates a trade-off between cost and performance (expressed as average packet delay). The proposed routing scheme is quasi-static. It can adapt to changing environments, and is well suited to networks extended by transparent bridges.

1 Introduction

In this paper we address the routing problem between autonomous networks connected both directly and via a public network hierarchy. The latter consists of a cross-country backbone and a set of regional networks. A typical regional network interconnects several of these autonomous environments that are clustered in a geographic area. In some instances local transmission facilities are used to provide direct communication between two networks. This local link, although hidden from the rest of the public domain, violates the strict hierarchy for those two networks.

An example of such a networking environment would be the current Internet in the USA. The Internet naturally forms a hiearchy which consists of a nation-wide backbone, a set of regional networks, and a set of campus and research laboratory networks. However it is not strictly hierarchical [COM88][NET88]. Two campuses which belong to the same region may possess a local link that connects them direclty. As such they do not follow a strict hiearchical structure. To communicate they may either use the Internet or the local transmission facility. In the Internet, computers, called gateways, provide

*This work was supported in part by a grant from AT&T program in Telecommunication and in part by the Benjamin Franklin Partnership Program

all interconnections among physical networks. Gateways play a key role in the Internet communications. They route packets based on the destination network address not on the individual host address thereby keeping the routing tables to a manageable size.
The routing protocol presented here can be used for the communication between two adjacent locally connected, autonomous networks that are connected to the hierarchical backbone network via a backbone (or *core*) gateway that uses hierarchical routing. Our routing algorithm is superimposed on top of the existing routing scheme and is only used when a choice in routes is available. The networks are attached to the local link via a *border* gateway. The proposed routing scheme can adapt to changing environments, and it is well suited to networks which are extended by transparent bridges. By adopting this scheme we are able to use the local transmission facility efficiently with possibly some increase in the average packet delay.
Routing strategies are roughly classified into static, quasi-static, and dynamic. A static routing strategy chooses the paths to send packets to the destinations independently of the instantaneous states of the communication channels in the system and keeps the routing table fixed for a long period of time. A dynamic routing strategy determines the paths for sending packets according to the instantaneous states of the channels. A quasi-static routing strategy lies in between these two. Much work has been done on hierarchical routing using the above strategies. However all these strategies assumed a homogeneous networking environment.
We propose in this paper a quasi-static proportional routing algorithm that can operate in the described non-hiearchical network scenario. In our discussions we consider how a source host routes packets to the destination host in an adjacent network if they are connected through the backbone network and have a local facility too. Through the backbone network packets may travel several hops to reach to the destination network even though the destination is in an adjacent network.
This paper is organized as follows. In section 2, we present the model description and problem formulation. We propose the routing protocol and solve the optimization problem in section 3 and 4. The numerical results for several weighting factors are presented in section 5, followed by the conclusion in section 6.

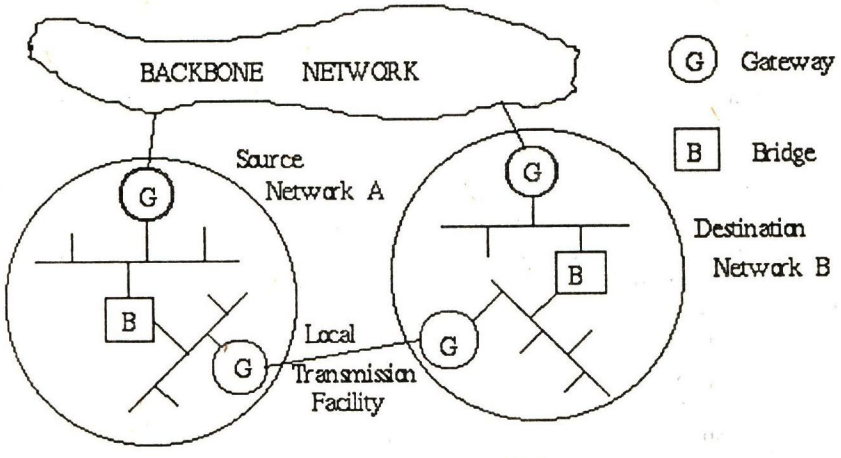


Figure 1. An Example of a Regional Network

2 Model Description and Problem Formulation

We assume that each private network is extended by transparent bridges and forms an autonomous system that falls under one administrative entity. Gateways in an autonomous system cooperate closely to propagate routing information. They usually operate at the network level[COM88][NET88][TAN88].
Transparent bridges are easy to install and popular in the Ethernet extension. Since the routing algorithm used by these bridges is Baran's backward learning, the routing procedure depends on the location of the source LAN and the destination LAN.

- If the destination is on the same LAN, the bridge discards the packet.
- If the destination LAN is different from the source LAN, forward the packet.
- If the destination is unknown, use flooding.

An example of a regional network is given in figure 1. We simplified the given network for our problem formulation as shown in figure 2. The routing scheme in the backbone is assumed to be based on virtual circuits using the shortest path. Let the source host be in network A and the destination be in network B and γ_{SD} be the total arrival rate at node 1 destined for network B. Let λ_1, λ_2 denote the rates of the packet substreams at the gateway which are to be sent to the destination network through the backbone network and the local network respectively. The packet length is assumed to be exponentially distributed with average length $\frac{1}{\mu}$[bits/packet]. We use Kleinrock's independence assumption to solve for each node and successive link independently.

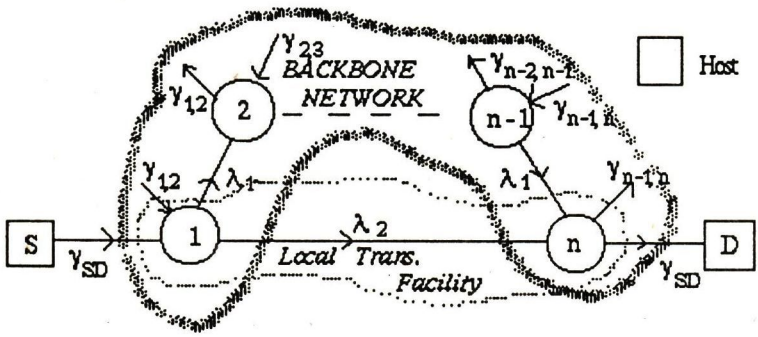


Figure 2. Model for the Formulation

The function of node 1 is to divide the incoming traffic γ_{SD} into λ_1 and λ_2. Node 1 needs a proper objective function to obtain the optimal λ_1, λ_2. The traditional objective function for making the routing decisions involves the minimization of the average packet delay[GAL77]. However, adopting this for the proposed networking environment could result in the poor utilization of the less costly local facility. In order to overcome this situation we propose a modification to the traditional objective function. We choose a linear combination of the average number of packets $E[N_1]$ in the backbone network and the average number of packets $E[N_2]$ on the local link that orginated at the source

network. By changing the weighting factor β we can adjust the trade-off between the cost of operation and the user performance expressed as average delay.
Based on the traffic assumptions mentioned above, each queue in each node and successive link can be modeled as a M/M/1 queue. Using the results of the M/M/1/ queue, the problem can be formulated as follows[KLE75]:

$$\min_{\lambda_1,\lambda_2} \; E[T_w] = \frac{1}{\gamma_{SD}}\{\beta E[N_1] + E[N_2]\}, \;\; \beta \geq 1$$

$$E[N_1] = \lambda_1 \sum_{i=1}^{n-1}[\frac{1}{\mu C_{i,i+1} - (\lambda_1 + \gamma_{i,i+1})} + \frac{d_{i,i+1}}{\mu}]$$

$$E[N_2] = \lambda_2[\frac{1}{\mu C_{1,n} - \lambda_2} + \frac{d_{1,n}}{\mu}]$$

Subject to: $\lambda_1, \lambda_2 \geq 0$, $\lambda_1 + \lambda_2 = \gamma_{SD}$, $\mu C_{1,n} > \lambda_2$, $\mu C_{i,i+1} > \lambda_1 + \gamma_{i,i+1}$, $i = 1, \ldots, n-1$
Given: $\gamma_{i,i+1}$, $d_{i,i+1}$, $C_{i,i+1}$, μ, γ_{SD}, $C_{1,n}$

Where for the link between node i and node $(i+1)$ we define:
$d_{i,i+1} \triangleq$ the processing and propagation delay[sec/bit]
$C_{i,i+1} \triangleq$ the link capacity[bits/sec]
$\gamma_{i,i+1} \triangleq$ the interfering traffic[packets/sec]

The cost bias is reflected by assigning a proper value to the weighting factor β. In case of $\beta = 1$, this linear combination represents the average packet delay from the source network to the destination network (the traditional objective function). Minimizing this function results in the lowest average delay but possibly poor network utilization of the local facility. $\beta \gg 1$ indicates a '*weighted*' packet delay, most packets for the adjacent network will not be sent through the backbone network because the backbone costs are high ($\lambda_1 \downarrow$, $\lambda_2 \uparrow$). Dependent upon the utilization on the local facility they could suffer a little longer delay to reach the destination.

3 Routing

In a study of routing schemes[RUD76] proportional routing was defined as selecting one of several possible paths with *a certain probability* whenever a packet is to be transmitted. In this proposal *a certain probability* is defined as *the threshold.* The gateways use proportional routing for the routing decision after solving the optimization problem in order to obtain the threshold. In [RUD80] it was shown that the performance of a session-based adaptive scheme was at least as good as packet-based schemes in terms of network utilization and processing requirements. The proposed routing algorithm is adaptive on a session basis even though the formulation and optimization is packet-based. The reason being that most networks use virtual circuits not datagrams and all routing is done on a session by session basis.

3.1 Proportional Routing

Our routing algorithm works on the assumption that the Address Resolution Protocol(ARP) is used[COM88]. In other words, a scout (or *request*) packet is first transmitted to physically locate the destination host. In our case, the gateways act as the intermediate destinations for both gateways. Without loss of generality we assume that the backbone gateway solves the optimization problem and sends the optimal value $(\lambda_1^*, \lambda_2^*)$ to the border gateway. Then the border gateway sets the threshold for the comparison at $\frac{\lambda_2^*}{\gamma_{SD}}$. Whenever a new session destined for the adjacent network is initiated by a host, both gateways in the local environment receive the broadcasted connection request packet. Only the border gateway reacts however. The border gateway generates a random number $\epsilon[0,1]$ and compares this with the threshold. If the generated number is less than the threshold all the packets from that session will be routed through the local link. If not, the border gateway sends a *HELP* message with the session identification number to the backbone gateway indicating that it is responsible for that session and discards the request packet for that session.
The backbone gateway in turn saves the request packet destined for the adjacent network for a given time, t_{max}. If it does not receive any *HELP* message for that session from the border gateway within t_{max}, it discards the packet. Once a route has been assigned to a session, all its packets will be routed to the appropriate gateway on the path.
Let $t_p(i, ba)$, $t_p(i, bo)$ be the packet delay from host i to the backbone gateway and the border gateway respectively. Let ω be the set of the nodes in the source network and t_{com} be the communication delay between the backbone gateway and the border gateway. Then t_{max} is given as;

$$t_{max} = \max_{i \epsilon \omega} \mid t_p(i, ba) - t_p(i, bo) \mid + t_{com}$$

If the *HELP* message is lost during the transmission, the host will fail to receive the call set up confirmation message from either one of the gateways. We assume in this case that the host will time out and re-broadcast the connection request packet.

3.2 Algorithm

The algorithm is classified into three classes according to the location of the destination.

3.2.1 Local Address

The destination is on the source network and recognizes its own network address in the request packet. All the gateways on the network ignore packets with a local address.

3.2.2 Adjacent Network Address

The algorithm is given as follows:

- The source host sends out a request packet because it has no knowledge of the destination and the network is transparent.
- Both gateways receive the request packet.

- The border gateway generates a random number if it receives a packet destined for the adjacent network and compare this with the threshold.
- If the number is less than the threshold the border gateway transmits the packet through the local link.
- Else it sends a *HELP* message with session identification number to the backbone gateway and discards a request packet for that session. The backbone gateway takes care of that session after receiving this message.

To reduce the overhead load generated by the communication between the two gateways for routing decisions, we can let the gateway that has the larger proportional rate send the *HELP* message to the other gateway instead of always assigning that role to the border gateway. Therefore when $\lambda_1^* > \lambda_2^*$ the backbone gateway is responsible for sending the *HELP* message to the border gateway when needed, and vice versa when $\lambda_1^* < \lambda_2^*$.

3.2.3 Remote Address

The destination can be reached through the backbone network only. The backbone gateway takes care of that because the border gateway has no information about the destination.

If the source network is extended by source routing bridges instead of transparent bridges which are prefered in token ring extension, we need to make some modifications at the host end. In this type of network each host maintains a table of all the physical addresses and full topology of its own network[NET88]. By informing the host of the superimposed routing scheme it can send packets to both gateways and the algorithm is the same as in the network extended by transparent bridges. However, if the source wants to send to only one of them, it must know the threshold and generate a random number for the comparison. This means that proportional routing is done at the source host even though the threshold is computed at the gateway.

4 Optimization Algorithm

The optimization technique to obtain the threshold is developed by using the Frank-Wolfe method[BER87][FRA73]. We assumed that the computation is done at the backbone gateway. The gateway solves the optimization problem if necessary(quasi-static), not based on the packet or session. Figure 3 shows the feasible line and moving direction to reach the optimal point. The algorithm at the backbone gateway is as follows:

1. Let $x^* = (\lambda_1^*, \lambda_2^*)$ be the optimal point and $x^o = (\lambda_1^o, \lambda_2^o)$ be the initial point on the feasible line. Let $\overline{x}^{up} = (x_1^u, x_u^u)$, $\overline{x}^{down} = (x_1^d, x_2^d)$ be two end points of the feasible line.

2. If $\nabla T_w(x^o)(\overline{x}^{up} - x^o) \geq 0$, find a stepsize $\alpha \epsilon [0,1]$ which minimizes

$$T_w[x^o + \alpha(\overline{x}^{down} - x^o)]$$

$$\text{then } x^* = x^o + \alpha^*(\overline{x}^{down} - x^o)$$

else then find a stepsize $\alpha \epsilon [0,1]$ which minimizes

$$T_w[x^o + \alpha(\overline{x}^{up} - x^o)]$$

$$\text{then } x^* = x^o + \alpha^*(\overline{x}^{up} - x^o)$$

3. Send this result $(\lambda_1^*, \lambda_2^*)$ to the border gateway.

After receiving $(\lambda_1^*, \lambda_2^*)$ the border gateway sets the threshold for the routing decision at $\frac{\lambda_2^*}{\gamma_{SD}}$. Observe that the objective function is a convex function and the feasible region is a convex set since the constraints are linear. Therefore any local minimum is a global minimum[BAZ79].

5 Numerical Results

For the numerical study we chose a simple model as shown in figure 4. The optimal solutions for the substreams and average packet delay for several weighting factors are shown in Fig. 5 and 6. As expected if we set β very high the packet prefers to be sent through the local trunk even though the average delay is increasing.
In figure 5 the horizontal axis represents the arrival rate [packets/sec] at network A destined for network B and the vertical axes represent the optimal substreams and the average packet delay from A to B. The local link rate is 9600bps and the backbone network rate is 64Kbps. The backbone network utilization factor without γ_{SD} is assumed approximately to be 60% for the analysis.

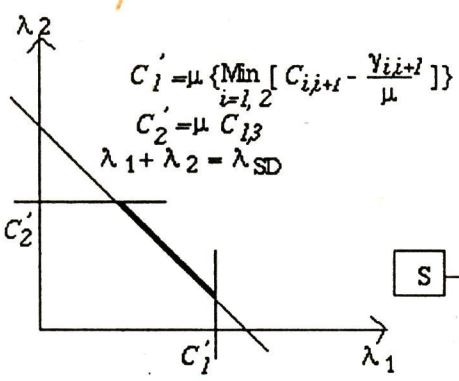


Figure 3. Feasible Region

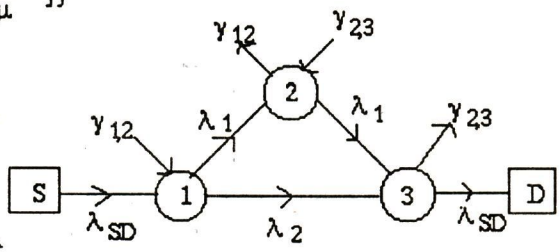


Figure 4. Model for the Analysis

In figure 5.(a) and (b) we set β to 1, which shows the performance for the traditional objective function based exclusively on delay. Up to $\gamma_{SD} = 3$ only the backbone network is used ($\rho_1 = \frac{\lambda_1}{\mu C} = 0.66$, $\rho_2 = \frac{\lambda_2}{\mu C} = 0.0$). For $\gamma_{SD} > 3$ the local link is also used for data transmission In figure 5.(c) and (d), β is set to 5. Up to $\rho_2 = 0.5$ only the local link is used with a little increase in the average packet delay as compared to the results for $\beta = 1$.
In figure 6 the backbone transmission rate is 1.544Mbps which is the rate of a T1 carrier. In case of $\beta = 1$ the local trunk is not used until $\rho_1 = 0.78$. If we set β to 10, then only the local facility is used till it reaches a utilization of $\rho_2 = 0.5$. In this case we observe that a higher β was needed to force traffic over the local link as a T1 facility costs much more than the local 9.6Kbps link, this increase in $\beta$is natural.

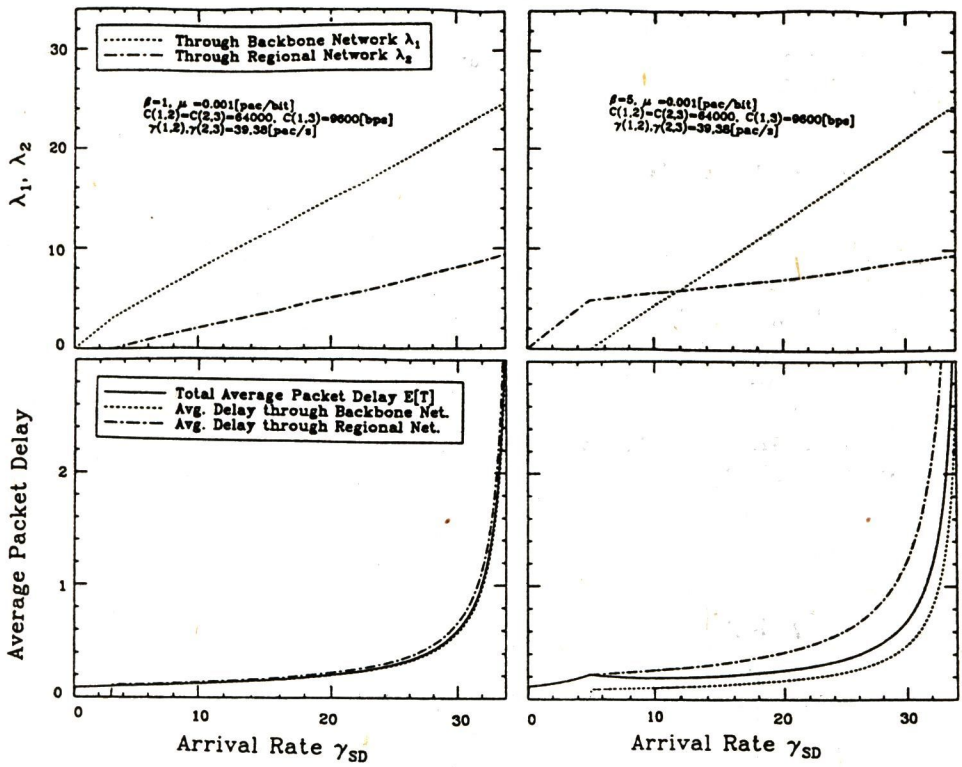


Figure 5.(a) (b)

Figure 5.(c) (d)

6 Conclusion

In this paper we proposed a routing protocol based on proportional routing for the communication between two adjacent networks which are connected to each other by a hiearchical high speed backbone network and a direct low speed local transmission link.

By defining a weighted average delay as our objective function we were able to make efficient use of the local transmission facility which is less expensive than the backbone transmission facility. This scheme is well suited to networks extended by transparent bridges. The advantage of this routing scheme is that it can be superimposed on top of a hiearchical routing algorithm with only minor modifications at the gateways. From the optimization technique we could obtain the threshold which is used for the routing decision at the gateways and find the average packet delay when this scheme is adopted.

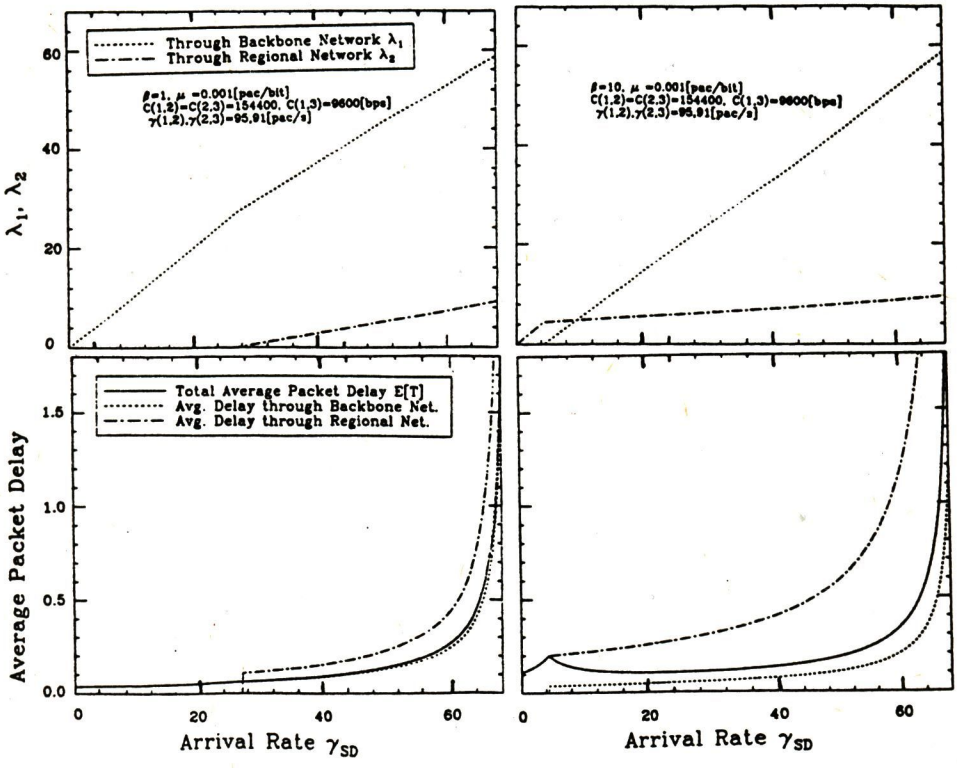


Figure 6.(a) (b)

Figure 6.(c) (d)

References

[BAZ79] Bazaraa, M., and Shetty, C.M., **Nonlinear Programming: Theory and Algorithms**, John wiley & Sons, 1979, New York.

[BER87] Bertsekas, D. and Gallager, R., **Data Networks**, Prentice-Hall, 1987, Englewood Cliffs, NJ07632

[COM88] Comer, D.E., **Internetworkin with TCP/IP: Principles, Protoclos, and Architecture**, Prectice-Hall, 1988, Englewood Cliffs, NJ07632

[FRA73] Fratta, L., Gerla, M., and Kleirock, L., "The Flow Deviation Method: An approach to Store and Forward Communication Network Design", *Networks.* 3, 1973, pp. 97-133.

[GAL77] Gallager, R., "A Minimum Delay Rouitng Algorithm Using Ditributed Computation", *IEEE Trans. on Comm.*, Vol.Com-25, N0. 1, Jan. 1977, pp. 73-85.

[KLE75] Kleinrock, L., **Queueing Systems. Volume 1: Theory**, John Wiley & Sons, 1975, New York.

[NET88] Special issue of IEEE Network Magazine, Jan. 1988.

[RUD76] Rudin, H., "On Routing and Delta-Routing: A Taxanomy and Performance Comparison of Techniques for Packet-Switched Networks", *IEEE Trans. on Comm.*, Vol.COM-24, No. 1, Jan. 1976, pp. 43-59.

[RUD80] Rudin, H. and Mueller, H., "Dynamic Routing and Flow Control", *IEEE Trans. on Comm.*, Vol. Com-28, No. 7, July 1980.

[TAN88] Tanenbaum, A.S., **Computer Networks**, Prentice Hall, 1988, Englewood Cliffs, NJ07632

ARQ ERROR CONTROL FOR MULTIPOINT SATELLITE COMMUNICATION

Stephan Jacobs and Ulrich Quernheim

Institute Informatik IV, RWTH Aachen,
Ahornstr. 55, D-5100 Aachen, West Germany
Phone +49-241-804523
FAX +49-241-806295

There are a number of reasons for using satellite networks for data communication; this is particularly valid for point-to-multipoint data distribution. Transmission errors may be met by FEC or ARQ techniques. A main aspect of ARQ error control in multipoint communication is the buffering of the acknowledgements. In this paper we discuss two buffering strategies combined with go-back-N and selective repeat ARQ protocols and with multicopy techniques. Throughput evaluation shows the advantage of acknowledgement buffering and of multicopy protocols over single copy protocols.

1. INTRODUCTION

In the next years use of geostationary satellites for data communication will increase. Satellite networks have a number of advantages, for instance flexibility and simple broadcast facilities for the distribution of identical information. Transmission protocols must take into account the special features of satellite links (e.g. long distance for geostationary satellites, 2*35,800 km or 0.27 s). Data transmission via satellite links is affected by errors, e.g. caused by rain fading. Due to the expected lack of free frequencies (today: 4/6 GHz, 12/14 GHz) and other reasons (antenna gain etc.), higher frequencies will be used in future satellite systems (e.g. 20/30 GHz). These frequencies involve stronger fading effects. To ensure a relatively reliable transmission an FEC (forward error correction) or ARQ (automatic repeat request) error control technique may be applied to the satellite link. Reliability includes correct, in-sequence transmission without loss or duplication of data. In this paper we restrict to ARQ protocols, because FEC cannot guarantee error free data transmission. In satellite networks go-back-N and selective repeat ARQ protocols may be applied..

Distribution of identical information to multiple receivers is simplified by the broadcast property of satellite channels; all stations in the beam coverage zone may receive the satellite signal. An increasing number of satellite networks with many small terminals will be established (e.g. VSAT-nets), thus broadcast (data) transmission will increase considerably, too. With point-to-multipoint communication error control by an ARQ protocol is more complicated than with point-to-point transmission; some stations may receive a frame correctly, whereas other stations may reject the same frame. Many variants of the basic ARQ protocols for point-to-point communication have been studied, cf. [9]. In [7], [12], and [13] ARQ protocols for point-to-point satellite communication and in [1], [4], [17] multicopy protocols (transmission of more than one frame subsequently) are discussed. Numerous protocols for multipoint communication have been analyzed, cf. [2], [3], [6], [8], [14], [16]. In [6] Gopal, Jaffe compare single copy go-back-N protocols, and in [16] Wang, Silvester suggest multicopy stop-and-wait and go-back-N protocols for multipoint communication, which use the outcomes of previous transmissions of a frame in different extent.

In this paper we discuss three multipoint multicopy variants of go-back-N and selective repeat, respectively, which differ in their ability to buffer and use the acknowledgements of previous transmissions of a frame not being accepted by all receivers. The GBN_0 (SR_0) protocol does not store the ACKs, whereas the GBN_all (SR_all) and GBN_all* (SR_all*) protocols do (for GBN_0 and GBN_all cf. [6] and [16] in the single copy and multicopy case). The multicopy protocols are adaptive in the way that the number of copies of a frame sent subsequently depends on the number of receivers, the round trip delay, and the frame error rates. The number of copies is optimized partly analytically and partly by simulation. Additionally, in the GBN_all* (SR_all*) protocol the number of copies in a retransmission may be different from that in the first transmission; it depends on the number of stations having rejected that frame and is chosen optimal.

2. ARQ ERROR CONTROL

Using an ARQ protocol he bit stream to be transmitted is subdivided into frames, which are subsequently numbered and contain control information and redundant bits for transmission error detection. There are three main categories of ARQ techniques, stop-and-wait, go-back-N, and selective repeat. Over satellite links with their large propagation delay *stop-and-wait* protocols are inefficient because after sending a frame the link is idle until the sender receives a positive or negative acknowledgement (ACK/NACK) from the receiver station. In contrast to this frames are transmitted continuously with go-back-N and selective repeat techniques, respectively. Most protocols contain a window mechanism for flow control; a sender may only send a specified number of frames in advance without having received the ACK for the first one. If a frame is negatively acknowledged by a receiver (or if a timeout occurs) the sender restarts transmission beginning with that frame using a *go-back-N* protocol. All N frames sent between the erroneous one and its retransmission are ignored by the receiver. The sender needs a buffer of size N+1 frames (corresponding to one round trip delay) for those frames which may be rejected and retransmitted. The receiver, however, only needs a buffer for one frame. Using a *selective repeat* protocol, only the rejected frames are retransmitted. Since a receiver station must rearrange frames which are out of sequence, it needs a buffer for at least N+1 frames.

3. POINT-TO-MULTIPOINT ARQ PROTOCOLS

3.1 Environment

The scenario considered here comprises one sender and K receivers interconnected via a broadcast satellite channel (one uplink and K downlinks); an ARQ protocol is used. Every frame transmitted by the sender is received nearly simultaneously by all K receivers, which are assumed to sense the channel continuously. Acknowledgements are transmitted to the sender via a suitable return channel (possibly a terrestrial channel). If p_u (p_d) is the probability that a frame is transmitted correctly from the sender to the satellite (from the satellite to a receiver, assuming that p_d is independent and the same for all receivers), then the probability that a frame is correctly received by a station (K stations) comes to $p = p_u * p_d$ ($p_u * p_d{}^K$), cf. fig.1. Therefore, the throughput decreases if the number of stations increases. Thus, sometimes (high error rates, many receivers) it may be favourable to transmit more than one copy of a frame subsequently.

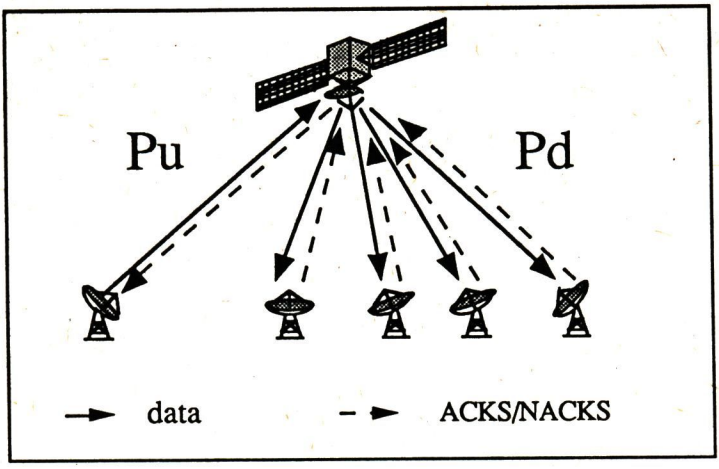


Fig.1: Multipoint satellite communication

We describe the three go-back-N protocols and the differences of the corresponding selective repeat protocols. An example of the respective performances is given in figures 2 to 5. The sender uses a list of expected acknowledgements (LEA) for those frames which have already been transmitted, but have not yet been accepted by all receivers.

3.2 Multipoint Go-Back-N Protocols

The three multicopy go-back-N protocols differ in the ability of the sender to remember acknowledgements of copies not yet accepted by all receivers and in the number of copies of a frame sent subsequently. The receiver operations are identical for the three protocols. The receiver first checks the FCS to determine whether or not the frame has been received correctly, and returns an ACK or a NACK, respectively, containing the frame number. If this frame number is the one expected by the receiver, it accepts the frame, which is delivered to the upper layers of the protocol hierarchy. If the predecessor of the frame has not yet been accepted or if the frame has already been accepted by the station earlier it will be discarded. Even if the frame is not expected, it is necessary to send an ACK to prevent deadlocks using the GBN_0 protocol described below. The sender transmits frames and expects ACKs/NACKs continuously. After sending m copies of a frame an associated time-out-counter is set; m is chosen optimally, its value depends on K, N, and the frame error rates. The protocols differ in the way how the outcomes of previous transmissions of a frame are used, cf. [6], [16]:

GBN_0: The sender receives the ACKs/NACKs for the m copies of a frame sent subsequently. If not all stations have accepted the frame at the same time (at least one of the m copies), the sender retransmits this and all subsequent frames. The retransmission of a frame is handled by the sender in the same way as the first transmission; the sender does not remember the outcomes of the first transmission of m copies when he checks the acknowledgements for the retransmission. So it is possible that all stations have received the frame correctly (in the first or second transmission), but the sender starts a third transmission of m copies.
Two conditions have to be fulfilled so that ACKs are accepted. Because of go-back-N all predecessors must have been received correctly, and because of the "0"-protocol all ACKs have to be received by the sender at the same time.

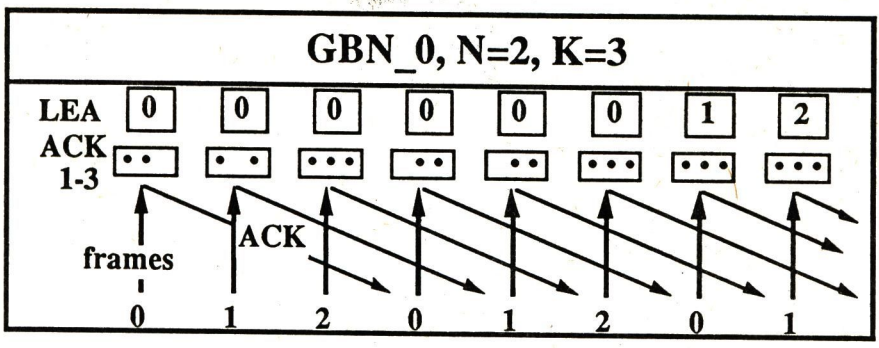


Fig.2: Performance of GBN_0

GBN_all: The LEA contains entries for each receiver, whether a particular station has accepted a frame or not. An ACK will only be registered, if the ACKs of this receiver for all previous frames have already been registered. So it does not matter, if a retransmitted frame is not received correctly by a station, which received this frame correctly the first time. If all receivers have acknowledged a frame, this frame can be discarded, the associated time-out-counter can be stopped and the ACKs of the next frame will be expected. LEA contains entries not only for each receiver but also for all expected frames.
Now only one condition must be fulfilled so that ACKs are accepted. All predecessors of the respective frame have to be received error free and acknowledged by all receiving stations.

GBN-all:* Here the number m' of copies during a retransmission of a frame depends on the number K' of those stations not having accepted this frame instead of K. For m' the optimal value of m for K' receivers in GBN_all is chosen.

In the single receiver case (point-to-point) the protocols coincide.

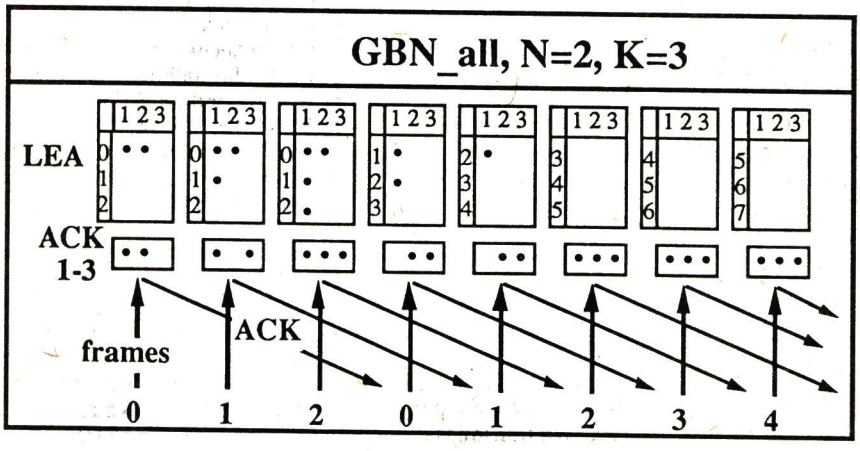


Fig.3: Performance of GBN_all

3.3 Multipoint Selective Repeat Protocols

Using selective repeat protocols all receivers have buffers to store frames which are out of order and will be rearranged later. Frames are accepted even if their predecessors were not error free (exception: buffer overflow). So the sender may accept an ACK even if its predecessor was a NACK. Therefore, only erroneous or lost frames have to be retransmitted, but not all successors. Again we consider three protocols (SR_0, SR_all, and SR_all*); they differ in the corresponding way how ACKs are used as the go-back-N protocols do.

SR_0: One condition must be fulfilled so that ACKs are accepted. Because of the "0"-protocol the ACKs of all stations have to be received by the sender at the same time.

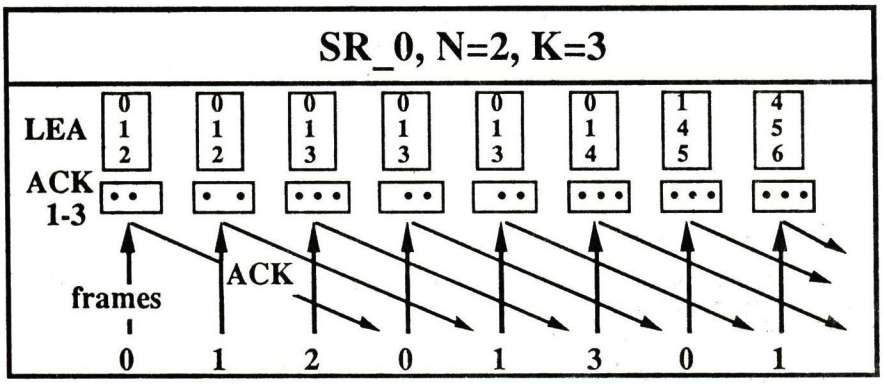


Fig.4: Performance of SR_0

SR_all: There is no condition relative to predecessors or ACKs of other stations which has to be fulfilled so that ACKs are accepted.

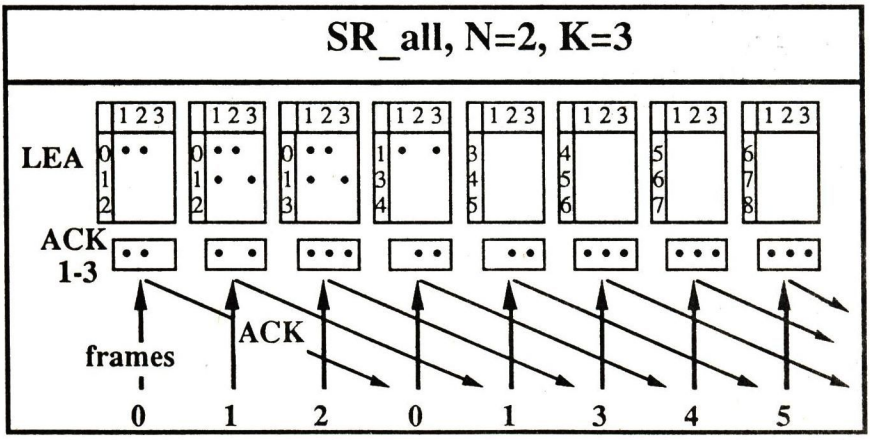


Fig.5: Performance of SR_all

SR-all:* Here the number m' of copies during a retransmission of a frame depends on the number K' of those stations not having accepted this frame instead of K. For m' the optimal value of m for K' receivers in SR_all is chosen.

In the single receiver case (point-to-point) the protocols coincide again.

The protocols may be classified by the conditions which have to be fulfilled so that ACKs are accepted (fig. 6).

	ACK at the same time	error free predecessor
GBN 0	●	●
GBN_all		●
SR 0	●	
SR_all		

Fig.6: Classification of protocols

4. THROUGHPUT COMPARISON

We compare throughput of the protocols in satellite networks with a round trip delay of about 0.54 s. The throughput T of a protocol (as function of the frame error rates and the number of receivers) comes to

$$T = \frac{\text{number of frames transmitted successfully to all receivers}}{\text{total number of transmitted copies of frames}}.$$

We make the following assumptions:
- All receiving stations have equal properties.
- Uplink and downlink errors are randomly distributed with frame error rates b_u and b_d. Uplink errors affect all receiving stations, whereas a downlink error affects the corresponding station only; b_d is equal for all stations. The downlink error processes are independent, because most fading effects are local.
- The transmission of acknowledgements over a return channel is error free.
- The sender is saturated; there are always frames waiting for transmission.

Throughput of the GBN_0, SR_0(infinite buffers), and SR_all/SR_all*(infinite buffers) protocols is computed analytically. For performance evaluation of GBN_all, GBN_all*, SR_0, SR_all and SR_all* (finite buffers) simulation has been used.

4.1 Throughput Analysis

Let $p_{m,K}$ be the probability that a single frame transmitted m times subsequently (a station accepts, if it receives one error free copy) is accepted by all K receivers, then

$$p_{m,K} = \sum_{i=0}^{m} \binom{m}{i} b_u^{m-i} (1-b_u)^i (1-b_d^i)^K .$$

GBN_0: If m copies of a frame are sent subsequently, the frame is accepted by all receivers with probability $p_{m,K}$; if the frame is not accepted and must be repeated later, the following N copies of other frames are also lost. Thus, throughput may be computed as

$$T_{GBN\_0} = \max_m \left(\frac{p_{m,K}}{m + (1-p_{m,K})\, N} \right).$$

GBN_all, GBN_all:* The actual system state may be expressed by the K-tupel $(n_1,...,n_K)$, where n_i indicates the number of those frames accepted by receiver i which have not yet been accepted by all receivers, $0 \le n_i \le [(N + 1) / m]$. As the receivers have identical properties, we may assume $n_1 \le ... \le n_K$. Due to the huge number of states exact analytical evaluation of the throughput seems to be impracticable, thus we have used simulations. For single copy GBN_all Gopal, Jaffe [6] have proposed and validated an approximation model. Unfortunately, it cannot easily be extended to the multicopy case.

SR_0(infinite buffers): If we assume infinite buffers at the receiver, throughput is equal to the probability that a frame sent in m (m optimal) subsequent copies is accepted by all receivers,

$$T_{SR\_0(\infty b)} = \max_m \left(p_{m,K} / m \right).$$

SR_all, SR_all (infinite buffers):* Here the outcomes of previous transmissions are considered. As we assume infinite buffers, frames may be buffered at a receiver for an arbitrary time. So it is unfavourable to transmit copies of a frame which might be unnecessary. Therefore, throughput is maximized, if m=1 is chosen, and comes to (cf. [16] for b_u =0)

$$T_{SR\_all(\infty b)} = T_{SR\_all^*(\infty b)}$$

$$= 1 / \sum_{m=1}^{\infty} m \Pr \left(\begin{array}{c} \text{a frame is accepted by all K receivers} \\ \text{exactly after transmission of m copies} \end{array} \right)$$

$$= 1 / \sum_{m=0}^{\infty} \Pr \left(\begin{array}{c} \text{a frame is not accepted by all K receivers} \\ \text{after transmission of m copies} \end{array} \right)$$

$$= 1 / \sum_{m=0}^{\infty} (1 - p_{m,K}) \; .$$

Throughput of SR_all(infinite buffers) is an upper bound for the throughput of all realistic (finite buffers) ARQ protocols.

SR_0, SR_all, SR_all (finite buffers):* To take into account finite buffers at the receivers simulations are used for the performance evaluation of these protocols.

4.2 Simulation

The simulation program for the point-to-multipoint scenario is written in QNAP2 language, cf. [10], [11], and [15]. QNAP2 (Queueing Network Analysis Package) is a program package developed by INRIA (Institut National de Recherche en Informatique et en Automatique) and Bull for evaluation of queueing networks based on the principles of parallel processes.

The duration of a simulation run was chosen to 20 s with a transmission rate of 1,000 frames/s, that means 20.000 frames were sent and additional 600 frames were generated to get a stationary behaviour. One simulation, consisting of 10 or 20 simulation runs, consumed between 10 and 25 hours CPU-time on a CELERITY 1260 D processor. To calculate confidence intervalls we used batch means algorithm, cf. [5], always computed for a confidence level of 95%.

4.3 Results

We evaluated the throughput of five protocols (GBN_0, GBN_all, GBN_all*, SR_0, SR_all) as function of frame error rates varying the parameter m, that is the number of copies, and choosing the optimal value for it. Results presented in fig. 7 to 11 base on a transmission rate of 1,000 frames/s. Due to the propagation delay of a satellite link (geostationary orbit) N=540 is chosen.

GBN_0 (analysis): In the case of high frame error rates (e.g. b_u, $b_d \geq 0.03$) throughput may be increased by transmitting a sequence of copies instead of a single one. Assuming $b_u = b_d = 0.03$ m_should be 4 for more than 15 stations and 3, if there are less (fig. 7).

GBN_all, GBN_all (simulation):* The optimal number of copies is nearly the same as above. If the satellite network comprises many receivers GBN_all, GBN_all* lead to a better throughput than GBN_0 (fig. 10: $b_u = b_d = 0.03$, fig. 9: $b_u = b_d = 0.1$).Throughput results of GBN_all vs. b_u, b_d ($b_u = b_d$) are presented in fig. 11.

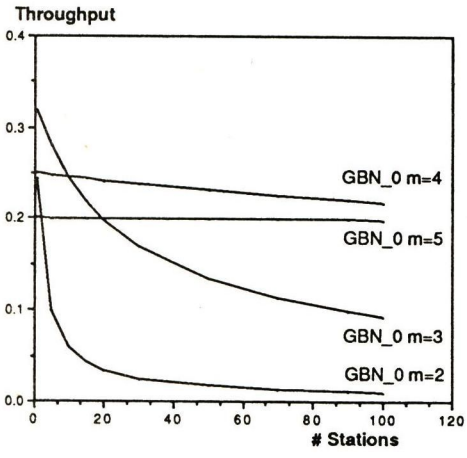


Fig. 7 : Throughput vs.# Stations
$b_u = b_d = 0.03$

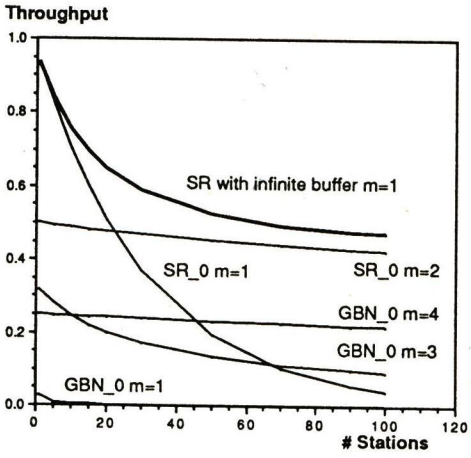


Fig.8 : Throughput vs.# Stations
$b_u = b_d = 0.03$

SR_0, SR_all with infinite buffer (analysis): The SR_all protocol with infinite buffers and m =1 leads to the best throughput using an ARQ protocol. If only a few (K<5) receivers are listening, throughput of the SR_0 protocol does not differ significantly from the throughput of the SR_all protocol. But with increasing number of stations K, throughput decreases rapidly. So it is better to choose two copies (if K>20) transmitted subsequently (fig. 8).
SR_all (simulation): The optimal values of m are 1 (K≤5) and 2 (K>5), cf. fig. 10.

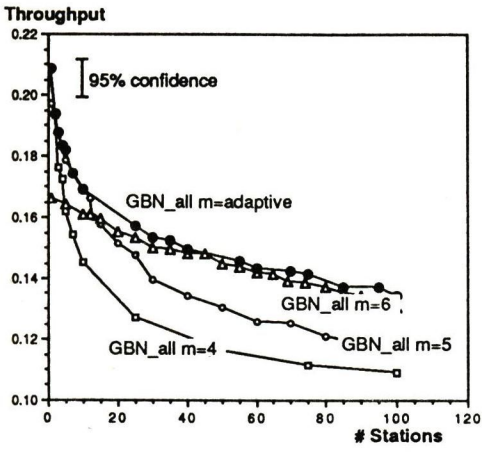


Fig.9 : Throughput vs. # Stations
$b_u=b_d=0.1$

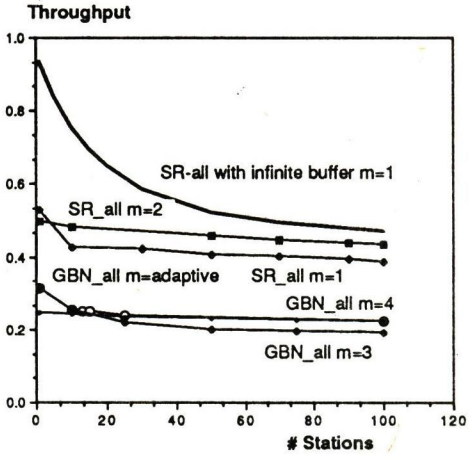


Fig.10 : Throughput vs. # Stations
$b_u=b_d=0.03$

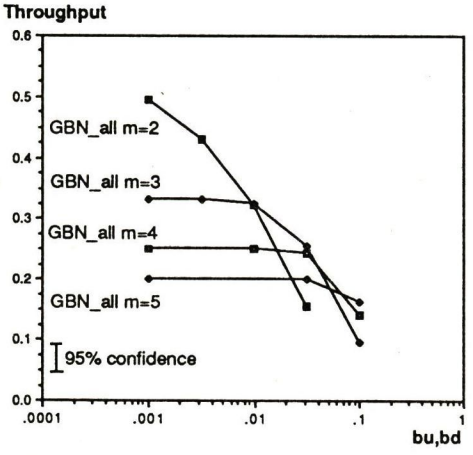


Fig.11 : Throughput vs. frame error rates
number of stations = 10

5. CONCLUSIONS AND FUTURE WORK

In this paper we have discussed some multicopy go-back-N and selective repeat protocols for multipoint satellite communication. Throughput evaluations show that for relatively high frame error rates (e.g. during fading periods or caused by small antennas) buffering ACKs increases throughput considerably. Moreover, multicopy protocols may perform much better than single copy protocols. The optimal number of copies sent subsequently depends on the

number of stations, the frame error rate, and the chosen protocol. The importance of buffering ACKs and of multicopy transmission increases with the number of receivers. Due to the large round trip delay in satellite networks the advantages of multicopy transmission and ACK buffering are particularly evident. Thus, during fading periods a satellite channel may continue data transmission even though throughput decreases, whereas the link breaks down if a single copy protocol is used.

Next we will consider the scheduling of acknowledgements of many receivers in multipoint ARQ protocols; for every frame acknowledgements of all receivers have to be transmitted separately to the sender. But without reducing throughput considerably, receivers may send an acknowledgement for a sequence of frames (e.g. 10 or 100), respectively, so that the round trip delay is increased correspondingly. Furthermore, we want to evaluate the performance of multipoint ARQ protocols considering inhomogeneous receivers and burst errors which are typical of satellite links.

REFERENCES

[1] H. Bruneel, M. Moeneclaey, "On the Throughput Performance of some Continuous ARQ Strategies with Repeated Transmissions", IEEE Trans. Commun., pp. 244-249, March 1986.

[2] S. B. Calo, M. C. Easton, "A Broadcast Protocol for File Transfers to Multiple Sites", IEEE Trans. Commun., p. 1701-1706, 1981.

[3] S. R. Chandran, S. Lin, "A Selective-Repeat ARQ Scheme for Point-to-Multipoint Communications and Its Throughput Analysis", Proc. ACM SIGCOM Conference, pp. 292-301, Stowe, VT, August 1986.

[4] Y. Chang, C. Leung, "On Weldon's ARQ Strategy", IEEE Trans. Commun., pp. 297-300, March 1984.

[5] G. Fishman, "Principles of Discrete Event Simulation", J. Wiley & Sons, New York, 1978.

[6] I. S. Gopal, J. M. Jaffe, "Point-to-Multipoint Communication over Broadcast Links", IEEE Trans. Commun., vol. COM-32, pp. 1034-1044, September 1984.

[7] M. Hübner, F.Sausen, O.Spaniol, "Evaluation of Application-Oriented Satellite Link Protocols", Proceedings IFIP TC 6 International Conference on Inform. Network and Data Comm., Ronneby Brunn, Sweden, May 1986.

[8] K. Mase, T. Takenaka, H. Yamamoto, M. Shinohara, "Go-Back-N ARQ Schemes for Point-to-Multipoint Satellite Communications", IEEE Trans. Commun., vol COM-31, pp. 583-590, April 1983.

[9] J. M. Morris, "On another Go-Back-N ARQ Technique for High Error Rate Conditions", IEEE Trans. Commun., vol. COM-26, pp. 187-189, 1978.

[10] D. Potier, "New Users' Introduction to QNAP2", INRIA Rapports Techniques No. 40, October 1984.

[11] "QNAP2 Reference Manual, version V03", INRIA and Bull, May 1984.

[12] U. Quernheim, "Satellite Communication Protocols - a Performance Comparison Considering On-Board Processing", Proc. EUROCON'88, 8th European Conference on Electrotechnics, Stockholm, Sweden, pp. 269-272, 1988.

[13] A. R. K. Sastry, "Improving Automatic Repeat-Request (ARQ) Performance on Satellite Channels under High Error Rate Conditions", IEEE Trans. Commun., vol. COM-23, pp. 436-438, 1975.

[14] D. Towsley, S. Mithal, "A Selective Repeat ARQ Protocol for a Point-to-Multipoint Channel", Proc. INFOCOM'87, pp. 521-526, San Francisco, CA, March 1987.

[15] M. Véran, D. Potier, "QNAP2: A Portable Environment for Queueing äSystems Modelling", INRIA, Rapports de Recherche, No. 314, June 1984.

[16] J. L. Wang, J. A. Silvester, "Optimal Adaptive ARQ Protocols for Point-to-Multipoint Communication", Proc. INFOCOM'88, pp. 704-713, New Orleans, March 1988.

[17] Jr. E. J. Weldon, "An Improved Selective-Repeat ARQ Strategy", IEEE Trans. Commun., pp. 480-486, March 1982.

AN APPROXIMATE ANALYSIS OF MULTI-HOP S-ALOHA NETWORKS WITH OR WITHOUT IMMEDIATE ACKNOWLEDGMENTS

Jozef WOZNIAK

Institute of Telecommunications
Technical University of Gdansk
Gdansk, Poland

Two operational network schems, based on the S-ALOHA protocol are considered.Their usefulness in multihop homogeneous networks is analysed. In the first case positive acknowledgments are sent back to source stations piggybacked to data packets. Therefore the sender has to wait a random time interval for acknowledgment. In the second channel access scheme the time axis is divided into slots consisting of two separate segmentes, one for data packets and the other one for acks. In this case acknowledgement is immediate. In both cases it has been assumed that stations are equipped with four basic buffers /buffers of new and transit packets and buffers of their copies/. For the sake of simplicity, memoryless properties of the input processes and service time distributions have been assumed. An approximate analysis demonstrates that the second model ensures much better performance parameters under moderate and heavy load conditions.

1. INTRODUCTION

The first use of a random access protocol for computer communications was the centralized ALOHA system with a star topology and single-hop transmission [1]. More recently, the concept of a packet radio network allowing multi-hop communication has been developed [2],[3]. The performance of such a network is analysed in the present paper.

We consider two operational network schemes based on the S-ALOHA protocol. In the first case we assume that positive acknowledgments /acks/ are sent back to source stations piggybacked to data packets exchanged between nodes. Therefore the sender has to wait a random time interval for acknowledgment. In this scheme network channels are utilized according to the traditional S-ALOHA procedure.In the second case we divide the time axis into slots consisting of two separate segments. The first of these segments is used for data packets, the second for acknowledgments. It is easy to verify that all positive acks can be sent without interference. With this scheme we achieve almost instantaneous acks. Nevertheless, when there is no successful transmission during the first segment, the second part of a slot is not used, resulting in a certain decrease of channel utilization. In both cases we assume that each station is equipped with four basic buffers. Buffers of new and transit packets are used to store up to K packets waiting for transmission to other nodes.

Buffers of copies are assigned to packets waiting for acks. Since in the second case information about successful or unsuccessful packet transmission is known by the end of a given time slot, the size of this buffer can be limited to one packet only.

Here we analyse homogeneous networks and neglect edge effects. Simultaneously, we assume that the number m of neighbour stations is large enough for the arrival processes to have memoryless / or quasimemoryless/ properties. We investigate station and network performance, i.e., throughput and delay, taking into account simple queueing models with priorities. We give approximate performance parameters and also present the conditions for the successful use of each of these schemes.

2. CHANNEL ACCESS SCHEMES AND PERFORMANCE ANALYSIS

2.1. Model A

In the first case, users get access to the common channel according to the S-ALOHA scheme. The only modification, as compared to the traditional S-ALOHA, is that we introduce priorities into the service of the buffers illustrated in Fig. 1. A higher priority is given to transit traffic. In this model the highest service priority is given to buffers of copies.

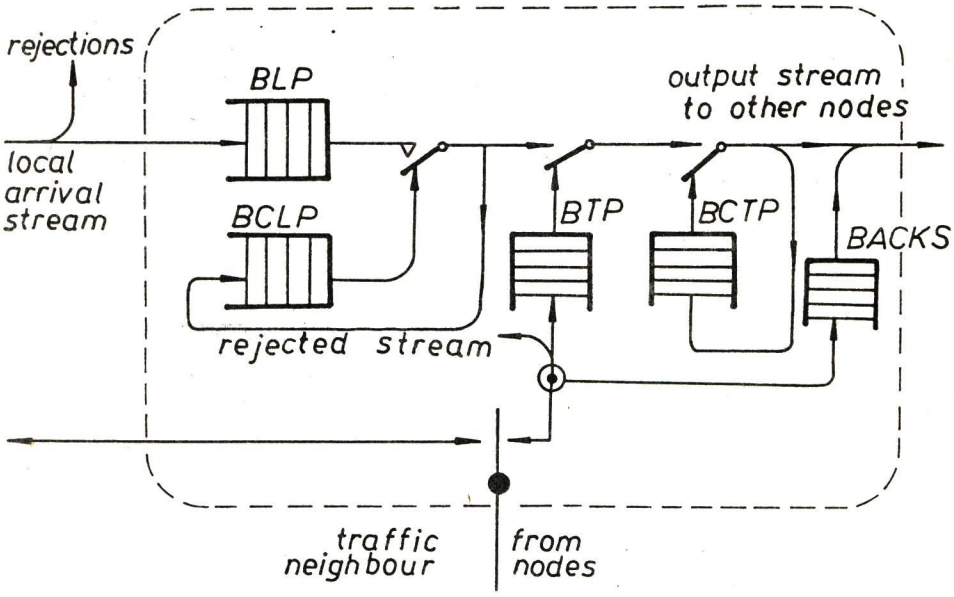


Fig. 1. Buffer configuration at a node. BLP - buffer of local packets, BCLP - buffer of copies of local packets, BTP - buffer of transit packets, BCTP - buffer of copies of transit packets, BACK - buffer of acks.

The numbers of packets in these buffers are interconnected.This means that in a given node different input and output packet streams form processes with the memory. The precise analysis of this buffer configuration becomes enormously complex. So for the sake of simplicty and to enable us to

obtain relatively simple final expressions we shall apply an aproximate analysis, assuming memoryless properties for these processes. We shall consider disrete - time Markov chain models for all the buffers presented in Fig. 1, using the same method of analysis in both network models. The buffers will be served in the following order: BCTP, BCLP, BTP and BLP. Additionally, we shall assume that there are always m-1 other identical stations within hearing range of each network station.

Under the above conditions one can analyse all buffers independently, by considering buffer input and output as practically memoryless processes. All the buffers can be analysed as in [4], [5]. This method leads to relatively simple expressions for the channel throughput, average queue lengths in buffers, and the average delay times of packet transmissions. This also allows us to get approximate input rates satisfying stable channel conditions.
For each network station we can describe the station channel throughput as the sum of four components

$$s = s_l + s_t = s_{l1} + s_{l2} + s_{t1} + s_{t2} \tag{1}$$

where

s_l , s_t - local and transit channel throughputs respectively,

s_{l1} , s_{l2} - local throughput components related to buffers BLP and BCLP respectively,

s_{t1} , s_{t2} - transit throughput components transmitted from buffers BTP and BCTP respectively.

Therefore, the total throughput in a local area network with m nodes is

$$S = ms \tag{2}$$

A very important factor influencing the network /station/ performance in networks with contention protocols is the total traffic . For a single station this is

$$g = g_{l1} + g_{l2} + g_{t1} + g_{t2} \tag{3}$$

while the total traffic in a channel used by m stations is G = mg.
The separate terms of expression (3), related to buffers BLP, BCLP, BTP and BCTP respectively, are defined in the following way:

$$g_{l1} = \Pr(\text{no time-out in BCLP})\ \Pr(\text{no time-out in BCTP})$$

$$[1 - \Pr(\text{K copies of packets in BCLP})]$$

$$\Pr(\text{no data packets in BTP}) \cdot [1 - \Pr(\text{no data packets in BLP})] =$$

$$= \Pr^*(t^*_{BCLP})\ \Pr^*(t^*_{BCTP})\ (1 - P_{KclP})\ P_{OtP}\ (1 - P_{OlP}) \tag{4}$$

$$g_{l2} = \Pr(\text{no time-out in BCTP})\ \Pr(\text{time-out in BCLP}) =$$

$$= \Pr^*(t^*_{BCTP})\ \Pr(t^*_{BCLP}) \tag{5}$$

$$g_{t1} = \Pr(\text{no time-out in BCLP})\ \Pr(\text{no time-out in BCTP}) \cdot$$
$$[1 - \Pr(\text{K copies of packets in BCTP})]$$
$$[1 - \Pr(\text{no data packets in BTP})] =$$
$$= \Pr^*(t^*_{BCLP})\ \Pr^*(t^*_{BCTP})\ (1 - P_{Kctp})\ (1 - P_{Otp}) \qquad (6)$$

$$g_{t2} = \Pr(\text{time-out in BCTP}) = \Pr(t^*_{BCTP}) \qquad (7)$$

On the other hand the general expressions for the channel throughput components for a given station are:

$$s_{l1} = g_{l1}(1-g)^{m-1}(1-\alpha P_{Ktp})(1-g)^{m-1} =$$
$$= s'_{l1}(1-g)^{2(m-1)}(1-\alpha P_{Ktp}) = g_{l1}(1-\alpha P_{Ktp}) \qquad (8)$$

$$s_{l2} = g_{l2}(1-g)^{2(m-1)}(1-\alpha P_{Ktp}) = s'_{l2}(1-g)^{m-1} \qquad (9)$$

$$s_{t1} = g_{t1}(1-g)^{2(m-1)}(1-\alpha P_{Ktp}) = g_{t1}(1-P_{Ktp})\beta^2 \qquad (10)$$

$$s_{t2} = g_{t2}(1-g)^{2(m-1)}(1-\alpha P_{Ktp}) = g_{t2}(1-\alpha P_{Ktp})\beta^2 \qquad (11)$$

where

$\beta = (1-g)^{m-1}$ - the probability of no transmission through the channel during the transmission in question,

$\alpha = \frac{s_t}{s}$ - a factor describing the probability that a received packet is a transit one and must be stored in BTP.

In the above formulae s'_{l1}, s'_{l2} /and s'_{t1}, s'_{t2}/ denote the probabilities of successful packet reception at a destination node after their transmission from buffers BLP or BCLP /or BTP and BCTP/ respectively.
Data packets are always directed to only one of the m-1 stations located within range of the source station. After successful packet reception, this packet is relayed with a probability of one by the receiving station and an acknowledgment piggybacked to a data packet is always sent back to the source station before the time-out t^* for this packet expires. This means that the main cause of time-outs /i.e., packet retransmissions/ are interferences in the channel.
A packet received without interference, carrying an ack may be rejected due to the lack of space in buffer BTP. Nevertheless, the ack will always be accepted.
To find the proper expressions for the station throughput /eqs. (8)-(11) / we must analyse all the buffers shown in Fig. 1. Markov chain analysis can be used to study the performance of a network whose nodes are provided with buffers [6] .
To analyse the behaviour of a BCTP we consider the simplified Markow chain illustrated in Fig. 2. All states k of a BCTP are divided into two substates, k' and k''. Retransmissions caused by time-out take place only in substates k''. The analysis of this buffer is very complex. Notice that servicing packets of copies in a BCTP /and in a BCLP as well/ is not arranged according to FIFO, since acks may arrive

out of transmission order. In accordance with previous assumptions, time-out on transit and local packets is denoted by t^*.

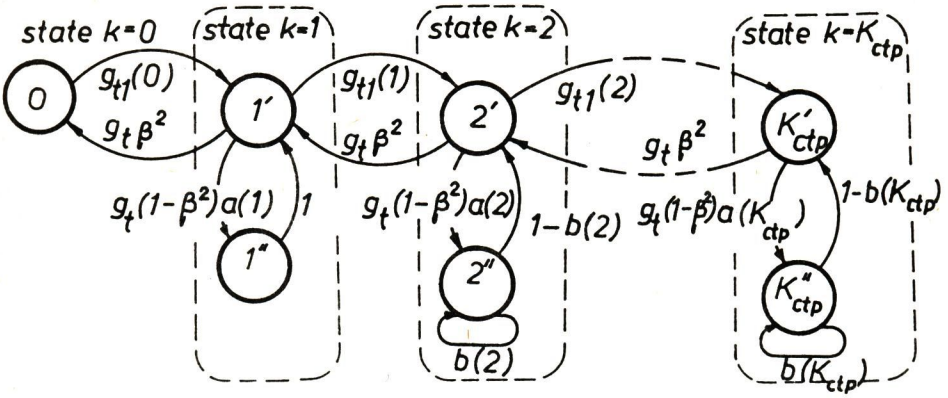


Fig. 2. Markov chain model for BCTP.

We assume that the acks rate is constant and equal to $g_t \beta^2$. The conditional probabilities of time-out /retransmission/ clearly depend on the number of packets in a BCTP.
The transmission rates g (k) from a BTP are also influenced by these buffer states. From a simple analysis it follows that for $t^* < K_{ctp}$ buffer states $k > t^* + 1$ cannot be reached. When $k = t^* + 1$ we always have one packet that needs to be retransmitted. It is possible to insert in a BCTP only one packet from a BTP during a slot. Thus, when $k = t^* + 1$ we have $t^* + 1$ packets spending 0,1,..., and t^* time slots there. The last packet must be retransmitted. From the above one can see that the transit rate $g_{t1}(k)$ of new arrivals from a BCT falls dramatically when a BCTP is almost full. Using the Markov chain illustrated in Fig. 2 we get

$$P_k = P_0 \frac{\prod_{i=0}^{k-1} g_{t1}(i)}{(g_t \beta^2)^k}, \quad k = 0,1,2,..,K, \quad t^* > K_{ctp} \tag{12}$$

where

$$P_0 = \frac{1}{1 + \sum_{k=1}^{K_{ctp}} \prod_{i=1}^{k} \frac{g_{t1}(i-1)}{g_t \beta^2}} \tag{13}$$

Additionally, the conditional probabilities $P_{k''}$ of time-out are as follows:

$$P''_k = P(t^* | k) = \frac{g_t (1 - \beta^2)\, a(k)}{1 + g_t (1 - \beta^2)\, a(k) - b(k)}, \quad k = 1,2,...,K_{ctp} \tag{14}$$

Thus, the average probability of time-out takes the form

$$P(t^*) = \sum_{k=1}^{K_{ctp}} Pr(t^* | k) \quad P_k = g_t (1 - \beta^2) \sum_{k=1}^{K_{ctp}} \frac{a(k) \; P_k}{1 + g_t (1 - \beta^2) \; a(k) - b(k)} \tag{15}$$

In the above formulae a(k) is the probability proportional to the number of events related to k' with one packet waiting t^*- 1 time slots.
A BCLP can be analysed in a similar way.
Now let us consider a BTP. When analysing this buffer we shall assume that the transition rates between states are identical, i.e., g_{t1}
Pr (new tp. arrival | k) = γ_t , and Pr (tp. transm. | k) = $g'_{t1} = \frac{g_{t1}}{1 - P_{0tp}}$

For this buffer we have the state probabilities

$$P_k = P_0 \, \rho_{tp}^{k-1} = \rho_{tp}^{k-1} \frac{1 - \rho_{tp}}{1 - \rho_{tp}^{K_{tp}+1}} \; , \; k = 1,2,\ldots,K \tag{16}$$

where

$$\rho_{tp} = \frac{\gamma_t}{g'_{t1}} \tag{17}$$

A BLP can be similarly analysed. In this case the probability of new arrivals of local packets Pr (new lp. arrival | k) is λ , and Pr (lp. transm. | k) equals $g'_{l1} / (1 - P_{0lp})$, $k = 0,1,2,..,K_{lp}$.

To complete our analysis, we need to evaluate packet transmission delays for local and transit packets. For packets from a BTP and BLP successfuly transmitted at the first attempt, delys D_{BLP} and D_{BTP} are obtained by Little's formula:

$$D_{BTP} = \frac{N_{BTP}}{S_{t1}} + \frac{t^*}{2} \tag{18}$$

$$D_{BCLP} = \frac{N_{BCP}}{S_{l1}} + \frac{t^*}{2} \tag{19}$$

In the above formulae N_{BTP} and N_{BLP} describe the respective average queue lengths in a BTP and BLP. For local and transit packets retransmitted from buffers of copies we have

$$D_{BCTP} = \left(\frac{g_{t2}}{S_{t2}} - 1 \right) t^* + \frac{t^*}{2} \tag{20}$$

$$D_{BCLP} = \left(\frac{g_{l2}}{S_{l2}} - 1 \right) t^* + \frac{t^*}{2} \tag{21}$$

2. 2. Model B.

In this model we analyse a set of buffers similar to that presented in Fig. 1. The only difference as compared to model A is that the size of the buffers of copies BCLP and BCLP are one packet only. As in model A these buffers are served with the highest priorities. Local and transit traffic buffers are served according to the Station Priority Scheme /SPS/ [7] based on the S-ALOHA protocol illustrated in Fig. 3.

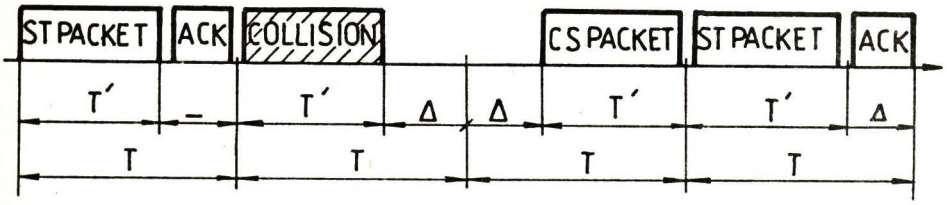


Fig. 3. Example of channel operation according to the SPS.

This scheme, used in a star configuration with the Central Station (CS), is similar to that described in [8] and states that:

- When a local station has a data packet to transmit, it operates according to the S-ALOHA protocol and sends the packet at the beginning of a time slot /see Fig. 3/.
- If a packet is not acknowledged at the CS, the station repeats its transmission after a randomly chosen delay.
- The CS transmits the ack of a correctly received packet in the same slot.
- If the CS has a packet to transmit it waits for a slot not used by stations.

In the papers quoted /i.e. [4],[5] and [7]/ the SPS protocol was used in a star network with the Central Station relaying the traffic from local stations [7] and perhaps creating some traffic of its own [4], [5]. Simple analysis of the SPS protocol also demonstrates its great usefulness in multihop networks. The SPS allows us to obtain almost instantaneous acknowledgments. In model B, contrary to the orginal SPS algorithm, all frames /slots/ have the same structure. Each station can start to transmit its packet only at the beginning of a slot, while period Δ is always reserved for ack packets /see Fig. 4/. An ack packet in the same slot following the transmitted one is considered a positive ack by a source node. Otherwise the transmitted packet is treated as a lost one /no ack in the same slot/. Ack packets are, as is easily verifiable, unique and their interpretation is clear. In this case the conditional probability of retransmission in state 1 is $1 - \beta$, since acks are

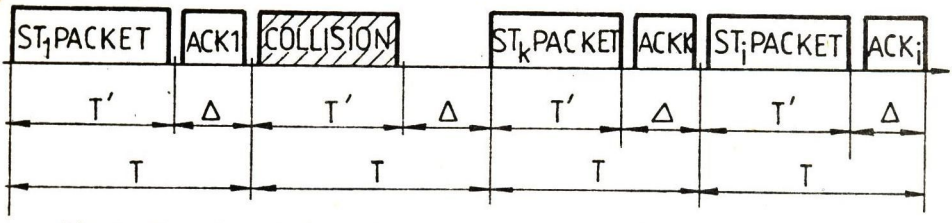


Fig.4. Example of channel operation in model B.

sent without interference. In model B, time-out is presumed to be less than one slot. By the end of a given slot we get a positive or negative ack. To avoid continuous retransmissions of packets we shall assume a p-persistent retransmission model. We can also distinguish four channel throughput components:

$$s_{11} = P_{0ctp} \cdot P_{0clp} \cdot P_{0tp}(1 - P_{0lp})\ \beta(1 - \alpha P_{Ktp}) =$$

$$= g_{11}\ \beta(1 - \alpha P_{Ktp}) \qquad (22)$$

$$s_{l2} = P_{0ctp} \quad P_{1clp}\, p\,(1 - \alpha P_{Ktp}) = g_{l2}(1 - \alpha P_{Ktp})\,\beta \qquad (23)$$

$$s_{t1} = P_{0ctp} \quad (1 - P_{0tp})\,\beta\,(1 - \alpha P_{Ktp})\,P_{0clp} \qquad (24)$$

$$s_{t2} = P_{1ctp} \quad p\,\beta\,(1 - \alpha P_{Ktp}) \qquad (25)$$

The correct formuale for the above probabilities can be obtained by analysing approximate queueing models related to appropriate Markov chains.

3. EXAMPLE RESULTS

To compare the effectiveness of these two channel access schemes in multihop networks we examine their performance parameters. We analyse the throughput and the expected packet delays with respect to a local area network with m homogeneous nodes in the communication range of each network station. We assume that the number m of neighbour stations is large enough for the assumption that all arrival streams are Poisson processes to hold. Under these conditions we get $mg = G$ and $\beta = e^{-G}$. Additionally we assume that the BTP and BLP have infinite capacities, and the probability of time-out in model A can be simply expressed as $g_t = g_t\,(1 - \beta^2)$ by ignoring the time-out value t^* and the BCTP size. The approximate formulae for the transit and local throughput values for model A are as follows:

$$S_{t1}^{(A)} \cong G_t^4 \qquad (26)$$

$$S_{t2}^{(A)} \cong G_t \quad (1 - \beta^2)\beta^2 \qquad (27)$$

$$S_{l1}^{(A)} \cong (G - G_t)\,\beta^4 \qquad (28)$$

$$S_{l2}^{(A)} \cong (G - G_t)\,(1 - \beta^2)\,\beta^2 \qquad (29)$$

while for model B they

$$S_{t1}^{(B)} \cong G_t\,\beta^2 \qquad (30)$$

$$S_{t2}^{(B)} \cong G_t\, p\,(1 - \beta)\beta \qquad (31)$$

To show the effect of Δ on the actual network throughput we have to consider the behaviour of $\delta\, S_{ti}^{(B)}$ and $\delta\, S_{li}^{(B)}$, i=1,2, where $\delta = T/(T + \Delta)$. Figures 5 and 6 illustrate the transit or total throughput variations for different values of $c = G_t/G$, p and δ. It clearly follows from Fig. 7 that model B is successful especially under moderate and heavy loads.
In Fig. 7 delays related to transit packets in a BCTP are shown. Even for a very short time-out $t^*_{min} = 2$ in model A, model B exhibits significantly shortes delays D_{BCTP}.

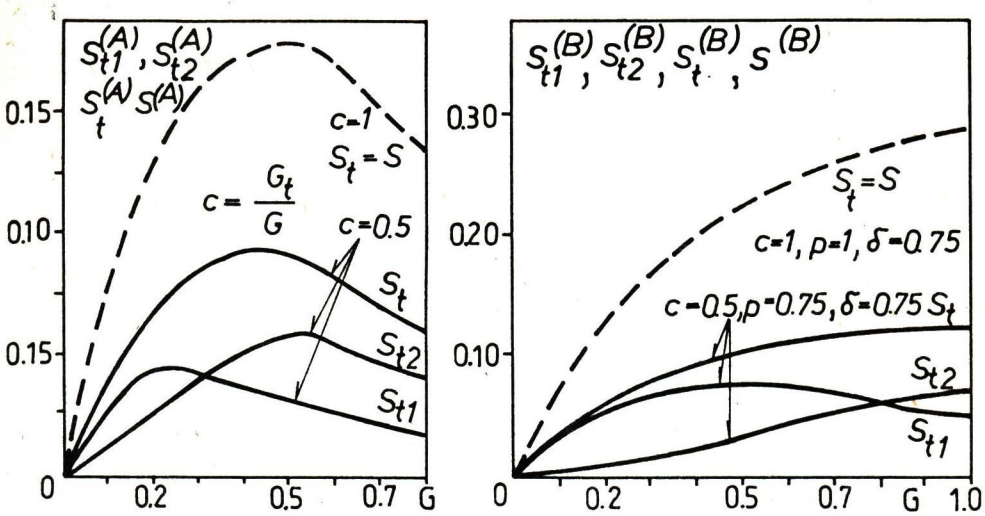


Fig. 5. Transit and total throughput variations vs. G

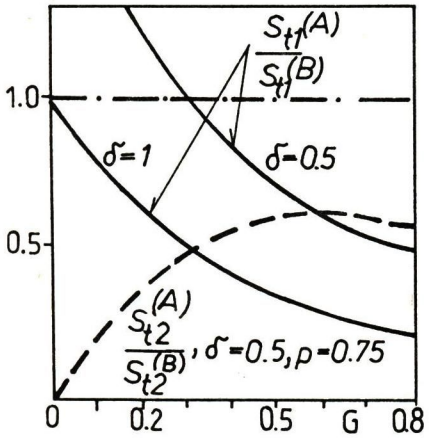


Fig. 6. Ratios of transit and total throughputs for models A and B.

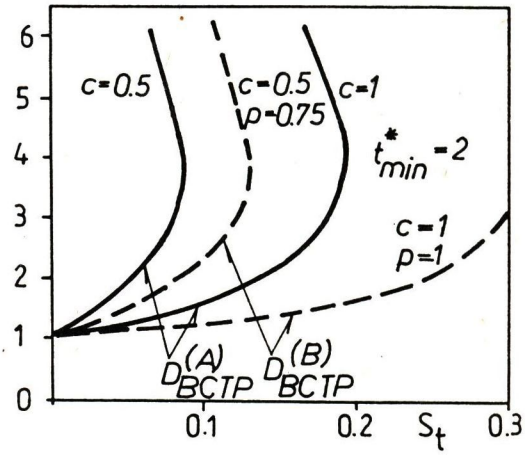


Fig. 7. D_{BCTP} delay time variations vs. S_t .

4. CONCLUSIONS

The main difference between the two models described concerns the method of acknowledgment. In the first case acks are sent piggybacked to data packets. Thus we observe their partial lost, due to packet interference. This results in long packet acknowledgment delays. In the case of limiting buffer sizes it decreases the channel /network/ throughput significantly. On the other hand, in model B, positive acks are sent immediately, i.e., with a delay of one slot. With this scheme buffers of copies can be of a

certain minimum size. Nevertheless, in this case there is certain decrease in the channel bandwith, caused by the special organizing Δ period during each frame used for ack purposes.
The approximate analysis shows that for sufficiently short periods Δ model B offes much better performance parameters for a regular, homogeneous network.Such a model is much more stable as well.

REFERENCES

[1] - Abramson N.: The ALOHA system: Another alternative for computer communications. Proc. Fall Joint Comp. Conf. AFIPS 1970, pp. 281-285.
[2] - Kahn R.E., Gronemeyer S.A., Burchfield J., Kunzelman R.C.: Advances in radio technology. Proc. IEEE, vol. 66, 1978, pp. 1468-1496.
[3] - Takagi H., Kleinrock L.: Throughput - delay characteristics of some slotted ALOHA multihop packet radio networks. IEEE Trans. Commun., vol. COM-33, 1985, pp. 1200-1207.
[4] - Wozniak J.: Study of a two-hop S-ALOHA star network with an intelligent reperater. Proc. 6 th Intern. Conf. on Systems Engineering, Coventry 1988, pp. 269-276.
[5] - Sobczak W., Wozniak J.: Performance analysis of a star S-ALOHA packet radio network with different models of the central station. Proc. COMPUERO'89, Hamburg 1989, pp. 4.112 - 4.117
[6] - Kleinrock L.: Queueing Systems. vol. I, Theory. New York. 1975.
[7] - Fratta L., Wozniak J.: Performance analysis of a two-hop star network with S-ALOHA multiple access scheme. Proc. 12 th ITC, Torino 1988, pp. 5.3B.6.1 - 6.9 or North-Holland.
[8] - Scholl M., Kleinrock L.: On a mixed mode multiple access scheme for packet-switched radio channels. IEEE Trans. Commun., vol.27, 1979, pp. 906-911.

Towards High Speed Networking in Local and Metropolitan Areas

Peter Martini
Department of Computer Science (Informatik IV)
Aachen University of Technology
Ahornstr. 55; D-5100 Aachen; West Germany
☎: +241/80-4522; FAX: +241/80-6295; e-mail: ...unido!rwthinf!martini

Abstract

In the 1980's, dramatically decreasing costs of optical fiber paved the way for the advent of a new generation of computer networks offering raw transmission capacities in the 100 Mbit/s range. And for the remainder of this century, there is no end in sight for the exponential performance growth of communications technology. Future capacities of fiber optic transmission lines will be enormous, with data rates limited not by the transmission lines, but by devices on either end of the lines. This paper provides an overview of the current status of upcoming standards for High Speed Local Area Networks (HSLANs) and Metropolitan Area Networks (MANs). It addresses applications, architectural alternatives, performance issues, protocols and unsolved problems.

I. Introduction

The IEEE project 802 provided the first generation of standard LAN protocols which increased the transmission rate from 1 Mbit/s experimental LANs to 10 Mbit/s. In late 1982, the Standards Committee X3T9.5 started work on the Fiber Distributed Data Interface (FDDI) which has resulted in a set of standards for the next order of magnitude in LAN performance. Basically aiming at the I/O Channel (backend) application, FDDI grew from the original ideas to satisfy other applications such as front-end high-performance Local Area Networks and LAN backbones. FDDI now is broad enough to be individually optimized according to the requirements of specific environments. With the development of FDDI-II the protocol even is able to cope with the requirements imposed by real-time applications such as digital voice, video, or sensor and control data streams.

In the IEEE standards committee on Metropolitan Area Networks (MANs) the need to support voice and other fixed-rate services has led to the use of time-slotted protocols, where even the largest files are divided into short segments and reassembled at the destination. When it came to the choice of the size of these segments, the need to facilitate interconnection of MAN's and bridging to wide area networks required the convergence with the B-ISDN format. Therefore, the MAN working group adopted a 5-byte header and a rather short payload field of 48-byte.

Of course, some LAN architectures have survived in the marketplace without the benefit of standardization. On the other hand, IEEE and ANSI standards have been the major force in the acceptance of LAN technology. In this paper we discuss advantages and disadvantages of protocols proposed for high speed networks. Our study is restricted to FDDI (cf. [1 - 4]) and DQDB (cf. [5]) which is the protocol proposed for MANs. However, it should be noted that the next step in speeding up communication systems is already in front of us. The FDDI standard development began in 1983, i.e. before the IEEE 802 standards were published in 1985. Likewise, work has already begun on the next order of magnitude in LAN performance while FDDI standards are being completed. Many new computer applications (e.g. graphics and image processing) either use or create massive amounts of data. Thus, a need for gigabit LAN performance has been generated. However, the question whether a gigabit LAN should be of the same basic architecture as the standard LAN has not yet been answered.

II. Fiber Optic Networks

The 1980's saw optical fiber emerging as the most important medium for future communication systems. Today, the data rates of optical transmission systems installed in the field and carrying commercial service exceed 1 Gbit/s and research has shown a potential throughput of thousands of Gbit/s per fiber. Compared to fiber optic lines providing highly reliable data transmission at high transmission rates with low losses, metallic conductor transmission lines are clearly inferior and will eventually become obsolete.

The network topologies used in LAN schemes designed to yield high performance at high speeds may be broadly categorized into busses, rings and stars. Each of these schemes has its relative merits in terms of network utilization, delay, implementation costs, capability to support different services, complexity and reliability.

In contrast to conventional LANs, the majority of fiber optic networks employs unidirectional transmission. Some years ago, *'folded bus structure'* and *'dual bus structure '* have been proposed (see e.g. [6]) to allow broadcast communication in unidirectional bus systems with passively coupled stations. Several alternative bus structures have also been discussed. On the other hand, the dual bus structure is the only one which is proposed by a standardization committee: DQDB is based on this dual bus.

Since there is nothing like a high impedance light sensor, light detectors use much more signal power than their electric counterparts. This is the reason why the number of passive taps that may be cascaded on a linear bus is fairly limited. Therefore, in DQDB read and write functions may be implemented using either passive or active techniques, where 'active' means that stations regenerate and repeat each symbol. In ring networks, a stream of suitably encoded symbols is transmitted sequentially from one active station to the next. Thus, the problem of power reduction does not exist in this topology. FDDI is an example of a LAN scheme designed to operate at high data rates using a ring topology.

Optical star networks have been successfully implemented to bridge large geographical distances in CSMA/CD-LANs. Furthermore, physical star structures are proposed for broadband ISDN (see e.g. [7]). One important reason is that security and privacy problems are much harder to solve in bus structures or ring networks. On the other hand, star structures are currently not within the scope of standardization activities for HSLANs or MANs.

III. Emerging Standards

In this section, we describe the emerging standards for local and metropolitan area networks, namely FDDI and DQDB.

A. FDDI-Overview

The Fiber Distributed Data Interface (FDDI) is emerging from ANSI as a standard for the lowest two layers, i.e. the physical and the media access control levels of the OSI reference model. The FDDI fiber optic ring network (cf. [1] - [4]) provides a general purpose interconnection (100 megabit per second) among computers and peripheral equipment. Default values have been calculated on the basis of 1000 physical connections and a total fiber path length of 200 kilometers.

Basic FDDI is defined by a set of four documents: The PMD (Physical Layer Medium Dependent) document specifies the digital baseband point-to-point communication between stations in the FDDI network. It provides all services necessary to transport a suitably coded digital bit stream from station to station. In contrast to this, the PHY (Physical Layer Protocol standard specifies encoding, decoding, clocking and data framing. The Medium Access Control (MAC) standard of FDDI is primarily concerned with the delivery of frames including their insertion, repetition, and removal. Finally, a SMT (Station Management) standard defines the control necessary at the station level to manage the processes underway in the various FDDI layers.

Two standards are being developed as extensions of the basic FDDI:

- SMF-PMD is the Single Mode Fiber (SMF) version of PMD which increases the permissible fiber links from 2 km to 60 km.
- HRC (Hybrid Ring Control) describes an upwardly compatible version of FDDI (known as FDDI-II) which provides both the packet transport used in basic FDDI and an 'isochronous transport'.

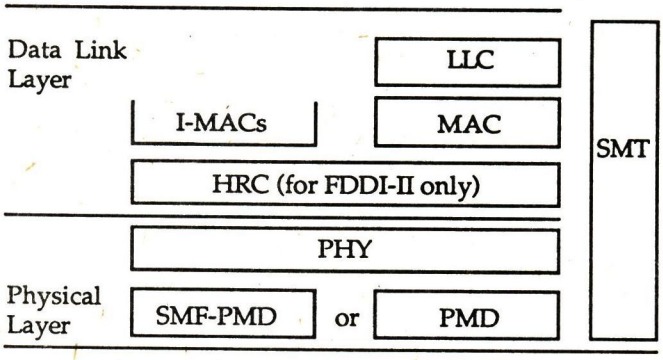


Fig. 1 Structure of FDDI Standards

In basic FDDI, a Token which is passed from station to station signifies the right to transmit information frames on the ring. If a station wants to transmit, it strips the Token from the ring. After the captured Token is completely received, the station shall begin transmitting its eligible queued frames. Each Frame starts with a Preamble which is at least 64 bit long. This preamble is followed by a starting delimiter (8 bit), Frame Control (8 bit), Destination Address (16 bit or 48 bit) and Source Address (16 bit or 48 bit). The length of the information field is variable, but is limited by the maximum frame length which is 4500 byte. This payload field is followed by a Frame Check Sequence (32 bit), End Delimiter (4 bit), and a Frame Status field ($\geq$ 12 bit) used for indicating errors or correct reception. In FDDI, a station immediately releases the token after transmitting a frame. Thus, pipelining may be achieved by multiple frames from several users simultaneously propagating around the ring.

FDDI supports two major classes of service, namely 'synchronous' (guaranteed bandwidth and response time for applications with predictable bandwidth and response time limits) and 'asynchronous' (dynamic bandwidth sharing for applications with bursty or potentially unlimited bandwidth requirements). Asynchronous service which instantaneously allocates bandwidth from the pool of ring bandwidth which is unallocated, unused, or both, is also used for applications whose response time requirements are less critical.

Asynchronous traffic is controlled by a Token Rotation Timer (TRT) which is maintained within each station. This timer is reset each time a token arrives before TRT reaches the 'operative Target Token Rotation Time' (T_OPR) which is negotiated during ring initialization via a 'Claim Token process'. FDDI guarantees an average TRT (or average synchronous response time) not greater than T_OPR, and a maximum TRT not greater than twice T_OPR.

Synchronous Transmission: Synchronous Frames may be transmitted whenever the station captures a token. In order to allow this kind of operation, each station has a known allocation of synchronous bandwidth which is allocated by the station management. It should be noted that the support for synchronous transmission is optional, and is not required for interoperability.

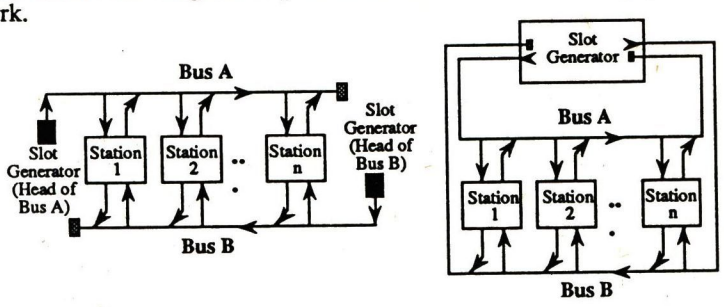
Asynchronous Transmission: Whenever an 'early' token is captured, i.e., a token arriving before TRT reaches T_OPR, the current value of TRT is saved in an asynchronous Token-Holding Timer (THT), and TRT is reset to time the next token rotation. THT is enabled (running) during asynchronous transmission and the difference between its current value and the target value (T_OPR) reflects the remaining asynchronous bandwidth available to this station.

The Medium Access Protocol described up to now supports fair access at a frame granularity. However, multiple levels of asynchronous priority may optionally be distinguished. For each implemented priority level, a threshold value is established, forming a set of threshold values. A Token may only be captured for transmission of a frame of priority i if the current TRT is larger than the associated priority threshold. Furthermore, the token may no longer be used for transmitting frames of a specific class if the THT reaches the threshold of this particular class.

Almost since the first proposal, FDDI has been criticized because it does not support telephony and other isochronous services. Meanwhile, a compromise has been reached resulting in the proposal called 'FDDI-II'. FDDI-II adds a circuit-switched service to the packet service of the basic FDDI by inserting a 'Hybrid Ring Control' (HRC) between Medium Access Control and Physical Layer. HRC multiplexes data between the packet MAC and the isochronous MACs (I-MACs). Details about the allocation of isochronous bandwidth may be found in [8].

B. DQDB-Overview

DQDB subnetworks consist of two unidirectional buses and a multiplicity of nodes along the buses as shown in Fig. 2. The buses, denoted Bus A and Bus B, support communication in opposite directions allowing full duplex communication between any pair of nodes on the subnetwork.



Open Dual Bus Topology Looped Dual Bus Topology

Fig. 2 Bus Topologies for DQDB

In DQDB, the nodes consist of an access unit and the attachment to each bus via one read and one write connection each, where the writing to the bus is a logical OR of the data from upstream stations with the data from the access unit. The read connection is placed logically ahead of the write connection and allows all data to be copied from the bus unaffected by the unit's own writing. It should be noted that nodes never remove data and only alter it when permitted by the access protocol.

The need to support voice and other fixed-rate services on the MAN has led to the use of a time-slotted protocol in which the data is divided into short segments and reassembled at the destination. When choosing the segment size for DQDB, it was an important task to keep compatibility with wide-area facilities in order to facilitate interconnection of MANs and bridging to wide area networks. Therefore, the MAN working group followed the decision of CCITT for a 5 byte header and 48 byte payload field, cf. Fig. 3.

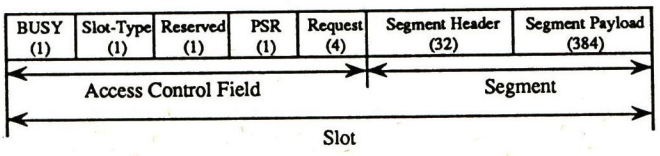


Fig. 3 Slot Format (length in bits)

Each slot consists of an Access Control Field and a segment, where a segment is nothing but a protocol data unit transferred between peer entities. The Access Control Field (ACF) is used to control the reading of a segment from a slot and the writing of segments into slots. The first bit of the ACF indicates whether the slot contains data (BUSY = 1) or the slot is available (BUSY = 0). In DQDB, there are two types of slots called 'queued-arbitrated' (Slot-Type = 0, used to transfer asynchronous segments) and 'pre-arbitrated' (Slot-Type = 1, carrying isochronous-sample segments), respectively. Access to these slots and the meaning of the request bits are discussed below. Finally, each slot contains a segment consisting of a segment header (not discussed in this paper) and a segment 'payload' of 48 bytes.

In DQDB, isochronous connection-oriented service uses assigned octet positions in particular slots for the transfer of individual octets of data. Details of pre-arbitrated slot access are not relevant for the study presented in this paper.

For bursty traffic, all nodes cooperate in sharing the network bandwidth by following the same 'distributed queue access protocol'. If the nodes were allowed to acquire any empty slot passing by, stations located closest to the slot generator could acquire all slots and prevent other nodes from transmitting. Therefore, a node requests a time slot on one of the busses ('downstream' towards the intended receiver) by setting a request bit on the reverse bus to make all 'upstream' stations know that an additional segment has been queued for access. By counting requests and unused slots, each node can determine its position in the distributed queue. Thus, each node always knows how many slots have been reserved by downstream nodes and when to access an idle slot.

More details about DQDB may be found in [9]. It should be noted that the DQDB standard defines four levels of priority. However, a study of how these levels may be used is not within the scope of this paper.

IV. Performance Analysis of FDDI

The performance of FDDI is heavily affected by the choice of the operational Target Token Rotation Time (T_OPR). This is extensively studied by Dykeman and Bux in [10]. Several interesting results of this analysis - which does not take into consideration synchronous transmissions - are discussed in the following. The rotation time of the token cannot be smaller than the ring latency T_L which is the sum of propagation delay in the fiber and station latencies. Choosing T_OPR = T_L means that no asynchronous data may be transmitted. In contrast to this, for T_OPR = ∞ each station capturing the token is allowed to transmit all the frames which are ready for transmission. Obviously, the maximum throughput may be achieved for T_OPR = ∞. In this case, a heavily loaded station may capture the token once and keep on transmitting frames without ever passing the token to its successor in the physical ring. Thus, protocol overhead due to token passing may be completely eliminated.

Of course, there may be applications requiring guaranteed response time which can only be achieved by small T_OPR. Then, we are faced with a tradeoff between small response time (i.e. guaranteed medium access within a certain time budget) and throughput.

The impacts of T_OPR and ring latency on the maximum throughput are studied in Fig. 4. Bux and Dykeman assumed 1.6 kbyte frame size, 5.085 µs/km propagation delay and 0.6 µs station latency. The number of stations is varied from 10 to 1000 and the fiber length is varied from 1 km to 200 km resulting in a ring latency from 0.011 ms to 1.62 ms. Fig. 4 illustrates that the impact of the ring latency is heavily affected by the operational Target Token Rotation Time T_OPR. This had to be expected since larger T_OPR means that more data may be transmitted during each token rotation.

It should be noted that the impact of frame size on the maximum throughput may be ignored since the transmission of MAC-overhead is included in the calculation of throughput. However, slightly better/worse results for larger/smaller frames are due to the rule that frame transmissions are not interrupted because of Token Holding Timer expiration.

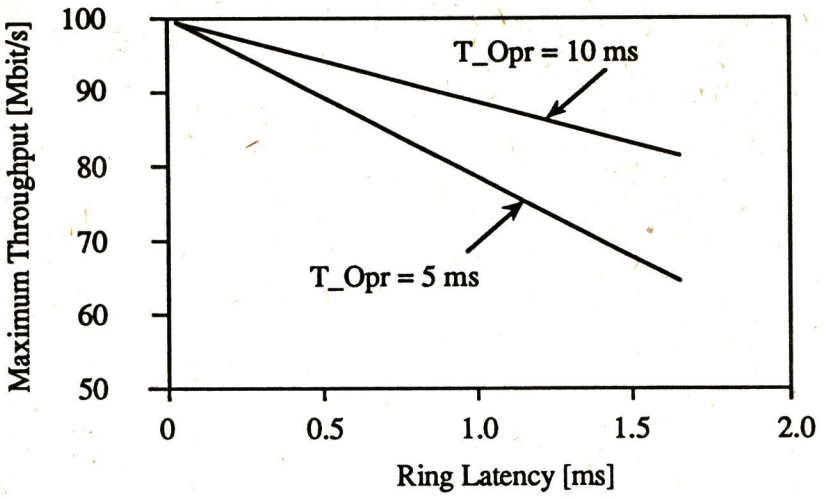


Fig. 4 Maximum Throughput vs. Ring Latency Single Cyclic Server (from [10])

Up to now we have studied the throughput assuming one priority level. In this case, each station has fair access to the common transmission channel. Results change decisively if we assume multiple priority levels. Now, there is no guaranteed throughput for low priority transmissions, i.e. transmissions associated with a smaller token holding time threshold. Instead, saturated senders belonging to the highest priority level monopolize the full channel bandwidth, i.e. tokens may never be used except for highest priority transmissions.

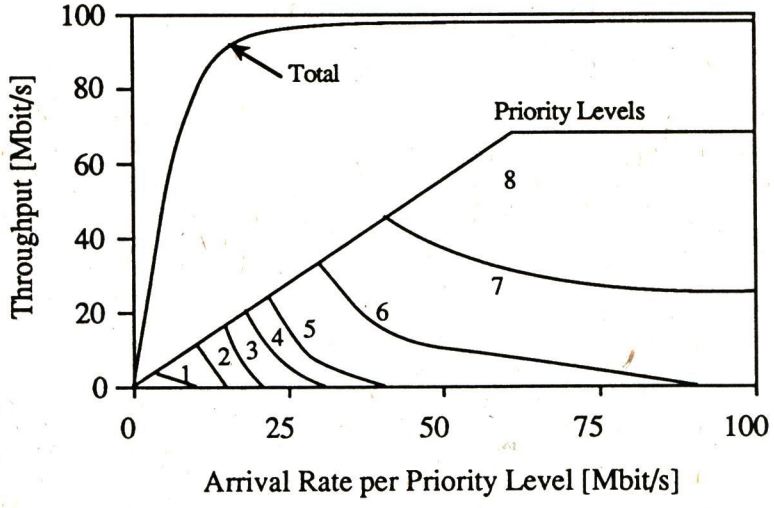


Fig. 5 Maximum Throughput vs. Arrival Rate (from [10])

The simulation results presented in Fig. 5 show the maximum throughput vs. the arrival rate per priority level. Eight different priority levels have been represented in the simulation model by eight different active stations, each trying to transmit frames at a station specific priority level. The token holding time thresholds are as follows:

Class 8: 100 ms	Class 7: 76.5 ms	Class 6: 56.2 ms	Class 5: 39.0 ms
Class 4: 25.0 ms	Class 3: 14.0 ms	Class 2: 6.2 ms	Class 1: 1.5 ms

The ring latency has been assumed as 1.0236 ms and frame size has been assumed as 1.6 kbytes. In contrast to Fig. 4 we do not assume saturated senders but exponentially distributed interarrival times with identical arrival rates of frames at each station.

The figure shows how increasing throughput of high level transmissions makes low level throughput decrease. In this example only the priority levels 7 and 8 have guaranteed throughput. Of course, throughput for a given priority level is not only determined by the token holding time threshold but also by ring latency, by the arrival rate of frames belonging to higher priority levels, and by the number of stations transmitting higher priority frames.

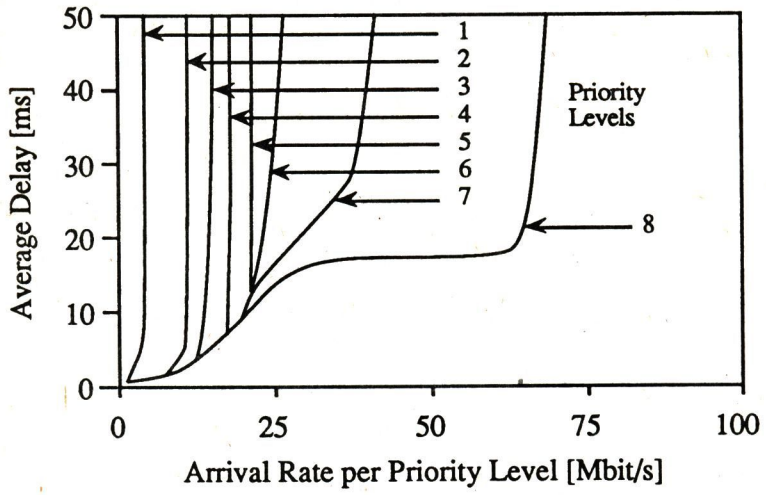


Fig. 6 Maximum Throughput vs. Arrival Rate (from [10])

Figure 6 shows results with respect to average frame transmission delays. For arrival rates between 30 Mbit/s and 60 Mbit/s the average delay for frames of priority level 8 is independent of the arrival rate! In this region frames with the largest transmission window benefit from the throughput reduction of other priority levels.

V. Performance Analysis of DQDB

Up to now, there are only few papers available dealing with the performance of DQDB. Of course, the main reason is that DQDB is a rather new protocol. In [11] Newman and Hullet presented a rather crude model of DQDB. The performance comparison included in [11] shows smaller packet delays for DQDB when compared to Token Passing or CSMA/CD. Similar results have been published in [12].

In the following, we refrain from a detailed performance comparison of FDDI and DQDB. The reason is that such a comparison strongly depends on numerous assumptions (source characteristics, data rate in DQDB, medium length of bus and ring, choice of T_OPR, number of stations, ...) which are very hard to justify. The comparison included in [13] to a large extent proves the conjecture that for light load DQDB yields smaller delays than token passing systems. The reason is as follows: For light load DQDB almost always yields immediate medium access without waiting time for a token. Furthermore, with the ring latency having a strong impact on the service characteristics of FDDI (cf. Fig. 4), DQDB has to be expected to show a better performance in communication systems covering very large distances.

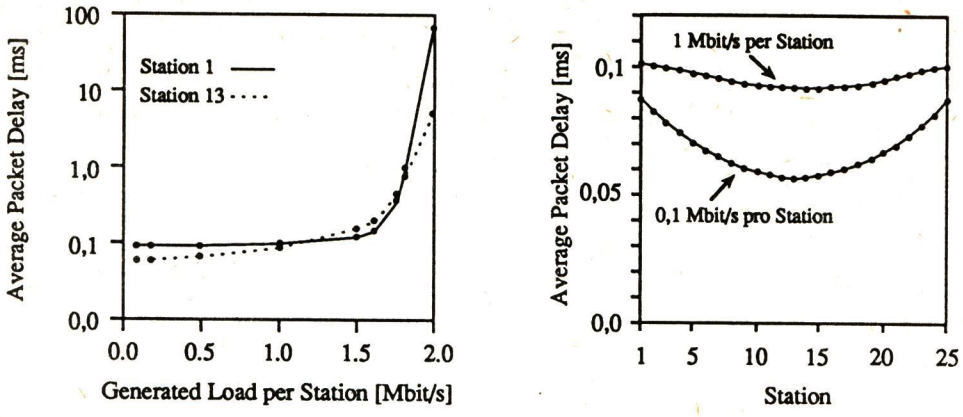


Fig. 7 Average Packet Delay for Light-to-Moderate Load

Fig. 7 shows the average packet delay (waiting time for medium access + transmission time) for queued-arbitrated traffic. We assumed 25 km medium length, 25 nodes (spaced equidistantly along the medium), 100 Mbit/s total data rate, i.e. 50 Mbit/s on each bus, and uniform distribution of destination addresses. The load generated by each station has been varied from 0.1 Mbit/s to 2 Mbit/s where we assumed a model of interactive traffic (details included in [9]) which also included packet segmentation.

Obviously, results obtained for station 13 (located in the middle) and results obtained for station 1 (located next to the slot generator, cf. Fig. 2) differ significantly (intervals of confidence have been 2 % to 3 % of the mean values). Similar results have also been published in [13] for a totally different load model and 100 km medium length.

To get a more detailed insight into the correlation of geographic location and packet delay, Fig. 7 also shows the results obtained for each station where we assumed 0.1 Mbit/s and 1 Mbit/s generated load per station, respectively. The reason for the different results presented in Fig. 7 is as follows: From a queueing theory point of view, the two buses may be modelled as independent servers. In DQDB, 'customers' desire service from a specific server determined by source and destination address. Thus, customers (segments) may have to wait for service by Bus A while the server representing Bus B is idle. For light load, the probability that both buses (independent servers) are simultaneously busy is very small. Therefore, the successor of a customer already waiting for service of bus A has a good chance to find bus B idle. Stations located in the middle benefit greatly from this effect because their load is balanced between the servers. In contrast to this, all packets generated by stations located next to the slot generator are destined to the same bus.

Of course, system performance is nothing but an answer to a dedicated load. Therefore, results would differ from the curves shown in Fig. 7 if different arrival processes would be assumed. However, for light load DQDB has to be expected to prove as a fair medium access protocol (all transfer times are very small). For this scenario, differences in transfer times may be ignored since the customer of a communication system generally does not become aware of delays in the order of milliseconds. The study included in [14] shows that this does not hold for heavy load.

In a heavily loaded communication system, some kind of congestion control procedure has to be invoked to prevent additional traffic from entering the network and/or to distribute the limited bandwidth resources fairly among the users. The ability to guarantee a given throughput/delay performance to the user is important in private networks. It is one of the essentials in public networks where users may demand a constant quality of service (regardless of the network load). A fail-safe solution to unfairness is to offer private, dedicated bandwidth. In DQDB, this is achieved by pre-arbitrated slot access. The drawback

of dedicated bandwidth is inefficiency. For that reason, DQDB is an integrated packet and circuit network, where the user can choose between queued-arbitrated service and pre-arbitrated service. In the following, pre-arbitrated service is left out of consideration. Instead, we focus on queued-arbitrate service.

In [14], we have presented an extensive study of fairness issues of DQDB. The main results of this study which assumes saturated senders are discussed in the following.

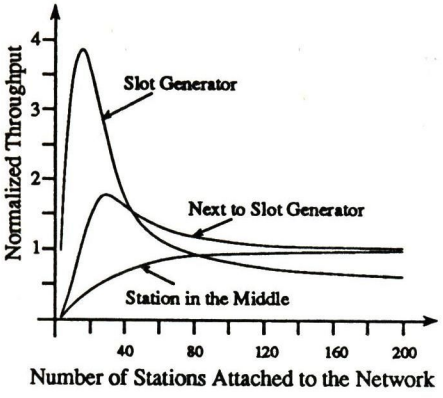


Fig. 8 Impact of the Number of Stations (100 km Bus Length)

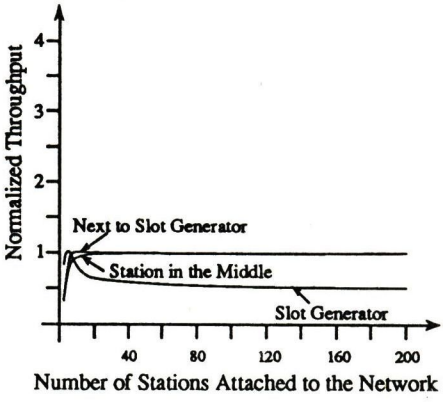


Fig. 9 Impact of the Number of Stations (10 km Bus Length)

Fig. 8 shows the throughput normalized to the throughput in the case of fair bandwidth sharing for stations spaced equidistantly on the bus. The figure compares the throughput of a station located in the middle to the throughput of the station next to the slot generator and the throughput of a station operating as slot generator. We assume a data rate of 50 Mbit/s per bus. For 100 km bus length and a small number of stations attached to the network, throughput is heavily affected by the station's geographic location. If only two stations are attached to each bus, both stations operate as slot generators. In this case, the protocol is fair since each slot generator may capture the whole bandwidth of one bus. With 3 stations attached to each bus, DQDB yields unfair bandwidth sharing. The situation gets even worse if more stations are added. The reason is as follows: Throughput of the slot generators decreases slightly with more stations attached to the network. On the other hand, stations attached far away from the slot generators do not really have a chance to capture a fair part of the capacity. Since the throughput shown in figure 8 is normalized to the throughput for fair bandwidth sharing (this throughput decreases for increasing n), the normalized throughput increases decisively.

Fig. 8 shows that the normalized throughput of the slot generator is upper bounded. This is due to the fact that waiting times increase with an increasing number of stations. This effect slows down stations enjoying preferential treatment because of shorter reservation times. Finally, it yields fair bandwidth sharing of each bus which means unfair treatment for the slot generators which transmit to one bus only. Therefore, the normalized throughput of slot generators approaches 0.5 for $n \rightarrow \infty$.

From Fig. 8, it comes clear that stations located next to slot generators achieve more throughput than stations located in the middle. The normalized throughput of other stations is in-between. These values are not shown in the figure.

With the number of stations increasing, all stations except the slot generators reach an equal share of the capacity. This is reflected in Fig. 8 by both the curve for the station located next to the slot generator and the curve for a station located in the middle getting close to 1 for an increasing number of stations. It should be noted that what seems to be an unfair treatment of slot generators is a fair treatment in the case of a looped dual bus topology where station 1 and station n are identical.

The extension of the region of unfair treatment depends heavily on the medium length. This comes clear from Fig. 9 which shows the normalized throughput vs. the number of stations for 10 km bus length. Now, DQDB has to be called fair if at least 20 stations are attached.

VI. Conclusion

In this paper, we have discussed protocols for high speed communication systems. Today, the most important protocol is FDDI. FDDI based products are already commercially available and additional vendors will follow soon. In the United States, many Federal agencies are currently planning FDDI networks. In West Germany FDDI rings have already or will soon be implemented at several universities including Aachen, Freiburg, Karlsruhe and Stuttgart.

FDDI is tailored to the requirements of private network owners (companies, universities, government agencies, ...) looking for a communication system which provides a high bandwidth interconnection among computers and peripheral equipment. In contrast to this, DQDB has been designed for public network providers. On the other hand, MAN installations may also be private: a single large company may lease private lines to connect several sites within a metropolitan area. This type of MAN is nothing but a very large LAN. However, dedicated cables will be too expensive for the majority of customers.

The (r)evolution of the standard proposed by IEEE 802.6 has shown that it is very difficult to predict the future of DQDB. Today, the dual bus proposal is supported by Telecom Australia, by AT&T and Bellcore. Prototype systems are already running in Perth and Melbourne. However, there is still a long road to a standard approved by the International Standards Organization (ISO) or by CCITT which is the organization of telephone industries. Approval will not be made more easy by the fact that DQDB shows unfair behavior for scenarios with few stations and large distances.

The fairness discussion for saturated senders has shown that DQDB is unfair in subnetworks interconnecting few stations and covering large distances. This scenario shows the limits for backbone networks interconnecting few stations scattered over large areas. For this special application, DQDB is questionable.

A way to reduce asymmetries in bandwidth sharing is to reduce the percentage of slots available for asynchronous traffic. With an increasing percentage of pre-arbitrated slots, the capability to transfer network control information remains unchanged. In contrast to this, the number of queued-arbitrated slots which are in transit between adjacent stations is reduced. In [9] we have shown that a large amount of isochronous traffic reduces unfairness due to geographical disadvantages. A similar effect is reached by making the stations not to try to operate at full network utilization. This approach which is suggested in [16] has been accepted by the DQDB working group during the meeting in Montreal by the end of September 1989.

Up to now we have restricted our discussion to protocols proposed by standardization bodies. Their activities show a trend towards increasing data rates and increasing geographic distances which is also reflected by a great variety of projects. A discussion of all these communication systems is far beyond the scope of this paper.

Future will show whether these are right who call FDDI 'the Ethernet of the future'. Whether it is FDDI or DQDB or something else, high speed communication systems for local and metropolitan areas may only reach a high number of installations if prices drop decisively and new applications requiring higher data rates than in conventional LANs become widely accepted.

References

[1] FDDI Token Ring Media Access Control, American National Standard for Information Systems, X3.139-1987

[2] FDDI Physical Layer Protocol (PHY), Draft Proposed American National Standard, X3T9/85-39, X3T9.5/83-15, Rev. 15, September 1987

[3] FDDI Physical Layer Medium Dependent (PMD), Draft Proposed American National Standard, X3T9/86-71, X3T9.5/84-48, Rev. 7.3, May 1988

[4] FDDI Station Management, American National Standard for Information Systems, X3T9/85 - X3T9.5/84-49, Aug. 1988

[5] Proposed Standard: DQDB Metropolitan Area Network, P.802.6_/D9, Unapproved Draft - Published for Comment Only, IEEE, August 7, 1989

[6] F.A. Tobagi, 'Fiber Optic Configurations for Local Area Networks', in *'Computer Networking and Performance Evaluation'*, T. Hasegawa et al., (Eds.), North-Holland 1986, pp. 163 - 172

[7] G.H. Domann, 'B-ISDN', *IEEE Journal of Lightwave Technology*, Vol. 6, No. 11, November 1988, pp. 1720 - 1727

[8] M. Teener, R. Gvozdanovic, 'FDDI-II Operation and Architectures', Proc. of the *14th Conf. on Local Comp. Networks*, IEEE Computer Society Press, 1989, pp. 49 - 61

[9] P. Davids, P. Martini, 'Performance Analysis of DQDB', to be published by IEEE in the *Proc. of the International Phoenix Conference on Computers and Communication (IPCCC-90)*, Phoenix, March 1990

[10] D. Dykeman, W. Bux., 'Analysis and Tuning of the FDDI Media Access Control Protocol', *IEEE Journal on Selected Areas in Communications*, Vol. 6, No. 6, July 1988, pp. 997 - 1010

[11] R.M. Newman, J.L. Hullet, 'Distributed Queueing: A Fast and Efficient Packet Access Protocol for QPSX', in *New Communication Services: A Challenge to Computer Technology*, P. Kühn (Ed.), North-Holland 1986, pp. 294 - 299

[12] R.M. Newman et al., 'The QPSX Man', *IEEE Communications Magazine*, Vol. 26, No. 4, pp. 20 - 28

[13] M.N. Huber et al., 'QPSX and FDDI-II Performance Study of High Speed LAN's', *Proc. of EFOC/LAN-88*, pp. 316 - 321

[14] P. Martini, 'Fairness Issues of the DQDB Protocol', *Proc. of the 14th Annual Conference On Local Computer Networks*, IEEE Computer Society Press, 1989, pp. 160 - 170

[15] P. Martini, 'Designing High Speed Controllers for High Speed Local Area Networks', *Proc. of GLOBECOM '89*, Dallas, IEEE, 1989, pp. 5.4.1. - 5.4.5.

[16] E. L. Hahne et al., 'Improving the Fairness of Distributed-Queue-Dual-Bus Networks', submitted to *IEEE INFOCOM '90*

BANDWIDTH ALLOCATION AND FLOW CONTROL IN ATM NETWORKS

Janusz Filipiak

Telecommunications Department
The University of Mining and Metallurgy
Al. Mickiewicza 30, 30-059 Kraków, Poland

The ATM network is envisaged as a bearer network accomodating mixed traffic types in which virtual circuits are established for transmission of different user entities: voice bursts, data packets, video screens, etc. In this paper a structure of network management and control is discussed. The control consists of accepting and blocking connection requests according to the thresholds worked out by the network management layer. The discrete time model of statistical multiplexing is then developed and used to determine the threshold values separately for each traffic class so as to avoid overload periods in the call and burst layers. This allows one to preserve the integrity of user entities and decrease the buffer space requirements. Flow control procedures are implemented in the cell layer to enforce the bandwidth allocated to a user.

1 VC set up, routing and flow control

The network is considered as a set of multi-point links L_m, $m = 1, \ldots M$. We shall assume that each link has a Link Controller (CTR_m). Stations attached to the link will be called the I/O stations.

To describe the connection set up and routing functions assume that the cell header includes three address fields:

1. I/O station address on the link
2. VCI = Virtual Circuit Identifier = Logical Channel Number (LC_1)
3. VPI = Virtual Path Identifier = Logical Channel Number (LC_2)

The I/O station address can be considered as a MAC layer address. The VCI and VPI are needed for cell routing. The protocol stack is shown in Table 1. The control and management functions are separated as described in [9].

Assume first that the network does not support virtual paths and that users generate VC request cells to establish virtual connections accross the network. The VC request must have at least two attributes: (a) destination address, and (b) traffic class. The VC request cells are processed by the segment controllers. Each CTR maintains a routing table and a table containing the flow control thresholds. CTR_m executes the following algorithm upon arrival of the cell requesting the VC set up for the j class call.

Step 1 The threshold $\hat{K}_{mj}$ on a number of calls in progress is checked to see if there is enough spare capacity on the link to support the requested connection. If the connection request is accepted the routing function is performed in Steps 2 to

4. In the opposite case the originating node is notified that the VC request was cancelled.

Step 2 The destination address is compared with a routing table to determine at which I/O station the VC request cell (and all subsequent cells of the call) must exit the link.

Step 3 The logical channel number LC_m is allocated to the request being valid on the link L_m

Step 4 CTR_m updates the address translation table at the I/O station where the VC request cell entered the link L_m. The address translation table associates the LC_{m-1} allocated to the VC on the previous link with the new LC_m and the address of the I/O exit station.

The procedure is repeated on all links on the way to destination.

TABLE 1

<table>
<tr><td rowspan="2">Executed in CTR</td><td>Bandwidth management</td><td rowspan="2"></td></tr>
<tr><td>VC set up</td></tr>
<tr><td rowspan="3"></td><td>Cell routing</td><td rowspan="3">Executed in the I/O station</td></tr>
<tr><td>MAC (ATM adaptation layer)</td></tr>
<tr><td>PMD layer</td></tr>
</table>

Cells of the call follow the route established by the VC request cell. They are routed according to the address translation tables. As mentioned, the address translation table defines the LC number and the address of the exit station for the next segment of the route based on the old LC number. The proposed routing technique splits the routing function between the I/O station and the Segment Controller. The time consuming and complicated routing of VC request cell is performed once per call by the higher layer procedure which resides in link controller CTR. Cell routing function is reduced to the address translation in the I/O station.

The Virtual Path is set up in a similar way as the VC. Once the VP is established the VP manager, residing in the originating node, allocates VC numbers to incoming calls. The VP manager is responsible for not overselling of the bandwidth, which was allocated to VP during the set up phase.

The above procedures are efficient provided that the buffers in I/O stations are not flooded with the cells which cannot be sent forward because of the congestion on the next segment. To counteract the congestion occurrence the threshold values $\hat{K}_{mj}$ must be determined with care. The procedures which we propose for that purpose are described in the next section.

We stress that in contrast to mono-service packet switched networks an integrated environment of ATM networks requires that the traffic class be defined in addition to the destination address during the call set up.

2 Bandwidth management

Assume that the bandwidth allocated to j class traffic on link m is c_{mj}. Denote by K_{mj} the number of j class calls connected on link m and assume that during a call an information is generated in bursts. The question is how many calls can be supported by the link using the allocated capacity c_{mj}.

The system behavior is modelled by means of a two-dimensional Markov chain (TDMC) (U, X), where U is a number of channels in a given traffic class being active in the burst layer, and X is a total number of cells waiting for transmission in the buffer.

TDMC transitions occur at epochs $\kappa\Delta t$. The number of active channels U changes at random independently of a number of waiting cells. Instead, the number of cells arriving to the segment in the time period Δt is determined by the number of active channels and equals αU, where α is equal to the length of the observation interval Δt divided by the packetization delay τ_1: $\alpha = \Delta t/\tau_1$.

Denote by L the number of cells which can be transmitted during the time interval Δt: $L = \Delta t/\vartheta_1$, where ϑ_1 is the cell transmission time. In what follows we shall assume that all traffic parameters are expressed in units of cell transmission time $\vartheta_1 = 1$.

L is treated as a size of a virtual frame. If the number αU of cell arrivals in a given frame is bigger than L, then the $\alpha U - L$ cells must wait in the IS buffers. The buffers are unloaded if in a given time interval αU is smaller than L, in which case the number of waiting cells is decreased by $L - \alpha U$.

Assume that the time scales are chosen so that the observation interval Δt is equal to the packetization delay. The number of cells arriving to the system in one frame L is then equal to U, $\alpha = 1$. In what follows we consider that situation to avoid notational complexity involved in the analysis of the general case.

Denote by $\pi_\kappa(i, x)$ the probability that the TDMC is in the state (i, x), where $i = u$ is a number of bursts in progress and x is a number of cells in the buffers, and let r_{ij} denote a probability that the number of active channels undergoes a transition from i to j during the time Δt. Disregarding the boundary conditions we have:

$$\pi_{\kappa+1}(j, x) = \sum_{i=0}^{K} r_{ij}\pi_\kappa(i, x + L - i) \tag{1}$$

where K is a number of channels active in the call layer.

Descriptions of that type were considered previously in [18] and [6]. A similar model, called the fluid approximation, was introduced in [11] and then used by O'Reilly, cf. [15] and [16]. Viterbi in [18] was able to derive a closed form expression on the mean delay in systems with $\hat{l} = 1$. That result can be applied to channels in which packets in a burst are not interspersed with silence intervals. A continuous time model similar to the one considered here was analysed in [2] and then used in [3].

To determine transition probabilities r_{ij} we assume that each source is described by specifying two parameters:

- η_2 =mean length of silence period
- ϑ_2 = mean burst duration

Variables with indexes '2' and '1' refer to burst and cell layers, respectively.

Define q and p as the probabilities of burst arrival and departure in the time interval Δt, respectively. We have $q = 1/\eta_2$ and $p = 1/\vartheta_2$. The transition probabilities r_{ij} can be calculated from the following equations:

$$r_{ij} = \sum_{k=j-i}^{\min(K-i,j)} \binom{K-i}{k} q^k (1-q)^{K-i-k} \binom{i}{k-j+i} p^{k-j+i} (1-p)^{j-k} \qquad (2a)$$

for $j \geq i$, and

$$r_{ij} = \sum_{k=i-j}^{\min(i,K-j)} \binom{i}{k} p^k (1-p)^{i-k} \binom{K-i}{k-i+j} q^{k-i+j} (1-q)^{K-k-j} \qquad (2b)$$

otherwise. The above equations were obtained by summing probabilities of all possible transitions from i to j that occur in one epoch as a result of different combinations of numbers of arrivals and departures.

The boundary conditions must be imposed on Eqs. (1). They complicate the model so that it has to be solved numerically.

Using the TDMC model we can determine the time congestion and cell blocking. The time congestion s is defined as a proportion of time that the system spends at the right boundary above L:

$$s = \sum_{i=L+1}^{K} \pi(i, N) \qquad (3)$$

The cells start to be blocked when the system enters the region $B = \{(i,x) : i = L+1 \ldots K;\ x = N-i+L+1 \ldots N\}$ of the state space close to the right boundary. Therefore, the blocking probability b is defined as follows:

$$b = \sum_{i=L+1}^{K} \sum_{x=N-i+L+1}^{N} (i - L - N + x)\pi(i,x) \qquad (4)$$

The TDMC model describes the traffic fluctuations in the burst layer. It determines the mean delay d_2 incurred by cells during overload periods in the burst layer. By the burst layer overload we mean a period of time when the number of on-going bursts is greater than the packetization delay.

The cells also suffer delays outside the burst overload periods owing to small traffic fluctuations. The latter delay will be denoted d_1 and called the cell layer delay. The cell layer delay can be determined from the M/D/1 model, cf. [8].

It is well known at present that under heavy loads the burst layer delay d_2 dominates the system performance. For small channel utilizations $r < \hat{r}$ the burst layer delay d_2 is negligible because the probability of the burst layer overload is very small. For larger loads d_2 increases rapidly resulting in long queues and high cell blocking. This is due to the fact that the overload periods in the burst layer are very long as compared to the cell transmission. The load here is defined as $r = K\varrho_2\varrho_1$, where ϱ_2 is a channel utilization by bursts during a call, $\varrho_2 = \vartheta_2/(\vartheta_2 + \eta_2)$, and ϱ_1 is a channel utilization by cells during a burst, $\varrho_1 = 1/\tau_1$.

The above result indicates that the channel load should be kept below the threshold value $\hat{r}$ to avoid the burst layer overloads. Long queues during the burst layer overloads

can be handled only with a difficulty, cf. [12]. When long queues occur a burst integrity is usually lost. To restore the burst structure complicated mechanisms need to be implemented in the cell layer decreasing an effective transmission rate. These problems can be avoided by keeping the channel utilization below the threshold value $\hat{r}$.

The threshold $\hat{r}$ depends on a traffic concentration. It can be determined using the TDMC model. For low traffic concentration the critical channel utilization $\hat{r}$ is very low. In such cases the allocated bandwidth is practically equal to a sum of peak bit rates of the multiplexed channels.

The characteristic $\hat{r}(c)$ is used by a network manager to reallocate capacities in the network and set up the call acceptance thresholds. Given the critical channel utilization $\hat{r}$, the traffic burstiness, and the packetization delay, the threshold on a number of accepted calls is determined from the formula:

$$\hat{K} = \frac{\hat{r}}{\varrho_1 \varrho_2} \tag{5}$$

There are several arguments supporting the proposed separation of the ATM network into several virtual subnetworks, one subnetwork for each traffic class. One of them is a difficulty in determining the cell transmission quality for mixed traffic types. Even if that could be done, however, the gains which can be obtained in practical systems from mixing different services seems to be marginal.

To illustrate that conjecture consider an example of voice and variable bit rate video. One VBR video channel requires the 4.5 Mbit/s bandwidth. Using 4.5 Mbit/s one can multiplex about four hundred voice channels with bandwidth utilization about .95, which means that a spare capacity needed to accomodate fluctuations of speech activity is about 200 kbit/s. One cannot expect big profits from overselling this capacity to the video traffic. Moreover, doing so one puts into a jeopardy the quality of voice transmission in all 400 channels. Similar arguments are valid for all traffic classes having different characteristics.

Assume now that there is a manager in the network NM attached to one of the CTR stations which executes the highest layer function in Table 1. The NM monitors the offered, blocked and carried traffic and reallocates the bandwidth between the services. A transmission quality in the cell and burst layers is ensured by keeping the load below $\hat{r}$ on each segment separately. Therefore, the task of the network manager consists of reallocating the bandwidth so as to maintain the low call blocking in each traffic class.

Denote by ϱ_{mj} the measured utilization of bandwidth c_{mj} allocated to j class traffic on the segment m. If $\varrho_{mj} < \hat{r}_{mj}(c_{mj})$ then c_{mj} is decreased so that for new c'_{mj} the condition $\varrho_{mj} = \hat{r}(c'_{mj})$ is fulfiled. The remaining bandwidth is reallocated to traffic types for which the call blocking is high.

3 Bandwidth enforcement

In order to fully describe the ATM traffic source the following parameters must be known:

Cell layer: ϑ_1 = cell transmission time

τ_1 = cell interarrival time during a burst (packetization delay)

Burst layer: ϑ_2 = mean burst duration

η_2 = mean length of silence period between two subsequent bursts

Call layer: ϑ_3 = mean holding time

τ_3 = mean call interarrival time

From the considerations of the previous section it follows that the cell and burst layer parameters determine the cell blocking and delay. If the actual parameters are different than the ones declared during the call connection phase then there is a risk of congestion collapse.

To explain this assume that due to statistical fluctuations or increased activity of ignorant or malicious users the overload periods occur in the burst layer. The cell blocking and delay increase in such case. Users which experience a degradation of transmission quality react by increasing an amount of information forwarded to the network. This increase may be observed in several layers. For instance, packetization delays may become shorter and bursts longer. This will further increase the congestion leading to a congestion collapse.

It is clear from the above arguments that some control mechanisms in the cell and burst layers need to be implemented at the User Network Interface, in order to police the user's activity during a call. The following algorithms can possibly be used for that purpose:

1. Adaptive Window Flow Control
2. Credit manager
3. Access Protection and Priority Control (APPC)

The APPC protocol, which applies mainly to multi-point connections, will not be discussed here. Its description can be found in [7].

3.1 Adaptive window flow control

The *sliding window flow control* is widely used at present in packet switched networks. Recently, a version of that mechanism known as the *adaptive window flow control* attracted a lot of attention, cf. [5], [13], [14]. The adaptive window flow control decreases loss from buffer overflow when an amount of buffering available in the network is limited. Assume that the end-to-end window flow control is applied. Sources update their windows using acknowledgements from cells that are successfully transmitted. If some cells are lost then the window size is reduced by the number of lost cells. If traffic carried by the network on a given path is bigger than the path capacity, cells are lost until the sum of window sizes equals the amount of buffering available plus n original window sizes, where n is the number of virtual circuits which can simulataneously be served by the path. After that the traffic is throttled at the inputs to the network and cells are not lost.

The described property of the adaptive window flow control can be used to decrease the network congestion. When the network is sligtly loaded the windows are not reduced. The window size decreases adaptively when the congestion occurs. When the overload period is over the window size is slowly restored to its original value.

Alternatively, the window size is restored only after a source has become inactive and then active again.

Using the adaptive window flow control the cell loss usually amounts to a window's worth of cells spread over all virtual circuits. The cell loss does not depend on the burst length which means that the average cell loss rate would decrease with increasing average burst length.

Although the adaptive window flow control has several desirable properties there are several problems with implementing that mechanism in ATM networks, related mainly to high transmission bit rates in such networks, a small cell size, and tight delay constraints for many services.

To illustrate a kind of problems which occur note that the signal propagation delay in a communication channel is 50 μs per kilometer, which means that the round trip delay over a distance of 1000 km is 10 msec. In order to apply an acknowledgment scheme the buffer space of the order of $10^{-2}c$ is required, where c is the transmission bit rate. Assuming that c equals 150 Mbit/s, the memory allocation or window size needs to be 1.5 Mbit. If blocks of cells are acknowledged rather than individual cells than some additional memory is also needed being equal to the block size.

The memory requirements themselves are not critical. Large semi-conductor memories are available at present at a low price. However, the delays of the order of 10 ms which occur in the case of erranous transmission are not acceptable. Moreover, in view of small cell size (48 bytes of information field) blocks of cells need to be retransmitted rather than individual cells. The adaptive window flow control mechanism with the block retransmissions does not have the nice features, which has been described above. Because all cells in the block are retransmitted even if only one cell is lost the network resources are wasted and the retransmission mechanism may trigger off the congestion period.

Another problem with the adaptive window flow control is an unfairness. Using that protocol an amount of cells lost on each path is not controlled. Some VC can incur much higher cell blocking than the others. The self regulating property of the adaptive window flow control is not very useful from that point of view.

3.2 Credit manager

The *Credit Manager*, also known as the *Leaky Bucket* algorithm, cf. [17] and [4], is currently being proposed by Bellcore as a bandwidth enforcement mechanism for the Switched Multi-megabit Data Services (SMDS), [1]. The SMDS, which is the User Network Interface protocol, provides for different *Access Classes*. Each Access Class is characterized in terms of the mean bit rate, *the Sustained Access Rate (SAR)*, and the burstiness of the information transfer characterized by the maximum burst length θ_{max}.

The mean burst interarrival time τ_2 equals $\vartheta_2 + \eta_2$. Each burst is composed on average of $\theta_2 = \vartheta_2/\tau_1$ cells. Denote by T the time interval τ_2/θ_2: $T = \tau_2/\theta_2$. Note that T is greater than the packetization delay τ_1.

The Credit Manager has the counter Q. During each time interval T the counter is increased by one, $Q = Q + 1$. Each time a cell is transmitted (sent from a user to the network) the Credit Manager decrements the counter Q by one unit $Q = Q - 1$. Cells cannot be transmitted if Q is equal to zero.

The counter content cannot exceed Q_{max}, the Maximum Unit Count. However, after reaching Q_{max}, which means that the user did not transmit any information for a period of time equal to TQ_{max}, the Credit Manager allows the user to transmit a burst of maximum $\hat{\theta}_2$ cells.

Assume that: (a) Q_{max} is chosen so that TQ_{max} is equal to the mean length of silence period η_2: $TQ_{max} = \eta_2$; (b) $\hat{\theta}_2 = \theta_2$; and (c) the peak bit rate c_{max} is not greater than θ_1/τ_1, where θ_1 is the cell length measured in bits. Then, the mean bit rate of information transfer accross the interface is not greater than the mean bit rate declared for a given Access Class.

Manipulating the Credit Manager parameters, T, Q_{max}, $\hat{\theta}_2$ as well as the peak bit rate c_{max} allows one to shape the cell and burst layer characteristics of traffic forwarded to the network.

4 Concluding remarks

The traffic management and control procedures for ATM network has been presented. The basic features of the proposed architecture comprise a separation of the transmission network into logical or virtual subnetworks (one subnetwork for each traffic class) and a separation of the decisionmaking into:

- The *call connection layer* with the call set up according to the thresholds and routing tables
- The *bandwidth management layer* determining the threshold values and entries of routing tables
- The *bandwidth enforcement layer* policing the user activity during a call

The separation of traffic management and control was previously discussed in a related paper, cf. [10]. We have indicated that the credit manager has attractive features as a bandwidth enforcement mechanism.

The quality of link operation has been determined using the two-dimensional Markov chain constructed in the burst layer. The analysis has shown that though in ATM systems cells have the same format for all traffic types, the service characteristics defined in the burst layer manifest themselves during periods of congestion and determine the system behavior.

References

[1] C. S. Albanese. Interdependencies among the SMDS Credit Manager parameters. In *3rd IEEE Workshop on Metropolitan Area Networks*, San Diego, CA, March 1989.

[2] D. Anick, D. Mitra and M.M. Sondhi. Stochastic theory of a data-handling system with multiple sources. *Bell Sys Tech J.* 61(8):1871 – 1894, October 1982.

[3] B. Maglaris, D. Anastassiou, P. Sen, G. Karlsson, and J.D. Robbins. Performance models of statistical multiplexing in packet video communications. *IEEE Trans. Commun.*, 36(7):834–843, 1988.

[4] M. W. Beckner, T. T. Lee, and S. S. Minzer. A protocol and prototype for broadband subscriber access to ISDN's. In *ISS'87 Proceedings*, Phoenix, 1987.

[5] W. Bux and D. Grillo. Flow control in local area networks of interconnected bridges. *IEEE Trans Comm*, COM-33(10):1058 – 1066, October 1985.

[6] A. Descloux. Contention probabilities in packet switching networks with strung input processes. In *Proc. ITC 12*, Torino, Italy, June 1988.

[7] J. Filipiak. Access protection for fairness in a distributed queue dual bus metropolitan area network. In *ICC'89 Proceedings*, Boston, June 1989.

[8] J. Filipiak. Accuracy of traffic modelling in fast packet switching. In *GLOBECOM'88 Proceedings*, Hollywood, Florida, 1988. Paper 49.4.

[9] J. Filipiak. M-architecture: A structural model of traffic management and control in broadband ISDN. *IEEE Communications Magazine*, 27(5), May 1989.

[10] J. Filipiak. Structured systems analysis methodology for design of an ATM network architecture. *IEEE J. SAC*, 7, 1989. *Special Issue: Telecommunications Network Design and Planning.*

[11] D.P. Gaver and J.P. Lehoczky. Channels that cooperatively service a data stream and voice messages. *IEEE Trans Comm*, COM-30(5):1153 – 1162, May 1982.

[12] San-qi Li. Why buffering: Study of information loss in packet voice systems. (to appear in IEEE Trans. Commun.).

[13] D. T. D. Luan and D. M. Lucantoni. Throughput analysis of an adaptive window based flow control subject to bandwidth management. In *International Teletraffic Congress*, Torino, June 1988.

[14] C. J. O'Neill. Congestion control in the broadband ISDN. 1988. Research Report 7926.

[15] P. O'Reilly. Performance analysis of data in burst switching. *IEEE Trans Comm*, COM-34(12):1259 – 1263, December 1986.

[16] P. O'Reilly and S. Ghani. Data performance in burst switching when the voice silence periods have a hyperexponential distributions. *IEEE Trans. Commun.*, 35(10):1109–1112, 1987.

[17] J. S. Turner. New directions in telecommunications (or which way to the Information Age). *IEEE Commun. Magazine*, 24(10), 1986.

[18] A.M. Viterbi. Approximate analysis of time-synchronous packet networks. *IEEE J. SAC*, SAC-4(6):879–890, September 1986.

PERFORMANCE EVALUATION OF TOKEN PASSING BUS LANS

Gábor PAPP and József HARANGOZÓ

MMG - AM
P.O.Box 59.
H-1300 Budapest
Hungary

Technical University
Dept. of Process Control
H-1521 Budapest
Hungary

The paper deals with the evaluation of end-to-end delay and throughput characteristics of token passing bus local area networks (ISO 8802.4 - IEEE 802.4). The investigation is based on queueing models. Models of identical and non identical stations, message and packet arrivals, priority and non-priority service classes are discussed. Validation of the analytical models are accomplished by discrete event simulation. Some numerical examples of MAP networks computed by a new program package are also presented.

1. PERFORMANCE EVALUATION OF LOCAL AREA NETWORKS

The two most important parameters that characterize the performance of a local area network are its delay and throughput. In detail:

D: delay, which is the time necessary for the transmission of a data unit, bit, frame, file, message, packet, command, etc.

S: throughput, which gives the number of data units, bits, frames, files, messages, packets, commands that are transmitted in a given period of time.

G: offered load, which gives the number of data units, bits, frames, files, messages, packets, commands that arrive to the station in a given period of time.

In general the performance of local area networks are examined on three levels [1],[2],[3],[4],[5].

1. We can exclusively examine the **medium** and its connection (layers 0-1). We do not examine this independently, we take the performance of the medium into consideration when examining the medium access control.

2. Examination of the performance of the **medium access control** (MAC). Basically the medium access control is that limits the performance of the network.
3. Finally we can examine the **end-to-end** performance (which is of greatest interest to the user), in other words, we can examine all (at most all seven) layers together. In general, these examinations use hierarchical models in which the lowest layer is the medium access control. This is the type of model that we are using.

Examination of the performance can be done analytically (using mathematical tools), with simulation (using computer programs) and via direct measurements. We performed two types of examinations: analytical and simulation.

Among the analytical methods the **queueing theory** is most frequently referred. Queueing theory is capable of describing the stationary stochastic behaviour of large systems. Our investigation was based on two-layer queueing models [3] which consist of a multiple-queue cyclic-service submodel for the token passing MAC sublayer and a closed queueing network submodel for the flow control in LLC sublayer or Transport layer. We used **discrete event** simulation to check the results of queueing theory.

2. ANALITICAL MODELS

2.1. Token Passing Bus Medium Access Control

Token Passing Bus MAC is ususally modelled by a cyclic-service queueing system. Arrivals are often described by an exponential distribution (M-type arrival process). We can talk about simple and bulk arrival. In case of bulk arrival messages arrive exponentially distributed. In this case a message consists of several packets.

Since the error rate of local area networks is extremely low, in the service model we do not take into consideration losses due to errors and noise. We assume that the token passing MAC operates normally, i.e. the logical ring does not collapse, there is no token loss or multiplication. With these assumptions a cyclic-service queueing system can be associated with a token passing MAC. The sum of packet transmission times (service times) and token passing times gives the token rotation time (cycle time). We consider the medium as the server.

Service time is a linear function of packet size. With no constraint on this, the service process is considered G-type. Thus a token passing MAC can be modelled by a M/G/1 cyclic-service queueing system [3],[6],[7],[8].

If the network consists of few stations and the offered load is low, then every station transmits all its packets in every cycle, i.e. system is CES (Cyclic Exhaustive Service). In the opposite case, i.e. when there are many stations and a large offered load, it is possible that a station can only transmit one packet per cycle, in this case system is called COS (Cyclic Ordinary Service). In summary: In case of large offered load COS, for small ones CES gives a reasonable approximation.

Let us introduce the following notations: N: number of stations in the network, C,V: capacity and velocity of the medium, M_t: total length of the medium, m: distance between the transmitter and the receiver, l: number of bits in a packet, d,w,x: total delay, waiting time and service time, t_t, t_p, t_r: transmission, propagation and token passing time, G: offered load in a station, T_{tt}, T_{tr}: transmission and receiving time of the token, L_s: mean bit delay in a station, T_g: interframe gap, L_t: number of bits in the token, b: number of pakets in a message. M,L,D,W,X,T_t,T_p,T_r,B are the mean values of m,l,d,w,x,t_t,t_p,t_r,b. c^2_l,c^2_x,c^2_r,c^2_b are the relative variances of l,x,t_r,b. j is the index of priority classes (j=6,4,2,0), i is the index of stations (i=1,2,...,N).

The following expressions hold with the above notations: $t_t=l/C$, $T_t=L/C$, $x=t_t+T_g$, $X=T_t+T_g$, $t_p=m/V$, $T_p=M/V$, $t_r=T_{tt}+t_p+T_{tr}$, $T_r=T_{tt}+T_p+T_{tr}$, $T_{tt}=L_t/C$, $T_{tr}=L_s/C$, $c_x^2=c_l^2\cdot T_t^2/X^2$. In case of equi-spaced stations the following equations hold for token passing [6]: $T_p=M_t/(3\cdot V)$, $c_r^2= M_t^2/(18\cdot T_p^2\cdot V^2)$.

The delay of the following systems in detail are (in this paper we do not deal with medium access throughput):

a) **packet arrival, without priority, identical stations** [9],[9]:

$$(1)\quad T_{CES} = \frac{N.G.X}{1-N.G.X}\cdot\frac{X.(c_x^2+1)}{2}+\frac{c_r^2.T_r}{2}+\frac{N.T_r.(1-G.X)}{2.(1-N.G.X)}+T_t+T_p,$$

$$(2)\quad T_{COS} = \frac{N.G.X}{1-N.G.(X+T_r)}\cdot\frac{X.(c_x^2+1)}{2}+\frac{c_r^2.T_r}{2}+\frac{N.T_r.(1+G.X+G.c_r^2.T_r)}{2.[1-N.G.(X+T_r)]}+T_t+T_p,$$

b) **message arrival, without priority, identical stations** [9],[8]:

$$(3)\quad T_{CES}=\frac{X}{1-N.G.X}\cdot\frac{N.G.X.(c_x^2+1)+(B.c_b^2-1)}{2}+\frac{c_r^2.T_r}{2}+\frac{N.T_r.(1-G.X)}{2.(1-N.G.X)}+T_t+T_p,$$

$$(4)\quad T_{COS}=\frac{1}{1-N.G(X+T_r)}.\frac{X.\left[N.G.X.(c_x^2+1)+(B.c_b^2-1)\right]+N.T_r.(B.c_b^2-1)}{2}+$$

$$+\frac{c_r^2.T_r}{2}+\frac{N.T_r.(1+G.X+G.c_r^2.T_r)}{2.[1-N.G.(X+T_r)]}+T_t+T_p,$$

c) packet arrival, with priority, identical stations [11],[11]:

$$(5)\quad T_{CES(j=6)}=\frac{N.[\overset{o}{\underset{j=6}{\sum}}G_j.X_j^2.(c_{xj}^2+1)]}{2.(1-N.G_{j=6}.X_{j=6})}+\frac{c_r^2.T_r}{2}+\frac{N.T_r.[1-(\overset{o}{\underset{j=6}{\sum}}G_j.X_j)]}{2.[1-N.(\overset{o}{\underset{j=6}{\sum}}G_j.X_j)]}+T_t+T_p,$$

$$(6)\quad T_{COS(j=6)}=\frac{N.[\overset{o}{\underset{j=6}{\sum}}G_j\,.\,X_j^2\,.\,(c_{xj}^2+1)]}{2.[1-N.(G_{j=6}.(X_{j=6}+T_r)]}+\frac{c_r^2\,.T_r}{2}+$$

$$+\frac{N\,.T_r\,.\,[1+\overset{c}{\underset{j=6}{\sum}}G_j\,.\,X_j+(\overset{o}{\underset{j=6}{\sum}}G_j)\,.c_r^2\,.\,T_r]}{2\,.[1\;-N\,.\;(\overset{c}{\underset{j=6}{\sum}}G_j\,.\,(X_j+T_r))]}+T_t+T_p,$$

d) packet arrival, without priority, nonidentical stations [9],[3]:

$$(7)\quad T_{CES(i)}=\frac{G_i\,.\,X_i^2\,.\,(c_{xi}^2+1)}{2\,.\,(1-G_i\,.\,X_i)}+\frac{G_i^2}{2\,.\,G_i^2\,.\,(1-G_i\,.\,X_i)}.$$

$$.\left[c_r^2.T_r.(1-\mathcal{S})+\frac{(N-1).G.X^2.(c_x^2+1)}{(1-N.\mathcal{S})}+\frac{N.T_r.(1-\mathcal{S})^2}{(1-N\,.\mathcal{S})}\right]+T_t+T_p,$$

$$(8)\quad T_{COS(i)}=\frac{1-N.\mathcal{S}-G_i.X_i}{1-N.\mathcal{S}-G_i.T_r}.\frac{1-N.\mathcal{S}}{(1-N.\mathcal{S}).N.\mathcal{S}+\overset{N}{\underset{K=1}{\sum}}(G_k.X_k)^2}.\left[\frac{N\,.\,\mathcal{S}}{2.(1-N.\mathcal{S})}.\right.$$

$$\left.\overset{N}{\underset{K=1}{\sum}}G_k.X_k^2.(1+c_{xk}^2)+\frac{N^2.\mathcal{S}.c_r^2.T_r}{2}+\frac{T_r}{2.(1-N.\;)}.\overset{N}{\underset{i=1}{\sum}}G_k\cdot X_k\cdot(1+G_k\cdot X_k)\right]+T_t+T_p,$$

where $G=\frac{1}{N}.\overset{N}{\underset{i=1}{\sum}}G_i$, $X=\frac{1}{N}.\overset{N}{\underset{i=1}{\sum}}X_i$, $\mathcal{S}=\frac{1}{N}.\overset{N}{\underset{i=1}{\sum}}\mathcal{S}_i=\frac{1}{N}.\overset{N}{\underset{i=1}{\sum}}G_iX_i$, $c_x^2=\frac{1}{N}.\overset{N}{\underset{i=1}{\sum}}c_{xi}^2$.

2.2. End-to-End Performance

Flow control is necessary for both local area and wide area networks. Generally, flow control is achieved by numbering the data units flowing through virtual channels. The information carried by numbering the data units is used for transmission control via the window mechanism.

In local area networks window mechanism based on connection oriented protocols with acknowledgement used of flow control on the LLC sublayer or transport layer. Networks with window mechanism can be described with a closed queueing network model [12], thus performance parameters can be calculated with mean value analysis [13].

Steps of end-to-end performance calculation are the followings [12],[13]:
1. We assign virtual channels of the network according to the traffic, and we specify the layers (entities) where these channels are.
2. We specify the mathematical model that can describe the layers or entities (FCFS, processors sharing, LCFS, pure time delay with phase type). We have to give the window size.
3. We solve the mathematical model with iterative mean value analysis.

Let us introduce the following notations: I: number of entities, R: number of virtual channels, V^r: window size in channel r, r = 1,2,...,R, $\underline{V}=(V^1, V^2, \ldots, V^R)$: window sizes, R(i): set of channels visiting entity i, i = 1,2,...I, Q(r): set of entities in virtual channel r. $S^r(\underline{V})$, X^r_i, $T^r_i(\underline{V})$, $N^r_i(\underline{V})$: throughput, service time, waiting time, number of packets in channel r, in entity i, if the window sizes are $\underline{V}$. We introduce $\tilde{S}^r(V')$, $\tilde{X}^r_i(V')$, $\tilde{T}^r_i(V')$, $\tilde{N}^r_i(V')$, $\epsilon^j_i(\underline{V})_r$ as follows:

$$(9)\quad \tilde{X}^r_i \approx X^r_i \cdot \Big(\sum_{j\in R(i)} N^j_i(\underline{V})\Big)/N^r_i(\underline{V}), \text{ or } \tilde{X}^r_i \approx X^r_i/\Big(1-\sum_{\substack{j\in R(i)\\ j\neq r}} S^j \cdot X^j_i\Big),$$

$$(10)\quad \tilde{T}^r_i(V') \approx \tilde{X}^r_i + \tilde{X}^r_i \cdot \tilde{N}^r_i(V'-1),$$

$$(11)\quad \tilde{S}^r(V') \approx V'/\sum_{i\in Q(r)} \tilde{T}^r_i(V'),$$

$$(12)\quad \tilde{N}^r_i(V') \approx \tilde{S}^r(V') \cdot \tilde{T}^r_i(V'),$$

$$(13)\quad \epsilon^j_i(\underline{V})_r \approx 0, \text{ if } j \neq r, \text{ or } \epsilon^r_i(\underline{V})_r \approx \tilde{N}^r_i(V') - \tilde{N}^r_i(V'-1),$$

Thus we obtain the following relations:

$$(14a)\quad T^r_i(\underline{V}) = X^r_i \text{ (pure time delay with phase type),}$$

$$(14b)\quad T^r_i(\underline{V}) = X^r_i \cdot \sum_{j\in R(i)} [N^j_i(\underline{V}) - \epsilon^j_i(\underline{V})_r] \text{ (LCFS and processor sharing),}$$

$$(14c)\quad T^r_i(\underline{V}) = X^r_i + \sum_{j\in R(i)} X^j_i \cdot [N^i_j(\underline{V}) - \epsilon^j_i(\underline{V})_r - \rho^j_i(\underline{V})] + \rho_i(\underline{V}) \cdot \sum_{j\in R(i)} \frac{S^j(\underline{V})}{S_i(\underline{V})} \cdot \frac{Var[x_i{}^j]}{2 \cdot X_i{}^j} \text{ (FCFS), where:}$$

$$\rho^r_i(\underline{V}) = X^r_i \cdot S^r(\underline{V}), \qquad \rho_i(\underline{V}) = \sum_{j\in R(i)} \rho^j_i(\underline{V}), \qquad S_i(\underline{V}) = \sum_{j\in R(i)} S^j(\underline{V}).$$

(15) $G^r(\underline{V}) = V^r / \sum_{i \in Q(r)} T^r_i(\underline{V})$,

(16) $N^r_i(\underline{V}) = S^r(\underline{V}) \cdot T^r_i(\underline{V})$.

The algorithm of iterative mean value analysis is the followings:

a) Set initial values $N^j_i(\underline{V})$, $G^r(\underline{V})$, $N_i(-1) = 0$, $r = 1,\ldots,R$, $i \in Q(r)$.

b-g) Calculate the throughput in the next iterative cycle:

b) $\tilde{X}^r_i$: (9), c) $\tilde{T}^r_i(V')$, $\tilde{S}^r(V')$, $\tilde{N}^r_i(V')$: (10),(11),(12), d) $\epsilon^j_i(\underline{V})_r$:(13),

e) $T^r_i(\underline{V})$: (14), f) $G^r(\underline{V})$: (15), g) $N^r_i(\underline{V})$: (16).

Thus we obtained channel throughputs and delays ($G^r(\underline{V})$ and $\sum_{i \in Q(r)} T^r_i(\underline{V})$).

3. SIMULATION

Because of the complexity of local area networks the analytical calculations - in many cases - are not to be done or the analytical results are inaccurate. Sometimes modelling the upper layers with queueing networks is not sufficient. In these cases simulation is the best way for examination. During the performance evaluation of local area networks simulation can often not be avoided [14],[15],[16],[17],[18],[19].

In order to create a working simulation model, we have to make some simplifying assumptions. Highly accurate simulation models can only be used in a mainframe or super computer enviroment. Due to the stochastic nature of the performance evaluation we neither chose a general purpose programming language nor a deterministic simulation language. Stochastic discrete event simulation languages are the best ones to describe stochastic processes. So we chose the GPSS (General Purpose Simulation System) for this purpose.

4. APPLICATION

4.1 Performance-analysing program package

Our performance-analysing program package is capable of computing the end-to-end performance of token passing bus LAN with any size and complexity. MAC level analysis is also possible. In case of end-to-end analysis medium access can be chosen to the lowest level, however, this is not compulsory. Thus general (wide network) analysis is also available.

The program uses the formulas of iterative mean value analysis of queueing networks and the formulas of token passing bus MAC. Apart from the most significant performance parameters (i.e. mean delay and throughput) it gives the necessary average station buffer size. With the program we can exactly show which are the stations, layers, entities that form a bottleneck in the network. We can also analysize what are the consequences if the configuration, the parameters of the network are varying. The performance analysing programs were written in Turbo Pascal.

The simulation programs were written in GPSS V. They served to check the analytical results. In order to obtain reasonable run time values simulation models use numerious simplifications and approximations. Because of the size limitation of this paper we can demonstrate the results of one sample system only.

4.2. A MAP Sample System

The sample network of Fig. 1. consists of four mainframes and eight small computers (STAT5-STAT12: cell controllers, data loggers, robot controllers, etc.). All the eight small computers have identical offered load parameters: each of them sends short status reports to the control room and to the data base manager. So we have 16 virtual channels (C1-C16). The four mainframes have the following offered load parameters:

STAT1: CAD-CAM system, sends files to one of the cell controllers (STAT6), to the data base manager and the gateway (C17-C19).
STAT2: Data base manager, provides files to the CAD-CAM system and the control room manager (C20-C21).
STAT3: Control room manager, regularly sends short control packets to the subsystems (C22-C29) and sends files to the data base manager and the gateway (C30-C31).
STAT4: Gateway, sends files to the control room and the data base manager.

The analytical model of the MAC is the following: COS, non-identical, without priority (8). This formula is capable of describing the various offered load parameters of the system. COS models the high load of the MAC

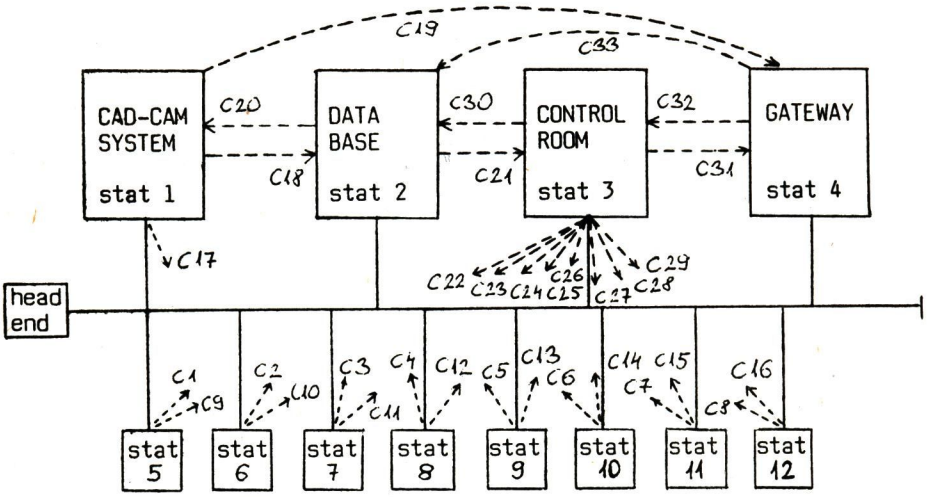


FIGURE 1

with good accurary (parameters: M_t = 2000 m, C = 10 MBPS, L_s = 5, L_t = 160, T_g = 2μs, v = 2 . 10exp(8) m/sec).

The analytical model of the upper layers is the following, (see Fig. 2.): All the seven layers are implemented. The LLC sublayer is connectionless, without acknowledgement. The network layer performs no significant tasks. Flow control is facilitated on the transport layer which is connection oriented, with acknowledgement and window flow control. Thus the LLC entity and the network entity are modeled as one pure time delay phase type node per station (X = 10 μsec). The entities of the transport layer are considered as FCFS nodes, where the service time is linearly proportional with the packet size (20 byte/msec). The entity-triplets of the three uppermost layers can be any type of nodes. Since these are not part of closed queueing network they are considered as simple elements of the total delay therefore we do not deal with these entities. Of course for the accurate calculation of the total delay, these entities should

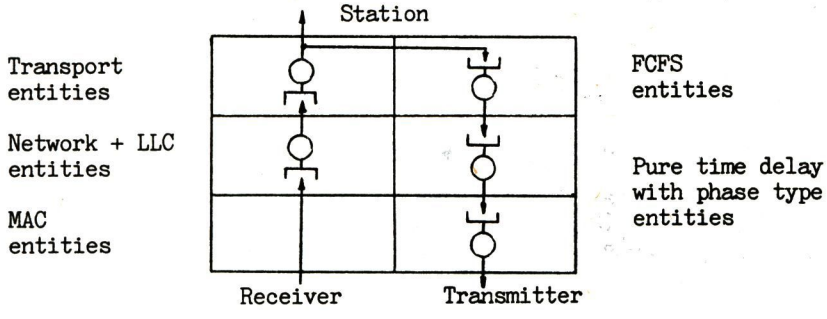


FIGURE 2

VIRTUAL CHANNELS	THROUGHPUT Analysis	Simulation	ERROR
C1,C3-C8	1.372	1.145	+20%
C2	1.366	1.140	+20%
C9,C11-C16	1.672	1.490	+12%
C10	1.663	1.454	+14%
C17	0.167	0.164	+2%
C18	0.105	0.093	+13%
C19	0.132	0.142	-7%
C20	0.093	0.109	-15%
C21	0.110	0.133	-17%
C22,C24-C29	1.675	1.609	+4%
C23	1.665	1.605	+4%
C30	0.119	0.103	+16%
C31	0.154	0.162	-5%
C32	0.141	0.152	-7%
C33	0.154	0.131	+18%

TABLE 1

also be considered. The average file size is 50 kbytes, the average size of the status- and control-packets is 500 kbytes. The deviation of the packet size is approximated with Erlang distributions for files Erlang 2-4, for status-packets Erlang 11-18). This way we can model that the length of files can very significantly while the difference among the status packets is only a few bytes. We assume explicit acknowledgement everywhere (the length of the acknowledgement is constant 50 bytes).

Table 1. compares the throughputs resulted in the analytical and simulation program. We can see that the analytical results fall within $\pm$ 20 % of the simulation results. For systems of such complexity these errors can be acceptable.

Evaluation of the results: stations STAT5-STAT12 send their status packets at a rate of 1-1.6 packets/sec, the delay towards STAT2 is 7 seconds and 0.6 seconds to STAT3. Although the load of STAT6 is greater, the performance difference between STAT6 and STAT5, STAT7-STAT12 is minimal. STAT3 sends its control packets at 1.6 packets/sec, with a 5 sec delay. This delay is too long, it is not acceptable, so the system designer must reduce it. The 0.1-0.17 packets/sec and 1-7 sec delay of file transfers between STAT1-STAT4 is acceptable. The detailed output of the programs (in this paper we omitted it) shows that the majority of the delay (80 %) comes from the delay of the LLC entities. Thus in order to increase the performance of the network, the processing rate of the LLC entitites should be increased. Increasing the capacity of the medium does not increase performance. On the other hand if we do not increase the processing rate of the LLC entities, several extra station with similar parameters can be connected additionally to the system without any reduction in the performance of the network.

SUMMARY

The first part of the paper gave a general overview of the performance evaluation of local area networks. The second part described the available analytical models that characterize the token passing bus MAC and the end-to-end performance evaluation. The third part dealt with the simulation of performance evaluation. Finally we used our program package to evaluate a MAP sample system and gave a general outline of the results.

Since the development and installation costs of process control systems are extremely high, the need for analytical and simulation performance evaluation is well founded. The models described in the paper - and therefore our program package - are capable of computing the performance parameters of a given system. Thus the program package evaluates the enviroment in which the system is usable and vice versa i.e. what is the required system configuration for a given enviroment. These models can be used to find bottlenecks of the network and to compute performance values for varying network parameters such as number of stations, offered load, service time distribution, transmission rate, propagation time, etc.

REFERENCES

[1] Tripathi, S.K., Huang, Y., Jajodia, S., Local Area Networks: Software and Related Issues, IEEE Transactions on Software Engineering, August 1987., pp. 872-879.

[2] Mitchell, L. C., Lide, D. A., End-to-End Performance Modeling of Local Area Networks, IEEE, Journal on Selected Areas in Communications, No. 6., Sept. 1986., pp. 975-985.

[3] Murata, M., Takagi, H., Two Layers Modeling for Local Area Networks, INFOCOM'87., San Francisco, Cal. 1987., pp. 132-140.

[4] Stallings,W., Local Area Networks - An Introduction, MacMillan Publ. Co.,New York, 1984.

[5] Stallings,W., Local Networks Performance, IEEE Communications Magazin, Febr. 1984., pp. 27-36.

[6] Sachs, S. R., Kau, K., Silvester, J., Token Bus Protocol Performance Analysis and Comparison with other LAN Protocols, Proc. 10th Conference of Local Computer Networks, IEEE, Minneapolis, Oct. 1985., pp. 1492-1498.

[7] Moraes,L.F.M.De,Rubin,J.,Analysis and Comparison of Message Queueing Delays in Token Rings and Token Buses Local Area Networks, IEEE Inter. Comm. Conf., Vol.1., 1984., pp.130-135.

[8] Heinula, J., Performance analysis, Modeling and Measurement in a Hierarchically Distributed Computer System, Acta Univ. Oul. C34., Oulu, Finland, 1986.

[9] Takagi, H., Analysis of Polling Systems, The MIT Press Cambridge, Massachusetts London, England, 1986.

[10] Ulug,M.E., Delay Analysis for the Multi-Priority Traffic of an Explicit Token Passing Bus, IEEE Phoenix Conference on Computers and Communications, 1987., 394-401.

[11] Papp,G., Performance Analysis of Industrial Local Area Networks, Dissertation, Technical University of Budapest, 1989.

[12] Reiser, M., A Queueing Network Analysis of Computer Communication Networks with Window Flow Control, IEEE Transactions on Communications, No.8., August 1979, pp. 1199-1209.

[13] Reiser, M., S.S. Lavenberg, Mean-Value Analysis of Closed Multichain Queueing Networks, Journal of the Association for Computing Machinery, No. 2., April 1980., pp. 313-322.

[14] Chien,J.Y., Performance Analysis of the 802.4 Token Bus Media Access Control Protocol, Proc. of National Bureau of Standards Workshop on Analytic and Simulation Modelling of IEEE 802.4 Token Bus, Apr. 1985.

[15] Loyer,B.A., Kolton,D., Performance Issues of 802.4 Token Bus LAN's, Proc. of National Bureau of Standards Workshop on Analytic and Simulation Modelling of IEEE 802.4 Token Bus, Apr. 1985.

[16] Saydam,T.,Sylvanus,F., Emulation of Token Bus Local Area Networks, Computer Networks and Simulation III., North Holland, 1986., pp.153-176.

[17] Summers,C.F., Weawer,A.C., Performance of IEEE 802.4 - The Basis of MAP, IECON '86, pp. 151-157.

[18] Jayasumana,A.P.,Jayasumana,G.G., Simulation and Performance Evaluation of 802.4 Priority Scheme, Proc. Simulation of Computer Networks, IEEE 1987., pp. 232-238.

[19] Muralidhar,K.H., Pimentel,J.R., Performability Analysis of Token Bus Protocol, INFOCOM '87, San Francisco, 1987., pp. 55-63.

PERFORMANCE MODELS OF ACCESS INTEGRATION POLICIES IN ISDN CHANNELS

O. EL DESSOUKI H. BARAKA

Electronics and Communication Department
Faculty of Engineering - Cairo University

ABSTRACT

This paper proposes a class of ISDN access integration for queued traffic. This class is based on user defined grade of service (access delay value). Three policies are presented: the Shortest Access Delay (SAD), SAD_SNOS and SAD_LNOS. Simulation is used to study the behavior of the user node under the proposed policies. A comparison is made between the proposed and the conventional policies FCFS, LNOS and SNOS. Moreover, the effect of preemption and increasing the number of servers are considered for all different policies.

1. INTRODUCTION

The information age had been categorized as a time of exploding demand for communications applications of all kinds. Integrated Services Digital Networks (ISDN) represents one of the milestones of this age. ISDN represents the evolution from the individual applications networks to the integrated communications systems. Currently, we have telephone, CATV and packet_switched data networks working in parallel, ISDN would integrate these different applications in one network. Moreover, ISDN offer the users a wide range of controlled performance options tuned according to their requirements and needs.

The subscriber loop represents the first interface between the users and the ISDN network. This represents the link from the user till the first ISDN access node. Access integration problem is concerned with the policy to integrate several and different varieties and types of services to reach the first ISDN node. This policy must attain two main goals: Integration and User_controlled grade of service.

In this paper, two classes of access integration policies are presented for all queued traffic. The first class is based on the access delay value needed : Shortest Access Delay policy (SAD). The second class considers the mixing of two parameters, the access delay value needed and the number of servers required by the service. Two policies are proposed based on these parameters: the shortest access delay_largest number of servers first (SAD_LNOS) and the shortest access delay_shortest number of

servers first (SAD_SNOS). Due to the complexity of these models, simulation is used to study the behavior of these policies under different load characteristics. Two performance parameters are considered: the global average delay of all types and the waiting time for each type of service under different policies. The first parameter gives an indication of channel utilization while the second relates to user satisfaction. A comparison is made between these policies and the conventional FCFS, LNOS, and SNOS policies. The effect of preemption is also considered for all different policies.

The paper is organized as follows: section 2 gives a survey of the access integration models with a formal definition for the complexity of the problem. In section 3, the different classes of access integration policies proposed are presented. The simulation model used and performance evaluation parameters for the access integration problem are discussed in details in section 4. In section 5, simulation results for the proposed policies and comparitive results with other access policies are presented. Section 6 gives conclusions and future extensions.

2. FORMAL MODEL AND PROBLEM COMPLEXITY

A central issue in the design, modeling and analysis of access integration policies is that of performance evaluation. Three general approaches towards the analysis of these problems can be identified as follows:

* Measurements
* Analytic
* Simulation Techniques

Measuremets provide the most direct means of access policies performance evaluation but it is the most expensive one as the environment must first exist before measurements can be taken. Mathematical analysis requires two main folds. First, considerable effort and skill is required to develop a performance model which accurately reflects the system under study. Secondly, the solution of this model, which is sometimes quite difficult. Thus, the modeler starts to simplify the considered model in order to get an analytic solution for the problem.

Access integration policies have been considered in the literature for all_queued traffic, blocked and a mixing of blocked and queued traffic [1],[2]. The policies considered in this paper are based on a class of queueing systems in which customers require service by more than one server, a customer cannot enter service until all required servers are available. A major contribution to the study of these queues is the work of Brill and Green [3]. Using a system point approach, they studied a multi_server queue in which customers require a random number of servers which are to begin and end service concurrently. Customers enter service in their order of arrival (FCFS). Based on system point theory, a general framework for deriving the waiting time distribution for each type of service is presented. Explicit solutions were derived for the two_server system with $\mu i = \mu$. Kraimeche and Schwartz [4] studied a model of a broadband channel carrying a mixture of narrow- and wide-band traffics. In [5] they extended the work of Green using generating functions method to present other access policies as LNOS and sorting policy. They solved the problem for two_servers case with two types of

customers. Serres and Mason [6] studied a multiserver queue with two types of customers (Narrow band blocked service and wide band waiting in an infinite waiting buffer), with a cut_off parameter for the wide_band customers in service.

In [7], the following problem has been proven to be NP complete: Given a set of m servers for n customers requiring a random number of servers b1,b2,.., bn, with $\sum$ bi ≤ 2m, and unit execution time, does there exist a sequence S with time limit w≤2 ? This statement is called the Variable Bandwidth Requirement (VBR) problem. The VBR problem was proven to be NP complete. For the general problem, the number of equations resulting for an N_dimensional markov space is equal to:

$$\prod_{i=1}^{n} Ni + \sum_{i=1}^{n} Ni$$

where Ni is the maximum number of customers of type i that may exist simultaneoulsy in service.

$$Ni = M / bi$$

where M is the number of channels, n is the number of priority queues i.e. the different types of services existing at the node, bi represents the required serves for type i customer. Even for the simple case of 3 types of services Type_1 requiring 1 server , type_2 requiring 2 servers and type_3 requiring 3 servers and a total number of servers equal to 24 servers, the resulting number of equations is equal to 2348 equations. It is therefore, clear for all practical cases that the problem is mathematically intractable. In this study access integration policies for practical cases are deeply investigated using simulation.

With simulation techniques, the user node can be modeled to any arbitrary level of details, and under different access policies schemes. The effect of the different parameters involved in the problem can be taken into consideration to reach a realistic model. This implies more computational time for the simulation, while for the analytic model, it makes the model mathematically intractable.

In the next section, different policies are described in greater detail in order to identify the proper simulation model parameters.

3. ACCESS INTEGRATION POLICIES

Services offered by ISDN differ in their nature, and their parameters. Access Integration policy adopted at an ISDN user node must integrate these varieties of services and achieve a fair compromise between the grade of service required by the user and the network utilization.

In this paper, the grade of service requested by the user is identified using two parameters: The bandwidth and the permissible access delay value. The bandwidth of a service is the number of channels required by the service for transmission. For Narrow band ISDN the number of channels available, according to CCITT recommendations, is equal to

2B+D for the subscriber loop and 23B+D or 30B+D for the network itself. A service cannot be transmitted unless the necessary bandwidth or number of channels exist simultaneously. The permissible access delay value is the delay a request encounter from a user node to the first ISDN access node. This defines the grade of service required by the user.

In this paper, three different classes of access integration policies are presented. These classes are based on the two parameters defined. First access policies based on the bandwidth required by the services. This class considers the network resources utilization in terms of channels availability and usage. These policies are called Number Of Servers access integration techniques (NOS_based). Two policies of this class are considered here namely SNOS and LNOS policies.. Secondly, policies based on the access delay value, that consider the grade of service defined by the user. These policies are called Access Delay access integration techniques (AD_based). The Shortest Access Delay policy is presented as a policy of this class. The last class of policies provides the compromise between the grade of service and the utilization. These policies are based on both parameters and they are called AD_NOS based policies. Two policies are considered namely the SAD_SNOS and SAD_LNOS policies.

In queueing terms, the above traffic system can be modeled as a multi_priority multi_server queueing system with M identical parallel servers FIG. 1 where customers (requests) are served to completion by a random (variable) number of servers. Upon completion a customer releases all his servers simultaneously. A customer i who requires bi servers arrives in a Poisson stream with rate λi, and its service time is exponentially distributed with mean $1/\mu i$ and requires di permissible access delay value.

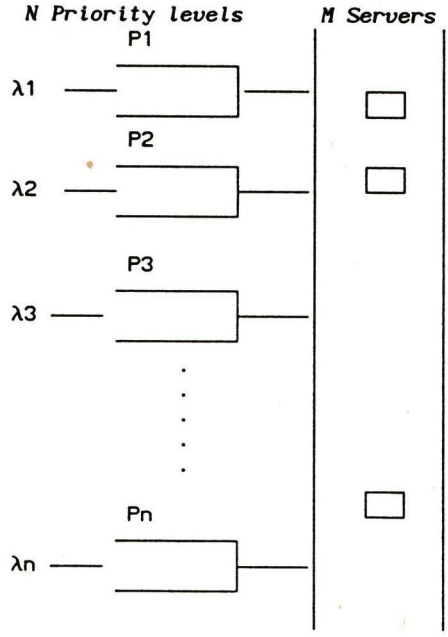


Fig. 1 Multi_Priority Multi_server Queueing Model

Each priority level Pi represents a type of requests that requires Ni number of servers, a permissible delay value di, service time $1/\mu i$ and an arrival rate λi where $Pi-1 > Pi > Pi+1$. The priority level is determined according to the parameter value or parameters values considered for each class of policies. For example, in SAD policy higher priority is given to lower access delay values. On the other hand, in SAD_SNOS and SAD_LNOS policies the priority level for a request is determined by the the number of servers required and the access delay value it may tolerate.

4. SIMULATION MODEL

The developed simulator consists of a number of major parts. The first part defines a number of global system parameters that are set by the user . This part uses a number of libraries allowing the system to be open to grow continuosly to provide more useful results with time. These libraries represent the different parameters that may affect the performance of the channel access. The first library is called the channels library. It contains the different types of channels that may exist in an ISDN network. D, B, H and E channels are defined with the corresponding specifications like bit rates. A number of extra slots are left free for future use. The second is the services types library. It contains different types of services like digital telephony, video teleconferencing, data transfer, etc...Each service type is defined by the number of channels required and the type of channels needed (bandwidth requirements). The nature of the service as either blocked or queued represents the type of the traffic. The grade of service required by the user is defined by the permissible access delay value. The third is the access integration policies library. It defines the policy of access integration. A unified structure for access integration policies has been developed for the simulator. An access policy is defined by the number and names of service attributes that are used to determine the service priority. Thus, an SNOS, LNOS, or SAD policy uses one service attribute (index) to determine the service priority. On the other hand, a SAD_SNOS policy uses two indices and so on. In this way, the user can define and add new policies to the system. The second part of the simulator deals with load generation. The user determines the number and type of channels, services and policies. Then, arrival rates are set, services requests are randomly generated from a poisson distribution with average arrival rate λi and mean service time $1/\mu i$ where $1 \leq i \leq k$.

The third part of the simulator is the main body in which the defined system is executed using an event driven technique. An arrival request arrival enters to service directly if and only if:

1- Enough free servers exist simultaneously

AND

2- No queued service request exists having higher priority than the arriving request.

Queued requests are served according to the priority level generated with the request if there are enough free servers existing.

The last part is the reports and output generation module. In order to measure the performance of the selected policy two key performance characteristics are calculated: the first is the average delay value of

all types of services as function of the load for all access policies. The second is the access delay variation among types of services for a certain policy. The average delay value represents the average delay encountered by a specific type of service under a certain policy, while the second parameter indicates the global delay encountered by all the types of services under a specified policy. Two access integration policies may give the same average access delay values, but they differ in the access delay between different types of services under the same policy. In order to choose an access policy, not only the average access delay must be low but also the variation in the access delay value for different types of services must not very large. This would makes the overall response time of the system under the selected policy acceptable for different types of services.

5. SIMULATION RESULTS

The main objective of this paper is to investigate and study the effect of two essential parameters that affect the access integration problem. These parameters are the number of servers and the access delay value. Access policies considered are based on these parameters individually (SAD, SNOS, LNOS) or on a mixing between them as SAD_SNOS and SAD_LNOS. A comparison is made between priority based policies and the FCFS policy. The effect of preemption is also considered and the six policies under the same load and generated arrivals are simulated for non_preemptive and preemptive case. Moreover, these policies are considered at different servers availability to determine the effect of increasing the available resources on the policies adopted.

Study Case:

The study case considers four different types of services that differ in the number of servers required and the delay value needed. Type_1 and type_2 require 2 servers each while their access delay requirements are 2T and T respectively. Type_3 and type_4 require 1 server with access delay values 2T and T respectively. Table 1 summarizes the input parameter values for the case under simulation.

Table 1: CASE STUDY

Types	Number of servers	Access Delay
Type_1	1	2T
Type_2	1	T
Type_3	2	2T
Type_4	2	T

Figure 2 represents the encountered access delay value of the different types of services (type 1 to 4) for each of the six policies without preemption allowed and under different loads. Figure 3 gives the encountered access delay value for the same types of services with the difference that preemption is allowed. Figures 4 and 5 plot the global average access delay value across all types of services under ceratin access policy for different loads. The results of the simulated model for the study case considered can be sumarized as follow:

First: Performance of Access Policies based on a single parameter.

The FCFS policy presents the best result of all the policies as the maximum encountered delay value for all the types is approximately equal to 0.54T [Type_4]. The FCFS policy offers a fair compromise between the differnt types as the minimum delay value for all the types is equal to 0.5T. [Type_2].

The SAD policy deteriorates the response of the system as it represents a constraint over the overall performance. It gives better performance for Type_2 and Type_4 having minimum access delay value and on the other hand the two other types are sharply deteriorated Type_1 performance in the SAD policy is below the corresponding Type_1 in the FCFS policy. In otherwords, the SAD policy deteriorates Type_3 {Max. access delay, 2 servers} while the three other types performance is better.

In the LNOS policy the two types of servers Type_1 and Type_2 are deteriorated on the expense of better performance for Type_3 and Type_4 while the reverse happens for the SNOS policy. The performance of LNOS is lower than SNOS. Moreover, the maximum delay for SNOS policy is lower than the corresponding maximum delay for LNOS policy.

Second: Performance of access policies based on two parameters

The SAD_SNOS policy gives better response time for Type_1 on the expense of Type_3 that is severly deteriorated. This is also the same result for SAD_LNOS policy. But the delay value for Type_1, Type_2 and Type_4 is almost zero for SAD_SNOS, while it is equal to 0.05T for SAD_LNOS. These results differ from the simple SAD policy in the sense that a very low response time is given to the three types having shorter access delays, while the service type of larger access delay is totally deteriorated. This means that access policies based on the shortest number of servers give better results than those based on the largest number of servers. Moreover, it results in better performance over the single parameter SAD policy especially in the case of SAD_SNOS policy.

Third: The effect of Preemption

As seen from figures, the preemption assures the policy adopted. Even, if a service with low priority enters service, a higher priority will preempt this service upon arrival. So, if certain anomalies would occur due to the randomness of arrivals, these anomalies are removed using preemption.

Fourth: Delay Variation for different types of services

It is clear from the figures that the delay variation for all the types under the FCFS policy is relatively small (max. delay=0.54T and min. delay = 0.5T). For all the other policies , this delay variation increases as the policies favor certain types on others so the delay variation between different types increase. For example in the SNOS policy the delay for Type_1 and 2 is almost zero , while for Type_3 and 4 is 1.5T sec.

Fifth: Average Response time for the system

This parameter reflects the overall response time of the system.

From the results, it appears that the average response time for all the policies is almost the same for preemptive and non_preemptive cases except for the SNOS policy. Thus, although the maximum average delay for a type of service under LNOS and SNOS policies were nearly equal, the SNOS overall response is clearly much better than the LNOS policy.

Finally, the same case study was considered with an increase in the number of channels (6 servers and 8 servers). According to the simulation results obtained we may conclude that the previous results of the case of four servers are the same for the case of six (or 8) servers except that the delay is totally reduced for all types. However, the ordering and the behavior of the policies did not change.

6. CONCLUSION

Access integration techniques for ISDN networks need special modelling and performance evaluation. Due to the multiplicity of parameters involved in the access integration problem and the complexity of the queueing models resulting,simulation techniques are used in this paper in order to evaluate the performance of the ISDN channels under different access policies. A main objective for the paper is to introduce access policies based on a certain grade of service defined by the user. In this paper, shortest access delay policy is proposed to fullfill the user grade of service requirements. A fair compromise between the channel utilization and the grade of service defined is accomplished using multi_criteria based policies. Two policies are proposed namely SAD_SNOS and SAD_LNOS.

Simulation results show that sorting requests on ascending order of number of servers gives better channel utilization than LNOS policies. SAD policies deteriorates the response times for large access delay users, while types of services with low access delay values have much better response time. SAD_SNOS and SAD_LNOS policies improves more the performance of low access delay types in addition to the lowest number of servers types. Preemption assures the policy adopted while removing the anomalies that may exist. Increasing the number of servers decreases the delay encountered by the different types while pertaining the same results for the different policies.

REFERENCES

[1] L.A. Gimpelson, Analysis of mixtures of wide- and narrow-band traffic, IEEE Trans. Commun. Technol., pp.258-266, Sept. 1965.

[2] L.Green, Comparing operating characteristics of queues in which customers require a random number of servers, Management Sci., vol. 27, no. 1, pp. 65-74, Jan. 1981.

[3] P.H. Brill and L.Green, Queues in which customers receive simultaneous service from a random number of servers: A system point approach, Management Sci., vol. 30, no.1 , pp. 51-68, Jan. 1984.

[4] B. Kraimeche amd M. Schwartz, Bandwidth allocation strategies in wide-band integrated networks, IEEE J. Select. Areas Commun., pp. 869-878, Sept. 1986.

[5] ————, Analysis of traffic access control strategies in

integrated service networks, IEEE Trans. Commun., vol. Com-33, pp. 1085-1093, Oct. 1985.
[6] Yves Desserres and L. Mason, A Multiserver Queue with Narrow and Wide_band restricted Access, IEEE Trans. COMM., June 1988.
[7] H. Baraka, New Access Control Management Techniques of ISDN networks, Ph.D. Dissertation, Cairo University 1989.

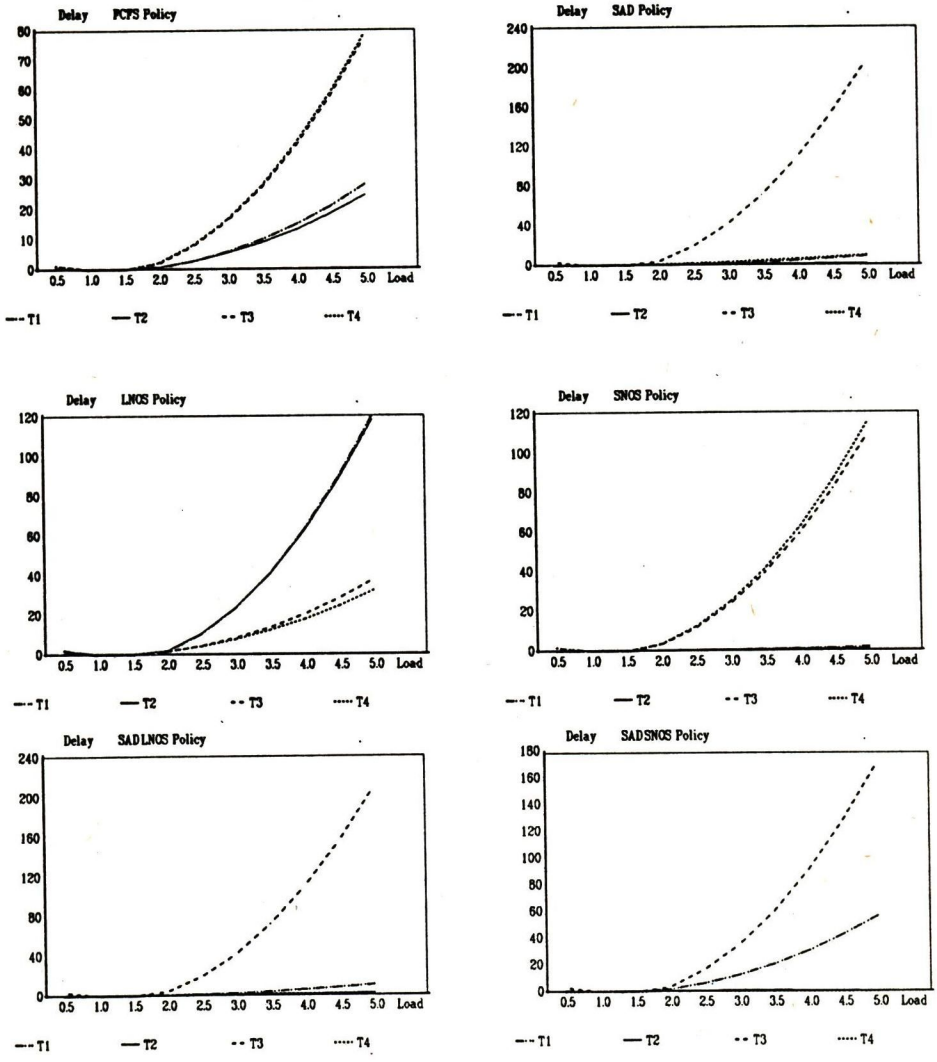


Fig. 2 Waiting Time for Different Policies Without Preemption

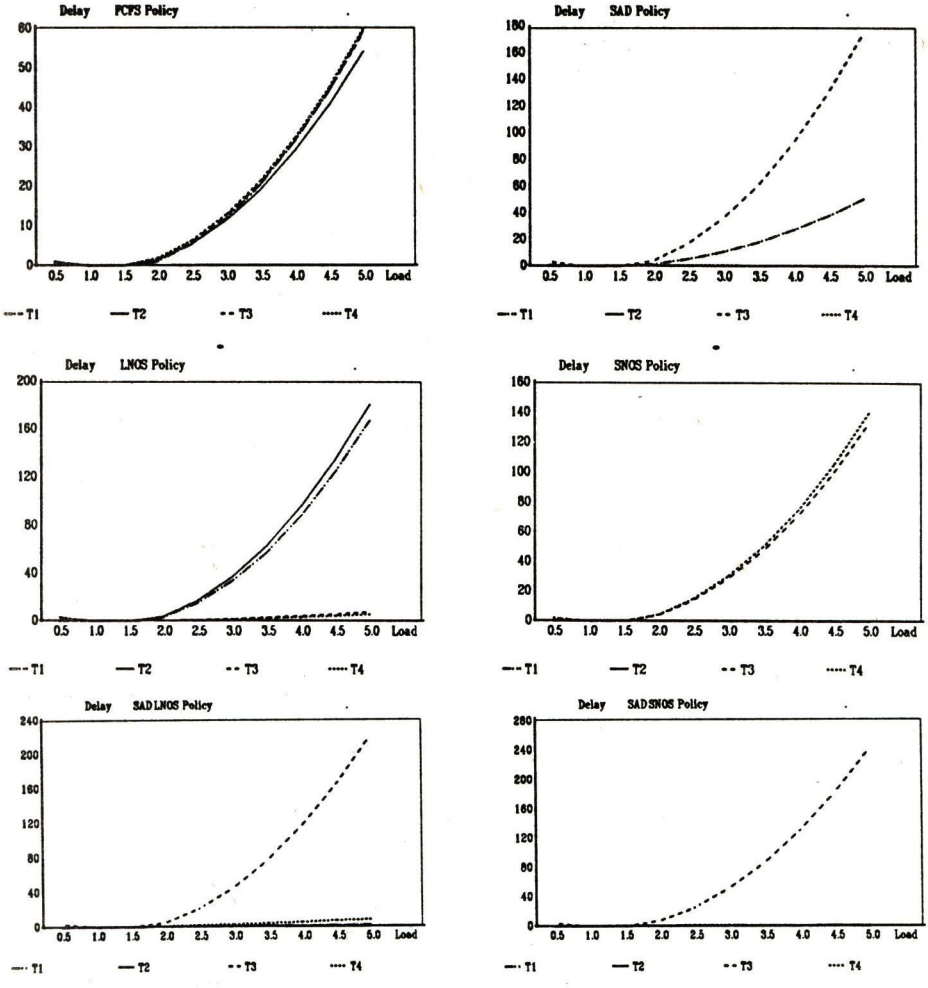


Fig. 3 Waiting Time for Different Policies With Preemption

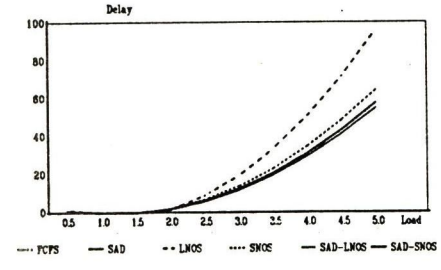


Fig. 4 Global Average Delay Without Preemption

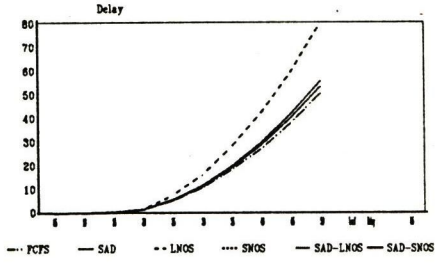


Fig. 5 Global Average Delay With Preemption

A SURVEY OF FAULT - TOLERANT LANS

Wojciech Molisz

Technical University of Gdańsk
Institute of Telecommunications
Majakowskiego 11/12
P1-80-952 Gdańsk, POLAND

The paper surveys typical fault-tolerant Local Area Networks. The review of network structures is followed by description of suitable fault location techniques. Then network reconfiguration methods are reported. Finally remarks on fault recovery are given.

1. INTRODUCTION

The increasing demand for the fault-tolerant LANs, capable of continued communication despite media failures and device outages can be observed. Perhaps the strongest requirements of fault-tolerance arise in the context of computer integrated manufacturing systems (CIMS). The network of a CIMS should be highly fault-tolerant in order to avoid the unacceptably high cost of the lost production. Certain CIMS operate in very special environment (e.g. in coal mines - a case in Poland), where the probability of unexpected damage of network components is high. There is additional motivation for fault-tolerance in such systems. Many other LAN applications, stimulating fault-tolerance can be reported.

Recent ISO proposition of enhancement of the 8802.5 Token Ring for multi - ring networks [6], as well as the new 16 Mbit/s IBM Token Ring , Version II [16] reflect the growing importance of fault-tolerance in network design.

Fault tolerance implies, that the whole distributed system (a network):
a) has high reliability
b) is able to detect the occurrence of faults
c) has mechanisms permitting to locate faults and their extent (range)
d) is able to isolate faults by means of network reconfiguration
e) has mechanisms providing fault recovery.
A degraded recovery (to a fault-free state, but with a reduced quality of service) is allowed.

Local area networks usually have bus, ring, or star (i.e. nonmeshed) structures. Fault-tolerance requires various forms of hardware and/or software redundancy .

In particular redundancy of network topologies should be ensured. The network structure should provide at least two independent communication paths for each pair of communicating entities (processes). Fault-diagnosis capabilities may also be directly related to the network connectivity.

The paper is organized into two main sections. Section 2 gives a survey of fault-tolerant LAN configurations. In section 3 we describe basic processes, needed to provide operational capability of these networks after a failure. Topics considered include: fault location techniques and fault isolation methods (by means of network reconfiguration). Finally some remarks on fault recovery are given.

2. A SURVEY OF FAULT - TOLERANT LAN CONFIGURATIONS

The basic LAN configurations are: a) bus, b) ring, c) star. Fault-tolerant LANs also have these structures, but with the media replication. The mixed topologies also exist. In this section we give a survey of typical fault-tolerant network structures.

2.1. Multibus, and other bus - type networks

Perhaps the first multibus configuration appeared in a 50 Mbit/s HYPERchannel network, offered by Network Systems Corp. since 1979 [4]. In HYPERchannel network, high speed, heterogeneous devices (e.g. mainframes or device controllers) are interconnected via a single or multiple (up to four) bidirectional 50 Mbit/s passive coaxial cable trunks. Bus interface units are called adapters. The HYPERchannel configuration is shown in Fig. 1.(The Loosely Coupled Network of Control Data Corp. has a similar configuration [5]).

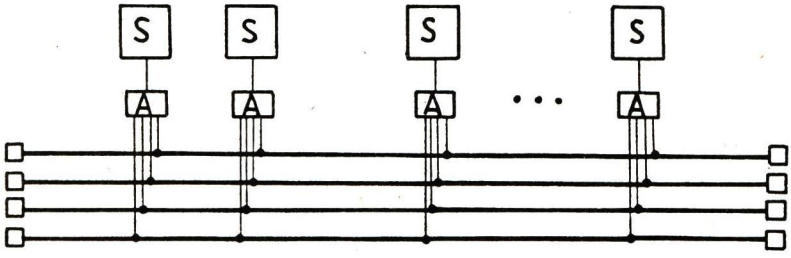


Fig. 1. HYPERchannel: a multibus HSLN configuration

Access protocol is based on a CSMA scheme with prioritized staggered delays. It allows the station with the highest priority to transmit as much as required, the second highest

station - similarily, but from the remaining capacity and so on. Access conflicts are solved through retransmissions without additional collisions.
The HYPERchannel configuration:
a) eliminates the single point of failure (if one trunk fails, others are available for transmission):
b) increases capacity (multiple simultaneous transmissions may be accomodated).

Fault location techniques and multibus network reconfiguration are described in section 3.

The fiber optic LAN, called: "SURVNET" , developed by the MITRE Corp. , has a completely different, fault-tolerant and survivable configuration (fig. 2) [7].

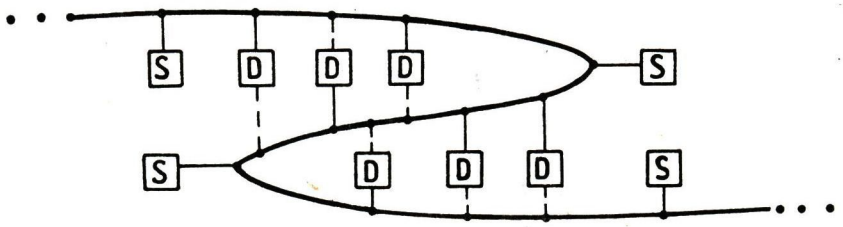


Fig. 2. SURVNET: a survivable bus LAN

Fault-tolerance and survivability of the SURVNET is achieved by the operation of its doubly connected nodes (D in fig. 2). Each D - node connects to the bus through its primary connection (PC) and secondary connection (SC) (continuous or dashed line in fig. 2, respectively). During normal operation the D-nodes use the PCs. After detection of a fault, both PC and SC are used (details in section 3).

SURVNET adapted the IEEE Standard 802.4 token passing bus access method protocol, with fault detection and recovery sections appended.

2.2. Multi - ring networks

An established way of enhancing the fault-tolerance of ring LANs is to design regular multi-connected loop networks, providing several alternate communication paths for each pair of stations. Various configurations have been proposed in the past.

The first class of networks of this kind is refered to as forward-loop-backward-hop (FLBH) [13]. An example of the FLBH network is shown in fig. 3a. Each station (node) in the FLBH net has a forward link connecting to its neighbour, and a backward link connecting to a node at some distance h (the

skip distance). We obtain different topologies by choosing different values of h. With h=2, FLBH becomes a "daisy chain". If h=1 - we have a standard dual ring network. The last structure is implemented in several networks, including the latest 16 Mbit/s IBM Token Ring, Version II [16].

Further performance and reliability improvement can be obtained in a 3-connected chordal ring class of networks. The chordal ring is a bidirectional ring network, in which each node has an additional bidirectional link (a chord) to some other node across the network (see fig. 3.b.).

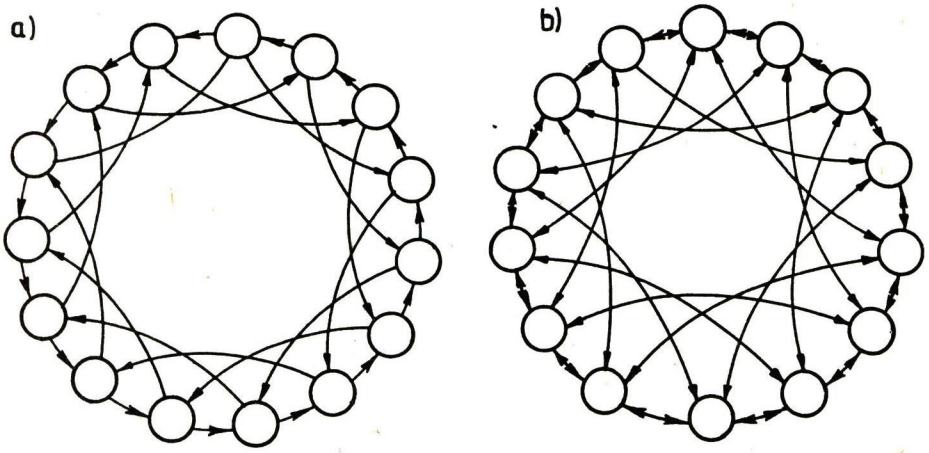


Fig. 3. Examples of multi - ring configurations:
a) FLBH networks, b) chordal ring networks

The price we pay for fault-tolerance and performance improvement in a chordal ring is the increased complexity of node interfaces and routing decisions.

Fault location and network reconfiguration methods in multi-ring networks are discussed in section 3.

2.3. Star and star - shaped ring networks

Another approach to improve reliability of a single ring network (compared to that of p. 2.2) is to bring all the interfaces to a central point. This results in a star - shaped ring with a wire center which allows easy maintenance and reconfiguration (fig. 4a).

A star - shaped ring topology was designed and implemented at IBM Zurich Research Laboratory [2]. A similar configuration was also implemented at the Massachussetts Institute of Technology [14].

Remarks on fault location and network reconfiguration are

given in the next section.
A "pure star" has a number of disadvantges. All ports must communicate through a central node, which failure is catastrophic. Redundancy must be provided.

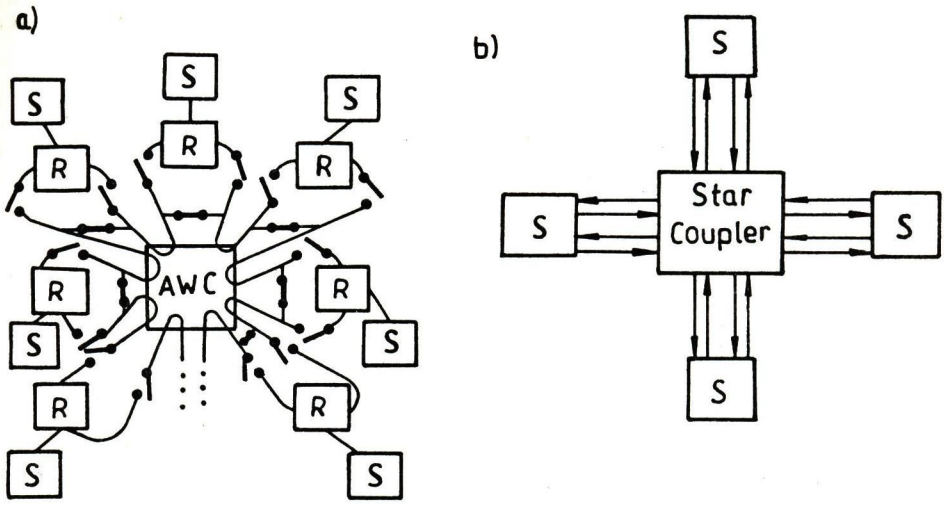


Fig. 4. Star configurations: a) a star - shaped ring topology; b) the CI topology

The reliable star configuration (the "CI" topology) is implemented in loosely coupled multicomputer systems of Digital Equipment Corp. (VAX Clusters). DEC is also proposing the "CI" as an LDDI (Local Distributed Data Interface), being standarized by the ANSI [15]. The "CI" (Computer Interconnect) (fig. 4b) is a dual path serial interconnect, with each path suporting a 70 Mbit/s data rate. Each path is implemented in two coaxial cables (one for transmission and one for reception). The star coupler is a passive device supporting up to 16 nodes.

3. FAULT LOCATON AND FAULT ISOLATION TECHNIQUES

In this section we focus on typical fault location and fault isolation techniques suitable for networks, described in section 2. We do not consider all of the cases.

3.1. Multibus networks and SURVNET

Since in practice most multiple bus systems use two buses (e.g. 85% of HYPERchannel installations [4]), we consider here a double bus, token passing example. During normal operation (i.e., no faults in the system) a logical loop and a monitoring station is established in each of the buses.

If a fault occurs in one bus segment it is detected by a monitoring station through:
a) absence of signals in the medium,
b) absence of a token for a period longer than the "token rotation time".
Then a monitoring station orders all of the stations to send status informations through the remaining, operational medium. Based on these informations, network reconfiguration decisions are prepared.

If both media fail, than each of the monitoring stations establishes a series of forward diagnostic loops to locate faults and their extent. These procedures are continued until loops which are not operational occur.Examples of first diagnostic loops are shown in fig. 5.

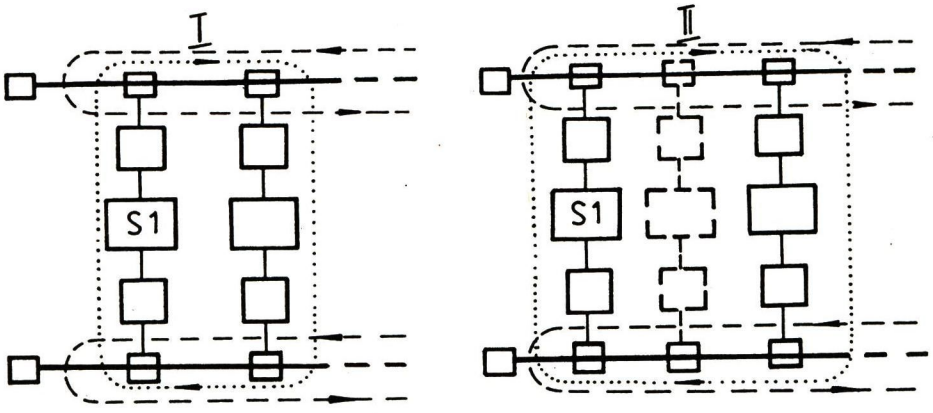


Fig. 5. First diagnostic loops (dotted lines) in a double bus , token passing network.

Based on data from the last operational diagnostic structures, conclusions on expected fault locations, and reconfiguration decisions are prepared [10], [12]. After reconfiguration a double bus network can operate in a double bus mode (certain stations have access to one medium only), or in a single bus mode. Also more then two virtual loops (of tokens) can be established in separated subnetworks of noncommunicating media segments.

Let us consider the SURVNET now. Each D-node examines periodically the bus segment between its two connection points for continuity. A test message is sent on PC and received on SC. If a fault is detected in this bus segment a D-node activates a bridging mode of operation, using both: PC and SC. Signals received by a D-node are repeated and broadcasted in both directions. The structure of SURVNET after reconfiguration is shown in fig. 6 (arrows indicate fault locations, while parallel continuous and dashed lines

are incident to D-nodes , being in a bridging mode).

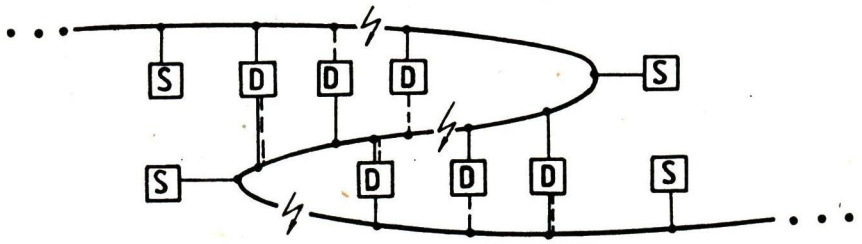


Fig. 6. SURVNET after reconfiguration

3.2. Multi - ring and star - shaped ring networks

Similar arguments as in p. 3.1. restrict our considerations to dual counterrotating rings (FLBH class, h=1). Typical access protocols are: token ring, and slotted ring. Generally fault location techniques are similar to those, described in p. 3.1. If multiple faults occur, then forward diagnostic loops are established in the token ring network, or backward diagnostic loops - in the slotted ring network. Details can be found in [9], [11]. The basic operational modes of a dual ring network after reconfiguration are: double loop , single loop (either of the loops), or the "C" loop (fig. 7).

Faults in a star - shaped ring network are located (and isolated) in cooperation of stations and active wiring concentrator (AWC). Each repeater generates timing signals when there is no message to send. Thus failure of a link, or repeater is immediately detected by the next repeater on the link by the loss of input signal. This station generates a beacon frame, including its address. When a beacon frame reaches the AWC it can identify the failure location and isolates it by closing a bypass switch.

4. CONCLUDING REMARKS

We gave a survey of most important fault-tolerant LAN topologies, followed by description of adequate fault location techniques, and network reconfiguration methods. After network reconfiguration further fault recovery actions take place. They depend on network architecture. Examples include: the reelection of monitoring stations, establishment of new virtual loops of tokens and so on.

Usually, in case of multiple faults, we observe a degraded recovery. A fault tolerant network, however, can continue its operation.

We presented a wide spectrum of the most important existing

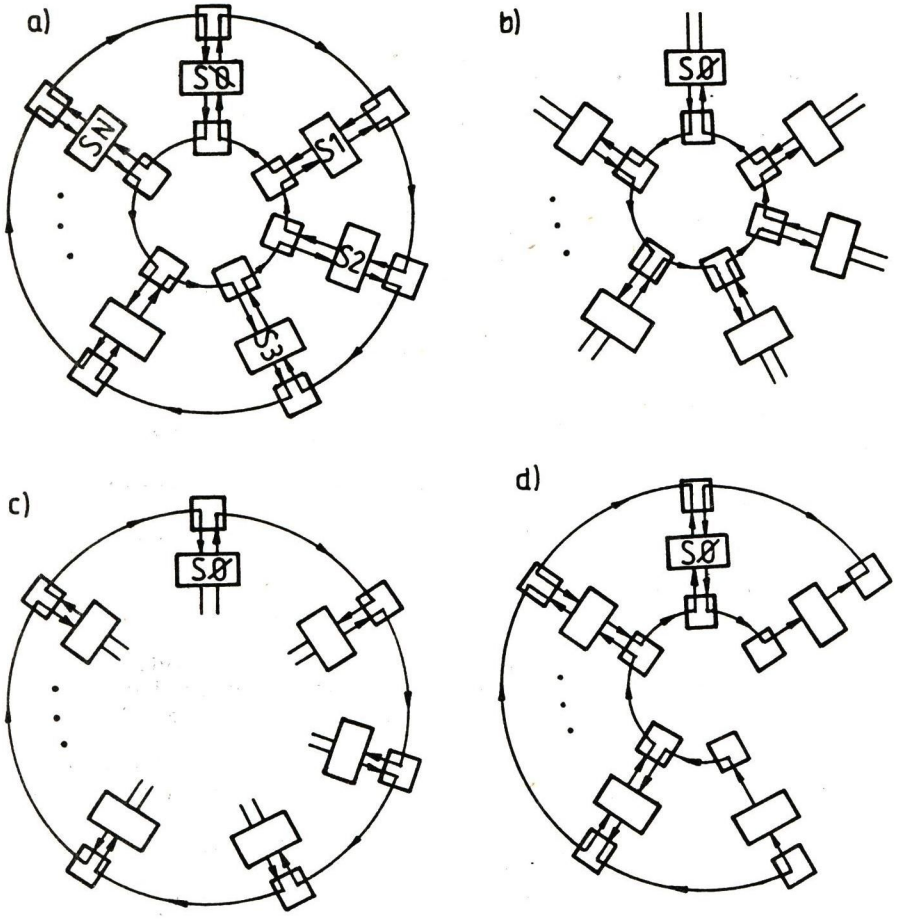


Fig. 7. Basic operational modes of a dual-ring network after reconfiguration : a) double - loop mode, b) inner - loop mode , c) outer - loop mode , d) "C" - loop mode

fault-tolerant LAN configurations. Other configurations exist (e.g. Expressnet, Fasnet), but they are not fault-tolerant.

A question arise : which network to choose ? The answer depend on the future application. The majority of structures reported here is offered by multiple vendors. Several users, however, would prefer some special designs for their applications. A detailed evaluation of network dependability should be made in such cases. Various problems arise here, due to complexity of Markovian models, dimensions of the state space, numerical solving of differential, or integral

equations, and so on. Such evaluations are out of scope of this paper. Our main task was to give a possibly complete survey of existing fault-tolerant networks. The inclusion of several hypothetical configurations was made for the completeness of presentation. In conclusion we can suggest the "natural" applications of homologues of the most popular configurations.

The double-bus structure is the simplest fault-tolerant instance of a multibus net. It is less expensive, than the quadruple-bus HYPERchannel network. We recommend the double-bus , token passing network for the MAP (Manufacturing Automation Protocol) [8] environment, to replace the existing IEEE 802.4 Standard, implemented there. The requirements of Computer Integrated Manufacturing Systems based on the MAP Standard , and in particular a loss , caused by a single medium failure, justify the usage of a double-bus network.

Similarily, a dual-ring configuration is the simplest acceptable instance of the multi - ring network. This structure is actually proposed by the IBM to replace the IEEE 802.5 networks. Similar configuration is also iplemented in the FDDI - II (Fiber Distributed Data Interface) Standard [1].

Both families of double media networks were investigated by the author ([9] - [12]). They are recommended here, since they are fault-tolerant at reasonable cost, offer good performability, and relative simplicity of implementation.

REFERENCES

1. Burr W.E., Zuqiu L.: "An overview of FDDI" ,Proc. EFOC/LAN 88 , Amsterdam, The Netherlands, June 29-July 1, 1988, pp.287-293.
2. Bux W., Closs F.H., Kuemmerle K., Keller H.J., Mueller H.R.: "Architecture and design of a reliable token-ring network", IEEE J. on Selec. Areas in Commun., SAC-1, (1983), pp. 756-765.
3. Chlamtac I., Ganz O.: "A study of communication interfaces for multiple bus networks", Computer Networks and ISDN Systems, 9 (1985), pp. 177-189.
4. Christensen G.S.: "Links between computer room networks", Telecommunications, 13 (1979), pp. 47-50.
5. Hohn W.C.: "The Control Data Loosely Coupled Network lower level protocols", Proc. of NCC, May 1980, pp.129-134.
6. ISO/IEC JTC 1/SC 6 N 5325: "New draft: enhancement of ISO 8802-5 Token Ring for multi-ring networks", March 1989.
7. Katz J.L., Metcalf B.D.: "SURVNET: a survivable network implemantetion", IEEE J. on Selec. Areas in Commun., SAC-6 (1988), pp. 950-958.
8. MAP : Manufacturing Automation Protocol, Version 2.1.,General Motors Corp. (GMC), Warren, Michigan, March 1985.

9. Molisz W.: "Automatic reconfiguration in local ring computer networks" (in Polish), National Telecom. Symp. KST 87, Bydgoszcz 1987, Proc. C, pp. 329-337.
10. Molisz W.: "Methods of fault-tolerance improvement in the IEEE 802 bus LANs" (in Polish), National Telecom. Symp. KST 88, Bydgoszcz 1988, Proc. C, pp. 328-337.
11. Molisz W.: "Fault-tolerance issues in local area networks", Proc. Fifth IASTED Int. Conf. "Reliability and quality control", Lugano, Switzerland, June 20-22, 1989, M.H. Hamza (Ed.), Acta Press. Anaheim, Calgary, 1989, pp. 61-64.
12. Molisz W.: "Management of fault-tolerant LANs" Proc. 9-th Int. Conf. "Remote Data Transmission", Most, Czechoslovakia, Nov. 7-10, CSVTS, Prague 1989, pp. 132-136.
13. Raghavendra C.S., Silvester J.A.: "A survey of multi-connected loop topologies for local computer networks", Proc. 1-st African Conf. on Computer Communications AFRICOM 84, 21-23 May 1984, Tunis, Tunisia.
14. Saltzer J.H., Pogran K.T.: "A star - shaped ring network with high maintainability". Computer Networks 4 (1980), pp. 239-254.
15. Strecker W.D.: "CI: a high speed interconnect for computers and mass storage controllers", Proc. of the Eight Int. Fiber Optics Commun. and Local Area Networks 1984, pp. 246-250.
16. Strole N.: "Inside token ring version II, according to big blue", Data Communications, No 1 (1989), pp. 117-125.

ERROR STATISTICS IN DATA NETWORKS

Jiří PŘIBYL and Igor VAJDA

Departement of Telecommunication
Czech Technical University
Prague, Czechoslovakia

Institute of Information Theory and Automation
Czechoslovak Academy of Sciences
Prague, Czechoslovakia

We consider the problem of measurement and efficient storage of error sequences in channels of telecomunication data networks. Various error statistics considered in the literature are reviewed and critically compared from the point of view of both the channel modelling and the code specifying. The hardware and software developed recently in this area at the Czech Technical University in Prague is briefly mentioned as well.

1. THE ERROR SEQUENCE

Let us consider a finite alphabet of r symbols with the Galois field structure, denoted by GF(r). Let $\underline{a} = (a_1,.., a_n) \in GF(r)^n$ be a message sent through a data network and $\underline{b} = (b_1,.., b_n) \in GF(r)^n$ the actually received form of this message. Then $\underline{e} = (e_1,.., e_n) \in GF(r)^n$ defined by

$$(1) \qquad \underline{e} = \underline{a} + \underline{b} \triangleq (a_1+b_1,\ldots, a_n+b_n)$$

is called an e r r o r s e q u e n c e of length n. Hereafter we shall be interested for simplicity in binary error sequences with the well known binary alphabet GF(2) = {0,1}.

There are many reasons why in a typical case all errors e_i will not be zeros of GF(r). These reasons have been discussed already in the fundamental papers of C. Shannon on information theory. In these papers one can find also the reasons, why the best description of the error sequence generator is probablistic, i.e. as a random sequence.

It is know from the literature (cf.e.g. Ephremidis and Snyder (1982)) that the well-fitted probabilistic models are quite complicated. In accordance with the philosophy of probability and statistics, the basic information about the probabilistic model is contained in the actually observed error sequence it-

self. One has to consult statistical literature in solving such delicate problems as how to choose the input sequence $\underline{a}$ in the p r o c e s s of measurement of $\underline{e}$, under what conditions to carry out the measurements, and how large should be n. There are however specialized papers on this subject (e.g. Adoul, Fritchman and Kanal (1972), Vajda (1984)) as well as papers describing the measurements and their results (e.g. Picardi and Trafton (1973)).

2. THE STORAGE PROBLEM

The first problem connected with the error sequence to which we call attention is how to store the error sequence effectively in the memory. The desired length n of the sequence $\underline{e}$ is usually very large, of order at least 10^6. Since the relative frequencey p of actual errors, i.e. of the events $e_i = 1$, in the sequence $\underline{e}$ is usually small, say $10^{-2} \leq p \leq 10^{-4}$, we have 99% or more zeros in this sequence and the direct bitwise storage of $\underline{e}$ is highly ineconomic. Considerable compression seems to be achievable by recoding $\underline{e}$ into the pair of integers (k, w) where $k = np$ is the total number of units in coordinates of $\underline{e}$ and where w denotes the order of the actually observed distribution of k units in the n positions of the sequence $\underline{e}$. Here, instead of n bits, the storage capacity of

$$\log_2 k_{max} + \log_2 \binom{n}{k_{max}}$$

$$\approx \log_2(np_{max}) - n[p_{max}\log_2 p_{max} + (1-p_{max})\log_2(1-p_{max})]$$

$$(2) \quad \approx np_{max}\left(\log_2 \frac{1}{p_{max}} + 1\right)$$

bits is sufficient.

This compression is however of little practical appeal as long as the complexity of computing the order w for the given distribution of k units and, conversely, of the distribution from the given k and w, remains unknown. We shall show that this computation requires not more than np_{max} log n bits of memory, but this solution in fact offers a simpler compression procedure as a whole with the adaptive memory capacity of $\log_2(np_{max}) + np\log_2 n$ bits.

Let k be as above and let us consider integers $m_1, .., m_k$ where $1 \leq m_1 \leq n$ denotes the position of the first unit in the sequence $\underline{e}$ and $0 \leq m_i \leq n-1$ for $i > 1$ denotes how many zeros

are situated between the i-th and (i-1)-st unit. Obviously,

$$k\log_2 n = np\log_2 n$$

bits are sufficient to describe the vector $(m_1,..,k_k)$. This memory capacity depends on the actual value of $k = np$. It never exceeds $np_{max}\log n$, which is for $p_{max} = 10^{-2}$ and $n \leq 10^7$ still less that (1), and is usually well below (1). The sequence $(k, m_1,..,m_k)$ is thus a suitable compression code for the error sequence $\underline{e}$ with the above stated capacity. On the other hand, the integer

$$w = \sum_{i=1}^{k} 10^i m_i$$

is related in one-to-one manner with the distribution of k units in the sequence $\underline{e}$ and can thus serve as the order required above.

In 1987 we equipped the czechoslovak computer SAPI 1 with the additional hardware and software necessary for the generation of a pseudorandom experimental input sequence $\underline{a}$ considered in (1) and, for control of the process of transmission of this sequence. The same computer at the receiver is able to evaluate the error sequence $\underline{e}$. The ontput of this computer may be the compression code $(k, m_1,..,m_k)$ for $\underline{e}$. But an additional reduction of demand on the memory capacity has been achieved by replacing too small integers m_i by the binary representation of blocks of digits $e_s, e_{s+1},...,e_{s+t}$ of fixed size t, where e_s is the first digit that follows the unit represented by the inleger m_{i-1}. This block covers in the memory the area normally reserved for m_i. In spite of at most k extra bits which have to be used as keys for how to read the momory areas reserved for m_1, m_2,.., this modification is very useful, particularly in channels with strongly bursting errors. In this arrangement the 20 kilobits reserve of internal memory of the computer suffices to store error sequences of length $n=10^6$ considered above. Such sequences can already be used for evaluation of error statistics considered in the following section.

3. ERROR STATISTICS

The error sequence $\underline{e}$ may be viewed as a statistical sample

from the unknown channel. In mathematical statistics any function of sample is called statistic. Some statistics are used directly in the process of choice of probabilistic model, others just to classify or decide some particular features of this model. The same situation is in telecommunication, where the "particular feature" is usually the convenience or inconvenience of some error detection or error correction code for the channel.

Simplest examples of statistics is the triple n, k, p defined in previous section. The length of error sequence, the number of errors and the error rate are basic descriptive statistics of any error measurement.

In Sec. 1 we have introduced another important statistics, namely the integer valued vector $(m_1, m_2,..,m_k)$ describing the gaps between errors and called gap statistics. (Note that the interpretation of m_1 as a gap presumes that the last bit transmitted before the error count started was bit by the error.) The distribution of gaps is a step rise distribution function with steps 1/k at points $m_{(1)},..,m_{(k)}$, where the latter is the ordered gaps statistics. Relative frequencies of steps in intervals (with linear or logarithmic division of the positive halfline) define histogram or density of gaps. Let us note that

$$\frac{1}{p} - 1 = \frac{n - k}{k}$$

is the expected gaps size.

Perhaps the most mysterious among error statistics are those related to the burst of errors. What is burst of errors?

Definition 1 (Alexander, Gryb and Nast (1960)). A burst with parameter t is any subsequence $e_s, e_{s+1},.., e_{s+t}$ of $\underline{e}$ such that $e_s = 1$.

Definition 2 (Přibyl (1978)). A burst with parameters t,d is any subsequence $e_s, e_{s+1},..,e_{s+t}$ of $\underline{e}$ such that $e_s = e_{s+t} = 1$ and all units in it, except e_1, are preceded by a string of at most d-1 zeros.

Definition 3 (Brayer (1968)). A burst with parameters $M \geq 2, 0 \leq \Delta \leq 1$ is a subsequence $e_s, e_{s+1},..,e_{s+t}$ of $\underline{e}$ with maximum possible t such that $e_s = e_{s+t} = 1$,

$$\sum_{i=s}^{s+t} e_i \geq M$$

and

$$\frac{1}{j+1} \sum_{i=s}^{s+j} e_i \geq \Delta$$

for every $1 \leq j \leq t$ such that $e_{s+j} = 1$.

D e f i n i t i o n 4 (Peterson, Weldon (1972)). A burst is the subsequence $e_s, e_{s+1}, \ldots, e_{s+t}$ of $\underline{e}$ containing all units of $\underline{e}$ with $e_s = e_{s+t} = 1$.

This review of definitions is far from being complete. We see that only the third definition satisfies the natural requirement that $\underline{e}$ is decomposed uniquely into b u r s t s and i n t e r v a l s between the bursts i.e. the complements of bursts (the definition 4 we consider trivial in the present centext).

Note that burst and intevals themselves are not considered to be statistics. Statistics are the l e n g t h o f b u r s t t+1 (cf. definition 3) and the l e n g t h o f i n t e r - v a l, the d e n s i t y o f b u r s t and the d e n - s i t y o f i n t e r v a l. By the density we mean in the present context simply the relative frequency of errors. Cumulative statistical characteristics of bursts and intervals are the (empirical) d i s t r i b u t i o n f u n c t i o n s o f l e n g t h s and o f d e n s i t i e s, for both bursts and intervals. Another cumulative statistics is the d i s t r i b u t i o n f u n c t i o n o f b u r s t c o n c e n t r a t i o n which is the ratio of lengths of burst and of the consecutive interval.

There and some other statistics useful for desing of detection and correction codes have been programmed into a statistical package for telecommunications at the computer SAPI 1 mentioned in the previous section. Simple graphics of tables of distribution functions is accompanying this package.

This package is a basis for another package on the some computer, which contains programs for selection of best-fitted multistate Markov chain models of errors in channels of telecommunication data networks. One advantage of probabilistic models against empirical error sequences $\underline{e}$ in simulation practice is that (1) potentially infinite sequences can be produced and, (2) slight modifications of parameters of these models may help to check the stability of simulation results or to investigate extremal situations not intercepted during measurements.

4. EXAMPLE

In Tab.1 one can see an e r r o r s e q u e n c e observed in a switched telephone line. This sequence is presented in term of g a p s t a t i s t i s t i c s ordered linewise from left to right.

836	1	4469	2	1	1	5	2	1	1
14	1	87	62	12	6	11	1	1	1
1	63	9	12	17	1	47	43	6	21
56	15	1	8	6	1	3	38	38	38003
3	347	9	8	8	600	7	61	16	31
1	41	3	15847	5	1	2	21	19	1
10	3	1	1	13	249	2072	45	9818	4
12	11	89	5	8	251	16280	24	176	13
621	85993	1	817	787	465	12	5	1298	2
49	1	7	17667	35644	226	202	1	1124	2

TABLE 1
Gap statistics of a switched telephone line

Tab. 2 presents the simplest statistics n,k,p of the error sequence of Tab. 1: the l e n g t h o f e r r o r s e - g u e n c e, the n u m b e r o f e r r o r s, and the e r r o r r a t e

n	234945
k	100
p	$4{,}3 \times 10^{-4}$

TABLE 2
The statistics n,k,p.

The d i s t r i b u t i o n o f g a p s for the sequence of Tab. 1 is presented in Tab. 3.

x \ d	10^0	10^1	10^2	10^3	10^4
d	.21	.47	.77	.89	.94
2 d	.26	. 6	.78	.91	.97
3 d	. 3	.63	.82	.92	.97
4 d	.31	.66	.83	.92	.99
5 d	.35	.71	.84	.93	.99
6 d	.38	.72	.85	.93	.99
7 d	. 4	.75	.86	.93	.99
8 d	.44	.75	.87	.93	.99
9 d	.46	.77	.89	.93	1

TABLE 3
Distribution of gaps

The remaining tables present statistics related to the bursts of errors according to Definition 3 for M = 2, = 0.01. Tab. 4 presents the distribution function of lengths of burst and Tab. 5 the distribution function of lengths of intervals.

x \ d	10^0	10^1	10^2
d	0	.333333	.733333
2 d	.2	.466667	.8
3 d	.266667	.6	.8
4 d	.333333	.6	.933333
5 d	.333333	.666667	.933333
6 d	.333333	.733333	1
7 d	.333333	.733333	1
8 d	.333333	.733333	1
9 d	.333333	.733333	1

TABLE 4
D.F. of lengths of burst

x \ d	10^2	10^3	10^4
d	0	.266667	.666667
2 d	6.66667E-02	.4	.8
3 d	6.66667E-02	.533333	.8
4 d	.133333	.533333	.866667
5 d	.133333	.6	.866667
6 d	.2	.6	.933333
7 d	.2	.6	.933333
8 d	.2	.6	.933333
9 d	.266667	.6	1

TABLE 5
D.F. of lengths of intervals

Tab. 6 presents the distribution function of density of burst and Tab. 7 the distribution function of density of intervals.

x \ d	10^{-2}	10^{-1}	10^0
d	0	.466667	1
2 d	0	.666667	1
3 d	6.66667E-02	.666667	1
4 d	.133333	.666667	1
5 d	.266667	.733333	1
6 d	.266667	.733333	1
7 d	.333333	.8	1
8 d	.4	.8	1
9 d	.466667	.8	1

TABLE 6
D.F. of density of burst

x \ d	10^{-5}	10^{-4}	10^{-3}
d	.8	.933333	1
2 d	.866667	.933333	1
3 d	.866667	.933333	1
4 d	.866667	.933333	1
5 d	.866667	.933333	1
6 d	.933333	.933333	1
7 d	.933333	.933333	1
8 d	.933333	.933333	1
9 d	.933333	.933333	1

TABLE 7
D.F. of density of intervals

At last, Tab 8 presents the distribution function of burst concentration.

x \ d	10^{-5}	10^{-4}	10^{-3}	10^{-2}
d	0	.214286	.428571	.857143
2 d	7.14286E-02	.357143	.714286	1
3 d	7.14286E-02	.357143	.785714	1
4 d	7.14286E-02	.357143	.785714	1
5 d	.142857	.428571	.857143	1
6 d	.142857	.428571	.857143	1
7 d	.142857	.428571	.857143	1
8 d	.142857	.428571	.857143	1
9 d	.142857	.428571	.857143	1

TABLE 8
D.F. of burst concentration.

The error sequence presented in this section was too short for the practical application. Therefore all results presented here are rather illustratitive. For longer error sequences all statistics are far more regular and their practical meaning is more evident.

5. APPLICATION TO CODES

The use of statistics of error sequences in the design of codes has been already well described in the literature (cf.e.g. Brayer (1968, 1971) or Barg, Vvedenskaya, Zyablov (1987)). The error statistics resulting from empirical error sequences or from error sequences generated by our probabilistic models have been used by our IBM PC programs for selection of best error correction codes from the classes of BCH codes (in particular Reed-Solomon codes), generalised binary Reed-Muller codes and generalised Hamming codes.

REFERENCES

Adoul J.P.A., Fritchman B.D., Kanal L.: A critical statistic for channels with memory. Trans. IEEE on Inform. Theory, IT-18 (1972), pp. 133-141.

Alexander A.A., Gryb R.M., Nast D.W.: Capabilities of the telephone network for data transmission. BSTJ 39 (1960), pp. 431-476.

Barg A.M., Vvedenskaya N.D., Zyablov V.V.: On error and erasure probabilities for block codes with incomplete decoding in a fading channel. Probl. Contr. and Inform. Theory 16 (1987), pp. 329-340.

Brayer K.: The improvement of digital HF communication through coding: I - interleaved cyclic codes. Trans. IEEE on Comm., COM-16 (1968), pp. 771-778.

Brayer K.: Error correction code performance on HF, troposcatter and satelite channels. Trans. IEEE on Comm., COM-19 (1971), pp. 781-787.

Ephremidis A., Snyder R.O.: Modelling of high error rate binary communication channels. Trans. IEEE on Inform. Theory, IT-28 (1972), pp. 549-555.

Peterson W.W., Weldon E.J.: Error-Correcting Codes (MIT Press, Massachusetts, 1972).

Picardi J.A., Trafton P.J.: Error behavior of tactical HF data transmission. Trans. IEEE Intern. Conf. of Communication (Seatle 1973), pp. 40/20 - 40/25.

Přibyl J. et al.: Research report on contract HS 5/78 B (Faculty of Electrical Engineering, Czech Tech. Univ., Prague 1978).

Vajda I.: Minimum chi-square estimates of multistate communication channel models. Probl. Contr. and Inform. Theory 13 (1984), pp. 343-356.

AN ALGORITHM FOR THE DESIGN OF MULTILEVEL CONCENTRATOR NETWORKS

G. Michael Schneider

Department of Mathematics/Computer Science
Macalester College
St. Paul, MN 55105

Mary N. Zastrow

Department of Computer Science
University of Minnesota
Minneapolis, MN 55455

Abstract

This paper presents an algorithm for the design of hierarchical, multi-level concentrator networks. The algorithm, based on the technique called clustering, is a generalization of an algorithm reported by P. McGregor and D. Shen and which appeared in the recent literature. The modification described here relaxes two of the assumptions made in the original McGregor/Shen formulation:

a) Solutions may have an arbitrary number of intermediate levels, instead of exactly zero or one.
b) The characteristics of a concentrator (i.e., capacity, cost, or local processing capability) can vary as a function of their placement within the solution.

These relaxed conditions lead to solution networks which can be lower in cost than those generated by the algorithm in its original formulation. This paper will present the algorithm and show some comparative results.

1. INTRODUCTION

One of the most well known and important research questions in computer network design is the *concentrator location* problem. In this problem, we have a large number of terminals at known locations and one central site to which all terminals must be connected. We wish to design a minimal cost tree network which connects all terminals to that central site, either directly or via one or more intermediate devices called *concentrators*. (Note that we are using the word concentrator in a generic, rather than technical, sense. Any type of line-sharing device - e.g., multiplexor, cluster controller, minicomputer - will be referred to here as a concentrator.)

This problem has been extensively studied in the literature. Exact solutions can be obtained by formulating the problem as

a 0,1 integer programming problem [1]. However, the computational complexity of the problem, especially with large networks, has motivated a number of heuristic approaches. One of the most widely used is the class of "Add/Drop" algorithms [2,3] which start with either a minimal (or maximal) network and then attempt to add (or drop) one concentrator at a time to see if it results in an improved design. A number of other heuristic approaches can be found in [4,5,6]. Excellent surveys of the work in this field are contained in [7,8].

A totally new approach, based on the concept of *clustering*, was reported in an article by McGregor and Shen in the January, 1977 issue of *IEEE Trans. on Communications* [9]. Their method involves the grouping together, or clustering, of geographically adjacent terminals based on some quantitative "similarity measure" which identifies the terminals as part of the same group. The similarity measure used in [9] is maximal terminal/center-of-mass (COM) distance. Namely, if a terminal is a distance d < dmax away from the COM of a growing cluster it is joined to the cluster, and is reset accordingly (Fig. 1).

After terminals have been clustered they may potentially be replaced by a concentrator located at (or near) the COM, if it is determined to be cost-effective.

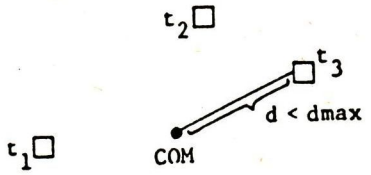


Fig. 1. Clustering of Terminals.

Results published in [9] compare the performance of the clustering method with an algorithm called ATD [6], believed to be one of the best reported in the literature up until that time. The results show that clustering produced solutions which were lower in cost (by about 5%) in 23 out of 25 test cases. And, it reduced the processing time needed to produce those solutions by about 85%. It appears that the clustering algorithm described in [9] is the best heuristic for the concentrator location problem currently available in the literature.

However, the McGregor/Shen clustering algorithm contains a number of restrictions which the authors themselves describe in the conclusion of the paper. Two of the most severe (in terms of constructing practical, "real-world" networks) are closely related - *staging* and *multicapacity concentrators*.

In the algorithm described in [9], the network topology is limited to a single intermediate layer - terminals must be

connected either to a concentrator or to the central site. (Fig. 2a) However, almost all centralized terminal-oriented networks (even moderate sized ones) allow multiple levels of concentrators - what is termed a staged network. Concentrators may then be connected to either other concentrators of the central site (Fig. 2b).

This ability to stage a solution is critically important in keeping communication costs manageable during overall systems design. There is a well-known "economy of scale" existing in the communications marketplace which says that we can purchase roughly 10 times the communication capacity for roughly twice the cost. Therefore duplicate long communication lines as in Fig. 2a, are highly inefficient and very expensive. It will almost always be more efficient to replace 2 long lines of capacity k with 2 local lines of capacity k to a new concentrator C, and one long line of capacity 2k to the central site (Fig. 3). Obviously, this is only possible when multilevel solutions are allowed.

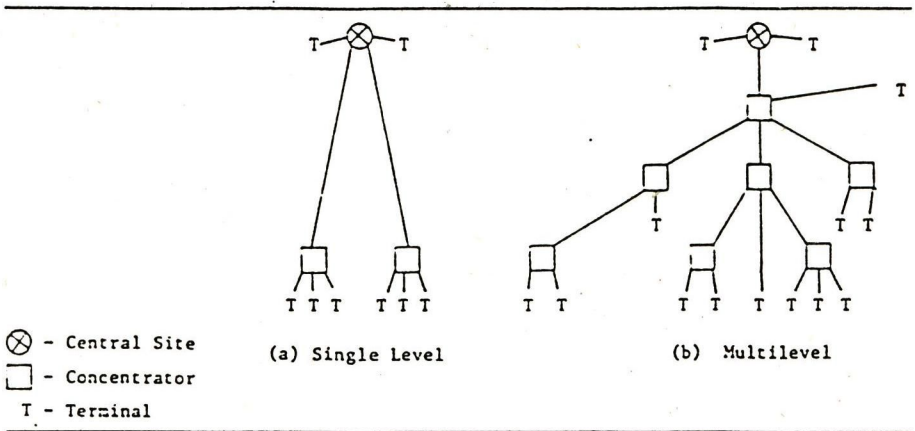


Fig. 2. Types of Concentrator Networks.

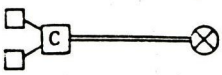


Fig. 3.

Closely associated with a multilayered topology is the idea of concentrators of varying capability. In real-world networks we frequently encounter line-sharing devices with widely varying performance characteristics - frequency or time-division multiplexors (FDM, TDM), statistical multiplexors (ITDM), cluster controllers, and programmable concentrators. While all these devices fall under the umbrella of the term "concentrator", they obviously differ markedly in cost and capability. Restricting solutions to a single uniform class of concentrator would again severely limit the utility of the ultimate solutions.

This paper will be presenting a modification of the McGregor/Shen algorithm described in [9] to relax the two restrictions stated above. Specifically, the algorithm will allow the user to:

a. Create hierarchical topologies containing an unbounded number of intermediate levels 1, 2, ..., k. Any device at level i can be connected to a device at level j, $i \leq j \leq k + 1$.
b. Parameterize the unique characteristics of concentrators at level i, $1 \leq i \leq k$. These parameters will describe the following three characteristics:
   1. Cost
   2. Input Capacity
   3. Local Processing Capability

Section II will describe the algorithm and data structures while Section III will present some comparative results obtained from coding and testing this algorithm. The program to implement this algorithm (in standard Pascal) is available by request from the authors.

2. THE ALGORITHM

This section of the paper is organized in the following fashion:

2.1. Overview
2.2. Data Structures
2.3. Notational Conventions
2.4. The Algorithm

2.1 Overview

Given the geographic location and mean traffic rate of M computer terminals, as well as the location of a single central site, X, this algorithm attempts to determine a minimal cost, multilevel concentrator network using a variation of the clustering technique developed by McGregor and Shen and described in [9].

The program performs its analysis in stages. Initially, the algorithm attempts to connect terminals (sometimes referred to as level 0 devices) to concentrators. If the algorithm succeeds in adding at least one cost effective concentrator at level 1, it then attempts to connect these level 1 concentrators, as well as any as yet unconnected terminals to concentrators at level 2. This process is repeated at levels 3, 4, ..., k until no concentrators are found to be cost effective at the current level. Finally, all unconnected devices and the level k concentrators are connected to the central site, level $k + 1$, completing the topology.

2.2 Data Structures

2.2.1 Nearest Neighbor Table (NNT)

The Nearest Neighbor Table is a list of all unassigned device pairs whose geographic separation is less than some upper bound, called *dmax*. *dmax* is a function of the level at which we are currently clustering. (A separation of a few hundred feet may be appropriate for clustering terminals into an FDM but is totally inappropriate for a regional concentrator.) Thus, *dmax* = *dmax*(level). The computation of dmax is described later in the paper.

```
type NNT = array [1..max] of distance-records;
    distance-records = record
                device1 : integer; (*index of the first
                                    device*)
                device2 : integer; (*index of the second
                                    device*)
               distance : real(*separation in miles*)
                                    end;
```

NNT is reconstructed for each level, *i*, in the developing solution. When constructing NNT for level *i* the devices represented in the 'device1' and 'device2' fields of the table are any devices - terminals or concentrators at levels 1, 2, ..., *i* - 1 - which are as yet unassigned to a device at a higher level.

2.2.2. Network Structure List (NSL)

NSL is a permanent record of the multilevel network being constructed by the algorithm. It includes a record of every 'device' contained within the solutions. The major item in the record is the unique *index number* of that device. Indices 0, ..., M are permanently assigned: device 0 is the central site, while devices 1, ..., M are the terminals. This information is input to the algorithm and is entered during initialization. All other devices M + 1, M + 2, ... are concentrators created by the algorithm. (There is no a priori upper bound on the number of devices or levels which can appear in the solution.)

```
type NSL = array [1..max] of device-records;
     device-records = record
                  v,h     : real;    (*geographic v,h
                                      coordinates*)
                  traffic: integer;  (*output traffic rate of
                                      this device*)
                  level   : integer; (*the level of this
                                      device in the multilevel
                                      solution*)
                  link    : integer; (*the index number of the
                                      device to which this
                                      device is upwardly
```

```
                                    connected, or 0 if not
                                    connected)
        speed   : integer;  (*the speed of the link*)
        end;
```

The *index* of a device is its position (i.e., subscript) within the NLS array. The *v, h* co-ordinates are as described in FCC Tariff 255. The distance (in miles) between 2 devices can be computed from their *v, h,* coordinates by:

$$\text{distance} = \sqrt{\frac{(v_a - v_b)^2 + (h_a - h_b)^2}{10}}$$

The traffic rate of the terminals (level 0) is an input parameter, $t_{0,k}$, $k = 1, 2, \ldots, M$. The traffic rate of a concentrator at level i, denoted $t_{i,k}$, is then simply the sum of the traffic rates of the incoming devices at lower levels multiplied by the *compression factor* for concentrators at level i, denoted α_i. $0 < \alpha_i \leq 1$.

The parameter α_i is a measure of the amount of local processing carried out by a concentrator and is described in more detail in the next section.

At any time in the developing solution, the only candidates for inclusion in the NNT, and therefore candidates for clustering, are those devices for which NSL. link = 0. The speed of the link (in bps) is chosen from a discrete set S = $(s_1, s_2, \ldots s_n)$ such that we satisfy the traffic demand with the minimal extra capacity. The set S reflects the current marketplace in data communications services.

2.2.3. The Clustering Workspace (CW)

The clustering workspace is a set of linked lists - one list for each device which is a candidate for clustering (i.e., which is currently unassigned to a higher level device.) See Fig. 4.

```
type head = record
     v,h  = integer; {weighted center of mass of the entire
                      cluster}
  traffic = integer; {sum of all traffic in the cluster}
     ptr  = ↑node {pointer to the cluster list}
          end;

     node = record
    index = integer; {NSL index of this device}
  traffic = integer; {traffic rate of this device}
     ptr  = ↑node {pointer to the next node in the list}
          end;
```

The smallest value of NNT[*i*] distance (i.e., the closest pair of unassigned devices) is found and a link is created between the list headed by device1 and device2, wiping out the head node of device2. See Fig. 5. Each linked list is grown until the sum of the traffic of all devices on that list is equal to or greater than the input capacity of a concentrator at this level.

Informally, we can say (as did McGregor/Shen) that CW is used to "roll snowballs", attaching geographically close devices together as we go. However, unlike McGregor/Shen, our CW must be reinitialized for every level since the set of unassigned devices changes at each pass.

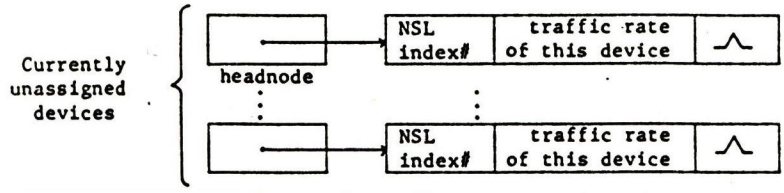


Fig. 4. Initial Structure of the Clustering Workspace.

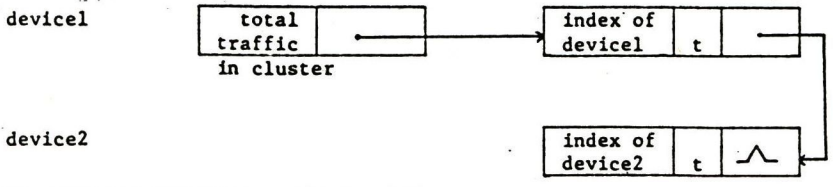


Fig. 5. Merging Items in the Clustering Workspace.

2.3. Notational Conventions

The notation used in the description of the algorithm is as follows:

k The number of concentrator levels in the solution $k \geq 0$. There is no a priori bound on k.

C_{ij} Concentrator j at level i, $i = 1, \ldots, k$, $j = 1,2, \ldots$

M The number of terminals.

T_i The terminals $i = 1, \ldots, M$. They may also be denoted $C_{01}, C_{02}, \ldots, C_{0M}$ or called level 0 devices.

t_{ij} The mean output traffic rate (in bits/sec) of device j at level i.

X The central site. Also referred to as $C_{k+1,0}$ - the level $k + 1$ device.

α_i The compression factor of all concentrators at level i, $1 \leq i \leq k$.

W_i The capacity (bits/sec) of all concentrators at level i, $1 \leq i \leq k$.

D_i The cost of all concentrators at level i, $1 \leq i \leq k$.

Fig. 6 shows a typical network design generated by the algorithm:

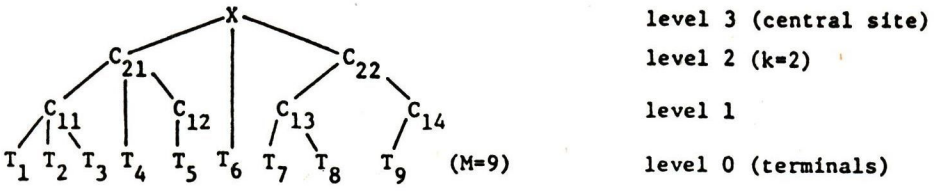


Fig. 6. Sample Solution.

2.4. The Algorithm

The structure of the solution must satisfy the following four constraints:

1. Every terminal $T_1, \ldots, T_M$ must be connected to exactly one device at a higher level l, $1 \leq l \leq k + 1$.
2. Every concentrator, C_{ij}, must be connected to exactly one device at a higher level l, where $i + 1 \leq l \leq k + 1$.
3. The total traffic into a concentrator at level i from all devices at levels l, $1 < i$, may not exceed the capacity, W_i of concentrators at level i.
4. The capacity of the central site, W_{k+1} must be sufficient to handle all terminals:

$$W_{k+1} > \sum_{i=1}^{M} t_{0i}$$

1. Initialization

a) Initialize all concentrator-based constants:

The *capacity* W_i, $i = 1, 2, \ldots, n$

The *compression* α_i, $i = 1, 2, \ldots, n$

The *cost* D_i $i = 1, 2, \ldots, n$

Since the number of levels, *k*, in the final solution is unknown, the algorithm assumes that the last constant value supplied applies to all unspecified higher levels:

$W_k = W_{k-1} = \ldots = W_n$

$\alpha_k = \alpha_{k-1} = \ldots \alpha_n$

$D_k = D_{k-1} = \ldots = D_n$

b) Input the set of available link speeds. The speed of any communication link must be a member of S:

$S = \{s_1, s_2, \ldots, s_n\}$

c) Set the level counter, l, to 1. l tells us the level in the solution we are working on l = 1, 2, ...

2. The Input Data

Input the following 4 classes of data:
a) The geographic *v*, *h* coordinates of the central site
b) M, the number of terminals
c) The geographic *v*, *h* coordinates of every terminal
d) The mean traffic rate of each terminal $t_{01}, \ldots, t_{0M}$

3. The Initialization of the Network Structure List

The input data from step 2 is entered into NSL. The level field is set to 0 for the terminals, and they are marked as currently unconnected to any other device.

```
for i := 1 to M do
begin
     NSL[i].v :=v;
     NSL[i].h := h;
```

```
    NSL[i].traffic := t0i;
    NSL[i].level := 0; {terminals are level 0 devices}
    NSL[i].link := 0; {they are currently unconnected}
    NSL[i].speed := 0
end
```

4. Create the Clustering Workspace

For each currently unconnected device, *i*, in NSL (i.e., NSL[*i*].link = 0) create a 2 node linked list of the form of Fig. 7.

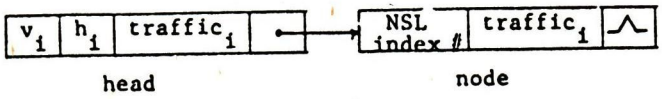


Fig. 7. Initial Contents of the CW.

5. Determine the Maximum Clustering Distance

As we mentioned earlier, the value for the maximum clustering distance, *dmax*(*l*), is a function of the level *l* we are currently at. Therefor, dmax must be determined on the basis of the current geographic characteristics of the unconnected devices. If *dmax* is too small, most of the concentrators will be underutilized. That is, the number of devices connected to C_{ij} will be less than W_i (Fig. 8a). Alternately, if *dmax* is too large our clusters may contain "spurrious" terminals which do not properly belong (Fig. 8b).

Naturally, as the mean distances between all available devices increase the maximum clustering distance must also increase:

$dmax(l) = f(\overline{d}_{CW})$

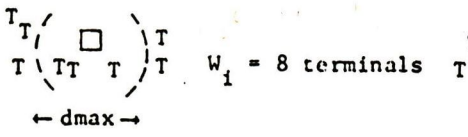


Fig. 8a.

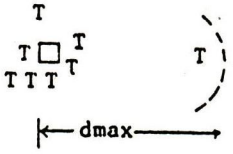


Fig. 8b.

where $\overline{d}_{CW}$ is the average interdevice distance of all devices with current entries in CW. However $\overline{d}_{CW}$ alone is insufficient to capture the appropriate value of dmax because it does not consider the capacity of the level l concentrators. For example, in Fig. 9, $\overline{d}_{CW} = 1.67$ for the three unconnected devices.

If W(l) = 2 then a maximum clustering distance of 1 would be adequate. However, if W(l) > 3, we would want dmax(l) > 2. Because of this dmax(l) must be dependent not only on d_{CW} but also on the ratio r:

$$r = \frac{W(l)}{N_{CW}}$$

where N_{CW} is the number of current entries in the CW.

As $r \rightarrow 0$, we either have a great number of unconnected devices or a very limited concentrator capacity, and should be able to fill a concentrator in a relatively small geographic area. As $r \rightarrow \infty$ we have either few candidate devices and/or a large capacity, and must increase our geographic search. Therefore, our first attempt at computing dmax was:

$$dmax(l) = \frac{\overline{d}_{CW} W(l)}{N_{CW}}$$

However, this formulation was insufficient. To be truly cost effective, concentrators should operate at or near the capacity W. (That is, the problem illustrated by Fig. 8a is more severe than that illustrated by Fig. 8b.) The incremental cost of adding another terminal to a concentrator (up to W) is usually small when compared to:

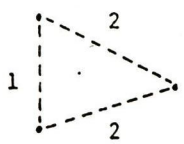


Fig. 9. $\overline{d}_{CW} = 1.67$.

a. The cost of a second concentrator
b. The cost of connecting two separate concentrators to a higher level.

Therefore, we tested a maximum merging distance computation of the form:

$$dmax(l) = \frac{K\,\overline{d}_{CW}W(l)}{N_{CW}},$$

where K is chosen to insure that all concentrators operate at or near capacity, but not so large as to create the problem of spurrious clusters (Fig. 8b). Tests of various terminal distributions and cost functions showed the best results available when K = 4. This value was used in all succeeding computations.

6. Create the Nearest Neighbor Table (NNT)

Calculate the distance, in miles, between every pair of devices in CW. If this value is less than or equal to dmax(l), then enter the indices of the devices and their distance into the next available entry in NNT.

7. Begin To Form A Cluster

Each device cluster, as it is being formed, is represented by a single "virtual node" located at the weighted center of mass of the individual devices belonging to the cluster. The traffic rate into the virtual node is the sum of the traffic of all devices in the cluster.

We initially determine the minimal value in the 'distance' field of NNT. Assume the indices of these devices are j, k. We now check the CW head nodes of these two device to see if they are candidates for merging into a single cluster. The sum of the traffic rates stored in the traffic rate fields of CW head nodes j and k must not exceed the concentrator capacity W(l). If not, then we can merge list k into list j where $j < k$. See Fig. 10.

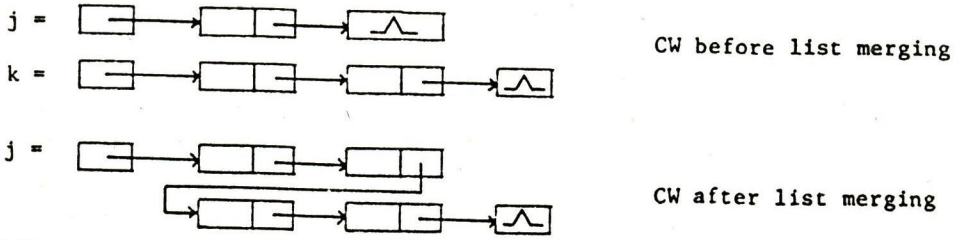


Fig. 10. Merging Items in the CW.

8. Reset CW

If the merge took place we now must recompute the three fields in the head node of CW[*j*]:

a) The new traffic rate will be computed to reflect the added traffic due to the newly added nodes.

b) The weighted center of mass must be recomputed the stored in the *v*, *h* fields of the head node.

9. Reset NNT

If the merge took place then we update NNT to reflect the fact that cluster *k*, which was merged into cluster *j*, does not, in a sense, exist any longer. That is *k* is no longer a distinguishable entity but a weighted portion of a "virtual node" located at the center of mass of cluster *j*. To do this we mark all NNT entries to reflect *k*'s "disappearance", and change the distances of all NNT entries involving *j* to reflect the new location of the center of mass. All devices connected to device *k* have their distance field set to ∞.

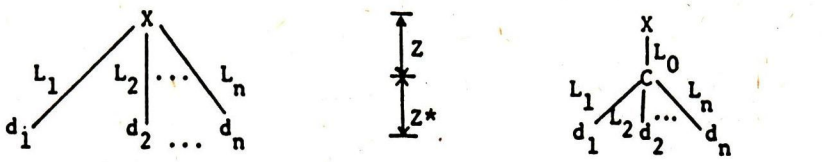
10. Complete The Clustering

Now go back to Step 7 and continue merging clusters until no distance entries in NNT are less than *dmax*(*l*). this implies that all nodes which can be merged have been merged. We now have a group of *n* clusters (represented by n linked lists in CW), no one of which has an input traffic rate which exceeds W(*l*).

11. Concentrator Placement

We now determine the cost effectiveness of a concentrator. We temporarily place a concentrator within each cluster formed by step 7-10. (A cluster will be defined now as any entry in CW which still has a head node.) The location of the concentrator is at the actual device within the cluster nearest to the theoretical center of mass. We lay out point-to-point lines of speed *s* from each device in the cluster to the newly placed concentrator and from the concentrator to the central site, s is chosen from S, the set of all possible link speeds. Using a cost procedure described in the next section, we find both the original cost of connecting all devices in the cluster directly to a central site, and the new cost of connecting all devices to the central site via the newly placed concentrator, including D, the cost of the concentrator itself. See Fig. 11.

This computation involves a potentially incorrect assumption - that the concentrator C will be connected directly to the central site, X. This is only true when the succeeding level, *l* + 1, does not contain any concentrators. However, since we do not as yet know the characteristics of level *l* + 1 we must



Old cost = $\sum_{i=1}^{n}$ cost of line L_i

New cost = $D_\ell + \sum_{i=0}^{n} L_i$

Fig. 11. Cost Computations.

make this assumption. Because of the economy of scale argument made earlier, this assumption will lead the algorithm to be overly optimistic in its placement of concentrators. This is because the cost of line L_0 of capacity n and distance Z will typically be much less than n lines of capacity 1 and of distance Z + Z* (where Z >> Z*). However, in our approximation the true value of Z will often end up being much less than the assumed value of Z. This is because the line L_0 will be connected to yet another intermediate device $C_{l+1,j}$ and not the central site, X. Since the distance is less, the economy of scale savings are less. However, test results have shown that the algorithm still produces improved results as long as concentrator costs, D_l, are small when compared to overall line costs.

If the cost factor improves then we permanently write the concentrator into NSL and connect all devices from the cluster into that device. The output traffic rate of this new device is set at:

$t_{ij} = \alpha_i$* (sum of the traffic of all devices in the cluster)

$0 < \alpha_i \leq 1.$

α_i is a measure of the local processing capability of concentrators at level i. Specifically, it measures the amount of information processed locally and not passed on to succeeding levels. This local processing capability could represent:

1) Local echoing

2) Local editing
3) Blank compression
4) Local data storage and retrieval
5) Other local processing services

For unintelligent devices, like FDMs and TDMs, $\alpha_i = 1$. For intelligent concentrators and front-end processors, we could have $\alpha_i < 1.0$.

If the cost factor did not improve, we simply disregard the cluster which has been formed and make no change to NSL.

Step 11 is repeated for every cluster contained in CW.

12. Move on to the next level:

$l := l + 1$

13. Repeat steps 4-10 until the operations in step 11 do not add any new concentrators to the network topology (i.e., do not modify the NSL)

14. Add the central site, X. to the Network Structure List and connect all previously unconnected devices to the central site X with a line of appropriate capacity. This completes the topology and the algorithm.

3. Computational Complexity

The rate determining operation in the algorithm is the initial execution of Step 4 - the generation of the Nearest Neighbor Table, NNT. The time required to generate NNT is $O(k^2)$, where k is the number of as yet unconnected devices. Initially, this is $O(M^2)$ and, for large M, becomes the limiting step. However, at succeeding levels the value of k drops by approximately:

$$k_{i+1} = k_i - (n_i * W(i) + n_i) \quad (k_0 = M)$$

$$= k_i - n_i(W(i) - 1)$$

where n_i = number of concentrators at level i, W_i = capacity of a concentrator at level i.

Thus, it is the initial generation of NNT which is the expensive operation. The succeeding levels require significantly less time. However, both the McGregor/Shen and

the Schneider/Zastrow algorithms must pay the high cost of this initial operation on NNT. The higher levels within the solution, which require relatively less in the way of computational resources and which can significantly reduce costs, are only generated by the algorithm described here.

The space complexity of the algorithm is also determined by the data structure NNT. In the worst case, the space needs of NNt are also $O(M^2)$. However, this is only possible in the pathological case where all M terminals are less than a distance d < dmax(1) from each other, where:

$$dmax(1) = \frac{4\,\overline{d}\,W(1)}{M},$$

$\overline{d}$ is the average interterminal distance, and W(1) is the capacity of a level 1 concentrator.

However, for this to happen, would require:

$$P = \frac{4W(1)}{M} >> 1$$

so that the average terminal separation, $\overline{d}$, is far less than the maximum clustering distance at level 1. But, typical values for the parameters W and M are:

W ≈ 4, 8, or 16 (capacity of a concentrator)
M ≈ 200 - 1,000 (number of terminals)

and typical values of P above would be in the range 0.01 to 0.025 - well less than 1. The average space needs for NNT will be significantly less than the worst case behavior of $O(M^2)$.

Because of the $O(M^2)$ space and time behavior of this algorithm, it appears that it will only be possible to apply this algorithm to moderately sized networks where M ≤ 1000 - 2000. Timing results discussed in the next section confirm this observation.

4. Results

The algorithm described in the previous sections was coded in the language Pascal and run on a CDC 6600 computer at the University of Minnesota. In order to demonstrate the improvements available through a multilevel solution, we also needed to develop a realistic cost procedure which more accurately reflects the true "economies of scale" available

in the communications marketplace. (McGregor/Shen used a simple linear communication cost function.) The cost analysis used in the following examples were based on actual rates for leased lines as described in F.C.C. Tariff 260 (effective February 2, 1979). The set S, of lines which were used is given in Table 1.

In addition, all necessary line termination costs (channel terminals, service terminals) and modem costs were included in the overall network cost. Because of these difference in cost procedures direct cost comparisons between results presented here and in [9] are not possible.

Three test cases were run with 50,100 and 200 terminals. Cost comparisons were made between the multilevel solution generated by the algorithm described in Section III and a solution which was artificially limited to one intermediate level. (This latter topology is the type shown in Figure 2a and the type which is generated by the algorithm reported in [9].) The results are summarized in Fig. 12. The conditions which generated these results were:

a) Terminals generated equal amounts of traffic and were randomly distributed in a grid of approximately 2000 miles square.

b) The central site was located in the center of the grid

c) $W_1 = 16$ terminals, $W_2 = 8$, $W_i = 4$, $i = 3, 4, \ldots$

d) $\alpha_i = 1.0$ for all i

e) $D_1 = \$128$, $D_i = \$420$, $i = 2, 3, \ldots$

The results show that for these cases the savings to be made were in the range of \$1000 - \$9000, or approximately 6% - 15% of the overall systems costs. These are not insignificant amounts and point out the savings which can be realized by eliminating replicated, and expensive, lower capacity communication links of the type shown in Fig. 1a.

Table 1. Set of Available Line Speeds.

Grade	Series	Speed (bits/sec)
Subvoice	1002	0– 55
	1005	55– 75
	1006	75–150
Voice	3000 ("A" rate centers)	300–9600
Wideband	8800A	>10,000

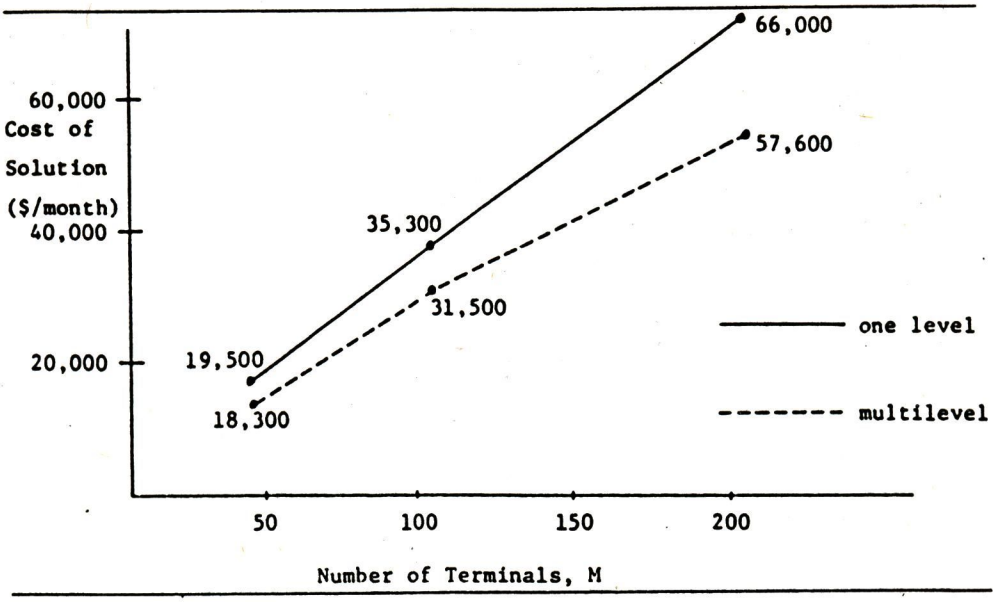


Fig. 12. Cost Comparison Between One Level and Multilevel Solutions.

Other test cases were run on the algorithm using varying topologies and concentrator parameters. They showed that the magnitude of the potential cost savings were strongly affected by the following characteristics:

1) An increase in M, a decrease in W(1), or both. As the number of terminals increases in relation to W(1), the capacity of a level 1 concentrator, we may generate an inordinately large number of level 1 devices in our solutions. Without the possibility of further intermediate levels we may have to connect these concentrators to the central site using expensive long lines.

2) The initial value of $\overline{d}$, the average interterminal distance between all M terminals. Obviously, as the terminals become separated by greater distances, the more important the economy of scale argument becomes, and the greater the cost savings available through intermediate levels.

However, in those realistic terminals networks which would make use of these algorithms, both of these conditions are usually met. In most networks M is typically in the range 100 - 1000 while W_1 is 8, 16, or, at most 32. Thus, $M >> W_1$.

Likewise, the M terminals in most networks are typically distributed over wide geographical areas - states, regions, or the entire country - leading to a large average interterminal separation. Finally, over the past few years the hardware costs of all types of line-sharing devices (D_1) has been falling much more rapidly than the cost of communication lines. Overall, then, one would expect the

typical savings to be gained through a staged, multicapability concentrator network to be significant. And, while pathological topologies can be constructed for which there are no cost savings (indeed even a loss because of some of our assumptions), the average reductions which can be expected for a realistic set of topologies should be in the range of about 5 - 20%.

As a final set of tests, we ran our algorithm or a test case containing 400 randomly distributed terminals. The algorithm generated a solution with 4 intermediate levels. The solution is shown visually in fig. 13 and summarized in Table 2. The value M = 400 was chosen specifically to be able to compare running times of our algorithm with the McGregor/Shen algorithm and the ATD algorithm reported in [9]. (Their algorithms were also run on a CDC 6600.) Times for generating solutions on networks with 50, 100, 200, and 400 terminals are summarized in Table 3. Table 3 shows that a multilevel solution to the concentrator network problem adds little to the execution time of the program. This is because, as mentioned earlier, the significant time is involved with construction of the first level of the tree - an operation which must be done by both the McGregor/Shen and Schneider/Zastrow algorithms. Levels 2, 3, 4, ... work on significantly smaller sets of devices (69, 13, 3, 1) in our 400 node test case. Thus, the higher levels are, in a sense, "free" since the bulk of the work has already been done.

Based on the timing results of the test cases listed in Table 3, it appears that it will be reasonable to run this algorithm on moderately large networks of $M \leq 1000 - 2000$. For such values of M, running times on a CDC 6600 can be expected to be in the range of 5 - 30 minutes. However, the $O(M^2)$ time complexity on initial generation of the NNT data structure probably precludes its use on very large networks, say M ~ 5000 - 10000. Rough computations show that for networks of this size running times may exceed 10 - 20 hours on a CDC 6600. Future research on this project will be directed at improving the efficiency of this algorithm so that it will be applicable to the design of very large hierarchical concentrator networks.

Table 2

Level	Number of Devices	
0	400	(terminals)
1	69	
2	13	
3	3	
4	1	
5	1	(central site)

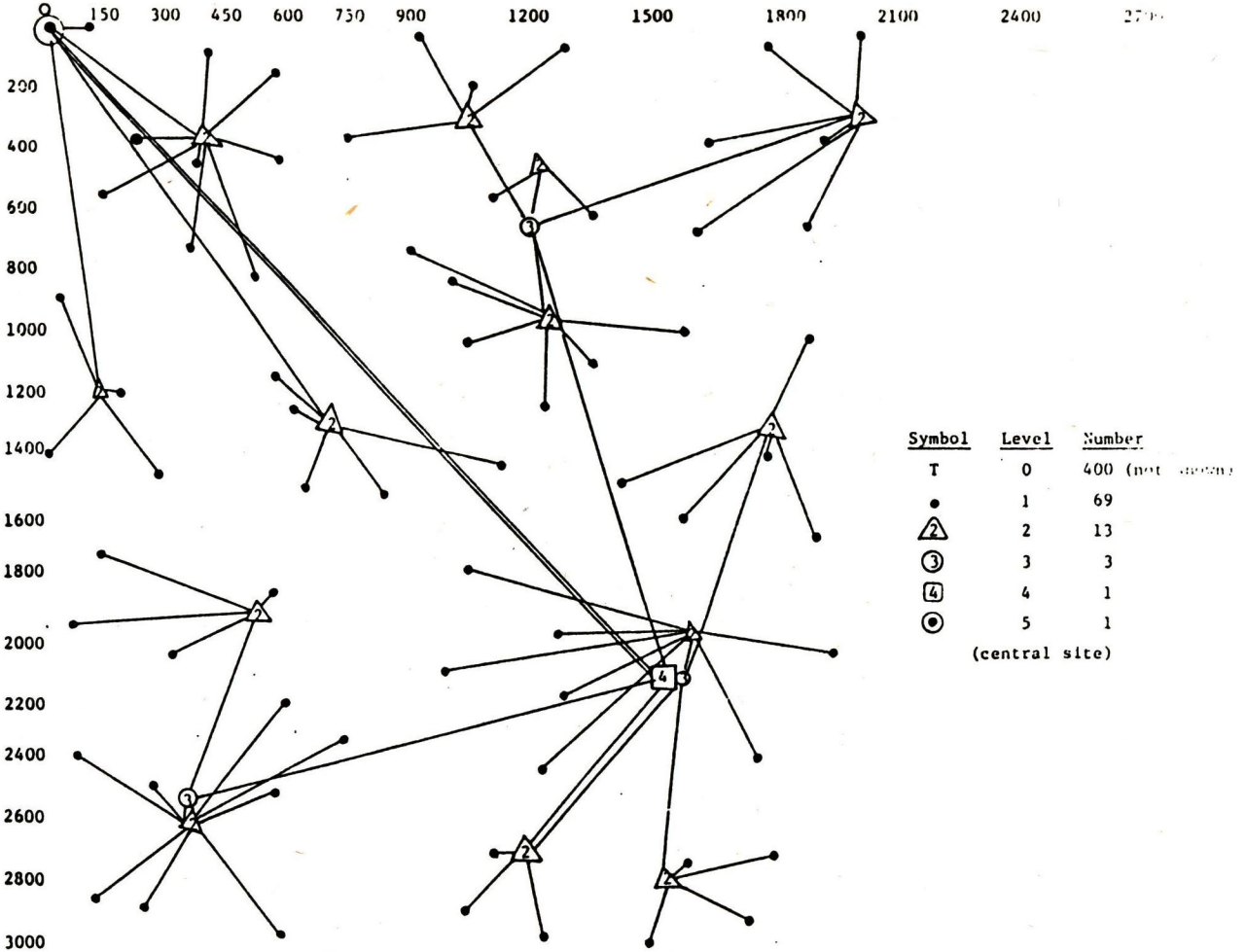


Fig. 13. Four-Hundred Node Test Case.

Table 3
Comparison of the Execution Times of Three Algorithms

	Execution Time (seconds)			
	50	100	200	400
McGregor/Shen	–	2.1	8.2	36.0
Schneider/Zastrow	3.2	4.0	11.9	40.2
ATD	–	9.4	67.7	275.0

References

[1] P. Gray, "Exact Solution to the Site Selection Problem by Mixed Integer Programming", Stanford Res. Inst. Tech Report 5205-28, 1967.

[2] A. Kuehn, M.J. Hamberger, "A Heuristic Program for Locating Warehouses", Management Sci, Vol. 9, July, 1963.

[3] L. Bahl, D. Tang, "Optimization of Concentrator Locations in Teleprocessing Networks", Proc. Symp. on Comp. Comm. Networks, Brooklyn, NY, 1972.

[4] D.A. Greenberg, "A New Approach For the Optimal Placement of Concentrators", Natl. Telecomm. Conf. Atlanta, GA, 1973.

[5] H. Frank, W. Chou, "Topological Optimization of Computer Networks", Proc. of the IEEE, Vol. 60, Nov., 1972

[6] L.S. Woo, D. Tang, "Optimization of Teleprocessing Networks with Concentrators", Natl. Telecomm. Conf., Atlanta, GA, 1973.

[7] Mischa Schwartz, Computer-Communication Network Design and Analysis, Prentice-Hall, Englewood Cliffs, NJ, 1977.

[8] R. Boorstyn, H. Frank, "Large Scale Network Topological Optimization", IEEE Trans. on Communications, Vol. COM-25, No. 1, January, 1977.

[9] P. McGregor, D. Shen, "Network Design: An Algorithm for Access Facility Location Problem", IEEE Trans. On Communication, Vol. COM-25, No. 1, January, 1977.

SOFTWARE RELIABILITY OPTIMIZATION IN DISTRIBUTED INFORMATION CONTROL SYSTEM

S.G.Trifonova, V.A.Tzanov* and R.T.Trifonov**

* Central Laboratory of Control Systems B.A.S. Sofia, Bulgaria
** Higher Institute of Mechanical and Electrical Engineering Department.of Automation, Sofia, Bulgaria

The object of the present report is distributed information control system (DICS). The problem of determining of optimal software components reliability with suggestion of given system hardware configuration is considered. For computing the optimal reliability parameters of components adaptive random search technique with constant values of parameters are used.

1. INTRODUCTION

The reliability assurance of technical systems is connected with additional funds, application of high technology and methods of repair and technical service.The rational utilization of funds, granted for reliability ensuring and support, is necessary due to the limited resources. As a result we can state the problem of optimal allocation of available resources so to achieve the given purpose. This problem may be reduced to the following two problems of constrainted optimization :

- to determine maximum high reliability at a preliminary set restriction of resources, given for reliability assurance,
- to achive preliminary fixed system reliability at minimum possible expenses of resources, intended to reliability assurance.

We divide the first problem into two subproblems:

- to determine the optimal number of reduntant components as simultaneously to meet the requirements of maximum high reliability and to observe some restriction of resources (cost, volume, memory, time, etc.),
- to determine reliability parameters of system components with the aim to achieve maximum high system reliability and to observe some restrictions of resources.

The problem of reliability optimization in technical system

was considered by many authors since 1956 [7] with different formulations and application of different optimization methods: linear programming, nonlinear programming, dinamic programming, integer programming, geometric programming, branch and bounds approach, variational methods, heuristic methods, simplex search, multiobjective optimization, etc. A review of the approaches could be found in [6,10].

The object of the present report is distributed information control system (DICS), which is actually a combination of hardware and software. For reliability investigation joint consideration of hardware and software is necessary, because of the impossibility to determine reliability out of the technical environment. The main source of failures in software are design errors, while the principal source of failures in hardware generaly is physical deterioration. In this paper we present the problem of determining of optimal software components reliability with suggestion of given system hardware configuration. Two cases are possible - optimization of software reliability components with fixed hardware reliability and joint optimization of software and hardware reliability parameters. In both cases it is possible to execute optimization with different by its very nature criteria (maximum system reliability, minimum cost, combined criteria, etc.). Application of various models for computing of software and hardware costs is suggested. According to consideration of system components of two kinds - software and hardware - it is possible to introduce separate resource constraints of two types. There is a good reason to introduce additional software resource constraints (execution time, operational memory). System reliability function may be found by general methods of paths and cut sets [1]. These methods allow to determine the reliability parameters of complex system (non serial-parallel structure). For computing the optimal reliability parameters of components we suggest to use adaptive random search technique based on the method of Bekey and Masri [2,5] with constant values of parameters, recommended by Pronzato et al.[8]. The global optimization algorithm does not need computation of derivatives, but only criterion evaluation. The objective function and constraints may be given in discrete form.

2. SOFTWARE IMPACT ON RELIABILITY OF DICS

In the case of software reliability evaluation of DICS we should take into consideration some peculiarity of distributed processing :

- the communication process, which determine in high degree working state of the system, have the following characteristics : complexity, parallelism and asynchronism;

- the software components are distributed in many hardware components.

Hense the system reliability depends on hardware and software component reliability, serially connected in the system reliability structure. When a redundancy in reliability assurance is used, we make decomposition in the reliability structure model by means of :

- identical to original copies of software moduls (Fig.1),

- new version of software moduls (Fig.2).

In DICS - system software copy, identical to software in failed hardware modul, may be executed without errors, because of data and time differences. The dependance between software and hardware reliability is bylateral. On application level the information processing is executed by programs, requiring input and output of extended data. As a result the state of the computer and peripheral devices influence software operation. There are many types of DICS, which differ in composition, structure and utilities. For this reason there is a great variety of mutual allocations and interconnections between software and hardware reliability moduls.

3. PROBLEM FORMULATION

Consider a complex system, consisted of two kinds of components: set of hardware compoonents $H = \{ h_1 , h_2 , \ldots h_K \}$ and set of software components $S = \{ s_1 , s_2 , \ldots s_l \}$, each having component reliability $R^h = \{R^h_1 , R^h_2 , \ldots R^h_K\}$ and $R^s = \{R^s_1 , R^s_2 , \ldots R^s_l\}$, respectively. Let construct a reliability model of this system. Consider a graph $G = (V, E)$ with vertex set $V = \{V_1, V_2 , \ldots, V_n\}$ and edge set $E = \{e_1, e_2, \ldots, e_{K+1}\}$, where $E = H \cup S$. Vertices does not fail and have fixed positions. An edge e_i, $i=1,2,\ldots,k+1$, has reliability parameter $R_i \in R^h \cup R^s$. The graph, together with the distinguished nodes V_{in} and V_{out}, define a network reliability problem - namely - calculate the probability for the nodes V_{in} and V_{out} to communicate. Different methods exist to evaluate this probability. These metods are quite general and can be used according to any complex system reliability. Most of these methods are based on minimum path sets or cut sets.

Let $A = \{ A_1 , A_2 , \ldots, A_m \}$ be set of minimum path set. The system reliability function is [1]:

$$R_{sys}(t) = P\left(\bigcup_i A_i \right) = \sum_i P(A_i) - \sum_{i<j} P(A_i, A_j) + \ldots + (-1)^{(m-1)} \cdot P(A_1, A_2, \ldots, A_m) \qquad (1)$$

To determine the optimum system and components reliability the following criteria could be used :

a) maximize the system reliability

$$\max R_{sys}(t) = f (R_1 , R_2 , \ldots, R_{K+1}) \qquad (2)$$

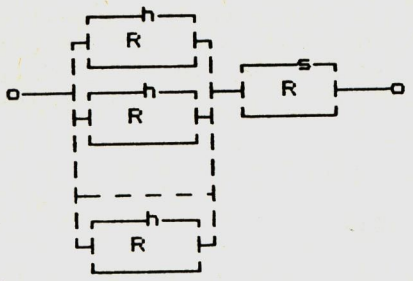


FIGURE 1

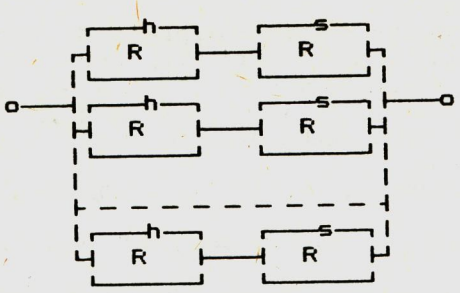


FIGURE 2

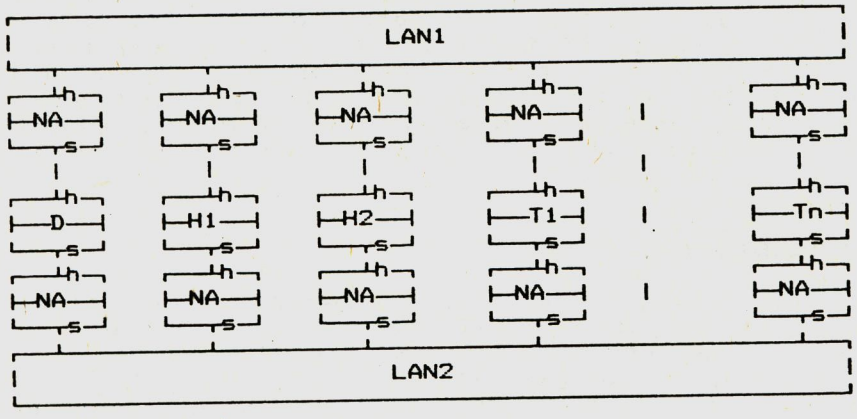


FIGURE 3

Generalized structure of the system. The elements are denoted as follows : NA-network adapter, D-Data Acquisition Subsystem, H1 and H2- pair of parallel reduntant control computers, T1...Tn-terminal (user) subsystems, LAN1 and LAN2- rings of LAN's transmission media.

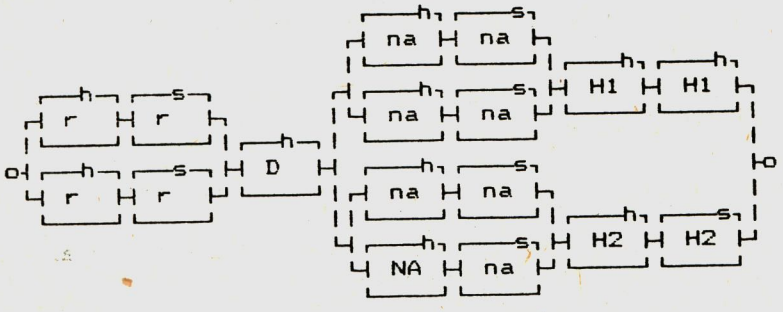


FIGURE 4

System's Reliability Structure Model.

b) minimize the total cost of the system

$$\min C_{sys} = \sum_i Ch_i + \sum_j Cs_j \qquad (3)$$

c) combined criterion [7] of the kind

$$\min Z_{sys}(t) = \sqrt{\alpha(C - C_O)^2 + \beta(R_{sys} - 1)^2} \qquad (4)$$

where α, β are weight coefficients and C_O is the lower bound for the total system cost.

Because of the limited resources for hardware and software the following restrictions must be stated :

$$\sum_i Ch_i < Ch_O \qquad (5)$$

$$\sum_j Cs_j < Cs_O \qquad (6)$$

where Ch_i, Cs_j are the costs of i- hardware and j- software components and Ch_O , Cs_O are the total costs of hardware and software.

For the system's components cost determining the following models have been applied :

a) exponential cost model [7] of the type

$$C_j = A_j \cdot \exp(B_j / (1 - R_j)) \qquad (7)$$

b) cost model [9] of the type

$$C_j = A_j \cdot R_j^{B_j} \qquad (8)$$

In view of the probability character of the variables, the following constraints exist :

$$0 < Rh_i < 1,\ i = 1,2,\ldots,k \qquad (9)$$

$$0 < Rs_j < 1,\ j = 1,2,\ldots,l \qquad (10)$$

$$0 < R_{sys} < 1 \qquad (11)$$

Thus is possible to introduce an additional software resource constraints (time, memory).

4. OPTIMIZATION ALGORITHM

To compute the optimum of the criterion chosen and the respective components reliability parameters the adaptive random search method, suggested by Bekey and Masri [2,5] has been selected, in modification with constant adaptive parameters, recommended by Pronzato et al. [8]. This global optimization algorithm does not require special properties of objective function (continuity, differentiability, etc.) and does not need computation of derivatives, but only a function

evaluation. This is necessary as the objective function may be a multimodal function. The algorithm consists of repeatedly alternating two phases. During the variance selection phase searching for global minimum algorithm randomly generate vector with multinormal distribution. Periodically the covariance of this distribution is selected among a predefined set of possible covariances by comparing the performance obtained with each of them. During the second phase (local minimum localization) the most successful variance is used for number of trials to obtain local minimum. The optimization alghoritm is described in more details in [5].

5. NUMERICAL EXAMPLE

A concrete distributed information control system (DICS), which is a kernel of computerized system for in-motion control of automobile park, is used as an illustration of our approach [11]. In the DICS the information exchange is performed in a restricted time loop through local area network (LAN). The network highway, control computers and some other devices are redundant. In this way a possibility to repair the main failed elements is provided. DICS structure is shown on Fig.3.

An approach of estimating of the lower bound of the system reliability is applied in the reliability model. Transforming of the primary system is based on the following rules :

- the failure of any element decreases system reliability or causes a system failure ;
- the assumption of absolute reliability of any system element is equivalent to its combining with another system element.

Applying these rules to the complex system from Fig.3 leads to the system's serial-parallel reliability model (Fig.4). In reliability model the system components are divided in :

- hardware components with reliability parameters

R^{h}_{r} , R^{h}_{r} , R^{h}_{D} , R^{h}_{na} , R^{h}_{na} , R^{h}_{H1} , R^{h}_{na} , R^{h}_{na} , R^{h}_{H2}

- software components with reliability parameters

R^{s}_{r} , R^{s}_{r} , R^{s}_{na} , R^{s}_{na} , R^{s}_{H1} , R^{s}_{na} , R^{s}_{na} , R^{s}_{H2} .

The system reliability function is

$$R_{sys}(t) = P^{h}_{r}.P^{s}_{r}.P^{h}_{D}.(2-P^{h}_{r}.P^{s}_{r}).(1-(P^{h}_{na}.P^{s}_{na}.(2-P^{h}_{na}.P^{s}_{na}).P^{h}_{H}.P^{s}_{H})^{2})$$

The following resources are available:

TABLE 1. Numerical results of optimization.

Case	1a	1b	2a	2b	3
R_{r}^{s}	0.9852105	0.8160225	0.9852105	0.9489335	0.9999943
R_{na}^{s}	0.7781791	0.5489071	0.7781791	0.8427139	0.9626893
R_{H}^{s}	0.9997625	0.8653896	0.9997625	0.9912577	0.9999261
R_{r}^{h}	0.9907777	0.9999380	0.9907777	0.7156768	0.9231163
R_{D}^{h}	0.9999844	0.9999973	0.9999844	0.9999201	0.9920319
R_{na}^{h}	0.9457073	0.9697820	0.9457073	0.5059475	0.9231163
R_{H}^{h}	0.9997309	0.9998957	0.9997309	0.9019762	0.9992003
R_{eye}	0.9944887	0.8647339	0.9944887	0.7533749	0.9859926
C_{r}^{s}	0.9910998	0.8851590	0.9910998	0.9690395	0.9999966
C_{na}^{s}	0.8602957	0.6977491	0.8602957	0.9024187	0.9774436
C_{H}^{s}	0.9998575	0.9169106	0.9998575	0.9947454	0.9999557
C_{r}^{h}			1.4916845	1.2272171	
C_{D}^{h}			1.4999859	1.4999280	
C_{na}^{h}			1.4505920	0.9966772	
C_{H}^{h}			1.4997578	1.4099649	
C^{s}	2.8512530	2.4998130	2.8512530	2.8662040	2.9773960
C^{h}			5.9420202	5.1337874	
C_{eye}			8.7932732	7.9999914	
Numb.of iter.	14104	14432	14104	14104	4920
Elaps. time (s)	933	1022	984	1002	266

Ch_O - for hardware,

Cs_O - for software components.

The optimization problem for this system will be to maximize the system reliability (4) with respect to constraints (5), (6) and (9), (10) , (11). For the system's component cost model we will accept (8), where a_j =1.5 , b_j =0.6 - for hardware , and a_j =1.0, b_j =0.6 -for software.

The total system cost is $C_{sys} = C^h + C^s$, where (7)

$C^h = \sum_{ii} C^h_{ii}$ - is total cost of hardware components, (8)

$C^s = \sum_{jj} C^s_{jj}$ - is total cost of the software components. (9)

The results of reliability optimization is shown in Table 1. The number of iterations and elapsed time are given. The optimization algorithm is realized in programming language FORTRAN'77 on IBM PC/AT.

Case 1.

We maximize the system reliability subject to (6) with

a) Cs_O = 3.0 respectively R_{sys} = 0.9944887 ,

b) Cs_O = 2.5 respectively R_{sys} = 0.8647339 .

Case 2.

We maximize the system reliability (6) with restrictions to total system cost

a) C_O = 9.0 respectively R_{sys} = 0.9944887 ,

b) C_O = 8.0 respectively R_{sys} = 0.7533749 .

Case 3.

We maximize the system reliability (6) with given hardware reliabilty parameters $R^h_r = R^h_{na}$ = 0.9231163, R^h_D = 0.9920319 and R^h_H = 0.9992003.

6. CONCLUSION

In the complex system, presented in this paper, system reliability is determined by software and hardware reliability. An approach of system reliability optimization by means of suitable software reliability parameter values and with separate restrictions on software and hardware is suggested.

REFERENCES

[1] Agrawal A., R.E.Barlow, A Survey of Network Reliability and Domination Theory, Oper.Research, vol.32, N:3, 1984, pp.478-492.

[2] G.A.Bekey, S.F.Masri, Random search techniques for optimization of nonlinear system with many parameters, Math. Comp.Simul., 1983, vol.25, pp.210-213.

[3] L.T.Fan, C.S.Wang, F.A.Tillman, C.L.Hwang, Optimization of systems reliability, IEEE Trans.Reliab., vol.R-16, 1967,pp.81-86.

[4] T.Inagaki, K.Inone, H.Akashi, Interactive optimization of system reliability under multiple objectives, IEEE Trans.Reliab., vol.R-27, 1978, pp.264-267.

[5] S.F.Masri, G.A.Bekey, F.B.Salford, A global optimization algorithm using adaptive random search, Appl. Math. Comp.,1980, vol.7, pp.353-375.

[6] M.Messinger, M.L.Shooman, Techniques for optimum spares allocation: a tutorial review, IEEE Trans.Reliab., vol.R-19, 1970, pp.156-166.

[7] K.B.Misra, M.D.Ljubojevic Optimal reliability design of a system - a new look, IEEE Trans.Reliab.,vol.R-22,1973, pp.255-258.

[8] L.Pronzato, E.Walter, A.Venot, J.-F.Lebruchec, A general-purpose global optimizer: implementation and applications, Math. Comp. Simul., 1984, vol.26, pp.412-422.

[9] F.A.Tillman, C.Hwang, Optimal reliability of a complex system, IEEE Trans.Reliab.,vol.R-19,1970,pp.95-100.

[10] F.A.Tillman, C.L.Hwang, W.Kuo, Optimization techniques for system reliability with redundancy - a review, IEEE Trans. Reliab., vol.R-26, 1977, pp.148-155.

[11] Tzanov V.A., S.G.Trifonova , Real-time diagnostic in distributed information system , Proceedings of the III Int. Conf.on Artifical Intelligence: Methodology, Systems and Applications (AIMSA'88), Varna,20-23 Sept.1988, North-Holland publ., pp.367-372.

O & M Experience Gained in the Introduction of ISDN
F.BLOME, A.HAMICH

At the time of going to press, the final version has not arrived yet.

SESSION 2

MESSAGE HANDLING SYSTEM AND DIRECTORIES

STANDARDS FOR MESSAGE HANDLING AND THEIR IMPACT

Walter Tietz

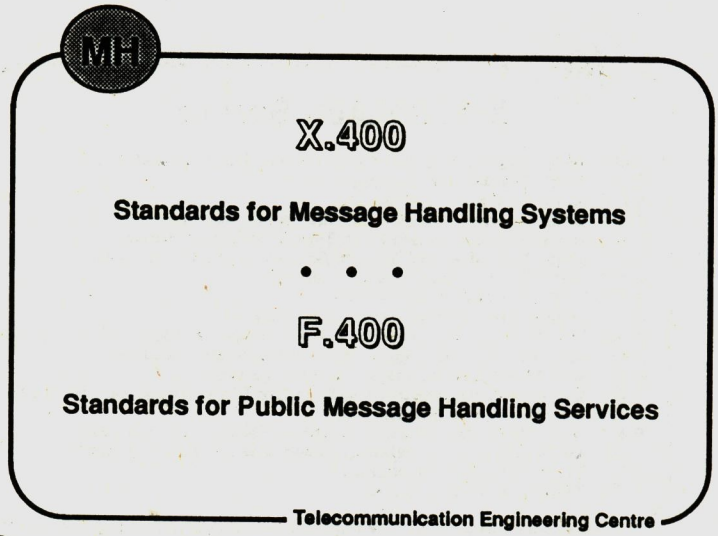
Deutsche Bundespost TELEKOM
Fernmeldetechnisches Zentralamt

Mainframe computers and dumb terminals in the first generation and intelligent terminals, workstations and personal computers in the following generations yielded a complex and in many cases incompatible mesh of system components people had to live with.

Numerous standards have been published to overcome incompatibility. The concept of Open Systems Interconnection (OSI) gave rise to new possibilities. Based on the OSI concept, Message Handling was taken up for standardization and led to a series of CCITT Recommendations titled X.400. The versions of 1984 are now accepted by industry and users and led to a new framework for a messaging infrastructure. The F.400 series in 1988 describe additional aspects of public Message Handling Services. The enhanced second generation of these standards was accepted by CCITT in 1988 in close collaboration with ISO and will pave the way for future developments.

The principles and models for Message Handling are outlined in this paper, taking into account the new field of Directories (CCITT Recommendations of the series X.500/F.500) which also requires careful attention and creates new opportunities for service providers.

An outline of public service providers for Message Handling and their involvement in joint arrangements for the interconnection of public systems and services is presented.



NOTES

Message Handling requires standardization

The X.400 series deals with Message Handling Systems, the F.400 series with Message Handling Services. Identical standards are established in cooperation with ISO.

Recommendations X.400 and F.400 are identical and are considered as the *flagship* of both series.

Furthermore, some additional Recommendations also dealing with Message Handling exist in the X.200 series and in the T series.

Many basic elements of service and optional user facilities are defined for the purpose of efficient messaging.

X.400/F.400 Series

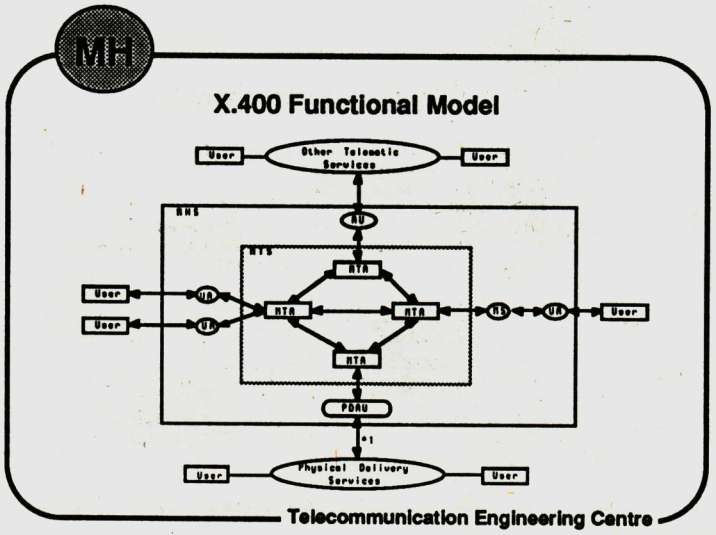
X.400 MHS: System and Service Overview (identical with F.400)
X.402 MHS: Overall Architecture
X.403 MHS: Conformance Testing
X.407 MHS: Abstract Service Definitions Conventions
X.408 MHS: Encoded Information Type Conversion Rules
X.411 MHS: MTS Abstract Service Definition and Procedures
X.413 MHS: MS Abstract Service Definition
X.419 MHS: Protocol Specifications
X.420 MHS: Interpersonal Messaging System
F.400 MH Services: System and Service Overview (identical with X.400)
F.401 MH: Naming and Addressing for Public MH Services
F.410 MH: The Public Message Transfer Service
F.415 MH: Intercommunication with Public Physical Delivery Services
F.420 MH: The Public IPM Service
F.421 MH: Intercommunication between IPM Service and Telex
F.422 MH: Intercommunication between IPM Service and Teletex
F.500 Public Directory Services
T.330 Telematic Access to IPMS

Telecommunication Engineering Centre

NOTES

Message Handling requires standardization

The Blue Books of 1988 can be ordered from the ITU in Geneva.



NOTES

Message Handling requires standardization

The functional model developed by IFIP was a start.
The version shown here is from 1988.
The Message Store and the Interworking Unit with Physical Delivery Systems are being added.

People and processes are considered to be users.

The definition of the system components conform to the architecture of Open Systems Interconnection (OSI).

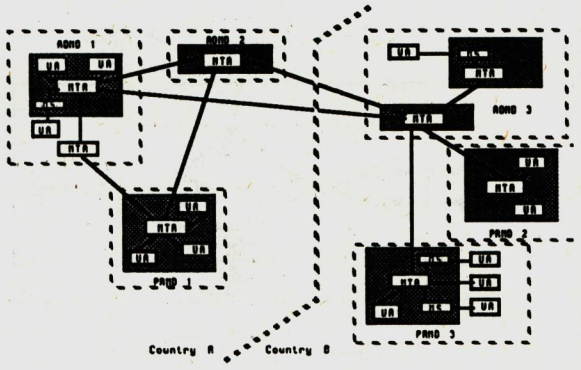
Protocols (P1, P2, P3, P7) were specified in 1988 to ensure interworking between the various system components.

Legend:

UA	User Agent
MTA	Message Transfer Agent
MS	Message Store
PDAU	Physical Delivery Access Unit
MHS	Message Handling System
MTS	Message Transfer System

MH

X.400 Management Domains



Telecommunication Engineering Centre

NOTES

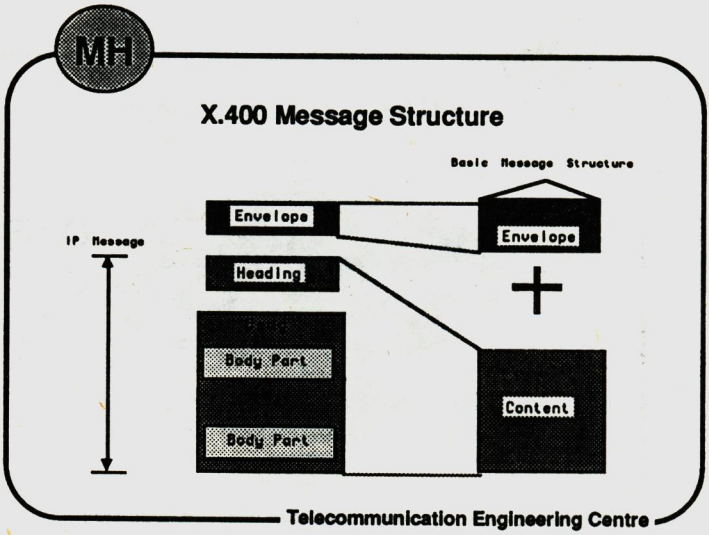
Message Handling requires standardization

Three pillars support the standardization of Message Handling:

- **The hierarchical structure**
- **The MHS naming conventions**
- **The MHS protocols**

Management Domains allow the establishment of a worldwide infrastructure for the exchange of messages.

Legend:
ADMD Administration Mangement Domain
PRMD Private Management Domain



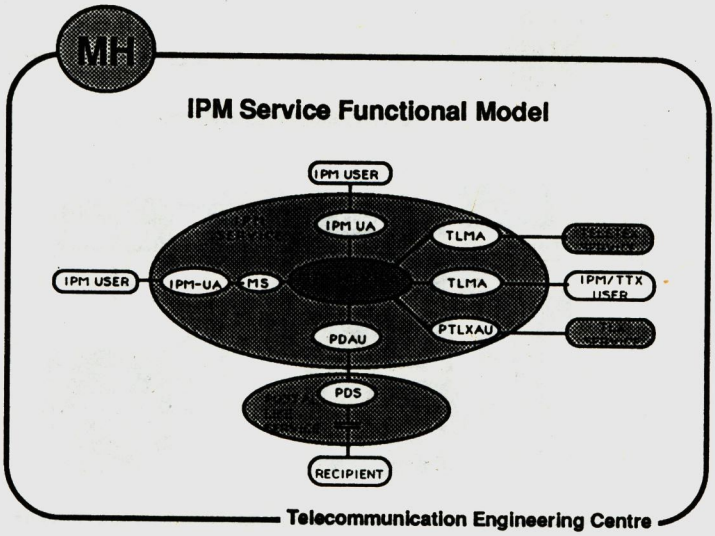
NOTES

Message Handling requires standardization

The content of a message is composed of a header and the body. The body may consist of several body parts.

The parts of a message may be differently coded. Voice belongs to the coded information types provided for in the standards.

The user surface ensures that the user can input, read and retrieve his messages in the way desired and accepted by him. No standards are provided in this respect. Manufacturers may follow up their own ideas and convince the client of their solution.



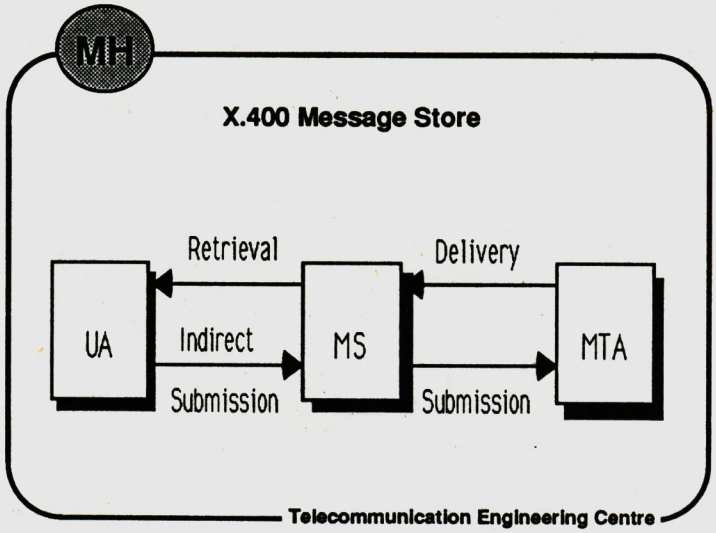
NOTES

Message Handling requires standardization

Via the MTS, various services may interact with each other. Access Units are the means allowing intercommunication.

Legend:

TLMA	Telematic Access Agent
PTLXAU	Public Telex Access Unit
PDS	Physical Delivery Sytem
IPM-UA	User Agent for Interpersonal Messaging
TTX	Teletex
MS	Message Store



NOTES

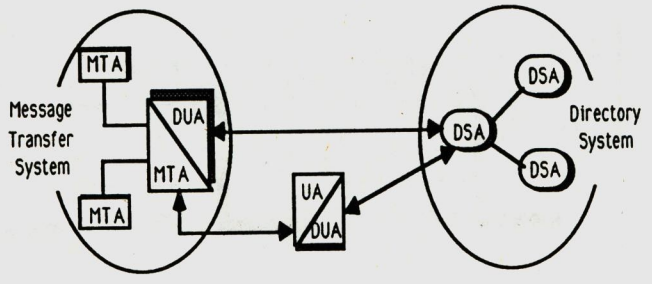
Message Handling requires standardization

Submission and delivery can also be fulfilled by a Message Store. Interactions are performed by means of the protocol P 7.

A remote UA can interact with a MTA independently if it has implemented the P 3 protocol.

MH

X.400 MHS and Directory System



Telecommunication Engineering Centre

NOTES

Message Handling requires standardization

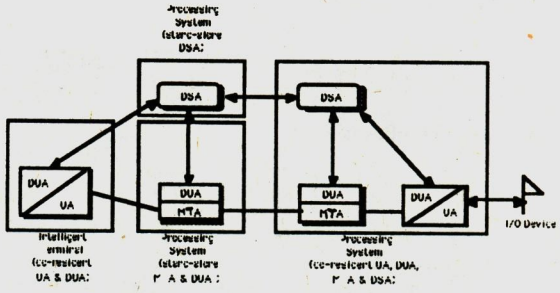
Message Handling Systems are heavily dependent on Directory Systems. The systems standardized in the X.500 series of CCITT Recommendations will contribute to a wider distribution of message handling.

Legend:

DSA Directory System Agent
DUA Directory User Agent
MTA Message Transfer Agent
UA User Agent

MH

X.400 Directory Configurations



Telecommunication Engineering Centre

NOTES

Message Handling requires standardization

It is assumed that components of Directory Systems are co-located with components of Message Handling Systems. These components can also be arranged separately and be consulted by the MHS.

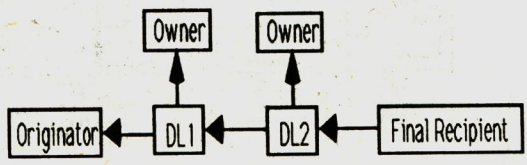
Legend:
DUA = Directory User Agent
DSA = Directory System Agent
UA = User Agent
MTA = Message Transfer Agent

MH

X.400 Distribution Lists

Sending of a message

Return of a Notification



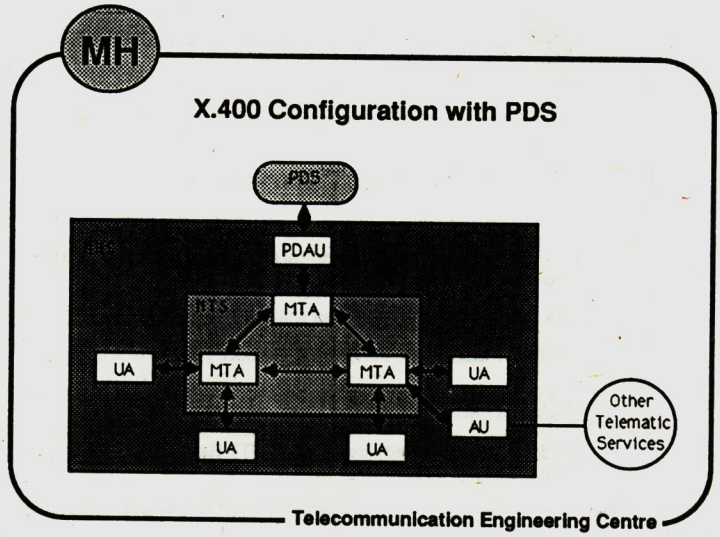
Telecommunication Engineering Centre

NOTES

Message Handling requires standardization

Distribution lists may be nested. Notifications are always sent back through the chain to the former distribution list. In a domain, certain policies apply to the treatment of the distribution lists and the sending of notifications.

Legend:
DL Distribution List



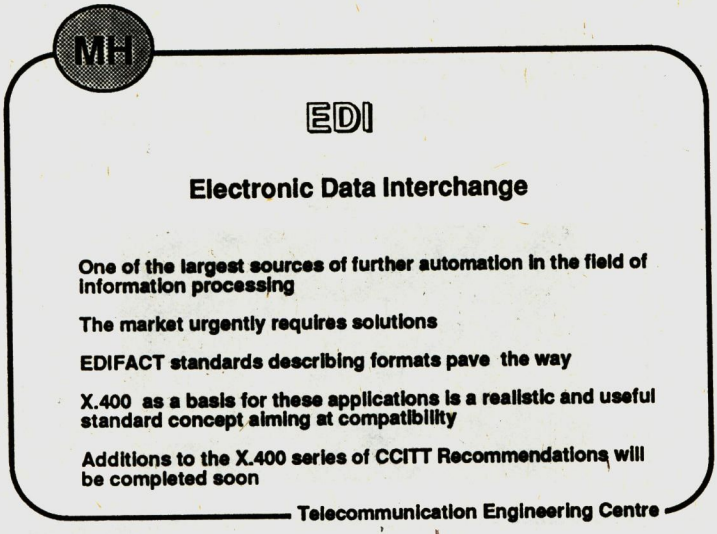
NOTES

Message Handling requires standardization

Physical Delivery by means of message handling was included in the standards in 1988.

Messages created by MH users or users of associated services can be delivered via the postal service and brought to the location of the user by a messenger.

The direction of messages to MH users and users of associated services (telematic, telex) is for further study.

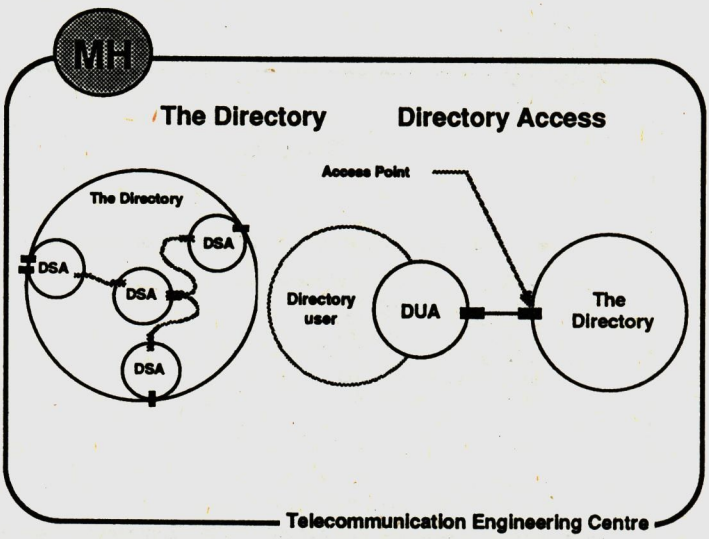


NOTES

Message Handling requires standardization

The potential of automation lies in the chaining of transactions across various applications. Assuming OSI implementations, there is a chance of compatibility between applications using different types of hardware and software.

In 1990 CCITT will issue a Recommendation concerning the use of MHS for EDI and another Recommendation concerning an EDI service in the field of Message Handling.



NOTES

Message Handling requires standardization

The X.500 series of CCITT Recommendations is another example of standards using the OSI structure.

The directory model assumes distributed systems and applications.

MH

Some Statistics

Electronic Mail forecasted for 1992 in various countries:

USA	13 000 000	mailboxes
UK	1 300 000	
Canada	650 000	
France	220 000	
Sweden	210 000	
Germany F.R.	210 000	
Italy	50 000	

Sources: Bis Mackintosh,
Eurodata, Managers Guide, Message Handling

Telecommunication Engineering Centre

NOTES

Message Handling requires standardization

Future trends should be observed closely.

MH

Number of TELEBOX Users

End of 1989 2093

Dominant Groups:

Freight Forwarding
DP Industry
Consultants
Universities
R & D
Press

Telecommunication Engineering Centre

NOTES

Message Handling requires standardization

The introduction of the pre-X.400 service went on slowly.
Various factors may have influenced
this development, e.g.:

- Existing infrastructure for text communication,
- Passive marketing because of uncertain legal conditions,
- Active competition filling the gap.

Distribution of Users in TELEBOX

Users in the Business Sector Nov. 87	%
Discrete Manufacturing	10
Process Manufacturing	1
Banking & Financial	1
Education & Research	10
Retail & Wholesale	3
Government	1
Utilities	
Air Transport	
Transportation (excl. air)	2
Insurance	
Health	
Other	72

Number of users included in the evaluation: 1 100

Telecommunication Engineering Centre

NOTES

Message Handling requires standardization

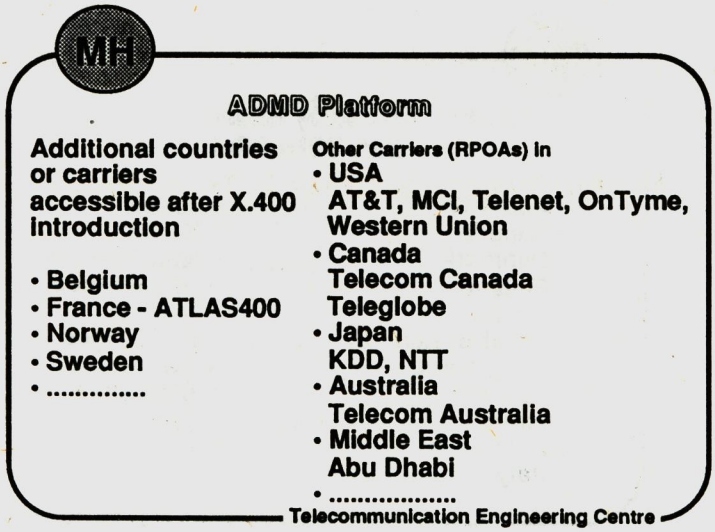
The figures are not representative as
the population is still far from having reached a critical mass.

NOTES

Message Handling requires standardization

The service providers in these countries use DIALCOM systems. The systems are interconnected with proprietary protocols.

All of them are moving towards X.400 implementations and will provide interconnections based on X.400 protocols sooner or later.



NOTES

Message Handling requires standardization

During the last CeBIT Fair in March 1990 many of these interconnections were demonstrated.

Commercial service has either started already or is in preparation.

Many Private Management Domains (PRMDs) are interconnected to the ADMDs in the respective countries. The worldwide messaging infrastructure is growing rapidly after a rather long period of implementation and acceptance of the X.400 standards.
The next step is to adapt the protocols to the 1988 versions and to expand coverage.

IAOG

International ADMD Operators Group

Goals

Promotion of Services based on F.400 / X.400

Establishment of International Infrastructure

Discuss and Prepare the Necessary Framework

Telecommunication Engineering Centre

NOTES

Message Handling requires standardization

On 20th April 1988, 25 organizations founded a group in Regensdorf near Zürich for the purpose of urging the implementation and offering of Message Handling Services based on X.400/F.400.

Other organizations (14) joined the activity afterwards. The last meeting took place at the end of May 1989 in Sidney, Australia.

IAOG Guidelines have been worked out and are being used as an input for further work in CCITT.

EXTENSIONS AND EVALUATION OF METHOD FOR MHS INTERWORK

YUKIO SAITO, MASAHIRO TERAMOTO and KAZUNORI UEGAKI

NTT Network Systems Development Center
Nippon Telegraph and Telephone Corporation (NTT)
Shinjuku-Ku, Tokyo, Japan

Nippon Telegraph and Telephone Corporation (NTT) started developing the NTT-MHS system soon after CCITT issued its X.400 recommendations in 1984. Interworking between this system and other MHSs has been confirmed through national and international events such as TELECOM'87 (1987, Geneva), INE'88 (1988, Tokyo) and '89 Telecom Japan (1989, Tokyo).
This paper describes protocol extensions that have been considered for the Japanese interworking environment and evaluates the method used for above interworking tests.

1. Introduction

Nippon Telegraph and Telephone Corporation (NTT) started developing the NTT-MHS applied to a public communication network facility soon after CCITT issued the X.400 recommendations in 1984 [1]-[4]. NTT is now enhancing the MHS to establish interfaces to ISDN networks called INS64. Interworking between this system and other MHS systems has been confirmed through national and international events such as TELECOM'87 (October 1987, Geneva), INE'88 (November 1988 , Tokyo) and '89 Telecom Japan (June 1989, Tokyo). An MHS system has been introduced to VAN (Value Added Network) companies and enterprises, to improve communications services including EDI services in Japan. Moreover, some MHS systems already provide G3FAX or telex delivery services linked with MHS services. These environments will require protocol extensions unified into a functional standard in Japan, at the same time they must forward MHS interworking on a future commercial basis.

This paper describes protocol extensions that have been considered for the Japanese interworking environment and evaluates the method used for above interworking tests.

2. Protocol Extensions Considered for The Japanese Interworking Environment

2.1 ADMD/PRMD Names

Within each country, ADMD/PRMD names should be managed by administrative authority so as to promote efficient MHS interworking. In Japan, the official publication containing the rules for registration of ADMD/PRMD names was released last July. It stated that domain names should be approved and designated by the Ministry of Post and Telecommunications (MPT), to prevent similar or duplicate domain names.

This rule was provisionally applied to '89 Telecom Japan and preceding interworking tests, and the following practical problems resulted regarding domain name characters. In addition to alphanumeric characters and spaces, a domain name may include graphic symbols, such as "-", because the rule states that a domain name shall consist of graphic characters of a PrintableString type defined by ASN.1 (Abstract Syntax Notation One). However, it was revealed that some systems could not recognize or would mistakenly recognize a domain name containing symbols, since these systems use symbols as command parameter delimiters for their man-machine interface. Because of this, all domain names used in the tests were restricted to combinations of letters and numbers. It is assumed that MPT will give relevant guidance to companies during the registration of their domain names. But, from an international point of view, it is necessary for every system to provide a mechanism for recognizing domain names correctly on man-machine interfaces, for example, a local directory translating a name without symbols into an actual domain name.

2.2 G3FAX and Telex Delivery

Some MHS systems already provide G3FAX or telex delivery services linked with MHS services. Moreover, the increased use of word processing software on PCs (Personal Computers) in Japan has allowed users to create complicated documents containing diagrams and images. However, Usage of the ORName elements for these services is not unified in Japan and existing G3FAX delivery services support only simple text containing only the carriage return and line feed control characters. Therefore, Unification of the ORName element usage for delivery of more complicated documents by G3FAX is a current requirement in Japan. The method shown in the following sections can satisfy this requirement. This method conforms to the X.400 recommendations and the international ADMD Operator's Group (IAOG) guidelines [5]. NTT is proposing this method to the Telecommunications Technology Committee (TTC) as a functional standard in Japan.

2.2.1 ORName Elements

The ORName elements used and their classification are shown in Figure 2.2.1-1. This form corresponds to Form1/Variant3 address of X.400 (1984) and Terminal O/R Address of X.400 (1988). A terminal-type element and an extendedNetworkAddress element were not defined in X.400 (1984).

Elements of ORName	Class (See Figure 3.3.2-3)	
	1984 Version	1988 Version
CountryName	M	
AdministrationDomainName	M	
PrivateDomainName	G	
NetworkAddress	M	
TerminalIdentifier	H	
terminal-type	not defined	H
extendedNetworkAddress	not defined	H
DomainDefinedAttributes	G	

Figure 2.2.1-1 ORName Elements for G3FAX and Telex Delivery

2.2.2 Setting Network Addresses

The network address of a G3FAX terminal and a telex terminal is set to an X121Address element in a NetworkAddress element. Usage of an X121Address element is as follows:

For telex, 8 + F.69 (= X.121).
For G3FAX, 9 + E.163 (= X.121).

In the latter case, since a terminal connected to a public switched telephone network (PSTN) is not always a G3FAX terminal, the rule in the following section should be required.

2.2.3 Service Method of Recognition

(1) For interworking between systems conforming to the 1984 X.400 standards and systems conforming to the 1988 X.400 standards, it shall be recommended that a DomainDefinedAttributes (DDA) element be used to distinguish the service requested, for example, G3FAX delivery or telex delivery, and that the value of a DDA element be a local matter. Additionally, it is also possible to designate in the address the kind of word processing software being used.

(2) If no terminal type information is provided in the address, the default type agreed bilaterally is assumed. For example, if the elements of a CountyName, an AdministrationDomainName and a NetworkAddress(X121Address) specified by "9 + E.163" only exist in the address, it means the request of G3FAX delivery.

(3) For interworking between systems conforming to the 1988 X.400 standards, the extension attribute of a terminal-type or an

extendedNetworkAddress for ISDN may be used in the address.

2.3 Use of Japanese Kanji Characters over X.400 Protocols

There are several standardized Japanese kanji codes such as JIS X0208, Shift JIS. In Japan, it is essential in developing an MHS network that messages including these codes can be transferred between MHS systems. Since the CCITT recommendations do not state how these messages shall be exchanged, domestic organizations, INTAP (The Interoperability Technology Association for Information Processing) and TTC, developed Japanese functional standards for MHS, respectively. However, because usage of protocol elements such as EncodedInformationTypes and BodyParts for transferring kanji messages is different, it is impossible to exchange kanji messages between two systems which implement these different functional standards without a conversion facility.

Therefore, companies which took part in '89 Telecom Japan were divided into two groups based on the implemented functional standard, and the demonstration was held with systems within the same group.

Consequently, for interworking, it must be possible for any conversion facility to be implement in both systems. Or it is necessary that , in future, usage of protocol elements for transferring kanji messages must be unified for higher interoperability.

2.4 Transfer of EDI messages

CCITT is studying EDI message transfer over the 1988 X.400 protocols. This protocol is called Pedi. On the other hand, with the 1984 X.400 protocols, the P0 method for setting EDI messages to the P1 Content and the P2 method for setting EDI messages to the P2 BodyParts are recognized and used internationally. The P2 method is easier to implement than the P0 method because changing the configuration of existing mail box is unnecessary. It is necessary in these cases to define how to designate the EDI syntax used, for example, ANSI X12, EDIFACT, or TDI. In Japan, TTC is studying EDI message transfer over the 1984 X.400 protocols, including the designation of domestic EDI syntax.

3. Evaluation of The Test Method

3.1 Test Steps

NTT adopted the following test steps to confirm technical capability.

(1) Step 1 - Exchange the following documents.
- product name and the software version.
- Protocol Implementation Conformance Statement (PICS) defined in X.403(1988).
- system conformance parameters.

- conformance certifications, if available.
- statement of conformance to relevant functional standards including any limitation.
- values of the interconnection parameters for OSI layers 3 to 7.
  Examples of Connection Data:
  X.121 DTE address, TSAP address, SSAP address, ADMD Name, MTA Name, Password, ASN.1 Coding, ORName for test, etc.
- test items.
- access method for a partner (UA addresses and ORNames).

(2) Step 2 - Check the static connectivity in the above documents.
(3) Step 3 - Register the connection data in the system.
(4) Step 4 - Test in on-line environment.
(5) Step 5 - Exchange and discuss test results.

If problems are found at individual steps, appropriate repair actions may be taken. Thorough execution of especially Step 2 makes possible, to a certain extent, the examination of connectivity with a partner before testing and the discovery of interworking problems as early as possible. As a result, testing will progress more smoothly.

CCITT and various international organizations are studying and developing conformance test methods and tools. The objective of these organizations is to validate implemented protocols so as to claim conformance to the CCITT recommendations or functional standards. Consequently, above conformance tests is not sufficient in case of starting an interwork bilaterally on a commercial basis, the above steps shall be required.

3.2 Additional Items for Static Connectivity Check before Testing

The connectivity check on X.400 services and protocol specifications is examined by a PICS defined in X.403 [6]. Moreover, we added several items to be checked to the IAOG guidelines. These items are not indicated by a PICS and depend on implementation. The following are examples.

- Can your system decode its partner's session user data encoded by the EOC length form?
- Does your system send the session user data of a CN SPDU encoded by the EOC length form?
- Which validation method does your system use, a NULL data or an MTA Name and Password?
- Can your system accept an MIP SPDU before achieving the negotiated check point size?
- Does your system send an MIP SPDU like above?
- Can your system handle protocol elements encoded by non-minimum length form?
- Does your system send protocol elements encoded like above?

At INE'88 and '89 Telecom Japan, each participating company examined the above items and was able to determine at an early stage whether its own system had any technical problems that would prevent proceeding to the interworking tests.

3.3 Connectivity Judgment on Classification of MHS Protocol Elements

Organizations promoting the OSI standards, like SPAG in Europe, NIST in the U.S.A, TTC and INTAP in Japan, provide their MHS functional standards to promote smooth MHS interconnection. They classify the treatment of protocol elements and give it a class symbol like M, R, G, H or X. This section describes protocol subsetting considerations and classification and connectivity judgments using such a classification.

3.3.1 Subsetting Protocol

The relations between the X.400 recommendations, functional standards and implementation specifications are shown in Figure 3.3.1-1. The following should be noted because each functional standard shows only the common specification to companies within each organization.

(1) They do not indicate handling of protocol elements of local matters.

(2) Where the implementation specification is upward compatible with its original functional standard, the higher level in each system can not be seen from the outside.

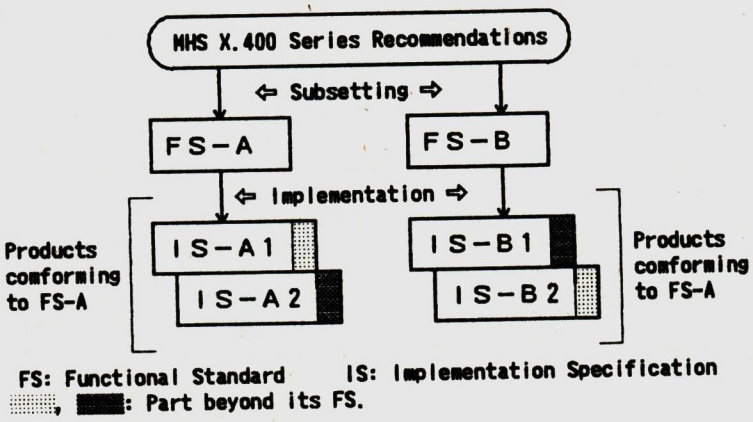


Figure 3.3.1-1 Relations between the X.400 Recommendations, Functional Standards and Implemantation Specifications

3.3.2 Classification Considerations

The method of classifying protocol elements can be arranged as in Figure 3.3.2-1. Figure 3.3.2-2, an example of Method 1, is adopted in X.403 Annex B. Figure 3.3.2-3, an example of Method 2, is adopted in functional standards such as SPAG, NIST, TTC, and INTAP. Generally, either of two methods is used to designate implementation specification of a system. Figure 3.3.2-4 shows the relationship between these methods and this will be very useful in judging connectivity and simplifying translation between methods.

Items	Method 1	Method 2
Base	System	Protocol Element
Feature	(1) Support is indicated by the implementation on origination and that on reception separately. (2) It is easy to understand the protocol handling on origination and reception.	(1) Support is indicated by the combination of protocol handling on origination and reception per element. (2) It is difficult to give it a class directly. (3) It is easy to compare betwee between FSs or ISs.
Range	Suitable for ISs because of its ease of undersatnding.	Suitable for FSs because of its ease of comparison.
Example	X.403(1988)	The FSs of SPAG, TTC, INTAP, etc

FS: Functional Standard IS: Implementation Specification

Figure 3.3.2-1 Classification Method

Class	Origination	Reception
X	These elements are not generated.	These elements are not supported. The absence of these elements should not be considered as a protocol violation.
S	These elements may be generated under some conditions.	These elements are supported. The absence of these elements should not be considered as a protocol violation.
M	These elements must always be generated, because they are required by X.400.	Appropriate action must be taken. The absence of these elemenmts should be considered as a protocol violation.

Figure 3.3.2-2 Example of Method 1

Origination / Reception		Method 1 M	Method 1 S	Method 1 X
Method 1	M	M	R	R
	S	+	G	X
	X	—	H	X

- : not aplicable inside of ···· : Method 2

Figure 3.3.2-4 Relationship Between Method 1 and Method 2

Class	Originating MD	Relaying MD	Delivering MD
X	Implementations are not required to generate these elements.	They must be relayed transparently	Implementations are not required.
H		Appropriate action must be taken. The absence of these elements should not be considered as a protocol violation.	Appropriate action must be taken. The absence of these elements should not be considered as a protocol violation.
G	These elements must be generated in some condition, e.g. by the command of a user, because they are optional in X.400.		
R	These elements must always be generated, although they are not required by X.400.		
M	These elements must always be generated, because they are required by X.400.	Appropriate action must be taken. The absence of these elements should be considered as a protocol violation.	Appropriate action must be taken. The absence of these elemenmts should be considered as a protocol violation.

Figure 3.3.2-3 Example of Method 2

3.3.3 Connectivity Judgment

Connectivity between systems can be judged by comparing a related class symbol of each protocol element. This section shows the table which indicates how each method judges connectivity under the following conditions.

B-Reception / A-Origination	M	S	X
M	C	C	C
S	N	C	C
X	N	C	C

C: Can connect
N: Can not connect

Figure 3.3.3-1 Connectivity Judgment of Method 1

(1) Two systems are considered to have connectivity when they can communicate with each other without a protocol error even if required services are not always performed.

(2) Two systems, A and B, for connectivity judgment are interconnected.

The tables for Method 1 and Method 2 are shown in Figures 3.3.3-1 and 3.3.3-2, respectively. The latter table can be derived from the former. In the table for Method 2, it is understood that the absence of a combination of class M and either class G , H or X for every protocol element of each system causes no interworking problems.

System A \ System B		M	O=M R=M	R	O=M R=SX	G	O=S R=S	H	O=X R=S	X	O=SX R=X
M	O=M R=M	C		C		N		N		N	
R	O=M R=SX	C		C		C		C		C	
G	O=S R=S	N		C		C		C		C	
H	O=X R=S	N		C		C		C		C	
X	O=SX R=X	N		C		C		C		C	

O : Origination
R : Reception

C : Can connect
N : Can not connect

Method 2	Method 1

Mapping of Method 1 and Method 2 is shown in Figure 3.3.2-4

Figure 3.3.3-2 Connectivity Judgment of Method 2

4. Conclusion

This paper describes protocol extensions that have been considered for the Japanese interworking environment and evaluates the method used for interworking tests. The following points were presented.

(1) It is necessary for every MHS system to provide a mechanism for the correct recognition of a domain name on a man-machine interface.

(2) For interworking between systems conforming to the 1984 X.400 standards and systems conforming to the 1988 X.400 standards, it is recommended that a DDA element be used to distinguish G3FAX and telex delivery services. As a result, it is also possible to designate in the address the word processing software used.

(3) In the future, usage of protocol elements for transferring kanji messages must be unified for higher interoperability in Japan, or it must be possible for any conversion facility to be implement in both systems.

(4) The P0 method and the P2 method are recognized for transfer of EDI messages over the 1984 X.400 protocols. The P2 method is easier to implement than the P0 method because changing the configuration of the existing mail box is unnecessary.

(5) The objective of a conformance test is to validate implemented

protocols so as to claim conformance to the CCITT recommendations or functional standards. Consequently, when starting an interwork bilaterally on a commercial basis, the test steps and additional questions shown in this paper shall be required.

(6) The Connectivity Judgment tables shown in this paper are very useful in checking the static connectivity.

ACKNOWLEDGMENTS

The authors would like to thank their many colleagues in the Packet Switching Systems Project Group. They particularly wish to express their gratitude for the guidance and encouragement received from Mr. Masayuki Nomura.

REFERENCES

[1] CCITT Recommendations X.400 (1984), Red Books.
[2] T. Nakayama and M. Nomura, Developments of MHS in NTT - Connection of Personal Computers to MHS -, ICCC'86, pp.61-66, 1986.
[3] A. Iwabuchi, Y. Shimazu, M. Nomura, and T. Itoh, Message Communication Network Services for Personal Computers, ICCC'86 pp.248-253, 1986.
[4] T. Soga, S. Fujiwara, and T. Kawahara, NTT-PCN: Development and Operation, ICCC'88, pp.533-537, 1988.
[5] IAOG GUIDELINES FOR ACCOUNTING, SERVICE AND OPERATIONAL ASPECTS FOR INTERNATIONAL MESSAGE HANDLING SERVICES PERFORMED VIA ADMINISTRATION MANAGEMENT DOMAINS, 1989.
[6] CCITT Recommendation X.400 (1988), Blue Books.

APPROACHES TO THE MHS MESSAGE STORE STRUCTURES

Valentin SABEV*, Frank FOERSTERLING**, Norbert KLEHN**

(*) Bulgarian Academy of Sciences
Centre of Informatics and Computing Technology
"Acad. G. Bontchev" Street, 25A
Sofia 1113, Bulgaria

(**) Academy of Sciences of the GDR
Institute of Informatics and Computing Technique
Rudower Chaussee 5
Berlin, GDR - 1199

In 1988, CCITT published the new recommendations of the X.400 series of Message Handling Systems (MHS), which are fully aligned with the ISO activities on Message Oriented Text Interchange Systems. The Message Store (MS) introduced is perhaps one of the most essential extensions of the functional MHS model compared with that worked out in 1984. This paper investigates several modes of a MS structure.

1. INTRODUCTION

Communication in its essence is the interaction aimed at exchanging a pieces of information among objects. It takes place in time and space as each process. Two types of communication are observed: information transfer and information archiving (recording) [4]. When the information is transferred, it is sent from a given point in space. Meanwhile it is stored in the information channel for a certain period of time during its transmission. When archived (recorded) the information does not change its location. It is stored from one period of time until another one (the coming one).

In order to improve the people communication CCITT worked out the new recommendation of the X.400 series of Message Handling Systems (MHS) [1], which was published in 1984. In the MHS model described in this recommendation special emphasis has been placed upon the information transfer in contrast to the information archiving especially that related to the end users which has almost not been dealt with. That shortcoming was avoided with the next edition of the X.400 recommendation published in 1988. For that purpose a new functional component known under the name of Message Store (MS) is introduced.

In order to distinguish the separate editions of the recommendations from 1984 and 1988 we are going to denote them by X.400(84) and X.400(88) respectively.

The present paper studies with a study of MS and meanwhile new ideas (modes) of their computer building up are given. The principles observed in our study are as follows:

- chronology. The study made follows the development of the MHS, especially some new ideas suggested by a number of researchers and developers which concern the adding of the store component to the MHS.

- location of computing tasks. It is well known that more than one computing tasks may be located in a computer node, i.e. the node may operate in a multitasking mode. Also it is quite possible a certain computing task to be distributed (located) in several nodes, i.e. the case of a distributed functional component is observed as well.

- concentration. Communication in its function is one of the necessary preconditions of every group (human) activity. It is well known that in case of a greater "density" with respect to the distribution of human activities (for example with offices) the Local Area Networks (LANs) are preferred. That is why we have given preference to the relation between a LAN and a MS in our study.

After a short overview of the X.400 MHS model in section 2, we will consider the physical configurations in section 3. Section 4 deals with the discussion of several modes of the MS structure.

2. THE MESSAGE STORE WITHIN THE MHS MODEL

This is a brief description of the MHS model used in the CCITT X.400 recommendations [1]. The main functional components of the model are shown in the fig. 1.

Message Handling Systems and Services enable users to exchange messages on a store-and-forward basis. The principal component of the MHS is the Message Transfer System (MTS) which consists of several Message Transfer Agents (MTA). A message submitted by one user, the originator, is conveyed by the MTS and subsequently delivered to one or more users, the recipients. User Agents (UA) help users to access the MHS. Message Stores (MS) provide storage for incoming messages and enable their submission, retrieval and management.

The MS plays an intermediary role between the UA and the MTA. One MS acts on behalf of one user only. It stores delivered messages and retains them for a subsequent retrieval by the user's UA. Messages received from the UA for submission are forwarded transparently to the MTA.

The MS maintains an information base containing the so-called entries. An entry represents a message and further constituent parts describing the message. The main operations of the MS are as follows:

- SUMMARIZE returns the summary counts of selected entries from the information base to the UA.
- LIST is used to search the information base for entries of interest and to return the selected information from those entries to the UA.
- FETCH is used to return the selected information from a specific entry in the information base to the UA.
- DELETE is used by the UA to delete selected entries from the information base.
- REGISTER-MS is used by the UA to register or deregister various pieces of information within the MS.
- ALERT enables the MS to inform immediately the UA about a new entry included in the MS.

The MS is perhaps one of the most essential extensions of the MHS model compared with those from 1984. The following reasons for introducing this new functional entity are identified:

- The UA can be implemented on a wide range of computers, including PC. These PC-UA are not available at any time because they are used for other applications as well.
- There is not enough space for a data base (mailboxes) on a PC.
- The user is able to access his mailbox from terminals at different locations (such as home, offices, hotels, etc.) by means of the MS.
- The MTA delivers messages to the UA in the sequence they arrive at the MTA. The MS allows the retrieval of messages as received.

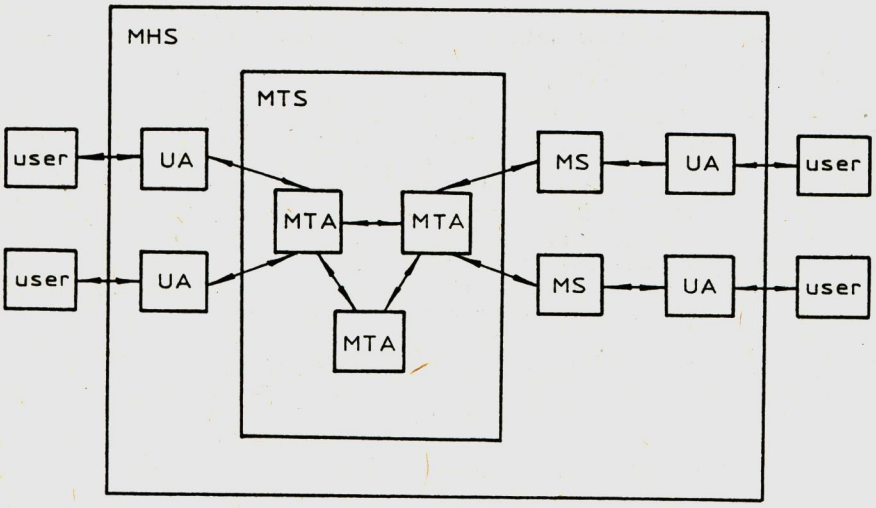


Fig. 1. MHS functional model

3. PHYSICAL CONFIGURATIONS

We have to consider several architectures of end-systems (fig. 2) connected to a wide area X.25 packet switched network (WAN). These architectures may be either LAN-oriented (fig. 2, item I and II) or Terminal-emulation-oriented (fig. 2, item III and IV).

The LAN-oriented architectures differ from each other in their topolology. Two of the possible LAN topologies are shown in the fig.2. It is bus-like (fig. 2, item I) or star-like (fig. 2, item II) respectively. In both cases, the router computer simultaneously acts as a LAN X.25 gateway and as a WAN host.

The architectures (fig. 2, item III and IV) belong to the Terminal-emulation-oriented ones. The third architecture shows a "stand-alone" host connected to a WAN, and "stand-alone" PCs connected to a WAN via a terminal concentrator, a so-called terminal communication controller (TCC) [3]. Such an architecture is usually used, if the host is under control of the PTT, and the PCs are located at the users'. It may happen, that for small organizations the TCC part of the architecture (fig. 2, item III) solves their communication task regarding MHS (fig. 2, item IV).

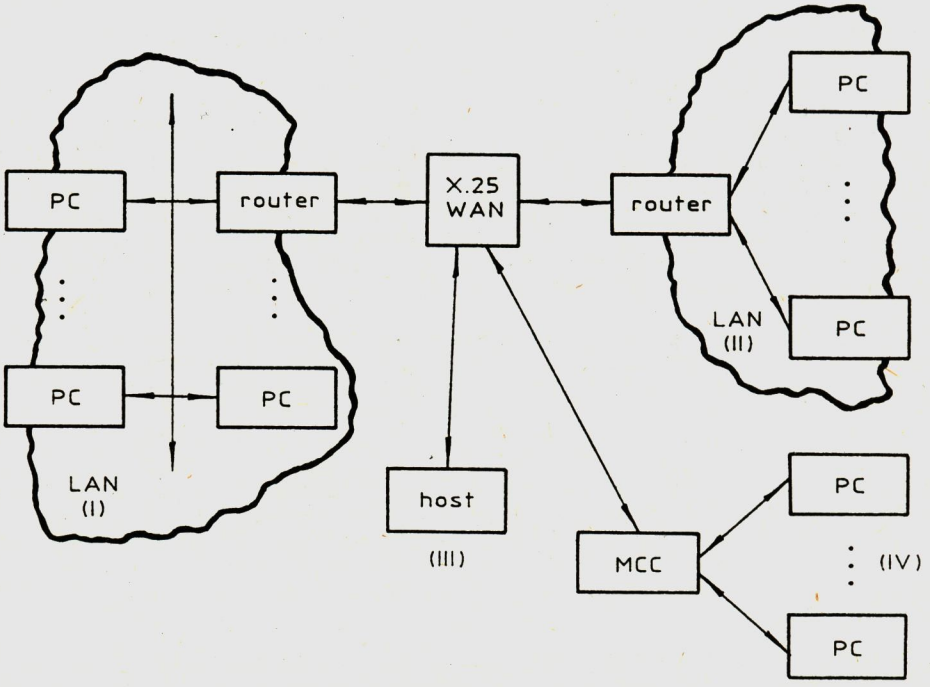


Fig. 2. Physical configuration of the end-SYSTEMS

The router of fig. 2 (item I) represents a XT-compatible 16-bit-computer or a 32-bit-computer. The router of fig. 2 (item II) and a host of fig. 2 (items III and IV) are 32-bit-computers. The PCs are XT-compatible 16-bit-computers.

Following the concentration principles chosen by us we are going to deal with the LAN-oriented MS.

Having in mind the various architectures of the end-systems it is important to find a solution of the functional MHS component independent of the underlying architecture. The best way to fulfil that task is to use the ideas of Open Systems Interconnection (OSI) [2,5]. The functional components of the MHS are located in the application process outside the application layer. In this sense, we are able to realize every functional MHS component with a strong standardized interface without any knowledge of the kind of the physical interconnection to its adjacent functional MHS component.

On the other hand, for reasons of the world-wide interconnection the communication through a WAN will be realized by means of the whole OSI protocol stack. Therefore the routers have to realize the OSI layer 1-7. Furthermore, the host which contains at least the MTA functional component has to realize a P1-protocol.

4. CONSIDERATIONS ON THE STRUCTURING OF THE MESSAGE STORE

Now we are able to investigate several modes for the locating of the MS in the end-system in connection with respect to its structure. The modes suggested which involve components for long-term message store or similar to them are called MS Servers in this paper. In case of a star-like topology it is appropriate to locate MS Server in the router. In case of a bus-like topology the following solutions may be suggested: a MS Server in a router if the latter is a 32-bit computer or outside the router if the latter is a 16-bit computer .

The underlying architectures may be neglected. Due to the location of the MHS components on different computers, it may be expected an influence to the speed of the MHS only.

4.1. Distributed User Agent

In order to avoid the shortcomings related to the message archiving for a long period of time ECMA published the mailbox service standard [8]. In fact this standard is an extension of the ECMA-93 MIDA [7] as well as of X.400(84). The extension is obtained by introducing the Mailbox Server and the Mailbox Client.

Having accepted the ECMA approach to the Mailbox Service we have given the first mode of storing messages for a longer period of time on fig.3. The MS Server consists of a Mailbox Server and a Message Carrier.

The Mailbox Server component is used for a long-term message store. This component consists of a set of Mailboxes and a register of Mailbox users. In order to ensure the communication with an e-mail system in particular with the MHS the Message Carrier operates following the X.400(88) recommendation. This component uses the P3 protocol in order to ensure the communication between the Mailbox Server and MTA of the MHS. In this way the Mailbox Server and the Message Carrier appear to be a UA with respect to MTS.

The (ECMA) P7 protocol ensures the communications between the Mailbox Client and the Mailbox Server. The main operation of the protocol are similar to those shown in section 2. The UA is distributed because the Mailbox Client and the Mailbox Server are separate objects. The support of various (different) access points of the P3 protocol is organized so that all the LAN users to be maintained by MTS.

The advantage of using a distributed UA is that the ready-made Software and Hardware are retained within the LAN, so that our efforts should be entirely directed to the creating of the Message Carrier. Moreover Submission Delivery Entity of MHS X.400(84) may be used as a Message Carrier. Another advantage is the attaching of the MS Server to a LAN. The high speed inside a LAN allows the requests between the Mailbox Client and the Mailbox Server to be transmitted in a necessary period.

However the disadvantage will be that the MHS X.400(88) model is not observed. Also the MS Server should always be ready for operation. Another disadvantage is that the communication will be provided by the ECMA P3 protocol which at this stage might cause some difficulties when using the two MHS versions, namely - X.400(84) and X.400(88).

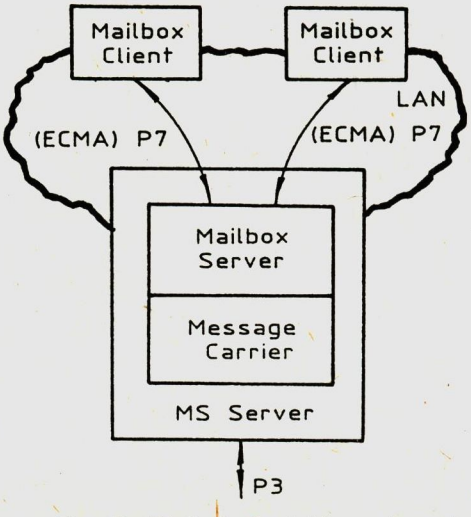


Fig. 3. Distributed User Agent

4.2. Multi Message Store

The second mode suggested (fig.4.a) follows entirely the MHS X.400(88) model. The MTA, MS and UA components correspond to those given by the recommendation from 1988 and mentioned briefly in section 2.

In this case the MS Server comprises a set of MS. Each MS is linked with a separate UA. The communication between these UA and MS is realized by means of the P7 protocol from the X.400(88) and the communication between MS and MTA from the MTS - the P3 protocol from the X.400(88).

This mode is similar to the first one (distributed UA) as regards to its structure. That is why we may conclude that it will show the same advantages and disadvantages.

However we have to point out that a number of MS instances are available in the MS Server. They have identical functions. For that reason many authors speak about a Multiuser MS [6]. We do insist on that idea. Meanwhile we think that there is no essential difference between the case of locating more than one MS in a computer and the case of one Multiuser MS. That follows from the fact that common parts of MS might be developed as a reentry program and that different entry points might be supported for the P3 and P7 protocols. Such a Multiuser MS is given in fig.4.b.

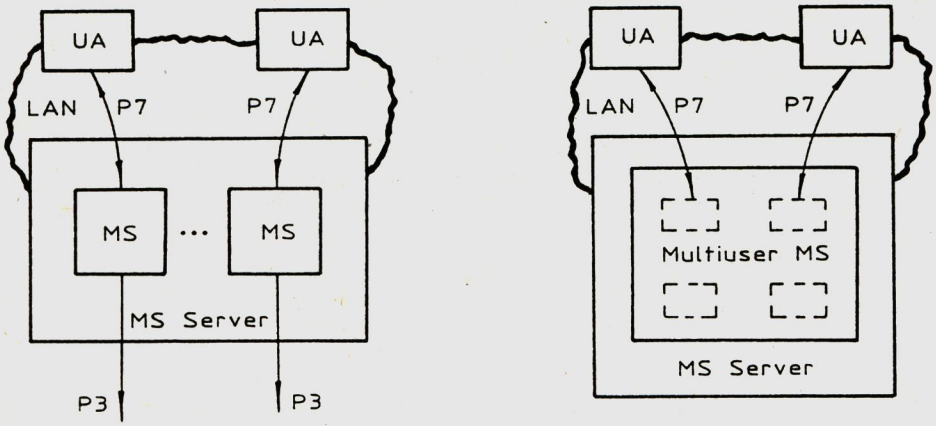


Fig. 4. Multi MS

4.3. Classical Message Store within ST/SYS

The MS structure mode proposed in this subsection coincides entirely with the requirements and the MHS configurations [1]. That is why we call it classical.

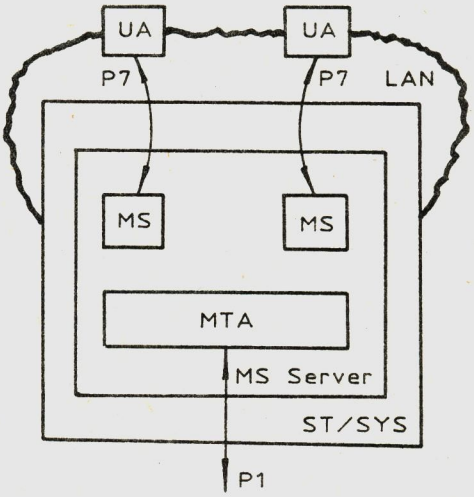


Fig. 5. MS inside ST/SYS

Here the MS Server consists of one MTA and several MS (fig.5). That means that it is both a storage and a transfer system (ST/SYS) in terms of X.400(88). A MS is envisaged for each UA. The MS and the UA communicate through a P7 protocol.

ST/SYS communicates with the remaining components of the MHS through a P1 protocol. And as it is well known it does not hamper the operation of the MHSs created after the X.400(84) and X.400(88) editions.

The fact that ST/SYS comprises a MTA lets us conclude that a MHS is available within the LAN which is a very good and positive LAN service. ST/SYS should be located in a 32-bit computer or a LAN dedicated station so that to ensure a good MHS.

4.4. Distributed Message Store

As we have already mentioned in section 2, the MS appears to be a link between the UA and the MTA. These functional components communicate through the P3 and P7 protocols respectively. Naturally the problem of dividing the MS into two (or more) parts was posed, namely - the first one aiming at communicating with the UA and the other one aiming at communicating with the MTA. This type of MS is called here a distributed MS.

A type of a MHS comprising a distributed MS is shown in fig.6. MTA and UA are entirely compatible with the X.400(88) recommendation. The distributed MS consists of the Message Keeper, Transfer Connector and Transfer Ensurer subcomponents which if connected appropriately appear to be a functional component intended for the MS. The subcomponents mentioned are characterized by the properties as follows:

- The Message Keeper subcomponent provides the messaging of the LAN UA through the P7 protocol. Since the operations of this protocol are rather peculiar it is very convenient to manage the data base of the messages stored by this subcomponent.

- The Transfer Connector subcomponent provides the communication with the MTA through the P3 protocol. It is appropriate to locate it in the host that contains MTA.

- The Transfer Ensurer subcomponent is aimed at ensuring the remote connection between the Message Keeper and the Transfer Connector.

The MS structure mode suggested has all the advantages of the previous ones. Together with them it has another advantage. It allows the connection of two electronic mail systems, which might lead to no observing all the of the X.400 requirements. For that purpose the Transfer Ensurer should perform the translation function of the protocols.

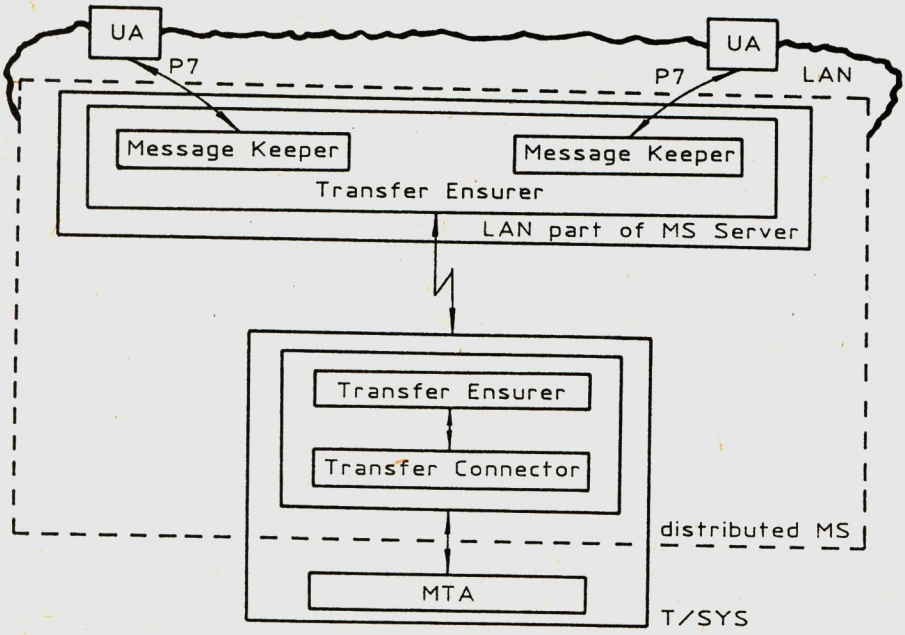


Fig. 6. Distributed MS

The electronic mail system including the distributed MS allows us to discuss what has recently been done from another point of view. The Transfer Ensurer and the Transfer Connector might be treated as an Auxiliary Unit (AU) functional component. This component is united with a MTA, i.e, a T/SYS in terms of a MHS X.400(88) will be obtained.

Meanwhile the Message Keeper and the Transfer Ensurer may be treated as a mail gateway providing the communication between the LAN UA and the corresponding MTA. In other words the LAN users are indirect users.

Let us conclude this subsection pointing out that the two studies of the distributed MS are identical.

5. CONCLUSION

After a short introduction to MHS X.400(88) we identified the reasons for the development of the MS. According to the physical configurations, we investigated several MS structures. The structures discussed above are applicable to the MHS. The concrete choice depends on its real implementation.

REFERENCES

[1] Message Handling Systems, CCITT Recommendations X.400-X.420; Melbourne, (1988).

[2] Cole, R. and Burns J., An Architecture for a Mobile OSI Mail Access System, IEEE Jrnl. on Selected Areas in Communications, 7 (February 1989),pp.249-256.

[3] Hennecke A. and Heymer V., OSI Communication Controller for Heterogeneous Computer Systems, in: Proc. COMNET'85, Budapest (1985).

[4] Hamming R.W., Coding and Information Theory (Prentice Hall, 1980).

[5] Information Processing Systems - Open Systems Interconnection - Basis Reference Model; ISO/International Standard IS7489, (1984).

[6] Saras Pazos J.A. and A. Gonzalez Lanceros ; A X.400 Based Model for a Multiuser Message Store; in: Proc. IBERICOM'87, Lisbon, Portugal (1987).

[7] MIDA - Distributed Application for Message Interchange, ECMA-93, (1985)

[8] MIDA - Mailbox Service Description and Mailbox Access Protocol Specification, ECMA-122, (1987)

A SIMPLE MULTI-PURPOSE INTERNET DIRECTORY

Nils MEULEMANS

Inter-University Institute for High Energy Physics
University of Brussels
Brussels, Belgium

The Simple Multi-Purpose Internet Directory (SMID) is based upon a UNIX environment, in which it is also integrated. Due to the uniform naming and addressing techniques used by protocols like FTP, SMTP, RSH and RLogin, it provides a straightforward and transparent interface to these services. The aim of this paper is not to specify fully a new directory service but to introduce a skeleton of a directory which can easily be tailored to local requirements.

1. INTRODUCTION

The ISO-CCITT Directory Standards [1] are currently still in a draft status. But even when they will be in their final status, still two problems or considerations, which are common to most of the application layer standards, remain. Base OSI standards are usually far too complex to be implemented as a whole and on a local basis, involving rather homogeneous communication systems, they will probably still have to compete with existing solutions. The first problem will in the future probably be dealt with by *Profiling* the base standards into more practical subsets but it will still take several years before these Functional Standards will be used fully on local networks. For what concerns directories, there is however still another problem. Currently only very few solutions exist for Directory Requirements. Furthermore, these requirements have only been specified quite recently (see [2] and [3]) and as such are not really reflected in any of these solutions, which in most cases are just restricted to "Name Servers" and lack the concept of "Yellow-Pages" oriented services.
With the SMID we have tried to propose a solution which could fill this gap on a short term basis, which could be used as an interim solution for as long as no products based on the OSI Base Standards are available and which in the future could even be linked to real X.500 based systems. This latter point is very interesting as it may turn out that the full OSI based standards do not really offer an acceptable alternative for local systems.

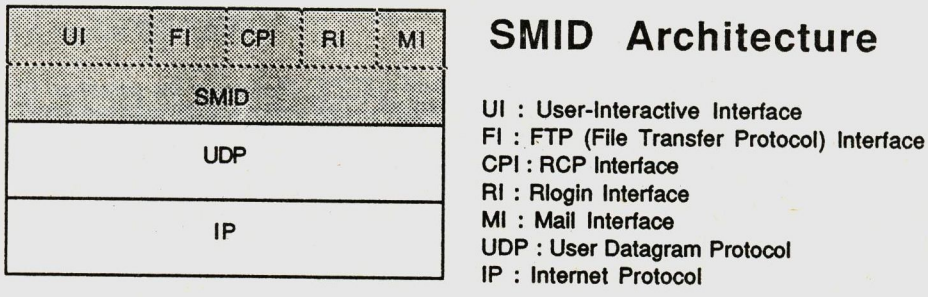
2. BACKGROUND INFORMATION

The Internet Protocol family [4] is comprised of the IP Transport Protocol, Internet Control Message Protocol (ICMP), Transmission Control Protocol (TCP) and the User Datagram Protocol (UDP). The IP protocol may be accessed through a "raw socket" which on most

UNIX systems is only available to the superuser. The ICMP is used to exchange control messages but will usually never be accessed directly by any application program. TCP provides a reliable, flow-controlled, two-way transmission of data. TCP uses the standard Internet Address format and, in addition, provides a per-host collection of "port addresses". Ports are the endpoints of a TCP connection and are mapped, under UNIX, to sockets, through which all kinds of communication take place. TCP however does not provide a broadcast facility, which is the basis of the SMID to support a distributed directory. For this reason SMID is built on top of UDP. UDP uses the same addressing techniques as TCP but, per se, it is connectionless. UDP is however an unreliable protocol which could result in packets arriving out of order or not arriving at all. The way we have overcome this problem is explained in the following paragraphs.

3. SMID FRAMEWORK

The SMID consists of two main Protocol Machines, the Requestor and the Responder. Both Protocol Machines are described in detail in the following paragraphs. The SMID is built on top of UDP and uses a dynamic socket allocation mechanism to be able to handle several requests in parallel. On top of the SMID, a twofold interface has been specified, one which is interactive and one which can be used by Real Time applications like FTP, rcp, rlogin and mail.

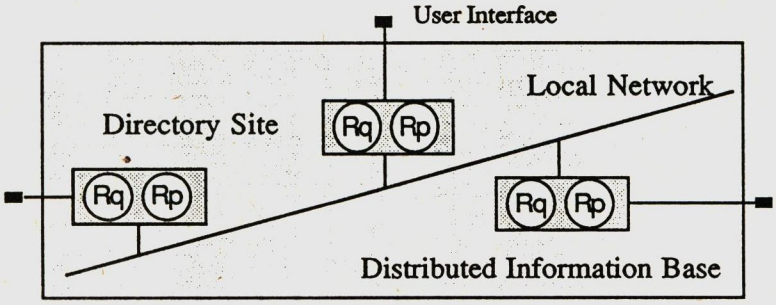


The interactive service is provided through a menu system while the Real Time service has a programmable interface or can be accessed through special versions of FTP, rcp, rlogin and mail. These new services have been called DFTP, Drcp, Drlogin and Dmail, where the "D" stands for *Directory*. Their functionality and way of operation are decribed in § 5.

4. SMID METHODOLOGY

The SMID is a Distributed Directory which consists of a number of Directory Sites. Each Directory Site contains at least a Requestor (Rq), which in functionality resembles the Directory User Agent (DUA) of X.500 (see § 7 for a short description of some X.500 concepts) and which will handle the incoming directory requests issued by the user through the User Interface. The Requestor will then broadcast a packet, indicating the type of request and a request-key, which will be used in searching the directory files, to all the Responders (Rp). The Responder can be compared with the Directory System Agent (DSA) of X.500. When receiving such a request the Responder will check its local Directory File and will

convey back to the Requestor all the matching entries. It is then up to the Requestor, through interaction with the User Interface, to select the correct entry or entries. Due to the unreliable characteristics of UDP, which is the carrier of all requests and responses, a special protocol has been specified (see § 6).
In the pilot implementation no additional files were needed to store the actual directory information since all information was kept in the password file (/etc/passwd). This turned out to have a number of rather interesting advantages as it inherits implicitly the security level of the password file and makes management part of the superuser's task. This latter point proves to be very important on medium sized machines in the category of up to about 20 users. The only problem we encountered was located on environments running NFS (Network File System) where management was more or less centralized and updating directory information was not really part anymore of the superuser's task.



This problem was solved by rewriting the standard *GETPW* (Access to the password file) commands in such a way that they carried an extra parameter (File Descriptor) with which a specific directory file could be identified. This allowed the Directory Information to be stored in any file (including generic YP files). Below a short description of the standard password file structure and related access routines can be found. Furthermore it explaines how the password file structure has been used to include the extra directory information and which extra routines had to be developed to have a more specialized access.
A standard entry in the password file looks somewhat like this:

LoginName : Password : UID : GID : Owner Info: Home Directory : Default Shell

With the exception of the "Owner Info" all fields have their predefined meaning within the UNIX system. On most UNIX systems the "Owner Info" field is of unlimited length and only the three first subfields, which in this case are separated by commas, have a predefined meaning. They are: FullName, Office and Phone Extension. Since all the information provided for these fields is only text, one can put here whatever he wants, but it should be kept in mind that on some systems commands like *finger* expect some sensible information here. In other words, *finger* displays user information in tables headed by: Login, Name, TTY, Idle, When and Office, where Name corresponds to the first subfield of "Owner Info" and Office to the second and the third.
All standard information retrieval functions accessing the password file return for each entry the following C structure:

```
struct   passwd  {
         char     *pw_name;
         char     *pw_passwd;
         int      pw_uid;
         int      pw_gid;
```

```
int     pw_quota;
char    *pw_comment;
char    *pw_gecos;
char    *pw_dir;
char    *pw_shell;
};
```

The fields *pw_quota* and *pw_comment* are unused; *pw_gecos* is the name given to, what we call here, the "Owner Info" field.

In order to be able to store more interesting information, we have extended the "Owner Info" field as follows:

Full Name, Office, Phone Extension, International Phone, Mail Address, Bussiness Address

Changing the structure of the "Owner Info" field is very easy since the information retrieval functions are independent of the length of this field and of the number of subfields it contains. The information rendering functions will scan the "Owner Info" string until an "end-of-string" character is encountered and will separate subfields on a *comma* basis.

5. SMID SERVICES

5.1. Service Specification

All services provided by the SMID are based on a simple "search-by-key" mechanism. The type and the syntax of the key are defined by the Service-Type parameter (see figures below). Before specifying the service, we should first make a distinction between the service provided by the Requestor and the service provided by the Responder. The Requestor offers its service directly to the User Interface and is at a rather high level of abstraction. When receiving a request through this latter interface, the Requestor will perform in some cases some preprocessing of the data provided, it will then map the request onto a more reduced set of service primitives, which is in fact the service provided by the Responder, and will forward it over the network to all connected SMID sites. A description of the User Interface Service, Interactive and Real Time, can be found respectively in paragraphs 5.2. and 5.3.. Below a description is given of the Responder Service.
The service provided by the Responder is based on the following three functions: *getpwuid*, *getpwnam* and *getpwfull*. The first two functions are standard UNIX C-functions. Respectively they search the password file based on a UID or a LoginName and return the first matching entry in the format listed in §4. Since UIDs and LoginNames should be unique on a single system, this first match is also the only match. Matching in both cases is evaluated on full integer and string equality. In most cases directory inquiries will be based on FullName searches; for this reason, the function *getpwfull* has been specified. This routine uses the standard *getpwent* function which allows to step through the password file entry by entry. Since matching in this case is not performed on full equality but allows for substring equality and even the use of wildcard characters, it could happen that several matching entries are found. In this case all these entries will be returned by the Responder.
Due to the unreliable characteristics of UDP a rather complicated protocol is used based on send-buffers, acknowledge-queues, timers, etc. For a more detailed description of this protocol we refer to §6.
Adding an extra service to the existing functionality usually implies three things. First an additional subfield in the "User Info" field of the password entry should be included. If this

new information is subject to being a search-key, new search functions will have to be built. This is very simple since the skeleton of such a function can be obtained from the *getpwfull* function. Furthermore, the representation of this information in the Interactive User Interface is rather straightforward. Whether any changes will have to be introduced to the Real Time Interface will depend on the nature of the new information.
Currently an Automatic Service Builder (ASB) is under investigation. The ASB is an interactive service which asks you which fields you would like to add to or delete from the "User Info" field and which service functions you need. It automatically updates your SMID implementation by including new procedures and new structures.

5.2. Interactive Interface

The Interactive Interface supports one of the two service classes provided by the SMID. Currently this service is provided through an interactive menu which looks like this:

```
Search by UID ..................... U
Search by Login Name .............. N
Search by Full Name ............... F
Search by Initials ................ I
Search by Department .............. D
Redisplay Previous Record ......... P

Display Main Menu ................. M
Quit .............................. Q
Your Choice :
```

The first five functions displayed in this box are mapped by the Requestor to the three Responder functions described in the previous paragraph. Each such mapping consists of a packet of the following format:

Request Packet

The *Request Id.* is a long integer (i.e. 4 bytes) which identifies each transmitted request. The Ids are used for synchronization of the Requestor and the Responder, for acknowledgements and for possible retransmission. The way they are generated and constructed is described in §6. In this paragraph it is also explained why it is impossible to generate duplicate Request Ids. *Service Type*, which is an integer, refers to the requested kind of service, offered by the Responder. Currently the following three alternatives exist: UIDGET (mapped to *getpwuid*), NAMGET (mapped to *getpwnam*) and FULGET (mapped to *getpwfull*).
The first three functions of the Interactive Interface are mapped, in a straightforward way, to these three service alternatives, the fourth one requires however some kind of preprocessing. To start with, it will only accept a number of characters which it will automatically separate by blanks. When the full range of initials are entered, the Requestor

will append the wildcard "*" to each character, matching everything upto the first blank, and will forward it as if it had been a request of the third type. It is up to the Responder to do the "Wildcard" matching of the *Key* field with the entries in the password file.
The "search-by-department" service is based on principles described earlier in this document. As opposed to the previously mentioned functions, a request of this kind will not be broadcasted but will just be forwarded to all the sites of the relevant department. Therefore their should be a relation between each department (name) and the machines belonging to that department. In the pilot this goal was reached through the fact that each machine carried as name the abbreviated name of the department followed by a number. The "search-by-department" is not fully described in this paragraph since it was still in a rather preliminary stage.
"Redisplay Previous Record" allows to scan through the buffer of received responses. And, finally, "Display Main Menu" redisplays the main menu box (see figure), which might be erased from the screen, when using the Interactive Interface, if more than one matching entry was found.

5.3. Real Time Interface

Basically the Real Time Interface provided by the SMID is based on the same Responder services as they are used by the Interactive Interface. The Real Time Interface is in fact nothing more than a programmable interface to the three Responder functions mentioned earlier on in this document.
The interface, which will be discussed in this paragraph, is one which is already at a higher level of abstraction, namely one which has been built into FTP, rcp, rlogin and mail. Respectively new servers have been specified, called DFTP, Drcp, Drlogin and Dmail. The general syntax of the related new commands is:

Command [options] userid[@department]

options

-U : *userid* interpreted as UID
-L : *userid* interpreted as Login Name
-F : *userid* interpreted as Full Name
-I : *userid* interpreted as Initials

These options are furthermore extended by the options of the original command, which it replaces.

userid

Identifies the account to which the connection (not an SMID connection) will be established. According to the allowed options it may be a UID, a Login Name, a Full Name or just some initials.

department

In some cases a department can be specified to make the interconnection more restricted. If a department name is used, the request will only be forwarded to sites belonging to that department.

- **Relation between *userid* and *user* or *username***

A "user" is a unique identification of a user on the overall network. An example could be nils@elem3. The *username* in this case would be "nils". Each time a *username* occurs in one of the original syntaxes of the commands supported by the SMID, also a *hostname* will appear. This means that username@hostname is in fact always semantically the same as "hostname -l username" which is also the same as "user". As

such the *userid* of the SMID request will, after a directory lookup based on the included option, be mapped to the *username*. Furthermore in each response, the UDP packet will contain the Internet Address of the responding machine. This address will be used to lookup, in the /etc/hosts or in the YP hosts files, the name of the machine; this name will become the *hostname*.

- FTP [-v] [-d] [-i] [-n] [-g] [host]

  FTP, the ARPA File Transfer Protocol, supports the transfer of files between two sites running TCP/IP. It is interactive in that it automatically sets up a connection to a requested system and afterwards prompts you explicitely for a LoginName of the account to which you want to login. Based on the destination specified in the initial command, DFTP will propose a list of Login Names which matched this destination. After the selection of a specific username, a password is demanded. When frequent transfers to the same account may occur, an autologin can be forced by specifying entries of the format <host, username, password> in the .netrc file in the user's home directory. Autologin can be disabled by using the -n option on the FTP command. When the password is accepted an interactive session (cfr. FTP), during which files can be exchanged between the interconnected systems, will start.

- rcp file1 file2
  [-r] file... directory

  Rcp is an alternative for FTP. Its advantage to FTP is that it is not interactive but performs the file transfer automatically. However, in order for the transfer to be successful, the LoginName of the issuer should exist on the remote machine. This drawback can be overcome by simulating RCP by a RSH (**rsh** host [-l username] [-n] command). The rsh command connects to the specified host and executes the specified command on that machine. Shell metacharacters are always interpreted on a local basis except when qouted. If we use rsh instead of rcp, we should distinguish between sending a file and retrieving one.

  sending : rsh rhost -l username cat ">" remotefile < localfile
  retrieving : rsh rhost -l username cat remotefile > localfile

  Drcp maps transparently to these rsh constructions.
  The main reason why FTP is interactive is because a password is required in most cases (see above). Neither rcp neither rsh require a password. This means that security has to be arranged in another way. For rsh it is up to the system manager or to the owner of a specific directory to specify who has access to which files. The superuser could say that two machines are equivalent by adding the foreign machine to his /etc/hosts.equiv file. In this case all users of one machine have access to their account on the other machine, if the account exists. A specific user can however put more particular restrictions by inserting hostnames and usernames in his local .rhost file.

- rlogin rhost [-ec] [-l username]

  The rlogin command connects your terminal on the current local system to the remote system, *rhost*.. Drlogin will map the syntax, as it is given in the beginning of this paragraph, to the one specified in the original rlogin.

- mail [-r] [-i] [-n] [-s subject] [user]

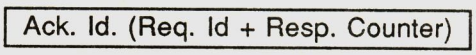
  Dmail will map, as specified above, to mail. When a directory search would result in a list of users, dmail allows to send mail to all of them or just to a selected subset.

6. SMID PROTOCOL

The specification which can be found in this paragraph describes only informally the protocol used by the SMID since still some changes of technical details can be expected. This paragraph mainly describes two aspects of the SMID protocol: broadcast requests from the Requestor to the Responder with retransmit extensions and responses from the Responder to the Requestor with buffering of information and acknowledgements. Summarizing, we could say that the main part of the Requestor-Responder protocol is concerned with providing a reliable transport service based upon UDP but still providing the broadcast functionality. The major problem lies in the fact that the Requestor can receive responses from several remote sites at once. Therefore it should be able to distinguish amongst acknowledgements of different Responders. Each time a Requestor receives a request form the User Interface it will broadcast a packet with the following format.

Request Packet

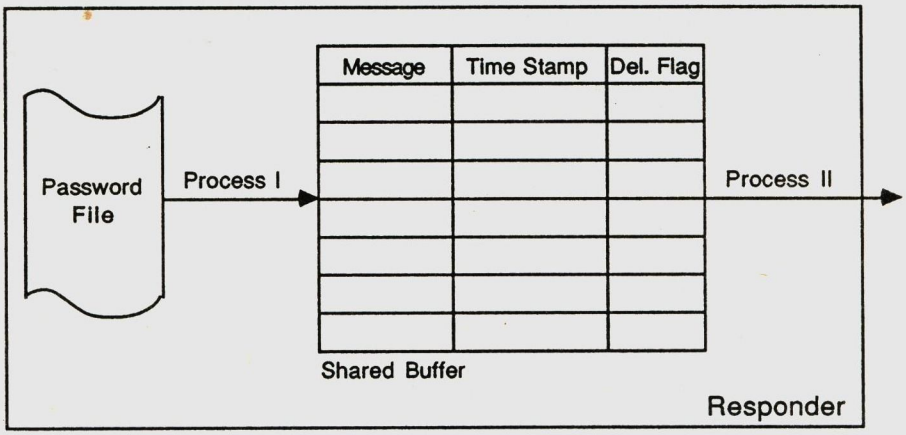
This packet will be kept in storage for possible retransmission if no answer was received from any site after a timeout. Before sending the initial request, the Requestor will initialize a FIFO queue which it will use to store all the incoming responses to the issued requests. In order to distinguish amongst responses issued on the same machine, but by another invocation of the Requestor, it will construct the Request Id. by concatenating the Requestor's Process Id. with a request counter. Once the packet has been sent, the request counter will be incremented by one (modulo Max Req. Id.). Bearing in mind that Request Packets are carried by IP Packets, duplicate ids can only occur when requests are sent from the same machine and within the same session, i.e. the same process id.. If we make Max Req. Id. big enough (e.g. 100 as in the SMID pilot), old requests will have died before duplicates can occur due to the 15 seconds default TTL (Time To Live) of an IP Packet. The Responder will also concatenate its own incremental Id. to this Request Id. for each response it returns to the Requestor. By stripping of this Id., the Requestor is able to check whether it was a response for its last request or whether it was a lost response (e.g. a response for a previously issued request by the same Requestor Process). On receiving a valid response, the Requestor will immediately answer with an acknowledgment. The reason why acknowledgements are not grouped lies in the fact that Responses can be received from several sites at once. In order to be able to group acknowledgements, the Requestor would be forced to keep a queue for each possible Responder. This would become impossible to be managed by one Requestor process.
An Acknowledge packet looks like this:



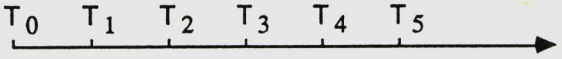
Acknowledge Packet

An Acknowledge Packet is distinguished from a Request Packet by the Responder due to the fact that it arrives on a socket opened by a child process of the Responder. Real Requests will only be received by the Responder Listener.

For the Responder site the protocol is a bit more complex since the Responder should handle also an acknowledgment queue. The figure, which you can find below, illustrates the Responder site. The Responder consists of two processes. The first process scans the password file, picks out the relevant entries and inserts them in the Send-Buffer (implemented as a Shared Memory queue in the experimental SMID). The other process will read this buffer and, when available, will forward the appropriate messages. This is possible due to the fact that, for reasons of simplicity, the Send-Buffer has been combined with the Acknowledge Queue and as such in some cases the Responder (Process II) will have to select older entries for retransmission. It should however be mentioned that the following figure only shows an abstract model of the Send-Buffer.



In reality the *message* part of the buffer is only linked through pointers to the rest of the buffer. The acknowledge mechanism, as it is described below, will in some cases have to replace messages in the buffer. Modifying pointers in this case is more efficient than copying messages. The Acknowledge mechanism uses the fields *Time Stamp* and *Del. Flag* of the Send-Buffer. Each time an entry becomes available in the buffer, Process II will mark the *Time Stamp* with "current time". The *Del. Flag* has been set to "false" by Process I, which inserted the entry and will be used to mark an entry as *deleted* rather than really removing it from the queue.



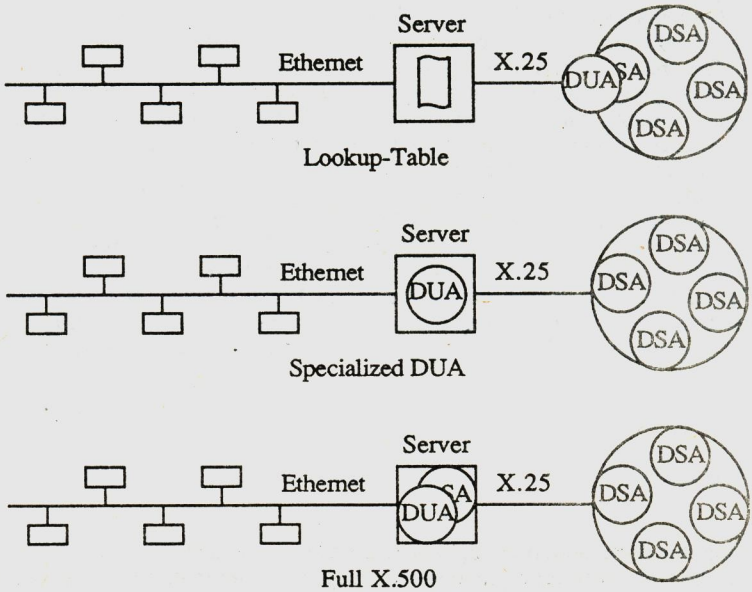
T_i : Real Time "Time Stamp"
$T_i - T_{i-1}$: Timer reinitialisation after interupt

The task of Process II is threefold as it should be able to deal with the following three events: new packet(s) available in the Send-Buffer, the arrival of an Acknowledge Packet and a timeout.
Whenever a new packet becomes available in the buffer, Process II will forward it to the machine on which the request was issued. It will furthermore set the *Time Stamp* field to the "current time" at which the message was sent. When an Acknowledge Packet arrives,

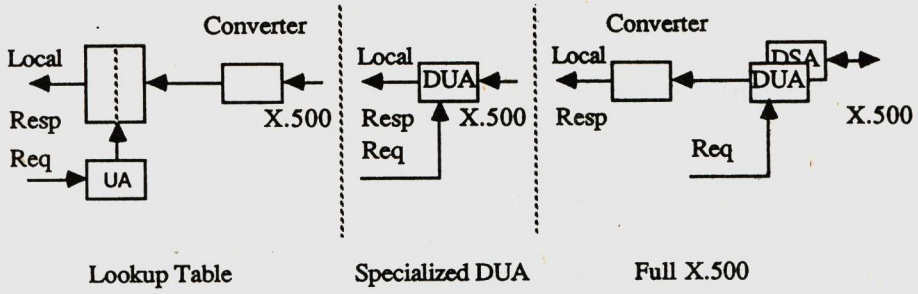
Process II will check whether it is an acknowledgment of the first non-deleted packet in this queue. If not, it will retransmit the message associated with this entry, delete it from its current position in the queue and append it to the end with the *Time Stamp* set to "current time". This procedure will be repeated until the acknowledge matches the first entry in the Acknowledge Queue. This packet will then be deleted from the queue and the timer will be reinitialized to the difference between the *Time Stamps* of the current first entry, in the queue, and the first one which was deleted due to the acknowledgment. When a timeout occurs, the first non-deleted packet of the Acknowledge Queue will also be deleted and appended to the end of the queue, as for the previous event. The timer will be reinitialized to the difference between the current first entry in the queue and the deleted one. If this value is negative (or zero), the same process, as described above, will be repeated until a positive difference is encountered.

7. SMID - X.500 INTERWORKING

This paragraph describes a few infrastructures for linking the SMID to existing X.500 pilots or even early products. It first however refreshes the reader's mind about some of the main X.500 concepts.
X.500 specifies a distributed directory in which information is stored in a number of DSAs (Directory System Agents). DSAs are organised in a hierarchical way and consist of a number of contexts. A (naming) context is a partial subtree of the DIT (Directory Information Tree, i.e. hierarchical form of the DIB, the Directory Information Base) defined as starting at a vertex and extending downwards to the leaf vertices or to one or more vertices each of which denotes the start of a new context. In order to resolve a directory name, it will have to go through a number of contexts before reaching a leaf node. In this latter node the information related to the directory name is stored. Although a DSA resides on one machine, one machine can handle several DSAs at one.



A DSA is accessed through a DUA (Directory User Agent). In order for a DUA to be able to decide whether theDSA, to which it is connected, is able to resolve the query, it should know which initial context is handled by that DSA.
In the *full X.500* example of the previous figure this would be (for the University of Brussels) "vub.ac.be". Anything which does not start with this initial context cannot be handled by the DSA (Server in the figure) and will as such be forwarded to a DSA at a higher level. The figure illustrates three possible infrastructures for the interworking of the SMID with an X.500 based system. In considering these alternatives we should however always keep in mind that the system should work in two directions, i.e. it must be able to handle outgoing calls as well as incoming ones. Let's first look at the outgoing calls.



For each of the alternatives mentioned in the beginning of this paragraph, the above figure shows the related configuration of the Server for outgoing calls. When using a Lookup Table, the UA (User Agent ≠ DUA of X.500) will search the table generated from a remote DSA. The converter will translate the X.500 information into a structure which can be understood by the local systems. When the Server contains a DUA, as in the second solution, this DUA will include the previously mentioned converter. Requests are directed immediately to the DUA. The advantage of this solution over the previous one is that the Lookup Table behaves in fact in a dynamic way. In the third alternative, where the Server becomes an X.500 context, the mechanism is more or less the same as for the Specialized DUA. The fact that the DUA accesses the DSA locally is transparent for the user and will differ, for the outgoing calls, from the previous alternative only in execution time. For what concerns the incoming calls, the latter solution is in fact the only one which, as it stands, provides a real acceptable solution. While in the first two cases updates will have to be sent to the remote DSA, on a regular basis, the third solution will probably contain, on a permanent basis, all the latest information of the local system in X.500 format or will at least have access to it through X.500 protocols. This means that any updates to local information bases will have to be forwarded immediately to the DSA. Procedures to cope with this problem, built on top of TCP, have been specified but have not yet been included in the SMID. The transition from this alternative to full X.500 is rather straightforward since it would just leave us with this latter DSA management procedures, outruling the SMID completely and returning the password file to its original format...an X.500 directory is born.

8. CONCLUSION

After reading this document, one might have the impression that for the SMID the accent has been put in the first place on implementation aspects rather than on precise specifications.

This results from the fact that, as it has already been stated in the introduction, full X.500 Directory specifications will be coming up in a few years, but that currently nothing exists which can fill the gap of the user requirements. Furthermore it was felt that writing specifications based on pilots and experiences gained from them, is probably better than specifying a directory from scratch following user needs. The big difference between a directory and any other application layer service, is that it provides a more or less new functionality which was not really used in the past but for which a need existed.
With the SMID we tried to prove that with a minimum of efforts (manpower as well as resources), based on existing infrastructures, a straightforward and "easy to use" directory could be built which could deal with the existing requirements. We say "could" because the functionality of the pilot SMID, as it is described here, probably does not cover the requirements of all organisational structures. We tried however to point out how easy it is, with the SMID concepts in mind and a pilot as the one described here in background, to adapt a basic SMID in order to cope with most local requirements.

ACKNOWLEDGEMENTS

This paper owes a lot to the contributions of Chris Van de Vijver (AGFA Belgium).

REFERENCES

[1] The Directory, Parts 1-8, X.500 (ISO DIS 9594) (ISO/TC 97/SC 21, 1988)
[2] C. Huitema, A proposal for a naming, data structures and name data distribution (May, 1988)
[3] Einar Løvdal, Directory Requirements for the COSINE FTAM Service (May, 1988)
[4] DDN Protocol Handbook, Volumes I, II & III (Defence Communications Agency) (Department Of Defense, 1985)

Directory Names and Schema - An Evaluation

Kai Jakobs
Technical University of Aachen
Department of Computer Science (Informatik IV)
Ahornstr. 55, D-51 Aachen, FRG
Tel.: +241/80-4521; Telex: 0832704 thac d
Fax: +241/80-6295; e-mail:..unido!rwthinf!jakobs

Abstract
The ISO/CCITT proposal on a directory system has recently been accepted as an international standard. However, the discussion still continues whether or not current functionality is sufficient. The paper contributes to this discussion, focussing on naming problems and the schema. Provision of alternative name forms is one of these problems. Integrity of names is also touched on. Some proposals are described and judged. Subsequently, we will review the directory schema concentrating on its impact on naming capabilities. Current recommendations are introduced and assessed.

1. A Brief Introduction

The ISO/CCITT proposal on directory services has finally been accepted as an International Standard/Recommendation (ISO 9594, CCITT X.500 series). Nevertheless, a considerable number of open questions and possible enhancements has been identified by standardization committees. This list includes keywords such as "replication and knowledge management", "short name forms" or "enhanced search".

Besides these topics there is the broad field of naming tasks. Since to a certain extent names may be regarded as a network's interface to (human) users it must be of primary importance to make naming as simple and user-friendly as possible. Thus, the first major topic of this paper is to discuss the standard's most important shortcomings in this area. Note that we will not discuss the directory explicitly. Those who are interested to learn more about the directory and its present functionality should read e.g. [Jak 89a] or [Hui 88], or should refer to the original standard [ISO 88]).

The second issue is the directory schema. The schema is a set of definitions and rules concerning the structure of the directory information tree. Thus, specification of a schema has strong impact on how entries can be named. Current evolution of scope and contents of the schema definition are considered and evaluated. We will particularly focus on interdependencies between schema and naming capabilities.

2. Current Status

Within this section we will briefly review basics of directory names and schema, respectively.

2.1. Directory Names

Names are supposed to be a means to identify a network's Object in a user-friendly way. In this context, the term user-friendly does not refer to eg. a nice, menu-driven and window based man-machine interface but to the system's capability of providing most sophisticated naming facilities such as eg. mnemonic names (that is, names that may easily be constructed from information a user usually possesses from eg. a business card).

Every real-world object is represented by an Entry, the totality of entries forms the Directory Information Tree (DIT). A name is assigned to every entry in the DIT to unambiguously identify this entry and thus the real-world object it represents. In the directory context, this unambiguous name is called Distinguished Name (DN) which is composed of a sequence of Relative Distinguished Names (RDN), each RDN being unambiguous relative to its immediate superior (cf. Fig. 2).

```
Name::=
 CHOICE { -- only possible solution for now --
   RDN Sequence}
RDNSequence::
 SEQUENCE OF RelativeDistinguishedName
DistinguishedName::= RDNSequence
RelativeDistinguishedName::=
 SET OF AttributeValueAssertion
AttributeValueAssertion::=
 SEQUENCE {Attribute Type, Attribute Value}
```

Fig. 1: Names in ASN.1

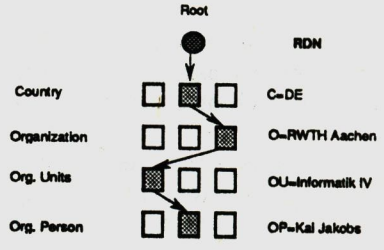


Fig. 2: Determination of DNs

Currently, DNs are the only kind of name supported by the directory; "..names which are formed in other ways are a possible future extension".

Besides the DN, the directory provides for some kind of alternative names of an object, called Aliases. Syntactically, an alias name is a distinguished name as well. An alias entry, however, just holds a pointer at the object entry rather than information about the object.

Names human users have to deal with are supposed to be user-friendly (this requirement is not really met, though). By now you get the impression that, as far as standardization is concerned, user-friendliness takes second place to management issues. With this paper we will try to take countermeasures.

2.2. The Directory Schema - A Very Short Introduction

The directory schema is "a set of definitions and constraints concerning the

structure of the DIT and the possible way entries are named..." [ISO 88].

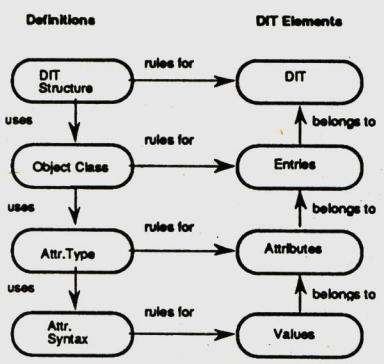


Fig. 3: The Directory Schema

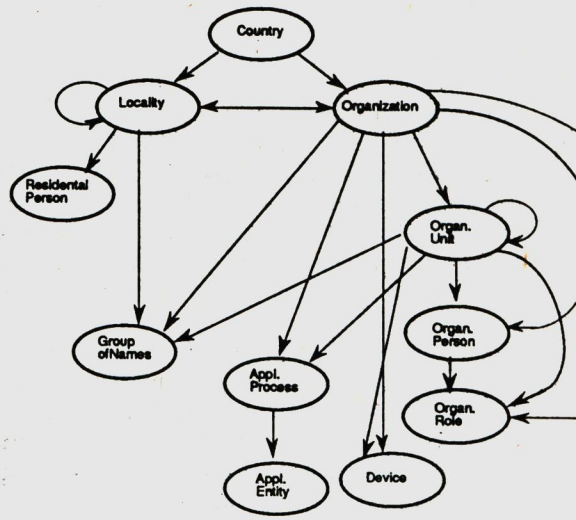


Fig. 4: Suggested DIT Structure

As we may see from fig. 3 the schema definition entails the relationship between object classes, specification of mandatory and optional attribute types allowed for each object class and the respective attribute syntaxes.

The schema is supposed to be distributed. That is, every administrative authority, responsible for a part of the DIT, specifies its own schema. Exchange of schema information between authorities is not provided.

In particular, specification of a schema includes defining the permitted hierarchical relationships between entries. That is, defining the schema also means defining the structure of the distinguished name of an entry, cf. Fig. 3. Fig. 4 shows the suggested DIT structure (i.e name form) as defined by the current standard.

3. Alternative Name Forms

We have already stated that an entry's name should reflect some common knowledge about the object it represents.

However, there may exist several different information about one single object. This may result in different attempts to create a distinguished name. For example (taking the author as a guinea-pig):

An American may try it this way:
{C=FRG, O=Techn. University of Aachen, OU=Dept. of Computer Science, OP=Kai Jakobs}

The German attempt might look like this:
(C=BRD, O=RWTH Aachen, OU=Fachbereich Informatik, OU=Informatik IV, OP=Kai Jakobs)

You can easily imagine that there are lots of other, perhaps even completely different looking names supposed to identify the same object. If you are lucky, one of those names may be the correct one, but you cannot be sure about that. You will find the directory's strategies to cope with the above problem rather unsatisfactory. At present, the only name known to the directory, and thus the only way to name an object is the distinguished name. For each entry, a Distinguished Attribute Value is determined for the RDN of that entry (the corresponding attribute type is defined by the schema, cf. sect. 2.2.). The one alternative name currently provided by the directory is the alias, syntactically being a distinguished name as well.

Although aliasing allows to better accommodate the users' needs, this approach's drawback simply is the resulting large number of alias entries necessary, increasing memory requirements as well as search time and search costs. What's more, it will obviously not be possible to provide an alias for every potentially meaningful combination of possible RDNs of an object. One thing to be said in favor of this approach is that it is fully in line with the current standard, which in fact is an argument that may not be underestimated.

Currently, at each level one single distinguished attribute value has to be selected for every entry. Additionally, it is permitted to use more than one Attribute Value Assertions (AVA) per entry. Using ASN.1, AVA ::= SEQUENCE {AttributeType, AttributeValue}. However, this may be used only to provide some supplementary information rather than an extended naming facility. Thus, a solution you may think of is admission of Multiple Distinguished Attribute Values. Now different attribute values may at each level be recognized as name parts. Thus, this approach can easily handle issues like the American/German sample problem shown above.

Unfortunately (since it is a very simple approach) this idea also has its drawbacks:

- The Name Resolution Process, this is the process determining whether or not a purported name really is a distinguished name will have considerable problems doing his job.

- Non-ambiguity of names will be hard to achieve and difficult to control. Probably, the user will much more often have to face a Name Error as a result of his search, due to an ambiguously specified name.

- The standard provides only for single distinguished attribute values. Thus, permission of multiple values would mean a modification of the standard which may be hard to carry through.

Another approach to enhance the directory's naming facilities is to extend the

current, quite restrictive definition of Name. Fig. 5 shows the ASN.1 notation of an extended definition of Name.

The nature of both ordered and unordered names should generally be clear from their respective names. The term "ordered" refers to the position of the single AVAs in the complete name. The two name forms may further be subdivided into Typed and Untyped names, respectively. Here, typed means that the respective attribute types have to be given in the name as well. Obviously, untyped doesn't require this.

Here again we may observe the well known dilemma user-friendliness vs. administration. Of course the user would prefer the untyped, unordered variant, meeting his requirements best. In contrast, system management would strive for either distinguished names as the only name form or, at maximum, for typed and ordered names (of which DNs may be considered as a subclass). From system management's point of view the list of problems that may occur with untyped and unordered names includes name resolution process, ambiguity of names and name matching. Nonetheless, at least the latter may be obtained by applying, admittedly complex, filters.

```
Name  ::=  CHOICE  {
    DistinguishedName
    ObjectIdentifier
    UnorderedName
    OrderedName   }

UnorderedName  ::=  SET OF AttributeValueAssertion
OrderedName  ::=  SEQUENCE OF AttributeValueAssertion

AttributeValueAssertion ::=
    SEQUENCE  {AttributeType, AttributeValue}
```

Fig. 5: Extended Definition of "Name"

Untyped and unordered names give considerable more user-friendliness. A user's name now looks much more like an address known from a normal letters. The weakness is obvious: searching will become much more time consuming without attribute types to be used to navigate through the name tree. In fact, regarding the present naming rules, it may be really hard since no fixed name form is defined (that is, some attribute types may appear more than once, cf. sect. 2.2.); the system will have considerable problems to determine the proper attribute type for a given value.

Providing Descriptive Names is another way how a more sophisticated naming may be achieved. The idea is to generally permit a name form which is, in contrast to the name forms discussed above, completely different from a distinguished name. Descriptive names, contrary to DNs, do not necessarily reflect hierarchical relations within the DIT (and, of course, can not be used to route through it). The idea of providing descriptive names first appeared in an older draft version of the directory [ISO 86], but has been removed in later documents due complexity reasons (management, name resolution, ambiguity

of names etc.). Anyway, increasing processing power may outdate these arguments.
Untyped and unordered names may be a subset of descriptive names.

4. Discussing the Future Schema

This chapter addresses modifications of today's schema definition currently under study in the standardization bodies. Since a complete discussion of this topic would go too far, we will focus on those aspects relevant for naming issues.

In general, you may state that the schema definition will be enhanced and become more precise. The schema will consist of a set of subschemas each of which administered by a schema authority. A subschema comprises the well-known object class and attribute definitions, respectively, and a set of structure rules, which in turn will consist of a set of name bindings (cf. Fig. 6). Currently, structure rules are supposed to be a local matter and are therefore not standardized by the directory.

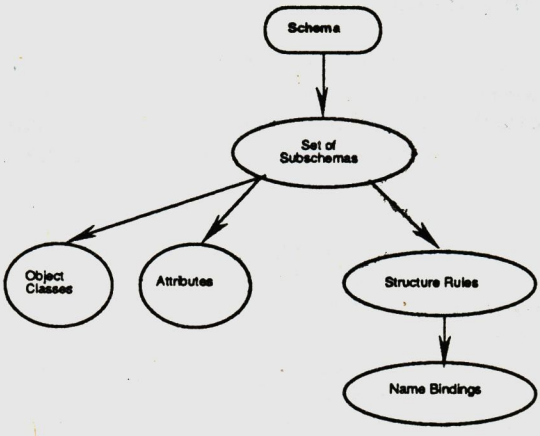


Fig. 6: Structure of the Schema

4..1. Object Classes

There is one major modification as far as object class definitions and their impact on naming is concerned. The current standard lays down that every entry belongs to exactly one object class, with this object class possibly being derived from exactly one superclass. Thus, requirements for eg. additional attribute types within an object class strive for defining a new class, either registered or unregistered. Of course, introducing a new object class alters the DIT within its domain and thus the possible name structure (cf. fig. 4). Therefore, altering the DIT should be done very carefully in order not to make naming a guessing game.

The new approach provides for entries pertaining to more than one object class not necessarily belonging to the same superclass chain. Every entry may belong to exactly one structural object class and an arbitrary number of auxiliary object classes, the structural object class being the one relevant for naming.

Provision of this mechanisms has a very nice effect: attribute types may easily be added to an entry of a given class without creating a new object class and thus changing the DIT structure. As a result, an object may for example originally belong to object class organitationalPerson and additionally become element of a newly created object class mhsUser (in this case, an auxiliary object class) without requiring the DIT structure to be altered.

4..2. Name Bindings and DIT Structure Rules

A name binding defines a hierarchical relation between two object classes permitted within the scope of one particular schema. Note, however, that name bindings only apply for structural object classes and that it is this object class defining location of the entry in the DIT. The complete set of name bindings forms the DIT structure rule.

Specification of structure rules provides information being of essential importance for naming issues. Remember the suggested DIT structure depicted in fig. 4. Due to the not exactly tree-shaped structure of the DIT there may very often be more than one way to route through the DIT to reach the destination object class.

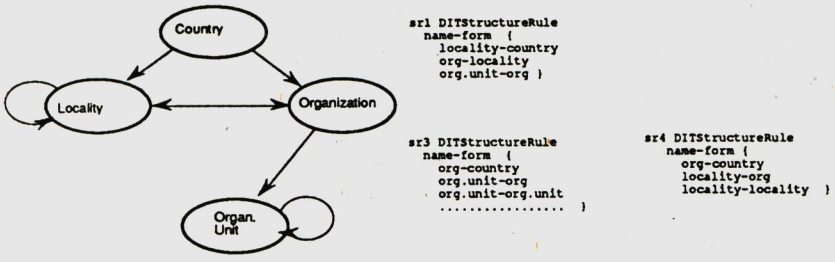


Fig. 7: Part of the old DIT structure and some possible structure rules derived

Applying structure rules, the depth of the DIT and, accordingly, the precise structure of a distinguished name may be specified. For instance, structure rules define the number of attribute types organizationalUnit permitted for a DN.

4.3. Name Integrity

Names were originally supposed to be long lived. It seems, however, that this assumption has been somewhat too optimistic; people may marry, move or change the employer, thus causing RDNs' changes.

Modification of an entry's RDN may have two effects (note that in accordance with the standard leaf entries may be modified by their respective owners whereas modification of non-leaf entries requires considerable administrative action and possibly multilateral agreements):

- Local copies of the name will become invalid
  Since the DS is not aware of information cached locally there is no direct way how cache entries may be kept up to date, especially since knowledge of cached entries would result in a considerable overhead.

- Aliases of the old name will become invalid
  No mechanisms are specified to dereference aliases. In fact, dereferencing is excluded explicitly mainly due to the fact that it would require backward pointers from an entry to its aliases. And this, in turn, would require global knowledge of alias entries for every single entry which is nearly impossible to achieve.

Imagine an entry has been deleted. Thus, it is possible to create a new entry with the same DN, representing a different real-world object, causing tremendous trouble to authentication and access control.

After having discussed ideas like to forbid re-use of names or to prohibit modification of RDNs (both of which result in unacceptable restrictions for the user, cf. eg. [Jak 89b], the following has been proposed [ISO 89]:

- Any entry to be removed will be marked.
- Marked entries remain part of the DIT for a certain time which will be determined by some administrative authority. This period may depend on the object class of the respective entry and perhaps also on memory requirements.

- During this period, creation of an entry with the same name will be prohibited by the system. The entry will then only contain naming attributes and perhaps a pointer at the new entry representing the same object. A user accessing the entry will be forwarded to the new entry and will be informed about the new DN of the object, for example via a Name Error.

- Any reference (from an alias entry) to a marked entry will result in a Service Error. The new DN may be reported to the requestor.

This approach achieves consistency of the directory for a reasonable time period.

4.4. Publication of Schema Information

We have stated above that specification and use of DIT structure rules will exactly define the structure of a distinguished name. However, this structure

may vary between administration domains. Thus, in order to be able to name an entry correctly, a user must be aware of the respective structure rule valid for the requested entry. Besides several more or less restrictive proposed solutions like eg. provision of one global schema, publication of schema information (a topic area absolutely not covered by the present standard) has turned out to be the only satisfactory way to manage this problem. In other words, it is necessary to distribute and possibly replicate schema information. This may be done using the same mechanisms to be provided to replicate "normal" data and knowledge information, respectively. However, replication and knowledge management are topics on their own and outside the scope of this paper. We only have to keep in mind that it will be a must to distribute schema information to reduce or even omit the guessing when trying to name an object.

5. Some Concluding Remarks

It is very likely that the directory will become one of the most important services in networking. This holds especially for the message handling world since a lot of "computer illiterate" people will use messaging services without having any specific knowledge about eg. network addresses and routing problems. Furthermore, it is known from the telephone network that introduction of the Yellow Pages lead to a remarkably enhanced acceptance of this network.

Currently, the directory is taking its first steps towards being an integral part of networking. Against this background it appears reasonable that the system should provide its service as user-friendly as possible. However, today's trends in standardization mainly focus on things related to management tasks rather than user-friendliness. To our mind it is completely absurd to make a service like the DS too complicated from its very beginning on rather than focussing on user's requirements like eg. simple, understandable and sophisticated naming facilities.

Within this paper we have discussed and judged some of the most important aspects related to naming and the schema. Most of the respective DS services turned out to be not adequate when compared to user's demands. In this area considerable efforts will have to be made.

Current naming strategies require too much specific knowledge from the user. Distinguished names are much to restrictive to be accepted as the only name form possible.

In addition, specification of the schema will need considerable more efforts. What's more, despite the claim of relieving the user from network specific details he will soon be forced to explicitly learn about remote schema information if he wants to get information from a DSA in a different domain.

After all, standardization of the directory goes on. The next, 1992 version of the

standard will look very different from the current one. Let us hope that it will be recognized that the directory originally was intended to provide user-friendly services.

6. References

[Hui 88]: Huitema, C.:
The X.500 Directory Service
Computer Networks and ISDN Systems, Vol. 16, pp. 161-6, 1988

[ISO 86]: International Standards Organization
ISO 2nd DP 9594, The Directory Part 1-7, Egham, 1986

[ISO 88]: International Standards Organization
ISO IS 9594, The Directory Part 1-8, Sydney, 1988

[ISO 89]: International Standards Organization; ISO/IEC JTC1/SC21 N3581
Working Document on the Directory Schema

[Jak 89a]: Jakobs, K.:
The Directory - Evolution of a Standard
IFIP TC6/TC8 Open Symposium on Network Information Processing Systems, K. Bojanov, R. Angelinov (eds.), Elsevier Publishers, North Holland 1989, pp. 281-90

[Jak 89b:] Jakobs, K.:
ISO's Directory Standard - Does It Require Enhancement?
to be published in Proc. 9th Int. Conf. on Remote data Processing DPD '89

Using Directory Services in Mobile Communication

Kai Jakobs, Frank Reichert
Technical University of Aachen
Informatik IV (Dept. of Computer Science)
Ahornstr. 55, D-51 Aachen, West Germany
Tel.: +49-241-80-4521; Fax: +49-241-80-6295
Telex: 08327804 thac d; e-mail: ..unido!rwthinf!jakobs

Abstract
Prospective benefits of the directory service for mobile subscribers to communication systems are discussed. The directory service as such is briefly reviewed. This is succeeded by a classification of mobile users. Specific requirements of the respective classes are identified. These requirements are matched against the directory's functionality, major technical problems are pointed out. Finally, we propose some necessary enhancements and modifications.

1. Some Introductory Remarks and Overview

Originally, the Directory Service (DS) has been intended to act as a service provider for the Message Handling System (MHS). This paper shows that a much broader community of users may benefit from the directory, which now also appears useful eg. for the community of mobile radio users. Of course, it must be stated here that MHS will be a major service for some class of mobile users as well.

First of all we will shortly review the directory's basic functionality. In particular, we will discuss authentication basics being of special relevance for mobile communication.

In order to differentiate we will introduce a rough classification of mobile users. These different classes of users call for different types of services.

Subsequently we will review the mobile user's requirements for an information service like the directory. Needs typical for the respective user class are pointed out.

Thus, the second main part of the paper discusses problems related to use of the directory in a mobile environment. These problems include association of directory user agents with directory system agents, problems of distributed authentication as well as required additional contents of the directory information tree. It should be noted that the directory may eg. serve for authentication purposes in a very general way, so that every user (including of course mobile users as well) wishing to use any service requiring for authentication will have to use the DS.

Another aspect particularly interesting for a mobile user will be the contents of the DIT. Currently, this contents is mostly limited to information more or less directly related to communication tasks. However, this will have to be enhanced towards more general information as well as very particular information typically required by mobile users. We will identify areas of information that will have to be included and discuss impacts of this inclusion on the DIT. These impacts include for example additional object classes and attribute types, respectively.

Finally, we will also discuss technical problems, i.e. requirements currently not being covered by the directory. This includes eg. naming and replication of data.

2. The Directory Service - A (Very) Short Survey

This section is intended to provide a very condensed overview of the subject. It is only supposed to introduce the topic to the reader. Those who have a deeper interest in the directory's functionality should eg. read [Hui 88] or should refer to the original standard [ISO 88].

The directory service provides a uniform naming scheme for and information about a network's resources (including e.g. hosts, processes, devices and human users). In terms of the DS, these resources usually are referred to as *Objects*. A non-ambiguous name is assigned to every object. In general, a DS has to provide three types of service:

- mapping name --> information
  For example, an object's name may be mapped on its network address.
- mapping name --> set of names
  A set of objects is identified by one single name.
- mapping information --> set of names
  This service establishes a "Yellow Pages" function.

The DS is modelled as a - highly distributed - *Client-Server System*. This may be characterized by a typically small number of hosts (the servers, the *Directory System Agents* (DSA)) providing callable services to the other hosts of the system (the clients, the *Directory User Agents* (DUA)).

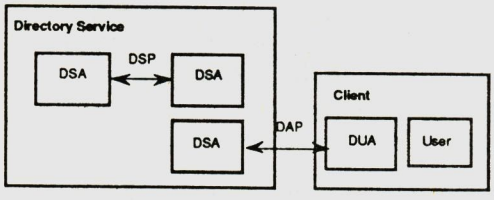


Fig. 1: The Directory Model
DAP = Directory Access Protocol
DSP = Directory System Protocol

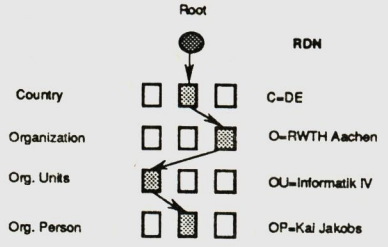


Fig. 2: A Sample Distinguished Name

A *Name* is assigned to every object. This name is called *Relative Distinguished Name* (RDN). Every RDN is non-ambiguous relative to its immediate superior. The sequence of RDNs of an object plus those of its

superiors forms the *Distinguished Name* (DN) of this object. The DN is globally unambiguous. The directory also provides for one kind of alternative names, called *Aliases*.

Every object is represented by an *Entry*, all the totality of entries forms the *Directory Information Base* (DIB). An object's name and the information stored in an entry are composed of *Attributes*. An attribute is a tuple <AttributeType, AttributeValue>.

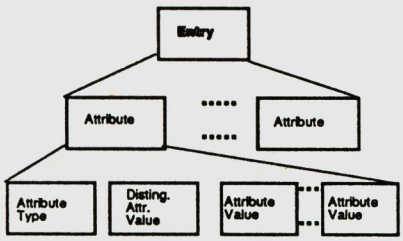


Fig. 3: Model of an Entry

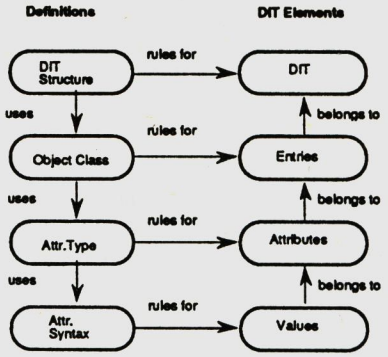


Fig. 4: The Directory Schema

Relation between objects in a network is usually hierarchical. Thus, the DIB will be structured in a (more or less) tree-shaped way. This tree is called *Directory Information Tree* (DIT). Information about objects is stored in the DIT. Every DSA holds data about a subset of objects and has *Knowledge* about objects known by other DSAs (to provide for distributed operations).

The directory's *Schema* specifies the structure of the DIT, defines *Object Classes* (eg. Country, Organization), *Attribute Types* (eg. country name or telephone number) and *Attribute Syntaxes* (eg. printable string or numeric string) permitted. The schema is composed from a number of subschemas each valid within one particular *Administrative Authority*.

Note that functionality of the ISO directory is based on the overall assumption that information stored will be long-lived. Thus, we will not consider the highly transient data required eg. for determination of traffic conditions. These data will have to be provided by a different, though to some extent directory-like service which may be called a "*GSM-Directory*" [Bau 89]. In the subsequent sections we will limit ourselves to the "ISO-directory"

3. Mobile Users and the Directory

Within this section we will discuss some relevant properties of the mobile environment, introduce a rough classification of mobile users and analyze requirements typical for the respective classes.

3.1. The Mobile Environment

Some characteristics of the mobile environment will have consequences for this very particular type of directory user. This is for example due to

limitations in bandwidth: Since the communication channel will also be used for voice and continuous traffic, it will not always be available for data transmission. Therefore, time may increase drastically to obtain the required information.

In the mobile communication environment depicted in Fig. 5 [Kre 89] users access the appropriate *Base Station* (BS), which in turn connects him to the *Traffic Management Center* (TMC) responsible. It appears meaningful to integrate DSAs at TMC level; since it is very likely that TMCs will mostly be located close to conurbations, this placement would be useful for non-mobile communication as well. However, to avoid unnecessary long distances between user and DSA, it may be useful to store at least those copies of data which are of only regional relevance in base stations as well.

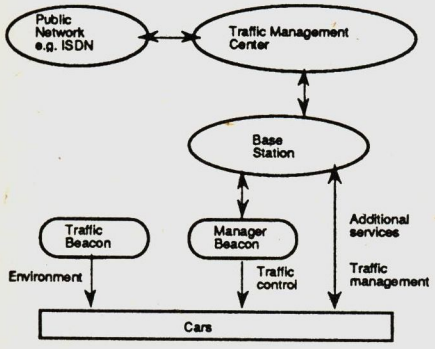


Fig. 5: Hierarchical Network Topology

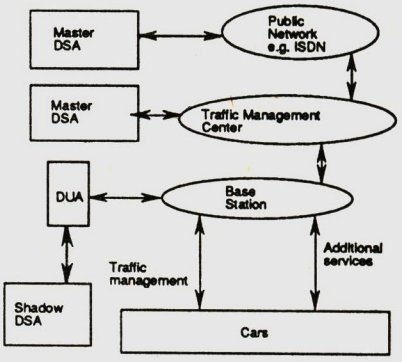


Fig. 6: Location of DSAs and DUAs

Another question concerns location of the directory user agent, DUA. You may think of three possible options:

- DUA functionality is provided by the user.
  This presupposes that the user for instance owns his private computing power. Although this may be true for most business travellers, it may not be assumed for the driver.
- DUA integrated in a TMC.
  This approach shows similar benefits than does integration in base stations. However, since the number of TMCs will be relatively small, this may easily become a bottleneck.
- DUA is integrated in a base station.
  This alternative should be preferred since it guarantees access for both classes of users independent of their respective equipment. Additionally, it may be possible for the owner of a private DUA to access the directory directly (i.e. via a TMC).

Fig. 6 shows possible location of the directory's elements in relation to those types of network nodes typical for the mobile network.

Another aspect important for the mobile user will be replication of information. Since the current standard does not provide for this functionality, this topic will be discussed in sect. 5.

3.2. Classes of Mobile Users

As far as requirements and necessities are concerned, you cannot speak of the mobile user as such. There is a broad variety of different demands depending on the particular mobile environment. Therefore, some classification is essential.

Before discussing his requirements it is necessary to specify what the term "mobile user" means for us although this might be clear intuitively. We will distinguish two classes of users with each class imposing different demands:

- The "Driver"
  As the name indicates, driver refers to the "pure" road traffic environment. Typically, a user of this class will be someone on a trip requiring:
  - information related to traffic conditions. This is rather short-lived information and will therefore have to be provided elsewhere.
  - Information closely related to travelling aspects. This may for instance include things like vacancies, sights or possibly route guidance information, eg. scenic routes.
- The "Business Traveller"
  This name typically indicates a user on a business trip. Common elements of this user's environment are the car, the laptop and some kind of network access unit; keyword: the mobile office. Actually, requirements of users of this class do not differ very much from those imposed by non-mobile message-handling users. The directory will typically be used to support MHS applications.

Of course, these two classes of users may not be said to be completely disjunctive. Nevertheless, this classification exhibits the basic differences between them.

3.3. Mobile User's Requirements

Requirements on the directory imposed by mobile users are twofold as well. They may be subdivided into:

- General requirements
  They are common to "normal" users and to mobile users rather than specific for mobile users. Due to the similarity of the business traveller's demands with common office requirements, we will associate these requirements with the business traveller.
- Special requirements
  These demands are typical for mobile users. They may as a first approach be associated with the driver's demands, that is, these requirements focus on classes of information typically needed in a traffic environment.

Thus, the two classifications introduced up to now may be reduced to one.

Business traveller

Demands of the business traveller category are well understood. They include use to message handling services and access to remote information services. Note, however, that access to the directory will also be required for message handling applications. More specific, the demands will include:

- Independence from internal network details.
  Nobody is willing first to learn details on eg. gateways or network topology in order to be able to address his partner correctly.

- User-friendly naming
  That is naming based on normal knowledge one usually possesses, let's say from a business card, rather than names like today's "maier%vax1.UUCP%germany.CSNET@ csnet-relay.ARPA" which is much more a nice memory training exercise than it is an easy-to-remember user name.
- General assistance
  This includes assistance for example in case a user name is not known correctly. On the other hand, this includes a service similar to the Yellow Pages. It is well known from the telephone network that after introducing yellow pages utilization of the network rose drastically. This very mere fact indicates importance of a service like this..
- Additional information services
  Such services may for instance include information on a recipient's user agent's telecommunication capabilities.

Of course this list does not claim for completeness, but it gives an impression about the kind of services usually to be required from the directory. Note that these requirements are not necessarily typical for a mobile user, but they are of major concern for him as well.

Driver

In addition to these demands the driver shows to have some more extensive, dedicated claims. His interests differ from those identified for the business traveller. This becomes easily understandable when comparing the different environments of the two classes.

As stated above, the drivers major needs for information concern aspects related to the traffic situation. These requirements turn out to be twofold:

- (usually short lived) information from his surrounding area, eg. on traffic jams.
- more general, long lasting information, take route planning, vacancies or sights as examples. This is a typical application of the directory.

There is one more point both classes of users have in common: the need for secure communication. This includes data integrity, data confidentiality and authentication services. In addition, services like eg. billing and accounting are based on authentication as well. This is of special importance since the mobile user will have access to wide area networks like eg. ISDN. In particular, this implies strong authentication, possibly in conjunction with well-defined access control capabilities. Actually, provision of strong authentication shows to be crucial (cf. sect. 4.3.).

4. Meeting the Demands

To be in line with the lists of requirements given above, we will first discuss if and/or how the business traveller's demands are met by the current (1988) directory standard.

4.1. The Business Traveller

When designing the directory it has always been one of the main intentions to relief the user from network internal details (which may be especially valuable for packet radio networks). This is achieved by assigning a name to every user

(i.e. object). In contrast to network addresses which may change quite often and unpredictably these names are intended to be long lived (cf. sect. 2).

It has been mentioned above that a name is assigned to every user. The directory standard calls those names user-friendly. Although this is not quite true yet, compared to network addresses which may be up to 40(!) digits long present directory names really are a considerable improvement.

As we know from the telephone network, things like phone books, yellow pages and telephone inquiries are essential. Thus, it is obvious that equivalences are required in data networks as well. The directory's general mapping functions provide these services (actually, that is exactly what the inquiry does: mapping a given name onto a network address - the phone number). By providing *List* and *Search* operations as well as *Filter* mechanisms the directory allows its users to retrieve specific information - the yellow pages service.

Data stored in entries associated with the network's users comprises a great variety of information, ranging for instance from details on postal delivery and telecommunication capabilities up to information on his business or research interests. In fact, almost all information a user wants to store in his entry may be found.

4.2. The Driver

As outlined above, the driver needs specific types of information currently not provided by the directory. Thus, it will be helpful to introduce new object classes and additional attribute types.

Information of the directory is kept in entries, each entry being associated with exactly one real-world object. Within the entry, information is stored as attributes (cf. sect. 2). Entries holding the same attribute types, i.e. the same type of information, form an object class. Currently, object classes as well as attribute types specified by the standard are more or less closely related to communication aspects. More "customary" information, as it is needed by the driver, is currently not intended to be provided by the directory.

However, provision of such information would be a minor problem. Every administrative authority is allowed to define new, appropriate attribute types. Additionally, entries may belong to more than one object class. Since this approach provision only of registered object classes does not alter the outwardly appearance of the directory [ISO 89].

4.3. Authentication

The directory standard includes an Authentication Framework, describing concepts for provision of secure communication. The framework distinguished between *Simple* and *Strong Authentication*, respectively. Let us consider simple authentication first.

As already pointed out in the standard [ISO 88], "..simple authentication is primarily intended for local use only, i.e. for peer entity authentication between one DUA and one DSA.." It is also stated that the password is valid only within the user's home domain. No procedures are specified how to authenticate at another DSA using this password. It follows immediately that simple authentication may not be applied in a mobile environment, where the user will frequently be attached to DSAs others than his home-DSA.

Employing strong authentication, the mobile user has to face the problem how to authenticate himself to a non-home DSA. This "distributed authentication" may work as follows:

- User A connects to the non-home DSA X, enciphering the message with his secret key.
- Given that X belongs to a different domain than the user's home-DSA Y, X tries to obtain A's public key from Y. In doing so, X traverses a certification path to reach the common point of trust.
- The result is returned to X from the point of common trust and may be conveyed to A.

The same mechanism may be applied for bilateral authentication of two communication partners A and B.

5. Technical Problems

Within this chapter we will discuss technical problems rising from use of the directory in mobile communication, i.e. demands not met by the current standard. Anyway, most of these deficiencies are subject to standardization work.

5.1. Naming

The first thing to be considered concerns the naming problem. Since to a certain extent the name is the user's interface to the network, this calls for a degree of user-friendliness as high as possible. It has, however, been realized by now that considerable additional efforts will have made to enhance the directory's naming capabilities (cf. eg. [Jak 89], [Neu 89]).

Remember the distinguished name depicted in Fig. 2. A DN (taking the author as a guinea pig) might look like this

Attempt A:

```
{C=DE@O=Techn.U.of Aachen@OU=Computer Science@OP=Kai Jakobs}
```

This is, of course, a nice, easy-to-remember name - for English speaking people. The German attempt may look quite different:

Attempt B:

```
{C=DE@O=RWTH Aachen@OU=Informatik IV@OP=Kai Jakobs}
```

However, another German might also try this:

Attempt C:

```
{C=DE@O=RWTH Aachen@OU=Math.-Nat.Fak.@OU=Fachgruppe
Informatik@OU=Lehrstuhl Informatik IV@OP=Kai Jakobs}
```

You see the problems:

- Different mother tongues of users may lead to different attempts in naming objects (attempt A).
- The DIT is not exactly tree shaped. Thus, it is possible that some name parts may occur more than once in a DN. Trouble is preprogrammed (attempts B+C).

Although the dilemma described above is not specific to mobile environment but is a general problem of the directory, its consequences are particular unpleasant for mobile users who will very often need the information faster than a common user does.

5.2. Replication of Information

It has been mentioned above (cf. sect. 3.1.) that directory system agents (DSAs) should be integrated in traffic management centers (TMCs). Referring to the network topology introduced in [Kre 89], there will be about one hundred TMCs (however, this number depends on the bit rate available between the TMCs and on the range of services to be provided). This number of DSAs should be sufficient to guarantee short response times. Nonetheless it would be helpful especially for the mobile user to replicate some part of the information. This will be of special value for the "driver-data" which will usually be of rather local relevance. Thus, this replicated data should be stored at base station (BS) level (cf. fig. 6, this does of course not mean that every BS will necessarily have to act as a DSA). Replication of selected data at BS level offers the mobile user a number of attractive features:

- Very short response times
  Since there will usually not be very many queries at a time and since the amount of data to be stored will be relatively small requested information will be very soon available.
- Less bandwidth required
  A high percentage of queries (especially those initiated by "drivers") may be satisfied locally.
- Better access to the whole system
  This is due to better accessibility of a DSA.
- Lower transmission costs
  Since a considerable amount of data will be stored locally (i.e. in the same cell the user is located) the number of distributed DS operations may be reduced significantly.

Taking all these advantages into consideration inclusion of computing power into a number of base stations to provide for DSA capabilities seems very desirable. This holds even despite the fact that BSs were originally supposed to be as simple as possible.

With the current directory standard replication of information is allowed but not supported; replication is said to relay on bilateral agreements. In particular, the list of topics to be discussed includes:

- Definition of the replication model
- Specification of additional services and protocol elements
- Consistency requirements
- Management and administration problems

A standard document on information replication is supposed to be approved by 1992.

6. Conclusions and Some Final Remarks

The directory is intended to serve all users of all data communication systems. Of course, that means it will also have to serve mobile users. This is of particular significance since on one hand side the "mobile office" will play an important role in the near future and on the other hand communication requirements in the road traffic environment are on the rise as well.

We have addressed requirements imposed on the directory by mobile users. A rough classification of two different types of users has been introduced

together with each type's specific requirements on the directory. Although in general directory services provide for considerable support of mobile users as well, we have to face restraints pertinent to mobile communication. These problems are mainly due to time constraints which may conflict with possibly time consuming distributed operations. The effect of extensive time consumption may also be observed in conjunction with data replication. Replication of data will considerably reduce communication time as well as communication costs. Additionally, naming problems, rising from the not exactly tree-shaped directory information tree may have greater influence on mobile users than they already have on normal users, anyway.

To sum everything up, you can say that the directory's services will make all mobile user's communication life a little more comfortable.

7. References

[Bau 89]: Bauer, R., Pfundstein, M.:
Realization of a Distributed Data Base in the European Mobile Radio Telephone System
Proc. Communication in distributed systems, Springer Verlag 1989

[Hui 88]: Huitema, C.:
The X.500 Directory Service
Computer Networks and ISDN Systems, Vol. 16, pp. 161-6, 1988

[ISO 88]: International Standards Organization
ISO IS 9594, Information Processing Systems - Open Systems Interconnection - The Directory Part 1 - 8, 1988

[ISO 89]: International Standards Organization
ISO SC21 N3569, On issues of the working document on the directory schema, Oslo 1989

[Jak 89]: Jakobs, K.:
ISO's Directory Proposal - Does It Require Enhancement?
to be published in Proc. 9th Int. Conf. on Remote Data Processing DPD '89, Karlovy Vary 1989

[Kre 89]: Kremer, W. et al.:
Entwurf einer Netzwerktopologie für ein Mobilfunknetz zur Unterstützung des öffentlichen Straßenverkehrs (in German)
Proc. Communication in distributed systems, Stuttgart, Springer Verlag 1989

[Neu 89]: Neufeld, G.W.:
Descriptive Names in X.500
Proc. SIGCOMM '89, ACM Press 1989

SESSION 3

MANAGEMENT AND MONITORING OF NETWORKS

A TRAFFIC GENERATING AND MONITORING TECHNIQUE FOR STANDARD AND INTEGRATED VOICE/DATA ETHERNET LANS

GY. CSOPAKI, T. MOLNÁR, CS. SZABÓ AND L. SZANDI

Technical University of Budapest
Stoczek 2, Budapest, Hungary, H-1111

Although the majority of existing Ethernet installations are slightly loaded, large-scale, heavily loaded networks are emerging. In such cases, estimating user performance measures of a large network is a task of great importance before that network is actually built.
The proposed novel technique produces artificial traffic which is highly similar to the composite traffic of a programmable number of real workstations with programmable activity parameters. It supports analysing traffic performances not only of standard Ethernet LANs, but also integrated voice/data local area networks. The software package includes a second group of measurement and monitoring functions, as well.

1. INTRODUCTION

Performance measurement in local area networks is an issue of rapidly increasing importance because of at least two factors. Firstly, a local area network has been often purchased in the past just because it was fashionable, and was not fully integrated into the everyday life of the given office, factory site, research department etc. Now LANs and their users seem to leave this childhood and enter the adult age, when someone buys a LAN if there is a real need in it. In this case, the installers and users want to know precisely what they can expect from the LAN they chose, in terms of the number of workstations which can be served by the LAN, in terms of response times etc.

Secondly, local area networks are becoming more and more utilized. The number of large LANs is increasing (several dozen stations is typical nowadays instead of a dozen which was the average number in the early years), and, on the other hand, the traffic generated by a single station is much greater than it was earlier because of the transition from personal computers to high-performance workstations. Configuring a larger LAN consisting of several dozen up-to-date workstation is a complex task which has to be executed carefully, trying to estimate the performance, prior to building it or expanding an already existing smaller LAN, with a reasonable accuracy.

Existing tools for analysing LAN performance are almost exclusively *traffic monitors* (which can be used for troubleshooting and maintenance purposes, too). Examples are some well-known off-the-shelf instruments made by BICC ISOLAN, EXCELAN [1], Hewlett-Packard etc. More sophisticated traffic measuring techniques and tools have been reported in an IEEE Network Special Issue, see e. g. [2], [3].

An essential function in performance measurements is *generating artificial traffic* with given characterictics. The network response and

the response of the individual stations can then be observed, the corresponding data gathered and processed by traffic monitoring stations, and a subjective judgement is possible by real users at real workstations while the system is loaded with the desired amount of artificial background traffic. This approach is particularly advantageous especially when an already existing "small" LAN is to be expanded. In this case, measurements can be carried out by introducing artificial traffic which is generated by modelling the expected load of the expanded system but before it has actually been built up.

Known LAN performance monitors support only sligthly this kind of performance estimation. There is usually a simple generating function built in in these instruments, which allows for generating periodically packets with a fixed of a user-definable content (e. g., destination address). This function is used primarily for testing individual stations of the network.

A novel feature of the performance measurement technique proposed in this paper is its advanced traffic generating capability. It is not only possible to generate packet sequences with constant or randomized interarrival times, but there is also a possibility of generating artificial traffic using the record made during the run of a special purpose simulation system PAPSYM [4]. This way, a rough simulation model can be constructed first, a series of simulations can be executed, and, using the results, the system can be loaded with a traffic characterized by more realistic parameters.

Another novelty of our technique is that it supports investigating not only standard Ethernets, but also a class of Ethernet-based integrated voice/data LANs. Using LANs for integrating voice and data services in a local environment is relatively new approach which has some advantages as compared to the trivial PABX solution. LAN-based and PBX-based solutions can be considered as alternatives, or, their combination can turn to be the optimal solution in a given environment. In Sec. 2, we shall biefly introduce the most important class of integrated voice/data protocols that are supported by our traffic generator in such a way that data thoughput-delay performance can be estimated when the system is loaded by a given number of voice connections.

The system, described as follows, comprise some measurement and monitoring fuctions, too, which are common in factory-made LAN analysers. Our goal was not an attempt to compete with these instruments, rather, only to complement the main traffic generation functions with some useful measurement ones, in order to support both groups of functions in a single software package.

2. INTEGRATING VOICE AND DATA ON ETHERNET

When implementing a real-time voice transmission service on a LAN, and thinking of connecting the LAN to the public telephone network (which is a natural extension), one faces the PTT recommendations on the quality of transmitted voice. This results in requiring a sevice from the medium access protocol a service which is completely different from that of data transmission, namely, the medium access sublayer must voice packets within a fixed (and low) delay, and, possibly, with no or negligible packet loss.

The standard CSMA/CD-type protocol is obviously not suitable because of its inherent random access nature, so there is a need in elaborating specific "integrated" medium access protocols (which will be implemented in specific "integrated" controllers) to ensure a specified delay with no packet loss. An additional requirement is also reasonable: the new

integrated protocol must be compatible with the standard data protocol in the sense that they utilize the same physical level information (carrier sensing and collision detection) and the integrated controllers can coexist with the standard ones on the same bus. As far as the physical implementation is concerned, it is desirable that the integrated protocols could be implemented by using the standard VLSI CSMA/CD controllers with as small amount of additional hardware as possible.

The most promising technique for integrating voice and data at the medium access layer is using the so-called train protocols, when, roughly speaking, a given portion of the time is used exclusively by voice sources and the remaining time is granted for data using the standard Ethernet protocol. This means organizing strings of packets from voice sources which will occupy a part of the so-called packetization period of the voice source. Data sources which follow the standard CSMA/CD protocol, must sense carrier during the string and, therefore, will not disturb it.

An appropriate mechanism to form "trains" from voice packets can be either (i) "hardware handshaking" between succeeding packets of the string, adding a sufficiently long header and trailer to each voice packet or (ii) a distributed timing mechanism ensuring that a given voice station can uniquely determine the time when it is allowed to transmit. For a detailed description and adequate references, see e. g. [5] and [6].

To simulate voice trains and to generate them artificially on an Ethernet system it is sufficient to periodically send a "large" packet the length of which corresponds to a given number of voice channels.

3. TRAFFIC GENERATOR

The estimating user performance measures of a large network is a taskof great importance before that network is actually built. The Ethernettraffic generator is intended to be a useful tool to accomplish this task.It produces artifical traffic which is highly similar to the compositetraffic of a programmable number of real workstations with programmableactivity parameters. The traffic generator (TRAGEN, TRAffic GENerator) is basically a software whichis implemented in an IBM PC XT/AT equipped with an Ethernet controllerboard and connected to the existing network in the same way as any other workstation.

The user has the possibility to specificate the traffic parameters or may use the output results of a simulation system, that generated traffic in advance for several number of workstations. There is possibility for the user to define special packets, that can be followed at the workstations. In this manner the user can generate special situations. With the traffic analyser the user may follow the packets and may observe the utilization of the channel and the different workstations.

The possible form of different traffic:

- periodic traffic;
- traffic with exponential distribution function;
- integrated voice/data traffic.

The generated traffic can be stored into a file and from there in a real-time mode the traffic can be generated into the channel.

There is another possibility to produce a file, when the file is generated with a simulation program.

During the traffic generation the user may define the following attributes:

- in case of periodic data
  - time interval

- length of packets
- type of packets
- duration of generation
- start time of generation
- source address
- destination address

-in case of non periodic data

- intensity of packet load
- length and type of packets
- duration of generation
- start time of generation
- source address
- destination address

The features of the traffic generator and traffic analyzer can be followed in the Fig.1. These utilities can be defined after the selection of traffic generation in the Fig.2. Here the user can define the generation into a file or can send the generated data directly into a file or can return into the operating system, because the whole menu-driven system is organized on a hierarchical manner (Fig. 3). If the user select the traffic generation directly to the channel ("Send data to the channel"), the following possibilities can be selected:

- traffic generation from file
- traffic generation directly to the channel, based on the user specification
- maximal traffic that is allowed by the hardware
- editing special packets and send to the channel.

In case the user select the traffic generation from file, the time will be written continouosly on the screen after the TIME keyword has been typed and the name of file containing the data for traffic generation can also be defined. Here the user can define the duration of generation.

When the "Generate from memory" is selected the user can specify that the interarrival time will be periodic or be based on exponential distribution function. The specification of exponential data traffic includes the definition of duration time, the average of free time in channel and the data length. In case of integrated voice/data traffic the user can specify the number of speaking partners.

The menu-driven input program will provide the syntactically correct data specification. If there is any error during the data input, an error message will be generated and the user should correct the error.

When the user activates the maximum traffic generation the message "Please wait during the generation", will be displayed, and the user will be informed about the time of termination of the traffic generation.

Special packets can be edited by selecting the "Generate specified data" bar from the menu "Traffic generation to channel". In this case (Fig.4.) the source and destination address, the packet type and length and the content of the packet can be specified. The content can be defined by ASCII characters or hexadecimal data.

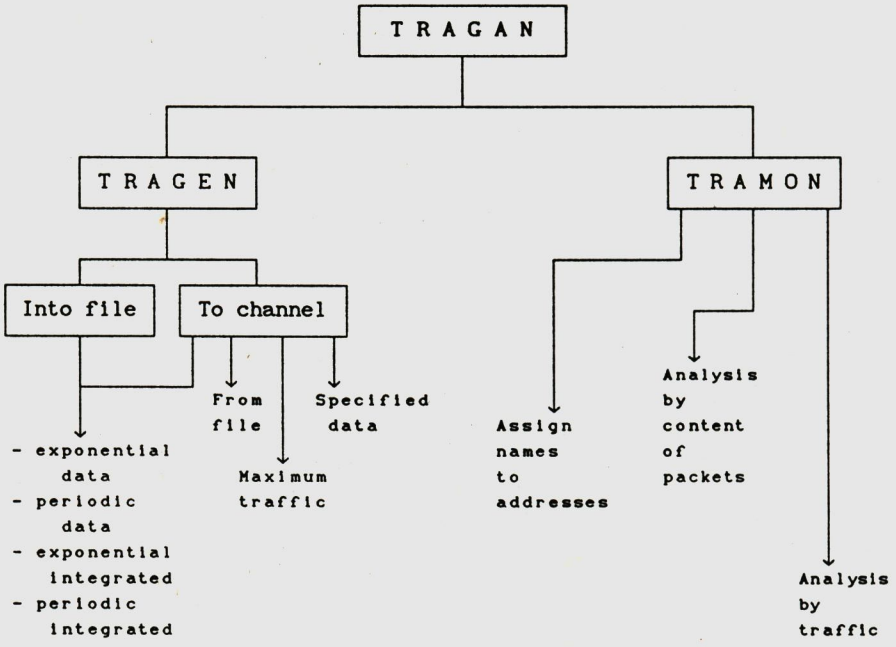


Figure 1.

The architecture of the TRAGAN program

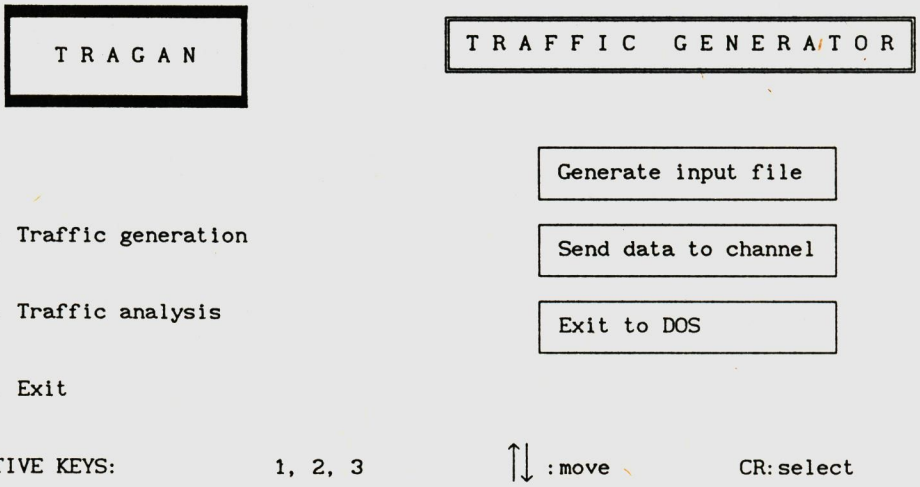


Figure 2.

The start of the traffic generator and monitor (TRAGAN)

Figure 3.

The first picture of traffic generator

Definition of a specified packet

Destination address: 22-22-22-22-22-22 Packet type: 00-80

Source address: 44-44-44-44-44-44 Data lenght: 500

DEC HEX	D A T A	A S C I I
0000 000	54 68 69 73 69 73 20-6F 6E 65 70 61 63 6B	This is one pack
0016 010	65 74 00 00 00 00 00-00 00 00 00 00 00 00	et.....
0032 020	00 00 00 00 00 00 00-00 00 00 00 00 00 00	.....
0048 030	00 00 00 00 00 00 00-00 00 00 00 00 00 00	.....
0064 040	00 00 00 00 00 00 00-00 00 00 00 00 00 00	.....
0080 050	00 00 00 00 00 00 00-00 00 00 00 00 00 00	.....
0096 060	00 00 00 00 00 00 00-00 00 00 00 00 00 00	.....
0112 070	00 00 00 00 00 00 00-00 00 00 00 00 00 00	.....
0128 080	00 00 00 00 00 00 00-00 00 00 00 00 00 00	.....

F1:base data F2:data field F3:ASCII field F4:send packet Esc:return

Figure 4.

The definition of specified data

4. TRAFFIC ANALYSER

The aim of traffic analyser (TRAMON, TRAffic MONitoring) is to monitor the behaviour of thenetwork, to detect and check errors. During the traffic analysing there is possibility:

- to filter packets in connection with the contents of packets and to store those (Analysis by content of packets);
- to monitor the whole traffic and to display the collected data (Analysis by traffic);
- to assign symbolic names to physical addresses (Assign names to addresses).

The user can specify names to addresses. For filtering of packets can be specified (Fig.5.):

- source address
- destination address
- type or inverse type

It is not compulsory to give the stop condition, because the user has the possibility to stop the process any time by hitting any key and after that the content of packets can be displayed.

When the user selects of "Analysis by traffic", any filtering condition can be specified and the monitoring will stop after that by pressing any key. During the monitoring the gathered information will continuously be displayed (Fig.6.). From these menus return will be made

to level "Traffic Analysis" and from where it is possible to return to DOS.

Giving the filtering conditions

Number	Source addresses	Destination addresses
1	44-44-44-44-44-44	FB-FB-FB-FB-FB-FB
2	67-67-76-76-76-77	5D-46-7A-9F-8C-D8
3	44-66-44-66-44-66	00-00-00-00-00-00
4	00-00-00-00-00-00	00-00-00-00-00-00
5	00-00-00-00-00-00	00-00-00-00-00-00

Filtering by type Filtering by inverse type No filtering by type

Type: 80-00

<->: move CR: ready Esc: abort

Figure 5.

Giving the special filtering conditions

Analysis by traffic

Source addr.	Σpack.	Σbyte	Dest. addr.	Σpack.	Σbyte
22-22-22-22-22-22	5	5000	44-44-44-44-44-44	5	5000
44-44-44-44-44-44	2	1000	22-22-22-22-22-22	2	1000

Number of error packets: Number of missed packets:

Press any key to stop the measuring!

Figure 6.

Analysis by traffic

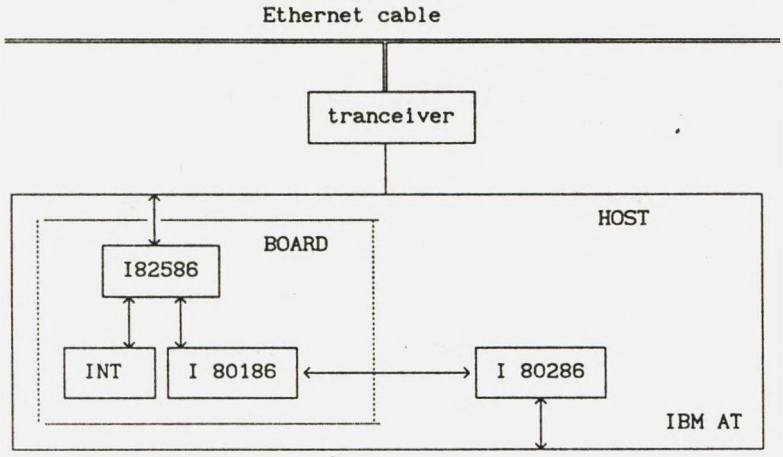


Figure 7.

The architecture of the workstation

5. IMPLEMENTATION

The traffic generator and analyser software can run on any IBM PC or a compatible machine which is equipped with an intelligent Ethernet controller board using Intel type 82586 controllers. Versions for other programmable controller chips can also be elaborated. The architecture of such a workstation is shown in Fig.7. The present implementation is based on the ACCORD's S-CORE board.

The software is written partly in Assembly and partly in Pascal and can be run under MS-DOS.

REFERENCES

[1] S. Spanier, "Design and Implementation of a LAN monitoring tool", Computer Communications, Vol. 11, No. 2, pp. 85-89, April 1988.

[2] D. Jacobson et al, "A master/slave monitoring measurement technique for an operating Ethernet network," IEEE Network, Vol. 1, No. 3, pp. 40-48, July 1987.

[3] M. Soha, "A distributed approach to LAN monitoring using intelligent high performance monitors," IEEE Network, Vol. 1, No. 3, pp. 13-20, July 1987.

[4] L. Jereb and L. Szandi, "Simulation method and program package for analysing traffic in integrated service local area networks," Report, October 1987 (in Hungarian).

[5] L. Fratta and Cs. Szabó, "CSMA/CD compatibel protocols to support real-time voice transmission," Proc. MELECON'87, 87 CH 2425-7, pp. 203-206.

[6] V. Garai, Cs. Szabó and T. Szép, "New multiple access algorithms for integrating voice and data on local area networks," Probl. Contr. and Info Thy, Vol. 17, No 6, pp. 381-396, 1988.

NETWORK MANAGEMENT IN A HETEREOGENEOUS COMPUTER CENTRE

D. GOLLNICK and H. LAMPRECHT

VEB Leitzentrum fuer Anwendungsforschung
Softwarebetrieb CAD/CAM
Jacques-Duclos-Str. 47/52
1156 Berlin
German Democratic Republic

The problems of network management become a main issue in a heterogeneous computer environment. Based upon practical requirements and in consistency with international development directed to standardized services of network management the basis of the network management tool MANset is demonstrated in this paper. In accordance with the vendor's DECnet compatible network architecture completed with modules for the integration of Personal Computers and Unified System (EC) mainframe computers, a gradual migration directed to OSI oriented solutions will be realized.

1. CONDITIONS AND MOTIVATION

Nowadays a computer centre rarely consists of homogeneous computers. Computer networks of different philosophies unite these heterogeneous capacities in the computer centre environment.

An economical, reliable computer centre management requires a qualified network management, which really means that network management becomes the central, regulating part as to operating, administrating, tuning and planning the computer centre.

At least 3 computer architectures are dominating in GDR:

- Mainframes of type EC.
  Most of them are compatible in a defined manner with IBM VM/370-Architecture.
- Minicomputers of CM-series.
  These series are very close to DEC's PDP- and VAX-series.
- Personal Computers.
  Most of them are fully IBM-compatible.

In the time passed away EC mainframe computers have been installed above all. That is why the software for economical data management and data bases existing on this architecture are very propagated.

A growing number of computer centres gets by use of the Robotrons 32-bit CM minicomputer K1840 and following an efficient, DECnet compatible network software.

Personal Computers, of course, are mainly used like working place computing capabilities. By the fastly growing number of Personal Computers the demand becomes more and more actual for a powerful heterogeneous network management.

Ranging as a base for physical communications there are the following mediums:

- Synchronous and asynchronous lines with V.24 (RS 232)-interface for low speed of transmission.
- ROLANET2, an Ethernet-based LAN, provided with communication facilities for CM minicomputers and Personal Computers.
- KIF-controller, which realizes communication between the EC mainframe computer's KIF (Small Interface of EC terminal controller) and the standard CM series bus providing relative high transmission rates in comparison with V.24-interface.

As communication software a software based on DECnet protocols (uniNet for Personal Computers, CMNET for K1840) and a special gateway software for communication between EC mainframe and DECnet compatible network software CMNET is used above all.

Particularly the DEC's network protocols and management tools are used for the communication between CM computers. There is no possibility for including the architecture of EC mainframe computers in the network management of DEC network architecture (DNA).

By developing a concept for the tool MANset presented in this article an attempt is made to meet the following requirements of network management in a computer centre:

- integration of most important network objects
- integration of all management functions (operating, managing, tuning, planning)
- guarantee flexibility and migration to standards
- gradual tool development by using already existing means.

2. RELATED WORK

After intensive occupation with protocols and services within homogeneous and also heterogeneous architectures in the last few years great attention is paid to the gradual realization of transparency in homogeneous environment as well as in the commercial and academic fields.

So IBM has published basic principles and protocols of its network management solution standard named NetView (/1/). By this way non-SNA- and non-IBM-compatibles can take part in IBM network management. Therefore IBM tries to stabilize its quasi-industrial standard NetView. NetView embodies a hierarchical three-point architecture including focal point, entry points and service points. A service point is a special type of entry point which forwards alerts from non-SNA devices. A number of vendors supplies NetView-based products using the service point mechanism.

AT&T and Digital Equipment have admitted migration in direction to standardized solutions by their announcements of Unified Network Management Architecture (UNMA, /2/) especially by DEC Network Architecture (DNA) phase V (/3/).

Hewlett-Packard's OpenView Network Management Architecture suggests a strategy for development which provides leveraging and coexistence of both OSI and non-OSI management solutions (/4/). At this moment in SC 21 of JTC 1 of ISO/IEC work related to network management is the most intensive one. Nevertheless in regard of the working time table of WG 4 and in regard of the experiences in standards stable standards for all five network management functional areas cannot be expected earlier than in the mid-nineties.

Under the conditions in GDR computer centres given above a solution similar to NetView is not possible. That is mainly caused by the existing communication hardware of EC mainframe computers. The inclusion of EC mainframe computers and Personal Computers in a heterogeneous computer network based on DECnet protocols is a solution be realized. Thereby a development towards standardized solutions is to be guaranteed.

Except for the Management Framework standard (/5/) there are no sufficient stable documents of management standard at this moment. An implementation based on the existing nonstable documents would deliver less functionality than already existing commercial solutions. It would be an interim solution without any future. This is not to be accepted.

3. ARCHITECTURE OF MANSET

Considering national conditions described in (1), the international experiences discussed in (2) and the demands mentioned above, a concept named MANset for network management in computer centres was developed to get a network management solution MANset.
Describing the MANset concept the term "managed object" is an important one and represents the computer network object be operated, administrated, tuned and planned.

Management activities require an extensive information base. It is required to store information on network configuration, current and historic performance, trouble events, security parameters, and accounting information. Therefore the management data base MANdb is the central part of MANset and consists of two fundamental components:

- "Managed Object Data Base", including describers of managed objects and attributes.

- "Event Data Base", including values of attributes which are originated from operating, administrating, tuning and planning of the network.

In MANdb entries are attached to the following functional area groups:

- configuration management

- fault management

- accounting mangement

- tuning management

- security management.

Between the entries of different functional area groups in the "Managed Object Data Base" there may exist relations. These relations are also described in MANdb, too.

Because of the advantages of a highly centralized network management in a computer centre (/6/) the main components of MANdb are attached to a special node. That means the administrative centre named MANct. The term "node" is usually interpreted as a "Managed Object" with processing capacity. In every node processes of the type "network management agent" (MANag) exist. These processes realize the possible functions decentralized. User initiated collecting of data or delivery of the network management data possibly filtered according to criteria are the minimum functions of MANag.

In the management centre named MANct the server process MANsv processes the relevant management data transmitted by the MANag processes and stores them into the "Event Data Base" according to the structure and rules given by MANdb.

The user interface MANif is the part which analyzes the MANdb corresponding to actual, periodical and visualizing information.
In accordance with functional area groups one can get information for operator, reports or carry out operating, managing, tuning or planning of the network. Independently from the contents of the visualized information MANif delivers an uniform layout for the user (operator, administrator) and guarantees a high integration in network management of computer centre.
The analysis of MANdb is effected by the module MANst for studying the information data base.

MANif is capable of displaying multiple windows simultaneously. The MANif start up display has two windows: a network configuration window and a command area window. In the configuration window, those devices which are "active" are distinguished from those that are "inactive". Selecting a special device (respectively other managed object) by the mouse the user can get more detailed information even in the sense of current or statistical meaning.

For relations between the main components of MANset see figure 1.

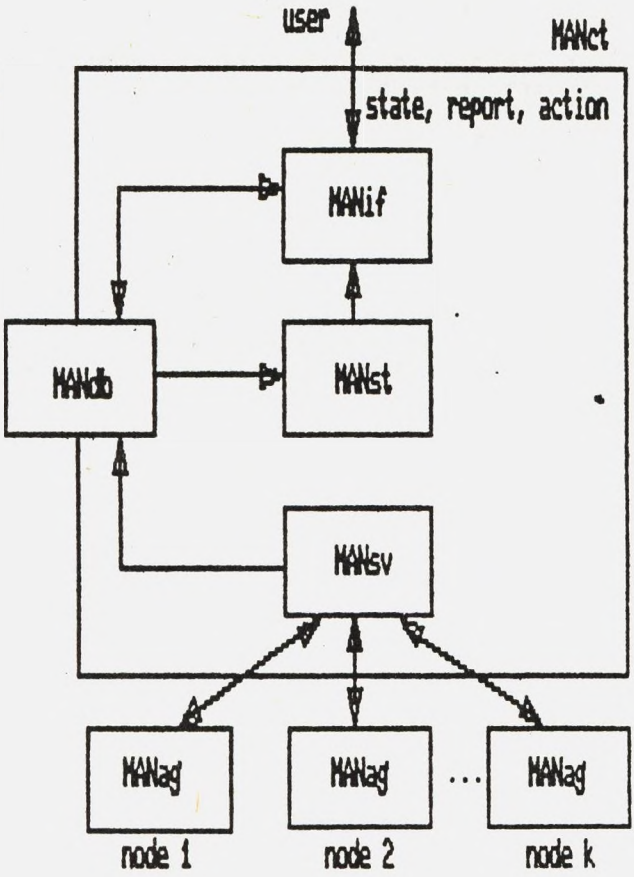


Fig. 1: MANset architecture

Regarding to (1) in today's computer centres new management components developed by customer have to be integrated into existing vendor-based structures and tools. In the case of MANset concept this demand is realized by including vendor architecture related components (of IBM's SNA, DEC's DNA, Robotron's RNA and others) in MANdb, MANsv, MANag.

4. IMPLEMENTING MANSET

Matching the concrete conditions the managed objects with their attributes and relations between attributes have to be defined. Extensibility of MANdb records is an important demand in implementation strategy.

See figure 2 for having an example of MANdb records.

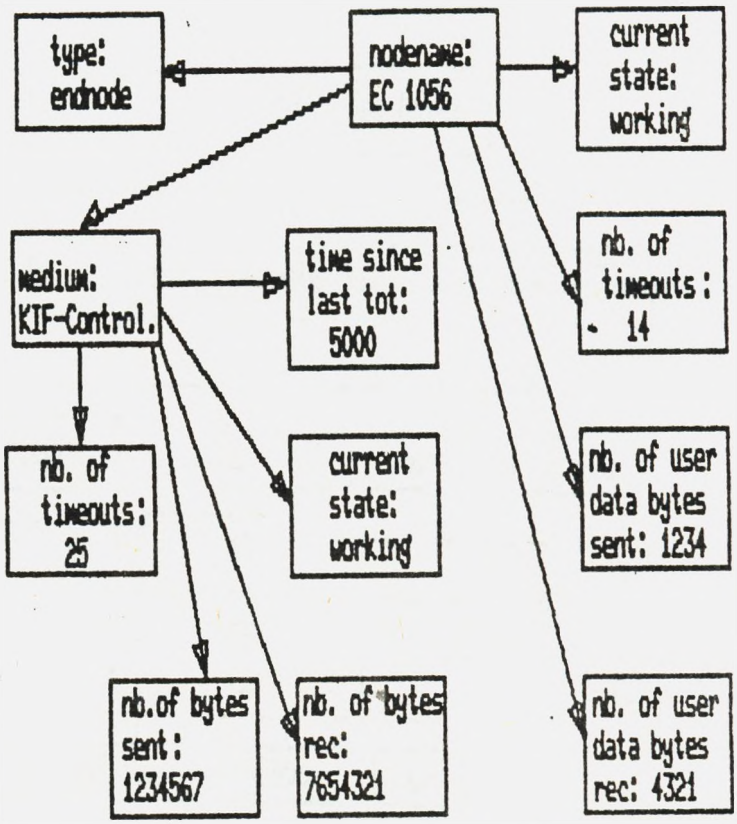


Fig. 2: Keyword relations in MANdb (example)

As discussed in (3) there has to be a basic vendor architecture existing in a heterogeneous network. For reasons described above DECnet-based protocols were selected as the vendor network base in our implementation concept of MANset.

Thus a Managed Object Data Base and an Event Data Base consist of a part "genuine database" known from DECnet and a part named "extended database". Using "extended database" it is possible to include elements of non-DECnet-architecture in MANdb.

For reasons of practicability (for instance data collection in other nodes) MANdb is implemented as a distributed database system. Because of DECnet compatible network protocol dominance MANct is located on a CM architecture minicomputer.

A 32-bit-based network management workstation will be the backbone of network management in MANset implementation. MANif is essentially based on the X-window standard (/7/) and has thus the qualities of environment portability and extensibility.

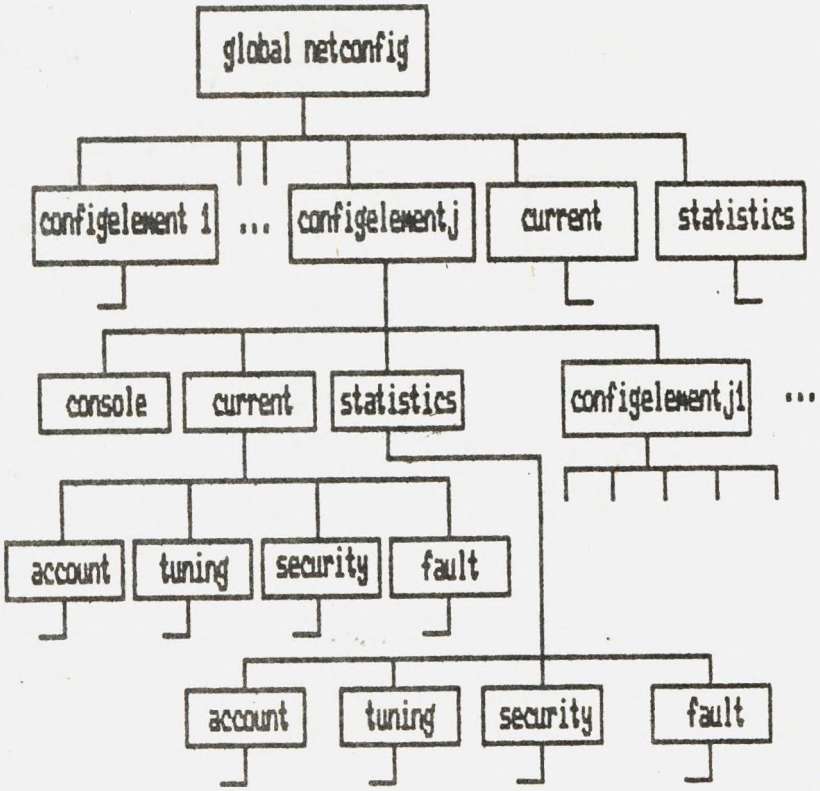


Fig. 3: Information tree of MANif

Figure 3 shows a scheme of information window hierarchy which can be used by the network (computer centre) operator. In the case of emergency events and faults the operator immediatly gets information about this by special icon signals.
Using the possibilities of the window hierarchy he can get more detailed information about event conditions.

With the aim of noncomplicated MANif extension a tool set is to be developed and included between the window system and the user interface modules. Thereby more modularity and more comfortable programming work will be achieved.

5. MIGRATION TO STANDARD SOLUTIONS

There are advantages while using any standard for network management integration, and more extensive advantages when an international standard based on the OSI Basic Reference Model is used. These include the following:

- A common communications platform permits one network for OSI user applications and network management.
- Network management expenses can be reduced through common, reliable specifications and software.
- Open naming conventions and standard data fields make management in a multivendor environment easier.
- There is increased demand for intelligent analysis and exploitation of expert system technology.
- Future OSI enhancements can be used to support network management.

Today there is no important vendor who denies not working in OSI-standard-direction. According to these announcements the communication software developer could wait for new releases of commercial vendor software. Nevertheless, marketing practice and concrete problems of heterogeneous environment (in our case the task of including EC mainframe computers in integrated computer centre management) do not allow a way such as this.

Realizing the concept discussed in (3) and using the DECnet movement towards OSI we expect less problems going from non-standardized MANset to a fully standardized MANset implementation. As described above developing of new components of MANset (as MANsv, MANag and other management components) shall be done in view of actual standard documents. A fully standardized network solution cannot be developed very soon because of nonstabilized standards.

6. CONCLUSIONS

The MANset concept will be realized in stepwise manner. The main tasks in refinement and extention are:

- data management using a DBMS
- use of expert system tools for MANst and further for network management automation
- extending MANset by complex systems for network installation and network consulting.

The necessity for using a DBMS is clearly seen while working on MANset development.

Further improvement of hardware functionality and productivity is the base for further development in the area of network management workstations. According to higher possibilities in graphical and computer capabilities MANif and MANst will get elements of artificial intelligence.

A heterogeneous computer centre environment requires architecture and implementation of network management with possibility of using commercially vendor-based components migrating in direction to Open Systems Interconnection standard.

By developing on management software MANset described above we hope to have found the solution of our management problems in a real environment.

REFERENCES

/1/ Willett M.,Martin, R.D., LAN Management in an IBM Framework, IEEE Network, 2(1988)6-12

/2/ Cohen, S.R., Kan, H.K., Raymond, J.P., Unified Network Management from AT&T, AT&T Tec. J., (Nov./Dec. 1988)121-135

/3/ Gottschalk, G., Manageable Multivendor Networks, in: Enterprise Event, proc. (Baltimore, 1988) 5-149-5-161

/4/ Klemba, K.S., OpenView's Architectural Models, in: Integrated Network Management, proc. (Boston, 1989) vol. 1 565-572

/5/ ISO/IEC JTC 1/SC21, Final Text of DIS 7498-4: Inform. Proc. Systems - OSI - BRM - Part 4: Management Framework, (April 1989)

/6/ Terplan, K., Kommunikationsnetze: Planung, Organisation, Betrieb, (Hanser/Prentice-Hall Intern., Muenchen, Wien, London, 1989)

/7/ Scheifler, R.W., Gettys, J., The X Window System, ACM Tr. Graph. 2(1986)79-109

AN EXPERIMENTAL OSI NETWORK MANAGEMENT IMPLEMENTATION FOR A PC NETWORK

Klaus Garbe

Informatics Center
Technical University Dresden
Dresden, GDR

ABSTRACT

In the recent few years, a remarkable progress in the field of OSI-standardization of network management can be observed.

A growing number of particular standards documents is being issued on architectural aspects as well as on services and protocols. Quite a lot of problems is still under discussion, and as a result some changes of the proposed functions can be expected. But nevertheless it is an interesting and necessary challenge to carry out first implementations on this base. This takes place in the r&d departments of several networks and software vendors, and it takes place in the research laboratories of some scientific institutions, too.

At the Informatics Center of the Dresden Technical University, a group of scientists and students have been working in the field of network management since 1982. Two experimental network management systems have been accomplished, one of them in the environment of a heterogeneous LAN including IBM and DEC compatible Computers, the other in the environment of a homogeneous LAN consisting of personal computers using a MS-DOS compatible operating system. The paper deals with the latter system. The implementation base, an experimental OSI software hierarchy is outlined. Then architecture, functions, the implemented CMIS-based services and protocols are discussed. Finally, some ideas about future developments are mentioned.

1. NETWORK MANAGEMENT PROBLEMS

For continuous operation, computer networks need some form of administration or management. Functions like activating and deactivating network software, monitoring the status of essential parameters, changing configuration, fault detection and diagnosis, event recording and security measures are to ensure high availability and steady conformance with user requirements. Network management appears to be a task of considerable complexity. It encompasses organizational as well as software and hardware aspects. This means, automatic acting facilities are not sufficient to perform network management. Human supervision and interaction is inevi-

table for a realistic and efficient approach. Thus, creation of automatic acting management components is necessary as well as tools for network administrators. As a matter of fact, management software is tightly interrelated with network software for normal user-initiated functions. This is one reason for the development of a network management architecture. For several other reasons it is useful to follow a worldwide unified approach for management architecture. One of them is the trend to internetworking, i.e. to interconnect several independent networks to a system. In such a case a need emerges for cooperation of several network administrators. This fact causes directly corresponding work in the standardizing bodies. A broad agreement could be reached with OSI model based management concepts. First industrial products claim to conform with the guidelines of OSI management.

2. A SHORT SURVEY ON OSI MANAGEMENT

Basic document of OSI management is the management framework [1]. It introduces three categories of management:

- systems management

- layer management

- management aspects of normal layer operation.

Responsibility for the latter two lies in the layer standards working groups. At present a process can be observed to become aware of the management needs of specific layers [2]. The largest amount of work is devoted to systems management, which comprises means for layer management, too. Systems management is responsible for overall monitoring, control and coordination of network activities. It is accomplished by systems management application entities (SMAE) in layer 7 and a systems management protocol between them. Every system in a network which is to be a subject of management must have a SMAE. The local management related environment embedded in which is a SMAE is a systems management process. Systems management processes can play either of two different roles: it may be a managing or managed process. Several systems may form a management domain. Overlapping of managing domains is feasible, in this case a management process of a system belonging to two different domains can play the role of a managed and a managing process as well [3]. All management relevant information forms conceptually the Management Information Base (MIB). For logical structuring the MIB an object oriented approach is being developped [4]. Systems management is divided into five functional areas:

- Configuration Management (CM)

- Fault Management (FM)

- Performance Management (PM)

- Accounting Management (AM)

- Security Management (SM).

Conceptual models for these areas are in a draft status, nevertheless several specific functions with relevance to more than one functional area have been isolated and documented in more detail. The most general functions for transferring management information constitute the Common Management Information Services and Protocols (CMIS/CMIP) [5]. They can be used to create or delete managed objects, to read (GET) or modify (SET) information in the MIB, to perform management related operations in remote systems (ACTION) and to collect information on relevant events (EVENT-REPORT). It can be expected that several changes will take place in near future upto final IS status. Nevertheless it seems to be useful to base first implementations on the above mentioned standards in order to get experience and to be able to evaluate newer versions. A short description of a case study of such a type is given below.

3. IMPLEMENTATION ENVIRONMENT AND GOALS

At present different working groups try to implement OSI protocol stacks on PC's [6], [7]. One of these consists of scientists and students of Informatics Center at Technical University Dresden, GDR. The goal of this work is to find out an agreeable way to integrate PC's in homogeneous or heterogeneous computer networks, whose software architecture is built according to the OSI model. Mainly, two specific problems arise in this context:

- PC's have scarce hardware resources in comparison with minicomputers or mainframes;

- as a rule PC's have a single-user, single-process operating system as for instance MS/DOS.

The first problem enforces a very efficient code and storage management or specific hardware solutions. The second problem is normally solved by building a multi-process shell around the native single-process operating system. In addition, the specific needs of a layered software system can be met by a buffer management, which supports efficient interchange of service and protocol data units in a multi-layer protocol hierarchy.

The hardware basis in our case consists of PC's of the types EC 1834 and A 7150 [8], [9], which are more or less XT-compatible. During the development it turned out that these PC's are too weak to support efficiently a full OSI protocol stack. Thus, we have to regard this hardware basis a an experimental one for development purposes. A portation of the software for its future use on more powerful PC's is planned.

The now used PC's are interconnected by means of a CSMA/CD-LAN, namely the ROLANET-1 product [10] with a transmission rate of 0,5 Mbps. As an operating system the MS/DOS-like DCP is used. The specific needs of a communication software packet are met by the SPSV system [11], which supports creating, suspending, resuming and deleting of processes, priority scheduling, event and time-out driven synchronization and buffer management. This SPSV system and all other software components of the OSI protocol hierarchy are written in the C programming language. On the base of an ISO 8802-2/3 like firmware implementation the following software layers have been realized:

- Transport layer according to ISO 8072,8073, classes 0,2,4;

- Session layer according to ISO 8326/8327, basic combined subset;

- Presentation layer according to ISO 8822/8823, kernel subset;

- File server according to ISO 8571 with certain constraints.

In addition to the file server another software system resides in the application layer. It is the systems management described below.

4. AN OSI-LIKE LAN-MANAGEMENT

4.1 Architecture

Usually a (single) human administrator is responsible for managing a LAN, a part or some specific aspect of it (a domain). He uses a dedicated station for his work, which has particular capabilities for this purpose. All other stations connected to the network in the realm of a given domain are objects of this activities. Thus, they have other management properties. Clearly, this situation corresponds to the role concept outlined in [3]. In our implementation, the active role is played by the management processor, whereas a management agent runs on all other stations. A command/message oriented administrator interface exists at the station with the management processor. Furthermore, there is the deposit for mass management data (logging facility). Between management processor and management agents a CMIP is executed. By means of a particular module, the network interface, an easy adaptation to the lower layers is to be accomplished. Upto now, direct interface to transport layer (in an experimental version without presentation and session layers) and normal interface to presentation layer were taken into account. A specific adressing mechanism allows access to layer objects to fulfil the management framework requirement to do most of layer management tasks by systems management. This adressing scheme is described in the fallowing clause in the context of the DISPLAY command.

4.2 Management functions

As mentioned in clause 2, management functions are usually grouped forming the five functional areas configuration management, fault management, performance management, accounting management, and security management. By examining the requirements of these functional areas in depth it appears that several general functions can be defined, which are useful in more than one or in every area at all. This way, ISO/IEC JTC1/SC21/WG4 has defined some general systems management functions like object management, state management, relationship management, management service control and others. In a similar manner, but with the aim to get more elementary functions which can be easily transformed into CMIS calls, in our implementation project some management functions have been specified. These functions are to fulfil simple but powerful management operations, initiated by an administrator command or by a programmed call. The latter way forms a possibility to build more complex management functions similar to the above mentioned ISO proposal. Each of our management functions can be unambiguously mapped onto a particular CMIS primitive.
The management functions hitherto defined are:

F1: Displaying management information

F2: Changing management information

F3: Starting software processes

F4: Stopping software processes

F5: Message transfer

F6: Event recording.

A short description of each function follows. For ease of description in this paper, the command syntax was somewhat simplified. It is to see as a prototype line oriented command system, which is the starting point for a more versatile window oriented interface now under development.

<u>Displaying management information</u>

The command format is:

$$\left\{\begin{array}{l}\text{DISPLAY}\\ \\ \text{DY}\end{array}\right\}\text{ES = esid,}\qquad \begin{array}{l}\text{ADR = address}\\ \text{NAME = symbol}\\ \text{LISTE = Listid}\end{array}$$

This command enables an administrator to specify any known endsystem with the ES=esid parameter (end system identifier), from which he wants to see management information. Management information can either be specified by a (physical) address, by a symbolic name or by a list ident. In the latter case a predefined report is displayed. In a preparatory step an

internal symbol table is created by using linker output. Thus, any external symbol of any C-module which is part of the communication software packet as a result of a single link pocedure can be accessed by the DISPLAY function. In this basic version no overlay structures ar allowed. In this manner a remarkable amount of layer management can be done. DISPLAY is mapped onto CMIS M-GET.

Modifying management information

The command format is:

$$\left\{\begin{matrix}\text{SET}\\ \text{SE}\end{matrix}\right\}\ \text{ES = esid,}\ \left\{\begin{matrix}\text{ADR = address}\\ \text{NAME = symbol}\end{matrix}\right\}$$

The old content of the specified management information is displayed. In can be directly changed on screen. After entry the modification is executed. It applies the same addressing mechanism as described with DISPLAY. SET is mapped onto CMIS M-SET.

Starting software processes

The command format is:

$$\left\{\begin{matrix}\text{START}\\ \text{ST}\end{matrix}\right\}\text{ES = esid, PROC = proctitle}$$

It activates a process with the name proctitle in the remote system. In the current implementation, a process is an activity in the sense of the above mentioned SPSV system. That means that the proctitle of a process which is subject of a START or STOP command must be a member of a process table internal to the SPSV system. This way, it is possible to accomplish software configuration processes or to initiate processes for testing, diagnosis or performance evaluation purposes. START is mapped onto a CMIS M-ACTION primitive.

Stopping software processes

The command format is:

$$\left\{\begin{matrix}\text{STOP}\\ \text{SP}\end{matrix}\right\}\ \text{ES = esid, PROC = proctitle [,SDEL] .}$$

It deactivates a process abruptly (SDEL) or after processing all outstanding events.

Message transfer

The command format is:

$$\left\{\begin{matrix}\text{SEND}\\ \text{SD}\end{matrix}\right\}\ \text{ES = esid, INFO = text.}$$

With a SEND command any appropriate character string can be transmitted from one administrator to another. It is mapped onto a CMIS M-EVENT-REPORT.

Event recording

In the present implementation, event recording can be performed by programs only. There are two forms, TRACE for recording subsequent events in wrap-around buffers in local (management agent) or remote (management processor) main memory. For later processing and analysis, trace data can be transformed to LOGs in secondary storage. Private and common logs can be defined. Trace and log files can be browsed or listed for different purposes.
For controlling the LOG/TRACE facility, some further commands are available.

5. ASSESSMENT AND FURTHER DEVELOPMENT GOALS

At present the system described above is in a complex testing phase. It is regarded as a basic mechanism to get experience and to derive further development goals. It is to serve as a laboratoy system to demonstrate possible forms of network management supplementary to lectures on computer networks theory. The simple administrator interface is appropriate for this purpose. For the use in real PC networks several enhancements are necessary, ecpecially concerning a more versatile interface for the network manager and the definition of a practical set of information forming a suitable management information base.
From a theoretical view, network management concepts are in a continuous evolution. As shown in clause 2, several new aspects of ISO management standards are worth for experimental implemen tation. Interesting new ideas like MIB as repository of all managed objects, the discriminator concept or the scope and filter function in CMIS are subjects for next investigations. Further tasks lie outside OSI. With respect to these, graphical administrator interfaces and KI techniques for decision support are to be mentioned.

REFERENCES

[1] Final Text of DIS 7498-4, Management Framework
ISO/IEC JTC1/SC21 N 3502. 24.4.1989

[2] Proposal for a New Work Item: Transport Layer Management.
ISO/IEC JTC1 N 201. 11.4.1988

[3] Systems Management Overview
ISO/IEC DP 10040. 7.2.1989

[4] Management Information Services. Structure of Management Information. Part 1: Management Information Model.
ISO/IEC DP 10165. 27.6.1989

[5] Management Information Service Definition. Part 2: Common Management Information Service.
ISO/IEC DIS 9595-2. 22.6.1989

[6] Mantelman, L.: OSI on PCs: Bringing the world to the desktop.
Data Communications International
April 1989 pp. 64-71

[7] Sommer, D.H.: The OSI-PC - an introduction
The International Open Systems 88: Online Publications.
Pinner. UK 1988. pp. 293-301

[8] Several papers on the PC EC 1834
rechentechnik/datenverarbeitung Berlin 28(1988)2

[9] Kerst, S. u. H. Riegel: Arbeitsplatzcomputer robotron A 7150.
Neue Technik im Buero, Berlin,. 31(1987)5. pp. 130-138

[10] Richter, J. u. B. Terpe: ROLANET 1- ein lokales Netz des VEB Kombinat Robotron
Neue Technik im Buero, Berlin, 30(1986)5 pp. 146-149

[11] Bez, Th.: Subprozess - Supervisor SPSV 2.0, Release 1.1 - Institut fuer Informatik und Rechentechnik. Akademie der Wissenschaften der DDR. Berlin 1989. Internal report.

Central System Support for Digital Switching Systems (CSS)
K.HOLZINGER

At the time of going to press, the final version has not arrived yet.

SESSION 4

PROTOCOL SPECIFICATION AND VERIFICATION

Formal specifications for protocols: Issues and experiences

Gregor v. Bochmann (University of Montreal),
Luigi Logrippo (Ottawa University) and
Behcet Sarikaya (Concordia University, Montreal)

February 1990

Abstract

With wide-spread acceptance of the ISO-OSI reference model and its standardized protocols in the areas of computer communication and information exchange, formal specifications have become an area of active research and development. This paper surveys issues and recent developments obtained mainly from our undergoing research. The discussion includes four important aspects of the area: protocol design and specification languages, validation of the resulting design and specification, implementation development and finally conformance testing and implementation assessment.

1. Introduction

Communication protocols are the rules that govern the communication between the different components within a distributed computer system. In order to organize the complexity of these rules, they are usually partitioned into a hierarchical structure of protocol layers, as examplified by the seven layers of the standardized OSI Reference Model [Somm 89, Larm 88].

As they develop, protocols must be described for many purposes. Early descriptions provide a reference for cooperation among designers of different parts of a protocol system. The design must be checked for logical correctness. Then the protocol must be implemented, and if the protocol is in wide use, many different implementations may have to be checked for compliance with a standard. Although narrative descriptions and informal walk-throughs are invaluable elements of this process, painful experience has shown that by themselves they are inadequate.

The informal techniques traditionally used to design and implement communication protocols have been largely successful, but have also yielded a disturbing number of errors or unexpected and undesirable behavior in most protocols. The use of specification written in natural language gives the illusion of being easily understood, but leads to lengthy and informal specifications which often contain ambiguities and are difficult to check for completeness and correctness. The arguments for the use of formal specification methods in the general context of software engineering [Somm 89] apply also to protocols.

The following activities can be identified within the protocol development process. These activities can be partially automated if a formal protocol specification is used [Boch 87c].

(a) Protocol design: The protocol specification is developed based on the communication service to be provided by the protocol. The protocol also depends on the underlying (existing) communication service; e.g. the protocol may have to recover from transmission errors or lost messages if the underlying service is unreliable. The design process is largely based on intuition.

(b) Protocol design validation: The protocol specification must be checked (1) for logical consistency, (2) to provide the requested communication service, and (3) to provide it with acceptable efficiency.

(c) Implementation development: The protocol implementation must satisfy the rules of the protocol specification; the implementation environment and the user requirements provide additional constraints to be satisfied by the implementation. The implementation may be realized in hardware or software.

(d) Conformance testing and implementation assessment: The purpose of conformance testing is to check that a protocol implementation conforms to the protocol specification, that is, that it satisfies all rules defined by the specification. This activity is especially important for interworking between independently developed implementations, as in the case of OSI standards. The testing of an implementation involves three sub-activities: (1) the selection of appropriate test cases, (2) the execution of the test cases on the implementation under test, and (3) the analysis of the results obtained during test execution. The sub-activities (1) and (3) use the protocol specification as a reference.

In the context of the OSI standardization process [OSI 83, Larm 88], so-called formal description techniques (FDT) have been developed to be used for the writing of formal specifications of OSI protocols and services. They are called Estelle [Budk 87], LOTOS [Bolo 87] and SDL [Beli 89]. After arousing initially much expectations, their use is still quite limited in the work of the standardization committees and in the wider context of industrial protocol development (for a discussion, see for instance [Boch 89n]).

Much research and development efforts have gone into the development of tools to be used in relation with these languages. Some of these efforts will be mentioned in the sections below. Some of the results obtained from these efforts are quite interesting and useful. In the following sections, we try to give an overview of the important results relevant to the different phases of the protocol development cycle, and to highlight the issues that must be addressed to make the use of formal specifications for communication protocols a practical approach.

This paper is written from the perspective of our own ongoing research projects, which are partly funded by a collaborative grant from the Natural Sciences and Engineering Research Council of Canada. We do not pretend to give a general overview of this broad area, but hope to provide a critical discussion of our research and its relevance in the general context of protocol engineering.

2. Protocol design and specification languages

2.1 Formal description techniques and their use

The purpose of the two FDTs that most concerns us in this paper, Estelle and LOTOS, was to support OSI's standardization process. The third comparable FDT, SDL, was developed at first as a specification language for telephony, and was enhanced to include protocol specification concepts for OSI at the same time that Estelle and LOTOS were developed.

Since the purpose of these languages was OSI specification, they should have been available much before the beginning of OSI standardization. What happened, instead, was that their design was started at the same time as OSI standardization, and they became

available only when the standards for the first six layers of OSI were substantially completed. This simple chronological problem explains partly the relatively low level of acceptance of these FDTs in the OSI context.

OSI standards were developed without using the FDTs, by the usual well-established informal specification methods: English text, combined with various types of diagrams and semiformal notation [Boch 89d]. Subsequently, small groups of experts set about to specify some OSI layers, and as of today the following layers have been specified by using the FDTs: A draft specification of the transport protocol has been developed in an earlier version of Estelle, however this project has been abandoned within ISO. This work has been completed within a SEDOS project in Europe [Diaz 89]. Specifications in LOTOS have been completed in the form of ISO Technical Reports for the Session protocol and service [ISO/IEC/TR9572 and ISO/IEC/TR9571]. Work is continuing, although at a slow pace, on Technical Reports for LOTOS specifications of the Transport protocol and service [Working Draft SC6/WG4 N421 and ISO/IEC/DTR10023]. As far as we know, no firm decisions have been taken yet to undertake the formal description of other layers, or to use the FDTs in the standards, such as application layer standards, that are currently being developed.

Experts who have used the existing formal specifications have reported that they are difficult to read. No real attempt has yet been made to use such specifications for other purposes, such as test case generation or verification. It is important to remember, however, that the complexity of these specifications is largely due to the inherent complexity of the described protocol standards, which are the result of many compromises. The use of FDTs during the standardization process could be helpful for avoiding such unnecessary complexity, since the complexities would immediately become apparent in the formal specifications of the standard proposals.

In order to provide a readable and reasonable complex example of a protocol specification, we have developed specifications in Estelle and LOTOS of a simplified class 2 Transport protocol [Boch 90]; a sketch of a specification in SDL is also given. The similarity of these different specifications provides some basis for the comparison of the three specification languages.

We believe that the following factors have a strong impact on the acceptance of formal methods in the area of protocol development, and similarly in the more general context of software engineering:

(1) User training: Assuming interest to learn a new specification language, it is important to provide adequate learning material. Tutorials for Estelle and LOTOS have only been available recently. Most complete example specifications of OSI protocols have been considered difficult to understand by the layperson, while some of the pedagogical examples, e.g. [FDT GL], have been considered irrelevant.

(2) Intuitive language features: It is very useful if the basic language features are easily understandable by the layperson. Also the use of graphic representations facilitates the initial acceptance of the language because of its intuitive flavor, especially for smaller specifications (although graphics often becomes cumbersome for larger specifications). Some examples of such features are FSM diagrams, entity-relationship diagrams for describing database structures, and inheritance relations in object-oriented languages. Graphic representations in SDL and the table-oriented structure of TTCN also provide an easy initial access to these languages. Currently, a graphic representation for LOTOS is being standardized.

(3) Relation between formal and informal specifications: Even the best formal specification will not replace an informal description of the specified system. The informal description will probably always remain the easier part to understand by the (human) designer and implementor. A straightforward relation between the informal and formal specifications will provide for easy cross-referencing between the two descriptions and promote the integration of the information provided by the two descriptions.

(4) Wide applicability: The FDTs seem to be applicable to other areas, in addition to the area of communication protocols; however, it is not clear how easily they can be adapted for writing object-oriented specifications. A wide applicability is an advantage, since the costs for development of support tools could be shared for a wide user community.

(5) Simple tools: Corresponding to intuitive language features (point (2) above), the functions provided by support tools should be intuitively easy to understand. In addition, it seems that the provision of a simple tool, possibly restricted in its functions, is better than the provision of a general tool which is difficult to use. For example, the reachability analysis tools based on FSM models have been found quite useful, although they are based on a restricted model and are not able to provide in practice a full analysis including interaction parameters.

(6) Integration into the general software/hardware development life cycle: Specifications are not used alone; as explained in Section 2.1, they are used throughout the protocol development cycle. Therefore the methods and tools related to formal specifications must be integrated with the other methods and tools used in the general software CASE or hardware CAD environment.

We expect that increased and more effective use of FDTs will result from increased knowledge of the languages in the community of protocol designers. It can then be expected that the specifier will start using FDTs early in the product or standard design phase, and the formal description will suit the product or standard much better than with the current "after-the-fact" methodology.

In the long term, as experienced with the FDTs increases, it will become more clear what are the desirable characteristics of such languages. Enhancements will be included in existing FDTs, and eventually better FDTs will be produced. However, judged by the example of what happened for programming languages, this evolution can be expected to be neither rapid, nor straightforward.

2.2 Example specifications

The results that we have had in our own work of application and enhancement of the existing FDTs have been encouraging and support the above conclusions. Following is a brief discussion of some of these experiences.

2.2.1 An industrial application of LOTOS

Gandalf is a Canadian supplier of communication equipments. Many types of standardized and proprietary protocols to coordinate and facilitate communication are an integral part of Gandalf's products. Informal or semiformal methods, such as various mixtures of English text, finite state machines, message diagrams, and program code fragments have previously been used to specify the protocols Gandalf has developed. These methods have resulted in errors, omissions and ambiguities being discovered during implementation and have

increased the cost of integrating new products and features into existing product lines. The errors appeared to be due at least in part to the imprecise semantics of the specification techniques mentioned above.

As a consequence, Gandalf wished to evaluate more formal specification methods. A pilot project was set up to this effect. In the first phase of this project, the language LOTOS was used for the specification of an existing Gandalf protocol, to see whether the language could be easily learned by protocol experts and to evaluate it as a specification method. In the second phase, LOTOS would then be used in the design of a new product. This application is, as far as we know, the first industrial application of LOTOS in North America.

The Gandalf project now has successfully passed the first phase. Gandalf specialists have learned LOTOS and have used it in order to specify certain aspects of a major company protocol, part of a distributed data PBX. It was not a simple protocol to specify, since it involved both software and hardware aspects, and since it had been developed incrementally over the years. LOTOS was found adequate for describing the architecture of the data switch both "in the large" (macro system design) and "in the small" (micro-system design). The project is now at the beginning stages of the second phase, where LOTOS are being applied in the development of a new protocol.

From this project, we are gaining valuable experience on the applications of LOTOS in an industrial environment [Logr 89]. In collaboration with Gandalf, we are planning to develop methodologies for the use of LOTOS through the whole life cycle of product development. Software tools and industrial methods will have to be developed, to help in the production and maintenance of specifications, in the development of implementations from specifications, and in the derivation of test suites for implementation. These are all current research topics, in which we hope that the continuation of the Gandalf project will provide contributions.

2.2.2 Different styles of LOTOS specifications

As most languages, LOTOS allows for a variety of different styles of using the language. The following styles have been identified in [Viss 88].

- Monolithic, where the specification is represented as a tree of choices. The parallel composition operator is not used in this style.
- State-oriented, where states are explicitly represented by using state variables.
- Constraint-oriented, where the specification is designed as the parallel composition of processes where each process separately enforces some set of behavior constraints.
- Resource-oriented, where parallel processes are chosen in such a way as to correspond to implementation modules.

The constraint-oriented style appears to be the most frequently used in the published OSI specifications mentioned above. It is our experience, however, that while this style works very well for specifying certain types of well-structured systems (such as the Transport Service or the telephone systems described in [Logr 90]), it leads to hard-to-read specifications when applied to complex systems where many cases have to be considered.

A researcher in our group has produced a complete LOTOS specification for the X.25 link layer protocol, LAP-B, including all options [Guer 89, Guer89a]. The specification consists of about 2 300 LOTOS lines. This specification constituted a useful exercise in LOTOS styles. In this specification, a mixture of resource-oriented, state-oriented, and monolithic styles was used. While unfortunately the unavailability of differently structured

LAP-B specifications makes it impossible to make comparisons, our specification appears to be fairly readable and has a reasonable degree of correspondence with the standard LAP-B specification. Furthermore, as mentioned in Section 5, the specification was successfully used for the derivation of test cases.

In view of these observations we believe that more experimentation is necessary on the question of LOTOS styles and their suitability in different contexts.

2.3 Semiformal description techniques in OSI

The semiformal techniques ASN.1 and TTCN are much more used in OSI than the FDTs. For the application of FDTs, it is therefore important to consider their relation with these semiformal techniques. As discussed below, there are still many unresolved issues.

2.3.1 ASN.1 (Abstract Syntax Notation One)

This is a notation for describing data structures [ASN.1, Neuf 90], similar to the data type definitions available in programming languages such as Pascal or ADA. It is applied to the description of OSI Application layer protocols, where it is used for the definition of the protocol data units (PDU's that is, the messages exchanged between different protocol entities). The notation includes a number of predefined data types, such as integers, reals, booleans, bit strings, octet strings and various kinds of character strings. It also allows the definition of composed data types, such as groups of elements (called SEQUENCE, corresponding to "record" in Pascal), a list of identical types (called SEQUENCE OF), a type of alternatives (called CHOICE, corresponding to Pascal's variant records), a TAG defining a code to distinguish between different alternatives, and others.

The main reason for the success of ASN.1 as specification language is probably the fact that it is combined with a standard encoding scheme for PDU's [ASN.1 C] which has been adopted for OSI Application layer protocols. Based on the information contained in the ASN.1 definition of the PDU structure, this scheme completely determines the PDU encoding, and can be used for implementing the encoding and decoding functions in a systematic manner, possibly automatically.

ASN.1 does not have the scope of an FDT, since it only describes data structures. However, it directly relates to the corresponding data structure definition facilities of the respective FDTs. Different scenarios for the interworking between ASN.1 and an FDT can be considered [Boch 89h]: (1) translation from ASN.1 into the corresponding FDT language constructs, (2) addition of the ASN.1 notation to the FDT, or (3) replacement of the corresponding FDT language constructs by the ASN.1 notation. The translation approach has been explored for Estelle [Boch 90c, Barb 89] and LOTOS [Boch 89h]. A similar approach could also be used for SDL. The approach allows the combination of tools for PDU encoding and decoding, based on ASN.1, with tools for implementation/simulation of Estelle, SDL or LOTOS specifications.

Most ASN.1 concepts can be easily translated into corresponding FDT concepts. However, the ASN.1 list structure (i.e. SEQUENCE OF) leads in Estelle to partly implementation-oriented data structures.

The experience with LOTOS [Boch 89h] shows that the abstract data type notation ACT ONE [Ehri 85], used in LOTOS, is very cumbersome for the description of simple data structures because of the lack of suitable notations. Unless an abbreviated notation for data

structures (see proposals in [Scol 87, Boch 89h]) is introduced into LOTOS, the definitions of data structures in LOTOS are of the order of 8 times longer than the corresponding definitions in ASN.1. This leads to very lengthy and unreadable descriptions of the PDU data structures.

2.3.2 TTCN (The Tree and Tabular Combined Notation)

This notation is relatively recent, and has been developed for the description of test cases for OSI conformance test suites [OSI C3]. As its name indicates, the language includes several different notations. The overall organization of the language is in terms of a collection of tables defining different aspects of a test case, such as service primitives, PDU's and their parameters, order of interactions, and constraints on parameter values. The interaction ordering is defined in terms of a conceptual tree where each branch represents a possible execution order. In addition to the tabular notation, a linear form of TTCN is developed for the exchange of test cases in machine-readable form. The ASN.1 notation can also be used for certain aspects of test descriptions.

When the need for a notation for OSI test cases arose around 1985, the responsible standardization subcommittee was not ready to adopt one of the developing FDTs for this purpose, which in the authors' opinion would have been a reasonable choice. Instead, a new language TTCN was developed, which seems in many respects quite "ad hoc". Its semantics is defined largely informally. In order to formally relate the defined test cases to the corresponding protocol specification, a definition of its semantics in terms of one of the FDTs would be desirable [Sari 88f].

As in the case of ASN.1, the application of FDTs in the context of OSI, requires a translation between TTCN and the FDT used for the protocol specification. As discussed in Section 5, translation from the FDT into TTCN is required when test cases are developed from the formal specification of the protocol. The validation of test cases in respect to a formal protocol specification also requires a translation. In some of our work we have explored the use of an intermediate formal language (so-called charts, see Section 5) which can provide a bridge between the different description techniques.

2.4 Object-oriented languages and other techniques

Object-oriented programming seems to be a buzz-word at the present time. Independently of its popularity, it seems that the object-oriented concepts of "object" and "inheritance" are in fact quite useful in the context of specification languages. Systems are conceived by humans in terms of "objects" which have an intuitive meaning, and the inheritance relations among the specified object classes provide a clear structure for specialization and generalization.

In the OSI standardization work on management of distributed systems and "Open Distributed Processing" (ODP), object-oriented description models are being used. For this purpose, two extensions of the ASN.1 notation are of particular interest:

(a) The notation for "remote operations" (ROSE) [OSI RO] which is used to define the operations which are provided by an object and can be invoked by other (remote) objects.

(b) A notation for defining object classes [OSI MO] including the concepts of object attributes and inheritance of properties among classes.

The writing of object-oriented specifications is not necessarily well supported by the FDTs [Cusa 89a, Blac 89]. In a research project in collaboration with BNR, we have developed an object-oriented specification language which combines certain aspects of LOTOS (e.g. rendezvous interactions) with inheritance and the view that "everything is an object"; in particular, no syntactic distinction is made between "data structures", abstract data types, and processes [Boch 89e]. We have also integrated certain concepts from (object-oriented) databases, which is important for real-time control applications. The intended initial application of this language is in the network management area; the development of a larger trial specification is in progress. A description of the architecture of the OSI reference model is discussed in [Mond 90].

At this point, one may wonder whether it is a good idea to develop a new language, instead of using an existing one. In particular, high-level logical descriptions can often be interpreted by logic programming languages, such as Prolog. Prolog has been found useful for the development of protocol validation and test selection tools [Sidh 88a], although as a specification language for communication protocols, it has certain disadvantages [Boch 85]. It has the advantage of being a general purpose language for which many tools already exist. We are presently experimenting with the use of Prolog for a knowledge-based approach to the specification of OSI application layer protocols, which facilitates the creation of a simple user interface for browsing through different parts of the specification, and the use of the same specification for simulation, test suite development and test result analysis [Boch 89j]. We also try to identify a process which leads in a systematic manner from the informal specification documents to the formal specification in Prolog.

3. Design validation

Several different techniques are known to do design validation. Most of them presuppose that the protocol is at least partly specified in some formalism. For example, some techniques content themselves with validating the general structure of the system. In such cases, only the aspects that should be verified need to be formally specified.

Other techniques attempt to verify all aspects of the system. In this case, the system needs to be entirely and rigorously specified in some FDT. No technique in this latter category, however, comes even close to validating real-life systems. Complete proofs have been developed for "toy" examples such as the well-known alternating bit protocol. Larger proofs become quickly extremely cumbersome.

However, if the entire specification is executable or may be simulated in some appropriate environment, it is possible to validate it by executing tests, that is, by performing and validating certain selected execution paths [Zafi 80]. This method will not cover all possibilities, however, by selecting appropriate paths to be tested, many errors can be found without too much effort.

Various execution and simulation environments have been developed for FDTs and other specification languages. In the case of Estelle and SDL, where tools for the semi-automatic generation of implementation code exist (see for instance [Boch 87c]), these tools may also be used for the validation of the specifications. Certain environments also provide for the simulation of performance parameters [Boch 87e] and/or provide facilities for observing the interactions exchanged between the different components of the specified system [Jard 85b]. It is also possible to combine the execution with the validation of predicates which are specified by the designer and should be satisfied in any execution of the system [Graf 89].

For the specification language LOTOS, two simulators have been developed independently [Eijk 88, Guil 88]. They both allow to execute the system in a step-by-step fashion, with the user providing the role of the environment and resolving nondeterministic choices. In an ongoing project [Saba 90], the simulation approach is used to validate a simplified specification of the Transport protocol [Boch 90] in respect to the OSI Transport service [OSI TS] written in LOTOS. A system of two interacting Transport entities and an underlying Network service are simulated by the LOTOS interpreter, and the resulting sequence of service primitives is validated against the service specification by using a trace analysis tool TETRA [Bell 89] initially developed for the automatic analysis of conformance test results (see Section 5.5).

In contrast to the simulation approach, the well-known reachability analysis explores systematically all possible execution paths. However, for realistic protocols, this technique is only applicable to an approximated specification restricted to a finite state machine model. In the case of LOTOS, certain tools only consider the so-called "basic LOTOS" subset which ignores interaction parameters. Related to CSP [Hoar 85] and CCS [Miln 80], this subset allows the application of similar exhaustive validation techniques, if the system is not too large [Najm 89] (also [Shir 89]).

A method in between reachability analysis and testing through simulation has been explored with a tool which provides for the generation of partial execution trees for LOTOS specifications including symbolically evaluated interaction parameters. This facility performs a sort of "eager evaluation" where all possible execution paths are explored, with symbolic values in lieu of the values to be provided by the environment. The symbolic tree evaluator has been enhanced to identify certain situations of repeated behavior [Guil 89]. A further step is a tool which provides the monolithic LOTOS specification corresponding to the tree [Quem 89, Ashk 90].

In the case of simple specifications, where the entire execution tree or monolithic specification can be calculated, these tools will provide a complete analysis of all possible behaviors of the specification. For example, deadlocks will be explicitly revealed by the tool, while other undesirable behaviors would be revealed by inspection. With larger specifications, however, the usefulness of this tool is greatly diminished by two factors. First of all, most such specifications branchout too quickly for the whole execution tree to be completed. Even worse, tests involving symbolic values cannot usually be evaluated to "true" or "false", thus a great number of unfeasible paths is generated. This problem can be reduced or limited by the use of appropriate specification styles [Guer 89a, Guer 89].

For the comparison of LOTOS specifications, several notions of equivalence may be used. Much work has been based on bi-simulation equivalence (e.g.[Miln 80, Miln 89, Brin 88, Najm 89, Shir 89]). However, the notion of testing equivalence, which is related to the "traces and refusals" of CSP [Hoar 85], seems to be more realistic, since it is based on whether two processes can be distinguished by executing testing experiments. Using such methods, it is possible to prove directly properties of traces (i.e. sequence of actions), such as: "the number of interactions at gate g will always exceed by one the number of interactions at gate h". Properties of this type are frequently stated for processes. Corresponding proof rules for LOTOS specifications have been developed [Gall 89]. Important immediate applications of verification principles have been found in the area of equivalence-preserving transformations. In LOTOS, a theory is being developed, supported by software tools, on how a specification written in one style can be transformed into an equivalent one written in another style. For example, one may wish to transform a specification written in an abstract, implementation-independent style, into another one that can be easily coded [Ashk 90, Quem 89, Viss 89, Eijk 89a, Leon 89].

Unfortunately by using these methods, it has only been possible so far to prove the correctness of small didactical examples. Semi-automated verification tools are being devised, but on the basis of what has been seen in the case of verification methods for functional programming languages, it can be assumed that the power of such tools will not go beyond proofs of specifications of a few hundred lines. This is an area where research is very active, however, some fundamental problems exist, by which progress can be expected to be slow.

4. Implementation development

The requirements to be satisfied by a protocol implementation include the protocol specification and usually additional constraints which are particular to the implementation project. These additional constraints may define such questions as "how does the implementation react to unexpected (invalid) user interactions?", "how many simultaneous connections should be supported?", or "what should be the performance of the implementation?". Based on these requirements, the implementation is usually developed in several steps of refinement using the standard software or hardware design and implementation methods.

In the case that a formal protocol specification is used, the additional constraints may be defined in the same formal specification language, and, depending on this language, may be translated into implementation code. The semi-automatic implementation based on Estelle specifications has been used in a number of cases. This topic has been covered relatively well in the existing literature [Boch 87h, Sidh 89]. Similar translation approaches can be used with other languages. In the case of LOTOS, it seems reasonable to impose certain language restrictions for the translation [Mana 89].

In the case that a multi-layer protocol structure is distributed over several operating system tasks or computers, that a communication service among several computers is to be simulated, or that a specified application is to be implemented in a distributed environment, it would be useful to have a uniform distributed implementation environment. In such an environment, a single system specification including multiple processes/modules could be automatically implemented by allocating each process/module to a particular operating system task or computer.

In the case of Estelle and SDL specifications, such a distributed implementation is relatively easy to obtain, since the processes/modules communicate by message passing, which is easily mapped on a message transmission communication facility available between the tasks/computers [Jard 89]. (For Estelle, the implementation is simplified if all dependents of a system activity or system process reside in the same task/computer).

In the case of LOTOS the situation is more difficult because of the rendezvous nature of interactions between the processes [Quem 89, Karj 88]. Several distributed algorithms for the implementation of rendezvous interactions between remote systems are known. The situation for LOTOS is particularly complex because more than two processes may participate in an interaction, and the processes involved cannot be determined statically. We have shown [Boch 89c] that a tree-oriented execution model for LOTOS can be combined with a virtual-ring rendezvous protocol, in order to obtain a scheme for the distributed implementation of LOTOS specifications. Present work is aimed at building a prototype implementation of a distributed LOTOS simulation environment, using multiple instances of an existing LOTOS interpreter [Logr 88], and at the realization of complete and fair interpretations in such a distributed environment.

As mentioned in Section 2, the integration with ASN.1 is an important issue for any specification language to be used for the OSI application-layer protocols. Some of our recent research projects were related to this question. While certain authors suggest the enhancement of existing specification languages with the data type notations of ASN.1, we have explored the possibility of translating the ASN.1 definitions of the PDU structures into the corresponding data type definitions of Estelle [Boch 90, Barb 89] and LOTOS [Boch 89h].

The translation for Estelle is relatively straightforward, except for the SEQUENCE OF construct which is translated into a list structure implemented by pointers. In order to obtain tools for supporting protocol implementations for this approach, we have built an ASN.1 translator and have adapted an existing ASN.1 tools which generates ASN.1 encoding and decoding routines [Yang 88] with an Estelle compiler [Este 87b] in such a manner that the C-code generated by the different tools is compatible and can be combined into a single protocol implementation.

The encoding and decoding of ASN.1 PDU's is a relatively tedious task that can be implemented in specialized hardware in order to obtain increased efficiency [Bilg 90]. It is to be noted, however, that the encoding/decoding modules (in software or hardware) are closely related to the data structures that are used within the implementation of the protocol implementation.

Sometimes the protocol specification is written in a form which is either not directly implementable by the translation approach discussed above, or would lead to implementation code which is not sufficiently efficient. In such cases, it is possible to first derive a more implementation-oriented description from the given specification, and then to apply the automatic translation approach to the implementation-oriented description. Much research has been done in the related general area of program transformations [Baue 79], where one wants to assure that the transformed program is equivalent to the originally given program or specification. Research along these lines is going on in relation with the LOTOS language (see Section 4), however, this is a difficult problem and it is not clear how useful this approach will be in the future.

5. Conformance Testing and Implementation Assessment

In the testing area, formal specifications of the protocol can be used for deriving test suites, validating manually developed test cases, and for analyzing test results. The use of TTCN for the description of OSI conformance test cases leads to the need for translation between various languages. The following subsections comment on related issues.

5.1 Test suite development

There are two main approaches for the design of a conformance test suite: manual design and semi-automatic design based on formal description of the protocol. There are several test suites that are manually developed and are being standardized [ISO 8882, and others]. We will concentrate on semi-automatic design from validated formal specifications and describe the techniques and tools available presently. Most of the literature deals with test generation from finite state machines [Sari 82, Sabn 85, Vuon 89, Fuji 90]. Test sequence derivation from FDT specifications has also received attention. First a comprehensive methodology based on Estelle has been developed [Sari 87]. Recently LOTOS based test derivation research has been active [Brin 88, Weze 89, Guil 88, Sari 89e, Trip 89a].

Estelle based test design methodology first applies syntactic transformations on the protocol specification and puts the specification into a form called normal form in which transitions contain single paths and local procedures/ functions are in-line replaced. The normal form transformations are applied to all the modules in case of modular specifications. The next step is to apply control flow and data flow abstraction on the specification. In the control flow abstraction the specification is treated as a collections of interconnected FSMs. This representation is easily obtained from the normal form transitions. The data flow abstraction represents the action part of the transitions as a data flow graph in which abstract service primitive (ASP) and protocol data unit (PDU) parameters become input/ output nodes and the data flow on these are determined from the assignment statements.

The FSM model of the specification makes it possible to use certain techniques to derive test sequences such as transition tours. In the case of modular specifications, transition tour generation considering the internal queues is rather complicated [Forg 90]. The resulting test sequence gives a control flow coverage of all the transitions. State recognition based techniques such as W- and D- methods could also be used; but their application to protocols seems to be controversial due to the fact that the choice of the state recognition sets/sequences requires the consideration of the context variables, rather than just the major state. There are also problems associated to incompletely defined FSMs [Vuon 89]. Since the transition tour coverage does not consider enabling conditions, some of the resulting paths are infeasible. It seems that avoiding these infeasible paths will remain in the domain of human intelligence.

The dataflow graph obtained can be used to derive the dataflow associated with "protocol functions". Since protocol functions are related to the understanding of the protocol, the derivation of them can only be done by the user.

A tool has been developed implementing this methodology. The control and dataflow graphs are visually displayed on a workstation. The test generation module assists the user in eliminating the infeasible paths. The dataflow graphs can be partitioned and named by the user. Then the test generation module derives the test sequences needed to completely cover the dataflow in each function. This way the resulting test sequences completely cover the control and dataflow in the specification. The dataflow coverage criteria is simple. A more involved dataflow coverage considering the definition and use of the context variables is studied [Ural 87a]. Presently it is not yet determined which coverage technique will yield better test sequences. The tool implementing the methodology has been applied to generate test sequences for transport protocol [Forg 89], FTAM [Barb 89], ISDN LAPD [Sari 89d] and ISDN Network layer [Amal 90].

In the case of LOTOS specification, the derivation of test cases is complicated due to the parallel composition and the nondeterministic nature of the specifications. Three different approaches to deriving test cases have been described in the literature.

(1) Canonical test processes: Given a basic LOTOS specification S of a protocol, an "inverse" specification T(S) can algorithmically be developed such that when an IUT is parallelly composed with T(S), it will apply all the test cases to establish the conformance of IUT to S. The method is named CO-OP after its main components, the sets Compulsory and Optional behaviors [Weze 89]. The T(S)obtained is a choice of all the test cases, except robustness tests. It is shown that T(S) is equivalent to the canonical tester defined earlier in [Brin 88]. It is also shown in [Weze 89] that if S is a parallel composition S: = B1 |||| B2 (contrary to intuition) we cannot simply obtain T(B1) and T(B2) and then parallelly compose them to get T(S).

A canonical tester is not intended for practical testing because T(S) can only be derived from basic LOTOS, although some efforts to extend the approach have been reported [Tret 89]. T(S) is another specification in LOTOS usually with an infinite behavior, and therefore it is difficult to derive a finite test suite.

(2) Interpretation of specification: Full LOTOS can be symbolically executed using an interpreter. Symbolic execution derives execution paths that can be used as test sequences [Guil 88]. Although such trees or expansions are usually partial, they can still contain enough information to be used in test case generation. In [Guil 89] this idea was applied in order to generate test cases for LAP-B based on the specification mentioned in Section 2.2.2. From the execution trees obtained from the interpreter, a FSM model was manually derived. Then traditional test selection methods for FSM's could be applied. The authors have not yet considered the dataflow aspects with the interpretation approach.

(3) Analysis of control and dataflow: Applying an extended FSM interpretation to LOTOS semantics, a LOTOS specification can be converted into an equivalent chart [Miln 84]. A translation algorithm of a subset of LOTOS into charts is reported [Karj 88]. This algorithm could be extended to full LOTOS but in some cases, i.e. parallel activation of an unknown number of connections, the chart obtained is infinite. Practice indicates that the chart construction algorithm leads to a state explosion even for medium size specifications such as [Boch 90] due mainly to state machine interpretation of parallel behavior. However using some heuristics, the number of states could be reduced to reasonable size. The chart obtained represents the overall behavior of the entity specified, i.e. the control and dataflow in the specification. The dataflow can be abstracted out and represented in graphical form. These graphs are similar to the graphs obtained from the Estelle specification of the same protocol. The complete traversal of the chart yields the test sequences (may-tests), considering nondeterminism (the "i" events) the tests can be converted into test cases (must-tests) [Trip 89a]. Another approach, also considering the control and dataflow of LOTOS specifications is reported in [Sari 89e].

5.2 Test specification in TTCN

It is important to note that the OSI conformance test cases are described in a newly developed notation called TTCN [OSI C3]. Since specifications written in this language are presented in the form of tables, screen-oriented editors have been developed, although a linear form of the language has also been defined.

In the case of semi-automatic test suite design, a translation from the language of the protocol specification into TTCN is required. If one wants to validate a TTCN test case in respect to a protocol specification, the inverse translation is necessary. Together with ASN.1, which can be used as part of TTCN, there are many translation issues. In addition, there are many proprietary test description languages for which test execution environments exist. Test cases written in TTCN will therefore often be translated into such languages for execution. The use of an intermediate language, such as the charts mentioned above, has been proposed as a common ground for all these translations. However, it is not clear what the best approach to this proliferation of specification languages is.

5.3. Test case validation

The tests obtained with the semi-automatic test design tools are expected to be correct (if the tool has been sufficiently debugged). However, manually developed test cases can be

expected to contain certain errors, due to the large amount of information that must be considered in their design. They can be validated against a formal specification of the protocol, using automated tools.

First, a test case can be validated by checking certain static conditions by a TTCN compiler. Second, the semantics of the test case must be validated against the protocol specification. In particular, each path in the test case leading to a PASS (FAIL) verdict must be shown (not) to be accepted by the formal specification. Additional validation may concern time-out behaviors or buffer sizes.

Two approaches to the automatic validation of test cases have been explored in our projects. After translation into LOTOS behavior expressions, the test case can be automatically compared with a given protocol specification written in LOTOS [Boch 89j]. The tool is an extension of what is described for test result analysis in Section 5.5. Another approach is the translation of both the test case and the specification into charts (see above) and their comparison at this intermediate level [Naik 90].

5.4 Test case adaptation and execution

Most standardized test cases, as well as most tests derived from formal specifications, usually can be classified as generic test cases for the local single-layer test architecture. In order to executed them with a test system using a different test architecture, such as the distributed test architecture [Rayn 87], they must be adapted. However, these necessary modifications are relatively straightforward in most cases.

The test cases must also be adapted to the implementation parameters of the protocol implementation, which are stated in the so-called PICS (protocol implementation conformance statement) and the PIXIT (protocol implementation extra information for testing). Certain values of test parameters are not specified in the generic test cases. They may be chosen during the test execution within the bounds provided by the implementation parameters.

Different approaches can be taken to execute test cases written in TTCN. Most systems for TTCN implementation support translate the test cases into implementation languages, such as C or specialized test description languages for which an execution environment already exists. The translation into Estelle can also be considered [Eswa 90], since good implementation environments for Estelle exist, and this language seems suitable for the specification and implementation of complete test systems [Linn 88]. It is to be noted that the use of ASN.1 in the test cases for Application layer protocols implies that ASN.1 support must be included in the test execution environment.

5.5. Test Result Analysis

Execution of each test case on an IUT will result in a conformance log containing the events that occurred with time stamps. Trace checking and test result analysis refer to the activity in which the conformance log (trace or test results) is analyzed manually, or by using the formal specification of the protocol [Boch 89m, Boch 89j] and a verdict of pass or fail is produced. Since TTCN test cases explicitly indicate the pass and fail cases in the definition, a test system executing TTCN test cases contains the result analysis already in the form of the test cases.

An automatic test result analysis in respect to the protocol specification is useful in the following situations:

(a) In the case that the IUT is subjected to ad hoc or random tests, for instance during debugging, or for complementing the standard conformance tests.

(b) For arbitration testing. This involves two or more systems that have already been tested individually, and which nevertheless turn out to have problems interworking. The testing architecture includes the tester passively observing the PDU's exchanged between the different systems. The tester includes a trace analysis module which checks the observed trace in respect to the specifications of all the systems and will notify any detected error.

(c) For validating the defined test cases. A suite of test cases for a given protocol can be very voluminous. Since most test cases are developed by informal methods, they may contain errors, that is, wrong verdicts. Automatic trace analysis can be used to check the verdicts of test cases with the formal specification of the protocol.

6. Concluding remarks

In our research we have used the three general purpose formal languages called LOTOS, Estelle and SDL, the data structure description notation called ANS.1 and the test specification language called TTCN for protocol design, validation, implementation development and assessment. While our research concentrated on LOTOS and Estelle it seems that presently ASN.1 and TTCN are more heavily used in industry. We discussed several ways in which heavier use of FDTs could be accomplished: developing easy-to-use tools, support for object-oriented techniques, pilot projects and so on.

There is a need to research towards arriving at a stage where protocol development activities could be supported with an integrated set of tools. The need to design validated specifications arouses the need to incorporate the PICS and PIXIT information in the formal specification, this in turn might require modifications in the languages. For example there is a need for an abbreviated notation for LOTOS data structures. It is interesting to investigate incorporation of PICS and PIXIT processing in test generation. Regarding ASN.1, it could possibly conclude that the best way to handle ASN.1 is by treating it as a tree structured data representation sublanguage instead of translating it into other languages. Finally, we point to the need for implementing protocols in parallel environments and by hardware.

Acknowledgements:

We would like to thank all our students and research associates who contributed to the experiences discussed in this paper, and are mentioned in the references. We would also like to thank Pierre Mondain-Monval, Daniel Ouimet and Lucie Lévesque for helping us to put together the final version of this paper. This work was supported by a strategic research grant of the Natural Sciences and Engineering Research Council of Canada.

REFERENCES

[ASN1 C] ISO IS 8825, "Information Processing - Open systems Interconnection - Basic Encoding Rules for Abstract Syntax Notation One (ASN.1)".
[ASN1] ISO IS 8824, "Information Processing - Open systems Interconnection - Specification of Abstract Syntax Notation One (ASN.1)".
[Amal 90] M.Amalou, G.v.Bochmann, publication departementale, 1990.
[Ashk 90] P.Ashkar, "A symbolic evaluator for LOTOS applications", M.Sc. Thesis, University of Ottawa, 1990 (in preparation).

[Barb 89] M.Barbeau, G.v.Bochmann, "Experience with automated verification tools: Application to discrete event systems", Prel. Proc. of Workshop on Automatic Verification Methods for Finite State Systems, Grenoble, June 1989 (due to an error, the paper is not included in the final proceedings published in Springer LNCS).
[Baue 79] F.L.Bauer, et al., "Towards a wide-spectrum language to support program specification and program development", Lecture Notes in C.Sc. 69, Springer Verlag, 1979, pp. 543-552.
[Beli 89] Ferenc Belina, Dieter Hogrefe, "The CCITT-Specification and Description Language SDL", Computer Networks and ISDN Systems, Vol. 16, pp.311-341, 1989.
[Bell 89] O.Bellal, "Analyse automatique de résultats de tests appliquée aux protocoles de communication", Master's thesis, Dept. d'IRO, Univ. de Montreal, 1989.
[Bilg 90] M.Bilgic, B.Sarikaya, "An ASN.1 encoder decoder and its performance", Concordia University, 1990.
[Blac 89] S.Black, "Objects and LOTOS", FORTE'89, Vancouver, 1989.
[Boch 85] G.v.Bochmann, R.Dssouli, W.Lopes de Souza, B.Sarikaya, H.Ural, "Use of Prolog for building protocol design tools", Proc. 5-th IFIP Workshop on Protocol Specification, Verification and Testing, Toulouse, June 1985, North-Holland Publ., pp. 131-147.
[Boch 87c] G.v.Bochmann, "Usage of protocol development tools: the results of a survey" (invited paper), 7-th IFIP Symposium on Protocol Specification, Testing and Verification, Zurich, May 1987, pp.139-161.
[Boch 87e] G.v.Bochmann, D.Ouimet and J.Vaucher, "Simulation for validating performance and correctness of communication protocols", Techn. Report, Department d'IRO, Université de Montréal, 1987.
[Boch 87h] G.v.Bochmann, G.Gerber, and J.M.Serre, "Semiautomatic implementation of communication protocols", IEEE Tr. on SE, Vol. SE-13, No. 9, September 1987, pp. 989-1000 (reprinted in "Automatic Implementation and Conformance Testing of OSI Protocols", IEEE, edited by D.P.Sidhu, 1989).
[Boch 89c] G.v.Bochmann, Q.Gao, C.Wu, "On the distributed implementation of LOTOS", FORTE'89 (IFIP), Vancouver, 1989.
[Boch 89d] G.v.Bochmann, "Protocol specification for OSI", to be published in Computer Networks and ISDN Systems.
[Boch 89e] G.v.Bochmann, et al., "Object-oriented databases: Modelling and specification of applications in the field of network management", Report for research contract CRIM/BNR, April 1989.
[Boch 89h] G.v.Bochmann and M.Deslauriers, "Combining ASN1 support with the LOTOS language", Proc. IFIP Symp. on Protocol Specification, Testing and Verification XI, June 1989, North Holland Publ., pp.
[Boch 89j] G.v.Bochmann and O.Bellal, "Test result analysis in respect to formal specifications", Proc. 2-nd Int. Workshop on Protocol Test Systems, Berlin, Oct. 1989.
[Boch 89m] G.v.Bochmann, R.Dssouli and J.R.Zhao, "Trace analysis for conformance and arbitration testing", IEEE Tr. on Softw. Eng., Nov. 1989, pp.1347-1356.
[Boch 89n] G.v. Bochmann, "Formal methods for describing distributed systems: A discussion of the experience in OSI standardization", Proc. IFIP Working Conf. on Decentralized Systems, Lyon, Dec. 1989 (invited paper).
[Boch 90] G.v.Bochmann, "Specifications of a simplified Transport protocol using different formal description techniques", to be published in Computer Networks and ISDN Systems.
[Boch 90c] G.v.Bochmann, D.Ouimet, G.Neufeld, "Implementation support tools for OSI application layer protocols", to be submitted to Software Practice and Experience.
[Bolo 87] T.Bolognesi and E.Brinksma, "Introduction to the ISO Specification Language LOTOS", Computer Networks and ISDN Systems, vol. 14, no. 1, pp.25-59, 1987.
[Brin 88] E.Brinksma, "A theory for the derivation of tests", Proc. IFIP Symposium on Protocol Specification, Testing and Verification, Atl. City, 1988.

[Budk 87] S.Budkowski and P.Dembinski, "An introduction to Estelle: a specification language for distributed systems", Computer Networks and ISDN Systems, vol. 14, no. 1, pp.3-23, 1987.
[Cusa 89a] E.Cusak, S.Rudkin, C.Smith, "An object-oriented interpretation of LOTOS", FORTE'89, Vancouver, 1989.
[Diaz 89] M.Diaz, et al., "The formal description technique Estelle", North Holland Publ. 1989.
[Ehri 85] H.Ehrig and B.Mahr, Fundamentals of Algebraic Specifications 1, Springer Verlag, 1985.
[Eijk 88] P.H.J. van Eijk, "Software tools for the specification language LOTOS", Doctoral Thesis, Universiteit Twente (1988).
[Eijk 89a] P. van Eijk. "Tools for LOTOS Specification Style Transformation" to appear in S. Vuong, Formal Description Techniques II, North-Holland
[Este 87b] NBS, "User guide for the NBS prototype compiler for Estelle", Final Report, Report no. ICST/SNA - 87/3, Octobre 1987.
[Eswa 90] S.Eswara, T.Berriman, P.vanHoutte, B.Sarikaya, "Towards execution of TTCN", Technical report, Concordia University, 1990.
[FDT GL] ISO Tech. Report, "Guidelines for the Application of Estelle, LOTOS and SDL", DTR 10167, 1989.
[Forg 89] B.Forghani, S.Eswara, V.Koukoulidis, B.Sarikaya, "An Estelle based test generation tool for modular Specifications", FORTE'89, Vancouver, 1989.
[Forg 90] B.Forghani, "Automatic test generation in TTCN from Estelle", M.Sc. Thesis, Concordia University, 1990.
[Fuji 90] S.Fujiwara, G.v.Bochmann, F.Khendek, M.Amalou, A.Ghedamsi, "Test selection based on finite state models: A unified view", submitted to IFIP Symposium on Protocol Specification, Testing and Verification, 1990.
[Gall 89] M.S.Gallouzi, "Trace analysis of LOTOS behaviors", M.Sc. Thesis, University of Ottawa, 1989.
[Graf 89] S.Graf, J.-L.Richier, C.Rodriguez, J.Voiron, "What are the limits of model checking methods for the verification of real life protocols?", in Lecture Notes in Computer Science "Automatic Verification Methods for Finite State Systems International Workshop", G.Goos and J.Hartmanis (Eds.), Grenoble, France, Springer-Verlag, June 1989.
[Guer 89] D.Gueraichi, L.Logrippo, "Derivation of test cases for LAP-B from a LOTOS specification", to appear in the Proc. of the 2nd FORmal TEchniques Symposium (Vancouver, December 1989).
[Guer 89a] D.Gueraichi, "Derivation of test cases for LOTOS LAP-B specification", M.Sc. Thesis, University of Ottawa, 1989.
[Guil 88] R.Guillemot, M.Hay-Hussein, L.Logrippo, "Executing large LOTOS specifications", Proc. IFIP Symposium on Protocol Specification, Testing and Verification, Atl. City, 1988.
[Guil 89] R.Guillemot, L.Logrippo, "Derivation of useful execution trees from LOTOS by using an interpreter", in "Formal Description Techniques", (Ed.) K.J.Turner (Proceedings of the First International Conference on Formal Description Techniques Stirling, Scotland, 6-9 September, 1988), pp.311-325.
[Hoar 85] C.A.R.Hoare, "Communicating Sequential Processes", Prentice Hall, 1985.
[ISO 8882] ISO, "X.25-DTE Data link layer conformance test suite", TC97/SC6, ,DIS8802 Part 2, 1989.
[Jard 85b] C.Jard, R.Groz, J.F.Monin, "VEDA: a software simulator for the validation of protocol specifications", Proc. COMNET '85 (IFIP), Computer Network Usage: Recent Experiences, North Holland, Oct. 1985.
[Jard 89] .Jard, J.M.Jazequel, "A multi-processor Estelle to C compiler to experiment distributed algorithms on parallel machines", Proc. IFIP Symposium on Protocol Specification, Testing and Verification IX, Twente, 1989.

[Karj 88] G.Karjoth, "Implementing process algebra specifications by state machines", Proc. IFIP Symposium on Protocol Specification, Testing and Verification, Atl. City, 1988.
[Larm 88] K.G.Knightson, T.Knowles and J.Larmouth, "Standards for Open Systems Interconnection", McGraw-Hill, 1988.
[Leon 89] G.Leon, C.Delgado Kloss , G.Gonzalez, M.A.Ruiz, S.Marchena, L.Santos, J.Navarro, "ASDE: Design of a transformational environment for LOTOS", to appear in S.Vuong, Formal Description Techniques II, North-Holland.
[Linn 88] R.J.Linn and J.P.Favreau, "Application of Formal Description Techniques to the Specification of Distributed Test Systems", Proc. of INFOCOM'88, pp.96-109, IEEE, March 1988.
[Logr 88] L.Logrippo, et al., "An interpreter for LOTOS: A specification language for distributed systems", Software Practice and Experience, Vol. 18(4), pp.365-385, April 1988.
[Logr 89] L.Logrippo, T.Melanchuk, R.J.DuWors, "The algebraic specification language LOTOS: an industrial experience", University of Ottawa, Computer Science Department, report TR-89-35.
[Logr 90] M.Faci, L.Logrippo, B.Stepien, "Formal specification of telephone systems in LOTOS: the constraint-oriented style approach", will be published in Computer Networks and ISDN Systems, preliminary version will appear in "Protocol Specification, Testing and Verification IX", North-Holland, 1989.
[Mana 89] J.A.Manas, T.DeMiguel and H.vanThienen. "The Implementation of a Specificatio Language for OSI Systems" In: P.vanEijk et al (Eds) The Formal Description Technique LOTOS. North-Holland 1989.
[Miln 80] R.Milner, "A calculus of communicating systems", Lecture Notes in Computer Science, No. 92, Springer Verlag, 1980.
[Miln 84] R.Milner, "A complete inference system for a class of regular behaviors", JCSS, 1984, pp.439-466.
[Miln 89] R.Milner, "Communication and Concurrency", Prentice-Hall, 1989.
[Mond 90] P.Mondain-Monval, "Object-oriented Model for the OSI Reference Model", submitted to IFIP Symposium on Protocol Specification, Testing and Verification, 1990.
[Naik 90] S.Naik, B.Sarikaya, "Static Validation of TTCN Test Cases" Technical report, Concordia University, 1990.
[Najm 89] E.Najm, "A verification oriented specification in LOTOS of the Transport Protocol", in "The Formal Description Technique LOTOS", P.H.J.van Eijk, C.A.Vissers, and M.Diaz (Eds.), pp.269-284.
[Neuf 90] G.W.Neufeld, "A Tutorial on ASN.1", to be published in Computer Networks and ISDN Systems, 1990.
[OSI 83] "Special issue on Open Systems Interworking", Proc. of the IEEE, Dec. 1983.
[OSI C3] ISO "DP 9646-3, Information Processing Systems - Open Systems Interconnection - Conformance Testing Methodology and Framework - Part 3: The Tree and Tabular Combined Notation (TTCN)".
[OSI MO] ISO/IEC JTC1/SC6 N5402 "Guidelines for the definition of managed objects" (Draft 1989).
[OSI RO] ISO IS 9072, "Remote Operations".
[OSI TS] ISO TC97/SC16, DP8072, "OSI - Transport service specification", 1983.
[Quem 89] J.Quemada, S.Pavon and A. Fernandez, "Transforming LOTOS specifications with LOLA. The Parameterised Expansion", in "Formal Description Techniques, K.J.Turner (Eds.), North-Holland, 1989, pp.45-54.
[Rayn 87] D.Rayner, "OSI Conformance testing", Computer Network and ISDN Systems, 14 (1987), pp. 79-98.
[Saba 90] Fayez Saba, "Validation en ligne de traces d'execution appliquee au protocole de Transport", M.Sc. thesis, Université de Montréal, summer 1990.

[Sabn 85] K.Sabnani, A.Dahbura, "A new technique for generating protocol tests", Proc. 9th Data Comm. Symp., 1985, pp.36-43.
[Sari 82] B. Sarikaya and G.v. Bochmann, "Some experience with test sequence generation for protocols", Proc. 2-nd Int. Workshop on Protocol Specification, Testing and Verification, North Holland, 1982, pp. 555-567.
[Sari 87] B.Sarikaya, G.v.Bochmann, E.Cerny, "A test design methodology for protocol testing" IEEE Trans. on SE, (May 1987), pp. 518-531.
[Sari 88f] B.Sarikaya, and Q.Gao, "Translation of test specifications in TTCN to LOTOS", Proc. IFIP Symposium on Protocol Specification, Testing and Verification, Atl. City, 1988.
[Sari 89d] B.Forghani and B.Sarikaya, "Automatic dynamic behaviour generation in TTCN format from Estelle specifications", Proceedings of the international workshop on Protocol Test Systems, Berlin (West), Germany, October 1989, pp.73-90.
[Sari 89e] B.Sarikaya and S.Lu, "A LOTOS based test design methodology", Technical report, Concordia University, 1989.
[Scol 87] ISO 97/21 N1540, "Potential enhancements to LOTOS", 1986.
[Shir 89] N.Shiratori, H.Kaminaga, K.Takahashi and S.Noguchi, "A verification method for LOTOS specifications and its applications", to appear in IFIP Protocol Specification, Testing and Verification IX, C.A.Vissers (Eds.), North-Holland, 1989.
[Sidh 88a] D.P. Sidhu and C.S. Crall, "Executable Logic Specifications for Protocol Service Interfaces", IEEE Transactions on Soft. Eng., Vol. 14, No.1, January 1988.
[Sidh 89] D.P. Sidhu and T.P. Blumer, "Automatic Implementation of Protocols", to be published in Computer Networks and ISDN Systems.
[Somm 89] Ian Sommerville, "Software Engineering", 3-rd ed., Addison-Wesley Publ. 1989.
[Tret 89] J.Tretmans, "Test case derivation from LOTOS specifications", FORTE'89, Vancouver, 1989.
[Trip 89a] P.Tripathy and B.Sarikaya, "Test generation from protocol specification", FORTE'89, Vancouver, 1989.
[Ural 87a] H.Ural, "Test sequence selection based on static dataflow analysis", Computer Communications, Vol.10, No.5, October 1987.
[Viss 88] C.Vissers, G.Scollo, M.v.Sinderen, "Architecture and Specification Style in Formal Descriptions of Distributed Systems", Proc. IFIP Symposium on Prot. Spec., Verif. and Testing, Atlantic City, 1988.
[Viss 89] C.A.Vissers, G.Scollo, M.v.Sinderen, E.Brinksma, "On the use of specification styles in the design of distributed systems", University of Twente, 1989.
[Vuon 89] S.T.Vuong, W.L.Chan, M.R. Ito, "The UIOv-method for protocol test sequence generation", proceedings 2-nd Int. Workshop on Protocol Test Systems, Berlin, Oct. 1989.
[Weze 89] C.D. Wezeman, "The CO-OP method for compositional derivation of conformance testers", Proc. IFIP Symposium on Protocol Specification, Testing and Verification IX, Twente, 1989.
[Yang 88] Y.Yang, "ASN.1-C Compiler for Automatic Protocol Implementation", Master Degree Thesis, Department of Computer Science, University of British Columbia, October 1988, 111pp.
[Zafi 80] P.Zafiropulo, C.H.West, H.Rudin, D.D.Cowan, and D.Brand, "Towards analyzing and synthesizing protocols", IEEE Tr. Comm. COM-28, 4 (April 1980), pp. 651-660.

Protocol Validation Using Process Algebraic Recursive Equations

Jan de Meer and Ramesh Ahooja

GMD-FOKUS
Hardenbergplatz 2
1000 Berlin 12
W. Germany
e-mail: jdm,rah@fokus.berlin.gmd.dbp.de

Abstract

Validation of algebraically specified protocols is considered. Consistency of protocol-underlyingservice behaviour and hence that of the provided-required service validates the protocol. Requirements on protocol behaviour are captured as *partial constraints*[0] satisfied by guarded recursive equations of algebraic processes. FoHowing the work of Bergstra, Klop, Hoare and Milner specifications in the form of guarded recursion equations which are free of abstraction are found to be suitable for validation of protocols. Consistency is proved by parallel composition of the relevant behaviours. A simplified version of the OSI Transport protocol (class 0) is validated as an illustrative example. The method can be adapted for specifications in the FDT LOTOS.

I Introduction

Validation of protocol behaviour specifications implies a proof of consistent behaviour. In the multi-layered OSI (Open Interconnected Systems) [1] framework, a protocol must be dually consistent. The protocol must interact consistently with the underlying service to provide the intended service to users in the layer above. Consistent interaction between the users (required service) and the intended service (provided service) validates the protocol.

Validation concerns have been extensively studied in the literature, see for example [2],[3]. The principles of structured program design [4],[5] coupled with concurrent programming language specifications [6],[7],[8] have provided the means to relate distributed system architecture to validation issues. Methodologies for validation have included finite state machine models for system analysis [9] and inductive assertion methods [10],[4],[6],[11]. Temporal Logic [12] and the method of invariants has been used to prove system properties [13], [11].

[0]The notion of partial constraints on distributed system behaviour has been introduced in [25]. Partial constraints form the basis of the Partial Constraint Methodology for the validation and verification of distributed systems.

The notion of Observational Equivalence [17],[18] of algebraic processes has been used to verify the intended service provided by a protocol/underlyingservice combination in [22] and [23]. Observational Equivalence between an intensional specification (protocol/underlyingservice specification) and an extensional specification (intended service specification) is demonstrated by means of a verification tool in [22]. The tool converts algebraic behaviours to Finite State Transition (FST) systems which reduces the decidability of observational equivalence complexity to polynomial time. However, the large number of states of real protocols make complexity concerns for deciding observational equivalence unavoidable.

The partial constraint methodology of design, validation and verification of distributed systems [25] form the basis of the validation method described in this paper. Requirements on protocol behaviour are captured as partial constraints describing simple relationships between the events of a communication environment. Partial constraints are then mapped onto a process algebra such as CCS [17], selected for behaviour specification. The mapping results in guarded recursion equations of processes which satisfy partial constraints. The specifications so obtained are partial specifications, in that the complete set of requirements are not specified. Validation of the intended service is proved by the deadlock-free parallel composition of all the specified processes in the communication environment.

The paper is organised as follows: The protocol validation problem is briefly stated in section II. Process algebraic specifications obtained from partial constraints are described in section III. Section IV briefly describes the validation method. Section V illustrates the method by means of an example consisting of a simplified Transport protocol. Section VI provides a brief comparison with other validation methods. Section VII concludes the paper.

II Protocol Validation

OSI protocol and service architectures

In order to motivate the validation method we briefly recall the relevant portion of the OSI Basic Reference Model (BRM).

The OSI BRM's multi-layered architecture describes two possible representations of the (n)-service (see for example [14], [15]):

(i) one which represents the behaviour resulting from the joint operations of the (n)-entities (layer protocol) and the (n-1)-service (underlying service).

(ii) and one which abstracts from the internal structure of services 1 to n.

Validation

The objective of validation is to determine whether the specified behaviour of the communication service or protocol is really what the user had in mind. It implies a demonstration, for example with the help of a simulator that the behaviour as defined by the specification or implementation satisfies user requirements. Thus validation implies proof of correct intention. It has the following aspects:

1. Consistent interactions between the protocol entities and the underlying service.
2. Consistent interactions between the protocol entities and the user
3. Validation of (n)-service provided by the protocol using its underlying service against the abstract (n)-service i.e. the intended service.

In the sequel, it is shown that if protocol behaviour is specified consistently in terms of the interaction behaviour of items 1 and 2, then a composition of the behaviour equations for the protocol and the underlying service represent service behaviour. The service behaviour so obtained when evaluated for correct intention provides validation of the protocol as in item 3 above.

Following [22], we take the point of view that systems designed with validation concerns incorporated into their specifications are more easily validated than others.

III Partial constraints and algebraic specifications

In this section *partial constraints* [25] are introduced and a mapping from partial constraints onto basic process algebraic behaviour expressions is defined. The use of partial constraints for system design results in guarded recursion equations of a selected process algebra. Any of the process algebras CCS[17], CSP[8], ACP[19] or LOTOS [16] can be selected for the mapping. The interested reader will find detailed information about the process algebras in the references.

In what follows we make the following assumptions:

1. Communication is an action (or a communication event), resulting from synchronisation or an interaction between actions. Actions of an environment are atomic.

2. Given an environment E of communication events we make the assumption that:

$\exists$ a modality of determination (a systematic process of determination is modally 'possible') by means of which behaviours and related assertions on behaviours of communication systems can be determined.

If β is a set of behaviours and Pc the set of *partial constraints* (defined below) on behaviours in E, then the assumption implies that:

(i) $\exists$ a method of determining partial constraints pc and behaviours B in E

(ii) $\exists$ a definition of provablility, s.t. $\forall$ B $\varepsilon \beta \exists$ a pc ε Pc, s.t. B $\models$ pc.

Definition III.1 Partial Constraints

The set Pc contains well formed formulae (wff) on the actions of an environment E and the partial constraint operators 'or', 'precedes', 'and', 'not', '$\forall$' and '$\exists$'. They are called partial constraints and denoted by pc. For example:

$\forall \alpha,\beta \varepsilon$ E

(i) ' ' (the empty pc) is a wff

(ii) 'α', 'α or α','α or α or α...' are wff

(iii) 'α or β' is a wff

(iv) 'α precedes α' is a wff

(v) 'α precedes β' is a wff

(vi) 'α and α' is a wff

(vii)'α and β' is a wff .....etc.

The operator 'or' is interpreted to be a non-deterministic choice such that: the pc 'α or β' true implies that it is possible to make a determination in E in which *either* α *or* β can occur. Similarly, 'α precedes β' implies '$\alpha < \beta$' or α occurs before β.

Complete definitions, asssociated theorems and examples of the partial constraint methodology are given in [25].

Behaviour descriptions can be modelled by action trees. Action trees do not have node information, but their edges are labelled by action names. Alternative behavior is modelled by branching the edges of the action tree, and sequential behavior is modelled by connecting the edges sequentially. The example in Figure III.1 shows sequential and alternative behaviour involving actions α and β:

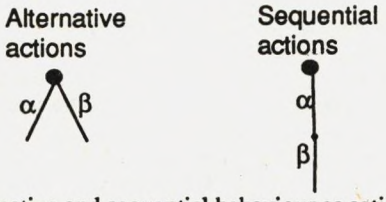


Figure III.1. Alternative and sequential behaviour as action trees

Mapping partial constraints to algebraic behaviour expressions.

The mapping from partial constraints to algebraic behaviour expressions can be defined by the following:

Definition III.2 Partial constraint mapping
In a communication environment E, the actions in behaviour expressions and partial constraints are self-referential.

By letting Π be a set of behaviour specifications in a given behaviour model (process algebra) the smallest set Pc of partial constraints in E are interpreted by defining the satisfaction relation $\models$ in Π x Pc as follows by induction on the structure of Pc:
(i) The behaviour expression B0 (inaction) $\models$ ' ' (the empty pc)
(ii) B $\models$ pc ==> B satisfies pc (always)
(iii) B $\models \neg$pc iff not B $\models$ pc
(iv) B $\models \cap$ pc(i) iff B $\models$ pc(i) $\forall$ i $\in$ I, the set of integers
(v) If B1 $\models$ pc1 and B2 $\models$ pc2 iff B = B1 'op' B2 $\models$ pc1 'o' pc2

Where the partial constraint operators o are related to the operators: 'op' of the process algebra under consideration as follows:
If 'op' = - non-deterministic or- then, 'op' $\models$ 'or';
If 'op' = - sequentiality-, 'op' $\models$ 'precedence';
If 'op' = - parallel composition-, 'op' $\models$ 'and';
If 'op' = - inaction-, 'op' $\models$ ' ';

The operators '$\forall$', '$\exists$' and 'not' are indirectly related to the operators of the procees algebra . Other operators defined in a process algebra are not considered.

System Architecture and Specification

The essential architecture of a distributed system 'S' consists of a protocol process 'P' distributed into component processes 'PE1' and 'PE2' (say), and a non-distributed process 'med' (say) providing the underlying communication service. The process 'med' may itself consist of decomposed processes 'med1' and 'med2'. The system 'S' operates in a communication environment E in which user processes called 'usr' (say) use the services of the distributed system.

Partial constraints based on requirements are constructed for the protocol processes 'P', 'PE1' and 'PE2', as well as for the processes 'med' and 'usr'. The partial constraints so obtained are mapped onto the behaviour expressions of a selected process algebra. The Recursive Definition Principle (RDP) [19] states that guarded recursive equations of algebraic processes have unique solutions. No hiding of actions is recommended in CSP and ACP. Recursion in the behaviour expressions ensures the satisfaction of associated partial constraints. The specifications are partial, in that the complete set of requirements is not specified.

The method makes it possible to define constraints which are of the "end-to-end" and "local" type. Constraints so obtained, unambiguously yield specifications which are easily validated.

IV The Validation Method

The service "S" provided by the protocol is obtained by use of the formula:

$S = P \parallel med$ ....eqn_IV.1

Where the parallel composition operator (||) is defined as in CCS and LOTOS, and full synchronisation is assumed on the gatelists of P and 'med'.

If the composition yields (by simulation, for example) deadlock-free traces, the protocol P behaves consistently with the underlying service 'med' to provide the service "S" to the users.

In order to show that the service "S" provided by the protocol is what the user had in mind, i.e. it is of correct intention, we can construct partial constraints from the traces obtained from eqn_IV.1 and compare them to partial constraints of intended service. Consistency of partial constraints then validates the protocol.

Another way of validation is to make use of the formula:

$S+ = usr \parallel P \parallel med$ ....eqn_IV.2

where the process 'usr' represents partial specifications of user behaviour with respect to the required service of concern in the validation problem.

V An example - Validation of the Transport protocol

Consider the Connection establishment phase of the OSI transport layer protocol (class 0). (The Transport protocol is considered in more detail in [26], [21] and [25]). To keep the presentation of ideas simple and comprehensive, a subset of all the required behaviours is considered. Disconnect requests or indications are not considered, but they can be easily treated by the method.

Figure V.1 shows the architecture of the system. The distributed system environment consists of user processes, a protocol and an underlying service. The logical structure of the protocol consists of two protocol entitites PE1 and PE2 respectively. The combination of PE1 and PE2 is the protocol process, with two pairs of interaction points t1 and t2, and p1 and p2 respectively. The interaction points t1,t2 are shared with user processes of the environment - usr1 and usr2 respectively. While p1 and p2 are shared with the underlying service.

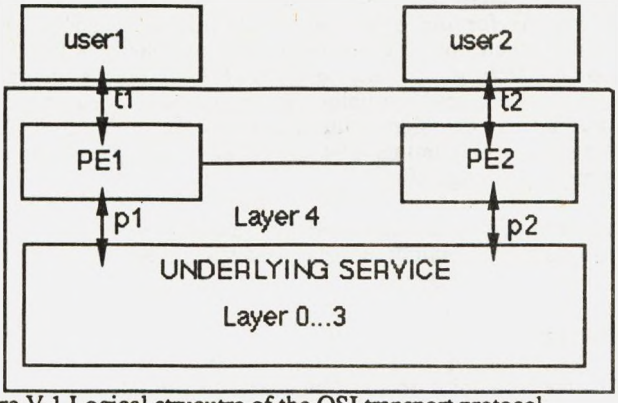


Figure V.1 Logical strucutre of the OSI transport protocol.

In order to model data type value passing on interaction of processes, it is assumed that the actions of an environment E can be indexed by values of predefined data type sorts, and that the set of values of all data types are countable. For example:

The partial constraint

"p1(send/output)CR TPDU precedes p2(receive/input) CR TPDU"

....pc_V.1

constraints a generic action p1 (the action occurs at the interaction point p1 of the process) indexed by a value: '(send/output)CR TPDU' to precede a generic action p2 (the action occurs at the interaction point p2 of the process) indexed by any value of the CR TPDU sort. The index values come from countable sorts (for example, the CR TPDU sort), making the actions unique. An index value represents a typical, well defined value of the associated predefined sort. Note that the indexes are not parameters, but interpreting them as such makes the specification readable. In principle system behaviour can be specified by a unique labelling of all the actions. But this would make the specification unreadable.

By convention we assign the symbol ! to the '(send/output)' component of the index and ? to the '(receive/input)' component of the index.

A second convention is the same as the matching convention followed in LOTOS for synchronisation of actions. In LOTOS, actions with common generic labels can synchronise if action parameters have the form:

1. ?type_name and ?type_name (i.e. same type_name)
2. ?type_name and !value_id (i.e. value_id is of sort type_name)
3. !value_expression and !value_expression
(i.e. the same value_expression)

In the specification given below the LOTOS type of synchronisation convention is assumed for the indexed actions.

Requirement descriptions are obtained from the standard for the OSI connection oriented transport protocol - [IS2]. It is assumed that simple canonical transformations of the behaviour mechanisms described in the standard yield partial constraints (as in pc_V.1). For example, the procedure describing Connection establishment (cf.clause 6.5.4 of [IS2]) contains the following statement:

"A transport connection is established by means of one transport entity (the initiator) transmitting a CR TPDU to the other transport entity (the responder), which replies with a CC TPDU."

Where CR TPDU and CC TPDU are a connect request and connect confirm transport protocol data units respectively.

By assigning roles of intitiator/responder to both the gates p1 and p2 we deduce the following partial constraint:

"p1(send/output) CR TPDU precedes p2(receive/input) CR TPDU
precedes p2(send/output) CC TPDU
precedes p1(receive/input) CR TPDU"
or
"p1(send/output) CR TPDU precedes p1(receive/input) CC TPDU

In a similar manner partial constraints for PE1 and PE2 are obtained and mapped onto process algebraic behaviour expressions. A LOTOS-like notation for the specification of processes satisfying the partial constraints is used to obtain:

```
protocol[t1,p1,t2,p2] = PE1[t1,p1,t2,p2] [] PE1[t1,p1,t2,p2]
                                                        ....eqn_V.1
where
PE1[t1,p1,t2,p2]  =
      t1?TConreq ; p1!NDatareq!Conreq ; protocol[t1,p1,t2,p2]
      []
      t1?TConresp ; p1!NDatareq!Conconf ; protocol[t1,p1,t2,p2]
      []
      p1?NDataind?Conreq ; t1!TConind ; protocol[t1,p1,t2,p2]
      []
      p1?NDataind?Conconf ; t1!TConconf ; protocol[t1,p1,t2,p2]

PE2[t1,p1,t2,p2]  =
      t2?TConreq ; p2!NDatareq!Conreq ; protocol[t1,p1,t2,p2]
      []
      t2?TConresp ; p2!NDatareq!Conconf ; protocol[t1,p1,t2,p2]
      []
      p2?NDataind?Conreq ; t2!TConind ; protocol[t1,p1,t2,p2]
      []
      p2?NDataind?Conconf ; t2!TConconf ; protocol[t1,p1,t2,p2]
```

The partial behaviour of the underlying service processes obtained from its partial constraints is given by:

```
unds[p1,p2] =
          p1?NDatareq?Conreq ; p2!Ndataind!Conreq ; unds[p1,p2]
          []
          p2?NDatareq?Conreq ; p1!Ndataind!Conreq ; unds[p1,p2]
          []
          p1?NDatareq?Conconf ; p2!Ndataind!Conconf ; unds[p1,p2]
          []
          p2?NDatareq?Conconf ; p1!Ndataind!Conconf ; unds[p1,p2]
                                                        ....eqn_V.2
```

The partial behaviour of the user processes must satisfy the required service.

The user processes are given by:

```
usr[t1,t2] = user[t1,t2]  [] user2[t1,t2]
                                                    ....eqn_V.3
where
user1[t1,t2] =   t1!TConreq ; t1?TConconf ; usr[t1,t2]
                 []
                 t1?TConind ; t1?TConresp ; usr[t1,t2]
user2[t1,t2] =   t2!TConreq ; t2?TConconf ; usr[t1,t2]
                 []
                 t2?TConind ; t2?TConresp ; usr[t1,t2]
```

For this specification, parallel composition is the same as in LOTOS, so that stepwise synchronisation of matching events of two processes occurs, at common interaction points. For example, two actions at the interaction point 'p1' can synchronise with exchange of messages if the two processes offer: '!NDatareq!Conreq' and '?NDatareq?Conreq' respectively. Where '?Conreq' for example is an index representing a predefined sort of a pdu (protocol data unit) type and '!Conreq' an actual value in a manner similar to that of ACTONE data types.

Validation Formula

Parallel composition of all the processes results in the expression:

```
S = usr[t1,t2] |[t1,t2]| protocol[t1,p1,t2,p2] |[p1,p2]| unds[p1,p2]
                                                    ....eqn_V.4
```

Common actions with matching events of the two processes result in interactions. Interactions are possible at each recursion of the processes. Those actions not common to any of the processes are not synchronised, but interleaved. The composition results in traces which yield partial constraints which are consistent with the traces. Behaviour expressions satisfying the partial constraints represent possible behaviours of the system. The set of traces observable for the behaviour specified in eqn_V.4 are:

```
T1!TConreq , p1!Ndatareq!Conreq , p2!NDataind!Conreq , t2!TConind,
t2!TConresp , p2!NDatareq!Conconf  , p1!NDataind!Conconf , t1!TConconf
or
t2!TConreq , p2!NDatareq!Conreq , p1!NDataind!Conreq , t1!TConind ,
t1!TConresp , p1!NDatareq!Conconf , p2!NDatareq!Conconf , t2!TConresp
                                                    ....tr_V.1
```

The set of traces tr_V.1 represent possible behaviours of the connection service.

The trace tr_V.1 forms the basis of determining the constraint:

```
" 't1!TConreq precedes  p1!Ndatareq!Conreq precedes p2!NDataind!Conreq
   precedes t2!TConind precedes t2!TConresp precedes p2!NDatareq!Conconf
   precedes p1!NDataind!Conconf precedes t1!TConconf'
or
  't2!TConreq precedes p2!NDatareq!Conreq precedes p1!NDataind!Conreq
  precedes t1!TConind precedes t1!TConresp precedes p1!NDatareq!Conconf
  precedes p2!NDatareq!Conconf precedes t2!TConresp' "
                                                    ....pc_V.2
```

The constraint pc_V.2 can be compared to the time sequence diagrams of the OSI Transport service definition standard (cf. pps. 9, Figure 4 [IS3]).

For example, the time sequence diagrams of the transport service contain the sequence of transport service primitives given in Figure V.2:

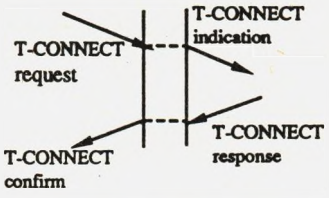


Figure V.2 Successful TC establishment

The following partial constraint based on the time sequence diagram of Figure V.2 describes successful TC establishment when TS user1 (interacting with TS provider at interaction point t1) or TS user2 (interacting with TS provider at interaction point t2) is calling/called TS user:

" (t1/t2)!TCONreq precedes (t1/t2)!TCONconf " ....pc_V.3

It is easy to deduce pc_V.2 from tr_V.2, and also that pc_V.3 is consistent with pc_V.2. Therefore, the protocol as specified in eqn_V.1 is valid with respect to intended service provided to the user as required in eqn_V.3.

VI Comparison with other methods

The method described in this paper is based on the partial constraint methodology elaborated in [25]. The methodology introduces and formally defines the notion of *partial constraints*. Partial constraints provide a simple but powerful tool to associate modal assertions with algebraic behaviour expressions. Thus providing the basis for an assertional approach to validation and verification. The method avoids the complexity of observational equivalence analysis for verification.

The method is similar to the one proposed in [22] in that the design strategy used evolves from the constraint-oriented style of specification [24]. However, it differs with the verification-oriented style of [22] from two aspects. Firstly, the specifications relate distributed guarded recursive processes by means of non-deterministic choice rather than by interleaving. This results in the natural elimination of a large number of undesirable behaviours which considerably slow down simulation experiments. Secondly, reducing algebraic specifications to FSTs to show observational equivalence makes the complexity of proof unavoidable.

Our method is similar to the one method proposed in [23] in that the validation formulae developed are identical. However, the structure of the component processes participating in the parallel composition of the formulae makes validation by simnulation tractable. Also, [23] basis validation on observational equivalence analysis of complete specifications. So that complexity concerns similar to those for [22] apply.

VII Conclusions

We have described a specification approach for validation of distributed systems in general and OSI-protocols in particular. With respect to the required service it can be shown that the protocol has been implemented correctly or with respect to the protocol the intended service is shown to be valid.

We have used fundamental results obtained from the process algebras CCS, ACP and CSP that, guarded recursion equations of algebraic processes specified with no concealment of interactions have unique solutions. These results form the basis of developing a validation methodology which reduces considerably simualtion times required to prove consistency of behaviours. Additionally, the method makes the validation problem tractable by avoiding the complexities of proving observational equivalence of processes via FSTs.

The results of this approach have already been applied in [20], [21] to derive test sequences from a process algebraic specification of protocols.

Bibliography

[1] 'IS7498': Information processing systems-Open systems interconnections-Basic Reference Model(1984).

[2] R. T. Yeh, Editor, "Program Validation", in Current Trends in Programming Methodology, Vol. II, Prentice-Hall, Inc. Englewood Cliffs, New Jersey 07632.

[3] M. Paul, H.J. Siegert, Editors, "Distributed Systems", in Lecture Notes in Computer Science, Edited by G. Goos and J. Hartmanis, Springer-Verlag, Berlin, Germany 1985.

[4] E. W. Dyjkstra, "Guarded Commands, Non-Determinacy and Formal Derivation of Programs", in CACM 18, 19775, pps. 453-457.

[5] D. Parnas, "Designing Software for Ease of Extension and Contraction", in Proceedings of the third International Conference on Software Engineering, IEEE-ACM, May 1978, pps. 264-277.

[6] L. Lamport, "The Hoare Logic' of Concurrent Programs", Acta Informatica 14,1980.

[7] L. Lamport, "Specifying Concurrent Program Modules", ACM Transactions on Programming Languages and Systems 6, 2 (April 1983).

[8] C. A. R. Hoare, "Communicating Sequential Processes", Prentice-Hall International, Englewood Cliffs, New Jersey, U.S.A., 1985.

[9] C. A. Sunshine et al, " Specification and Verification of Communication protocols in AFFIRM using state transition models",in IEEE Transactions on Software Engineering, Vol. SE-8, No. 5, September 1982.

[10]C. A. R. Hoare, "An Axiomatic basis for Computer Programming", Comm. of the ACM 12,(10), Oct. 1969, pps. 576-580,583.

[11]L. Lamport, "An Assertional Correctness Proof of a Distributed Algorithm", Science of Computer Programming 2,3 (Dec. 1982), pps. 175-206.

[12] A. Pneuli, "The Temporal Logic of Programs", Proceedings of the 18th Symposium on the Foundation of Computer Science, ACM, Providence, Rhode Island, Oct. 1977, pps. 46-57.

[13] L. Lamport, ""Sometime" is Sometimes Not "Never"", in Proceedings of the seventh Symposium on Principles of Programming Languages, ACM SIGPLAN, Las Vegas, Nevada, Jan 1980.

[14] "IS8072": Information processing systems-Open Systems Interconnection-Transport service definition.

[15] "IS8073": Information processing systems-Open-Systems Interconnection-Connection oriented transport protocol specification.

[16] "IS8807": "LOTOS - A Formal Description Technique Based on the Temporal Ordering of Observational Behaviour", International Standard IS 8807, September 1988.

[17] R. Milner, " A calculus of Communicating systems " Lecture Notes in Computer Science, No. 92, eds. G. Goos and J. Hartmanis, Springer-Verlag, Berlin, Heidelberg, 1980.

[18]R. Milner, "The Calculus CCS And Its Evaluation Rules", in Control Flow and Data Flow: Concepts of Distributed Programming, M. Broy (ed) NATO ASI, Series Vol. F14, Springer-Verlag, Berlin, 1984.

[19] J.A.Bergstra, J.W.Klop, "Process Algebra: Specification and Verification in Bisimulation Semantics", in Reprint from CWI Monographs 4 North Holland 1986, pp61-94.

[20] Jan de Meer, "Derivation and Validation of Test Scenarios based on the Formal Specification Language LOTOS", 6th IFIP IWS on PTSV, Montreal, 1986

[21] R. Ahooja, J. Burmeister, J. de Meer and Axel Rennoch, " Method of Open Test Sequences and their Implementation generalised in a Conformance Test Tool", to appear in the proceedings of the second IWPTS conference to be held in Berlin, W. Germany, in October 1989.

[22] E. Najm, "A verification oriented specification in Lotos of the Transport Protocol", Seventh IFIP international meeting on Protocol Specification, Testing and Verification, Zurich,May 5-8,1987 (North-Holland 1988).

[23] J. Parrow, "Verifying a CSMA/CD-protocol with CCS", Eighth IFIP international meeting on Protocol Specification, Testing and Verification, Atlantic City, New Jersey, USA, June 7-10,1988 (North-Holland 1989).

[24] C. A. Vissers, Giuseppe Scollo, Marten Van Sinderen, "Architecture and Specification Style in Formal Descriptions of Distributed Systems", Invited paper, Eighth IFIP international meeting on Protocol Specification, Testing and Verification, Atlantic City, June 7-10,1988.

[25] R. Ahooja, "On the Design of Distributed Systems", Ph.D. thesis submitted t o Concordia University, Montreal - Canada, December 1989.

[26] R. Ahooja, "Towards Verification of Communication Protocol Specifications in LOTOS, Canadaina Conference on Electrical and Computer Engineering, Vancouver, Canada, 1989.

A SYNTHETIC LOCALIZED APPROACH TO PROTOCOL VALIDATION

Kassem Saleh and Robert Probert

Protocols Research Group
Department of Computer Science
University of Ottawa
Ottawa, Ontario, Canada.
K1N 6N5

In this paper, we present a simple and efficient protocol validation technique for the detection and correction of logical design errors. The communication model consists of two entities asynchronously exchanging messages via lower modules modelled by reliable FIFO channels. The communicating entities are represented by acceptance trees, traces and sets (ATTS), a tree-based model which is general enough to express any labelled transition-based system. Very limited coupling between the two entities during the validation process is required, therefore this technique is denoted a localized approach to validation. The validation of the entities can be executed in parallel and the computations that are performed during the validation are very simple. Moreover, this technique can be integrated into an incremental synthetic protocol design methodology which guarantees freedom from logical errors and satisfaction of service requirements. Therefore, this technique can be used for the validation of existing designs, and for the synthesis of new designs.

1. INTRODUCTION

Protocol design consists essentially of the construction of interacting entities which cooperate to provide a set of specified services to the service users while guaranteeing that no errors will be encountered. We can recognize two types of protocol design errors:

(1) errors encountered during the progress of the interactions between the communicating entities. These *progress errors* can exist in any protocol and are independent of service requirements. These errors are often called *logical errors*, and include such errors as deadlock, unspecified reception and channel overflow [17].
(2) errors resulting in incorrect service (with respect to the service specification) provided by the designed protocol to the service user. We refer to such errors as *semantic errors*.

Two major approaches to protocol design have been proposed and applied: the *analysis* approach and the *synthesis* approach [17].

In the *analysis* approach, the protocol designer starts with a preliminary version of the protocol in which the validation aspects are often overlooked. Protocol validation is then performed in a separate post-design activity and is based on analysis techniques intended to detect errors and omissions in the design. The sequence of re-design, analysis, error detection and correction is applied iteratively until the protocol design becomes error-free. Analysis techniques have been classified under five categories [14]: closed covers [4], reachability analysis [12,16], divide-and-conquer [8,13], partial state exploration [9,10], and localized [1,7,15].

In the *synthesis* approach, various methods are used to construct or complete a partially specified protocol design such that the interactions between the constructed or completed protocol entities proceed without encountering any logical error and (ideally) provide the requested services. Furthermore, using a synthesis method, the validation of the protocol is a by-product of the method itself. Synthetic design methods have been surveyed and compared in [11].

The purpose of this paper is to report on a simple and efficient validation technique based on a localized approach for the detection and correction of logical design errors. The technique uses the concept of synthesized and inherited traces at nodes of a tree-based model of communicating entities. Moreover, this technique can be integrated into an incremental synthetic protocol design methodology which guarantees freedom from logical errors and satisfies the service requirements.

The major advantage of using a localized approach is that the validation procedure is not based on a global reachability analysis, therefore, the explosion and termination problems are avoided. Furthermore, the advantages of our technique over other localized validation techniques are: i) the global information required at each local entity is minimal and very trivial to obtain and maintain, ii) the computations involving local and global information are very simple, and iii) the technique can be used to validate any labelled transition system-based specifications.

This paper is organized as follows. In Section 2, the Acceptance Tree, Traces and Sets *(ATTS)* model is introduced and operations on the model and its components are defined. In Section 3, the validation technique is described. In Section 4, the technique is applied to the X.25 Call Setup procedure. Section 5 concludes the paper and presents some issues which are currently under research.

2. THE MODEL: ACCEPTANCE TREES, TRACES AND SETS (ATTS)

In this section, we first introduce the *ATTS* model, then define some operations to manipulate the model, and derive some key properties.

The *ATTS* model is based on the concept of *acceptance trees* introduced in [5]. The semantics of the model is quite similar to that of the acceptance trees in the case of deterministic systems. However, a minor modification is

introduced for non-deterministic systems. In fact, the ATTS model is an adaptation of the acceptance trees to communication protocol design and validation activities.

Definition 1. An *ATTS* is a tree that carries semantic information useful to perform computations on the behaviour of an entity it describes. The nodes of an ATTS tree are labelled by an acceptance set and an acceptance trace. An acceptance set at node n, denoted by $A(n)$, includes all events that the process can perform at n. An acceptance trace at node n, denoted by $t(n)$, is the sequence of event occurrences that leads to n starting from the root node of the ATTS. This trace uniquely identifies a given node in the ATTS tree. Therefore, a node n is uniquely characterized by the tuple $<t(n), A(n)>$. Note that t(n) of the root node is a null trace (ε) and $A(n)$s of the leaves are empty sets (Φ). Each root node identifier is a tree name, however each leaf node identifier may be either: i) own tree name (recursive or cyclic behaviour), ii) a new tree name (enabling a new phase), or iii) *end* or *exit*.

Internal non-deterministic behaviours can also be expressed in the ATTS model. For a node n, at which the process can proceed non-deterministically to node n', $A(n)$ will include events of the form *ie* for every e in $A(n')$. Moreover, $t(n')$ will be equal to $t(n)$. The ATTS model is general enough to express deterministic or non-deterministic behaviours, synchronous or asynchronous and terminating or non-terminating (recursive) communicating systems. Furthermore, the model can easily express any labelled transition-based systems: Garavel [2] transforms LOTOS [6] specifications to a labelled transition system, and Yuan [15] developed a procedure for transforming a finite state machine specification into a tree.

Definition 2. The *final* traces of process P, denoted by F(P), are traces at the leaf nodes of the ATTS representation of P.

Definition 3. *Last* (t) is the last event in the trace or event list t.

Definition 4. An event in a trace can be either a *send* event, denoted by *!e*, or a **receive** event, denoted by *?e*. If e is a *send (receive)* event, then σ(e) is *send (receive)*. If σ(e) = *send (receive)*, then σ(^e) = *receive (send)*.

Definition 5. Two events e1 and e2 are said to be *matching* events, denoted by *Matching* (e1, e2), if i) σ(e1) = *receive* and σ(e2) is either *receive* or *send* such that e1 = e2, or ii) σ(e1) = *send* and σ(e2) = *receive* such that e1 = e2.

Definition 6. The length of a trace t, denoted by *Length* (t) (or L(t)), is the number of events in the trace or event list t.

Definition 7. *Height* (n) (or $H(n)$) is the height of the node n in the corresponding *ATTS*. H(n) is equal to $L(t(n))$.

Definition 8. The first x events of a trace t is denoted by *(x) t*. *Init*(t) is *(L(t) -1) t*.

Definition 9. The last x events of a trace t is denoted by *(x-) t*.

Definition 10. The position of event *e* in the trace *t* is denoted by *Pos* (*e*, *t*).

Definition 11. The *ith* event in the trace *t* (with 0 < i <= *L*(*t*)) is denoted by [i]*t*.

Definition 12. The *concatenation* of two traces t1 and t2 is denoted by t1@t2.

Definition 13. The number of occurrences of an event *e* in a trace *t* is denoted by #(*e*, *t*).

Definition 14. An event e_k is said to be *feasibly executable* at process behaviour Pi communicating with Pj, that is, e_k is in A(Pi), if:

1. $\sigma(e_k)$ = *send*, or
2. $\sigma(e_k)$ = *receive* and #(?e_k, t(Pi)) <= #(!e_k, t(Pj)) and,
   ∀ e_i, e_k in t(Pj) / $\sigma(e_i) = \sigma(e_k)$ = *send*, *Pos*(e_i, t(Pj)) < *Pos*(e_k, t(Pj)) and *Pos*(?e_i, t(Pi)) < *Pos*(?e_k, t(Pi)).

Definition 15. For two traces t1 and t2 (in distinct processes), the *first-send-not-received* (t1,t2) is the event !e1 such that ∀ !e1, !e2 in t1, ∄ ?e1, ?e2 in t2 and Pos(!e1, t1) < *Pos*(!e2, t1).

Definition 16. The characterization of *compatible* traces depends on whether the communication progresses synchronously or asynchronously in the system. This is related to the degree of coupling between the communicating entities: a close or tight coupling implies a synchronous communication, and a loose coupling implies an asynchronous communication. The general case corresponds to asynchronous communication, in which a time delay occurs between the transmission of event *e* and its eventual reception at a peer entity. Synchronous communication corresponds to the special case in which the time delay is null.

In the general case, two traces t1 and t2 are said to be *compatible* if:

1. ∀ !e in t1, ∃ ?e in t2 / *Pos*(?e,t2) >= *Pos*(!e, t1), and ∀ ?e in t2, ∃ !e in t1 / *Pos*(!e,t1) <= *Pos*(?e, t2).
2. ∀ !e in t2, ∃ ?e in t1 / *Pos*(?e,t1) >= *Pos*(!e, t2), and ∀ ?e in t1, ∃ !e in t2 / *Pos*(!e, t2) <= *Pos*(?e, t1).
3. If $!e_i$ and $!e_j$ in t1 and *Pos*($!e_i$,t1) > *Pos*($!e_j$,t1) then *Pos*(?ei, t2) > *Pos*(?ej, t2).

Moreover, if t1 and t2 are final traces in *F*(P1) and *F*(P2), respectively, they are compatible if they satisfy the three conditions listed above and if their leaf node identifiers are the same, that is, if t(n1) = t1 and t(n2) = t2, Id(n1) = Id(n2).

This general definition corresponds to the case of both synchronous and asynchronous communication between processes P1 and P2. In the case of synchronous communication, the general case is reduced to the following rule:

t1 and t2 are compatible if: t1 = ^t2, that is: ∀ !(?)e_i in t1, ∃ ?(!)e_i in t2 / *Pos*(e_i, t1) = *Pos*(e_j, t2).

Lemma 1. For successful synchronous communication between processes P1 and P2, the following 2 conditions are satisfied: i) ∀ t1 in *F*(P1), ∃ t2 in *F*(P2) / t1 = ^t2, and ii) ∀ t2 in *F*(P2), ∃ t1 in *F*(P1)/ t2 = ^t1.

Lemma 2. In the case of asynchronous communication between P1 and P2, the following 2 conditions are satisfied: i) ∀ t1 in *F*(P1) , ∃ t2 in *F*(P2) / t1 and t2 are two compatible traces, and ii) ∀ t1 in *F*(P2), ∃ t2 in *F*(P1)/ t1 and t2 are two compatible traces.

Note that Lemma 1 is a necessary and sufficient condition to avoid deadlocks and unspecified receptions in the design, however, Lemma 2 is only a necessary (not sufficient) condition.

Definition 17. Two traces, t1 and t2, are said to be simultaneously *executable*, if they can be decomposed into t'1, t"1 and t'2, t"2, respectively, in which, t'1 and t'2 are compatible traces (or empty traces) and every event in t"1 and t"2 is a feasible executable event.

Definition 18. *Trace decomposition.* From the definitions of compatible and executable traces, we can conclude that traces t1 and t2 of two communicating processes can be decomposed into (t'1@t"1) and (t'2@t"2), respectively, in which the subtraces t'1 and t'2 are compatible and the subtraces t"1 and t"2 are executable. Since the lengths of compatible subtraces must be identical, only three possibilities exist when decomposing t1 and t2:

1. t'1 and t'2 are empty, then t1 and t2 are executable traces but not compatible.
2. t"1 and t"2 are empty, then t1 and t2 are compatible, therefore, executable.
3. no subtrace is empty, then t1 and t2 are said to be executable but not compatible.

Definition 19. Two traces, t1 and t2, are said to be *colliding*, denoted by *Colliding*(t1,t2), if ∃ x (x> 0) / (x) t1 and (x) t2 are not subtraces of compatible traces t'1 and t'2, respectively.

Lemma 3. For every pair of incompatible colliding traces t1 in *F*(P1) and t2 in *F*(P2), synchronizing traces must exist in *F*(P1) and *F*(P2) for handling collisions.

Corollary 1. Unspecified receptions occur if and only if there exist incompatible colliding final traces that do not satisfy Lemma 3.

3. THE VALIDATION TECHNIQUE

In this section, we first describe the features of the validation technique, and then we outline two algorithms for the implementation of the technique.

3.1. Features of the validation technique

The validation technique consists of two phases. In the first phase, the ATTS representations of the processes to validate are augmented with some semantic information in the form of traces of two types: synthesized and inherited. In the second phase, the detection and correction of logical design errors are performed.

Given the ATTS representations of two communicating entities P1 and P2, two types of traces are computed for each node of the ATTSs: the inherited trace carries information about the ATTS where the node exists, and the synthesized traces transfer information about traces from the other ATTS. The validation procedure is essentially based on the examination of the inherited and the synthesized sets of traces at each of the nodes of the ATTS representing the process under validation. Synthesized traces at one node are propagated upwards to their (unique) parent node depending on some rules for compatibility and executability. The errors detected by the procedure can be corrected before proceeding with the validation.

The validation is done in a bottom-up fashion which results in an early detection of complex erroneous behaviours and an efficient technique to detect errors since a trace at a leaf node identifies a unique path in the tree. The validation procedure is then applied twice: first, P1 is validated with respect to P2, and then, P2 is validated with respect to P1 (or vice versa). Very limited coupling between the two entities during the validation process is required, therefore this technique is classified under the localized approach to validation. In addition, the validation of the entities can be executed in parallel and the computations that are performed during the validation are very simple. Finally, the technique could be extended to validate against the service requirements in addition to logical errors.

3.2. Algorithms

In the following, we present two algorithms which implements the validation technique: Algorithm-G for the generation of inherited and synthesized traces at nodes of an ATTS representation of a process, and Algorithm-D for the detection and correction of logical design errors in the given protocol.

Algorithm-G for the generation of inherited and synthesized trace sets:

Input: ATTS representation of communicating entities to validate.
Output: ATTS representation augmented with inherited and synthesized traces.

Steps:

1. Initialization. For each of the communicating entities to validate, generate its ATTS representation.
2. Generate the inherited and synthesized traces at the leaf nodes of each of the ATTSs.
   For a leaf node n in P1's ATTS:
   . $\mathbf{Ip}(n) = t(n)$.
   . $\mathbf{Sp}(n) = \{ t / t \in F(\text{P2}) \text{ and } H(n) >= L(t) \} \cup$
   $\{t / \forall \text{ tr} \in F(\text{P2}) \text{ and } H(n) < L(\text{tr}), t = (H(n))\ \text{tr} \}$.
3. Generate the inherited and synthesized traces at other nodes.
   For a non-leaf node n in P1's ATTS:
   . $\mathbf{Ip}(n) = t(n)$.
   . $\mathbf{Sp}(n) = \{ t \in \mathbf{Sp}(n\text{-}1) / L(t) < H(n) \} \cup$
   $\{ Init(t), t \in \mathbf{Sp}(n\text{-}1) / Matching(Last(t), Last(\mathbf{Ip}(n\text{-}1)),$ and
   $Init(\mathbf{Ip}(n\text{-}1))$ and t are executable$\}$
   Where $\mathbf{Sp}(n\text{-}1)$ denotes the union of the sets of synchronized traces at nodes dependent of n.

Algorithm-G shows that the only information transferred from the ATTS representing P2 is the synthesized sets of traces at the leaves of P1. These traces correspond to the inherited traces at the leaves of P2. The synthesized sets of traces at nodes other than the leaves are generated using the synthesized sets of dependent nodes propagated upwards in the ATTS.

Definition 20. *Collision handling synthesis (analysis) rule.* Let t1 in F(P1) and t2 in F(P2) be two colliding traces. Suppose t1 and t2 are decomposed into t'1 @ !e1 @ t"1 and t'2 @ !e2 @ t"2, respectively, where t'1 and t'2 are compatible traces. The handling of a collision caused by the simultaneous transmission of e1 and e2 can be done by considering (verifying that) one of the three cases:

a. P1 has a higher priority than P2. Add t11 = t'1 @ !e1 @ ?e2 @ t"1 to F(P1), and t21 = t'2 @ !e2 @ ?e1 @ ^t"1 to F(P2).
b. P2 has a higher priority than P1. Add t11 = t'1 @ !e1 @ ?e2 @ ^t"2 to F(P1), and t21 = t'2 @ !e2 @ ?e1 @ t"2 to F(P2).
c. P1 and P2 have equal priorities. Add t11 = t'1 @ !e1 @ ?e2 @ t"1 @ ^t"2, t12 = t'1 @ !e1 @ ?e2 @ ^t"2 @ t"1 to F(P1), and t21 = t'2 @ !e2 @ ?e1 @ t"2 @ ^t"1, t22 = t'2 @ !e2 @ ?e2 @ ^t"1 @ t"2 to F(P2).

Note that cases a and b correspond to the winner/loser approach to synchronization introduced by Gouda [3]. However, case c corresponds to the no-winner/no-loser collision resolution (which is characteristic of multiple concurrent updates in a distributed database application).

Lemma 4. The application of the collision handling rule to every pair of colliding traces avoids the occurrence of unspecified receptions.

Lemma 5. Only incompatible colliding traces can be used to detect unspecified receptions.

Based on Lemma 5, the following optimization rule can be applied for the

elimination of useless synthesized traces generated by Algorithm-G at a node n of an ATTS:

$\forall\, t \in \mathbf{Sp}(n)$, if *not-Colliding* ($t$, $\mathbf{Ip}(n)$) is true then eliminate t from $\mathbf{Sp}(n)$.

Lemma 6. A protocol design causes an overflow iff $\exists$ t1 (t2) in F(P1) (F(P2)) / $\forall$ e in t1 (t2), $\sigma(e)$ = send.
Before proceeding to Algorithm-D, we state a theorem for proving the correctness of the validation technique.

Theorem 1. The satisfaction of the rules for compatibility (Lemmas 1 and 2), collision handling (Lemmas 3,4, and 5) and overflows (Lemma 6) guarantees that deadlocks, unspecified receptions and overflows are avoided.

Algorithm-D for the detection and correction of logical design errors:

Input: ATTS representation augmented with inherited and synthesized traces.
Output: ATTS representation free from deadlock and unspecified receptions.

Steps:
1. For each t1 in F(P1) there must exist t2 in F(P2) such that t1 and t2 are compatible.
   Similarly, for each t2 in F(P2), there must exist t1 in F(P1) such that t1 and t2 are compatible.
2. Detect the existence of *unspecified receptions* at each node of the ATTS.
   At a node n, let t be in $\mathbf{Sp}(n)$.
   . If *nonMatching*(*Last*(t), *Last*($\mathbf{Ip}(n)$)) and *First-sent-not-received*(t, $\mathbf{Ip}(n)$) = !e and ?e $\notin$ $A(n)$ then Unspecified Reception (?e) at n.
3. If at node n a reception is missing, add new traces according to **Definition 20**.

Step 1 guarantees that no deadlock is possible for synchronous communication. Step 2 checks if, for every pair of colliding traces, all possible traces for handling the collision are present in the behaviour of the entity to validate. Step 3 generates all those synchronizing traces absent from the behaviour of the entity as a result of the detection of colliding traces and unspecified receptions.

The correction of an unspecified reception (?e) at a node n, characterized by $<t(n), A(n)>$, is done by the generation of a new final trace $t = t(n)$ @ ?e @ t'. The trace t must be compatible with another trace in F(P2). Furthermore, t must be a correct trace with respect to the service definition, hence, a semantically valid trace. Since the new final trace in F(P1) is compatible with another trace in F(P2) and because of Lemma 5, the re-application of the validation is not necessary.

In the case of non-deterministic ATTSs, all internal events i must be converted to !δ (send dummy event), where δ is not to be received by any other ATTS. Then, the two algorithms described above are applicable for the

validation of the non-determinstic communicating entities.

The validation technique for one ATTS can be applied in two ways: 1) two-phased: first apply Algorithm-G to generate all synthesized traces at all nodes, then apply Algorithm-D for the detection of logical errors, or 2) interleaved: for every node, first apply Algorithm-G, then apply algorithm-D. Furthermore, the validation of the ATTSs can be performed concurrently. But based on Lemma 2, if the validation of the ATTSs is applied sequentially, the validation of one ATTS may optimize the validation of the other ATTS. The termination of both algorithms is also guaranteed since both the number of final nodes and the number of pairwise incompatible colliding traces is finite.

4. EXAMPLE

In this section, we demonstrate the application of the validation technique to the Call Setup procedure of X.25. The ATTS representations of an incomplete specification of the DTE and DCE involved in the Call Setup procedure of X.25 is given in Figure 1. The application of the technique detects the omission of possible traces and suggests possible corrections, as shown in Figure 2. The example shows, informally, how the technique can be applied to the synthesis of new traces or behaviours, while satisfying the service requirements.

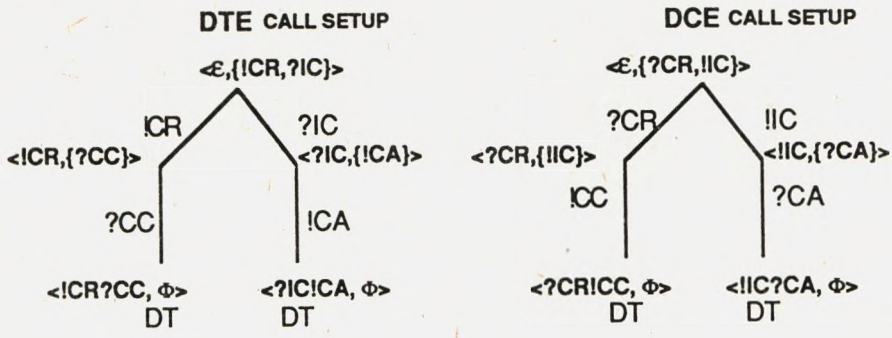


Figure 1. ATTS representations of DTE and DCE.

The application of Algorithm-D indicates that two unspecified receptions ?IC and ?CR exist in DTE and DCE, respectively. Two new traces must be generated: !CR?IC @ t1 in DTE and !IC?CR @ t2 in DCE. Based on Lemma 2, these two traces must be compatible. t1 and t2 are chosen according to the protocol requirements. For example, a reception of an incoming call (?IC) by the DTE after the transmission of a connection request (!CR) by the same DTE must be ignored by the DTE. This corresponds to the (English) procedures for Incoming Call handling specified in the X.25 protocol. This imposes that the new traces should be: !CR?IC?CC in DTE and !IC?CR!CC in DCE.

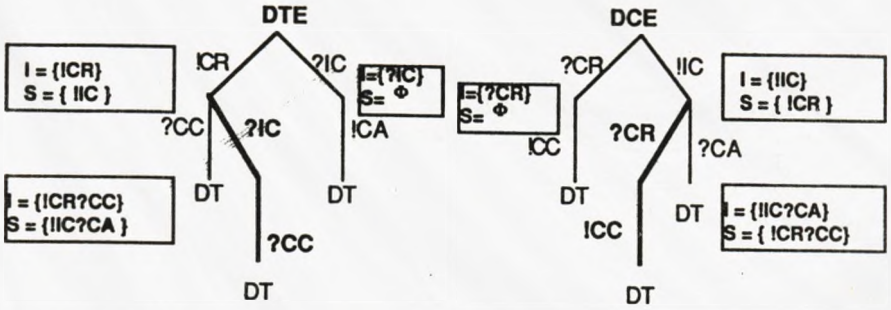


Figure 2. Synthesized and inherited traces at nodes of DTE and DCE.

5. CONCLUSIONS AND FUTURE WORK

In this paper, a protocol validation technique was proposed for detecting logical errors such as deadlocks and unspecified receptions. The technique is very simple and requires very little computation on the model of communicating entities. Very limited coupling between the communicating entities is required during validation, therefore, the explosion problems of traditional global validation techniques are avoided. Error correction is based on the generation of new event occurrences, hence the emphasis on generation (or synthesis) of new traces in the model. Therefore, we claim that the technique can be integrated into a synthetic design method. The validation technique can also be applied to non-deterministic behaviours in the communication model and can be generalized to involve more than two communicating entities. Finally, the tree-based modeling formalism (ATTS model) we used can be applied to any labelled transition system.

The advantages of our technique over other global and localized validation techniques are: i) the global information required at each local entity is minimal and very trivial to obtain and maintain, ii) the computations on local and global information are very simple, and iii) the technique can be applied to any labelled transition system-based specification. The main limitation of the technique and the model at their current stages is that no validation against the service requirements is performed. In our opinion, this can be achieved when we integrate the validation technique into a synthetic design method, in which existing and synthesized traces are validated against the service requirements.

ACKNOWLEDGEMENTS.

The authors gratefully acknowledge support of this work by the Telecommunications Research Institute of Ontario (TRIO) and the Natural Sciences and Engineering Research Council (NSERC).

REFERENCES

[1] D. Brand and P. Zafiropulo, "On communicating finite state machines", *Journal of the ACM*, Vol. 30, No. 2, April 1983, pp. 323-342.

[2] H. Garavel and E. Najm, "TILT: from LOTOS to labelled transition systems", in *The formal description technique LOTOS*, pp. 327-336, North-Holland, 1989.

[3] M.G. Gouda and Y.T. Yu, "Protocol validation by maximal progress state exploration", *IEEE Trans. on Commun*, Vol. COM-32, 1983, pp. 94-97.

[4] M.G. Gouda, "Closed covers: To verify progress of communicating finite state machines", *IEEE Trans. on Software Engineering*, Nov. 1984, Vol. SE-10, No.6, pp. 846-855.

[5] M. Hennessy, "Acceptance trees", *Journal of the ACM*, Vol. 32, No.4, Oct.1985, pp. 896-928.

[6] ISO- Information processing systems - Open systems interconnection- 'LOTOS- A formal description technique based on the temporal ordering of observational behaviour'. IS 8807. 1989.

[7] Y. Kakuda et al., "A new algorithm for fast protocol validation", *Proc. COMPSAC'86, IEEE,* 1986, pp. 228-236.

[8] S.L. Lam and U. Shankar, "Protocol verification via projections", *IEEE Trans. on Software Engineering*, Nov. 1984, Vol. SE-10, No.6, pp. 325-342.

[9] F.J. Lin, P.M. Chu, and M.T. Liu, "Probabilistic verification using reachability analysis: The state space explosion problem and relief strategies", *ACM SIGCOMM'87 Symp.*, Aug. 1987.

[10] N.F. Maxemchuck and K. Sabnani, "Probabilistic verification of communication protocols", in *Seventh IFIP Intern. Symp. on Protocol Specification, Testing, and Verification*, May 1987, pp. 307-320.

[11] R. Probert, and Kassem Saleh, "Survey and critique of synthetic protocol design methods", *Technical Report TR-89-41*, Department of Computer Science, University of Ottawa, Canada, Oct. 1989.

[12] C.H. West, "An automated technique for communication protocol validation", *IEEE Trans. on Commun.*, Vol. COM-26, No. 8, Aug. 1978, pp. 1291-1295.

[13] S.T. Vuong and D.D. Cowan, "A decomposition method for the validation of structured protocols", in *Proc. of INFOCOM Conference*, April 1982.

[14] M.C. Yuang, "Survey of protocol verification techniques based on finite state machine models", in *Proc. of Computer Networking Symp.*, 1988, pp. 164-172.

[15] M.C. Yuang and A.Kershenbaum, "Parallel protocol verification using the localized approach: the two-phase algorithm", in *Ninth IFIP Intern. Symp. on Protocol Specification, Testing, and Verification*, June 1989.

[16] P. Zafiropulo, "Protocol validation by duologue-matrix analysis", *IEEE Trans. on Commun.*, Aug. 1978, Vol. COM-26, No. 8, pp. 1187-1194.

[17] P. Zafiropulo, et al., "Towards analyzing and synthesizing protocols", *IEEE Trans. on Commun.*, April 1980, Vol. COM-28, No. 4, pp. 651-661.

LIVENESS VERIFICATION AS A TOOL FOR PROTOCOL VALIDATION

dr. László Zs. Varga

Central Research Institute for Physics
Budapest P.O.Box 49
1525 Hungary

This paper presents a new algorithm called Liveness Validation (LV) for the validation of the global state transition graphs of protocols. The LV algorithm detects more complex validation properties and is not less effective than the reachability analysis methods. First we will show that the liveness property implies all the features of a valid protocol and then we present the LV algorithm. We will prove the properties and the exactness of the LV algorithm and evaluate the efficiency of LV. Our work is a part of the efforts to construct the PROCONSUL protocol consultant system, which is an automated protocol development environment.

1. INTRODUCTION

The design of a protocol means the explicit construction of two or more interacting asynchronous processes. These processes together form an implicitly given system, which is very difficult to understand. While the behaviour of the component processes are relatively easy to understand, the whole system may contain hopelessly many improbable sequences of events, which are easily overlooked and may cause errors. There is need for automated validation of the whole system.

The automated validator symbolically executes a formalized description of the protocol and examines the generated states of the system if they do not satisfy some not desired properties, e.g. deadlock property. There are two basic types of automated validators: The *partial validators* [1][2] generate only a part of the reachable states of the system. They do not prove the absence of errors, only indicate their presence with certain probability. The *exhaustive validators* [3][4][5][6][7] generate the complete state space of the analyzed system. They are able to indicate the absence of errors, but their applicability is limited by the number of states. With the performance improvement of computers this limit becomes weaker.

The automated validators, especially the exhaustive validators are usually based on the reachability analysis, which is simple to implement, but is unable to detect complex validation properties like liveness. Reachability analysis only detects the simple properties like deadlock or buffer overflow freeness.

The aim of this paper is to present a new exhaustive validation algorithm, which detects more complex validation properties and is not less effective than the reachability analysis methods. First we will show that the most important property is liveness, since it implies the other important validation properties. Then we will present our algorithm called LV (Liveness Validation) algorithm and demonstrate it on a simple protocol. We will analyze the

algorithm and prove that it always terminates in finite graphs, generates each transition of the graph exactly once, and detects liveness if and only if the graph satisfies the liveness property. We will also discuss the efficiency of the LV algorithm. Our work is a part of the efforts to develop an automated protocol development environment called PROCONSUL [8][9].

In section 2 we will examine the validation properties, in section 3 we will present the LV algorithm, in section 4 the properties of the algorithm will be proved, and in section 5 the efficiency will be discussed. Section 6 sums up the conclusions.

2. PROTOCOL VALIDATION AND LIVENESS

We model the protocols with communicating finite-state machines [10] shown on Figure 1. In the model there are finite-state machines (Process P_i), which can receive and place messages from and to FIFO channels (C(i,j)). The finite-state machines are described by their state transition graph. A pair of example graphs is shown on Figure 2. The states are numbered and each transition of the graph has a label showing the content of the message by a mnemonic and the direction of the message by + or - sign. The state of the communicating system is described by a global state (q_1,c_{12},c_{21},q_2), which consists of q_i: the state of Process P_i and c_{ij}: the list of messages in channel C(i,j). The empty channel is denoted by E. Executing the transitions of the finite-state machines one can generate the global state transition graph (GSTG). Figure 3 shows the GSTG of our example protocol.

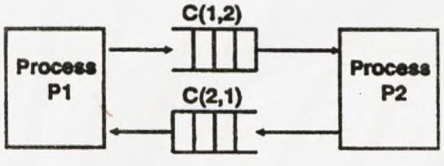


FIGURE 1 Communicating Finite State Machines

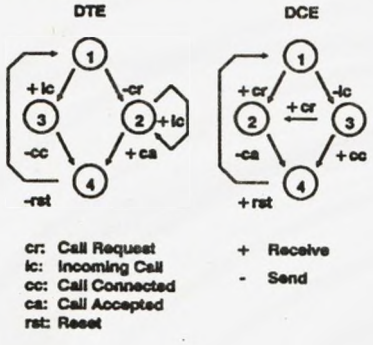


FIGURE 2 State Transition Graphs of Finite State Machines

There are some properties, which (if they exists in the GSTG) indicate errors of the protocol [11]. They are the following:

(1) *unspecified reception freeness*: there is no such state in the GSTG, in which one of the processes may receive a message, but in the state transition graph of the process the destination state for this message reception is not specified.

(2) *nonexecutable transition freeness*: there is no such transition in the state transition graph of the process, which does not give a transition in the GSTG.

(3) *buffer overflow freeness*: in each of the states of the GSTG the number of the messages in the channels is under a limit.

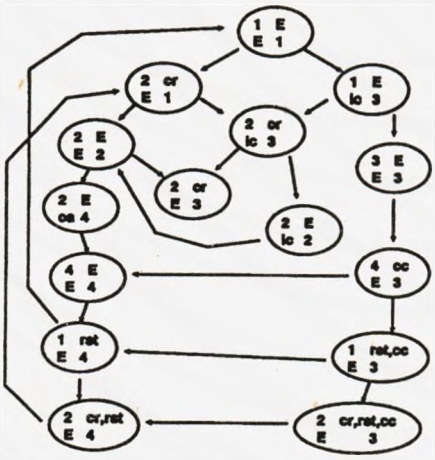


FIGURE 3 Global State Transition Graph

(4) *deadlock freeness*: there is no such state in the GSTG, which has no successor.

(5) *recovery from failures*: after a failure the protocol will return to normal execution within a finite number of steps.

(6) *liveness*: from each reachable state any other reachable state is reachable in the GSTG.

The validation of the (1), (2), and (3) properties can be regarded as a "side effect" of an exhaustive validation. The validation of (1) is executed when the new states are generated in the GSTG. If a new state can not be determined, then (1) is not true. Property (2) can also be validated when the GSTG is generated. The transitions from the state transition graphs of the processes that are used in the GSTG state generation can be marked. If at the end of the exhaustive validation there is an unmarked transition, then (2) is not validated. The (3) property can also be checked at the generation of a new state in the GSTG. If the number of messages in a channel of a new state exceeds a limit, then (3) does not hold.

The exhaustive validation methods has to concentrate on the (4),(5),(6) properties. Reachability analysis can validate (4). If a global state s can be reached and no other reachable state can be generated from s, then it is a deadlock state. Reachability analysis generates a tree, that covers the GSTG, but it does not say anything about the structure of the GSTG, so it is unable to validate (5) and (6). But if we could validate (6), then we could also validate (4) and (5).

Lemma 1. If liveness is true for a GSTG, then recovery from failures and deadlock freeness is also true.

Proof: Recovery from failures: Let us denote the initial global state s_s and the global state entered on failure s_f. Both s_s and s_f are reachable, so if liveness is true, then s_s can be reached from s_f, i.e. the protocol can recover from failure.

Deadlock: If liveness is true, then from any reachable state s_r the initial state s_s can be reached, i.e. each reachable state s_r has at least one successor, so there is no deadlock. #

From the above considerations we can see, that liveness is the most important validation property. It implies that the GSTG does not contain closed loops like in Figure 4. Deadlock is a special case of closed loop, it contains only one state.

The problem with liveness validation is that at a first glance it may require more computing power than reachability analysis. In the next section we will try to construct an algorithm, which refutes this.

3. THE LV ALGORITHM

Our LV algorithm is somewhat similar to reachability analysis, i.e. it will generate a reachability tree that covers the GSTG, but this reachability tree will be a special one. When this tree will be completed, we will know, that if s_i and s_j are states in the GSTG and there is an arc of the reachability tree from s_i to s_j, then there is a path in the GSTG from s_j to s_i. It means, that this tree will be such, that if in the tree an s_r state is a descendant of the initial state s_s, then this s_r state is reachable and there is a path in the GSTG from s_r to s_s and thus from s_r every reachable state is reachable. If our tree covers the whole GSTG, then liveness is true for the GSTG.

To demonstrate how LV works let us take the directed graph of Figure 5 as an example GSTG. (s_i = state i, t_i = transition i) To construct the special reachability tree the LV algorithm will use a list, the members of which are alternatingly states and transitions of the GSTG. This list is called as OPEN_PATH, the sates and transitions in this list will be marked as OPEN. When the LV algorithm finds a state or a transition, that will be a member of the special reachability tree it generates, it will mark that state or transition as CLOSED. If the state or transition is on the OPEN_PATH then the marking will be not CLOSED but POTENTIALLY_CLOSED.

The operation of LV is a depth-first backtrack search. It starts from the initial global state, takes a transition not investigated yet, applies this transition to the initial state, generates the new state, and adds this transition and state to

FIGURE 4 No Closed Loops if Liveness

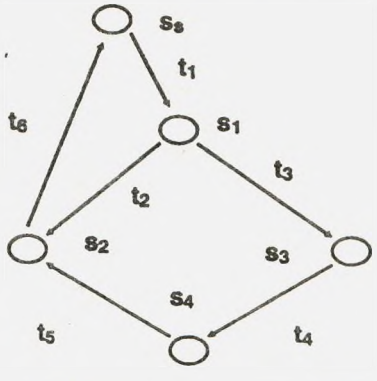
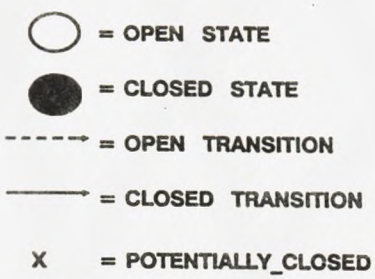


FIGURE 5 An Example GSTG

the OPEN_PATH. Then it repeats the same with the newly generated state until there will be a loop on the OPEN_PATH, i.e. a newly generated state can be found on the OPEN_PATH. This situation is shown on Figure 6. At this point LV backtracks: marks the states and transitions in the loop as POTENTIALLY_CLOSED and removes the last transition from the OPEN_PATH. Then the depth-first backtrack search continues from the last state of the OPEN_PATH. This situation is shown on Figure 7.

LV will also backtrack, when in the last state of the OPEN_PATH all of the transitions have been investigated as on Figure 7. In this case LV can backtrack only if the last transition on the OPEN_PATH is POTENTIALLY_CLOSED, otherwise LV fails, which means that liveness is not true for the graph. If LV can backtrack, then it will mark the last state and transition on the OPEN_PATH as CLOSED and removes them from the OPEN_PATH. This phase is shown on Figure 8.

LV has to backtrack not only when a newly generated state is the same as one on the OPEN_PATH, but also when it is the same as one of the CLOSED states. This situation is shown on Figure 9. Though in this case there is no explicit loop in the OPEN_PATH, we know, that there is an implicit loop,



Explanation of signs on FIGURE 6 ... 11

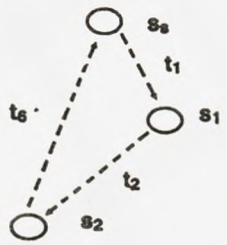


FIGURE 6 Loop in OPEN_PATH

FIGURE 7 There are no More Transitions at the Head

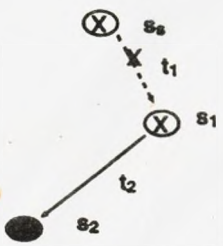


FIGURE 8 Closing a State

because we can ascend through CLOSED transitions to a state on the OPEN_PATH. Let us call this state on the OPEN_PATH as s_a. In this situation LV will do the following: It marks the states and transitions on the OPEN_PATH between s_a and the last state as POTENTIALLY_CLOSED and removes the last transition from the OPEN_PATH. This situation is shown on Figure 10.

LV finishes with success if it starts to backtrack from the initial state.

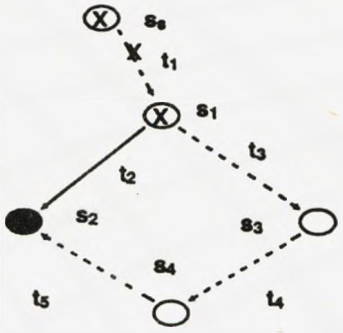


FIGURE 9 OPEN_PATH Reaches a CLOSED State

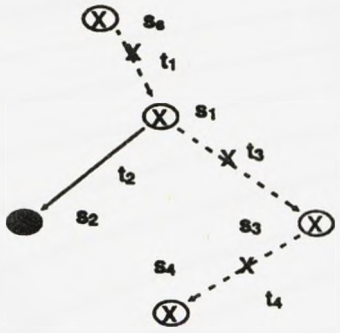


FIGURE 10 Backtrack from a CLOSED State

The detailed description of LV is the following:

algorithm LV

```
$1  make a list called OPEN_PATH consisting
    solely of the initial state, mark the initial state
    as OPEN
$2  make a tree called REACHABILITY_TREE
    consisting solely of the initial state
$3  STATE1 := head(OPEN_PATH)
    TR := a transition originating from STATE1,
     which has not been investigated yet
$4  if TR = NIL then
      a) mark STATE1 as CLOSED,
         remove the OPEN marking from STATE1,
         OPEN_PATH := tail(OPEN_PATH)
      b) TR1 := head(OPEN_PATH),
         OPEN_PATH := tail(OPEN_PATH)
      c) if TR1 = NIL then
             exit with SUCCESS
      d) if TR1 is not marked as
          POTENTIALLY_CLOSED then
             exit with FAIL
      e) mark TR1 as CLOSED
      f) goto $3
$5  OPEN_PATH := concatenate(TR.OPEN_PATH),
```

```
    mark TR as OPEN,
    STATE2 := apply(TR,STATE1)
$6  if STATE2 is not in REACHABILITY_TREE then
      a) OPEN_PATH := concatenate(STATE2,OPEN_PATH)
      b) mark STATE2 as OPEN
      c) add TR and STATE2 to REACHABILITY_TREE
      d) goto $3
$7  if STATE2 is marked as CLOSED then
      a) move backward on CLOSED transitions
              until the first OPEN state is found
      b) STATE2 := the first OPEN state
$8  mark the states and transitions on
    OPEN_PATH from STATE2 to the head of the
    OPEN_PATH as POTENTIALLY_CLOSED
$9  OPEN_PATH := tail(OPEN_PATH)
       /* remove TR from OPEN_PATH */
$10 goto $3

end_algorithm LV
```

4. PROPERTIES OF LV

In this section we will discuss and prove the properties and correctness of the LV algorithm. Since the global state transition graphs that have to be validated are usually finite graphs, we will first prove the following lemma:

Lemma 2: The LV algorithm will always terminate on finite directed graphs.

Proof: The LV algorithm can operate infinitely only if it infinitely loops on step $3. Since in step $3 a new TR transition is always taken and there are finite number of transitions, it cannot operate infinitely. #

The next theorem states, that successful termination of LV is sufficient for the validation of the liveness of a graph. In order to prove Theorem 1 we will prove few lemmas.

Theorem 1: If G is a finite directed graph and LV algorithm terminates with SUCCESS on G, then liveness is true for G.

Lemma 3: If t_{ij} is a transition of a finite directed graph from state s_i to state s_j and t_{ij} is marked as POTENTIALLY_CLOSED or CLOSED, then there is a path in the directed graph from s_j to s_i.

Proof: A state can be marked as POTENTIALLY_CLOSED in step $8 of LV. At the beginning there is no CLOSED transition, so the first transition can be marked as POTENTIALLY_CLOSED, only if STATE2 in step $5 was marked as OPEN. In this case the OPEN_PATH formed a path from s_j to s_i. A transition can become CLOSED only in step $4e and only if it is marked as POTENTIALLY_CLOSED, so there is a reverse path in the graph for the CLOSED transitions. Thus if STATE2 in step $5 is CLOSED, then there is a path from that state to an ascendant of s_i on the OPEN_PATH and thus from s_j to s_i. #

Lemma 4: If an s state is marked as CLOSED, then all of the immediate successors of s are added to the REACHABILITY_TREE. Moreover if this s state is not the initial state and LV does not exit with fail, then there is exactly one t_s transition to s, which is marked as CLOSED.

Proof: State s can become CLOSED only in step $4a and only if there is no such TR transition originating from s, which has not been investigated in step $3. If a TR transition has been investigated and the destination state to which this transition takes was not in the REACHABILITY_TREE, then this destination state is added to the REACHABILITY_TREE in step $6. A transition can be marked as CLOSED only in step $4e after step $4a. If LV does not exit, then the transition CLOSED in step $4e is the only t_s transition to s, which is marked as CLOSED. #

Lemma 5: If G is a finite directed graph and LV algorithm terminates with success on G, then the REACHABILITY_TREE contains all the reachable states from the initial state and all the states and transitions of the REACHABILITY_TREE are marked as CLOSED.

Proof: LV can terminate with SUCCESS, only if the OPEN_PATH is empty and the initial state is marked CLOSED in step $4a. If the OPEN_PATH is empty, then there is neither OPEN nor POTENTIALLY_CLOSED state, because such states can only be found on OPEN_PATH, so all the states of REACHABILITY_TREE are CLOSED. If the initial state is CLOSED, then from Lemma 4 all of its immediate successors are added to the REACHABILITY tree and the transitions leading to them are CLOSED. Since the states in REACHABILITY_TREE are CLOSED, all of the immediate successors of the initial state are added to the REACHABILITY_TREE and the transitions leading to them are CLOSED. We can continue the same way, so all the reachable states from the initial state are added to the REACHABILITY_TREE. #

Proof of Theorem 1: From Lemma 5 the REACHABILITY_TREE contains all the reachable states and all the transitions and states in this tree are CLOSED. This with Lemma 3 means, that there is a path in the G graph from each reachable state to the initial state and thus to each reachable state. So liveness is true for the G graph. #

The following theorem states, that successful termination of LV is necessary for the validation of the liveness of a graph.

Theorem 2: If liveness is true for a finite directed graph, then LV algorithm terminates with SUCCESS on G.

Proof: From Lemma 2 will LV terminate. Let us suppose, that it terminates with FAIL. At termination in step $4d there is a state STATE1, which is marked as CLOSED and the last transition on the OPEN_PATH is TR1, which takes to STATE1 and is not marked as POTENTIALLY_CLOSED. From Lemma 4 all of the immediate successors of STATE1 are added to the REACHABILITY_TREE. These successors cannot be OPEN, otherwise TR1 would be marked as POTENTIALLY_CLOSED. So the successors are all CLOSED. From these CLOSED successors we cannot ascend to the OPEN_PATH, otherwise TR1 would be marked as POTENTIALLY_CLOSED. Since the successors are CLOSED, from Lemma 4 all of the successors of the successors are added to the REACHABILITY_TREE. The same is true for the successors of the successors as for the immediate successors of STATE1, so we can conclude, that none of the states that are reachable from STATE1 can be on the OPEN_PATH. It means that the initial state can not be reached from STATE1, so liveness is not true for G. This is contradiction, so LV can not terminate with FAIL, it can terminate only with SUCCESS. #

5. EFFICIENCY OF LV

In this section we will discuss the complexity of LV. We will also discuss the first experimental results of the implementation of LV.

The complexity of reachability analysis consists of two parts: one is the generation of each transition in the GSTG, the other is the comparison of the newly generated states with the previous ones. For the second part an improved method is given in [3]. The complexity of the first part is proportional to the number of transitions in the GSTG.

The complexity of algorithm LV consists of three parts. The first two is the same as in reachability analysis, the third is the marking of the states and transitions. Since a state or a transition can be marked only once as OPEN, only once as POTENTIALLY_CLOSED and only once as CLOSED, the marking process is also proportional to the number of transitions in the GSTG. One might say, that the markings not only has to be placed, but also has to be checked many times in step $8 of LV. This can be avoided if we use pointers to jump through those transitions, which are marked as POTENTIALLY_CLOSED and would be repeatedly marked. Figure 11 shows the use of such pointers.

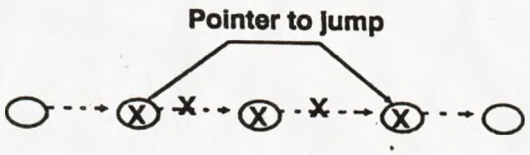


FIGURE 11 Jumping through States

The liveness validation method derived from the definition is the following: Let us generate a reachability tree from each state of the GSTG and check each pair of the states if one can be reached from the other. If the GSTG has n states, then the complexity of this liveness method is the following: n times the complexity of a reachability analysis plus n*(n-1) times a check. From the above considerations we can see that the complexity of the LV algorithm is not worse than that of the reachability analysis, but better than the liveness validation method derived from the definition.

The LV algorithm is implemented in the C programming language on an IBM AT machine and has been tested on sample graphs. The run time of LV on a graph with 2^4 nodes and 2^4 transitions from each node is approximately 2 minutes.

6. CONCLUSIONS

In this paper we have shown, that automated protocol validation is necessary and in theory the best way of validation is liveness validation. We proposed the LV algorithm for the validation of liveness and proved that the LV algorithm is sufficient and necessary for this. We showed, that the complexity of the LV algorithm is not worse than that of the reachability analysis.

The LV algorithm is planned to be a validator in the PROCONSUL system, which is an automated protocol development environment under development at the Central Research Institute for Physics.

ACKNOWLEDGEMENTS

The author wishes to thank Katie Tarnay and Géza Lõrincze for their help and their work to establish the possibilities for this work.

REFERENCES

[1] D.D Dimitrijevic, M.-S. Chen: "Dynamic State Exploration in Quantitative Protocol Analysis", in Proceedings of the Ninth IFIP WG 6.1 International Symposium on Protocol Specification, Testing, and Verification, 6-9 June 1989, Univ. of Twente-Enschede, The Netherlands

[2] C. West: "Protocol Validation by Random State Exploration", in Protocol Specification, Testing, and Verification, VI, B. Sarakaya, G. Bochmann (Editors), Elsevier Publishers B.V. (North Holland), 1986

[3] G.J. Holzmann: "An Improved Protocol Reachability Analysis Technique", Software-Practice and Experience, Vol.18 (2), February 1988

[4] G.J. Holzmann, J. Patti: "Validating SDL Specifications: an Experiment", in Proceedings of the Ninth IFIP WG 6.1 International Symposium on Protocol Specification, Testing, and Verification, 6-9 June 1989, Univ. of Twente-Enschede, The Netherlands

[5] P. Zafiropulo: "Protocol Validation By Doulogue-Matrix Analysis", IEEE Tr. on Communications, Vol. COM-26, No.8, August 1978

[6] C.H.West: "General Technique for Communications Protocol Validation", IBM Journal of Research and Development, Vol.22, No.4, July 1978

[7] J-R. Zhao, G. Bochmann: "Reduced reachability analysis of communicating protocols: A new approach", in Protocol Specification, Testing, and Verification, VI, B. Sarakaya, G. Bochmann (Editors), Elsevier Publishers B.V. (North Holland), 1986

[8] H. Koenig, K. Tarnay: "An Intelligent Protocol Workstation", Network Information Processing Systems (ed. Boyanov), North Holland Publ. Co., Amsterdam, 1989

[9] K.Tarnay, S. Wagner-Dibuz: "A Protocol Consultant", "Windows to the World" proceedings of the DECUS Europe Symposium 1988

[10] D. Brand, P. Zafiropulo: "On Communicating Finite-State Machines", Journal of the ACM, Vol.30, No.2, April 1983

[11] P.M.Merlin: "Specification and Validation of Protocols", IEEE Tr. on Communications, Vol. COM-27, No.11, November 1979

Experience in Using Estelle* for the Specification and Verification of a Fieldbus Protocol: FIP

M.L. BARRETTO, J.-P. COURTIAT, P. de SAQUI-SANNES

LAAS/CNRS
7 avenue du Colonel Roche
31077 Toulouse Cedex - France

E.mail: courtiat@laas.laas.fr, pdss@laas.laas.fr

ABSTRACT

Formal Description Techniques (FDTs) are the mandatory support of new software methodologies devoted to the design of complex distributed applications. This paper discusses an experience using the Estelle* FDT for the specification, simulation and verification of a fieldbus protocol (FIP standard proposal). The experience relies on the use of the ESTIM software as well as the facilities offered by Estelle* for describing a broadcasting mechanism.

INTRODUCTION

Specific software methodologies are required to cover the life cycle of distributed systems, due to the high complexity of protocols that rule the communications inside such systems. For that purpose, Formal Description Techniques (FDTs) have been introduced to permit isolation of design errors at a very early stage of the software life cycle, namely during the specification phase. Two FDTs, Estelle [ISO 9074] and LOTOS [ISO 8807], have been studied within the SEDOS (Software Environment for the Design of Open distributed Systems) project (November 1984 - October 1988) of the European ESPRIT program [Diaz 89]. These two FDTs, which have recently reached the IS (International Standard) status within ISO, have been chosen due to their technical complementariness (Estelle refers to communicating state machines while LOTOS refers to the CCS process algebra). The objective of the SEDOS project was the development of Estelle and LOTOS, the production of trial specifications, the assessment of verification techniques and the prototyping of a consistent set of tools supporting the different phases of the design of complex OSI protocols and services. Starting from the results of SEDOS, two new ESPRIT projects were elaborated, namely SEDOS/Estelle/Demonstrator (July 1986 - December 1989) and LOTOSHERE (April 1989 - March 1992).

Major contributions achieved at LAAS dealt with the Estelle approach and were mainly devoted to the definition of a formal semantics for Estelle, the proposal of some extensions (essentially a Rendez-Vous mechanism) to Estelle and the prototyping of the ESTIM simulation/verification environment. The SEDOS experience led to identifying a convenient subset of the Estelle language which appears to be particularly suitable for protocol simulation and verification purposes: Estelle* which integrates the Rendez-Vous extension previously mentioned [Courtiat 88]. Estelle* is supported by the ESTIM (Estelle SimulaTor based on an Interpretative Machine) tool environment [Saqui 88] which provides a set of nice capabilities for the simulation and verification of specifications.

This paper presents therefore an experience in using Estelle* and ESTIM for the specification, simulation and verification of a fieldbus protocol. Fieldbus protocols are non OSI protocols intended to provide real-time communication services in a factory environment. The time related aspects of the specification are therefore of primary concern and it will be shown how ESTIM provides capabilities for handling them in a natural way by defining, for the Estelle "delay" clause, a semantics derived from Time Petri nets. The paper uses a protocol validation methodology that combines the original simulation capabilities of ESTIM together with verification capabilities via projection techniques, as implemented by PIPN (Prolog Interpreted Petri Nets) [Azéma 85, 89], a verification tool also developed at LAAS-CNRS. This methodology, applied to the FIP fieldbus protocol, makes it possible to automatically derive the service provided by a protocol from the Estelle* specification of this protocol.

In a first section the paper will give a general overview of Estelle*, with regards to the standardized version of Estelle (ISO IS 9074). The validation methodology implemented by the ESTIM environment will be discussed in the second section. Finally, in a third section, the FIP case study will be presented and some results dealing with the FIP protocol verification will be shown and assessed.

I. THE ESTELLE* LANGUAGE

I.1 - Estelle*: A Dialect of Estelle

An Estelle specification [ISO 9074] describes a hierarchical architecture of nondeterministic sequential components (module instances) exchanging messages (called interactions) through bidirectional links between their ports (called interaction points). Both the hierarchy of modules and the structure of their links may change over time, making thereby the configuration of modules a dynamic one. As the Estelle FDT has already been detailed in several papers [Budkowski 87] [Linn 87] [Tenney 88] [Courtiat 87a], it will not be further discussed here from a tutorial point of view. Only the features related to the Estelle* definition and those required for the FIP case study will be presented in more depth.

Estelle* [Courtiat 88] may be seen as a simplified version of Estelle (only a nondeterministic transition firing rule is considered), on the top of which some new features (essentially a Rendez-Vous mechanism) are introduced in order to obtain OSI protocol specifications with a significant higher level of abstraction. Estelle*, which is the result of our experience in applying Estelle to several OSI protocols, gives the opportunity to bridge the gap between Estelle and models commonly used in the Petri net community (Petri nets, Predicate/Transition nets, Predicate/Action nets, ...) by introducing an explicit merging mechanism between two Estelle* transitions. The main rationales for the features added to (or removed from) Estelle [ISO 9074] have been detailed in previous papers [Courtiat 87c] [Courtiat 89] and will be reminded shortly in the next subsections.

I.2 - Estelle* Nesting Principles

Estelle* assumes a non deterministic firing rule for the transitions of the specification (cf. the case of a specification whose root is attributed by "systemactivity"), and therefore does not consider the "systemprocess" concept defined in the Estelle standard. It has been shown in [Courtiat 88] that the non deterministic firing of the specification transitions does represent an asynchronous parallelism among the module instances which are not in a "ascendant/descendant" relationship, as soon as two sufficient conditions are satisfied by the specification (which may be a "dynamic" one):

① there is no "priority" clause associated with any transition defined with a "when" clause;

② once initialized ("init" statement) and attached ("attach" statement), the module instances are autonomous with respect to their parent module instance; in particular, any decision of "detach" or "release" is local to a module instances and depends on the interactions it receives from the lower layer service (PDUs from the peer entity or interactions generated by the service).

Consequently, the valid firing sequences expressing the global behavior of the specification may directly be obtained from the reachability graph of the specification global states instead of from the reachability graph of the specification global situations [Dembinski 86] [Budkowski 87] which expands much more faster than the former. This approach is further consistent with verification methodologies based on reachability analysis, as for Petri nets [Genrich 79] [Diaz 82] [Courtiat 84] [Juanole 84] and models based on communicating state machines [Danthine 80] [Brand 83], commonly used for protocol modeling and verification. One may further point out that most of the Estelle specifications produced within the ESPRIT/SEDOS project satisfied directly the sufficient conditions stated previously [Diaz 89], e.g. the specifications of the OSI Presentation service, of the CCITT recommendation T.70, of a significant subset of the OSI FTAM protocol (SPAG A.111 Profile) and of the OSI Session protocol.

I.3 - Introducing a Rendez-Vous Mechanism into Estelle

From an external point of view, a module instance is a "black box". Access in and out of that box is made through finite sets of interaction points (possibly through exported variables). Each interaction point of a module instance x has an associated unbounded FIFO queue which receives and stores interactions sent to x through that interaction point. This FIFO queue may be either a dedicated

queue of a module instance interaction point ("individual queue" discipline) or it may be shared by several interaction points of the same module instance ("common queue" discipline).

The FIFO queue mechanism is the basic communication mechanism in Estelle, as it allows module instances, having linked interaction points (either directly connected or indirectly connected through several attachments), to exchange interactions. The fact of representing a service primitive interaction between a service user and a service provider in a non atomic way (i.e. a service primitive is seen as a message appended to a queue) may introduce, in the case of a connection oriented service, some new problems which have to be explicitly dealt with, in a case by case basis, by the Estelle specification. Three types of problems have been identified so far : service primitive collisions, connection endpoint allocation conflicts and difficulties in expressing backpressure flow control [Courtiat 88] [Courtiat 89].

In order to get round these problems, an extension has been proposed in [Courtiat 87b] for Estelle: the so-called Rendez-Vous mechanism, that allows a more abstract representation of the interactions between protocols entities of the adjacent layers in the OSI system. The inclusion of the Rendez-Vous mechanism within the Estelle standard (IS 9074) is currently being investigated within the ISO FDT group under the framework of question 48.5 (JTC1/SC21/WG1). The Rendez-Vous introduces new clauses (called synchronization clauses) which make it possible to express an explicit synchronization with value passing between two transitions of two distinct module instances (the informal semantics corresponds therefore to transition mergings). This mechanism has extensively been described and formalized in [Courtiat 89] and its usefulness demonstrated on several complex OSI protocols (like the Session and Transport protocols).

I.4 - Estelle* "Delay" Clause

In Estelle, the clause "delay(tmin,tmax)" defines the minimum time tmin (respectively the maximum time tmax) during which an enabled transition must (respectively may) be delayed. The "delay" clause is fundamental for the specification of protocol time-outs, hence its semantics should be defined in a complete and formal way.

As stated in paragraph 5.3.5 of the Estelle IS [ISO 9074] "The computational model for Estelle is intentionally formulated in time-independent terms; one of the principal assumptions of this model is that nothing is known about the execution time of a transition in a module instance. Knowledge of execution speeds is considered implementation dependent". A consequence of these statements is that an Estelle specification with transitions containing "delay" clauses is not self contained, as it requires the specification of an external "time process".

A particular semantics of the "delay" clause is considered in this paper, relying on the semantics of Time Petri nets [Merlin 76] [Roux 85]. The starting point of this semantics is the assumption that the firing time of a transition is very short with respect to the time values expressed in the "delay" clause; consequently, this firing time is assumed to be equal to zero. This approach, consistent with the assumptions dealing with time progression in Estelle, makes it possible to update the time evolution at the level of the global model, which is very interesting in order to model, e.g. protocol time-outs; furthermore, the time evolution does not directly depend on the number of transitions which have been fired, nor on the speed of the processors involved in the implementation [Dembinski 87].

II - OVERVIEW OF THE ESTELLE* VALIDATION TOOLSET

The ESTIM tool has been developed in the ML functional Meta-Language, following a rapid prototyping approach [Saqui 89]. ESTIM permits the validation of protocol specified in Estelle*, in the sense that several verification capabilities have been added to the original simulation ones. Those enhancements, in addition to the handling of real-scale specifications, as the present application and many others [Alami 89] [Ayoub 89], were possible thanks to the interfacing of ESTIM with two other SEDOS tools: the Estelle translator [Chari 89] and PIPN [Azéma 85] [Azéma 89]. The Estelle translator parses an Estelle* specification and outputs an intermediate form of this specification which is in turn coded into ML by GENESTIM [Lenotre 87]; the specification ML abstract tree may finally be loaded by the Estelle* interpreter which is the basis of the ESTIM tool. According to the semantics of a

specification attributed by "systemactivity", this interpreter is invoked each time a fireable transition has been selected, either by the user or by the simulator kernel.

Recent enhancements of the ESTIM tool concern the generation of the reachability graph of an Estelle* specification. Analyzing this graph has particularly been investigated with protocol verification in mind, with the specific purpose of automatically deriving the quotient automaton that characterizes the service provided by a protocol, starting from the reachability graph of an Estelle* specification of this protocol; this is the basic idea of the PIPN implemented projection technique. The main steps of the validation methodology are depicted by the scheme below:

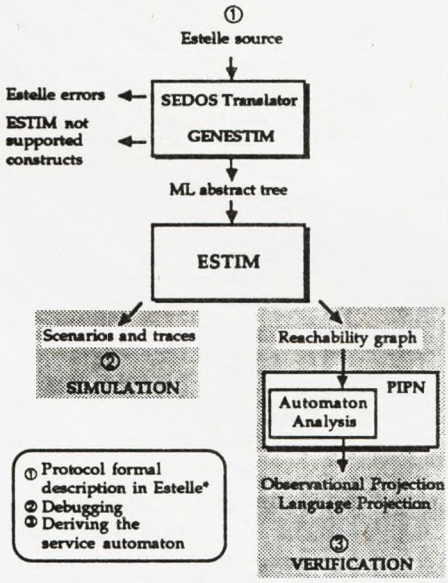


Fig. 1 - The Estelle* validation toolset

Further details on the design and implementation of the Estelle SEDOS tools are available in [Diaz 89]. Next sections address the use of ESTIM as well as its simulation and verification capabilities.

II.1 - The Estelle Simulator of ESPRIT-SEDOS Project

The first step in the simulation process consists in preparing the Estelle source, particularly by writing the PDUs handling Pascal procedures and functions that are often declared as primitive in an Estelle specification. Simulation driving commands (e.g. breakpoints or a Pascal-like "writeln" instruction not available with Estelle) may be inserted in the Estelle source by using "qualifying comments", an Estelle extension which is supported by the SEDOS Translator [Chari 89] [Saqui 89].

Along the simulation, the user is assisted in monitoring the run-time behavior of its specification by providing a symbolic access to all the Estelle objects; in particular, the user will be able to access all the components of the current global state:

- the specification architecture, i.e. the existing module instances, their headers and bodies, the module instance tree and the interaction point configuration (connections and attachments);
- the module instance local marking that may even be directly modified by the user in order to simulate exception conditions (handling of protocol errors for instance);
- the environment and the store of the specification;
- the dynamic firing time intervals associated to the transitions.

A step by step simulation permits analysis of the protocol entity behavior by observing the way local variables are modified after each transition firing as well as interactions newly stored in queues;

therefore, additional test modules should be written (in Estelle) for each module under test and, then, for all the modules at a same hierarchical level. The SEDOS project experience made further appear that state/events tables provided by protocol standards are always implemented by the modules instances which are the leaves of the specification instance tree. Therefore, a down-top approach is considered for module instance debugging. Finally, a non deterministic transition firing permits analysis of the global behavior of the specification; random simulation may be used for detecting potential unspecified receptions as well as deadlock states.

ESTIM simulation facilities were basically intended to debug complex Estelle specifications before implementing them. The new features introduced in Estelle* - particularly the Rendez-Vous mechanism - make it possible to obtain a greater level of confidence in a specification by analyzing the associated reachability graph.

With the assumptions made for Estelle* (see Section I), the reachability graph of an Estelle* specification is the reachability graph of the specification global states. The one transition firing rule implemented in ESTIM handles a nondeterministic selection of transitions that is systematically performed following a transition firing; therefore, it seems natural to try to apply to Estelle* the results of verification techniques successfully experimented in the field of protocol modelling, relying on reachability graph analysis [Genrich 79] [Danthine 80 [Diaz 82] [Courtiat 84] [Yuang 88] [Azéma 89] [Berthomieu 88] [Holzmann 89].

In the general case, the reachability graph of an Estelle* specification is not finite, due to unbounded nature of FIFO queues as well as to Estelle dynamic constructs (e.g. the release of a module instance). Therefore, ESTIM implements verification for static specifications only, while permitting to assign bounded capacities to FIFO queues. It should be stressed that ESTIM does not translate an Estelle* specification into Predicate/Transition Nets; interfacing ESTIM with PIPN was only motivated by the opportunity of using a tool - for reachability graph analysis and reduction purposes - which is available at LAAS.

II.2 - Verification via Projection of Estelle* Descriptions

The protocol validation is considered in this paper as the results of the successive application of simulation and verification techniques on the specified protocol.Two classes of properties are generally considered with protocol verification in mind: *general properties*, e.g. deadlock freeness, which should be satisfied by any protocol, and *specific properties* which depend on the protocol semantics, i.e. the service it is supposed to provide. Specific properties are further classified into *safety properties*, which guarantee partial correctness (the required properties are satisfied as soon as the protocol progress), and, on the other hand, *progress properties* which guarantee that the protocol progresses actually.

Reachability graph generation is a well-proven verification technique for Formal Description Techniques that rely on finite state machine models [Yuang 88]; properties may be checked on this graph, or, at least, on a reduced graph. The intention of reducing a protocol reachability graph is that the resulting quotient automaton may be interpreted as a model of the service provided by the protocol; in other words, the service is the protocol *projection*, as soon as an adequate equivalence relation is found for graph reduction computation. For instance, projections relying on [Kohavi 78]'s language equivalence permits observation of the interaction exchanges at the interface between a protocol entity and its user. For the purpose of distinguishing the protocol internal events [Azéma 89], we also consider graph projections relying on [Milner 80]'s observational equivalence which was originally introduced for checking equivalence among CCS behavior expressions.

In addition to an equivalence relation that preserves "interesting" properties of the protocol, projections rely on a *partition* between the system whose behavior is to be observed and its external world. One major interest of the projection concept is that it makes it possible to observe the inter-module communication inside a specification which is, basically, a "closed world" [ISO 9074]. For a 3-level model of an OSI protocol, the observed events may be the interaction exchanged at the service boundaries [Saqui 89b]. Therefore, the partition determines the way the arcs of the reachability graph are labeled:

- an arc that corresponds to a rendez-vous offer is attributed with a 3-uple: (instance identifier, "no queue" interaction point identifier, interaction identifier)

• otherwise (i.e. the corresponding fired transition has no synchronization clause), the arc is not labeled (λ is the classical notation for a null label)

From a user point of view, the verification methodology discussed so far may be presented as follows:

① Generating the reachability graph, starting from the specification initial state (i.e. the state entered when the static initialization is completed);

② Selecting the events to be observed by defining a partition, i.e. a boundary across the synchronization channels (channels between "no queue" interaction points);

③ Choosing either the observational or the language equivalence for graph reduction purposes;

④ Comparing the reduced graph with the service specification.

Steps ① and ② are handled by ESTIM; projections algorithms (step ③) are computed by PIPN. ESTIM uses a PIPN format file to store the reachability graph of an Estelle* specification, hence this graph may then be reduced ("projected") by PIPN.

III - A CASE STUDY: THE FIP PROTOCOL ARCHITECTURE

A French Working Group - including both industrial and research practitioners - was set up in 1982, in order to investigate opportunities for local-area networks promotion in the field of industrial process control; the major issue of this work is the FIP proposal ("Flux d'Informations issues du Processus", Informations Flow issued from a Process), an example of fieldbus protocol [Galara 85] [FIP 88] [Thomesse 87]. Fieldbus protocols are non ISO protocols, appropriate to cope with real-time constraints that govern interconnections among the basic elements of manufacturing automation environments; they specifically permit communications among e.g. "intelligent" actors, sensors, regulators or programmable automata. The FIP protocol architecture discussed in this paper is further intended to reduce the cabling systems among the basic elements in a factory.

An experimental standard has been published for FIP [FIP 88], relying on the following protocol architecture:

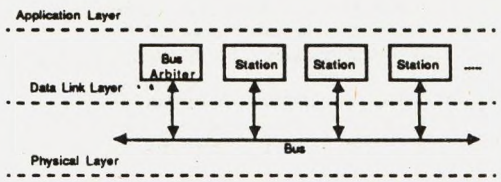


Fig. 2 - Structure of the FIP fieldbus

A station connected to the bus, typically a regulation unit or an input/output device, may be either a producer or a consumer with regards to the informations transmitted on the bus: one information is essential and unique, at a given time : the *identifier* (ID) that characterizes the producer station. A station receiving an ID and further recognizing it as its own access code to the producer status, becomes the one producer on the network. In parallel, one or several consumer stations may be connected to the network. The basic idea of FIP consists in refreshing, either periodically or aperiodically, the memory of consumers; for that purpose, it uses a cyclic schedule, starting with the four steps algorithm below:

a) The arbiter broadcasts an identifier frame on the medium; this frame (ID_DAT) contains the identifier of the variable to be produced;

b) One station (and one only) recognizes ID as its own identifier, hence, as previously mentioned, this station reaches the producer status; meanwhile, other stations - potentially, consumer stations - are aware of the producer identifier;

c) The producer station broadcasts on the medium a response frame (RP_DAT) containing the variable value associated with the identifier sent by the arbiter;

d) A consumer station (cf. step b) seizes the variable value (RP_DAT) which was sent by the producer. RP_DAT is also received by the arbiter, as a mean for checking whether the previously sent ID_DAT had been received correctly.

The cycle ends as soon as a new ID_DAT is broadcast by the arbiter.

III.2 - Describing the FIP architecture in Estelle*

III.2.1 Structuring the specification

Fig.2 defined a layered architecture from which the Estelle* specification structuring was quite straightfordwardly derived. In fact, several Estelle* specifications have been written, relying on the architecture depicted by Fig. 3 below, in order to either describe different services of the FIP standard (periodic vs aperiodic) and/or to adapt the specification towards the requirements of simulation and verification (see also Section IV).

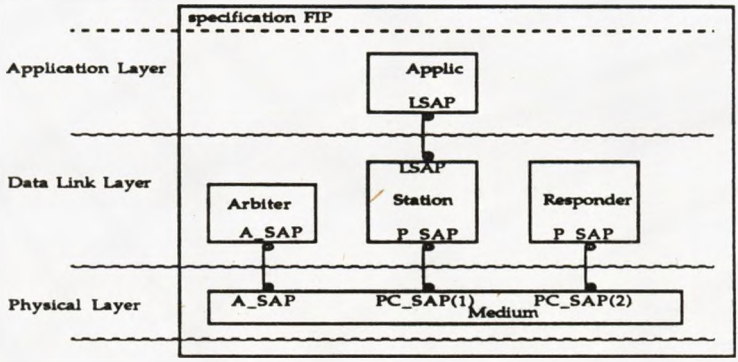


Fig. 3 - The FIP architecture in Estelle* (configuration for verification)

For verification purposes, a static and "minimal" configuration of modules is considered in this paper, including the fundamental elements of the FIP Data Link layer: the arbiter, a producer and a consumer. Frames loses may be represented through the non deterministic selection of transitions in Estelle*, by adding specific transition inside the arbiter and the FIP station.

The main interest of the FIP example lies in the the broadcasting medium.

III.2 Expressing a broadcasting mechanism in Estelle*

Let us now consider a particular scenario for communication on a FIP fieldbus: a new frame should not be broadcasted by the arbiter towards the other stations unless the latters would eventually have received the previous frame. Obviously, this mechanism implies a strong and simultaneous synchronization between, on one side, the arbiter and, on the other side, the producer and consumer stations. Therefore, broadcasting may be elegantly specified by means of a multi-way rendez-vous mechanism, as supported by Petri Nets [Club FIP-LAAS]; unfortunately, only a two-ways rendez-vous is available with Estelle*. Furthermore, the native FIFO queue mechanism of Estelle is basically an asynchronous communication mode, hence nothing can prevent the arbiter to send a new frame before a station received the previous frame.

Nevertheless, broadcasting may be expressed through the Time Petri Net based semantics of the "delay" clause [Courtiat 87c] which is implemented by the ESTIM tool, regardless of whether a FIFO communication or a two-ways rendez-vous is used (between the medium and the other modules). The proposed solution may be characterized as follows (notation : (tmin,tmax)) :

① At the application level, a clause "delay (0,∞)" is assigned to each transition of the Application module concerned by an emission of a service primitive; thus, theses Application transitions are always enabled. Choosing a temporal interval (0,∞) is further consistent with the fact that the Application interactions may be sent in a completely asynchronous way;

② Dealing with the medium, it should be first noted that broadcasting a frame from the arbiter to the stations requires two distinct transitions: the first one receives the interaction sent by the arbiter module and the second one (TB) broadcasts it in turn towards the stations. TB is assigned with a clause "delay (TM,TM)", where TM denotes the propagation time on the medium;

③ Finally, at the FIP station level, the reception transitions are assigned with a clause "delay (0,0)" (in fact, reception transitions in Estelle are mandatorily assigned with a zero-valued "delay" clause).

The Time Petri Net based rule for time progression ensures that, as long as a transition for which tmin=0 remains fireable, no transition for which tmin>0 will be fired. Therefore, the points ①②③ above guarantees that the transition TB, delayed by TM, may not be fired unless all the FIP stations received the frame sent by the arbiter. In fact, there is no chance for TB to be fired as soon as one (or several) transition, delayed by (0,tmax), remains still fireable (tmax is a positive integer value or infinite). One may observe that this approach leads to binding to 1 the length of interaction point queues; finally, using a rendez-vous communication was required at verification stage, as the current version of ESTIM handles verification with respects to the interactions which are exchanged by Rendez-Vous.

IV - VERIFICATION VIA PROJECTIONS OF THE FIP DATA LINK PROTOCOL

The FIP specification architecture introduced by previous section has been simulated and verified within the ESTIM environment presented in Section 2. Starting from a verification methodology already experienced in the field of OSI protocols [Saqui 89a] [Saqui 89b], a partition has been defined over the FIP specification; typically, the partition permits observation of the interactions exchanged at the boundary between the application module and the FIP station.

In the remainder of this subsection, a particular situation is considered for FIP periodic services : variable values are transmitted and, possibly, lost by the medium. Two types of confirmation primitives, respectively the positive and negative ones appear in the automaton below ; in addition, a station may loose ID_DAT while the arbiter may loose RP_DAT.

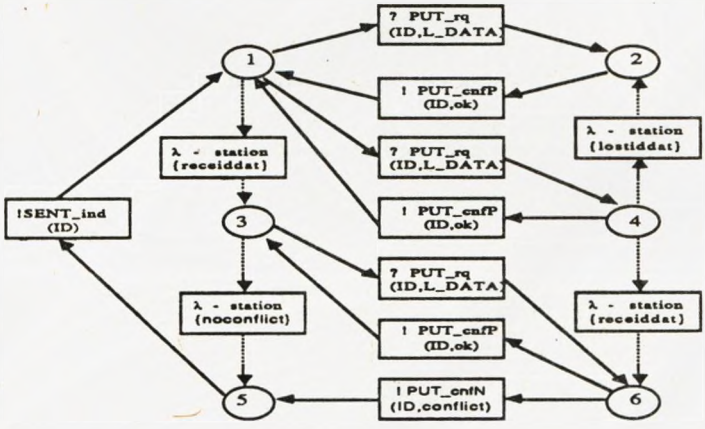


Fig. 4 - Service automaton

The automaton in Fig. 4 considers a particular verification configuration which may be described as follows:

– a station produces variable values associated with the identifier generated by the arbiter. An identifier (ID) associated with an Application primitive PUT_rq is always known and identical to the identifier sent by the arbiter;

– the arbiter specifies the duration for an elementary cycle, by sending a second identifier which is not known by the stations. After sending this identifier, a time-out is triggered; next elementary cycle may then start at expiration of this time-out;

– the responder consumes the variable value associated with the identifier generated by the arbiter.

As for other FIP services that we verified, the service whose previous automaton refers to, satisfies the following properties: there exist no unspecified reception; the protocol model is bounded

(hence it may be implemented) as well as reinitializable (repetitive behavior). The Estelle* specification has been checked for partial correctness properties and deadlock freeness.

CONCLUSION

The Estelle Formal Description Technique has been widely and, until now, quite exclusively applied to protocols developed within of the OSI Reference Model framework. This paper addressed a new application field for Estelle: fieldbus protocols (with a specific emphasis on the FIP standard proposal).

Focus has been laid on a verification methodology which relies on two major issues of the ESPRIT-SEDOS project : Estelle*, a dialect of ISO Estelle, enhanced with a Rendez-Vous mechanism and, on the other hand, ESTIM, a simulation/verification environment which implements the Petri Net based semantics proposed by LAAS for Estelle. One major advantage of this semantics is that it handles a time progression rule which makes possible to express a broadcasting mechanism which is fundamental with respect to the FIP protocol architecture. On the other hand, the verification via projection technique permits comparison of a reduced automaton of the protocol formal specification with its service specifications.

There are now two major issues for this work: first, an implementation of a multi-way rendez-vous into ESTIM is under study. On the other hand, current investigations deal with the specification (in Estelle*) and verification of protocols built upon the FIP service, e.g. the MMS (Manufacturing Message Specification) standard. For this purpose, Estelle would eventually have to be enhanced with ASN.1 (Abstract Syntax Notation 1), an ISO standardized notation [ISO 8824] for description of Application-layer data.

ACKNOWLEDGEMENTS

The authors are thankful to M. DIAZ (SEDOS manager), P. AZEMA, J.-C. LLORET, F. VERNADAT (for PIPN development), B. BERTHOMIEU (for technical support in ML) and J.-F. LENOTRE (for GENESTIM development).

REFERENCES

[Alami 89] R. ALAMI, J.-P. COURTIAT, M. DIAZ, A.-M. DRUILHE, J.-F. LENOTRE, V. MAZZOLA and L. STEFFAN, "Using Estelle for the Design of a Flexible Assembly Cell", French-Brazilian Workshop on Distributed Systems, Florianópolis, Brazil, September 1989 (in French).

[Ayoub 89] M. AYOUB DIT AYADI, "Experience in Using Estelle for the Formal Description of the OSI Presentation Service", Proc. of the IEEE Singapore International Conference on Networks, June 1989.

[Azéma 85] P. AZEMA, G. PAPAPANAGIOTAKIS, "Protocol Analysis by Using Predicate Nets", in Proc. of the 5th IFIP Workshop on Protocol Specification, Testing and Verification, Moissac-Toulouse, June 1985.

[Azéma 89] P. AZÉMA, F. VERNADAT, J. C. LLORET, "Requirements Analysis for Communication Protocols", Workshop Méthodes de Vérification Automatique, Juin 1989.

[Berthomieu 88] B. BERTHOMIEU, M. DIAZ, "Specification and Verification of Time Dependent Distributed Systems Using Time Petri Nets", submitted for publication to IEEE Transactions on Software Engineering, July 1988.

[Brand 83] D. BRAND, P. ZAFIROPOULO, "On communicating finite-state machines", Journal of the ACM, April 1983.

[Budkowski 87] S. BUDKOWSKI, P. DEMBINSKI, "An introduction to Estelle: a specification language for distributed systems", SEDOS report 104, September 1987.

[Chari 89] V. CHARI, J-F. LENOTRE, E. MARIANI, "The Estelle Translator", in [Diaz 89].

[Club FIP-LAAS] "Couche Liaison de Données - Modèle en Réseaux de Petri Labellés à Prédicats", Dossier Technique, 1988.

[Courtiat 84] J-P. COURTIAT, J-M. AYACHE, B. ALGAYRES, "Petri nets are good for protocols", in Computer Communication Review, VOL.14, N°2, 1984.

[Courtiat 87a] J. P. COURTIAT, P. DEMBINSKI, R. GROZ, C. JARD, "ESTELLE: un langage pour les algorithmes distribués et les protocoles", revue TSI, Mai 1987.

[Courtiat 87b] J. P. COURTIAT, "Contribution à la Description Formelle de Protocoles", Thèse de Doctorat d'Etat, Université Paul Sabatier, Toulouse, Décembre 1987.

[Courtiat 87c] J. P. COURTIAT, "A Petri net based semantics for Estelle", IFIP Workshop on Specification, Verification and Testing for Estelle", Zurich, May 1987, North-Holland, C. West - H. Rudin Editors, 1987.

[Courtiat 88] J. P. COURTIAT, "Estelle*: A powerful dialect of Estelle for OSI Protocol Descriptions", 8th IFIP International Symposium on Protocol Specification, Verification and Testing, Atlantic City, June 1988.

[Courtiat 89] J. P. COURTIAT, "Introducing a Rendez-Vous Mechanism in Estelle: Estelle*, in [Diaz 89].

[Danthine 80] A. DANTHINE, "Protocol representation with finite-state models", IEEE Transactions on Communications, Vol. COM-28, N°. 4, pp. 632-643, April 1980.

[Dembinski 86] P. DEMBINSKI, "Estelle semantics", Report ESPRIT-SEDOS/B1/054, June 1986.

[Dembinski 87] P. DEMBINSKI, S. BUDKOWSKI, "Simulating Estelle specifications with time para-meters", Proceedings of the 7th International Workshop on Protocol Specification, Testing and Verification, Zurich, May 1987, H. RUDIN & C.H. WEST (ed).

[Diaz 82] M. DIAZ, "Modelling and Analysis of Communication and Cooperation Protocols using Petri Nets based Models", Tutorial paper, Computer Networks, December 1982.

[Diaz 89] M. DIAZ, J. P. ANSART, J. P. COURTIAT, P. AZEMA, V. CHARI, "The Formal Description Technique Estelle - Results of the SEDOS Project", Ed. M. Diaz & al.,North Holland, 1989.

[FIP 88] FIP, Spécification des Services "Couche Application", DRAFT 3 - Norme Expérimentale, N° 88-0004, CGEE ALSTHOM, 01/02/1988 — FIP - "Couche de Liaison de Données" - document C 46-603, Mai 1988 — Document C 46-601, "Architecture et Présentation Générale" — Document C 46-604, "Couche Physique" — Document C 46-605, "Gestion du Réseau".

[Galara 85] D. GALARA, J-P. THOMESSE, "Groupe de Refléxion FIP", Réseaux Locaux Industriels, MINISTERE DE L'INDUSTRIE ET DE LA RECHERCHE.

[Genrich 79] H.J GENRICH, K. LAUTENBACH, "The Analysis of Distributed Systems by means of Predicate/ Transition Nets", Lecture Notes on Computer Science - Springer Verlag, 1979.

[Holzmann 89] G.J. HOLZMANN, J. PATTI, "Validating SDL Specifications: an Experiment", Proceedings of the 9th IFIP International Workshop on Protocol Specification, Testing and Verification, Eschendende (The Netherlands), June 1989.

[ISO 8807] "LOTOS, A formal Description Technique Based on the Temporal Ordering of Observational Behaviour", ISO IS 8807, November 1988.

[ISO 8824] Specification of Abstract Syntax Notation One (ASN.1), ISO IS 8824, August 1987.

[ISO 9074] "Estelle - A formal description technique based on an extended state transition model",IS 9074, November 1988.

[Juanole 84] G. JUANOLE, B. ALGAYRES, J. DUFAU, "On Communications Protocol Modelling and Design", Lectures Notes in Computer Science, Springer Verlag, 188, Advances in Petri Nets 1984, ed. G. ROZENBERG.

[Kohavi 78] Z. KOHAVI, "Switching and Finite Automata Theory", Mac Graw-Hill, 1978.

[Lenotre 87] J-F. LENOTRE, "The Estelle to ML Generator Reference Document", SEDOS report 116, October 1987.

[Linn 87] R. J. LINN, "Tutorial on the Features and Facilities of Estelle", ICST report, National Bureau of Standards, 1987.

[Merlin 76] P.M. MERLIN, D.J. FABER, "Recoverability of communication protocols, Implication of a theoretical approach", IEEE Transactions on Communications, September 1976.

[Milner 80] R. Milner, "A Calculus of Communicating Systems", Lectures notes in Computer Science, Vol. 64, 1980.

[Roux 85] J.L. Roux, "Modélisation et analyse des Systèmes Distribués par les Réseaux de Petri Temporels", Thèse de Docteur-Ingénieur INSAT, Toulouse, 1985.

[Saqui 88] P. de SAQUI-SANNES, J. P. COURTIAT, "ESTIM: The Estelle Simulator Prototype of the ESPRIT/SEDOS project", in "Proceedings of the First International Conference on Formal Description Techniques (FORTE 88), North-Holland, 1988, K. J. Turner (ed), pp. 15-29.

[Saqui 89a] P. de SAQUI-SANNES, J. P. COURTIAT, "ESTIM: Simulating ESTELLE Descriptions of OSI Protocols, Proc. of the IEEE Singapore International Conference on Networks' 89, 17 - 20th July 89.

[Saqui 89b] P. de SAQUI-SANNES, J.-P. COURTIAT, "From the Simulation to the Verification of Estelle Specifications", in Proceedings of the 2nd International Conference on Formal Description Techniques (FORTE 89), Vancouver (CANADA), S.T. VUONG (ed), North-Holland.

[Tenney 88] R. L. TENNEY, "A Tutorial Introduction to Estelle", Invited paper at the 1st International Conference on Formal Description Techniques, Stirling, September 1988.

[Thomesse 87] J-P. THOMESSE, M. GAULT, J-P. LERARE, M. MESBAH, "FIP: une proposition de standard de bus de terrain, RGE, N° 11, Décembre 1987.

[Yuang 88] M.C. YUANG, "Survey of Protocol Verification Techniques Based on Finite State Models", Proc. of the IEEE Computer Networking Symposium, Washington, April 1988.

SESSION 5

PROTOCOL IMPLEMENTATION AND TESTING

EXPERIENCE WITH THE ASTRIDE TESTING APPROACH

Omar RAFIQ* and Richard CASTANET**

(*) Laboratoire TASC
Département d'Informatique
Université de Pau
Avenue de l'Université
64000 Pau - France

(**) Laboratoire LABRI
ENSERB
Université de Bordeaux I
351, Cours de la Libération
33405 Talence - France

In COMNET'85, we have introduced some elements of a new testing approach for conformance testing of OSI protocol implementations : the astride testing approach. In this paper, this approach is described by using the recent ISO conformance testing terminology. Its benefits, shortcomings and cousins are discussed. Dedicated test tools based on SDL of CCITT, are presented. Finally, an example of test realization taken from the common ISO transport, is developed.

1. INTRODUCTION

To ensure that OSI products conform to the relevant standards, testing procedures comprising methods and tools are essential. ISO has designed a multi-part standard [ISO9646] specifying a general methodology for OSI conformance testing. One can find a survey of this work in [Rayner 87]. Four abstracts methods are proposed : the local test method and three external methods which are the remote test method, the distributed test method and the coordinated test method. In this paper, we deal only with single layer test.

In the local test method, all test software is placed in the system under test (SUT). It is made up of three parts : an upper tester (UT), a lower tester (LT) and test coordination procedures (TCP). The (N) service and (N-1) service boundaries of the (N) implementation under test (IUT) also called points of control and observation (PCOs) are directly observed and used by the test software. The UT interacts with the (N) service interface by sending/receiving (N) service primitives also called (N) ASPs (abstract service primitives) and the LT interacts with the (N-1) service interface by sending/receiving (N-1) ASPs including (N) PDUs. The TCP govern the cooperation between the LT and the UT during testing. Due to the fact that the test software has full control and observation of all the external access points of the IUT, the local test method is the most powerful [Dssouli 85]. However, it is rarely considered in practice because conformance testing is generally the responsability of institutions undertaking the activity from centralized site. Nevertheless, recently, [Zeng 89] has introduced the ferry clip : an approach based on the ferry principle [Zeng 88] and using remotely the local test method.

In the external test methods, the LT is separate from the SUT and communicates with the (N) IUT via an appropriate (N-1) layer service provider. This (N-1) service is assumed to

function correctly. External test methods vary according to the level of requirement or standardization put on the TCP, on the access to the layer boundary above the IUT, and on the requirements on an UT. Among the three external methods proposed by ISO, the distributed test method is the only one requiring access to the upper boundary of the IUT and it is very applied in real testing and more particularly in development testing. However, it is very difficult to guarantee synchronization between the distributed LT and UT [Sarikaya 84, Linn 84]. To get arround this problem, we have designed the astride testing approach [Rafiq 85a, Rafiq 85b] with which the UT is moved in the same system than LT : the testing system (TS), and replaced by the astride test responder above the IUT in the SUT. This technique simplifies considerably the realization of the distributed test method.

In recent papers [Chan 89, Chanson 89, Mackert 88, Zeng 88, Zeng 89, etc..], various approaches, tools and experiences of protocol testing are reported. In this paper, the astride testing approach is described by using the recent ISO conformance testing terminology [ISO9646]. Its benefits, shortcomings and cousins are discussed. Dedicated test tools based on SDL of CCITT, are presented. Finally, an example of test realization taken from the common ISO transport [ISO8072, ISO8073], is developed.

2. THE ASTRIDE APPROACH

Staircase Implementation Model

Schematically, the staircase implementation model of OSI layers, consists in implementing each (N) layer above a part of the (N-1) layer boundary and leaving the other part accessible to the user. Figure 1 shows a synoptic scheme of this arrangement. No information flows through any intermediate level, i.e. only the visible interfaces can be invoked by the user.

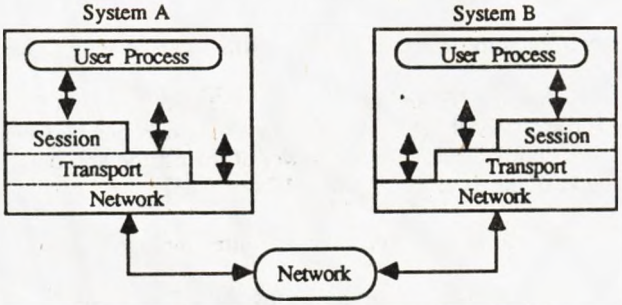


Figure 1 - **The staircase Implementation Model**

A significant advantage of staircase implementation model is generality : a user can communicate and interwork with other systems not only through the highest implemented level but through any of the existing levels. Earlier [Rafiq 85a], this model has been adopted by Bull within the SPS7 and Questar400 systems. Furthermore, users are not the only group that an OSI implementation should accomodate, it should also enable convenient and incremental installation and testing of communication layers. Access to service boundaries provided by staircase implementation was very helpful for designing the astride testing approach.

Testing Architecture

The astride testing architecture shown in figure 2, is compatible with the ISO distributed test method. The (N) IUT is implemented over the (N-1) service boundary in its working environment, according to the staircase implementation model. The UT moved in the TS, communicates with the astride responder (AR) via an extra connection called the testing connection (TC). The role of AR is to apply whatever is sent by the UT to the IUT and to communicate the indications of the IUT to the UT. Therefore, the control protocol between the UT and the AR is elementary.

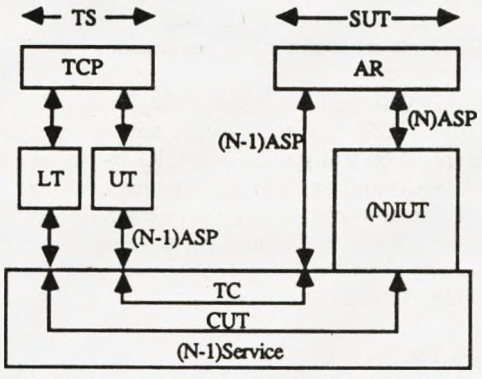


Figure 2 - The Astride Testing Architecture

The term astride comes from the fact that the designed responder (AR) has to be implemented upon two communication levels : the (N) level under test and a lower level ((N-1) in figure 2). The LT communicates with the (N) IUT via a connection under test (CUT). The TC is established by the UT at the beginning of testing.

Benefits and Shortcomings

The astride approach has some interesting benefits which can be summarized as follows :

(1) Due to the facts that, the UT and the LT are in the same system (TS), the connection between the UT and the AR (TC) is reliable and the connection between the LT and the IUT (CUT) is also reliable, the testing power of the astride approach is very close to the testing power of the local test approach [Dssouli 85].

(2) To carry out testing, only one connection is required through the IUT.

(3) Tests can be launched either from the TS by the LT or from the SUT by the UT. Therefore, the capabilities of the IUT to be initiator or acceptor of communication functions can be tested.

(4) According to the role of the AR, it is small and easy to implement. Therefore, all SUT configurations small and large systems, are covered by the AR.

It has also two peculiar shortcomings :

(1) Exchanges between the AR and the UT add extra time delay to the real communication. Fortunately, it is not a serious problem in practice.

(2) It requires two exposed interfaces within the SUT.

Cousins of the Astride Approach

The astride approach has two cousins in the literature : the loop-back ferry approach and the ferry clip distributed approach. The loop-back ferry approach introduced in 1985 [Zeng 85] and consolidated in 1988 [Zeng 88] is exactly the astride approach with the TC passing through the IUT. Unfortunately, this configuration is not efficient because the IUT is a part of the test software. However, it can be tolerated, only if there is no access to a lower service boundary allowing the TC to bypass the IUT.

The other cousin is also due to Zeng [Zeng 89]. It is an application of the ferry clip approach. The latter is a technique for carrying out remotely the local test method, by using two testing entities : an active ferry clip (AFC) in the TS and a passive ferry clip (PFC) in the SUT, communicating through a transfer service independent of the (N)IUT and called the ferry transfer service (FTS). Figure 3 shows a synoptic scheme of this approach. However, if the (N) IUT is integrated over the (N-1) service, access to the lower (N) IUT service boundary can be achieved only indirectly through the (N-1) service as indicated in the figure 4. The ferry clip approach becomes the ferry clip distributed approach which is exactly the astride approach. This is another way to show that the configuration of the distributed test method which is the most close to the local test method is the astride testing approach.

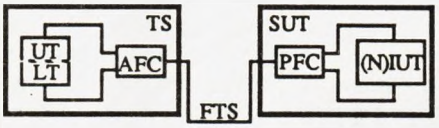


Figure 3 - **The Ferry Clip Approach**

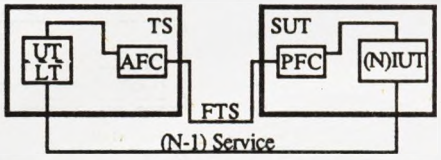


Figure 4 - **The Ferry Clip Distributed Apprpoach**

3. TEST TOOLS

To carry out the astride testing approach, whereas the implementation of the AR does not pose any peculiar problem, the realization of the test system tools is not trivial. Furthermore, both interactive and automatic testing have to be supported. Interactive testing is generally necessary to test whether or not the IUT performs the elementary and basic functions. After that, automatic testing is required for a more realistic communication. This section describes the main elements of the test system tools integrating interactive and automatic modes, that we have designed and realized for the French manufacturer Bull. Figure 5 shows the architecture of this system.

The interactive mode is provided through a set of commands allowing the user to construct and to send messages on TC and CUT as well as to set and to reset timers for waiting for responses. This set of commands, is in fact a subset of LTP (language for testing protocols), the language we have designed to carry out the automatic mode [Chraïbi 86]. The aim of LTP is to combine a protocol specification language, SDL (specification and description language) [Belina 89] of CCITT, to describe the test skeleton, with the C programming language to handle all objects related to the test realization. Now, there is a technique : TTCN (tree and tabular combined notation), designed by the ISO particularly for describing OSI conformance testing [ISOTest3 89]. Fortunately, there is no problem to translate TTCN descriptions into LTP programs. Other languages have been used in other systems [Mackert 88].

SDL

SDL is based on extended finite state models (EFSM) : FSM [Danthine 80] with additional context variables. FSM are defined in terms of states (waiting for message), transitions (execution of treatment), inputs (incoming message) and outputs (outgoing message). In SDL, a system is defined by a set of blocks which are connected to each other and to the environment by channels. The behavior of each block is modelled by one or several processes communicating with one another by means of signals (messages) through channels. Each process is described by an EFSM using the following keywords : state (STATE, ENDSTATE), input (INPUT), task (TASK), output (OUTPUT) and transition (NEXTSTATE).

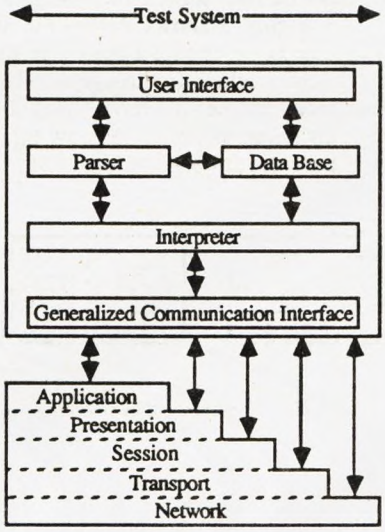


Figure 5 - **Test System Architecture**

Software Environment

All test system tools are written in C and developed on Bull SPS7 system which provides a real-time operating system called SPART and the SPIX system compatible with UNIX systemV. SPART designed by Bull-Sems, allows the carrying out of real-time tasks which are necessary for the test, for instance : inspecting the status of CUT and TC at regular intervals, instantaneous reaction (response) to the reception of message from CUT and TC, setting timer to fixe the time delay to wait for a message from the IUT, resetting timer, and so on. SPART is in fact coupled with SPIX. Therefore, programs dedicated to SPART are written and debugged under SPIX and afterwards run under SPART. This approach is choosen in order to allow programmers to use the UNIX facilities for developing real-time software.

User Interface

The user interface takes as input either commands to carry out interactive testing or an LTP program for automatic testing. It performs text-creation and modification functions of an ordinary text editor.

Parser

The syntax of LTP is described by a contex-free grammar with the LALR (1) property, in Backus-Naür Form [Aho 87]. The related parser is built up by using LEX and YACC tools [Aho 87]. It performs the lexical and syntax analysis, as well as semantical checking peculiar to SDL (state declaration, incompatibility between the declaration and use of signals, for instance). By associating actions with the grammar rules, i.e. C routines to be invoked when the rules are used, a file of quadruples representing the test skeleton (EFSM) is produced. This quadruples list is the actual input of the interpreter [Chraïbi 86].

Data Base

The data base is a library containing the identifiers, formats, types and codes of the ISO PDUs and ASPs parameters required for the test. This data base is filled and up dated through the user interface. It is used by the parser for checking the structures of PDUs and ASPs described in the LTP programs, and by coding/decoding functions of the interpreter.

Interpreter

The role of the interpreter is to control and to manage the testing carrying out. Using the quadruples list and the data base (library of PDU and ASP structures), it executes the test specified by the user and performs the interactions between the TS modules and the operating system. The test progress is performed essentially by functions calls. These functions are either related to the actions of transitions or routines connected to the testing environment. We can distinguish three main types of function :

- the context functions which manage the global context of the TS (initialization, buffer allocation, etc.) and the TC (LT) and CUT (UT) contexts (addresses, counters, etc.);
- the processing functions (test control,sending and receiving of messages,timer handling,etc.);
- the coding functions (to built the (N)PDUs and the (N)ASPs to be packed into the (N-1)ASPs) and the decoding functions (to recognize the (N)PDUs and the (N)ASPs extracted from the (N-1)ASPs).

Generalized Communication Interface

The staircase implementation model provides access to different communication levels. In order to make access to these levels uniform, a generalized communication interface (GCI) has been designed. This interface is similarly invoked for using any of the available communication services. Pratically, the implementation of the GCI leans on the communication system provided on the Bull SPS7 system. According to this one, the OSI services are used by means of a global access method based on a message exchange mechanism. The latter allows the user to access to the different communication services through one primitive for sending user requests and another one for receiving service responses.

4. TEST REALIZATION

Testing an IUT involves applying one or more test cases, a complete set of actions required to achieve a specific test purpose [Sarikaya 89]. According to [ISO9646], conformance testing is performed by executing a test suite which is defined as a set of test groups together with an ordering information. Test groups are a set of related test cases such as connection establishment, normal data transfer, connection release, and so on. A test case

is made up of a number of test events, an indivisible unit of test such as sending or receiving a single PDU.

We give below an example of test group written in LTP and dealing with the connection establishment, the normal data transfer and the connection release of the ISO transport (class2) [ISO8072, ISO8073]. This example is a simplified one and gives just an outline of the declaration part, of the processing part and of message parameters.

The transport service primitives used are : TCONreq, TCONcnf, TDTreq and TDISind, and the TPDUs used are : CR, CC, DT, AK, DR and DC. Briefly, the test group consists in establishing a transport connection initiated by the UT. TDTreq is then sent by the UT and received by the LT in form of successive DT TPDUs : DT0, DT1 and DT2; the initial credit is equal to two TPDUs. After that, the connection is released by the LT. During the test, if a time-out occurs or an unexpected message (OTHERS) is received, a related comment (mess) is printed and the test is stopped.

```
SYSTEM Transport;
CHANNEL FROM LT WITH CC, AK, DR;
CHANNEL TO LT WITH CR, DT, DC;
CHANNEL FROM UT WITH TCONreq, TDTreq;
CHANNEL TO UT WITH TCONcnf, TDISind;
BLOCK Class 2;
SIGNAL TCONreq ( INTEGER nousr, notrp, credit, class, altclass;
                 BOOLEAN expdata; CHARSTRING ndata, tsaped, tsaping);
SIGNAL TCONcnf ( INTEGER nousr, notrp, credit, class, altclass;
                 BOOLEAN expdata; CHARSTRING ndata, tsaped, tsaping);
SIGNAL TDTreq  ( INTEGER nousr, notrp; CHARSTRING usr_data);
SIGNAL TDISind ( INTEGER nousr, notrp, reason; CHARSTRING usr_data);
SIGNAL CR ( INTEGER credit, destref, sourceref, class, opt, tpdusize; CHARSTRING tsaping, tsaped;
            INTEGER adopt, altclass; CHARSTRING user_data);
SIGNAL CC ( INTEGER credit, destref, sourceref, class, opt, tpdusize; CHARSTRING tsaping, tsaped;
            INTEGER adopt, altclass; CHARSTRING user_data);
SIGNAL DT ( INTEGER destref, tpdu_nr, eot; CHARSTRING user_data);
SIGNAL AK ( INTEGER credit, destref, yr_tu_nr);
SIGNAL DR ( INTEGER destref, sourceref, reason; CHARSTRING user_data);
SIGNAL DC ( INTEGER destref, sourceref);
PROCESS Test_Group;
DECLARATION_PART;
CONST
  time=10;
  mess1  = "network connection refused";
  mess2  = "network connection breakdown";
  mess3  = "receiving OTHERS when waiting for CR";
  mess4  = "timeout when waiting for CR";
  mess5  = "receiving OTHERS when waiting for TCONcnf";
  mess6  = "timeout when waiting for TCONcnf";
  mess7  = " all DTs have been correctly received ";
  mess8  = "DT received not valid";
  mess9  = "timeout when waiting for DT";
  mess10 = "receiving OTHERS when waiting for DC";
  mess11 = "timeout when waiting for DC";
  mess12 = "end of test";
  mess13 = "receiving OTHERS when waiting for TDISind";
  mess14 = "timeout when waiting for TDISind";
TYPE STATES = ( Closed, LT_Waiting_for_CR, UT_Waiting_for_TCONcnf, Open,
                LT_Waiting_for_DC, UT_Waiting_for_TDISind);
VARIABLES  TIMER t1, t2, t3, t4, t5;
           INTEGER n;
```

```
FUNCTION openNCUT (INTEGER vcid, refcalled, refcalling);
FUNCTION DT_ok( );
PROCESSING_PART;
START Closed;
    TASK        CALL n = openNCUT (300, 9006, 9005);
    DECISION (n);
    CASE (0) :  TASK CALL print (mess1);
                STOP;
    CASE (1) :  OUTPUT UT.TCONreq (  nousr=01, notrp=00, credit=2, class=2, altclass=2,
                                     expdata=1, tsaped=TSTID, tsaping=TSTID);
                TASK SET (time, t1);
                NEXTSTATE LT_Waiting_for_CR;
    ELSE :      TASK CALL print (mess2);
                STOP;
    ENDDECISION;
    STOP;
ENDSTATE Closed;
STATE LT_Waiting_for_CR;
  INPUT LT.CR;
    TASK            RESET (t1);
                    CC.destref=CR.destref;
    OUTPUT          LT.CC (  credit=2, destref, sourceref=10, class=2, opt=0, tpdusize=08,
                             tsaped=TSTID, tsaping=TSTID, adopt=01);
    TASK            SET (time, t2);
    NEXTSTATE       UT_Waiting_for_TCONcnf;
  INPUT OTHERS;
    TASK            RESET (t1);
                    CALL print (mess3);
    STOP;
  INPUT Time_out;
    TASK            CALL print (mess4);
    STOP;
ENDSTATE LT_Waiting_for_CR;
STATE UT_Waiting_for_TCONcnf;
  INPUT UT.TCONcnf;
    TASK            RESET (t2);
                    TDTreq.notrp=TCONcnf.notrp;
    OUTPUT          UT.TDTreq (nousr=01, notrp, usr_data);
    TASK            SET (time,t3);
    NEXTSTATE       Open;
  INPUT OTHERS;
    TASK            RESET (t2);
                    CALL print (mess5);
    STOP;
  INPUT Time_out;
    TASK            CALL print (mess6);
    STOP;
ENDSTATE UT_Waiting_for_TCONcnf;
STATE Open;
  INPUT LT.DT;
    TASK            RESET (t3);
                    CALL n=DT_ok ( );
    DECISION (n);
    CASE (0) :  TASK    AK.yr_tu_nr=AK.yr_tu_nr+1;
                        SET (time, t3);
                NEXTSTATE Open;
    CASE (1) :  TASK    AK.yr_tu_nr=AK.yr_tu_nr+1;
                        AK.destref=CC.destref;
                OUTPUT LT.AK (credit=1, destref, yr_tu_nr);
                TASK    SET (time,t3);
                NEXTSTATE Open;
    CASE (2) :  TASK    CALL print (mess 7);
```

```
                    DR.destref=CC.destref;
                OUTPUT LT.DR (destref, sourceref=10, reason=0);
                TASK SET (time,t4);
                NEXTSTATE LT_Waiting_for_DC;
    ELSE :      TASK    CALL print (mess8);
                    DR.destref=CC.destref;
                OUTPUT LT.DR (destref, sourceref=10, reason=0);
                TASK    SET (time,t4);
                NEXTSTATE LT_Waiting_for_DC;
    ENDDECISION;
    STOP;
  INPUT Time-out;
    TASK        CALL print (mess9);
    STOP;
ENDSTATE Open;
STATE LT_Waiting_for_DC;
  INPUT LT.DC;
    TASK        RESET (t4);
                SET (time, t5);
    NEXTSTATE UT_Waiting_for_TDISind;
  INPUT OTHERS;
    TASK        RESET (t4);
                CALL print (mess10);
    STOP;
  INPUT Time_out;
    TASK        CALL print (mess11);
    STOP;
ENDSTATE LT_Waiting_for_DC;
STATE UT_Waiting_for_TDISind;
  INPUT UT.TDISind;
    TASK        RESET (t5);
                CALL print (mess12);
    STOP;
  INPUT OTHERS;
    TASK        RESET (t5);
                CALL print (mess13);
    STOP;
  INPUT Time_out;
    TASK        CALL print (mess14);
    STOP;
ENDSTATE UT_Waiting_for_TDISind;
ENDPROCESS Test_Group;
ENDBLOCK Class2;
ENDSYSTEM Transport;
```

5. CONCLUSION

The astride testing approach for realizing the ISO distributed test method has been presented. It simplifies considerably the synchronization problems encountered in the distributed test and it is close to the most powerful approach : the local test method. Dedicated TS tools have been designed and developed. This TS is used to carry out conformance testing of ISO transport and session implementations. In waiting for standardized tests, those are constructed manually from specifications, by using empirical principles based on our knowledge of OSI protocols and services. Our experience with the astride approach shows that it is easy to realize and to carry out. Finally, It is worth noticing that it has been chosen by Telenet GmbH (West Germany), for testing a videotex interworking protocol [Zeng 88] and by the Institute of Informatics and Computing Technique (DDR), for testing OSI protocol implementations of the research network DELTA [Heymer 88].

6. REFERENCES

[Aho 87] A. V. Aho, R. Sethi and J. D. Ullman, " Compilers, Principles, Techniques, and Tools ", Addison-Wesley, 1987.

[Belina 89] F. Belina and D. Hogrefe, " The CCITT-Specification and Description Language SDL ", Computer Networks and ISDN Systems, Vol.16(4), 1989, pp. 311-341.

[Chan 89] R. I. Chan, B. R. Smith, G. Neufeld, S. T. Chanson, W. B. Davis, S. T. Vuong, H. L. See and S.Chan, " A Software Environment for OSI Protocol Testing Systems ", Protocol Specification, Testing and Verification, IX, E. Brinksma, G. Scollo and C. Vissers (Editors), North-Holland, 1989.

[Chanson 89] S. T. Chanson, B. P. Lee, N. J. Parakh and H. X. Zeng, " Design and Implementation of a Ferry Clip Test System ", Protocol Specification, Testing and Verification, IX, E. Brinksma, G. Scollo and C. Vissers (Editors), North-Holland, 1989.

[Chraïbi 86] C. Chraïbi, O. Rafiq and R. Castanet, " SDL as a Testing Language ", Proceedings of the Eighth International Conference on Computer Communication (ICCC'86), Munich, West Germany, September 15-19, 1986, P.J. Kuehn (Editor), North-Holland, pp.502-507.

[Danthine 80] A. Danthine, "Protocol Representation with Finite State Models", IEEE Transactions on Communications, Vol.28(4), 1980, pp.632-643.

[Dssouli 85] R. Dssouli and G.v. Bochmann, " Error Detection with Multiple Observers ", Protocol Specification, Testing and Verification, V, M. Diaz (Editor), North-Holland, 1985, pp.483-494.

[Heymer 88] V. Heymer, " Conformance and Interoperability Testing of Computer Communication Protocols ", Proceedings of the IFIP Open Symposium on Network Information Processing Systems, Sofia, Bulgaria, May 9-13, 1988, K.Boyanov and R.Angelinov (Editors), North-Holland, pp.41-47.

[ISO8072] ISO, " Transport Service Definition ", IS 8072.

[ISO8073] ISO, " Connection Oriented Transport Protocol Specification ", IS 8073.

[ISO9646] ISO, " OSI Conformance Testing Methodology and Framework - Part 1 : General Concepts, Part 2 : Abstract Test Suite Specification, Part 3 : The Tree and Tabular Combined Notation (TTCN), Part 4 : Test Realization ", Part 5 : Requirements on Test Laboratories and Clients for the Conformance Assessment Process ", ISO 9646.

[Linn 84] R. J. Linn, " An Evaluation of the ICST Test Architecture after Testing Class4 Transport ", Protocol Specification, Testing and Verification, IV, Y. Yemini, R. Strom, and S. Yemini (Editors), North-Holland, 1984, pp.611-621.

[Mackert 88] L. Mackert, R. Meyer, U. Scheere, J. Schneider,R.J. Velthuys,J. Burmeister,J. de Meer and I.Schröer, " A Generalized Conformance Test Tool For Communication Protocols ", Proc. of the Eighth Inter. Conf. on Distributed Computing Systems, San Jose, California, June 13-17, 1988, pp.180-188.

[Rafiq 85a] O. Rafiq, R. Castanet, C. Chraïbi, J. P. Goursaud, J. Haddad and X. Perdu, " Towards an Environment for Testing OSI Protocols ", Protocol Specification, Testing and Verification, V, M. Diaz (Editor), North-Holland, 1985, pp.533-544.

[Rafiq 85b] O. Rafiq and J. Haddad, " Description of Protocol Testing Scenarios ", Proceedings of the IFIP TC6 Working Conference COMNET'85, Computer Network Usage : Recent Experiences, Budapest, Hungary, October 1-4, 1985, I. Csaba, K. Tarnay and T. Szentivanyi (Editors), North-Holland, pp.623-644.

[Rayner 87] D. Rayner, " OSI Conformance Testing ", Computer Networks and ISDN Systems, Vol.14(1), 1987, pp.79-98.

[Sarikaya 84] B. Sarikaya and G.v. Bochmann, " Synchronization and Specification Issues in Protocol Testing ", IEEE Transactions on Communications, Vol.32(4), 1984, pp.389-395.

[Sarikaya 89] B. Sarikaya, " Conformance Testing : Architectures and Test Sequences ", Computer Networks and ISDN Systems, Vol. 17(2), 1989, pp.111-126.

[Zeng 85] H. X. Zeng and D. Rayner, "The Impact of the Ferry Concept on Protocol Testing", Protocol Specification, Testing and Verification, V, M. Diaz (Editor), North-Holland, 1985, pp.519-531.

[Zeng 88] H. X. Zeng, Q. Li, X.F. Du and C.S. He, " New Advances in Ferry Testing Approaches ", Computer Networks and ISDN Systems, Vol. 15 (1), 1988, pp.47-54.

[Zeng 89] H. X. Zeng, S. T. Chanson and B. R. Smith, " On Ferry Clip Approaches in Protocol Testing", Computer Networks and ISDN Systems, Vol.17 (2), 1989, pp.77-88.

EXPERIENCE IN CONFORMANCE TESTING WITH THE TEST SYSTEM TEKOS

Uwe BUHRANDT, Volker HEYMER,
Jochen HINKELMANN, Peter TOLLKUHN

Academy of Sciences of the GDR
Institute of Informatics and Computing Technique
GDR, Berlin

TEKOS is a conformance test system which implements the external distributed test method in the form of the ferry concept and is dedicated to the test of upper layer OSI software. The system is embedded in a tool environment for the preparatory as well as the evaluation phase of the test process. A main design goal was the minimization of the dependencies from the implementation under test. Experience in using TEKOS together with the tools and directions of further work are discussed.

1. INTRODUCTION

To support the interworking of real open systems developed at different sites tests are necessary with specific test goals. They can be classified in

- software tests,
- conformance tests,
- interoperability tests,
- performance tests,
- robustness tests.

With the software test the program modules have to be tested as stand alone software. After this testing we can assume, that the program runs well in its local system environment.

Conformance testing provides the proof against the specification of the implemented protocols, especially against the specification of standards. A certificate on conformance is the formal basis for the integration of a newly developed communication component into a network.
It turned out that conformance testing does not guarantee the right behavior of the new components in the network. Therefore an interoperability testing has to be performed to prove the interworking of the conforming systems.
From the point of view of the network user the possibility of interworking of different systems is not the only requirement. He also wishes to know about performance constraints and about the behavior in the case of errors in single components. This information may be obtained through performance and robustness tests.

A number of test systems has been developed in the past, e.g. /LINN83, RAFI86/. Most of them are dedicated to certain protocols or profiles of protocols. Their application by independent and commercial test centers requires the parallel running of several systems, often available on different hardware and equipped with different user interfaces. So it was found an up-to-date task to find a common solution of a test system architecture and methodology which can be used for and tuned to both different protocols and single/multiple layer implementations.

In this paper we will describe the architecture of the test system TEKOS and the tool environment in which it is embedded. TEKOS is a test system, which implements the external distributed test method in the form of the ferry concept /ZENG86/. Tools are available for the preparatory phase of the test process as well as for the evaluation of the test results.

In section 2 a description of the overall test environment is given. The chosen architecture of the test system is explained in section 3. Experience in using TEKOS together with its tools and directions of further work are discussed in section 4 and 5.

2. TEST ENVIRONMENT

Different test goals require also different test techniques. But nevertheless some methods and test data are good for several tests. Therefore it seems to be reasonable to develop a common test environment for almost all the tests.

The concept of our test environment is given in fig. 1. It consists of the test system TEKOS as the kernel, and of tools for the test process preparation and the test result evaluation.

The main goal of our approach is to develop a test system architecture and a test methodology which enable a division of the overall system into parts independent from the tested layer and and into parts devoted specifically to the implementation under test (IUT).

The first test system which stated such characteristics was the GENEPI /ANSA82/. In papers of the Fraunhofer Institute /WATS88/ and the University of British Columbia /CHAN89/ this quality is mentioned also.

Doubtless the way to go consists of

- the division of the protocol specific information from the generic parts of the test system and
- the provision of an apparatus to include the protocol specific parts during the test process into the data description and the protocol control flow.

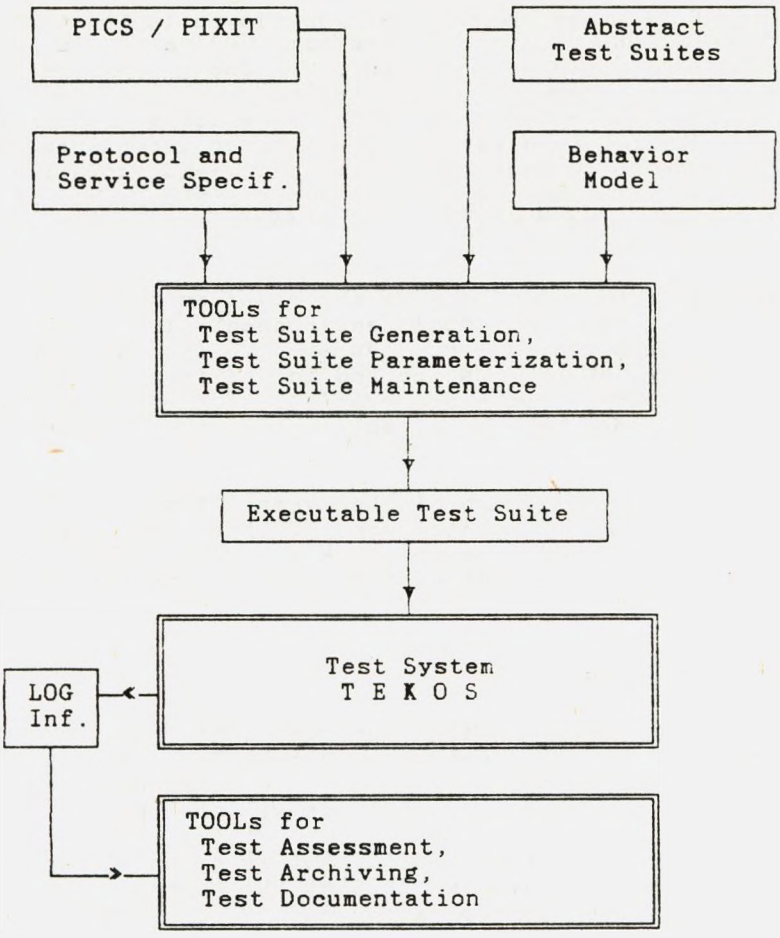


Fig.1 Test Environment of the Test System TEKOS

The different approaches are mainly characterized by different ways of division of the two types of information and by different points in the system architecture where these information types are matched again.
A further independence goal of the test system is related to the specific system under test (SUT). It can be tackled the same way as the protocol independence.

The dependencies are:

- the structure of the protocol data units (PDU)
- protocol specific relations between PDUs, e.g. enforced concatenation of PDUs

- (abstract) service primitives (ASP)
- the protocol control flow (logic and time dependent)
- values of parameters
- the used transmission service to the IUT
- interfaces between upper tester, lower tester and IUT

Our approach to solve the problem includes the tool part of the test environment and the test system themselves and consists of:

- a formal specification of test cases for the given protocol with a protocol independent language; this is done by tools in the preparatory phase of the test;
- a formal specification of the PDUs, ASPs and of predefined parameters in special description parts; this is done by tools on the basis of the formal specification of the protocols or if this doesn't exist by hand; these parts also contain references to default parameter values;
- the derivation and completion of the specific PDUs and ASPs on the basis of structure files; this is provided by the encoder module of the test system;
- the definition of a service interface independent of the used transmission service; this interface is provided by interface handlers which do the mapping onto the real service used;
- the transformation of the ASPs into real ones by the responder module at the upper point of control and observation (PCO).

Tools have been developed to aid in

- the derivation of test suites,
- the completion of test documents (Protocol Implementation Conformance Statement (PICS), Protocol Implementation Extra Information for Testing (PIXIT)),
- the transformation and parameterization of test suites,
- the assessment of tests,
- the archiving of test results,
- the generation of test reports.

All the tools are equipped with a common window based user interface. While the regime setting functions use the menu-windows of the window system, the tools operate with self-opened windows.

In the preparation phase of the test process executable test suites are derived in concordance with the chosen test purpose. The sources for this derivation are

- existing abstract test suites,
- the specification of the service and the protocol to be tested.

In the first case the use of abstract test suites delivered by OSI, CCITT or certification bodies is assumed. These test suites consist of a description of the test event tree together with parameters or parameter limitations. On the

basis of PICS, PIXIT and time and cost restrictions, they are completed with values for the open parameters and events related to the implementation of the service at the SUT.

The tool applicable to this process sets appropriate values to the parameters of the test suite according to the provisions of the PIXIT. A considerable problem is the integration of the complicated data structures of the PDUs and service data units (SDU) and the operations on them. The degree of automation of this process mainly relies on a formal notation of the abstract test suites, e.g. with TTCN.

In the case of the absence of standardized test suites the executable test suites must be derived from formal descriptions (SDL, ESTELLE, LOTOS). There is a large amount of publications dealing with this problem, e.g. /SARI88/ for ESTELLE-specified protocols and /STEE86/ for LOTOS descriptions. We use suites which are derived by methods of the theory of experiments with finite state machines /GRUN88/ at the Institute of Electronics and Computing Technique in Riga.

It should be mentioned, that the test cases for the basic interconnection test and a first set of capability test were derived by hand.

Among the tools for the preparatory phase of the test process there is a test suite management system. The test cases are stored as abstract and as executable ones. The first are written into the data base system without special requirements. The executable test cases are stored in the form of test event trees to whom parameter tables with different parameter values are connected. Each test case is defined by its tree and a dedicated value table.

Another tool supports the production of the necessary test documents in
- completing the PICS with respect to the relevant static conformance requirements,
- completing the PIXIT without conflict with the related PICS.

The kernel of our test environment is the test system TEKOS, which is described in section 3.

Executable test suites are the main input of the test system. The description of parameterized executable test suites forms the so called test program. Containing all the layer specific and IUT specific information, it is the main prerequisite to minimize the dependence of the test system from the test object.
The procedural course of the test suite and the events on the PCOs are specified in the test program . It consists of a declaration part, a message type description part, an action part, a test step description part, a default behaviour description part, and a default value part. In the declaration part constants, variables, messages and functions are declared. The types of the messages used must be included in

the message type description part. This part contains structure descriptions of the PDUs and of the SDUs of the underlying layer and the IUT in an abstract TEKOS specific format.

The action part is composed of statements to input and to output messages. Expected messages can be specified and distinguished through predefined parameters. Further statements exist for time out handling, control of the program flow, arithmetic and logic operations, and verdict assignment.

Predefined test steps, default behaviour on input messages and default values for messages can be described also. A history component supports the maintenance of consistency between all these parts.

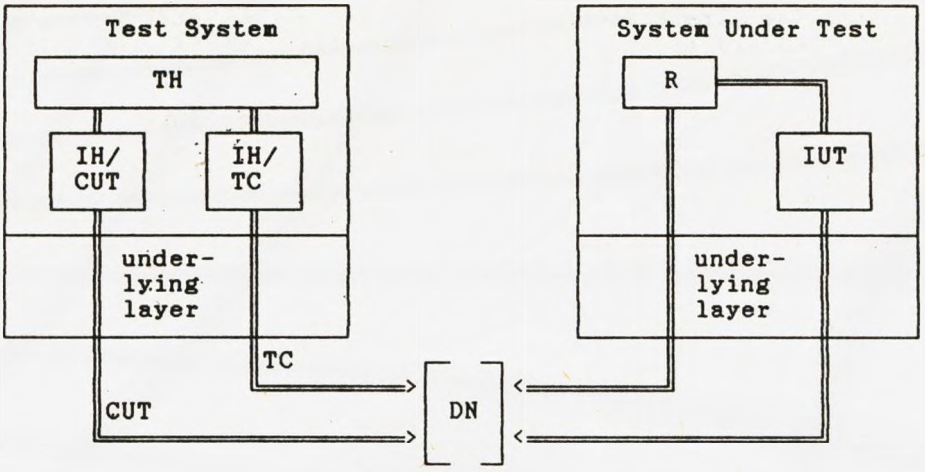
At the beginning of the test campaign of a new IUT the dialogue oriented work is characteristic. This way the test operator gets some feeling of the IUT and may complete his understanding of the PICS and PIXIT. Later in this campaign the large amount of test cases is processed automatically. The behavior is too complicated and too fast for real time control by the operator.

The third part of our test environment consists of tools for the evaluation of the test process, the archiving of the test results and the generation of the test reports. The need of a software tool to aid the test evaluation exists because of the considerable number of records in the conformance log information.
In the case of conformance testing the evaluation of the conformance log of the test is predefined in the test suite by the verdict assignments "pass", "fail" or "inconclusive". Based on this data a test report generator produces the report forms that have to be delivered to the clients.
The archive of the test results contains the test report information.

3. TEST SYSTEM ARCHITECTURE

In fig. 2 the main components of the test system and the SUT are depicted. To ensure an easy coordination of the actions of the upper tester and the lower tester as described in the abstract test methods of /IS9646/ an architecture was chosen similar to the astride concept of /RAFI86/ or the ferry concept of /ZENG86/. The functions of the upper and the lower tester, and the test coordination are included in the test system. In the SUT a responder is incorporated, whose functions are determined by the test system.

The test system itself consists of a test handler and of interface handlers for the connection under test and for the test connection. The structure of the test handler is shown in fig. 3. The Init and End components fulfill local functions of initialization and termination of the test system.



TH - Test Handler
IH - Interface Handler
CUT - Connection Under Test
TC - Test Connection
DN - Data Network
R - Responder
IUT - Implementation Under Test

Fig.2 Architecture of the Test System and the SUT

The main component of the test handler is the test program interpreter. It interprets the test program and puts log information about the test course into the conformance log of the current test campaign.

Abstract service primitives for the lower layer service interface and for the service interface of the IUT as well as protocol data units are handled during the interpretation. They are transformed from the representation of our test program language into the formats of the underlying layer and of the test protocol and vice versa.

In the case of an output of a message the formation function is fulfilled by the encoder. Incoming messages are transformed by the decoder. These components are independent of the tested layer and the specific service primitives of the underlying layer and the IUT.

The mapping of service primitives into specific interface data units is done by the interface handler of the connection under test (IH/CUT). The IH/CUT is layer dependent. It has to be modified when the test system uses different underlying layer services to manage the connection with the SUT.

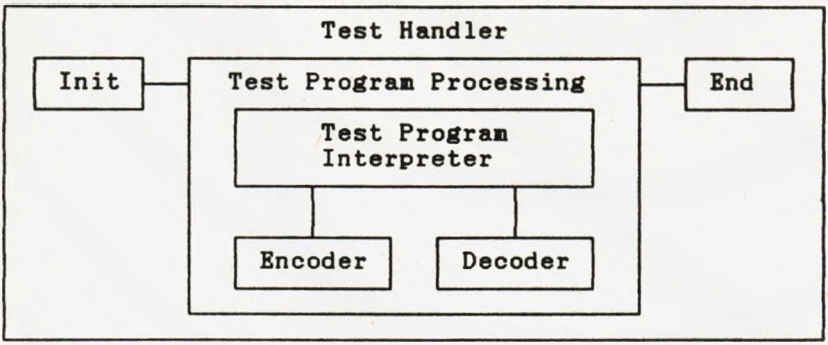


Fig.3 Structure of the Test Handler

The interface handler of the test connection (IH/TC) is determined by the test protocol and depends on the transmission service to support this protocol. This service is based on a low layer and in general will not be changed when testing different layers or layer groups.
The specific format of the SDUs of the IUT is mapped by the Test Responder (TR) to the ASPs. This logical part of the test system is located at the SUT.

The chosen architecture of the test system shows the possibility of building a system with a generalized architecture for testing different IUTs. The parts which have to be rewritten to test different IUTs can be located in specific modules which are of small size.

4. EXPERIENCE

The implementation of the first version of the test system TEKOS was done on a minicomputer K 1840 (VAX/VMS compatible). The programs of the test system are written in MODULA-2. The basic idea how to match the component structure onto a program structure was to consider the architecture under the aspect of possible parallel work. This led to a subdivision of the whole system into several processes. The processes communicate through events. This solution enables a well-defined interface between the groups of components formed this way.

The most sophisticated test campaign had been carried out for an OSI session layer implementation. It revealed failures within both the session layer implementation and the first version of TEKOS.
More than 5.000 test cases had been generated with the above mentioned tool covering the session protocol subsets BCS and BAS.

The testing was provided mainly during the protocol development to support the session implementation group. Besides the elimination of errors of the test system the most important results were got in connections with the handling of large amounts of data. The improvement of the test suite management, the parameterization of the executable test suites, the automation of the test campaign control and the test assessment tools was most urgent.

The final test campaign with all the test cases took one man-week for running the tests and about two man-month for the evaluation of the test results. To reduce the needed time the test result evaluation tool has to be improved to support the following functions:

- investigation of the conformance log in accordance with several given criterias;
- optional displaying of each parameter of the input and output messages;
- providing both interactive and batch mode operation of the tool.

The lack of test case selection criteria forced us to run all the test cases not knowing their specific weight in the test process. Therefore the error classes found during this test campaign are of importance for guiding further test. They are the following:

- Errors in parameter encoding and sequencing, wrong length indicators, invalid parameter values.
- Erroneous translation of the state transition tables of the standard into the implemented protocol machine.
- Insufficient measures for error detecting by the IUT.
- Wrong interpretation of the protocol.

5. CONCLUSIONS

The test system TEKOS as presented in this paper is an implementation of a test system architecture which is applicable to different IUTs.
The gained experience generated new ideas to improve the layer independence and to modify the test program notation. Doing this together with a better module management will also allow the execution of interoperability tests.
Independent of the delivering of abstract test suites by standardization bodies the demand on tools for generating own test cases on the basis of formal descriptions grows. They are necessary not only for conformance testing, but also for interoperability testing, performance and robustness assessment.
Not underestimating the use of the common sense for test case selection, the large amount of reasonable test cases requires test selection criteria based on test coverage measures.

Experiments have to be done to evaluate the time behavior of

the test system in more detail to guarantee its right behavior in the case of high speed application testing.

The presented research continues in a next phase which has been dedicated
- to fundamental research on test assessment, especially on test case selection and the test coverage measure,
- to the development of tools for a formal test suite description and test derivation from formal protocol specifications,
- to the further development of the test system.

REFERENCES

/ANSA82/ Ansart J.P.; GENEPI/A - A Protocol Independent System for Testing of Computer Communication Protocols, in: Network Inf. Proc. Syst., North-Holland, 1988.

/CHAN89/ Chanson S.T., Lee B.P., Parakh N.J., Zeng H.X.; Design and Implementation of a Ferry Clip Test System; Protocol Specific., Testing, and Verification IX; North-Holland, 1989.

/IS9646/ OSI Conformance Testing Methodology and Framework. ISO DIS 9646, 1989.

/GRUN88/ Grunskij I. S., Petrenko A. F.; Constructing Test Experiments with the Automata that Describe Protocols; Avtomatika i Vychislitelnaja Technika, N.4, 1988.

/LINN83/ Linn R.J., Nightingale J.S.; Some Experience with Testing Tools for OSI Protocol Implementations; Protocol Specific., Testing and Verification III; North-Holland, 1983.

/RAFI86/ Rafiq O. et al.; Towards an Environment for Testing OSI Protocols; Protocol Specific., Testing, and Verification VI; North-Holland, 1986.

/SARI88/ Sarikaya B., Eswara S., Koukoudilis V., Barbeau M.; A Formal Specification Based Test Generation Tool; Concordia University Montreal, 1988.

/STEE86/ Steenbergen C.; Memorandum Conformance Testing of OSI Systems; MSc-Thesis, Univ. of Technology T.H. Twente, Netherlands, 1986.

/WATS88/ Watson K.S.; Architecture of the Fraunhofer Conformance Test Systems; ENTERPRISE Conf., Baltimore, 1988.

/ZENG86/ Zeng H. X., Rayner D.; The Impact of the Ferry Concept on Protocol Testing; Protocol Specific., Testing, and Verification V; North-Holland, 1986.

SEQUENCE GENERATION FROM EFSMs FOR PROTOCOL TESTING

Alberto Faro (*) and Alexandre Petrenko (**)

(*) Istituto di Informatica e Telecomunicazioni, Universita' di Catania, viale A.Doria 6, 95125 CATANIA (Italy)
(**) Institute of Electronics and Computer Science, Latvian SSR Academy of Sciences, Riga, USSR

ABSTRACT

As known an important problem in computer nets is to verify that protocol implementations conform to the standard specifications. Generally these specifications are given by Extended Finite State Machines (EFSMs). The paper presents a method to detect all the specified faults of an Implementation Under Test (IUT) described by an EFSM; in particular for each fault a partial characterization set is defined together with the related partial checking sequence. These sequences are merged into a final test sequence which represents the test set for the given protocol entity.

1. INTRODUCTION

Conformance testing of protocol implementations is an important step to develop real open computer systems able to communicate via a common network [1]. The choice of the necessary tests is the key to solve this problem. Exhaustive testing is not realistic, therefore the only reasonable and constructive purpose of a test is to detect some faults, but not to show the absence of all of them. Heuristic tests may be quite effective with respect to their fault detection capability but their effectiveness depends on the experience of the designer. On the contrary tests produced by formal methods guarantee the detection of all the specified faults. This is why methods for checking experiments on abstract machines [2] have been used to test protocol entities represented in form of FSMs (Finite State Machines) or EFSMs (Extented Finite State Machine) [3,4,5]. However the fault model assumed in these applications does not reflect specific EFSM properties and may become unadequate to detect real possible faults (see sect.2).

The purpose of this paper is to demonstrate the necessity and feasibility of new test generation methods for detecting particular faults of a given IUT specified by a reference EFSM. Section 2 deals with the problems arising from the application of the method (based on the classical notion of characterization set [7]) for testing IUTs derived from not strongly connected FSMs. Section 3 generalizes this notion of characterization set. This allows us to specify different types of errors and to construct the necessarily testing sequences. Section 4 deals with the definition of local

errors and the construction of the related partial checking sequences; their concatenation represents a test set (i.e. a test scenario skeleton). Section 5 gives an application of the proposed method for testing a real protocol example.

2. PROTOCOL TESTING BY THE CHARACTERIZATION SET APPROACH

Let E be an EFSM, which has the set of inputs X, the set of outputs Y, the set of control states S, the set of contexts C, where $c \in C$ is k-tuple of context variables $V=\{v1,v2,\ldots,vk,\}$, d(vi) is the domain of vi, the number of different contexts is $|C|=|d(v1)|x\ldots x|d(vk)|$, R is the set of predicates $P=P(v1,v2,\ldots,vk)$, O is the set of different operations on variables from V, $so \in S$, $co \in C$ are initial control state and context. In the following we deal only with finite deterministic EFSMs without spontaneous transitions. Thus, the EFSM E is:

$$E=(X,S,Y,V,R,O,h,so,co) \qquad (1)$$

where $h:(X \times S \times R) \to (S \times Y \times O)$ is the transition function. A pair (s,c) is called a state of the EFSM. The state space is the set $G=(S \times C)$. In order to get a test sequence for a given EFSM methods from the theory of experiments on abstract automata [6],[7],[8] dealing with a pure FSM can be used. For this aim, several papers [4],[5] transform the reference EFSM into the equivalent FSM for further derivation of test sequences by known generation methods. The size limitation may force to test only submachines of a given EFSM [4],[11]. Suppose that from an EFSM E given in form (1), one has got an equivalent FSM A:

$$A=(X,(S \times C),Y,f,g,(so,co)) \qquad (2)$$

where $(S \times C)$ is the set of states of FSM, $f: (X \times S \times C) \to (S \times C)$ is the transition function, $g: (X \times S \times C) \to Y$ is the output function. Known test sequence construction methods work only if a given FSM is reduced and strongly connected. If A is such a machine, the existing methods [7] firstly need that we find the characterization set consisting of those input sequences of A producing different output sequences for different states. Then the testing tree is constructed: its nodes are the FSM states, its branches are the FSM transitions. Each transition is represented only once. The initial state (so,co) is the root of this tree.

A test sequence Q is constructed by traversing all the partial paths of the testing tree or subtree (i.e. the paths from the root to every node of this tree) and by applying to every node each sequence of the characterization set followed by a reset sequence to come back to the root of the tree. The characterization set may have only one sequence; in this case it is called the distinguishing sequence [6,9]. Let Z be the class of FSMs describing all the different implementations of A and $A' \in Z$ be an IUT, we introduce the checking sequence Q, as defined in [7],[9], that allows us

to verify A'=A or A'<> A, provided that the number of states of each A'∈Z is not more than that of A. Each part of Q, which forces FSM from (so,co) into (so,co) represents an individual test [1]. Now we consider the case in which A is not a reduced strongly connected automaton, but at same time let us assume that A satisfies the following relations:

$$\forall\, si,sj\in S,\ i <> j,\ \exists\ c\in C,\ \exists\ \alpha\in X^*,\ g(\alpha,si,c) <> g(\alpha,sj,c); \qquad (3)$$

$$\forall\, ci,cj\in C,\ i <> j,\ \exists\ s\in S,\ \exists\ \alpha\in X^*,\ g(\alpha,s,ci) <> g(\alpha,s,cj); \qquad (4)$$

where X* is the set of all input sequences of the alphabet X. From (3) it follows that no pair of control states of E can be substituted by one state and from (4) that the same is true for the contexts. An EFSM E is called <u>the unredundant EFSM</u> if (3),(4) hold in its FSM form A. Under our assumptions an FSM derived from the given unredundant EFSM has unreachable and/or equivalent states. Quite understandable and natural decision to delete the unreachable states from G=(SxC); however this leads to difficulties in the specification of faults detected by the resulting checking sequence, that is in the definition of the class Z.

We illustrate the above problem by using the EFSM E given by the state table shown in Fig.1, where the initial state (so,co) is (0,0), whereas P1,P2,P3,P4 are the following predicates: P1=¬v1, P2=v1, P3=¬v1&v2+v1&¬v2 and P4=¬v1&¬v2+v1&v2. Let us construct the equivalent FSM A with 12 states as shown in Fig.2. Only 7 states of A are reachable from the initial state, as it follows from the testing tree (Fig.3). By deleting all the unreachable states from A we can get the reduced strongly connected FSM A' with 7 states. The characterization set of this automaton has only one sequence (distinguishing sequence) Xd=x2x2x1 (Fig4). According to the above given scheme we construct the checking sequence Q to distinguish the given reduced FSM A' from all the unequivalent FSMs whose state number is equal to or less than 7, as follows: Q = α1x1α1x3α2x2α3x3x1α1x3x3α5x3x2 α2x2x2α6x2x1α3x2x3α7x3x3x1α1x3x3x2α5x3x3x3α1x2x2x1α1x2x2x2α3 x2x2x3α1x2x3x2α4x2x3x1α1x2x3x3α7x2x3x2x1α4x2x3x2x2α4x2x3x2x3 α7, where αi=Xd.ri, i=1,2,...,7 (Fig.4).

Now let us go back to the given EFSM and try to inspect the fault detection capability of the checking sequence constructed above on the base of Xd=x2x2x1. Assume IUT executes by error v1:=0 (instead of v1:=1) when performing the transition from s0 to s1 under x2. Then E goes into (1,0), but not into (1,2). The reaction to Xd in (1,2) should be y1y3y1, but in (1,0) it is actually y2y2y2 (Fig.4) and the error will be finally found. Now let us assume another fault dealing with the context variable v2 in the transition from s1 to s2 under x3. IUT executes v2:=0 (instead of v2:=1) and goes into the state (2,0) instead of passing to (2,1). The reaction to Xd in (2,0) and (2,1) is the same: y1y1y1, and therefore the error cannot be revealed (Fig.4).

Thus, the above constructed checking sequence based on Xd=x2x2x1 detects some fault operations on context variables but not all of them. It is obvious that the second fault increases the number of states of A' from 7 to 8, but both faults are equally feasible in an implementation of EFSM E. In the next section we present the proper solution of the discovered problem.

3. COMPLETE AND PARTIAL CHARACTERIZATION SETS

To solve the above problem we firstly introduce some necessary constructions. Let G=(SxC) be the state space of the given completely specified EFSM E; $\alpha \in X^*$ be an input sequence. We define the following set called the reachable state space of E, $Go \in G$, that is:

$$Go=\{(s,c) \mid f(\alpha,so,co)=(s,c) \ \& \ f(\alpha',s,c)=(so,co)\}.$$

We assume that the EFSM is strongly connected (i.e. the initial state can be reached from any reachable state), as follows: $\{(s,c) \mid f(\alpha,so,co)=(s,c)\}=Go$

By $\alpha(s,c,G')$ we denote the set of input sequences $\alpha \in X^*$ distinguishing state (s,c) from all unequivalent states of $G' \in G$:

$$\alpha(s,c,G')=\{\alpha \mid g(\alpha,s,c) <> g(\alpha,s',c'); \ (s',c') \in G', (s',c') <> (s,c)\} \qquad (5)$$

Here we consider only unredundant EFSMs; from (3),(4) it follows that $\alpha(s,c,G') <> \emptyset$ for these machines. Then the union of all $\alpha(s,c,G')$ with $(s,c) \in G''$ gives $\alpha(G'',G')$ which is the set of input sequences distinguishing all the states of the set $G'' \in G$ from all the unequivalent states of the set $G' \in G$. $\alpha(G'',G')$ is called a complete characterization set of a given EFSM if G''=Go, G'=G; whereas $\alpha(G'',G')$ is called a partial characterization set if $G'' \in Go$, $G' \in G$.

In particular, if G'=G''=Go, then $\alpha(Go)$ is simply the characterization set [7] of a reduced strongly connected FSM whose state set is Go. And finally $\alpha(Go)$ is the distinguishing sequence [6],[9], if $|\alpha(Go)|=1$. For the example under consideration we have $\alpha(Go,G)=\{x3; x2x2x3x3x3\}$ which allows us to recognize any reachable state or show that the IUT went by an error to an unreachable unequivalent state (Fig.5). The complete checking sequence has the structure of the partial checking sequence given in sect.2 with $\alpha i=\alpha i1,\alpha i2$; $\alpha i1=x3ri1$; $\alpha i2=x2x2x3x3x3ri2$; i=1,2,...,7. Fig.5 shows that correctly $\alpha(Go,G)$ does not distinguish between (0,0) and unreachable (0,2) since they are equivalent; analogously unreachable (0,1) and (0,3) are correctly not distinguished by $\alpha(Go,G)$. However $\alpha(Go,G)$ does not distinguish also between (1,1) and (2,0), which are neither equivalent nor reachable states.

The separation of states into control states and contexts can be useful for the characterization set construction

under certain assumptions about the implementation of an EFSM. The most frequently used hypothesis about IUT assumes that errors derive from the implementation of the output function g and not from the transition function f. A checking sequence Q detecting faults of this particular type can be constructed rather easy by making a tour through all the branches of the testing tree without applying any additional sequence to the nodes (a transition tour) [8],[3],[4],[5].

Let us consider the transitions to the state (s,ci)∈Go; the complete characterization set α(Go,G) is generally needed in order to recognize the result of any fault transition into an arbitrary unequivalent state (s',c')∈G. Now we suppose the error can be only in the resulting control state s' but not in the final context ci. To verify the state (s,ci) under this assumption we have to distinguish (s,ci) not from all the (s',c')∈G, but only from the subset of states having context ci, that is:

$$G(ci)=\{(s',ci)\,|\,(s',ci)\in G \;\&\; (s',ci)<>(s,ci)\}.$$

If we put G(ci) into (5), then α(s,ci,G(ci)) gives the input sequences, which allow us to distinguish (s,ci) from all (s',ci)∈G, (s',ci)<>(s,ci).

If Go(ci)={(s,ci)|(s,ci)∈Go}, then the set α(Go(ci),G(ci)) is called the context ci characterization set. We consider only unredundant EFSMs, therefore α(Go(ci),G(ci))<>∅. The above set distinguishes any reachable state between each others and from all the unequivalent unreachable states with the same context ci∈C. To construct a test sequence Q detecting these faults we have to build |C| context characterization sets α(Go(ci),G(ci)). Q formed by these sets is called the control state checking sequence. As an illustration we costruct the context characterization sets for our example (Fig.6):

For co: Go(co)={(0,0),(1,0)}, G(co)={(0,0),(1,0),(2,0)},
α(G(co),G(co))=x2x2.
For c1: Go(c1)=(2,1), G(c1)={(0,1),(1,1),(2,1)},
α(Go(c1),G(c1))=x3x3.
For c2: Go(c2)={(1,2),(2,2)}, G(c2)={(0,2),(1,2),(2,2)},
α(Go(c2),G(c2))=x2.
For c3: Go(c3)={(1,3),(2,3)}, G(c3)={(0,3),(1,3),(2,3)},
α(Go(c3),G(c3))=x1x3x2.

The control state checking sequence, based on the produced context characterization sets, has the same structure as the partial checking sequence of sect.2, where α1=x2x2x1, α2=x2x2x1, α3=x2x1, α4=x1x3x2x3x1, α5=x3x3x1, α6=x2x2x1, α7=x1x3x2x1 (Fig.6).

In similar way we can treat faults which influence a context (but not a control state at the same time). A test sequence Q detecting these errors is based on sets α(Go(si),G(si)), si∈S. These sets are called control state si characteriza-

tion sets. If (3),(4) hold, then this set is not empty. In our example there are the following characterization sets:

For so: Go(so)=(0,0), G(s)={(0,1),(0,3)},
α(Go(so),G(so))=x3x2;
For s1: Go(s1)={(1,0),(1,2),(1,3)},
G(s1)={(1,0),(1,1),(1,2),(1,3)},
α(Go(s1),G(s1))=x2x2.
For s2: Go(s2)={(2,1),(2,2),(2,3)},
G(s2)={(2,0),(2,1),(2,2),(2,3)},
α(Go(s2),G(s2))=x2x2x3x3.

By using these sets a test sequence is constructed in the same way as the above given control state checking sequence and it is called the context checking sequence Q. Thus, untill now we have considered four different hypotheses about possible faults of EFSM implementations, that is four classes Z1, Z2, Z3, Z4 of IUTs:

- class Z1: faults can be in outputs and in context or control states. A complete checking sequence Q1 is based on α(Go,G).
- class Z2: faults can be in outputs and in control states. A control state checking sequence Q2 is based on α(Go(ci),G(ci)).
- class Z3: faults can be in outputs and in contexts. A context checking sequence Q3 is based on α(Go(si),G(si)).
- class Z4: faults can be only in outputs. A partial checking sequence Q4 is a tour of the testing tree, that is, α(G)=∅.

In sect.4 we show that it is also possible to specify some smaller subclasses of IUTs in order to ease the test sequence generation with the penalty of worse fault coverages.

4. CHARACTERIZATION SETS FOR LOCAL FAULTS

We define a certain type of faults to specify a class of EFSMs under test. A fault is called local if it can force the EFSM not into the correct state (s,c)∈Go but into some unequivalent state (s',c') which belongs to the set of possible final states G(s,c)∈G (i.e. faults with a limited consequence). To detect a local fault in any transition into the state (s,c) a partial characterization set α(s,c,G') has to be constructed.

To illustrate this we use our example. Suppose that the test designer has decided to construct tests for local faults resulting in erroneous operations on the context variable v1 (0 or 1 only). He assumes that errors migth be only when value of v1 has to be changed. There are four transitions in the testing tree changing the value of v1: f(x2,0,0)=(1,2), f(x1,2,2)=(0,0), f(x3,2,2)=(0,0), f(x2,2,3)=(0,0). To verify these transitions we define the following partial characterization sets: G(1,2)=(1,0), α(1,2,G(1,2))=x1, G(0,0)=(0,2). States (0,0) and (0,2) are equivalent, therefore a possible

local fault operations on v1 in the transitions to (0,0) are not essential (i.e. they do not change the EFSM). Thus, the assumption about the limited local faults, resulting only from erroneous context variable v1 values leads to the construction of the following simple test sequence x2x1x2x1, where the first symbol forces the EFSM from (0,0) into (1,2), the second symbol is simply the characterization set and the last two symbols move it back into the initial state. In the same way we can construct a sequence which checks operations on context variable v2.

5. APPLICATION OF THE METHOD

This section shows the application of the proposed method for detecting faults resulting from erroneous operations on context variables of a flow control protocol. Flow control is a typical mechanism that adapts the data transmission rate of a protocol entity to the data reception rate of its remote partner. This mechanism is present in many well known protocols as for example CCITT-X25 and ISO transport protocol. For simplicity we apply the method only to test the transmitting entity but it can be applied in the same way to the receiving one. Transmitter may be in two states: data transfer and wait for ack. It behaves as follows:

a) Data transfer state: It sends the message received from the user to the remote entity if the sequence number v1 of the message is within the window, that is v1<v2+W where v2 is the lower window edge and W is the window size; otherwise it goes into the state "wait for ack" by ignoring the request coming from the user. It updates the lower window edge v2 when it receives an ack from the remote entity, that is it puts v2:=N where N is a parameter of the received ack used by the receiver to acknowledge (N-v2) received messages. This on condition that v2<N<v1.
b) Wait for ack state: It simply ignores the request to send message coming from user. It behaves as in data transfer state when it receives an ack from the remote entity.

Note that the above flow control mechanism could be obtained by using only the data transfer state. However if the transmitter consists of only one control state it is easy to verify that it has to perform more operations to handle the possible inputs than those needed by the above version. The EFSM specifying the transmitter is given in fig.7, where:

- so is the data transfer state;
- s1 is the wait for ack state;
- x1 is the input coming from the user to send message to the remote entity;
- x2(N) is the input coming from the remote entity to acknowledge (N-v2) received messages;
- y(v1) is the output message numbered v1 directed to the remote entity;
- P1 is a predicate that is true when v1<v2+W;
- P2 is a predicate that is true when v2<N<v1.

Now we construct the sequences checking context variables v1 and v2 in state so. For simplicity we assume that remote entity always sends ack with correct N. This allows us to consider only a partial testing tree shown in Fig.8, where we have assumed W=2 and v1,v2 $\in$ {0,1,2,3}. The number of different contexts is 16 and four of them (i.e. v1v2 = 30,01,12,23) are not reachable as it can be seen in Fig.8. We limit our analysis to two cases: i) errors may be only in the value of v1$\in${0,1,2,3}, ii) errors may be only in the value of v2$\in${0,1,2,3}.

The characterization set for the first case consists of x1, whereas the characterization set for the second one consists of x1x1x1. This result is obtained by applying the proposed method and it can be verified by using the table in Fig.9. A partial checking sequence constructed from the partial testing tree by the sequence x1x1x1 allows us to discover errors actually either in the value of v1 or in the value of v2. In general it seems to us that for arbitrary window W the length of the above sequence is W+1. However this sequence does not distinguish the contexts v1v2=20,31,02,13 among each others. The goal of discovering both the above types of errors at the same time can be reached at the price of increasing the computation complexity and the length of the resulting test sequence. The example shows that the construction of a partial checking sequence does not need the complete specification of the protocol entity; it needs only the necessary subprotocol.

6. CONCLUSIONS

This paper shows how to full the gap between the protocol implementation tests produced by a simple transition tour of an equivalent FSM and those produced by a cheking sequence which distinguishes a given EFSM from any unequivalent EFSM having the same or less number of states. Heuristic and random tests, e.g. [10], fall into this domain by their fault detection capability. However they cannot reach the testing purpose (in the above given sense) because they do not have it at all. The approach outlined in this paper allows the test designer to generate interesting from his viewpoint tests and to achieve the testing purposes formulated in terms of his assumptions about faults in an IUT. The approach is based upon a generalizied notion of the characterization set. A particular type of a characterization set reflects an assumed model of possible faults in a protocol implementation from a given EFSM. We note that not only partial but also complete characterization sets exist even for those EFSMs which are neither completely specified nor deterministic up to a certain limit. This is demonstrated by a companion paper [13].

REFERENCES

[1] ISO/Working draft for OSI Conformance Testing Methodology and Framework - ISO/DP9646, 1987.
[2] Gill,A.Introduction to the Theory of Finite State Machines, McGraw-Hill 1962.
[3] Bochmann,G.V.;Sarikaya,B.Some Experience with Test Sequence Generation for Protocols.in Proc.2th Int.Workshop Protocol Specification,Testing and Verification,1982.
[4] Favreau,J.-P.;Linn,R.J.Automatic Generation of Test Scenario Skeletons from Protocol Specification written in ESTELLE.in Proc.6th Workshop Protocol Specification, Testing and Verification,1986.
[5] Castanet,R.;Sijelmassi,R.Methods and Semi-automatic Tools for Preparing Distributed Testing. Proc.6th Workshop Protocol Specification, Testing and Verification,1986.
[6] Gonenz,G.A Method for Design of Fault Detection Experiments.in IEEE Trans.on Comp.,vol C19-6,1970.
[7] Chow,T.S.Testing Software Design Modeled by Finite State Machine.in IEEE Trans.on Software Eng. vol SE4-3,1978.
[8] Naito,S.Tsunoyama,M. Fault Detection for Sequential Machines by Transition Tours. Proc.IEEE Fault Tolerant Comp.Conf.,1981.
[9] Kohavi,Z.Switching and Finite Automata Theory. N.Y.McGraw-Hill,1978.
[10] Linn,R.J.;McCoy,W.H. Producing Tests for Implementations of OSI Protocols.in Proc.of 3th Workshop Protocol Specification, Testing and Verif. N.-H.Publ. 1983.
[11] Alfonzetti,S.;Casale,S.;Faro,A. A Theory of Subprotocols.in Alta Frequenza,vol.LII,1,1983.
[12] Petrenko,A.F.;Timokhovich,E.A. An Approach to the Certification of a Protocol Implementation.in Proc.of All-Union Conf. Computer Networks (in Russian),Riga,USSR,1987.
[13] Faro,A.; Petrenko,A. Checking experiments for incompletely specified and non deterministic FSMs. Internal report under publication.

	x1	x2	x3
so	so,y1	s1,y2,v1:=1	s1,y3,v1:=0
s1	P1: so,y2; P2: s1,y1	P3:s2,y1,v2:=0; P4: s1,y2	s2,y2,v2:=1
s2	so,y1, v1:=0 v2:=0	P1: s2,y1; P2: s1,y3	P3: so,y2, v1:=0,v2:=0; P4: s2,y2

Fig.1. The control state table of the EFSM E.

s	v1 v2	c	x1	x2	x3	Go
0	0 0	0	(0,0),1	(1,2),2	(1,0),3	*
	0 1	1	(0,1),1	(1,3),2	(1,1),3	
	1 0	2	(0,2),1	(1,2),2	(1,0),3	
	1 1	3	(0,3),1	(1,3),2	(1,1),3	
1	0 0	0	(0,0),2	(1,0),2	(2,1),2	*
	0 1	1	(0,1),2	(2,0),1	(2,1),2	
	1 0	2	(1,2),1	(2,2),1	(2,3),2	*
	1 1	3	(1,3),1	(1,3),2	(2,3),2	*
2	0 0	0	(0,0),1	(2,0),1	(2,0),2	
	0 1	1	(0,0),1	(2,1),1	(0,0),2	*
	1 0	2	(0,0),1	(1,2),3	(0,0),2	*
	1 1	3	(0,0),1	(1,3),3	(2,3),2	*

Fig.2. The state table of FSM A.

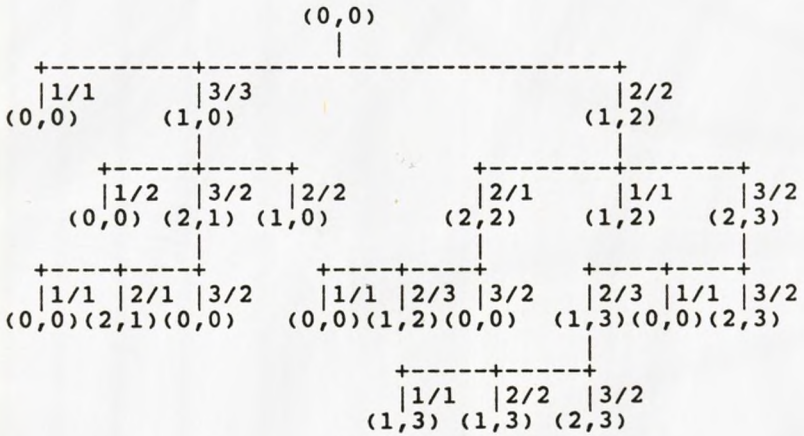


Fig.3 The testing tree of FSM A.

N	state	output to Xd=x2x2x1	reset seq.ri
1	(0,0)	2,1,1	-
2	(1,0)	2,2,2	-
3	(1,2)	1,3,1	x2x1
4	(1,3)	2,2,1	x3x1
5	(2,1)	1,1,1	-
6	(2,2)	3,1,1	-
7	(2,3)	3,2,1	x3x1
8	(0,1)	2,2,1	
9	(0,2)	2,1,1	
10	(0,3)	2,2,1	
11	(1,1)	1,1,1	
12	(2,0)	1,1,1	

Fig.4. Reactions of E to Xd.

state	x3	ri1	x2x2x3x3x3	ri2
(0,0)	3	x1	2,1,2,3,3	x1
(1,0)	2	x3	2,2,2,2,3	x1
(1,2)	2	x1	1,3,2,2,2	x1
(1,3)	2	x1	2,2,2,2,2	x1
(2,1)	2	--	1,1,2,3,2	x1
(2,2)	2	--	3,1,2,3,2	x1
(2,3)	2	x1	3,2,2,2,2	x1
(0,1)	3		2,2,2,2,2	
(0,2)	3		2,1,2,3,2	
(0,3)	3		2,2,2,2,2	
(1,1)	2		1,1,2,2,2	
(2,0)	2		1,1,2,2,2	

Fig.5. Reactions of E to α(Go,G).

α(Go(ci),G(ci))	state	output sequen	ri
x2x2	(0,0)	2,1	x1
	(1,0)	2,2	x1
	(2,0)	1,1	--
x3x3	(0,1)	3,2	--
	(1,1)	2,2	--
	(2,1)	2,3	x1
x2	(0,2)	2	--
	(1,2)	1	x1
	(2,2)	3	x2x1
x1x3x2	(0,3)	1,3,1	--
	(1,3)	1,2,3	x3x1
	(2,3)	1,3,2	x1

Fig.6. Reactions of E to α(Go(ci),G(ci)).

	x1	x2(N)
so	P1: so,y(v1), v1:=v1+1 ¬P1: s1	P2: so, v2:=N ¬P2: so
s1	s1	P2: so,v2:=N ¬P2: s1

Fig.7. Control state table of EFSM.

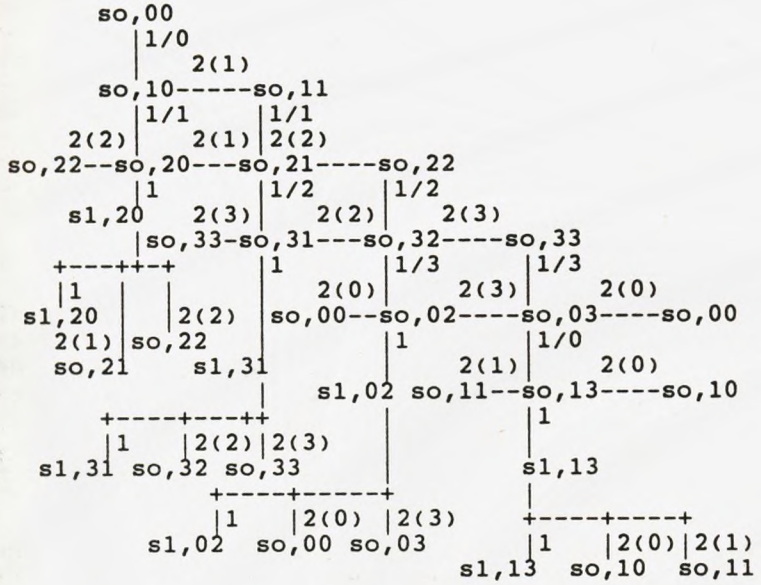


Fig.8. Partial testing tree of EFSM E

v1v2	x1	x1	x1
00	0	1	-
10	1	-	-
20	-	-	-
11	1	2	-
21	2	-	-
31	-	-	-
22	2	3	-
32	3	-	-
02	-	-	-
33	3	0	-
03	0	-	-
13	-	-	-
01	0	1	2
12	1	2	3
23	2	3	0
30	3	0	1

Fig.9 Reactions of EFSM to x1x1x1

CONFORMANCE TESTING SYSTEM FOR THE KASK NETWORK: SOME IMPLEMENTATION PROBLEMS

Artur DUBIELEWICZ, Iwona DUBIELEWICZ,
Witold KOMOROWSKI, Janusz RATAJCZAK

Computing Centre,
Technical University of Wrocław,
Wrocław, Poland

The conformance testing system realized as a part of the National Academic Computer Network (KASK project) follows the suggestions and recommendations given by ISO in DP 9646 (January 1988). The prototype system is implemented on an IBM PC/AT compatible computer with the IBM SDLC ADAPTER

The problems presented in the paper arose during the design as well as the implementation processes. The most important of them are: standardization of the interfaces to the transport and network layers, implementation of the translator of TTCN notation, definition of universal Tester Driver-Responder Protocol.

1. INTRODUCTION

The National Academic Computer Network (KASK project) is a heteregenous network implemented since 1981. Hence only the layers 1-3 are implemented according to the OSI standards, the transport layer being a modified version of the early recommendation of ISO 8073 standard and higher layers being no standards implementations at all.

Considering the current architecture of the KASK network, the conformance testing system is limited and can only test the implementations of the transport protocol. The main reason for the realization of the conformance system for the KASK network is to enable:
- conformance testing for the transport protocol implementations, and
- non-standard testing, i.e., testing of some special features and behaviours of the implementations of modified transport protocol in the various KASK host computers.

Hence, the main goal of this system is to guarantee correct and reliable testing in a simple way which can be easily repeated. It can be reached by
- using standard conformance tests, and
- automation of the testing process, as much as possible.

In our system, these postulates are fulfilled by means of:
- computer-aided selection of the tests to be executed ,
- automatic translation of any test suite/group/case written in TTCN,
- automatic generation of all the combinations of the test parameters values,
- automatic DRIVER - RESPONDER communication and control,
- automatic registration of all the events and suspicious situations observed during testing in a human-readable form.

2. CONFORMANCE SYSTEM CHARACTERISTICS

The distributed test method is applied in our conformance system. It requires direct access to the upper boundary of the implementation under test (IUT) and consequently the detailed specification of the control and observation requirements at both ends should be given.

The realization of the distributed method implies that much more work must be done to connect the upper tester to the transport protocol layer in every tested system. To minimize these efforts, the upper tester is designed in such a way that only one part of it, namely this one which communicates with the IUT, will be constructed and coded separatly for every system. The rest of the upper tester is designed as a portable module, coded in ahigh level language, and can be easily implemented on any system.

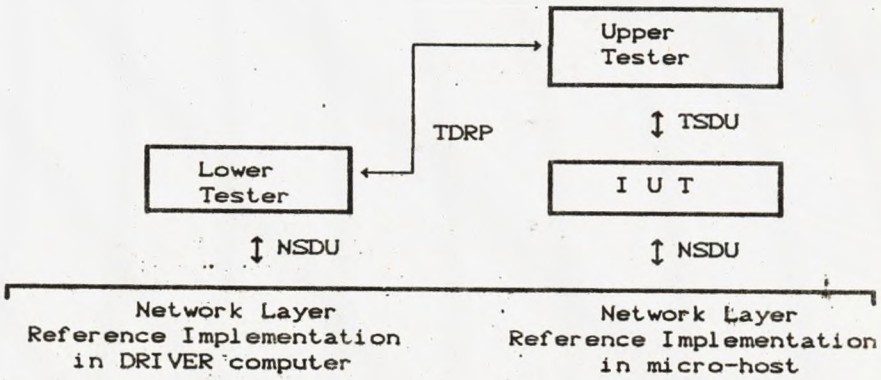


FIGURE 1
Architecture of the conformance testing system
for the KASK network.

The communication between both parts of the testing system is achieved by means of the Tester Driver-Responder Protocol (TDRP), which was designed especially for this purpose. The architecture of the conformance testing system for the KASK network is shown in Fig.1.

As mentioned above, the testing system consist of two main elements: DRIVER and RESPONDER. The DRIVER is realized as the encoder/decoder of the protocol data units (PDU). It means that the transport protocol data units - TPDUs (placed in the network service data units -NSDU) - are sent, accross the interface between the DRIVER and the network layer, to the RESPONDER.
This type of conformance system architecture guarantees that the testing of the most interesting features of the protocol implementation is possible. However, it needs the following two assumptions to be fulfilled:
1. Implementation of the lower layers are correct implementations of standards,
2. There exists a well-defined interface to the lower layer theDRIVER cooperates with.

Our conformance testing system is operating in a special network software environment, designed and coded for the micro-host [4].The main features of this network software, useful for our purposes, are:
- a special "network kernel" (as the DOS shell) which realizes functions such as programs scheduling and activation, buffers and timers management, interrupts handling,
- the 1-4 ISO protocol layers,
- due to well-defined interface between every two adjacent layers, they can be added or (ex)change in an easy way.

In the conformance system we used the kernel and 1-3 layers programs. The DRIVER is placed above the network layer. The RESPONDER is situated above the transport layer (which is the layer under testing) within the microcomputer which operates as a micro-host in the KASK network.

In the software we used, the second requirement previously mentioned is obviously realized, and the first one will be fulfilled after intensive testing of the microhost software.

The DRIVER's program is situated in an IBM PC/AT compatible computer equipped with 4MB extended RAM and an SDLC Adapter card which enables the connection of this computer to a KASK node.

3. FUNCTIONAL CHARACTERISTICS

The conformance testing system offers some functions called operator commands. The main screen of our system, with a complete list of operator commands, is shown in Fig. 2.

The most interesting among the listed commands is the RUN_TEST command which activates the testing process. The execution of this command is shown in Fig.3.

Due to the difficulty of describing the execution order of each test case of the suite, it is proposed to create an independent description in form of a text file named Suite

Configuration File (SCF). The name of this file constitutes the parameter of the RUN_TEST command and such information is sufficient to determine the order of test suite execution.

The SCF consists of the following two parts:
- description of formal test parameters, and
- description of the test suite execution structure.

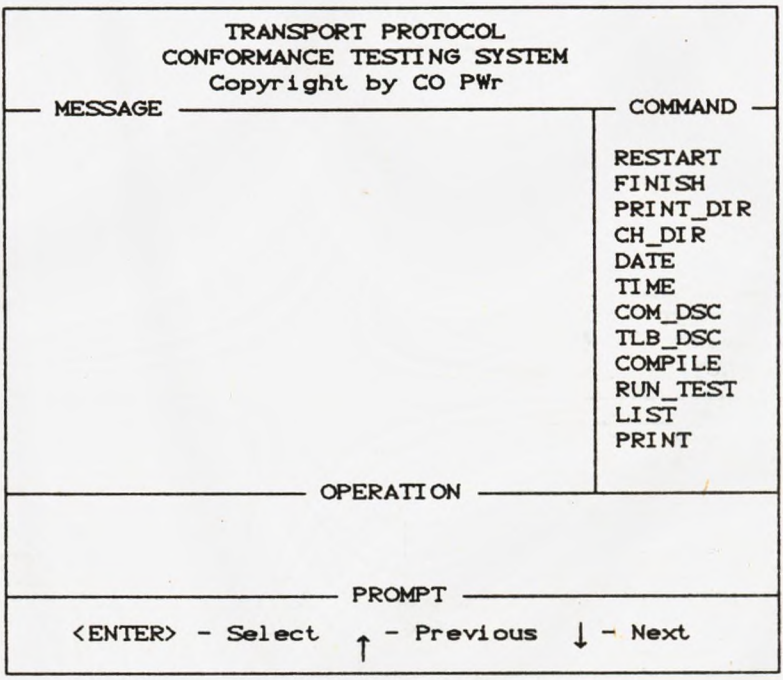


FIGURE 2
Main screen of the conformance testing system.

The first part of SCF is used to define values of all parameters which will be taken into account during the case test execution. Every test suite parameter can be given as a single value or as a set of values. Test will always be executed for all combinations of parameters values. It relieves the user from running multiple test cases with different values for the same parameters.

The second part of the SCF defines the test suite execution structure. It takes a graph form. It defines an execution sequence of test cases in the suite.

The graph is described by a list of nodes. For each node it should be given the following information:
- a node name,

- a set of test cases attached to that node,
- a list of successors.

The order of the nodes in the SCF is not important. Test cases will be executed according to the order they appear in the list associated with the currently processed node. Similarly, the order of the list of successors determines the order in which the test case sets will be executed.

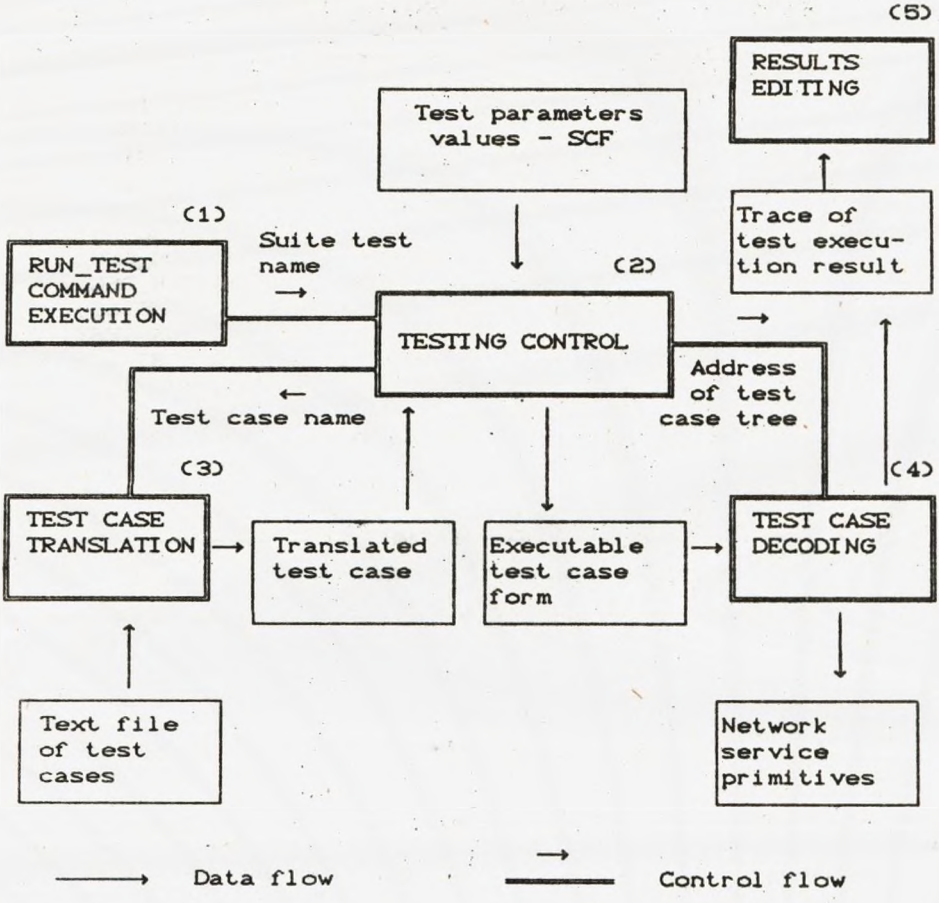


FIGURE 3
Data flow during RUN_TEST command execution.

To overview the testing history in a more readable form, the process of decoding the values of ASP and PDU parameters is performed on the operator demand, using the PRINT command. The decoded data can also be recorded on a hard/floppy disk and then printed.

The other commands are auxiliary ones and enable the operator to obtain information about actual time and data, the directory content and also some additional information on the test suites existing in the system (with a short description about their structures and purposes).

4. DRIVER ARCHITECTURE

The software architecture of the DRIVER is consistent with the data flow in the system (shown on Fig.3). It consists of 5 modules:

- *Command Interpreter*, whose main function is to read, recognize, and execute the commands of the operator; it calls the other DRIVERs modules according to the function which should be performed,
- *TTCN Translator* performs the lexical and syntactical analysis of the test case; it generates the internal form of the abstract test case; this internal form is the tree structure (with the complex data structures) which describes the types and values of all the objects defined in the test.
- *Code Generator* activates and controls the test suite/group/case execution; it creates the executable test case from the given internal form of the abstract test case and activates the Test Controller module. Correct execution of all test cases or detection of an incorrect test result terminates the operation of Code Generator, and an appropriate message is sent to Command Interpreter.
- *Test Controller* interprets the executable test case; it communicates with the network layer by sending and receiving service primitives; it compares the expected behaviour with the actual one and generates the result of the test execution and send this information to Code Generator.
- *Registration module* performs the decoding/encoding of data gathered during the testing process (it prints/displays them in a readable form).

All modules but one are coded in C language. Only the TTCN translator is coded in TURBO Pascal v.5.0. Communication between C and Pascal modules was achieved by means of the 'spawn' construct of the C language. The input parameters sent to the Translator take the form of program parameters (as defined in C language). But the ouput parameters and data sent/generated by Translator are placed in the memory pointed to by the C program (the address of this memory is one of the input parameters). The way in which the other modules are activated and communicate among themselves is defined by the software environment in which they operate [4].

5. RESPONDER

The Responder resides in the tested system and interacts directly with IUT via Transport Service interface. It interacts indirectly with the Driver by means of a specially defined Test Driver-Responder Protocol (TDRP).

The Responder consists of two parallell parts - the controlling responder (TRC) and the testing responder (TR). The connection handled by the TRC is used to exchange the TDRP commands and responses while the second connection handled by the TR is used for the actual test traffic. Both parts of the Responder are almost identical but while the TRC's behaviour is defined at system generation time, the TR's behaviour can be programmed and then monitored from the Driver by means of TRDP.

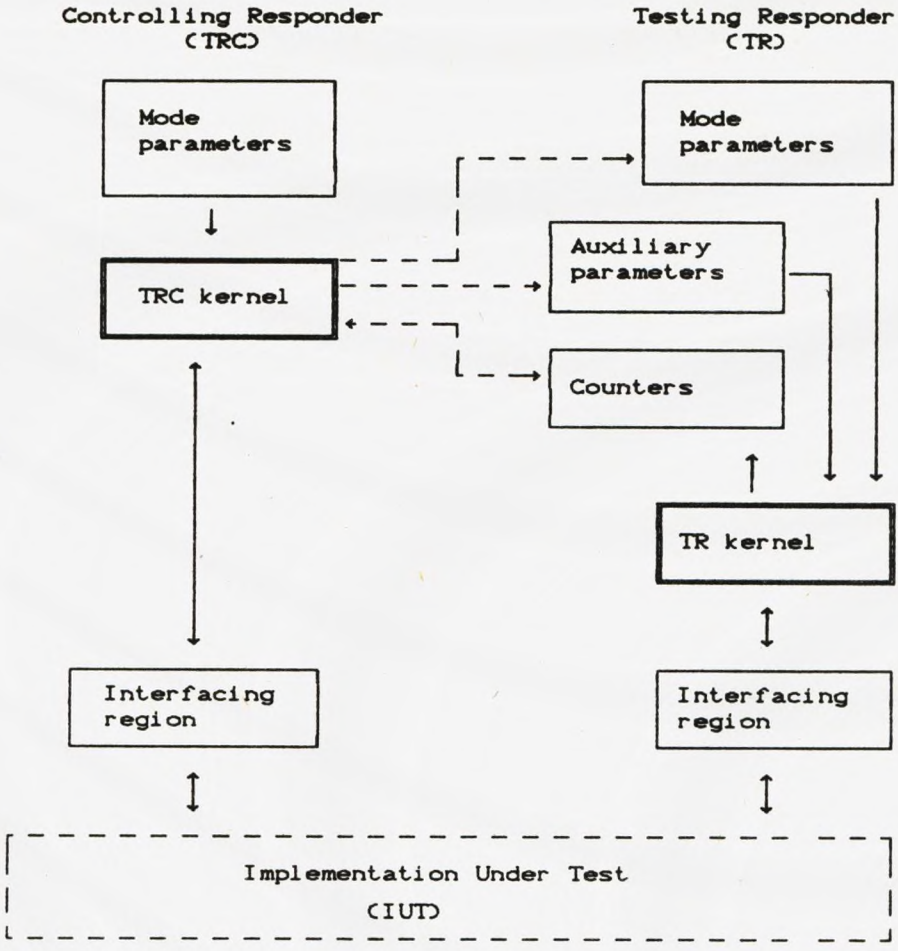


FIGURE 4
Software architecture of the Responder.

Every responder acts as a finite-state machine - it makes a specified response to every event. The mapping between events and responses is defined as the operation mode of the responder. The parameters of the TR's mode (i.e., definition of TR behaviour) as well as some auxiliary parameters are set from Driver when the actual testing responder is generated (Fig.4.).

Almost all events occur as results of incomming primitives (from IUT) and most responses consist of sending outgoing primitives (to IUT). Several counters are connected to TR and they count incoming and outgoing primitives when the test runs. There exists one "counter event" (CEV) which can be used as pseudo-random event, occuring when a specified counter reaches a specified value. The actual CEV may be defined by an appropriate TDRP command.

The TDRP has two elements: commands and responses. Commands are issued by the Driver and are embedded in Transport Data Units. Responses are issued by the Responder (its controlling part - TRC) and are also packed within T_DATA_req primitives. Most commands are used to set or change the Responder mode (its testing part TR), to set auxiliary parameters, or to clear counters. There are two special commands for generating a testing responder of "caller" or "called" type. The last group of commands consists of response requests.

Responses relay back to the Driver the information requested by the just received TDRP command. A response might consist of mode parameters, auxiliary parameters, values of counters, and parameters of last received DISCONNECT or CONNECT primitives.

6. CONCLUSIONS

It can be easily seen that the main goals listed in the introduction have been reached in the implementation of our conformance testing system .

The most attractive and original feature of this system is the way of definition of parameters values and execution structure for any test suite/group/case. The idea to determine the special file with the necessary data to perform the testing is very useful and flexible.

From our observations it is evident that the system described above should be implemented on a some much more powerful computer system. It already appears seen for very simple conformance tests. Because the implementation of the conformance testing laboratory for the Polish computer networks (in the meaning of the ISO 9646 part 4,5) is planned, experience obtained during this implementation is very useful. In the future, we would also like to implement the standard version of the TTCN language, to design and construct responders for other protocols, and to try to

implement the coordinated architecture. This latter seems to be very attractive from our point of view because it does not require the precise boundaries between two successive layers to exist (in the older, operating networks existence of the such boundaries is very rare).

REFERENCES

[1] Dubielewicz A.,Dubielewicz I.,Koleśnik K., Misiura L., Ratajczak J., Język definiowania i uruchamiania testów w sieci KASK. Rap.CO PWr. SPR 21/88. Wrocław 1988.

[2] Dubielewicz A.,Dubielewicz I., Misiura L., Ratajczak J., Model systemu testowania zgodności implementacji protokołów ze standardami. Dokumentacja programowa. Rap.CO PWr. SPR 22/89. Wrocław 1989.

[3] ISO/IEC JTC 1/SC 21 No 2041 Rev., 2nd DP 9646-1: Information Processing Systems - OSI Conformance Testing Methodology and Framework - Part 1: General Concepts.

[4] Bicz T. i inni, Mikrokomputer kompatybilny z IBM PC jako mikrohost w sieci KASK - założenia funkcjonalne. Raport SPR CO PWr, Wrocław 1988.

[5] Janczewski K., Stanisz A, Koncepcja implementacji protokołów ISO/OSI w sieciach LAN i WAN KASK na mikrokomputerach kompatybilnych z IBM PC. Raport SPR CO PWr, 1988.

COMMENTS ON TTCN IMPLEMENTATION IN THE KASK NETWORK

Iwona Dubielewicz and Krystyna Kolesnik

Computing Centre
Technical University of Wroclaw
Wroclaw, Poland

In the conformance testing system for the KASK network, conformance tests are described in the TTCN notation. So the necessity of the realization of TTCN translator has been arisen. The subset of TTCN - MP grammar (defined in DP 9646, September 1988) which was called TTCN-MP0, was realized. In the paper are given: detailed information about the assumed limitations for the TTCN - MP grammar, the internal representation of the test tree and data structures, and some remarkes on the realization of the TTCN-MP0 translator. The problems connected with the development of the current implementation in future work are also discussed.

1. INTRODUCTION

In the middle of 1987 the TTCN notation was accepted for use in the conformance system for the KASK network [6] as the language of test description. The main reason of this decision was the ability to use standard conformance test suites in our network environment, as well as the ability to express in this language some non-standard tests used for other purposes. The appropriate ISO documents were studied, and at the end of 1987 the proposition given in the ISO 9646 Annex F [1] was accepted for realization. Some modifications and corrections were made because of many mistakes noticed in this publication. It was also obvious that in our realization we should follow the changes introduced in subsequent ISO documents dealing with this subject. But we have taken into account one serious limitation: the implementation of the TTCN language should be ready in to September 1989 and as a matter of fact, the version of the language close to this proposed in the interim output Teddington/Stockholm, September 1988 [2], has been implemented .
For the time being there exists no standard test suites expressed in this language. So we have written down some simple tests cases, groups, and one test suite for the transport protocol, class 0 [4] which enables us to verify many concepts of the language as well as to check the translator realization. We have got some interesting

experiences in both of these areas.
We have assumed some limitations in TTCN implementation and in the paper detailed the information on these changes is given. We also describe the organization of the translation process and the translator realization. Our remarks on disadvantages and difficulties in implementing some of the language constructs are also presented.

2. LIMITATIONS ON TTCN-MP.

The following main constraints are assumed to the implementation of the subset of TTCN-MP called TTCN-MP0.1:

1. the allowed predefined types are limited to: Integer, Boolean, Bitstring, Octetstring, Generalstring;
2. the length of Characterstring must be explicitly given after the type name, in a form as in the example:
   Octetstring [24]
3. the form of character string constants is as follows:
   for Bitstring: '011..01'B
   for Octetstring: '35AF..5'H
   for Generalstring: 'abc ..x'
4. boolean and arithmetic expressions are limited to simple ones, i.e., to one- or two-argument expressions;
5. operators for Integer type are limited to: +,-,mod;
6. in boolean and arithmetic expressions, only named constants may be used.

The following changes have been introduced to the language:

C1. ASPs and their parameters have an internal code given after the name of ASP or its parameter e.g.:
   N_DATA_IND [9]

C2. mandatory PDU parameters are enclosed in **$PCI...$End.PCI** and optional PDU parameters are enclosed in **$VCI ... $End.VCI**;

C3. the constraints declaration part must be written prior to the behaviour part,

C4. the code and decode elementary events are enclosed in braces (it allows to distinguish them from the arithmetic expression),

C5. the elementary events list (in the event line) must end with ';'. It allows to distinguish the reference to constraint list from the arithmetic expressions (if they both are present in the event line).

The following language elements are not implemented :

N1. user-defined operators ,
N2. timer type declaration section,
N3. constraint list parameters,
N4. generic variables,
N5. tree attachment,
N6. pseudoevent 'repeat'.

These restrictions are not significant. Some of the constructions defined by the omitted syntactic rules can be easily expressed by some others (e.g., "repeat" by "goto" and counter, etc.), but they cause that the source tests text becomes longer. However, they have not got any influence on

the object code form or size.
The static semantic is extended in the following sense:
- every operation of the ASP receiving (denoted as "?") should be time-limited, i.e., in the level which contains "?" should also be "elapse" or "timeout" line event, or both,
- only one "receive" operation can be awaited in one point of control and observation on any test tree level.
Our interpretation of the TTCN constructs follows the operational semantic given in [2] with some little modifications. These modifications deal with the 'snapshot' semantics of the "timeout" and "elapse" events (due to system software environment).
In our realization the operation called "Undo" [2] is not implemented. Restoring old values of variables which have changed their values due to an assignment operation executed during the "send" line interpretation requires so much effort that the testing process is stopped in this situation.

3. TRANSLATION PROCESS

The TTCN (source) program is translated in two passes. In the first one the entire file with the source program (test suite description) is split into many smaller files, each of them containing only one test case (also written in TTCN). So the results of this phase are the libraries for test case, test steps and default steps. Every library consists of a test catalogue, which associates the test case name with the test case file.
The libraries contain tests in the source form. So, every test case must be translated as many times as it is required to be executed. This approach to organizing test libraries had to be accepted, because in other circumstances problems with saving and accessing the complex structure of the compiled test are unresolvable in our system.
The second phase of the translation is performed on the demand of the user who can indicate the test suite, test group and test case he wants to be executed. During this phase, lexical and syntactical analysis are performed, and the generation of the data structure (which constitutes the abstract test case internal form) is done. The syntactical analyzer is type of LL(1) analyzer.
Any error detection causes the erroneous line to be written down in the listing file together with the error description. The erroneous entity will be omitted in the following analysis. During the translation of the declaration part the following objects may be omitted: test part, test section, simple object, compound object, compound object parameter, compound object constraint list depending on the type and size of the erroneous entity being analysed. During translation of the dynamic part, elementary event, reference to constraint list, or a whole event line may be omitted. Translation is continued from the beginning of the next object of the same type as the erroneous one. This sort of error detection may cause that some errors remain hidden or may constitute the source of some false errors. Only errors

causing memory to be exhausted when storing the test tree will stop the translation process.
The output of the translation process is a complex data structure which constitutes the internal form of the test. This structure consists of a test header record and a test tree. The header "points to" the root of the tree. The test tree consists of nodes, each of them represents one event line. The node points to the list (ev) of elementary events which the line consists of. The following objects:
- test parameters (PAR),
- test constants (CONST),
- test variables (VAR),
- ASPs and PDUs,
- ASP/PDU parameters (co_ob_par),
- constraints lists (co_l_el)

are treated as parameters of the elementary events in the translator environment.
Every simple object of the language, i.e. constant, variable, test parameter, ASP/PDU parameter, is stored in a special (standardized for translation purposes) structure called "valrec".
The connections among the elements of the test data structure are shown on the Fig. 1.
The test tree structure obtained as the result of the translation process represents the abstract test case. Obtaining the executable test case requires the assbciation of all the formal test parameters with some actual values as well as the generation of the proper form of ASPs and PDUs is done. This is performed in the next step, during the test interpretation.
The interpreter is working on the test tree structure; it executes the subsequent event lines, and within a line, elementary events (according to their respective priorities), which the line consists of.

4. TRANSLATOR REALIZATION

The translator was written in TURBO-PASCAL on an IBM PC/AT computer.
All data structures which constitute the internal form of the test are placed in dynamic memory; only the translator data structures, necessary during the translation process, are placed in PASCAL static memory.
Data structures generated by the translator are passed to the other modules coded in C language (the interpreter as well as the whole conformance testing system is coded in the C language [5]).

5. CONCLUSIONS.

The restrictions and simplifications mentioned in the Section 2 (1-6 and C1-C4) remarkably decreased the size of the translator and the duration of its implementation (both of

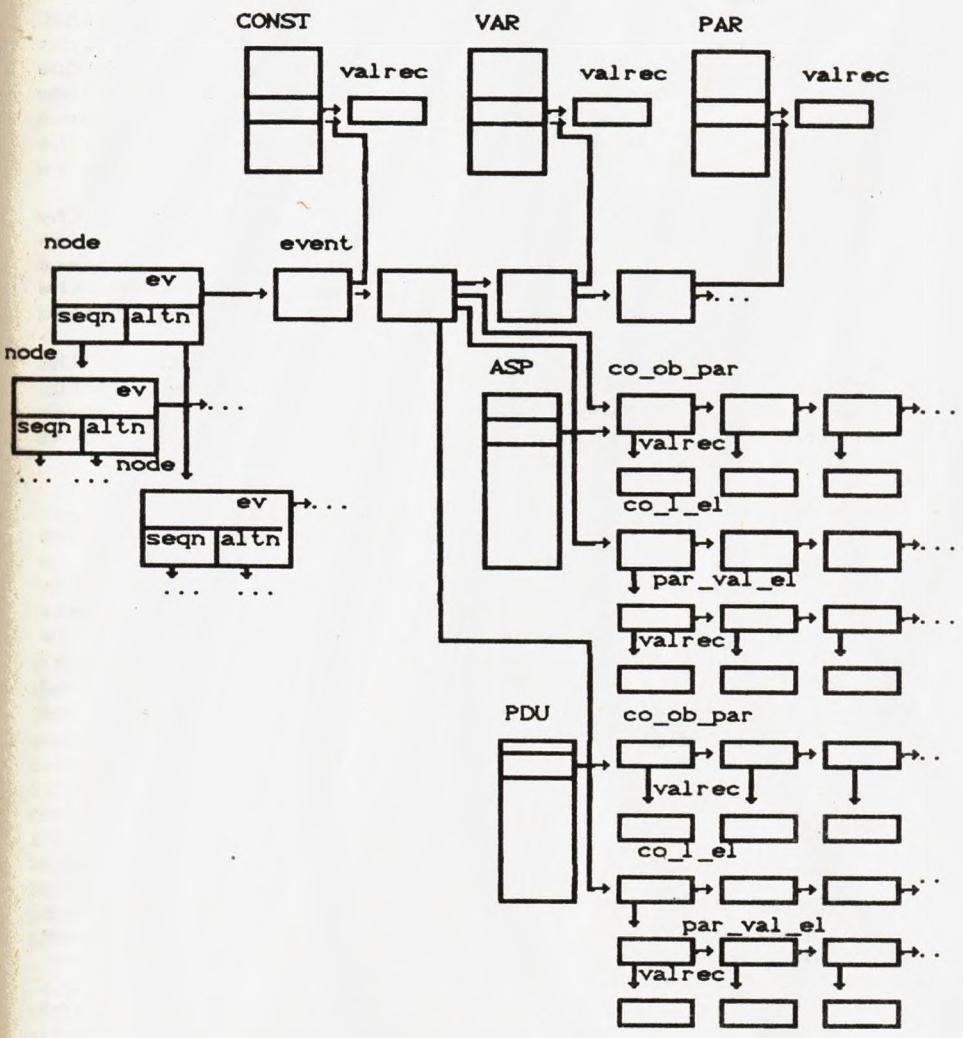


Fig 1. Structure of the data elements connections in the Translator data output.

these features were very important to us). Moreover, from our point of view, placing the constraints part before the dynamic part allows for a one-pass translator with full checking of proper use of constraints lists and values.
Some of the changes listed as C5-C6 in Section 2, have been approved to avoid the ambiguities which arose during the translation process conducted according to the LL(1) method.

After writing some simple test cases, we can also say that the solution using the libraries of ASP and PDU definitions is very rational (unfortunately, we have not realized it due to lack of time). Other propositions are unpractical because the very long declaration part is needed for these definitions and , moreover some mistakes are likely to be introduced that can in turn introduce unpredictable events in the testing process.
Other observation deals with the test tree expansion (by attaching all the subtrees and default trees). This expansion is very cumbersome because the demand on the memory increases rapidly. In our realization operating memory must store the full test tree. This can be very large for some complicated tests.
We have planned to implement the standard version of the TTCN-MP, but it will be possible on an another, more powerful computer, with the large memory (actually the conformance system and TTCN translator are placed in the IBM PC/AT compatible computer with 640 kB operating memory).

We have also another trouble. We do not collect the test cases in their internal form but as source text. So we have to translate a test case every time it is called to execution. This type of operation causes the translation process to be very time consuming. Hence the testing process can last very long for a large test suite. Since it cannot be performed in one session, there is an ability to execute only a part of the required suites or groups of test cases , and the rest can be performed subsequently [3]. It makes the testing a little effortless.

REFERENCES

[1] ISO/IEC JTC 1/SC 21 No 2041 Rev., 2nd DP 9646-2 : Information Processing Systems - OSI Conformance Testing Methodology and Framework - Part 2 : Abstract Test Suite Specification.

[2] ISO/IEC JTC 1/SC21 Information Retrieval, Transfer and Management for OSI, The Tree and Tabular Combined Notation (TTCN), Teddington/Stockholm output, 19 Sept. 1988

[3] Dubielewicz A.,Dubielewicz I.,Koleśnik K., Misiura L., Ratajczak J., Język definiowania i uruchamiania testów w sieci KASK. Rap.CO PWr. SPR 21/88. Wrocław 1988.

[4] Dubielewicz I., Komplet testów zgodności klasy 0 protokołu transportowego. Rap.CO PWr. SPR 24/88. Wrocław 1988.

[5] Dubielewicz A.,Dubielewicz I., Misiura L., Ratajczak J., Model systemu testowania zgodności implementacji protokołów ze standardami. Dokumentacja programowa. Rap.CO PWr. SPR 22/89. Wrocław 1989.

PROTOCOL SOFTWARE ENGINEERING TOOLS FOR IMPLEMENTATION OF A GENERAL PURPOSE OSI STACK

Ari P. Ahtiainen

Nokia Research Center
Espoo, Finland

Jukka Keskinen, Jari Simolin, Kalle Tarpila and Ismo Turunen

Nokia Telecommunications
Espoo, Finland

Making reliable communications software according to OSI specifications has turned out to be a formidable task. This is especially true when a full stack of protocols has to be implemented in such a fashion, which is general enough to serve a whole family of applications. In this paper we describe a computer aided software engineering environment CVOPS, which has been used to increase productivity of a large software project and the maintenability and portability of OSI communications software products. We discuss some common problems in implementing a stack of OSI protocols and point out some major design choices made for our end system environment, which is DX 200 digital switching system.

1. INTRODUCTION

It has been more than ten years since the specification and standardization of Reference Model for Open Systems Interconnection (OSI) started, yet rather few implementations have been done. One reason is that such implementation work is a formidable task. OSI communications software tends to grow large and difficult to handle. Although the telecommunication industry has a long lasting tradition for running large communications software projects, the implementation of OSI protocols has some characteristics of its own.

To begin with we give a short description of our operational environment, the DX 200 switching system. Then we discuss some common problems in implementing OSI stacks. The solutions we propose are based on a computer aided software engineering (CASE) approach. A CASE toolbox called CVOPS provides a simple paradigm of a virtual protocol machine environment, which includes tools for implementation and testing of portable computer communications software.

Our protocol architecture TOSI (Telecommunication OSI) is presented in figure 1. It has some special features as compared with "a normal stack". For example, the bottom of the stack is two-branched because

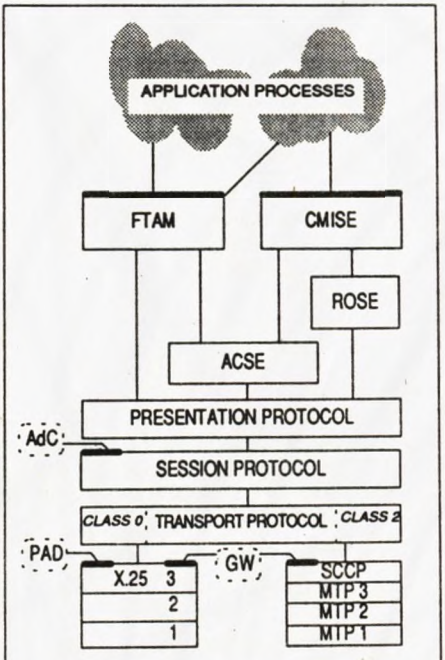


Figure 1: The DX 200 OSI architecture.

we are able to operate either on an X.25 Packet Switched Public Data Network, or on CCITT's Common Channel Signalling System No 7. We also implement a network layer routing function (GW) between X.25 and SS7. Because of the two different networks we must have two different classes, 0 and 2, of the Transport Layer. The Session and Presentation Layers are general, providing services to all kinds of Application Service Elements (ASEs) of the Application Layer. For historical reasons the Session Layer also has a direct application interface (AdC). The Application Layer consists of several ASEs. There is FTAM for file transfer and management and CMISE for network management. They both use ACSE for application association control, and CMISE makes also use of remote operations of ROSE.

DX 200 is a digital switching system with a wide range of applications, from small local exchanges to network and transit exchanges [1], [12], [16], [17]. The number of subscribers can be up to 40 000 and the maximum traffic handling capacity exceeds 400 000 BCHA. The first DX 200 was taken into operation in May 1982.

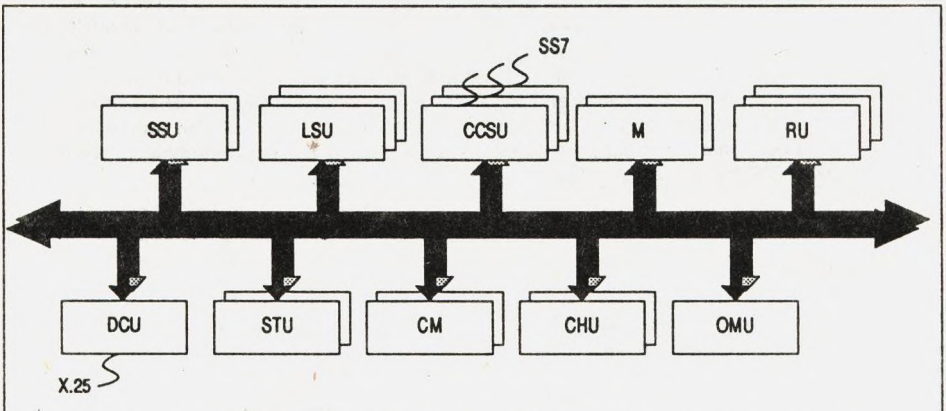


Figure 2: A Simplified architecture of the DX 200 Switching System.

The decentralized architecture of DX 200 is based on loosely coupled microcomputers (figure 2). Nowadays we use Intel 80386 microprocessors running our own operating system DMX. Each type of microcomputer is dedicated to a certain function, for example the STU (Statistical Unit) collects statistics and the CCSU (Common Channel Signalling Unit) handles Signalling System No 7. Usually there are several microcomputers of each type in order to achieve reliability and processing capability. Microcomputers communicate via a message bus. The message bus and microcomputers are made fault-tolerant by duplicating the hardware and software (in addition to some other strategies not discussed here).

Due to its highly distributed nature the DX 200 architecture is not a typical environment as an OSI end system. In the rest of the paper we present our method of integrating two very complicated issues: the seven layers of OSI and the DX 200. First, we will shortly describe some general problems in implementing OSI.

2. SOME COMMON PROBLEMS IN OSI IMPLEMENTATION

2.1 System design choices

The OSI Reference Model is a good conceptual model. It would be even better if that model could be used also as an implementation model, because it is easily understood and provides a natural way for modularization of the software. However, this is not possible because individual layers cannot

just be put together to make a complete multi-layer system - or cannot be put together without some kind of run-time environment for protocol entities.

The OSI Application Services, e.g. FTAM and CMISE, can be implemented in many ways. However, there are three basic views on how to implement a multi-layer communication architecture. Before introducing them we give some definitions:

- o An Application Service Block (ASB) implements a specific OSI Application Service, e.g. FTAM. ASB consists of required Application Service Elements (ASE), e.g. FTAM-ASE and ACSE-ASE, and all the protocol layers needed below.

- o An OSI Service Block (OSB) is a set of ASBs implementing the dedicated OSI stack, i.e. the OSI Application Services in the target system.

First we characterize the OSI Service Block according to the way, how concurrent connections are managed:

- o Application Service Block can manage several concurrent connections (A)

or,

- o Application Service Block is capable of managing only one connection at a time. To provide multiple connections processes are duplicated for each application (B).

View (A) can be called a multi-server view and view (B) a single-server view. Both views can be characterized more precisely by describing how software is divided into executable processes.

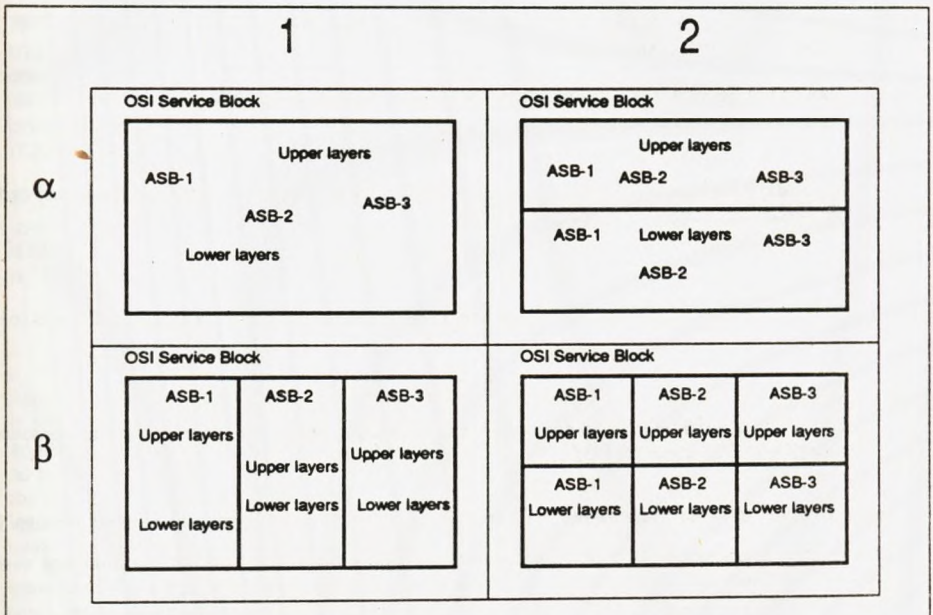


Figure 3: Different constructions of the OSI Service Block.

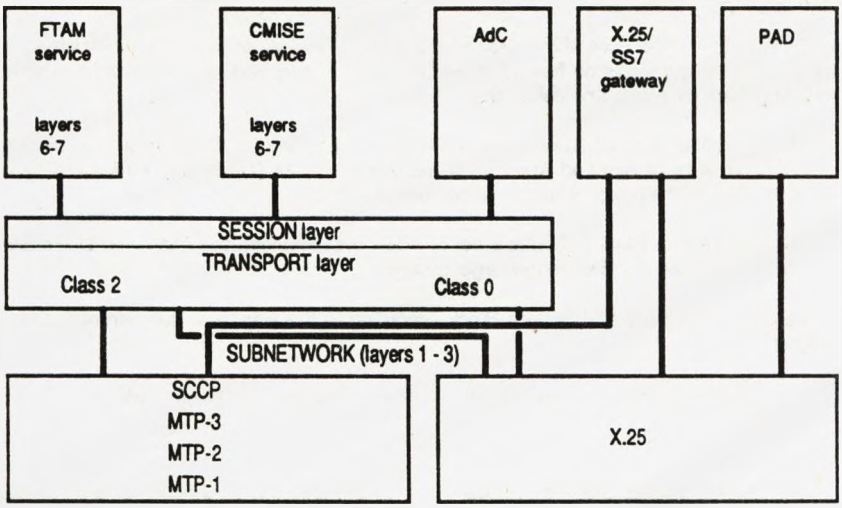


Figure 4: The implementation architecture of the DX 200 OSI.

Again, the OSI Service Block can be constructed in four ways (figure 3) deduced from the combinations of the two groups below:

- Application Service Block gathers all seven layers together into executable process (1)

or,

- Application Service Block is divided horizontally into multiple processes so there are certain processes associated with specific layers (2)

and

- OSI Service Block gathers certain parts of all ASBs together into an executable process (α).

or

- OSI Service Block is divided vertically into multiple processes so there are certain processes associated with specific ASBs (β).

Figure 4 shows that our implementation can be classified as a (A2β)-model architecture: the OSI Service Block is composed of distinct Application Service Blocks which are able to manage several concurrent connections. Each ASB is composed of at most three processes. What were the reasons behind our design choices? In fact there were only two issues that had to be considered:

- the characteristics of OSI protocol layers and the ASBs to be implemented
- impacts and possibilities of the DX 200 operational environment

Normally the OSI stack is divided in two functional parts:

- Application, Presentation and Session layers, i.e. the OSI upper layers, are closely related to the applications using OSI services mapping one connection to each application.

- Transport, Network, Link and Physical layers provides a uniform Transport service, which hides the actual network possibly composing of several different subnetworks

To achieve maximum performance, the latter part should be implemented with co-processors. For historical reasons usually only the protocols up to the network layer reside in separate hardware plug units. The X.25 Packet Level interface is a well known example, but it should be clear that soon all data communication hardware manufacturers will also support the Transport service. In any case, the OSI stack should be implemented in multiple processes according to view (2).

The Application layer is constructed from Application Service Elements. Thus, the Application layer offers a set of services provided by different ASEs. The actual relationship between an ASE and the Presentation layer is not visible outside. Therefore it has become a rather common practice to do the encoding and decoding of data units to transfer syntax and vice versa in the ASE software. This leaves only the plain Presentation layer protocol itself as a common part to all Applications. In order to make the management and configuration of OSI software easier, the OSI Service Block should be divided into smaller independent ASBs according to view (β).

The position of the Transport layer in OSI software is critical when considering whether upper layers of ASB should support multiple concurrent connections. If Transport protocol classes 2, 3 and 4 are used to multiplex several transport connections to one network connection and the Transport layer is part of the process implementing an ASB, multiple connections should be supported according to view (A). Together with choice (β) done above this means there is no way to multiplex Transport connections from distinct ASBs to the same Network connection. Therefore we have selected to make a separate process for Transport and Session layers as depicted in figure 4.

The DX 200 operational environment also had impact on the splitting of the OSI stack. By placing OSI upper layers in every microcomputer where OSI services were needed we assured maximum concurrence. To avoid software redundancy, the stack was implemented as separate Application Service Blocks which were placed in certain microcomputers only when needed. We also shared the load by splitting the stack horizontally in two parts. Session and Transport layers were placed both in DCU and CCSU to avoid overhead in message transfer.

In fact there was also a further reason to divide the stack between the Presentation and Session layers. Both Transport and Session layer standards are in a very stable state while standardization is still going on with the upper layers. By splitting the OSI stack between Presentation and Session layers we were able to achieve a stable interface for the future OSI Application Services.

2.2 Functional profiles

Although there are stable standards for nearly every individual layer, making an OSI implementation is not just "take the standard and code it". On the contrary the first thing is to refine the standards against functional requirements. Each layer consists of functional units, protocol classes, options and so on which have to be profiled. In order to be able to communicate with other OSI implementations the standards have to be interpreted correctly. In the OSI context, "correctly" means the way other implementors have interpreted the standard. However, individual layer profiling is only the first step towards a multi-layer communication software implementation.

2.3 Local implementation matters

OSI standards include many unspecified subjects (so called "local implementation matters") which are common to most of the layers, so that general solutions are needed. It is said that "local implementation matters" are the major challenges in OSI implementation. These matters cover

subjects such as flow control between layers in one stack, interfaces between layers and to the host system, and memory and buffer management.

Communication between adjacent layers can be synchronous or asynchronous. In both cases the implementation can be based on procedure calls or message passing. The imitation of asynchronous interface by means of procedure calls leads to practices based on 'upcall' mechanism [18].

Addressing of the entities within the stack is problematic if the stack is composed of several processes. The two biggest problems are the management of concurrent connections and the scheduling of protocol entities in one connection. OSI implementation problems are discussed more thoroughly in the references [2], [6] and [20].

2.4 Overall project management

The OSI stack is a complex software packet having tens of thousands of code lines. Therefore many people are involved in its production and all this must be well managed. Designers and programmers have to obey some common rules to produce maintainable and understandable software. Methods and tools to support the software life cycle are needed. The software life cycle consists of several phases: designing, specification, coding, testing and maintenance. In the OSI implementation a tool is needed in every phase. Tools shorten implementation time and some tools (e.g. code generators) not only ensure conformity of the implementation but also provide code reusability.

3. CASE TOOLS FOR OSI PROTOCOL SOFTWARE

The implementation of the DX 200 OSI stack has been developed by using a computer aided protocol software toolbox. This CVOPS toolbox contains a family of tools for the specification, implementation, testing and simulation of communication protocols in a portable C language environment. CVOPS tools support high-level specification of protocols in the form of extended finite state automaton [3] and data unit specifications based on ASN.1 language [9]. The skeleton of protocol entities are then derived automatically from these specifications.

CVOPS also has a simple language for specification of layered protocol architectures, which allows an implementation to be constructed from a library of implemented protocols. A run time support package (CVOPS kernel) includes routines for running protocol entities and is available for several well-known operating systems including System V and BSD UNIX, VAX/VMS, Intel RMX and MS-DOS. For the DX 200 project a portation to DMX operating system has been performed.

The current CVOPS toolbox consists of four independent, but integrated tools: the CVOPS Main system, CASN compiler for ASN.1, CATTS test system and system builder BUILD. These are further described below.

3.1 CVOPS Main System

CVOPS[1] - C language Virtual OPerating System - is a development environment for any distributed system built up from a set of entities communicating with each other by message passing [5], [7], [13], [14]. In communication protocols each entity or ASE is modelled as an extended finite state automaton which is called a virtual task. The kernel of the system is VOPS, which acts as a scheduler for such virtual tasks thus assuming the role of operating system. The host operating system however sees the entire CVOPS system as one process.

Virtual tasks communicate asynchronously by messages exchanged via priority based FIFO queues. Scheduling of the virtual tasks is based on the priority of the queues. VOPS takes the oldest message from the highest priority non-empty queue and invokes the body function of the receiving virtual task and passes a pointer to the message. The invoked virtual task acts according to state

[1] CVOPS was originally developed by Technical Research Center of Finland during the Technology Programme FINPRIT 1983-87 of the Technology Development Center of Finland.

transition statements given in the extended automaton specification. The task usually manipulates its internal variables and sends messages to other tasks. Execution of those statements is an atomic operation in the CVOPS process. When the handling of the message has been completed, the control returns from the task body to VOPS. When all queues are exhausted, VOPS is suspended. Input from the environment (e.g. user keyboard, network devices, adjacent processes) or timer expirations brings the VOPS up again.

A typical structure of a CVOPS system is presented in figure 5. It shows how two protocol layers, (N)-layer and (N-1)-layer, can be implemented. One virtual task (VTASK) handles one connection at a time. Concurrent connections are handled by duplicated virtual tasks. In this case an enity virtual task (ENTITY-TASK) is needed in every layer. It chooses a free virtual task in connection establishment and it solves local addressing problems by serving as a message router for every incoming service primitive. The connection tasks take care of the actual protocol and message parameters, independently of other connections.

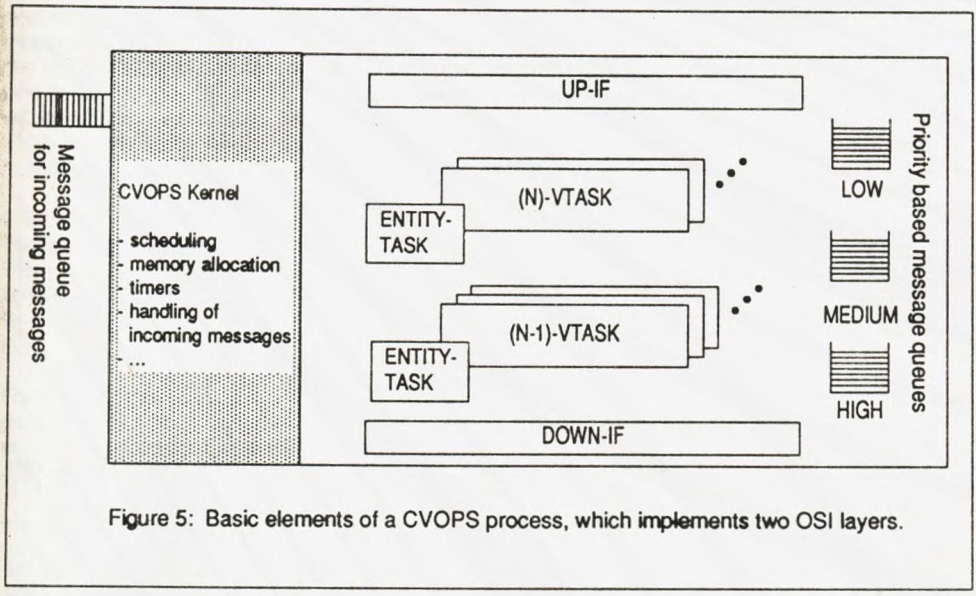


Figure 5: Basic elements of a CVOPS process, which implements two OSI layers.

Special types of virtual tasks called (device) drivers (modules UP-IF and DOWN-IF in figure 5) are needed to implement the interface between the host operating system and the CVOPS process. The CVOPS process needs a driver for every external source of events. For standard types of interfaces like BSD sockets and NetBIOS some CVOPS drivers are available, but in most cases a driver must be tailored according to system specific features.

The purpose of the driver is to encapsulate system specific features, such as inter-process flow control, from the protocol entities. Although the neighboring virtual task is no longer an adjacent layer entity, but a special purpose device driver, one can use an unmodified layer service interface.

This simple paradigm gives the protocol implementor an impression of asynchronously communicating protocol entities yet avoiding some of the overhead evident when implementing each entity as a single (real) process. As compared to systems making direct use of 'upcalls' the CVOPS environment is more friendly, since there is seldom need to follow the low level control flow, which nevertheless is based on upcall mechanism.

The structure of a single virtual task is illustrated by figure 6. Normally a virtual task has three interfaces; for the layer above (UPParameters) and below (DOWNParameters) and one which is the

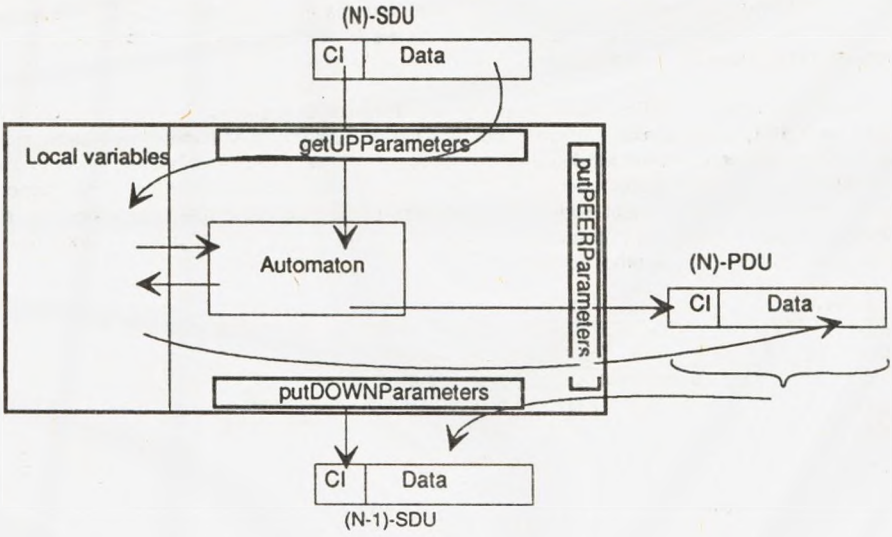


Figure 6: General structure of a virtual task.

virtual interface to the peer entity (PEERParameters). For every interface there are two parameter functions: a "get*Parameter" function is used for reading incoming parameters and a "put*Parameters" for writing outgoing parameters.

Figure 6 also shows the information flow when a service primitive (say (N)-SDU) comes from the upper layer. It is read by the getUPParameters function, which stores necessary data to local variables of the task. The Control Information (CI) of the SDU is given to the automaton which then manipulates local variables. It constructs the associated (N)-PDU by determining the type of the PDU (CI) and some other fields, and then calls putPEERParameters function to do the actual encoding. After the encoding of the PDU, the function putDOWNParameters is called, which makes the associated (N-1)-SDU. This is the way in which service primitives and protocol data units move from one layer to another. However, there is usually no need to actually copy any user data, since the virtual tasks share a common address space. Each virtual task only adds (or removes in the case of upcoming message) its own information in front of the previous user data.

After specifying the interface events and state automaton, the implementation of a virtual task is completed manually by writing the set of parameter functions (UP, DOWN and PEER) and initialization functions for the virtual task. Data units attached to the events are described using C language structures or ASN.1 descriptions. In case of layer entities using ASN.1 structured data the results of the ASN.1 compiler (see below) is used to implement most of the parameter functions.

This step by step mechanism is repeated in a similar manner regardless of protocol specific features. Thus the overall software engineering may be based on this scheme. In fact this process can be further automatized as discussed below in 3.4. The reusability of virtual task implementations from one protocol stack to another - assuming that the interfaces remain unchanged - is also achieved.

3.2 CASN - Compiler for ASN.1

Abstract Syntax Notation One (ASN.1) is a high-level language standardized by ISO and CCITT for defining data types. ASN.1 has become a common publication language for all the Application layer protocol specifications. Its usage is also being extended to other protocol families like MAP/TOP, GSM and TCP/IP.

The use of ASN.1 in specification and implementation of application data units is two-fold. First, the syntactic structure of all data units is described in this abstract notation, resulting in the abstract syntax of the protocol [9]. Secondly, any data unit specified in ASN.1 has a machine-independent bit-level representation. This machine-independent transfer syntax can be derived by applying Basic Encoding Rules which have also been standardized for ASN.1 [10].

The CVOPS toolbox includes a Compiler for ASN.1, called CASN[2] [4], which is used in both steps needed for implementation of data unit presentation. First the data units attached to CVOPS messages for all interfaces of a virtual task are described by using ASN.1. CASN expects as input one or more ASN.1 modules for each of the interfaces of a virtual task and translates the ASN.1 data type definitions into standard C data type definitions. CASN also assigns variables for storing ASN.1 values for each virtual task. The values of these ASN.1 variables are represented in local syntax, which is determined by the C compiler. This approach is diiferent from some previously reported methods [2], [15], where ASN.1 values are stored locally in tree structures.

CASN also generates C language functions for encoding and decoding values of these variables. These functions have access to run-time message buffers, where protocol data units are represented in the standard transfer syntax. This conversion is part of the responsibilities of the Presentation layer even if the encoding and decoding functions are accessed via a virtual task of the Application layer. All the low-level conversion routines and message buffer management routines are included in the CVOPS run time system.

For testing and probing facilities built in the CVOPS system another set of functions may be generated by CASN Compiler. These test functions are used to fill and print ASN.1 structured data units during the testing phase. It has turned out that the amount of work spent in testing a protocol entity may be decreased by this automatic generation of test benches.

In the implementation of such Application Service Elements like FTAM and CMISE, the use of CASN compiler had significant impact on the time and total effort needed for writing the CVOPS code. Together with the CVOPS code generator for protocol automata it was possible to produce automatically as much as 50 to 70 percent of the code lines for these virtual tasks. CASN compiler also provides some options which allow quick prototyping of a protocol entitity, which also turned out to be a useful feature during the development process.

On the other hand the use of automatically generated data type definitions leads easily to many conversions between external message structures and CVOPS parameter blocks. These conversions are performed by the CVOPS device driver tasks, which are also dependent on the application environment. For the interface between network management application and CMISE we were able to use ASN.1 structured messages thus avoiding the need for adaptive conversions.

CASN Compiler itself has been developed in C language. A language processor tool MIRA[3] has been used to implement the parser for the ASN.1 language. Current version of CASN is running both in UNIX and MS-DOS.

---

[2] CASN Compiler for ASN.1 has been developed by Nokia Research Center

[3] MIRA Program has been developed by Expert Software Systems N.V., Belgium.

3.3 Computer aided test system CATTS

CATTS[4] (Computer Aided TOSI Test System) is a conformance testing system, which is an implementation of the ISO Local Test Method [19]. It is meant for protocol developers to be used during protocol implementation phase, although it can be used for normal conformance testing purposes, too.

CATTS consists of a simple language in which all test cases are written. This aTDL (abstract Test Description Language) is based on TTCN [11]. In a aTDL test suite developers can describe all inputs and expected outputs and their parameters. CATTS supports test suite structuring and management so that every individual test case or many test cases as a whole can be easily rerun. CATTS also includes a test result analyzer which automatically evaluates the test result judging the test to be passed, failed or inconclusively terminated. CATTS expects that implementation under test has been developed by using CVOPS and CASN.

CATTS consists of several software modules. The main modules are implemented with software tools LEX and YACC. CATTS can be used in MS-DOS and UNIX environments.

3.4 System Generator BUILD

As described in 3.1 above each virtual task and hence an entire CVOPS process consists of a few basic components (e.g. put*Parameters and get*Parameters functions), which are hand-coded in a similar way from one task to another. This observation has led to the development of BUILD[5] which is an application generator especially tailored for software engineers working with the CVOPS toolbox.

BUILD takes as an input high-level specification files for a CVOPS process: system architecture definition and definitions for interface events, automata and internal items of virtual tasks. Having analyzed this input, BUILD can generate C source code files for the implementation of virtual tasks and Makefiles for producing an executable image of the system.

So far BUILD has been successfully applied for building simulators of distributed systems, e.g. the DX 200 switching system itself. The work to extend BUILD with capabilities needed for protocol software development and providing this tool with a graphic user interface has started and will result in further improvements to the CVOPS toolbox.

4. CONCLUSION

Implementation of a distributed general purpose OSI stack is a formidable task. As in other big software projects, software tools for the whole protocol life cycle are needed. However, implementing an OSI system also presents other difficulties. The specifications of OSI protocols are insufficient and ambiguous so they must be interpreted in a proper way to be able to communicate with other manufacturers' stacks. Although the OSI Reference Model is a well designed conceptual model, it cannot be used as an implementation model without adaptation. To construct a multi-layer system from individual OSI layers a separate run-time environment is needed.

In this paper we have pointed out the major difficulties in implementing the OSI system and presented a CVOPS toolbox which supports a multi-layer communication system design, coding, testing and maintenance. It also offers the needed run-time support for individual protocol layers; it solves the internal flow control, addressing and memory management, the scheduling of individual protocol entities and also supports concurrent connections.

---

[4] CATTS system has been developed by Nokia Telecommunications.

[5] BUILD system has been developed by Nokia Research Center.

REFERENCES

[1] Ahava, H., Ilmarinen, J.; Switching System Applications for Rural Networks, pages 275-289 in Part 2 of 5th World Telecommunication Forum, Geneva 22-27 October 1987.

[2] Bochmann, G. v., Deslaurier, M. and Bessette, S., Application layer protocol testing and ASN.1 support tools, Proc. of IEEE GLOBECOM Conf., Houston, Dec 1986, pp. 767-771.

[3] Bochmann, G.v., Usage of Protocol Development Tools: The results of a survey, in: Protocol Specification, Testing and Verification VII, North-Holland, 1987.

[4] How to use CASN Compiler for implementation of a virtual task in CVOPS, vers. 1.0, Technical Report, Nokia Research Center, 1990.

[5] CVOPS User's Guide, Technical Report, Telecommunications Laboratory, Technical Research Center of Finland, 1990.

[6] Gantenbein, D., Hauser, R., Mumprecht, E.; Implementation of the OSI Transport Service in a Heterogeneous Environment, pages 215-241 in HECTOR, vol II: Basic Projects, Springer-Verlag, 1988.

[7] Harju, J., Karila, A., Kuittinen, J., Malka, J.; CVOPS - A Tool for the Implementation and Testing of Computer Communications Software, Technical Report, Telecommunications Laboratory, Technical Research Center of Finland, 1986.

[8] ISO, Information Processing Systems - Open Systems Interconnection. Basic Reference Model, ISO/IS 7498, May 1983.

[9] ISO, Information Processing Systems - Open Systems Interconnection - Specification of Abstract Syntax Notation One (ASN.1), ISO/IS 8824:1987 (E) with DAD1.

[10] ISO, Information Processing Systems - Open Systems Interconnection - Basic Encoding Rules for Abstract Syntax Notation One (ASN.1), ISO/IS 8825, December 1987.

[11] ISO, Information Processing Systems - Open Systems Interconnection - OSI Conformance Testing Methodology and Framework, ISO 9646, Part 3: The Tree and Tabular Combined Notation (TTCN).

[12] Jokiaho, T.; An Operating System for SDL. CCITT - SDL Implementor's Forum, Florence, 6-9 September 1982.

[13] Karila, A.; Portable Protocol Development and Run-time Environment, Lisenciate Thesis, Department of Technical Physics, Helsinki University of Technology, 1986.

[14] Karila, A.; VOPS - a Portable Protocol Development and Implementation Environment, in Proc. of IFIP - International Conference on Data Communications, Theory and Practise, Rönneby Brunn, Sweden, 1986.

[15] Pope, A. R., Encoding CCITT X.409 presentation transfer syntax, ACM Comp. Comm. Review 14, 4 (Oct 84), pp. 4-10.

[16] Purho, T., Aflatuni, Z., Soitinaho J.; Operational Reliability of the DX 200 Switching System, in Proc. Annual Reliability and Maintainability Symposium, 1987.

[17] Pöyhönen, P., Lahdelma, R.; Multi-model Reasoning in DX 200 Digital Switching System Diagnosis, in International Switching Symposium, Phoenix, Arizona USA, March 15-20, 1987.

[18] Rose, M.T., The Open Book: A Practical Perspective on Open Systems Interconnection, Prentice-Hall, 1990.

[19] Sarikaya, B., Conformance Testing: Architectures and Test Sequences, Computer Networks and ISDN Systems 17 (1989). pp. 111-126.

[20] Svobodova, L.; Implementing OSI Systems. Research Report #62131, IBM Research Division, Zurich Research Laboratory, 1988.

The Pre-Implementation and Verification of LOTOS

E. Najm, J. Queiroz*, A. Serhrouchni+

INRIA
Domaine de Voluceau - Rocquencourt
BP 105 78153 Le Chesnay CEDEX
FRANCE

Email: najm@inria.inria.fr
queiroz@mozart.inria.fr
ahmed@wagner.inria.fr

The present paper deals with the formal translation of LOTOS into an implementation oriented model. An approach named Pre-implementation, which is based on a model named PRIMOL, is described. A subset of LOTOS is used to illustrate this approach. PRIMOL is a model of Hierarchically Cooperating Extended Finite State Machines on which LOTOS specifications can be translated. PRIMOL is endowed with a transformation and verification framework.

1. Introduction

The Process Algebras approach for the description of concurrent and distributed systems has emerged in the early 80's [Mil80,Bou84,Hoa85,BeK85] and has gained growing interest in the protocol research and designers community. LOTOS [IS8807,BoB87,NaB86,Bri88] is one such approach which has reached the status of International Standard.

Process Algebras have a common syntactic and semantical framework: a) processes are denoted as terms of an algebra, i.e., expressions built using a predefined set of "behaviour" constants and operators; b) the semantics of terms is given by means of their translation into labelled transition systems, this translation being usually defined by associating a set of axioms and/or inference rules to each of the "behaviour" constants and operators [Plo81].

*+ Universite Pierre et Marie Curie
Laboratoire MASI
4, Place Jussieu
75252 PARIS Cedex 05 - FRANCE

* Universidade Federal de Pernambuco
Departamento de Informatica
Av. Prof. Luiz Freire, S/N
50.000 RECIFE / PE - BRAZIL

The conciseness and power of the syntax together with the self-contained mathematical semantical model make Process Algebras very appropriate for the formal and implementation independent description of concurrent and distributed systems. Thus Process Algebras constitute a solid starting point for the development of conformance and implementation theories [dNH84,BSS87] which aim at the derivation of verified implementations. In the present paper we only address a first formal step of the challenging problem of derivation of implementations from LOTOS specifications, i.e., the mapping of LOTOS objects onto real/physical objects. Although our discussion is applied to LOTOS, it is not specific to this language and may be used for Process Algebras in general.

Implementation is a wide problem which covers various aspects such as parallel computing, operating systems, user-interface, testing and programming methodologies. In the present work we are only concerned with an approach, called Pre-Implementation, the aim of which is to formally bring LOTOS specifications closer to implementation.

There is an apparent contradiction in the idea of mapping formally defined concepts on real systems which are, by nature, not completely formalised (at least for the time being). In fact, specification-to-implementation is a complex trajectory which may greatly benefit from the use of formally defined intermediate steps before reaching the point where the mapping on real objects should take place.

Pre-Implementation is based on a language/model called PRIMOL, the elements of which are called pre-implementations. The main characteristic of PRIMOL is to be a formal model which is based on what intuitively is considered to be close to the implementation: Cooperating Extended Finite State Machines. Another important characteristic of PRIMOL is to have a nice transformational framework. The aim of the Pre-implementation phase is to first translate LOTOS specifications into pre-implementations and then, using the PRIMOL to PRIMOL transformation framework, to interactively transform these pre-implementations in view of their verification and possibly of their actual implementation.

2. Existing Implementation Approaches for Process Algebras /LOTOS

The problem of formally deriving implementations from high-level specifications has been variously addressed in the recent past. In [Mil84] and [Par85], the authors have defined a translation mechanism from pure CCS regular processes into charts (a particular kind of transitions diagrams). An adaptation of this idea to LOTOS, including value-passing and multiway synchronization, can be found in [Kar88]. [Ver86] also contains a mechanism for translating a finite subcalculus of Meije

[Bou84] into minimal finite state automata and [GaN89] describes a two steps translation of a subset of LOTOS with value-passing: first into an extended petri net model and then into finite state automata.

In [BSS87] possible different definitions to the concept of implementation, as related to specification, are developed. The authors have named it "reduction" when it indicates partial (or total) resolution of non-determinism in the specification, "extension" when it means the addition of functionalities not included in the specification, and "refinement" when it denotes transformations on the inner structure of the specification without altering its semantics. The tool described in [QPF88] uses the concept of refinement: LOTOS specifications are transformed using the expansion theorems defined in [IS8807] allowing for the replacement of parallel processes by sequential ones. An extension to the language was needed in order to achieve these transformations.

The LOTOS Implementation Workbench presented in [MaM88] describes a translator of LOTOS specifications with "C annotations" into C code runnable on UNIX. This translator provides an interactive framework for the resolution of non-determinism. In this tool, LOTOS is used for synchronization and control whereas the "C annotations", that are added on LOTOS actions, provide means to act upon the UNIX environment.

LOTOS interpreters/simulators [BFL86,Eij89,Tre89] are another interesting implementation approach. They exercise the semantics of the language by reading the text of the input LOTOS specification (or an abstract syntax tree representation of it) at each interpretation step. These tools are suitable for early debugging since the interpretation is closely based on the source text. Ecrins [MaS87] is an interpretor of user definable Process Algebras which also provides mechanisms to prove laws of strong equivalence between terms.

3. The Pre-implementation Approach

For a given Process Algebra, considered as a specification technique, Pre-Implementation merely consists in:

- an "appropriate" model called Pre-Implementation model the elements of which are called pre-implementations,
- a formal, semantics preserving, mapping from specifications into pre-implementations (a refinement in the sense of [BSS87]).

By "appropriate" it is meant that the Pre-Implementation model has the following properties:

a) the Pre-Implementation model should share the same semantical domain as the Process Algebra,

b) the semantics preserving mapping from specifications into pre-implementations should be simple and straightforward,
c) the Pre-Implementation model should be as close as possible to what intuitively is considered to be implementations on real distributed systems. The following is a list of implementation concepts that would with advantage be formally reflected in the Pre-Implementation model:
   - a distributed system is a set of cooperating sequential processes,
   - sequential processes are modelled by "Extended Finite State Machines",
   - sequential processes use an imperative-like language to operate on data stored in some memory,
   - communication between processes is modelled by a controlled access to a shared storage,
   - the configuration of a set of cooperating processes is controlled by a higher level process.
d) the Pre-Implementation model should be endowed with a simple and formal transformational mechanism allowing for user-driven transformations to be applied on pre-implementations in view of their verification and of their actual mapping on distributed systems (this last point is still an open research topic). This property is of paramount importance. It implies that the Pre-Implementation model is flexible so that to allow for different implementations to be derived from a given pre-implementation depending on the real implementation environment.

The design of our Pre-Implementation model has been inspired by results of [Bou87] and [dSi85]. [Bou87] presents a language using two levels of semantics: one which is symbolic, i.e., a description of a mechanism to calculate what "imperative instructions" may apply in a given state; a second level which is operational, i.e., it describes how a system evolves (how the data is modified) due to the operations of the "imperative instructions" that were calculated in the first step. [dSi85] defines a mechanism for translating Process Algebras into the SCCS-Meije calculus [Mil83,Bou84]. This translation mechanism is based on the association of a SCCS-Meije finite automaton to each of the Process Algebra operators. Then the translation of, say, *op(B1,B2)* is a SCCS-Meije term which is the parallel composition (with appropriate restrictions) of the automaton associated to *op* with the result of the translation of *B1* and *B2* . Roughly speaking, this SCCS-Meije term denotes the "synchronization" of the behaviours of *B1* and *B2* by the automaton associated to *op*. It is an adaptation of this translation mechanism that has been used for the design of Pre-Implementation models.

4. Presentation of Primol

PRIMOL is a model of Hierarchical, Cooperating, Extended Finite State Machines (EFSMs). More precisely, a PRIMOL pre-implementation is a

hierarchy, the nodes of which are a special kind of EFSMs sharing a collection of data type definitions and a set of typed value-identifiers. Each node in the hierarchy is responsible for the "synchronization" and "control" of its subnodes. These concepts are formally presented in sections 4.1 and 4.2 which are devoted to the syntax and semantics.

The hierarchy in PRIMOL is inherited from the algebraic structure of LOTOS behaviour expressions. When translating a LOTOS specification into a pre-implementation, each PRIMOL node is roughly the result of the mapping of a LOTOS operator from the source specification. The LOTOS to PRIMOL translation mechanism and the syntactical condition of its applicability are briefly presented in section 4.3.

An advantage of the hierarchical structure is that it sets an easy and simple framework for PRIMOL to PRIMOL, semantics preserving, transformations: a transformation is the product of the application of a set of elementary transformations and an an elementary PRIMOL to PRIMOL transformation is simply defined to be as the result of merging of a node and one of its subnodes into a new node. Section 4.4 describes this transformation mechanism and presents its properties and its advantages.

4.1. The syntax of PRIMOL

PRIMOL is an intermediate, i.e., non end user oriented, representation. The two main requirements that led to its design are: a) the ease of mapping of LOTOS specifications and b) the definition of a simple framework for syntactical, semantics preserving, transformations. For convenience, it has been endowed with a concrete syntax which can be found below. This concrete syntax uses many of the syntactical constructs of LOTOS (Meta-syntactical conventions: terminals are in **bold** type face or between " ", | is an alternative symbol, [...] denotes 0 or one instance, { ... } denotes 0 or more instances, { ... }+ denotes 1 or more instances and underscored non-terminals are defined in LOTOS [IS8807]):

```
pre-implementation =  PRE-IMPLEMENTATION specification-identifier
                                  formal-parameter-list data-type-definitions
                            { node }+
                      END-PRE-IMPLEMENTATION

node =        NODE          node-identifier
        [     SUBNODES      node-identifier { "," node-identifier } ]
              STATE-VARIABLES "[" state-identifier { "," state-identifier } "]"
              STATE-INITIALIZE "[" state-expression {"," state-expression } "]"
        [     SUBNODES-INITIALIZE node-identifier { "," node-identifier } ]
        {     FROM      state-condition-expression
              DO        action-expression
              TO        next-state }
              ENDNODE
```

```
state-condition-expression = state-expression { "," state-expression }
action-expression = ground-action | action-from-subnode | seq-actions | sync-actions
ground-action =   "(" [   PREAMBULE preamble-expression ]
                      LABEL label-identifier VIS visibility
                  [   OFFERS offers-expression ]
                  [   PROVIDED premiss ]
                  [   SUB-INIT node-identifier { "," node-identifier } ] ")"
action-from-subnode =     action "(" node-identifier ")"
seq-actions =     pre-action action-expression
sync-actions =    SYNC action-expression { "//" action-expression }+
                  RELABEL-TO label-identifier ENDSYNC
pre-action =      label-pre-action | values-pre-action | sub-init-expression
label-pre-action =     IF-IS label-identifier | NOT-IN"{" labels "}"
                  |    RELABEL-TO label-identifier
values-pre-action =    LET identifier-equations | PROVIDED premiss
sub-init-expression =  SUB-INIT node-identifier { "," node-identifier }
preamble-expression =     atom-preamble | seq-preamble | par-preamble
atom-preamble =        values-pre-action
seq-preamble =         atom-preamble seq-preamble
par-preamble =         "(" preamble-expression { "//" preamble-expression }+ ")"
labels =               label-identifier { "," label-identifier } | i
offers-expression =    offer-list { "//" offer-list }
offer-list =           experiment-offer { experiment-offer }
next-state-expression = state-expression { "," state-expression }
visibility =           i | e
state-expression =     state-identifier "=" state-value
state-identifier =     node-identifier
label-identifier =     gate-identifier
```

The syntax of a node is quite similar to the classical notation of Extended Finite State Machines. The originality resides in the action-expressions. These make use of action-variables (these are called action-from-subnode) and their syntax is: **action**(node-identifier), where node-identifier is the name of a sub-node. It would be too lengthy to comment all the aspects of the syntax here. Instead we will present some examples in the next subsection which is devoted to the semantics.

4.2. The semantics of PRIMOL

4.2.1. Informal introduction

An execution step (a global transition) of a pre-implementation involves the cooperation of a sub-tree of nodes rooted at the highest node. Each of the nodes of the sub-tree is involved with one of its transitions and synchronizes the involvement of those of its subnodes that are referenced in the action-expression of this transition (more precisly, subnodes are referenced in action variables). Perhaps the best informal description of this mechanism may be given through examples. The three examples given below present simple Lotos behaviour-expressions together with their translation into PRIMOL. Facing the first PRIMOL text, a (natural) graphical representation of it is also given.

Example 1 BE0 = BE1 [> BE2

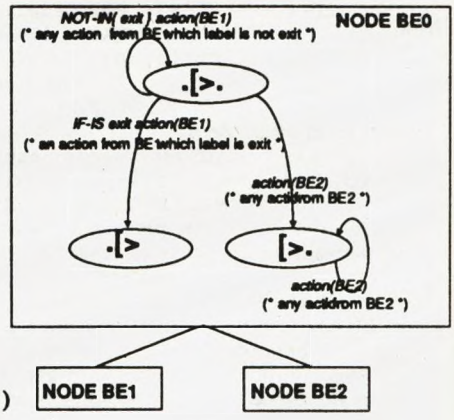
```
NODE    BE0
    SUBNODES    BE1, BE2
    STATE-VARIABLES     [BE0]
    STATE-INITIALIZE    [BE0 = .[>. ]
    SUBNODES-INITIALIZE BE1, BE2

        FROM    [BE0 = .[>. ]
        DO      action(BE2)
        TO      [BE0 = [>. ]

        FROM    [BE0 = [>. ]
        DO      action(BE2)
        TO      [BE0 = [>. ]

        FROM    [BE0 = .[>. ]
        DO      IF-IS exit action(BE1)
        TO      [BE0 = .[> ]

        FROM    [BE0 = .[>. ]
        DO      NOT-IN{exitt} action(BE1)
        TO      [BE0 = .[>. ]
    ENDNODE
```



Example 2

BE0 = g ?x:s !E [f(x)] ; BE1

```
NODE    BE0
    SUBNODES    BE1
    STATE-VARIABLES     [BE0]
    STATE-INITITIALIZE  [BE0 = .; ]

        FROM    [BE0 = .; ]
        DO (    LABEL g VIS e
                OFFERS ?x:s !E
                PROVIDED f(x)
                SUB-INIT BE1    )
        TO      [BE0 = ;. ]

        FROM    [BE0 = ;. ]
        DO      action(BE1)
        TO      [BE0 = ;. ]
ENDNODE
```

Example 3

BE0 = let (x : s = vex) in BE1

```
NODE    BE0
    SUBNODE     BE1
    STATE-VARIABLES     [BE0]
    STATE-INITIALIZE    [BE0 = .LET ]
    SUBNODES-INITIALIZE BE1

    FROM    [BE0 = .LET ]
    DO      LET x:s = vex action(BE1)
    TO      [BE0 = LET. ]

    FROM    [BE0 = LET. ]
    DO      action(BE1)
    TO      [BE0 = LET. ]
ENDNODE
```

4.2.2. Notations and definitions

Syntactical notations

for a given node *nd*, a transition *t*, an action-expression *ae*, a preambule-expression *pre* and an offer-expression *oe*:

- *nd-id(nd)* is the node-identifier of *nd*,
- ^*nd* is the tree rooted at the node *nd*,
- for *ND* a node or a tree:
  - *state-identifiers(ND)* is the set of *state-identifiers* occuring in *ND*,
  - *value-identifiers(ND)* is the set of *value-identifiers* occuring in *ND*,

- *from(t)* is the *state-condition-expression* of *t*, *action(t)* is the *action-expression* of *t*, *next(t)* is the *next-state-expression* of *t*,
- a transition *t* will be also denoted by a triplet (*from(t)*, *action(t)*, *next(t)*),
- *ground-action*(s) will be sometimes denoted by their quintuplets: (*pre* , (*g*,*vis*) , *bex* , (*si*)) where *pre* is a preambule-expression, *g* is a *label-identifier*, *vis* is a *visibility* (i or e), *bex* is a premiss, and *si* is a list of *node-identifiers* (the sub-init part of the *ground-action*).
- *AV(t)* (resp. *AV(ae)*) is the set of action-variables occuring in *t* (resp. in *ae*),
- *parameters(pve)* is the set of (typed) *value-identifiers x* occuring in *pve* in the form: *let x=any s*,
- *Input(oe)* is the set of (typed) *value-identifiers x* occuring in *oe* in the form: *?x:s*
- for *txt*, $old_1,\ldots,old_n$, $new_1,\ldots,new_n$ pieces of text, $txt[new_n/old_n,\ldots,new_1/old_1]$ is a new piece of text obtained from *txt* by replacing all occurences of old_i by new_i.

Partial functions

for f:D $\longrightarrow$ D' and g:D $\longrightarrow$ D' partial functions

- $f(x) = ?$ denotes the fact that f is not defined for x
- $f \oplus g$ is the partial function defined by:

  $(f \oplus g)(x) = f(x)$ *when* $f(x) \neq ?$ and $g(x) = ?$

  $(f \oplus g)(x) = g(x)$ when $g(x) \neq ?$ and $f(x) = ?$

  $(f \oplus g)(x) = ?$ when either $(f(x) = g(x) = ?)$

  or $(f(x) \neq ?$ *and* $g(x) \neq ?)$
- $f \backslash g$ is the partial function defined by:

  $(f \backslash g)(x) =$ if $g(x) \neq ?$ then $g(x)$

  if $g(x) = ?$ then $f(x)$
- $f \Rightarrow g$ is a predicate which is true when $f \backslash g = f$

Semantical domains and functions

for a given pre-implementation *pre-imp* and a given node *nd* of *pre-imp*

- *Values* denote the union of the value-domains of all sorts defined in data-type-declarations of *pre-imp*
- *State-values* denotes the set of all *State-value(s)* occuring in *pre-imp*
- for *ND* a node or a tree, a control-state *S* of *ND* is a partial function:

  S:state-identifiers(ND) $\longrightarrow$ *State-values*
- for a tree ^*nd*, a variable-state V of ^*nd* is a partial function:

  V:value-identifiers(^nd) $\longrightarrow$ *Values*
- for *vex* a value-expression, *val(vex at V)* is the value of *vex* at variable-state *V*
- if *sex* is a *state-condition-expression*, or a *next-state-expression* or a list of *node-identifiers*, then

  sex denotes the partial-function:

  sex : *state-identifiers* $\longrightarrow$ *State-values*

  wich is naturally associated to *sex*

(in the case where *sex* is a list of *node-identifiers*, $\overline{sex}$ is the function which is the union of the local-initialisations of the nodes that are in the list and recursively of the local-initialisations of the subnodes that are initialised in the nodes of the list)

About the semantics of *preambule-expression(s)*
preambule-expression(s) constitute the "imperative" part of PRIMOL. Their semantics is such that for any *preambule-expression pre*, one can associate a parametrized boolean-guard, denoted *pre-guard*, and a parametrized partial function, denoted *pre-func*, both defined on *variable-states* and both parametrized by the set of assignments *(parameters(pve) ⟶ Values)*. Let *U* be such an assignment, the domain and range of *pre-guard* and *pre-func* are given below:

pre-guard(pre,U)(.) :
(value-identifiers ⟶ Values) ⟶ *{true, false}*
pre-func(pre,U)(.) :
(value-identifiers ⟶ Values) ⟶ *(value-identifiers ⟶ Values)*

Moreover, *pre-guard* and *pre-func* satisfay the following condition:
if *pre-guard(pre,U)(V) = true* then *pre-func(pre,U)(V) ≠ ?*

Notation
for $O = O_1,...,O_n$ an *offer-list*,
pre a *preambule-expression* ,
V a *variable-state*, and
U an assignment *U:*
parameters(pve) ⟶ *Values* verifying *pre-guard(pve,U)(V)=true*,
W an assignment, *W:Input(O)* ⟶ *Values*
then *Values-vector(pre,U,V,W,O)* $=_{def}$ $v_1,...,v_n$
where v_i = if O_i = *?x:s* then *W(x)*
if O_i= *!E* then *val(E at pre-func(pre,U)(V))*

4.2.3. Formal semantics of Primol

We will use the Structural Operational Semantics formalism [Plo81] to describe how a single transition can take place in a PRIMOL pre-implementation. Furthermore, in line with what was mentioned in an introductory subsection, the semantics of PRIMOL will be given in two sets of rules: a first set which describes what are the symbolic actions that are firable from a given (control) state, and a second set which describes the actual transitions that can take place when a symbolic action is firable.

We will assume the existence of a syntactical transformation (which is too lenghty to describe here) on *ground-terms action-expressions*, i.e., *actions-expressions ae* with $AV(ae)=\phi$. This syntactical transformation denoted by *ground-form* is such that for any *ground-term ae*, the result of *ground-form(ae)* is either "*?*"or a *ground-action*. Intuitively, *ground-form*

replaces an action-expression by a simplified one which is semantically equivalent. *ground-form(ae)* yields a"*?*" when there is incompatibility between actions that should be synchronized in a *sync ... endsync*, or if one of the *label-guards* is false

In the inference rules, SYMBOLIC1,...,SYMBOLIC3, OPERATION1, OPERATION2, given below the notation *(ND,S)* stands for "*ND* at state *S*" (*ND* here is either a node or a sub-tree and *S* a *control-state*), and (*^nd,S/V*) stands for "*^nd at control-state S* and *variable-state V*".

SYMBOLIC 1

$$\frac{s \Rightarrow \overline{\text{from}(t)}}{(nd, s) \xrightarrow[\text{symbolic}]{(action(t),\ next(t))}}$$

In this rule, *t* is a transition of node *nd*. $s \Rightarrow from(t)$ denotes the fact that *nd* is at state *s* where *t* is firable. This rule infers that if $s \Rightarrow from(t)$ then the node *nd* can participate to the global transition by the application of (*action(t)*, *next(t)*).

SYMBOLIC 2

$$\frac{(\hat{}nd_1, S_1) \xrightarrow[\text{symbolic}]{(ae_1, ne_1)} , \ldots , (\hat{}nd_p, S_p) \xrightarrow[\text{symbolic}]{(ae_p, ne_p)} , (nd, s) \xrightarrow[\text{symbolic}]{(ae, ne)} , \quad S \Rightarrow (s \oplus S_1 \ldots \oplus S_p),\ AV(ae) = \{action(nd\text{-}id(nd_1)), \ldots, action(nd\text{-}id(nd_p))\}}{(\hat{}nd, S) \xrightarrow[\text{symbolic}]{(ae\,[\,ae_i \,/\, action(nd\text{-}id(nd_i))\,],\ (ne_1,\ldots,ne_p,ne))}}$$

This rule shows how a given node *nd* synchronizes each of its transitions with those of its subnodes (if any). The result of this synchronization is obtained by replacing the *action-variables* occuring in a firable transition of *nd* by *action-expressions* that are firable in the corresponding subnodes.

SYMBOLIC 3

$$\frac{(\hat{}nd, S) \xrightarrow[\text{symbolic}]{(ae, ne)}, \quad ga = ground\text{-}form(ae)}{(\hat{}nd, S) \xrightarrow[\text{symbolic}]{(ga, ne)}}$$

This rule infers that an *action-expression* can be replaced by its *ground-form*.

OPERATION 1

if *ga* is a *ground-action* where
ga = (*pre*, (*g*,*i*), $O_1 \,//\, \ldots \,//\, O_n$, *sp*, (*si*)),
V is a *Variable-state*,
U is an assignment $U: Parameters(pre) \longrightarrow Values$

W is an assignment $W: Input(O_1 \,//\, \ldots \,//\, O_n) \longrightarrow Values$
such that

$pre\text{-}guard(pre,U)\,(V) = true,$

$Values\text{-}vector(pre, U, V, W, O_1) = \ldots =$
$= \ldots = Values\text{-}vector(pre, U, V, W, O_n)$

$val(sp \text{ at } pre\text{-}func(pre,U)(V) \setminus W) = true$

then

$$\frac{(\hat{}nd, S) \xrightarrow[\text{symbolic}]{(ga,\, ne)}}{(\hat{}nd, S / V) \xrightarrow[\text{operation}]{i} (\hat{}nd, S' / V')}$$

This operational rule discribres how a given tree of nodes $\hat{}nd$, rooted at a node nd, evolves due to the application of an internal (with visibility i) ground-action ga, calculated using the symbolic rules. Note that the resulting label of the transition is i.

where

$S' = S \setminus \overline{ne} \setminus \overline{si}$

$V' = (pre\text{-}func(pve, U)(V)) \setminus W$

OPERATION 2

if ga is a *ground-action* where

$ga = (pre, (g,e), O_1 \,//\, \ldots \,//\, O_n, sp, (si)),$

V is a *Variable-state*,

U is an assignment $U: Parameters(pre) \longrightarrow Values$

W is an assignment $W: Input(O_1 \,//\, \ldots \,//\, O_n) \longrightarrow Values$

such that

$pre\text{-}guard(pre,U)\,(V) = true,$

$Values\text{-}vector(pre, U, V, W, O_1) = \ldots =$
$= \ldots = Values\text{-}vector(pre, U, V, W, O_n)$

$val(sp \text{ at } pre\text{-}func(pre,U)(V) \setminus W) = true$

then

$$\frac{(\hat{}nd, S) \xrightarrow[\text{symbolic}]{(ga,ne)}}{(\hat{}nd, S / V) \xrightarrow[\text{operation}]{g\ v_1 \ldots v_n} (\hat{}nd, S' / V')}$$

This operational rule discribes how a given tree of nodes $\hat{}nd$, rooted at a node nd, evolves due to the application of an external (with visibility e) ground-action ga, calculated using the symbolic rules. Note that the resulting label of the transition is $g\ v_1 \ldots v_n$.

where

$S' = S \setminus \overline{ne} \setminus \overline{si}$, $V' = (pre\text{-}func(pre, U)(V)) \setminus W$

$v_1 \ldots v_n = Values\text{-}vector(pre, U, V, W, O_1)$

4.3. About the Translatability of LOTOS into PRIMOL

As it may be noticed, PRIMOL is a model where the number of value-identifiers and nodes used in any given pre-implementation is fixed. This implies that a syntactical constraint on LOTOS specifications is needed in order to ensure their translability into PRIMOL. This condition is the same as the one which ensures the finiteness of dynamic

process creation in LOTOS. Such a condition is described and proved in [GaN89]. The statement of this condition is given below, followed by the definition of the notion of *architectural occurrence* which is used in it.

Condition: A LOTOS specification is translatable into PRIMOL iff:

a) all the occurences of parallel operators in this specification are *architectural*,
b) any non *architectural* occurence of an enable or a disable operator in a process definition is such that there is no direct or indirect instantiation of this process from within the behaviour-expression which is the left argument of this operator.

Definition: an occurence, in a process-definition, of a LOTOS operator is said to be *architectural* iff there is no direct or indirect instantiation of this process from within the behaviour-expressions that are the arguments of this operator.

4.4. PRIMOL to PRIMOL transformations

The design of PRIMOL was conditioned by the ability of applying simple syntactical transformations. Any PRIMOL to PRIMOL transformation is in fact a sequence of elementary transformations.

4.4.1. The PRIMOL to PRIMOL elementary transformations

An elementary PRIMOL to PRIMOL transformation is merely the merging of a node and one of its subnodes into a new node. In fact, any couple formed by a node and one of its subnodes can be merged. This merging mechanism is described in the following paragraph.

Let *nd* be a node and *nd'* one of its subnodes as given in the table below:

	nd	*nd'*
NODE	*n*	*n'*
SUBNODES	*sub*	*sub'*
LOCAL-STATE-VARIABLES	*lst*	*lst'*
LOCAL-INITIALIZE	*loc-init*	*loc-init'*
SUBNODES-INITIALIZE	*sub-init*	*sub-init'*
TRANSITIONS	*trans*	*trans'*

The transformed pre-implementation is obtained by replacing nodes *nd* and *nd'* by a new node *ND* obtained from *nd* and *nd'* by the following syntactical transformations:

A preliminary transformation:
First node nd should be transformed so that the initialisations of subnode nd' occuring in the transitions (action-expressions) of node nd become next state expressions (since nd and nd' are going to be merged). This first transformation is discribed below.

node *nd* is replaced by *another node n_nd* which is similar to *nd* except for the transitions *trans={...,ti,...}* which are transformed using the following scheme:

for each transition t_i = *(from(ti), action(ti), next(ti))* obtain a new transition tt_i by replacing for each occurence of a *sub-init-preaction*, say *si*, such that *n' ∈ si* ,

si is replaced by *si ∪ sub-init' - {n'}*

and in *ti*, *next(ti)* is replaced by *next(ti) ∪ loc-init'*

The transitions *ttrans* of *n_nd* is the set *{...,tti,...}*.

The new node *ND* is given below:

NODE	*N*
SUBNODES	*SUB*
LOCAL-STATE-VARIABLES	*LST*
LOCAL-INITIALIZE	*LOC-INIT*
SUBNODES-INITIALIZE	*SUB-INIT*
TRANSITIONS	*TRANS*

where

N = *n*

SUB = *sub ∪ sub' - {n'}*

LST = *lst ∪ lst'*

LOC-INIT = if *n' ∈ sub-init* then *loc-init ∪ loc-init'*
if *n' ∉ sub-init* then *loc-init*

SUB-INIT = if *n' ∈ sub-init* then *sub-init ∪ sub-init' - {n'}*
if *n' ∉ sub-init* then *sub-init*

TRANS = *trans1 ∪ COMB (trans2,trans')*

where *trans1* and *trans2* are such that:
ttrans = trans1 ∪ trans2,
∃ t ∈ trans1 , action(n') ∉ AV(t),
∃ t ∈ trans2 , action(n') ∈ AV(t)

and where
COMB(trans2 ,trans') =def
{ (fr,ae,ne) | ∃ t ∈ trans2, ∃ t' ∈ trans' such that:
fr = (from(t) AND from(t')) ,
ae = action(t) [action(t') / action(n')])
ne = (next(t),next(t')) }

Theorem: elementary transformations are semantics preserving (modulo strong equivalence).

proof: ommitted.

4.4.2. About using transformations in PRIMOL

The flexibility of the transformational framework of PRIMOL is such that many fields of applications can be envisaged. Following is a list of application examples.

a) a pre-implementation can be transformed into a single-node pre-implementation. The EFSM of this single node can be used for different purposes:
   - deadlocks can be detected by the analysis of the control part of this EFSM,
   - many properties can be checked by expanding the value-identifiers of this EFSM and thus obtaining a (finite) labelled transition system which in turn can be analysed using existing theories and algorithms [BoS88, Naj88,...] and tools [Ver86,...],
   - general test derivation theories [BSS87,dMe86,Bri88a,...] and EFSM based test theories can be applied to this EFSM,
   - the step by step interpretation and the simulation of the source LOTOS specification can be more efficiently conducted using this EFSM.

b) a pre-implementation can be transformed into a two-level pre-implementation. These two levels can then be mapped onto a real implementation where the root node is mapped on a "kernel" process and the subnodes are mapped on real "operational" processes.

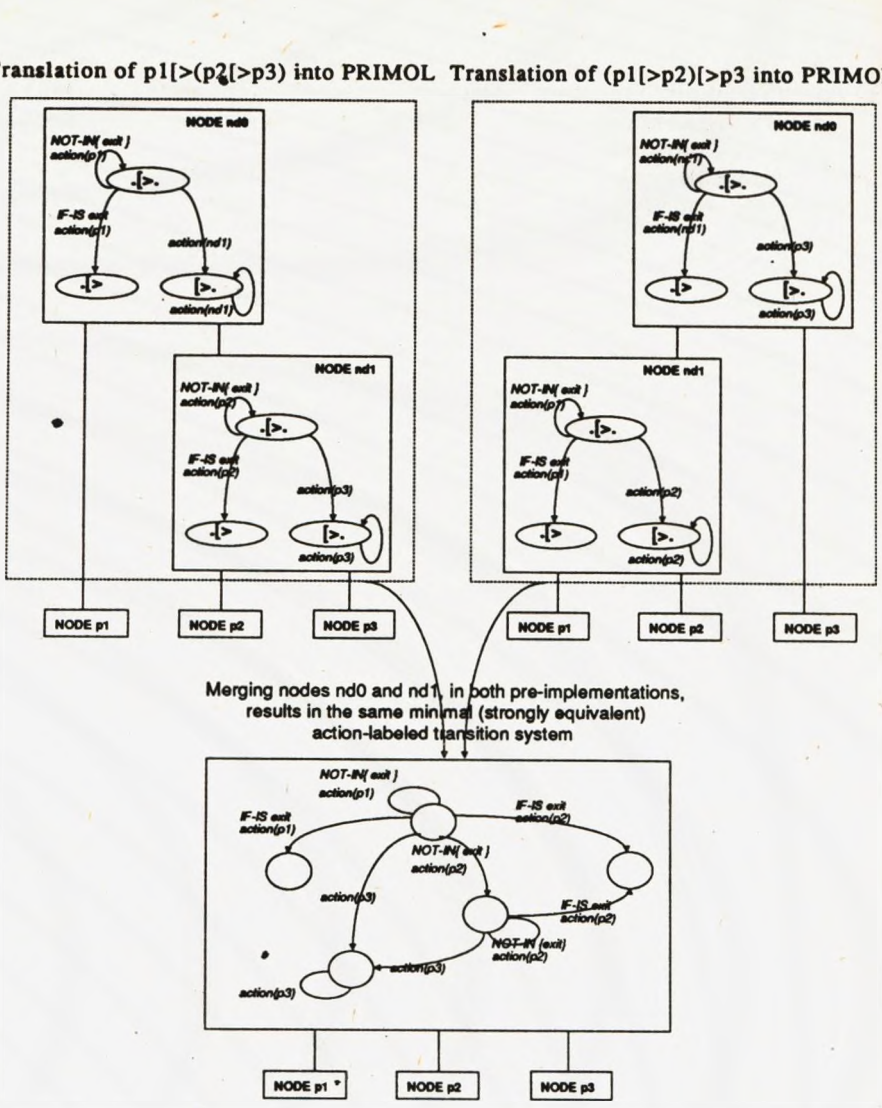
c) PRIMOL can serve as a theorem proving framework for LOTOS. In order to prove the equality (strong, weak, testing, ...) of two open behaviour-expressions, one can translate these two behaviour-expressions into two two-level pre-implementations. The roots of these two pre-implementations may then be compared using an adaptation of the verification framework used for the finite labelled transition systems (an example is given below).

5. Example

One of the advantages of Pre-Implementation is that it flatens all LOTOS operators of a given *behaviour-expression* into the same basic elementary construct, i.e., EFSM(s). This makes it an interesting approach for checking equivalences between LOTOS *behaviour-expressions*. Furthermore, open *behaviour-expressions*, i.e., with *behaviour-variables*, can also be translated into *pre-implemetations* where the *behaviour-variables* are mapped on empty *sub-nodes* and referenced through the *action-from-subnodes* occuring in the *action-expressions* of the parent node. This implies that PRIMOL is also a framework for theorem proving on *behaviour-expressions*. An illustration of this last is the proof of the associativity of the disable operator in LOTOS. This problem can be stated as follows, where ≡ is the strong equivalence equality sign:

$$\forall\, p1, p2, p3 \quad (p1 \;[>\; p2) \;[>\; p3 \;\equiv\; p1 \;[>\; (p2 \;[>\; p3)$$

The proof is established by translating the two sides of the equality into PRIMOL and then transforming each of the two pre-implementations into a two-level pre-implementation where the leaves correspond to p1, p2 and p3. The proof is obtained by cheking the strong equivalence of the finite automata that constitute the transition part of the root nodes of these two-level *pre-implemetations*. This proof mechanism is graphically depicted below.



6. Conclusion

We have presented an approach and a model that can serve as a promising starting basis for an integrated workbench for Process Algebras in general, and LOTOS in particular, covering the topics of verification and implementation. However, non-determinism and fair implementation issues have not been addressed. Furhermore, The PRIMOL model is likely to be enhanced on aspects such as: the extension of the translatable subset of LOTOS, the definition of a more dynamic representation for the semantics, the study of strategies for the

application of transformations, the design of new transformations and the study of a verification framework. A tool, named LOTOMAT, is now under development. The first version of this tool is expected to include the translation of LOTOS to PRIMOL, the PRIMOL to PRIMOL transformations and the translation of PRIMOL into finite labelled transition systems.

As it may have been noticed, PRIMOL has commonalities with Estelle [IS9074,BuD87], an EFSM based standardized language. However, there is one main difference: the inter-process communication which is through Fifo queues in Estelle and through multiway rendez-vous in LOTOS/PRIMOL. The design of PRIMOL seem to indicate that LOTOS and Estelle could have been defined in such a way that the one is the pre-implementation model of the other. Perhaps [Cou88] is a step in this direction.

A tool called PIL was developed at INRIA. PIL Translates LOTOS specifications into PRIMOL and then allows the user to interactively choose the nodes to be merged and performs the transformations automaticaly. The example given in section 5. has been treated using PIL.

Acknowledgments

The authors would like to thank their colleagues at INRIA for their support. Special thanks to J.B. Saint, P. Cunha and A. Rizk for their editorial remarks.

References

[BeK85] Bergstra, Klop: "Algebra of Communicating Process with Abstraction", Theorical Computer Sciences 37,pp. 77-121, 1985.

[BFL86] J.P. Briand, M.C. Fehri, L.Logrippo, S.Obaid: "Executing LOTOS Specifications", In Protocol Specification, Testing and Verification, North-Holland, Volume 1,pp. 73-84, 1986.

[BoB87] T. Bolognesi, E. Brinksma: "Introduction to the ISO Specification Language LOTOS", Computer Networks and ISDN Systems, Vol 14, pp. 25-59, 1987.

[BoS88] T. Bolognesi, S.A Smolka: "Fundamentals Results for the Verification of Observational Equivalence: A survey", 7th Iternational Workshop on Protocol Specification, Testing and Verification, North-Holland, Amsterdam, Eds. H.Rudin and C.West, 1988.

[Bou84] G. Boudol: "Notes on Algebraic Calculi of Processes", Advanced Nato Scholl series on Logics and Models for Verification and Specification of Concurrent Systems, Springer Verlag, 1984.

[Bou87] G. Boudol : "Communication is an abstraction", INRIA, RR636, March 1987.

[Bri88] H. Brinksma: "On the Design of Extended LOTOS", Ph.D. Thesis, University of Twente, November 1988.

[Bri88a] H. Brinksma: "A theory for the derivation of tests", Eighth International Symposium on Protocol Specification, Testing, and Verification,Atlantic City, New Jersey, Ed Sudhir Aggarwal and Krishan Sabnani,1988.

[BSS87] H. Brinksma, G. Scollo, C. Stennbergen: "Process Specification, their Implementations, and their Tests", 6th International Workshop on Protocol Specification, Testing and Verification, North-Holland, Amsterdam, Eds. B.Brinksma and G.V. Bochmann, 1987.

[BuD87] S. Budkowski, P. Dembinski: "An Introduction to Estelle: A Secification Language for Distributed Systems", Computer Networks and ISDN Systems 14, North-Holland, 1987.

[Cou88] J-P. Courtiat: "Estelle*: a powerful dialect of Estelle for OSI protocol description", 8th IFIP Symposium on Protocol Specification, Testing and Verification, Atlantic City, June 1988.

[dMe86] J. de Meer: "Derivation of Test scenarios based on the formal specification Language Lotos", 6th International Workshop on Protocol Specification, Testing and Verification, North-Holland, Montreal, Eds. G.V. Bochmann, B. Sarikaya, 1986.

[dNH84] R. de Nicolas, M.C.B. Hennessy: "Testing Equivalences for Processes",Theoretical Computer Sciences 34, pp. 83-133, 1984.

[dSi85] R. de Simone: "Higher Level Synchronizing Devices in Meije-SCCS", Theoretical Computer Sciences 37, pp. 245-268, 1985.

[Eij89] P.V. Eijk: "The design of a simulator tool", The Formal Description Technique Lotos, pp 351-390, North Holland, Eds. M. Diaz, P.V. Eijk, C. Vissers 1989.

[GaN89] H. Garavel, E. Najm: "Tilt from Lotos to Labelled Transition Systems", The Formal Description Technique Lotos, pp 317-336, North Holland, Eds. M. Diaz, P.V. Eijk, C. Vissers 1989.

[Hoa85] C.A.R. Hoare: "Communicating Sequential Processes", Prentice-Hall International, 1985.

[IS8807] IS 8807: "Information Processing Systems - Open Systems Interconnection, LOTOS - A Formal Description Technique Based on the Temporal Ordering of Observational Behaviour", May 1988.

[IS9074] IS 9074: "Information Processing Systems - Open Systems Interconnection, Estelle - A Formal Description Technique Based on Extended State Transition Model", May 1988.

[Kar88] G. Karjoth: "Implementing Process Algebra Specifications by State Machines", Eighth International Symposium on Protocol Specification, Testing, and Verification,Atlantic City, New Jersey, Ed Sudhir Aggarwal and Krishan Sabnani,1988.

[MaM88] J. Manas, T. de Miguel: "From LOTOS to C", First International Conference on Formal Description Techniques, Elsevier Science Publishers, Amsterdam, Ed. K.J. Turner, 1988.

[MaS87] E. Madeleine, R. de Simone: "Ecrins, un laboratoire de preuve pour le calcul de process", INRIA, RR672 1987.

[Mil80] R. Milner: "A Calculus of Communicating Systems", Lecture Notes in Computer Science 92, Springer Verlag, 1980.

[Mil83] R. Milner: "Calculi for Synchrony and Asynchrony", Theoretical Computer Science 25, pp. 267-310 1983.

[Mil84] R. Milner: "A complete inference system for a class of regular behaviours". Journal of Computer and System Sciences, pp. 439-466, 1984.

[NaB86] E. Najm, J. Barre: "Lotos, un Nouveau Langage de Specification des Systemes Distribues", 3eme Congres De Nouvelles Architectures pour les Communications, Ed Eyrolles 1986.

[Naj88] E. Najm: "A Verification Oriented Specification in Lotos of the Transport Protocol", 7th International Workshop on Protocol Specification, Testing and Verification, North-Holland, Amsterdam, Eds. H.Rudin and C. West, 1988.

[Par85] J. Parrow: "Fairness Properties in Process Algebra, with Applications in Communication Protocol Verification". Phd thesis, Departement of Computer Systems, Uppsala University, Uppsala, Sweden, 1985.

[Plo81] G. Plotkin: "A Structural Approach to Operational Semantics", Daimi FN-19,Aarhus University, 1981.

[QPF88] J. Quemada, S. Pavon, A. Fernandez: "Transforming LOTOS Specifications With Lola: The Parametrized Expansion", First International Conference on Formal Description Techniques, Elsevier Science Publishers, Amsterdam, Ed. K.J. Turner 1988.

[Tre89] I. Tretmans: "Hippo a Lotos Simulator", The Formal Description Technique Lotos, pp 391-396, North Holland, Eds. M. Diaz, P.V. Eijk, C. Vissers 1989.

[Ver86] D. Vergamini: "Verification by Means of Observational Equivalence on Automata", Technical Report 501, INRIA, March 1986.

FUNCTIONAL AND PERFORMANCE PROTOTYPING CONCEPTS BASED ON SDL

Dieter Carl

Institute of Informatics and Computer Technics
Academy of Sciences of the GDR
Rudower Chaussee 5
Berlin
DDR-1199 (GDR)

The use of SDL in an ISDN terminal design is described and the gained experiences are presented. Some ideas about Rapid Prototyping of SDL specifications on the basis of PROLOG, SIMULA and C are mentioned. The SDL oriented tool kit under development is characterized.

1. INTRODUCTION

As a part of research work in the ISDN field we are working on the realization of a multifunctional ISDN terminal. First the telephony part was developed. We used a software engineering methodology based on SDL and CHILL. This methodology was supported by tools, which enable the analysis of SDL specifications before the beginning of any implementations. The logic of the application processes and the protocols of the D-channel were specified in SDL*, a subset of SDL'88 [1] and implemented in CHILL*, a subset of CHILL'84 [2].

2. USE OF SDL

2.1 SDL* - an SDL subset

For the given task an SDL subset named SDL* [3] was defined. The essential limitations in relation to SDL'88 are

- The use of data generators is reduced to some standard cases to enable the generation of some simple but not predefined modes, as e.g. enumeration modes, named value sets.
- SDL* processes can communicate only by exchanging signals.
- There are no global variables, neither in processes nor in procedures.
- The substructuring concept is available for blocks only.

- The service concept is not used.
- The use of JOIN is strongly limited, e.g. it is not permitted to jump into a DECISION from outside.
- The declaration of valid signal sets, channels and signal routes is mandatory.
- The declaration of the destination process in an OUTPUT statement (TO part) is mandatory. The only exception is the use of the VIA construction during the establishment of the name space of the system.

The main reasons for these limitations are

- To support a well structured design and avoid some error classes (e.g. side effects),
- To reduce the effort for learning and
- To reduce the effort for tool development.

2.2 Rules for the use of SDL*

We felt it usefully to have two levels of specification:

- The **systems definition,** which reflects the static structure of the system and
- The **process specifications**, which represent the flow of control and data.

In both cases the graphical syntax (SDL*/GR) is the main user interface. Some rules limit the degree of freedom in the graphical arrangements to simplify the task of automated processing of the specifications. Limited are e.g.:

- the permitted line flow,
- the permitted set of symbols at the beginning of a page and
- the use of jumps and connectors.

For the specification of layered software (as in the case of the D-channel handler) additional rules are useful:

- One block represents one layer.
- Each block contains a root process that is created together with the system. This root process creates all the other processes of the block dynamically.
- Source and direction of a signal are given by naming conventions according to the OSI Reference Model [4]. (e.g. N_CONN_req means a signal (primitive) that is directed from the (N+1)-layer to the N-layer, but N_CONN_ind is a signal from the N-layer to the (N+1)-layer).
- The encoding and decoding of messages is specified separately. This makes a process specification much more readable and simple data types are sufficient therefore.

According to our understanding a specification is complete,

if there are no requirements for further logical decisions in the implementation process. Such a specification we named an **Implementation Specification**. When specifying protocol layers a so called **Protocol Specification** has an independent significance. Such a specification contains the complete protocol logic but not the mapping of PDUs to IDUs.

The full formalization of a specification is necessary to automated or semiautomated validation, analysis, implementation and testing.

3. EXECUTION OF SDL* SPECIFICATION (RAPID PROTO TYPING)

3.1. Goals for analysis of specifications

One of the main goals of modern software engineering methodology is the error detection as early as possible. This includes the urgency of validation of a formal software design, in our case of SDL* specifications. There are two different validation techniques:

- the analysis of the **static properties** of the design by inspection of the SDL* text, which represents a given specification and
- the analysis of the **dynamic behaviour** of the design by execution (interpretation) of the specification.

By systematic inspection one can find such errors as missing INPUT corresponding with a given OUTPUT action, static unreachable states, saved but nowhere received signals. The main advantages of the systematic inspection are the relative simplicity of the realization and the possibility to guarantee the absence of some error classes.

But if one want to analyse more complex situations, then one has to take into account the dynamic relations inside the system.
By specification execution (so called **Rapid Prototyping**) one is able to carry out a complete or an incomplete analysis. 'Complete analysis' means that each of the possible transitions in the state space is executed at least once with each permutation of the allowed population of the signal input port and the variables. The serious problem of explosion of the state space and of the number of permutations in specifications of real world systems exist. Actually this problem is reduced by the development of new search and backtracking algorithms, but not solved in general. Above all the complexity of possible data populations remains as a serious problem.

Further we only consider the aspect of incomplete analysis of the dynamic behaviour. The execution of a specification requires two steps:

- first, the transformation of the specification in an exe-

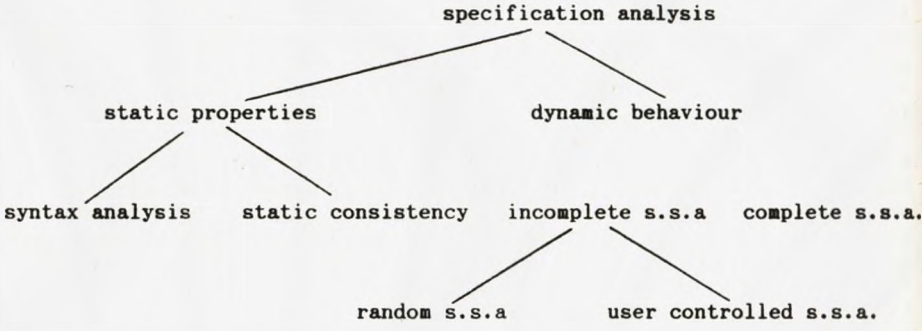
cutable computer program and
- second, the execution of this program together with an appropriate run time support system (this compound is called the **prototype**).

The specification can be executed in two different ways:

- as a **random state space analysis**, i.e. the interpretation of a specification on the basis of random branching and random environment behaviour [5],
- as a **user controlled analysis**, which can be compared with a normal test methodology.

The main advantage of the first mentioned method is the ability of analysis without an explicite test scenario. Therefore no additional work is needed. The analysis can start immediately after termination of the design process.
But if one wants to analyse the behaviour of a system under predefined conditions, e.g. under an extreme population of variables or under a rare signal combination, then the user control of analysis is demanded. This user control can be realized by an additional specification of a system environment according to the wanted test suite.
Thus, both methods support alternative goals and require different expense.

Figure 1 gives a summary on specification analysis goals.



s.s.a - state space analysis

Figure 1: Specification analysis goals

3.2. Random state space analysis using PROLOG

We found, that most of the design errors were of the kinematical error type, e.g. deadlock, endless loop, no return to the initial state, incorrect use of timers. Therefore we begun the development of a special tool based on PROLOG for the automated detection of such dynamical inconsistencies by

using the random state space analysis method.
The use of PROLOG was suggested by the following properties of this language:

- PROLOG allows the description of the logic of an analysis process in a natural way, the built-in inference mechanism is very efficient.
- Because of the logical foundation of PROLOG there is no need for additional assumptions (e.g. time semantics) and
- a PROLOG description is executable without any additions.

An executable prototype consists of the PROLOG equivalent of the SDL* specification and the PROLOG program PROEXE (PROlog EXecution Environment) [6], which gives the required run time support.
In the actual version of the prototyping environment for the transformation of an SDL* specification into PROLOG the following clauses are available, which are interpreted by PROEXE:

system, process	- definition of the process set of the system under analysis
start, stop	- start and stop of a process run
call, return	- call of a procedure and return from it
timer	- definition of a timer
sav	- save of a signal in the input port
input, trans	- input of the current signal from the input port and start of the corresponding transition
nextstate	- call of a new state
output	- output of a signal
setto, resetto	- set and reset of a timer
era, ass, inc	- manipulation of the value of a variable
chk	- check of the value of a variable
envsig	- definition of a signal from the environment

The analysis will be provided by a random controlled run of the PROLOG prototype. Therefore one can influence the random behaviour of the

- environment,
- timeouts and
- message loss

in the following manner:

Each time a 'nextstate'-clause is reached the PROEXE built-in scheduler is called. It performs a round robin scheduling. Before reactivating a process the current state of this process is checked. If at least one among the expected signals is from the environment, then with a user controlled probability p_{env} a legal such signal is generated and written at the beginning of the input port. If there is an active timer, then with a certain probability p_{tim} a timer signal is generated additionally and the timer is deactivated. If the

scheduler does not find a ready process, which can be activated, then it has to check for a deadlock state of the system. This is only the case if there is not any appropriate active timer or environment signal.
An important aspect is the modelling of signal loss. If PROEXE interpretes an 'output'-clause, a check is made for timeout monitoring of this signal in the destination process. If the signal is monitored, the send operation is suppressed with a user controlled probability p_{loss}.

The timer consistency can be examined easyly by interpreting of the 'setto'- and 'resetto'-clause. A missing RESET has as consequence an invalid timer signal.

The following diagnostics are made:

- deadlock of the whole system,
- discard of a message or timer signal,
- set of an active timer and reset of an inactive timer,
- missing destination process and
- updating or reading of an uninitialized variable.

For analysis purposes the gathering of statistical information is an important aspect. These are e.g. input port statistics, routing statistics, signal statistics. By such statistics it is additional possible to find out some abnormal reactions of the system.
For debugging purposes some tracing and report facilities are planned.

The following example gives a comparison of the SDL- and PROLOG-representation of a small part of a process specification and illustrates the straightforward transformation:

SDL	PROLOG
STATE s1; INPUT message; RESET timer; OUTPUT ack TO partner; NEXTSTATE s2; INPUT timer; TASK err := err + 1; DECISION err > 3; (TRUE): OUTPUT error TO monitor; NEXTSTATE null; (FALSE): SET(t,timer); NEXTSTATE s1; ENDDECISION; ENDSTATE s1;	trans(p,s1,message) :- resetto(p,timer), output(p,ack,partner), nextstate(p,s2). trans(p,s1,timer) :- inc(p,err,ERR), transalt(p,ERR). transalt(p,ERR) :- ERR > 3, output(p,error,monitor), nextstate(p,null). transalt(p,_) :- setto(p,t,timer), nextstate(p,s1).

As it is seen, the SDL to PROEXE transformation is more or less straightforward. But at least two problems remain:

- to distinguish variables with the same name but declared on different levels:

  Each variable is represented by a PROLOG fact of the type

  var(p_id, pd1_id, pd2_id, var_id, var_val) with

p_id	processidentifier
pd1_id	identifier of the first level procedure (if exists)
pd2_id	identifier of the second level procedure (if exists)
var_id	variable identifier
var_val	current variable value

- the realization of JOIN

This is solved by additional PROLOG procedures. The following example shows the principle:

The SDL sequence

```
   A;
     LABEL:
          B; C; ... NEXTSTATE sx;
  ...     JOIN LABEL;
```

is transformed to

```
   a.
     label.
          label :- b,c, ... nextstate sx.
          ...
          label.
```

The actual state of the whole system under investigation is represented by a set of PROLOG facts (database). These facts are manipulated by help of the PROLOG meta programming (retract/assert).

The main elements of the *PROEXE internal database* are:

control information

```
plist(Q)          Q: list of ready processes

currstat(P,S)     P: processname
                  S: name of current state

inpq(P,Q)         P: processname
                  Q: inputqueue

rand(R)           R: current state of random
                     number generator
```

lists of actual saved signals

```
savq(P,Q)      P: processname
               Q: list of saved signals
```

indicators of active timers

```
timout(P,T)    P: processname
               T: time interval
```

temprary buffers of message parameters

```
signal(P1,P2,Sig,Par) P1: name of sourceprocess
                      P2: name of destinationprocess
                      Sig: signalname
                      Par: list of signalparameters
```

To give an impression how to write an SDL oriented runtime support system using PROLOG we give the simplified implementation of the NEXTSTATE statement:

```
/* manipulation of input queue of the process P                          */

nextstate(P,S) :- envload(P,S),savload(P),timload(P,S),!,state(P,S).

state(P,S) :- inpq(P,Q),empty(Q),!,wait(P,S). /* wait if input queue empty  */
state(P,S) :- siginp(P,Sig),statalt(P,S,Sig). /* processing of first signal */
state(P,S) :- state(P,s).                     /* lock for next signal       */

siginp(P,Sig) :- inpq(P,Q),first(Q,Sig),!, /* remove first signal from queue*/
                 retract(inpq(P,Q)),del(Q,Q1),asserta(inpq(P,Q1)).

statalt(P,S,Sig) :- clause(trans(P,S,Sig),_),!,trans(P,S,Sig).
statalt(P,S,Sig) :- sigsave(P,S,Sig),!,fail.
statalt(_,_,_)   :- write('++ signal discarded ++'),fail.

                 . . .
```

```
/* random generation of a timeout signal and adding it into the queue      */

timload(P,S) :- timout(P,T),clause(trans(P,S,T),_),!,timalt(P,T).
timload(_,_) :- true.

timalt(P,T) :- ranswi,!,timstor(P,T).
timalt(_,_) :- true.

timestor(P,T) :- retract(timout(P,T)),retract(inpq(P,Q)),
                 add(Q,T,Q1),asserta(inpq(P,Q1)).

ranswi :- random(R),ptim(PT),R < PT.                    /* random branching */

/* random number generator                                                  */

random(R) :- retract(rand(Q)),R is Q mod 100, Q2 is 125*Q + 1,
             Q1 is Q2 mod 4096,asserta(rand(Q1)).
```

3.3. User controlled analysis using SIMULA

The above presented variant of random state space analysis is not able to detect errors of the following classes:

- missing reaction on a valid environment signal,
- missing timeout monitoring of a signal,
- lifelocks and
- 'local' deadlocks.

Because of the low probability of extreme situations the random analysis requires usually a lot of computer time to generate such conditions. To overcome these problems a user controlled analysis can be helpfully.

Therefore one can use the PROLOG oriented technique described above or a simulative method. In the following cases a simulation oriented analysis method is needed:

- if one wants to adjust the timers in a system in an optimal way and
- if one wants to carry out a performance analysis.

The following properties of SIMULA led to the decision, to take this language for the simulative analysis under user control:

- the objekt orientation (classes, virtual procedures),
- the coroutine concept,
- the built-in classes SIMSET and SIMULATION and
- the high level language support (e.g. in comparison with GPSS).

In our first approach the simulation program (the prototype) consisted of two levels, the run time support system COSIL (COmputer SImulation Language) [7] and a subclass of COSIL, which is the SIMULA equivalent of the SDL* specification. COSIL itself is a subclass of SIMULATION and offers the following classes for the SDL* to SIMULA transformation:

```
link class SERVER     - modelling of processors
process class TASK    - modelling of processes
link class EVENT      - abstact event data type
process class SIGNAL  - abstract signal data type
process class TIMER   - modelling of timers
link class  QUEUE     - abstract queue data type
class STATISTICS      - means to observe the running prototype
```

Each of these classes has some attribute procedures, which perform the operations on the object, e.g. via the following SERVER-procedures the scheduler is called:

```
wait(event_pointer) - blocking of the calling process,
deactivate          - self interrupt of the running process,
delay(dt)           - blocking of the running process for a
                      time intervall dt.
```

In connection with the deviation of prototypes from SDL* specifications there are to solve two fundamental problems:

- In SDL there is no possibility to specify any time consumption. But for a simulative analysis on the basis of SIMULA this is needed (SIMULA - 'hold' statement).
- For the purpose of performance estimations it is further necessary to have features to specify the mapping of processes to processors, the utilization of ressources, to determine the scheduling disciplines, and so on.

There exist some proposals to extend SDL for analysis purposes, but this is not a good approach, because for a given system one gets different specifications dependent on the goal of their use (consistency check, performance analysis, implementation). We propose an other solution [8]: we understand the run time support system (in our case COSIL) as a representant of the real underlying system. As a consequence of this view such properties as transmission delay are realized by controlling the run time support system. Such an approach allows a clear distinction between the logic of a specification and the properties of the underlying real or prototype system.
In the case of COSIL it is possible to model the transmission delay by mapping an SDL channel to a COSIL server. The delay parameters are controlled on the COSIL level, not on the SDL level.
As a successor of COSIL the new run time support system SDL MACHINE was developed [9], its interface is closer to SDL*. This simplifies the mapping from SDL* to SIMULA.

Because SIMULA was available only on mainframe computers, a C based simulation environment was implemented [10], which enables a relative easy portage of most of the SIMULA programs. Based on this tool a new C coded version of the run time support system SDL MACHINE is now available.

In connection with this idea we want to realize an evolutionary prototyping, i.e. that one goes step by step from the specification to the real world implementation.

An other aspect for future developments is the performance estimation of a design based on the corresponding SDL* specification. For that we shall work into two different directions:

- extension of the simulation environment to offer new features for model configuration and parametrization and
- development of a transformator from SDL* to the input language of the performance modelling tool SATURN [11], which is based on extended queueing networks.

3.4. An analysis example

The CCITT specification of the D-channel level 3 protocol [13] was changed for following reasons:

- The open ended character of our design with the final goal of a multifunctional terminal leads to a strong layered structure of our implementation. Therefore we defined an ISDN network service (N_CONN_req, N_SETUP_req, etc.) using as far as possible the OSI concepts. Thus e.g. an arriving SETUP_ACK message is transformed into a N_SETUP_conf primitive.
- a consequence of the introduction of a network service the definition of an extended collision handling was necessary because of the asynchronism of outgoing and incoming primitives.
- The test and demonstration of the terminal should be possible without using an ISDN exchange. We performed the required symmetrization of the level 3 protocol in the following way:
  - Utilization of the options of recommendation I.451, e.g. a terminal is allowed to send a CALL PROCEEDING message and a terminal has to accept a CONNECT ACK message,
  - Extension of the recommendation I.451 by permitting incoming SETUP messages without any call information and by accepting and processing of INFO messages with call information by the terminal.

In this context there were to solve further collision problems. Let us consider such a problem in more detail:
In the initial state the telephone application process knows two permitted inputs: the signal 'hook_off' and the signal

N_SETUP_ind (connection request from the side B). The transition related to the first signal ends with a transfer to the state 'ready_to_dial'. In this state a N_SETUP_conf is expected:

```
STATE null;
  INPUT hook_off;
    ...
    OUTPUT N_SETUP_req TO layer_3;   /* this leads to a N_SETUP_ind signal at
    ...                                 the partner side                    */
    NEXTSTATE ready_to_dial;
  INPUT N_SETUP_ind;
    ...
    OUTPUT N_SETUP_resp TO layer_3;  /* this leads to a N_SETUP_conf signal
    ...                                 at the partner side                 */
    NEXTSTATE incoming_call;

   ...

  STATE ready_to_dial;
  INPUT N_SETUP_conf;
   ...
```

Let us consider the case that two terminals are connected directly. If both terminals try to establish a connection simultaneously, then a collision arises from the fact, that both processes go into the state 'ready_to_dial' but no N_SETUP_resp can be sent. The consequence is a 'hanging system'

By random analysis we did not found this error, but an experiment based on a defined test suite showed the effect.

In our practical work we gained the following experience: About 80 per cent of the specification errors were found by Rapid Prototyping (by random and user controlled analysis). The remaining part was only detected in the debugging phase of the implementation. This is not the result of a bad prototyping technique but of incomplete prototyping test suites. This experience shows, that in the frame of an incomplete state space analysis good test scenarios are demanded.
Consistently we begun to develop algorithms for an automated generation of test suites from SDL* specifications.

4. SDL* ORIENTED TOOL KIT

To support the software design and implementation based on SDL* and CHILL* in cooperation with a team from the Humboldt University in Berlin the following tool kit is under development:

SDL* editing

- SDL*/GR editor that allows a graphical input via the keyboard or mouse and includes a document generation module,
- transformator of the internal representation of SDL*/GR to SDL*/PR including the syntax check of SDL*/GR,
- reverse transformator from SDL*/PR to SDL*/GR,
- checker of the syntax and static semantics of SDL*/PR specifications including the generation of an internal tree notation (SDL* parser).

SDL* analysis and verification

- random state space analyser for SDL* specifications (see 3.2),
- simulation environment for the user controlled analysis of SDL* specifications (see 3.3),
- SDL* performance analyser which includes facilities for hybrid (simulative/ analytic) model evaluation.

automated implementation

- automated translation of SDL*/PR specifications to CHILL* and C programs including the consistent updating of the design and the corresponding programs,
- SDL* oriented runtime support system SDL/EX.

test support

- test scenario generator based on SDL* specifications,
- automated comparison between the the input/output behaviour of the implementation and given test scenarios.

5. CONCLUSIONS

SDL is an useful language for the specification of process structured software. There are at least four aspects of using an SDL specification:

- documentation and common base for design discussions,
- automated analysis and verification of a design,
- automated implementation, and
- automated creation of test suits.

The last three aspects require the completeness of a specification and the absence of informal parts. By means of Rapid Prototyping one can find not only most of the specification errors but can get first implementation experience also.

The availability of powerful tools to support the development of distributed systems is of high priority.

ACKNOWLEDGMENTS

I should like to thank my colleagues W. Blume, Th. Bez, K. Sommer and W. Rommel for their informations, suggestions and discussions and for support in the preparation of this paper. I also want to thank J. Fischer and K. Ahrens from the Humboldt University of Berlin for their cooperation in the field of SDL tool design and implementation.

REFERENCES

[1] CCITT, Recommendation Z.100, Specification and Description Language (SDL), blue book, Geneva, 1989

[2] CCITT, Recommendation Z.200, CCITT High Level Language (CHILL), red book, Geneva, 1985

[3] Carl, D., SDL* - Sprachkurzbeschreibung, Version 1.2, Internes Arbeitspapier ISDN/AN/SWT-SB/001, IIR/AdW, Berlin, 1989

[4] CCITT, Recommendation X.200, Reference Model of Open Systems Interconnection for CCITT applications, red book, Geneva,1985

[5] West, C.H., Protocol Validation by Random State Exploration, Proc. 'Protocol Specification, Testing and Verification', Montreal, 1986

[6] Carl, D., PROEXE - Prototyping-Umgebung zur dynamischen Konsistenzanalyse von SDL*-Spezifikationen nach dem Verfahren der Zufallsanalyse, Internal Paper, IIR/AdW, Berlin, 1989

[7] Carl,D., Zur Verhaltensanalyse von Rechnernetzen, Dissertation B, Akademie der Wissenschaften der DDR, Berlin, 1986

[8] Bez, Th., An SDL Oriented Prototyping Environment, 2. National Workshop 'Software-Technologie für Kommunikationssysteme', Bad Saarow, 1989

[9] Fischer, J., Software-Technologie "Rapid Prototyping" bei der Entwicklung von Kommunikations-Software verteilter sowie eingebetteter Systeme, Dissertation B, Humboldt-Univ., Berlin, 1988

[10] Bez, Th., Burghardt, F., Simulation with C, IIR report, Academy of Sciences of the GDR, Berlin, 1988

[11] Burghardt, F., Wirth, K.-D., user manual SATURN, Academy of Sciences of the GDR, Berlin, 1988

[12] CCITT, Recommendation I.440/441, ISDN user-network interface data link layer, Temporary document, Geneva, Nov. 1986

[13] CCITT, Recommendation I.450/451, ISDN user-network interface layer 3, red book, Geneva, 1985

EXPERIENCE IN COMPUTER-AIDED PROTOCOL IMPLEMENTATION

Hartmut KOENIG

Technical University "Otto von Guericke" Magdeburg
Department of Informatics
PSF 124, Magdeburg, 3010, G.D.R.

The paper deals with problems of computer-aided protocol implementation. It considers the several implementation steps under the aspect of computer-aided implementation. It discusses in particular the requirements of the implementation specification, the implementation model and the implementation environment, presenting our experience gathered by the implementation of the specification languages PDL and ESTELLE.

1. INTRODUCTION

The large gap the protocol developer has to overcome from the specification to the practical realization of protocols has brought forth the desire to automate this process. Therefore, computer-aided protocol implementation has become one of the main topics in the field of protocol engineering. Besides efficiency, computer-aided protocol implementation brings further remarkable advantages such as the reduction of implementation errors, the avoidance of incompatibilities between different implementations, better maintenance, and independence of a certain implementor /Aggarwal 85/. The experience shows further that automatically generated implementations are better structured than hand-coded ones.

Systems for computer-aided protocol implementation have been reported in literature since the beginning of the eighties. They are mostly based on state-transition specification languages. The principal implementation approach is similar in many cases. The specification text is transformed into a high-level language code, often PASCAL, C or CHILL, which is connected with hand-coded procedures. The latter usually compri-

ses such components as process organization applied in the implementation, and the buffer and queue management, and the time control as well. In practice the proportion of this part differs between 30 and 60 per cent.

Computer-aided protocol implementation is a topic of research. There are a lot of problems to be solved. More experience is needed in this connection. The paper will discuss some of these problems. It considers the different implementation steps under the aspect of computer-aided protocol implementation presenting our experience in this field gathered by the implementation of the specification languages PDL /Koenig 86/ and ESTELLE /Koenig 89/, /Krohmann89/.

2. IMPLEMENTATION STEPS AND DECISIONS

The implementation process consists of several implementation steps which require a lot of detailed decisions. Implementation starts with the implementation design based on the service and protocol specification. The implementation design must also reflect the specific conditions of the system the implementation is going for. Furthermore some general implementation decisions have to be made concerning the quantitative parameters of the implementation such as the number of connections handled by an entity or the quality of service to be provided. The result of the implementation design is documented in the implementation specification. The base of the implementation specification is the implementation model which describes the logical structure of the implementation. It is possible and not uncommon that the implementation model can be directly derived from the structure of the protocol specification if it fits the requirements of the real implementation. Then the implementation specification must be mapped onto the real system. This includes many decisions such as the choice of the implementation language and the determination of a useful process and module structure. Finally, the implementation has to connected to the implementation environment, especially to the operating system.

In the following we want to focus our discussion on three topics: the implementation specification, the implementation model, and the implementation environment.

3. IMPLEMENTATION SPECIFICATION

The implementation specification is usually the basis which computer-aided implementation systems start from. The elaboration of the implementation specification is a complicated and lengthy process. There are no unified rules about what the specification should include. Usually they differ from the protocol specification in containing concrete algorithms for certain local actions of the protocol entity not specified so far, in defining the interface to the adjacent layers, and in calling routines from the implementation environment. This information has to be supplemented by the specifier because it cannot be derived from elsewhere. For computer-aided implementation systems it is therefore important to have a clear cut between the things the specifier has to do and those which can be made automatically. It can be seen that this depends on the specification language applied. In PDL, for instance, we have a special language level for implementation specification which includes language features not available at the other levels. Thus implementation specification here is a refinement with the help of the new language features. In ESTELLE we do not have such a special language level. In this case the line between the protocol and the implementation specification is often a question of taste unless there is a set of defined rules for this purpose.

Another important problem of implementation specification concerns its structure. Implementation specification often determines the implementation structure. The design of a useful implementation model is therefore a decisive task for computer-aided protocol implementation. We will discuss it below. But most of the specification languages give the specifier much freedom in further structuring their specification. These decisions can heavily influence the implementation. ESTELLE, for instance, provides a complicated module mechanism permitting sophisticated dynamic structures. Such solutions are mostly inefficient and difficult to implement. They should be avoided. Therefore tools should be provided supporting well-defined specification design or permitting a redesigning of the specification, respectively. A graphical tool for representing the module structure can be very useful for this purpose.

4. IMPLEMENTATION MODEL

The implementation model is the base of the implementation process. It determines the structure of the implementation and describes the interaction between its components. There is no unified implementation model for communication protocols. Its structure depends on various factors such as the layer to be implemented, the specification method to be applied, the implementation environment and subjective decisions.

The main problem of computer-aided implementation consists in working out an implementation model which will find a broad acceptance because computer-aided implementation provides only one solution the user must accept or not. This requires a careful analysis of all the factors influencing the implementation and experience with different implementation models as well. It may be possible that future implementation systems will provide several models applicable to different requirements but these will be very complicated to realize. Current systems generally provide only one model.

Figure 1 shows the implementation model we used in our ESTELLE-implementation. It generalizes the approaches proposed in /Serre 86/, /Heymer 84/, /Koenig 87/, /Zschiesche 89/.

The model consists of four basic components: the protocol handler, the lower and upper SAP-handler and the layer manager. The protocol handler creates the proper protocol. For each connection one handler will be established (see below). The lower SAP-handler maps the interaction between the peer entities onto the lower layer service whereas the upper SAP-handler serves for the exchange to the upper layer service. It is usually less complex than the lower one but it has been seen to be useful in many cases. The layer manager establishes the connection to the network management. It controls the different components of the layer and their cooperation including such functions as initialization, function test and error handling. Every component has its own input queue. The implementation model must be mapped onto a practicable process structure. Here are several solutions possible, e.g. one process which includes a subtasking for all components. In our solution we use one process for every component and subprocesses for each protocol handler (connection).

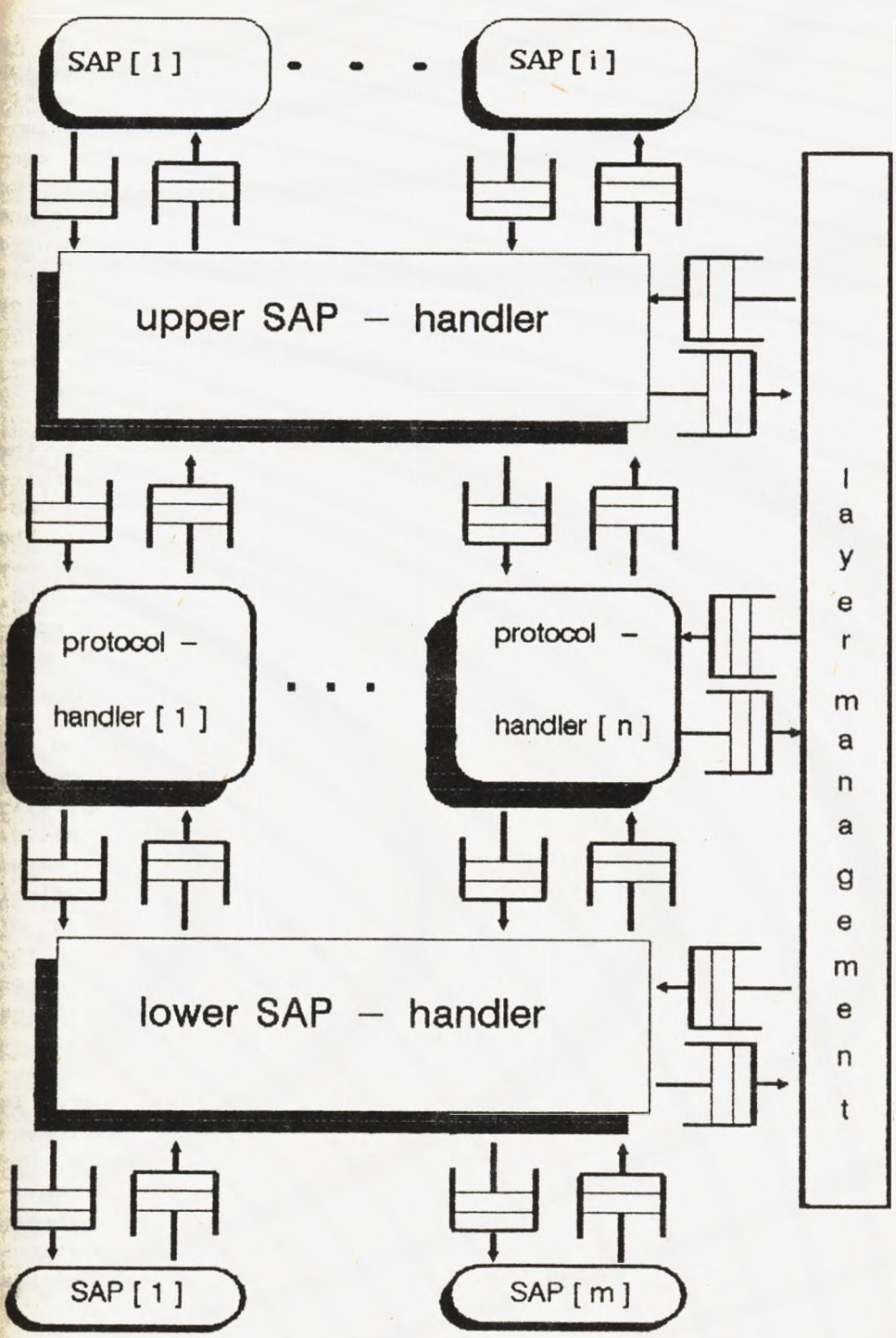


Fig. 1 :Structure of ESTELLE – implementation approach

An important problem in this connection is the question to what extent the protocol or the implementation specification respectively, are to reflect the future impolementation model. All the approaches refered to above use implementation-oriented specifications, i.e. specifications reflecting in a certain manner the future implementation structure. In our ESTELLE-approach we also recommend a specification structure which corresponds to the implementation solution presented above, whereby dynamic and static structuring are possible within the main modules. This recommendation is based on the experience that too complicated structures are cumbersome for computer-aided implementation and can even prevent it. We believe that the use of implementation-oriented specification is an acceptable pragmatic approach based on the complexity of protocols and the gathered implementation experience at the moment. Moreover the use of an unified architecture permits the shifting of some implementation decisions onto a more abstract level /Serre 86/. The stepwise complementation of the specification can be decreased. When is more experience has been gained with other specification techniques, e.g. LOTOS, this approach will certainly be changed. The forgoing discussion about useful specification structures supports this process too.

Finally, we want to discuss another important problem to be solved by finding the best implementation model: connection and event management. The information included in the specification text as the starting base of the compiler is limited. The specification considers only partly the existence of several connections. It assumes the existence of one theoretical connection. It implementation has to carry out the mapping of this virtual connection onto n real ones. The situation for the event management is a similar one. Specifications represent the appearance of events but not their concrete handling (recognition, choice and assignment). These aspects are also a matter for the implementation. All these things cannot be automatically derived from the specification because there is not enough information. These components have to be implemented by hand. It has been shown /Koenig 87/, /Zschiesche 89/ that it is useful to include them in the implementation environment (see below). On the other hand this lack of information is one of the reasons why computer-aided implementations are less efficient than hand-coded ones. The implementor has

to accept the means provided and cannot gain advantages from the detailled knowledge of the buffer and process management used.

5. IMPLEMENTATION ENVIRONMENT

The implementation environment has an important role for the protocol implementation. It carries out the integration into the operating system and the connection to the other layers including the network management. Typical functions of the implementation environment are, for instance, the process and subprocess management to carry out the protocol execution, resource management (buffer and queue management) and time control. For computer-aided implementation it seems to be useful to include connection and event management as discussed above.

The implementation environment can be directly created by the operating system if its functions are powerful enough. In most cases further functions must be provided which form a specific environment situated between the communication software and the operating system (see Figure 2). In computer-aided implementation the environment usually corresponds to the hand-coded part. Its proportion related to the automatically derived part varies between 30 and 50 per cent. Due to the link function the environment has to carry out it cannot be expected that this relation will change on principle.

The degree of automation, however, can be increased considerably by using a unified implementation environment defining a common interface with the communication software. Such an environment "smoothes" the operating system interface (see Figure 3) including possibly an extension or a confinement of the operating system functions. In /Zschiesche 89/ such an environment is described which has been implemented for various operating systems, e.g. MS-DOS and VMS, and practically applied to a hand-implementation of the transport protocol. This environment in a slightly changed form is also used as the kernel of our implementation system.

The use of such a unified environment can increase the proportion of the automatically derived part to almost 100 per cent, if a broad application of the environment is assured. The ef-

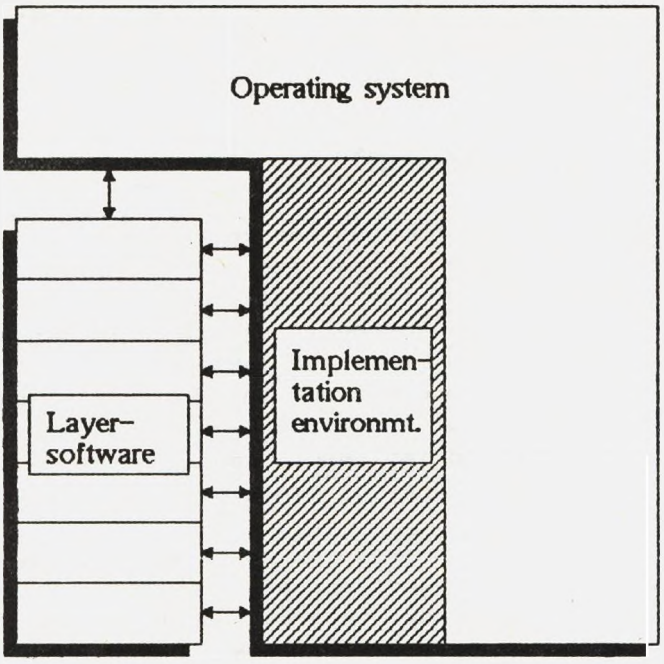


Fig. 2: Implementation environement

fort involved in the hand-coded part is also decreased due to its reuse. The unified implementation environment simplifies the design of the compiler for the communication software shifting most of these problems to the implementation of the environment.

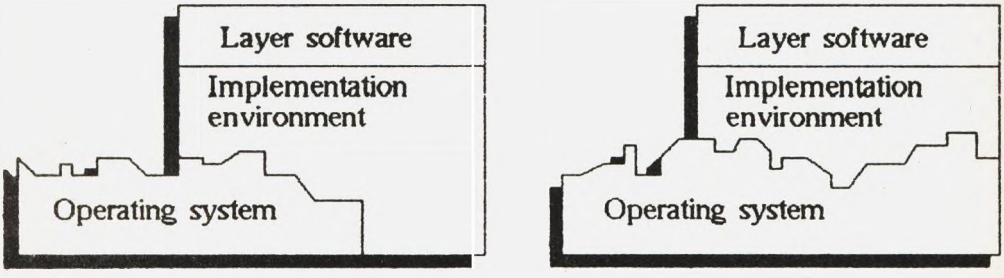


Fig.: 3: "Smoothing" of the operating system interface

6. CONCLUDING REMARKS

The paper has discussed experience in computer-aided implementation. The step to a daily practical application of such systems needs more time, yet. There are still many detailled problems to be solved. Problems which require further investigations are the character of the implementation specification, the elaboration of well-suited implementation models and the optimization of the implementation solutions by means of performance evalution.

REFERENCES

/Aggarwal 85/ Aggarwal,S;Kurshan,R.P.: Automated Implementation from Formal Specification. in Yemini,Y.; Storm,R,;Yemini,S.(eds): Protocol Specification,Testing and Verification, IV. North Holland, Amsterdam, 1985, pp. 127-136

/Heymer 84/ Heymer,V.: Verkopplung von Rechnersystemen zu Rechnernetzen. Diss.B, AdW der DDR, 1984

/Koenig 86/ Koenig,H.;Heiner,M.: The PDL-System - An Unified Approach to the Specification, Verification and Implementation of Protocols. in Csaba,L.;Tarnay,K.;Szentivanyi,T.(eds.): COMPUTER NETWORK USAGE: Recent Experiences. North Holland, Amsterdam, 1986, pp. 567-582

/Koenig 87/ Koenig,H.: Kommunikationsprotokolle - Prinzip und Entwicklung. Diss. B, TU Dresden, Informatikzentrum, 1987

/Koenig 89/ Koenig,H.;Tarnay,K.:A Workstation for Protocol Engineering. in Boyanov,K.;Angelinov.R.(eds.): Network Information Processing Systems. North-Holland, Amsterdam, 1989, pp. 129-137

/Krohmann 89/ Krohmann,O.;Koenig,H.: Spezifikation und Implementierung von Kommunikationsprotokollen auf der Basis von ESTELLE. Tagungsberichte 18. Jahrestagung "Grundlagen der Modellierung und Simulationstechnik", Rostock, 1989

/Serre 86/ Serre,J.M.;Cerny,E.; Bochmann,G.v.: A Methodology for Implementing High Level Communication Protocols. Universite de Monteal,Department d' informatique et de recherche operationelle, Publication #555, 1986

/Zschiesche 89/ Zschiesche,M.: Konzeption und Implementation portabler Komponenten des OSI-Referenzmodells dargestellt an der ISO-gerechten Transportschicht. Diss.A, TU Dresden,Informatikzentrum, 1989

NETWORK LOPOS - PERFORMANCE ANALYSIS

Jan PIŠTĚLÁK and Milan ŠÁREK

Research Institute of Electrical Rotating Machines and Technical University Brno, Czechoslovakia

The submitted paper analyze and evaluates the link protocol of the local area network LOPOS. There are three basic topics: the protocol validation by formal model, evaluation of its characteristics by means of simulation model and performance analysis on real network. The description of network is based on Extended Petri nets. The protocol is decomposed into processes of token circulation, buffer enquiries and acknowledgement transfer. Simulation model is written in SIMSCRIPT II and SIMULA 67 languages.

1. INTRODUCTION

LOPOS represents an original project of a local area network in Czechoslovakia. LOPOS was developed in the Research Institute for Computer Technique (VùVT) Æilina to enable form homogeneous "token-bus" LANs from the 16 bits and 32 bits minicomputers of the SMEP family. Currently 16 bits personal microcomputers PP 06 can be also incorporated into LOPOS network.

As mentioned above, LOPOS is a full original project. It consists of hardware adapter and software modules. The adapter (denotes AKM) is based on Z80 microprocessor with SIO module. Using of these elements determinate transfer rate to 0.5 Mbits/s. Access method "token-passing" is implemented by means of microprogram. Adapter has DMA to the memory of host minicomputer and to the communication medium (standard TV coaxial cable with impedance 75 Ohms). The maximum number of stations is 255.

Communication control rules are covered by network architecture containing five layers: physical layer, link layer, network services layer, presentation layer and application layer. Meaning of all layers is clear from a simple comparison of the LOPOS network architecture to the OSI Reference Model, which is suggested in the fig. 1. This architecture has a significant feature. There is the same interface between network services layer and presentation layer as in the network architecture of the SYRPOS 2 communication system (standard network software for SMEP family of computers). It enables to use all network utilities (corresponding to the presentation and application layers)

from the SYRPOS 2 system on the LOPOS environment and, hence there is an application compatibility between both systems.

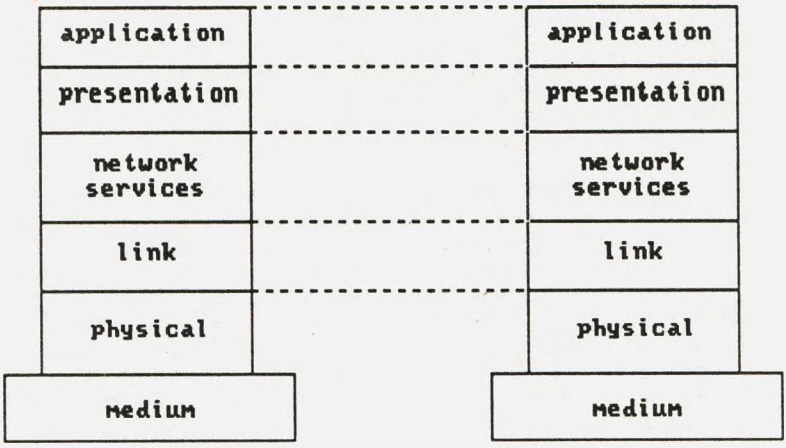


FIGURE 1

Presented paper is dealing with the LOPOS analysis from the point of view of link layer (see chapter 2, concerning with the model of a link protocol by means of a Petri net, and with the performance analysis by means of simulation model) and, outsequently, with performance analysis of a file transfer service on the level of an application layer (chapter 3). The last part of the paper is focused to the short discussion of the acquired results.

2. MODELS OF LINK LAYER

The formal description is based on Extended Petri nets. The protocol is decomposed into processes of token circulations, buffer enquiries and acknowledgement information transfer. An example of token circulations between two stations with acceptation of disturbances is illustrated in figure 2. The characteristics of the protocol are evaluated by means of marking graphs (figure 3).

The simulation model of the link layer of protocol was written in SIMSCRIPT II and implemented on the EC 1045. The simulation model includes six main events:

- end of network initialization,
- end of data transmission,
- generation of new requirement,
- generation of new disturbance,
- release of buffer,
- end of simulation.

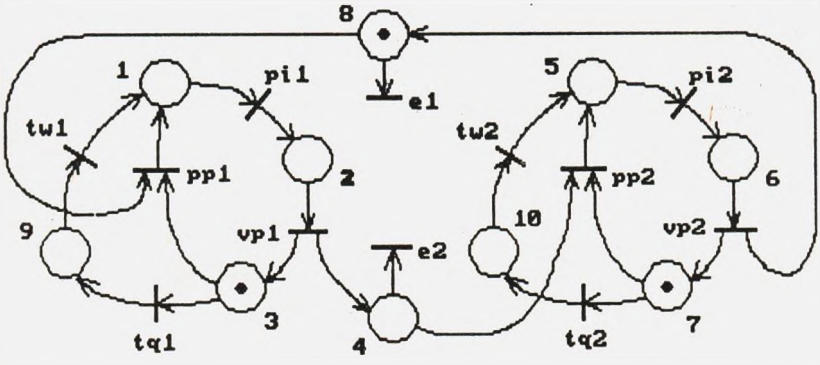


FIGURE 2

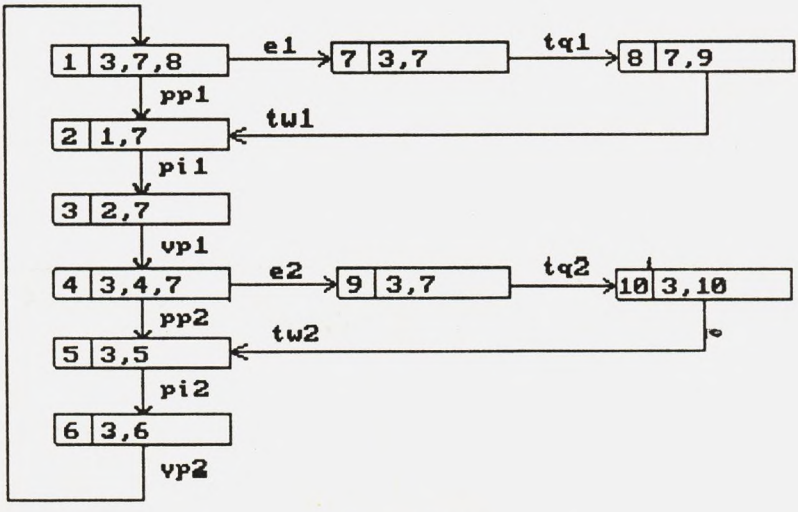


FIGURE 3

Input parameters of simulation model are:

- time of simulation,
- length of transmitted data,
- number of stations,
- medium time between requirements,
- medium time between disturbances.

Output parameters are:

- throughput,
- transport delay,
- complete delay.

We received many results. There are presented only two examples. The first is a throughput as a function depending on number of stations (figure 4). The second is a throughput as a function of time between requirements (figure 5). In both examples are accepted influence of time between disturbances.

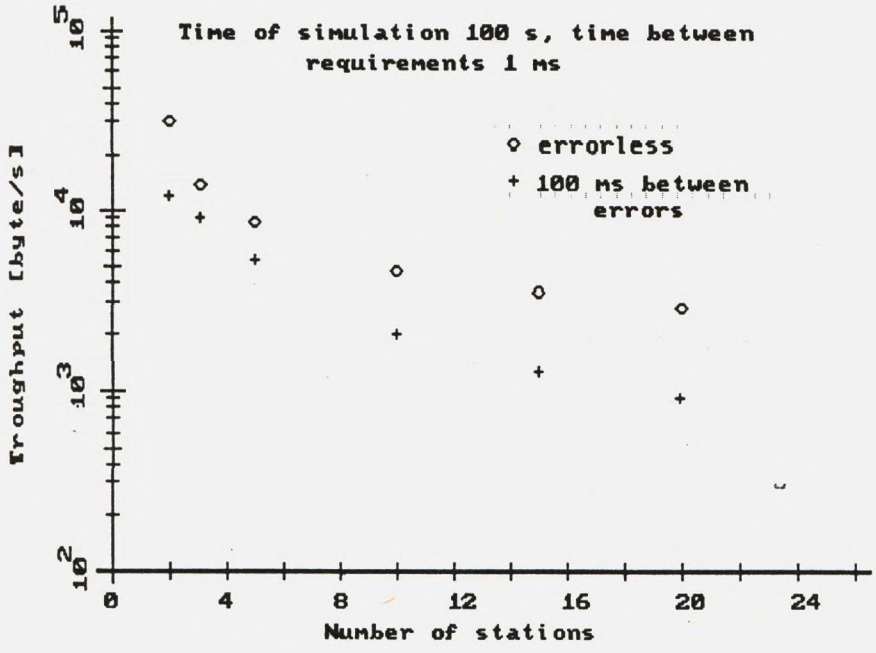


FIGURE 4

The results satisfy a user which is wishing to apply the LAN and wants to know its behavior in typical situations and configurations.

The simulation model was also rewritten to the SIMULA 67 language for the education purpose.

3. PERFORMANCE MEASUREMENT OF THE FILE TRANSFER SERVICE

File transfer service is based upon a mutual communication of two network processes NFT (utility) and FAL (server task). User activates file transfer service by NFT utility in station A and, subsequently, server task FAL will perform required action in the station B. Hence, performance measurement consists in time measurement of files in dependence on the given number of blocks (one block corresponds to 512 bytes). This situation can be described by following equation:

$$T_P = T_{SA} + T_C + T_{SB} \ , \qquad (1)$$

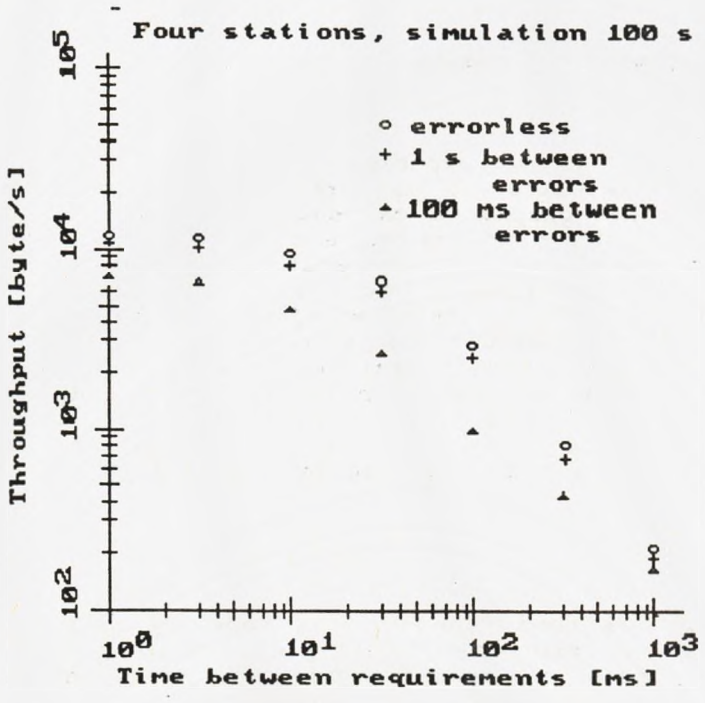


FIGURE 5

where T_C is data transfer time at the level of link layer, T_{SA} and T_{SB} represent overhead of operation system and network software in both communication stations (at the level of higher layers). T_C is associated with used "token-passing" access method, and, therefore, it can be further decomposed as follows:

$$T_C = T_W + T_D ,$$

where T_W is a waiting time for a token, and T_D is a transfer time of data. Both ones can be analytically approximated:

$$T_W = p.k.T_T \quad (k=1,\ldots, n-1), \quad T_D = N/r ,$$

where N is a length of transferred data (including all protocol headers), r is a transfer rate (0.5 Mbit/s) and n is a number of stations. T_T is a transfer time of token and k acquires values from 1 to n-1 according to the position of station within a logical ring. p reflects the fact, that T_p is the time measured at the level of the application layer, therefore, we must consider $k.T_T$ several times. It follows from mutual exchange of start messages between protocol stations in higher layers of the network architecture. Regarding the amount of transferred data, the length of this start messages may be neglected. Now, we can form final equation (2), which approximates T_p:

$$T_p = T_{SA} + T_{SB} + N/r + p.k.T_T \qquad (2)$$

T_R was measured in an experimental LAN with 2, 3 and 4 minicomputers. Results of measurements are in figure 6. Referring to this graph, we can see, that number of station has minimal influence to the value of T_p. On the other hand, station addresses of communication stations (it means incorporation of stations to the logical ring) have significant effect to T_p (transfer time from the station with address 4 to the station with address 1 is about 26% less then transfer time between stations with address 3 and 2). This reality reflects expression $p.k.T_T$ in (2). In the first case (transfer from 4 to 1) k is set to 1, on the second one (transfer from 3 to 2) k is set to 3 (i.e. n-1). Correctness of the equation (2) is validated by essentially linear form of curves.

4. CONCLUSIONS

Our approach has been based upon a top-down specification, which refines the protocol step-by-step. At every step in the iterative process of specification and verification, new situations have been taken into consideration. The formalism adopted is useful, not only for proving the logical correctness of the protocol, but also for making a performance analysis of the system designed under the protocol.

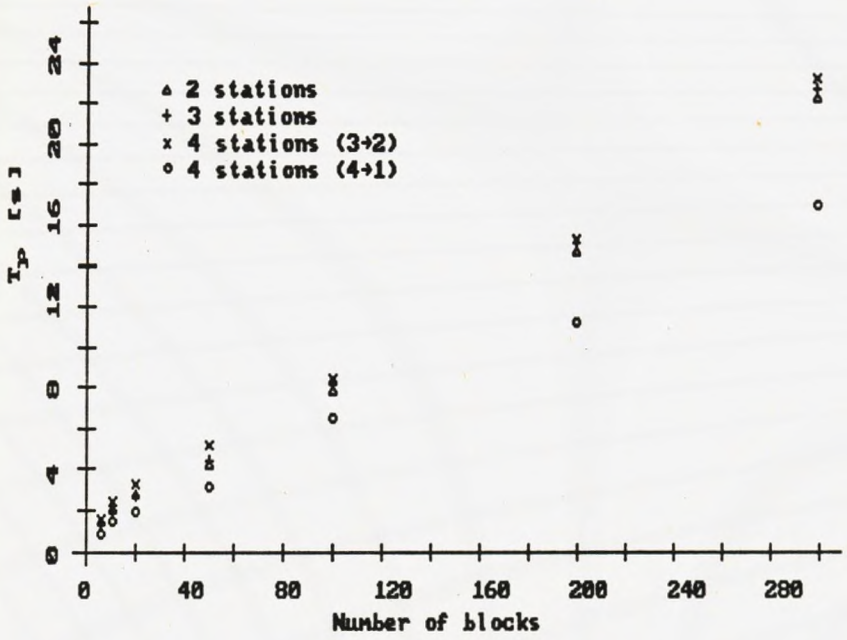


FIGURE 6

Part of results are formulated in a way that showed satisfy a user which is wishing to apply the LAN and wants to know its behavior in typical situations and configurations.

REFERENCES

[1] Sunshine, C., Computer Networks (1982) 6.
[2] Peterson, J.L., Petri net theory and modeling of systems (Prentice-Hall, New Jersey, 1981)
[3] Kotov, V.J., Seti Petri (Nauka, Moskva, 1984)
[4] Šárek, M., Protocol design in medical orientated local area network, in: Roger, F.H., (ed.), Medical Informatics Europe 85 (Springer-Verlag, Berlin, 1985) pp. 780-781.

SESSION 6

KNOWLEDGE BASED SYSTEMS IN COMPUTER COMMUNICATION

META-MANAGEMENT OF COMMUNICATIONS NETWORKS

Branislav Meandzija, William P.-C. Ho***

*University of California,
Riverside, CA 92521-0135

**MetaAccess Inc.
P.O. Box 21956
Sanata Barbara, CA 93121-1956

ABSTRACT

Interconnection of networks with different communications architectures, protocols, and management systems necessitates sophisticated methods and tools for integration on a variety of different integration levels. No general integration methods exist today which are applicable to the integration of architectures, protocols, and systems. In this article we argue that *meta-communication*, i.e. communication about communication rules, is such a general integration methodology. We summarize our efforts towards the development of an automated methodology for meta-communication.

We view meta-communication as a distributed design problem. Meta-communicating entities exchange partially specified communication rules. Each entity, or a meta-communication center, applies a standard composition principle on the individual partially specified rules in order to derive the complete protocol architecture. We report on an existing implementation of the meta-communication mechanism and on translators that transform the resulting specification into C code.

1. INTRODUCTION

As communications networks are moving to multivendor, multicarrier, and multitechnology, network users and administrators have to deal with increased integration complexity. Todays network topologies often include a variety of Local Area and Metropolitan Area networks that are serviced by a variety of Value Added networks and may be interconnected through different Wide Area networks. However, because of the increased integration complexity resulting from diversity, growth, and change, network interconnection is proceeding faster than network integration [7].

In the area of network management, integration problems are so acute that both industry and research have started a great amount of new activity. Meta-Management is a general integration methodology. It is based on **M e t a C o m m u n i c a t i o n**. Meta-communication is the capability of designing, specifying, and implementing communication rules at run-time. This means, communication is enabled solely through the meta-communication mechanism. Initially, the only known common denominator between communication partners is the meta-communication mechanism. That mechanism defines the communications rules and therefore defines the way in which the

components are to be integrated to a system. Thus, a meta-communication method is the basic tool for integration of communications systems. We have developed, [8], [9], and partially implemented, [11], [13], a methodology that automates meta-communication. Meta-Management is an application and extension of Meta-Communication to management systems. Before we look at our results in Section 3 and report on the implementation of meta-communication mechanism in Section 4 and 5, we briefly introduce the subject of network management, its key issues, and the state of the art in Section 2.

2. NETWORK MANAGEMENT

What is network management? Network management is the feedback process of monitoring and controlling a network. Its main objectives are to provide for availability and performance of network resources and to support the user community in whatever way necessary. Network management functions have been classified into five different functional groups by the ISO. These are configuration and name management, fault management, security management, performance management, and accounting management. Management systems available today can be classified into one of the following classes: tools for managing major communications architectures (eg.SNA, DECnet); physical subnet managers (eg. Modem, VSAT, Switch, LAN); specialized performance and security managers; and embedded management (eg. PBX).

Key issues of network management today are protocols, management information bases, analysis of network management data, control of the network, automation of network management functions, application programming interfaces, and presentation of network management information. Most of these issues are not adequately solved by the existing management systems. This forces most organizations to use more than one management system. The integration of these management systems commonly does not allow correlation and aggregation of data of the vendor-specific managers. Commonly, more than one point of control to the network is necessary because of the incompatibility of the different systems used.

There are three major approaches to integrated network management systems today. The first approach would produce the ideal integrated network management system but hasn't been used in any existing systems because of a lack of an appropriate underlying integration method. This approach is to simply have one integrated manager take charge of all devices in the network. This is technically difficult because of the great diversity of device types and manufacturers who use their own protocols. The first approach is illustrated in Figure 1.

The second approach is to integrate device managers and not the devices themselves. This approach has been used in products such as IBM NetView and AT&T Unified Network Management Architecture Integrator. It is illustrated in Figure 2.

The third approach is only slightly different from the second one as it allows for multiple integrated network managers and interaction between them. AT&T's UMA Integrator allows such an interaction.

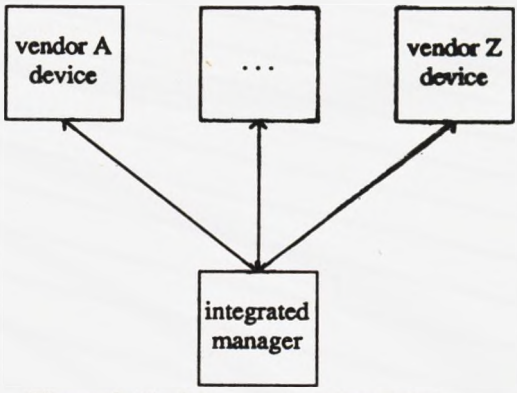


Figure 1: Ideal Approach to Integration

All currently existing approaches impose a common denominator platform and expect others to conform. As there is a number of different platforms, confusion is the result. In summary we can conclude that there are no off-the-shelf solutions for complex heterogeneous nets, that there is only limited interoperability, and that rapid technology cycles make integration/automation efforts difficult. However, diversity, growth, and change demand new solutions for integration of network management systems and network integration in general.

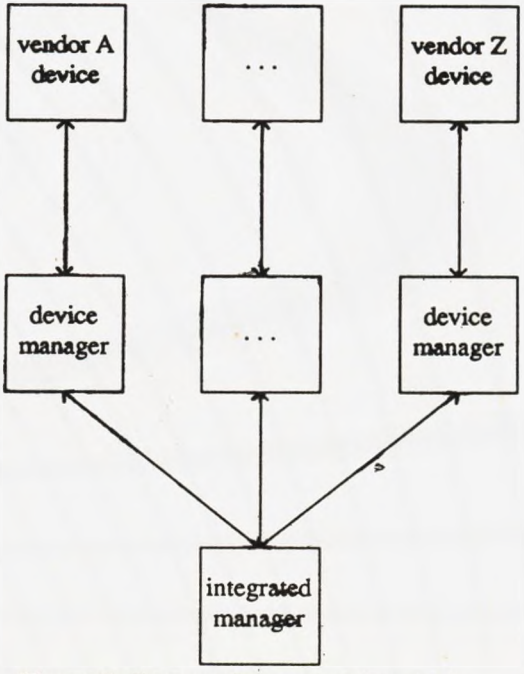


Figure 2: Common Integration Approach

3. META-MANAGEMENT

The meta-management/meta-communication approach allows us to *postpone the system design decisions until all communication characteristics are known.* This means we can solve each communication task in the most efficient way known. Thus, the meta-communication system is flexible to environments and applications. This should not only result in improved performance but is the key to solving integration problems. Another advantage of flexibility is the adjustability to changes in communications technologies. This should result in longevity of the design and implementation. Finally, it should be noted that meta-communication as it is defined in this paper can be easily modified to solve protocol conversion problems as defined in [4] and [5].

We view meta-communication as a distributed design problem. Meta-communicating entities exchange partially specified communication rules. Each entity, or a meta-communication center, apply a standard composition principle on the individual partially specified rules to derive the complete system architecture.

The system architecture (both, system specifications and system architectures of all communications sides, are represented) is specified with the abstract specification language of Archetype [10]. Partial specifications are given as predicates on the set of all Archetype specifications. Thus, each meta-communicating entity formulates constraints on the set of acceptable system architectures. Each entity meta-communicates its constraints to each other entity (or to a center). All entities (or the center) apply the same constraint composition principle on the accumulated constraints to deduce a set of constraints that specifies a single system architecture. All entities compile this specification into executable programs that are subsequently automatically loaded to drive the protocols of the specified system architecture.

Abstract specifications of Archetype are not directly translated into executable programs. This wouldn't be practical from the software engineering standpoint because of the very high level of abstraction of those specifications which was necessitated by the meta-communication methodology. The abstract specification language is descriptive and can be used to express a "Natural Language"-like specification of the system architecture. An abstract specification is transformed into a specification in the second language of Archetype, the data-driven, concurrent, executable specification language. A translator for the abstract specification language is being developed. A translator of the executable specification language into C has been implemented on UNIX 4.3. and SYSTEM V and has successfully produced protocol drivers for SUN workstations, the TI Explorer, and a VAX 11/780.

The crucial part in the meta-communication mechanism is the combination of constraints and the deduction of the system architecture. We have implemented this process using LISP and have successfully deduced system architecture specifications by combining sets of constraints. Figure 3 illustrates the entire meta-communication architecture.

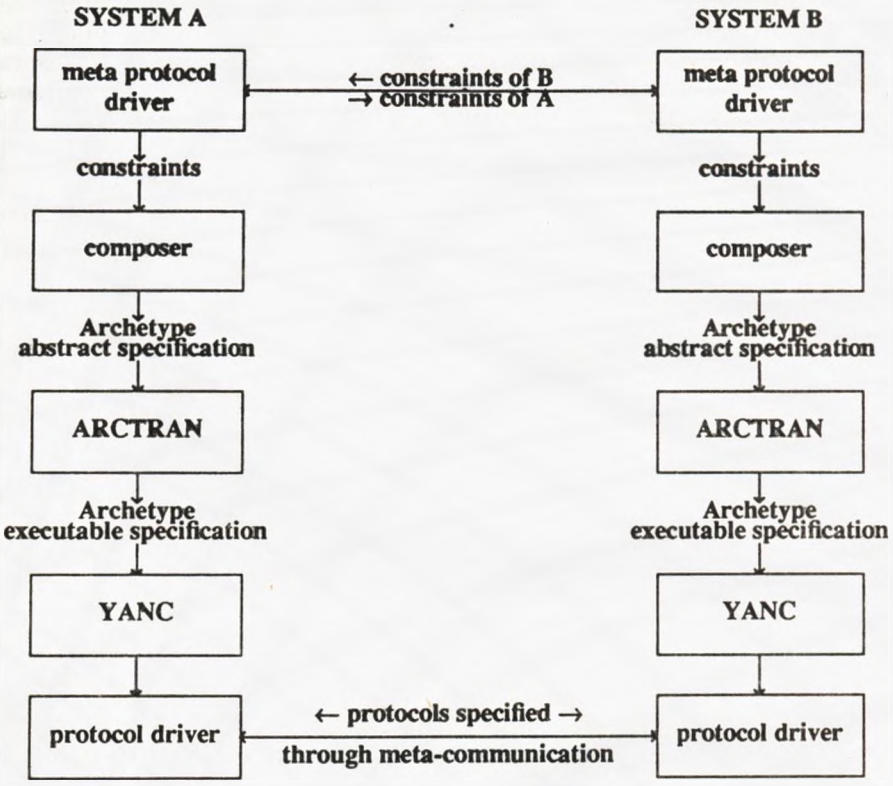


Figure 3: Meta-Communication Architecture

4. DESIGN LANGUAGE

The design language of Archetype is suitable for the automatic design process because of its abstract descriptive nature. A more conventional protocol specification technique such as the state-transition approaches [2] would make the specification of constraints for the design process complex because of the level of detail required.

An abstract specification is represented as a tree. Each node of the tree is labeled with an archetype, which dictates the shape of the subtree. The subtree of a node is the refinement of that node's archetype. The tree is specified as a list structure. Each node is represented as a list in which each child node is represented (recursively) as a sublist. Thus, the tree is structured as a nested list.

Each node is labeled by an archetype. The archetype can be refined by a subtree, by an explicit refinement specification (will be explicated later), or can be structured with the substructures possibly refined.

The label MEDIUM is used to indicate the axis along which exchange of information units is to take place. Everything below a node labeled MEDIUM is regarded as one, possibly distributed, information processing system.

Each archetype consists of a description and, optionally, an explicit refinement. Three types of descriptions are distinguished. The atomic description is of significance only to the node for which it is the label. Currently, we distinguish only between two types of functions of atomic descriptions. The purpose of DATA is to indicate information which, at that particular level of interpretation, is not to be analyzed any further. The purpose of NEGOTIATE(<id>) is to indicate a piece of information that is to be identified by <id> and dynamically negotiated among communication partners.

A thorough introduction to Archetype can be found in [10]. The application of Archetype to meta-communication has been reported on in [11]. Figure 4 illustrates a simple, incomplete, design specification in Archetype.

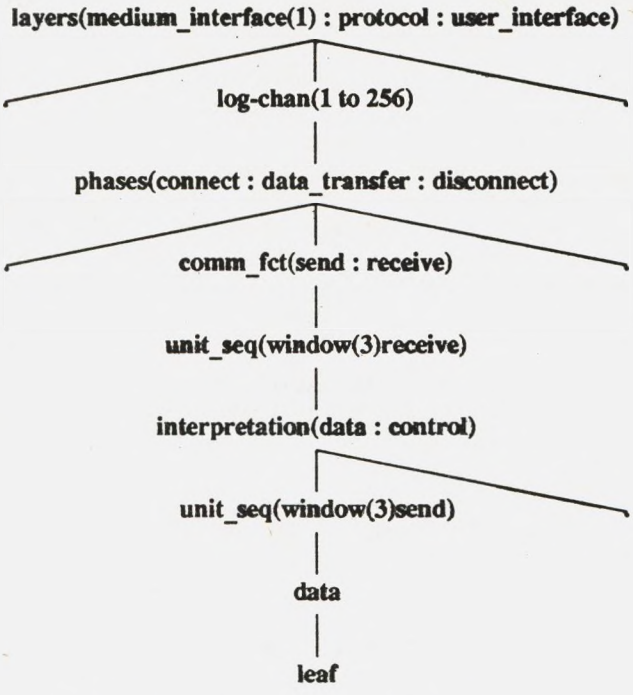


Figure 4: Archetype Abstract Specification

Archetype, as it is currently implemented, is not sufficient for Meta-Management. It needs to be extended to describe Network-Management systems architectures and atomic operations. The following syntax would provide such an extension for the atomic operations. It is based on the BBN Monitoring and Control protocol.

Operation	=	Control-op | Monit-op
Control-op	=	Component-op | Item-op
Component-op	::	(START | STOP | RESET) Vd-id Comp-id
Item-op	=	(READ | WRITE) Vd-id Comp-id Item-id
Monit-op	::	Monit-event Vd-id Comp-id Item-id
Monit-event	=	$(\text{Time})^{+}$ | Change
Change	::	From To
From	=	To = Items
Resources	=	Vd-id → Virt-device
Virt-device	=	(Comp-id → (Component → (Item-id → Item)
Component	=	Processors | Protocols | Soft-modules | Hard-modules | Assoc-dev | Data-struct | Activities | The-whole-dev | Ports
Items	=	State-var | Control-var | Config-item | Event-buff | Data-struct-value

The above syntax gives a basic idea of the type of Archetype extension necessary for Meta-Management. This extension is simple as the object of design is not a complex protocol driver but interrelations between sequences of high-level commands.

5. DISTRIBUTED DESIGN OF A PROTOCOL ARCHITECTURE

A partial specification of a protocol architecture is given by defining a set of predicates on all the syntactic domains of the grammar of Archetype. We can have multiple predicates for the same domain, which could represent distinct points of view or subsequent changes of the same point of view. Those predicates are acted on by a composition function in attempting to construct a single specification tree. The general methodology has been formally defined using [1] (see [9]).

5.1. THE PROTOTYPE

The currently existing prototype is written in LISP and operates on predicates that are written in LISP (for a complete description see [11]). A predicate formulates the constraints on the usage of an archetype. A meta-communicating entity specifies constraints on the usage of all archetypes. The weakest constraints are those which are formulated as predicates always yielding the value TRUE. The strongest constraints are those which are formulated as predicates that return true to a unique value in a unique position within the tree.

The prototype consists of two main modules. The *Candidate Value Extractor* constructs the set of candidate values from the predicates and stores those values in the *Candidate Value Base*. The *Synthesizer* is the core of the prototype. It recursively constructs the protocol architecture by using the *Candidate Value Base* to satisfy the composed predicates and dynamically maintaining a run-time environment. Starting at the root, the following steps are performed for each node to be generated:

- pick a candidate value belonging to a particular syntactic domain and label the node
  temporarily with that value

- apply the composed predicate on that node

- if composed predicate holds
  then label the node permanently and proceed with next node
  else if there are other candidate values for the same domain,
    then make the candidate value unavailable for this node and start
      with step 1 again
    else make this domain unavailable for this node and start with
      step one for the next domain

- if all domains have been tried unsuccessfully, then no specification can be generated

The information explicitly passed to each node in generation is the set of archetype descriptions and the set of archetype subtypes which have been applied along the path prior to that node. Once a leaf has been reached, the control falls back through the recursion levels until a node is reached for which not all of the children have been generated.

5.2. PREDICATE FORMULATION

Each predicate specifies four types of constraints: the constraints on the values of the variable parts of the archetype, the constraints on the archetypes used before this archetype, the constraints on the archetypes used in other subtrees, and the logical interrelations between those constraints.

All specified predicates are combined by the *composer* which uses the *composition function* to logically deduce a single protocol architecture. The *composer* does this by successively constructing nodes of the Archetype specification tree in a leftmost, depth first fashion. Therefore, predicates essentially express two types of context constraints on the usage of archetypes in the construction of the final specification tree. These are constraints with respect to predecessor nodes (vertical constraints) and constraints with respect to sibling nodes (horizontal constraints).

Vertical constraints are formulated based on a path history (used-at and used-st variables record the archetypes and their subtypes used in the nodes of a path). Figures 5 and 6 illustrate this.

```
(or (not (member 'PHASES used-at ) )
    (and (member 'PHASES used-at )
         (member 'DATA_TRANSFER used-st )
         constraint ) )
```

Figure 5: Vertical Constraint Example: restricting sequencing

The LISP function in Figure 5 defines a *constraint* in the specification (usage) of the data-transfer phase of a protocol. Graphically, this function defines two types of predecessor - successor relationships:

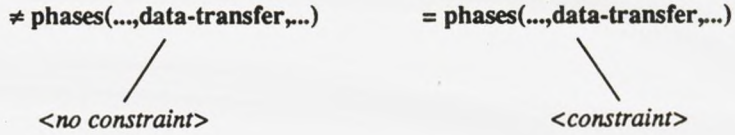


Figure 6: Vertical Constraint Example

Horizontal constraints formulations are independent of the sequence of node constructions by the composer. A dependency of the usage of archetypes in sibling nodes is expressed by allowing an explicit evaluation of the composition function on an artificially constructed node. This enables the out_of_sequence determination of the archetype that will be selected in any node. Therefore, any archetypes usage can be constrained dependent on the usage of archetypes in other nodes even though there is no predecessor-successor relationship between the nodes. Figure 7 defines a complete predicate for the PHASES archetype that formulates the horizontal dependency of usage of archetypes in layers and interfaces.

```
(defun is-wf-pha (node_id at st )
(or( and (member 'PROTOCOL st)
        (comp comp_opr '(is-wf-pha2 is-wf-pha3) node_id
            (append (lsub at '(LOG_CHAN) )
                    '(PHASES ADDR_MAP STREAM_MAP) )
            (append (lsub st '(PROTOCOL) )
                    '(USER_INTERFACE NORM_MODE)) ))
        (not (member 'PROTOCOL st ) ))
```

Figure 7: Horizontal Constraint Example: layer interdependency

A PHASES archetype can be used in the PROTOCOL subtree if and only if it is also used in the USER_INTERFACE subtree. The variable **comp_opr** is bound to the composition function (e.g. logical AND, OR, etc., or any other function returning a boolean value). The function **comp** is defined below. Figure 8 gives a graphic representation of the predicate of Figure 7.

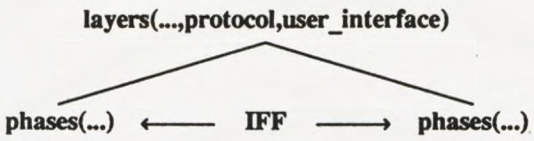


Figure 8: Horizontal Constraint Example

The composition function itself is easily defined because of LISP's ability to use functions as data. Predicate functions of meta-communicating entities are given as arguments to the composition function which uses the LISP primitive *apply* to apply the composition principle (simple logical operator or any other type of LISP function) on those predicate functions to construct the composed predicate.

```
(defun comp (comp_opr pfunc node_id at st )
        (setq pfunc (fcreat pfunc '(node_id at st) ) )
    (apply comp_opr pfunc )
)
```

Figure 9: Composition Function

6. CONCLUSION

We have introduced a methodology for automating meta-management/meta-communication for the purpose of designing, specifying, and implementing communication rules at run-time. The main purpose of the meta-communicating communications model is the integration of communications architectures, protocols, and management systems. All currently existing approaches to integration of communications systems are of an ad-hoc nature (see [12]).

We have implemented crucial substeps of the meta-communication methodology and will demonstrate a complete system in the near future. The composition of partial specifications to a complete specification of a protocol architecture has been successfully implemented and tested [11]. The translator for the Archetype executable specification language has also been implemented and has been used to automatically generate protocol drivers [13]. We are implementing the translator for the Archetype abstract specification language, and will be porting both translators to operating systems other than UNIX.

Each of the steps in the meta-communication methodology is under continuing investigation. On the topmost level we would like to provide a user-friendly way of specifying constraints that represent the knowledge of a meta-communicating entity. This could be accomplished by a knowledge-based system that automatically constructs these constraints from formalized application requirements and environment constraints. A possible approach to this problem is given in [6]. On the level of combining constraints further research is needed to clarify the relationship: composition function ←→ predicates. Also, both the abstract and the executable specification language of Archetype have only partially defined interfaces to surrounding environments. These and other issues are subject of currently ongoing research.

In summary, we believe that future communications systems will be based on automatic meta-communication. We have demonstrated the feasibility of the meta-communication approach through our methodology and through implementations.

REFERENCES

[1] Bjorner D., Jones C. B. *Formal specification & software development.* Prentice-Hall International, 1982.

[2] Bochmann G. V., Sunshine C. A. A survey of formal methods. In *Computer Network Architecture and Protocols*, Plenum Press, 1982, 561-579.

[3] Conry S., Distributed artificial intelligence in communications systems. Proceedings *IEEE Expert Systems in Government Symposium*, 1986.

[4] Green P. Protocol conversion. *IEEE Transactions on Communications.* March 1986.

[5] Groenback I. Conversion between TCP and ISO protocols as a method of achieving interoperability between data communications systems. *Journal of Selected Areas in Communications.* March 1986.

[6] Hayward N., Meandzija B. Deriving protocol architecture specifications from formalized network application service requirements and environment constraints. SMU-CSE, 87-CSE-3, (January 87).

[7] Herman J. Integrated Network Management - A Tutorial. *First International Symposium on Integrated Network Management.* May, 1989.

[8] Meandzija B. Archetype: a unified method for the design and implementation of protocol architectures. *IEEE Transactions on Software Engineering*, Vol. 14, No. 6, June 1988.

[9] Meandzija B., Ho W. P.-C. Towards truly open systems. *Proceedings IFIP International Symposium on Protocol Specification, Testing, and Verification* (Eds. H. Rudin, C. West), (May 1987). North-Holland, 1987.

[10] Meandzija B.: Design and automated generation of protocol architectures Using Archetype. *Proceedings 7-th International Conference on Distributed Computing Systems.* September 1987.

[11] Meandzija B., Ho W. P.-C., Banerjee S. Distributed Design through Stepwise Refinement. *Proceedings IFIP Symposium on Protocol Specification, Testing, and Verification* (Eds. S. Aggarwal, K. Sabnani). North-Holland, 1988.

[12] Meandzija B., Westcott J. (Eds.). Integrated Network Management, I. *Proceedings IFIP Symposium on Integrated Network Management.* North-Holland, 1989.

[13] Radigan J., Meandzija B. Y.A.N.C. (Yet Another Network Compiler). *Second International Symposium on Interoperable Information Systems.* November 1988.

Distributed/Cooperative Office

László Csaba[1)]
and
Erzsébet Csuhaj-Varjú[2)]

[1,2)]Computer and Automation Institute
Hungarian Academy of Sciences
H-1132 Budapest
Victor Hugo u.18-22
Hungary

Abstract

Motivated by the theory of distributed /cooperative problem solving systems we introduce a formal concept of a society model of office information systems. We define some notions to measure and characterize descriptive and behavioural properties of offices and illustrate them by presenting a result having comparative character.

1. Introduction

Computer supported collective activities have gained increasing interest recently. The necessity of developing intelligent general tools for handling theoretical and practical problems of cooperative and distributed activities in offices has obtained an emphasized importance. Office information systems have different models and approaches of description (see e.g.[1]), which express different aspects, as procedural [2,3], declarative [4] or object-oriented [5]. In [6] artificial intelligent techniques are used, where an office task execution is considered to be a problem solving action. Office information systems as societes are modelled in [7]. Based on the observation that usual computerized cooperative/distributed activities in offices, as real-time computer conferencing, distributed/cooperative text processing, meeting organizement, electronic bulletin board or electronic mail can be considered as distributed/cooperative problem solving actions showing common formal similarity, [8] introduces a highly formalized description for the model of [7]. In this paper we develop this model of by considering offices as societies of cooperating and communicating agents, which solve problems together. We present a formal definition of the office society, motivated by a generative paradigm of [9]. We propose some size and structural complexity measures and state some results of comparative character.

2. Office Societies

Activities, processes, relations in offices are versions of intelligent activities, processes and relations in communities.

By an **office** we mean a collection of functionally related **office societies,** which are communities of cooperating and communicating **office agents.** Office agents are intelligent office entities, represented by knowledge sources. The basic general rules of the *cooperation* and the *communication* are determined by collective bargaining by the agents and are fixed in the **social** and **communication agreements**, respectively, which are represented by mappings.

Office societies have **problems** to be solved and **goals** to be achieved. Goals are understood as missions, which are achieved by solving some well-defined problems.

The **solution** of the office problem is the result of the **problem solving process**, which consists of consecutive steps and to which a common abstract space, called the **office board** is associated. The office board consists of the **workboard** and a finite set of **communication boards.** The workboard contains the current partial solution(s) of the problem and the communication boards are for storing those information which concerned with the problem solving process and are intended to be exhanged by the agents. Each communication board is dedicated to a group of agents for privilegized levels of usage. Note that the workboard and the communication boards are handled in accordance with the social- and the communication agreement.

The problem solving process starts in an *initial state of the office board* (**problem initialization**). Then it is continued by consecutive **actions** of the office agents, up to achieve the solution. An action consists of modification of the contents of the workboard and that of some communication boards. There are *enability conditions* which determine the agent(s) executing the action. These conditions can be given in the form of an outer (static or dynamic) *control*, being fixed in the social- and the communication agreement, or can be given inside the agent. The process ends in a previously defined state of the workboard, accepted as the solution.

During the problem solving process the office board contains *strictly organized forms of nondecomposable knowledge pieces*, which are called **elementary office knowledge pieces** (office information pieces). We make distinction between elementary office knowledge pieces having **terminal** and **nonterminal character**. Terminal elementary office knowledge pieces convey full information (which cannot be developed) from the point of view of the agent, nonterminal elementary office knowledge pieces should be or can be developed. We assume the office problem solution and the partial

solutions to be built up from office knowledge pieces, that is, strictly organized forms of elementary office knowledge pieces.

Office agents, as knowledge sources, are considered to be *rule-based knowledge bases* with the ability of activation. The rules have condition-action format, where condition concerns the current state of the office board and the action is the set of modification steps executed by the agent on the workboard and on the communication boards (on each board at most one step). The *ability of activation* is determined as a set of conditions concerning the problem solving process and the content of the office board.

Usually, office agents can solve a lot of problems together. The collection of office problem solutions of office problems which can be solved by the same office society gives it **office problem solution space.**

In the following we give some examples for the possibility of the usage of this society model for intelligent distributed/cooperative office activities.

Examples:

Distributed/cooperative text preparation by using the facilities of an electronic mail system.

The persons writing texts together correspond to the agents, the mailboxes to the communication boards, the current version of the text to the workboard. The office actions are the corrections of the text and the handling of the messages through the mailboxes. For example, different distribution lists correspond to social- and communication agreements. An office problem is the preparation of a text about a fixed topic (with a fixed title). The office problem solution space is the collection of those texts which can be produced by this community. Office knowledge pieces are reasonable units of texts (sentences, paragraphs).

Computer conferencing.

The participants correspond to the agents. There are no separated communication boards, the common meassage space corresponds both to the workboard and the communication boards. A message is an office knowledge piece. Fixed topics of the conversation can be considered as problems.

3. Measure and classification

From the point of view of the *organization* and the *construction* of the computerized office we need information about the **size**, the **structure** and the **behaviour** of the office society.

The classification (classes of) office societies according to some formally *well-defined measures* is an essential point.

In the following we propose some ideas - some of them were mentioned in [8] - to measure size, structural and behavioural properties of office societies.

Size measures of an office society are connected with the office agents, with the communication boards and with the social- and communication agreements.
The following measures are of obvious sense:

- the number of office agents,
- the maximal number of rules of the office agents,
- the maximal number of nonterminal (terminal) office knowledge pieces occurring in the set of rules of an office agent.

The *size of a communication board* can be characterized by the *maximal number of total occurrence of elementary office knowledge pieces* in the board during the different problem solving processes of the society.

Structural measures refer to the process of problem solving and characterize the interactions of the agents. Such measure is for example: the *number of groups of strictly cooperating agents*, where by strict cooperation we mean the following: two agents are in strict cooperation if there is a problem solving process in the society in which the agents develop a knowledge piece produced by the other one. To determine independent groups of agents of a society is of obvious sense.

Another important structural complexity measure is the *maximum of the total occurrence of nonterminal elementary office knowledge pieces* in the partial solutions in the *workboard* during the problem solving processes. This measure can be interpreted as the number of *open questions* in the current step in the workboard.

Size and structural measures can be extended for classes of office societies having the same office problem solution space by considering the minimal value of the corresponding measure in the class.

Another important point of classification is the classification of the office societies according to the **strategies** applied by (groups of) agents in the different problem solving processes. Strategies can be *collective* and *self-strategies*. Collective strategies are concerned with the collective behaviour of the agents and are fixed, in some cases, by the social agreement. Self-strategies are characteristic only for the agent.

Without the aim of completeness we give some examples for self-and collective strategies. Self-strategies can be , for example, so called "minimal comment" and "full comment" strategy. In the first case the office agent modifies one office knowledge piece in the office board, and in the second case he/she modifies all knowledge pieces which he/she thinks necessary to be modified. Another self-strategy can be, for example, the so-called "minimal message" strategy, when the agent modifies such knowledge

piece in the communication board, which is built up minimal number of elementary knowledge pieces.

The following cases are examples for collective strategies: first, the strategy of step number limitation, when every office agent can execute consecutively (at least, at most, exactly) a fixed number of actions; the strategy of comparative competence, when the office agent performs the action which is the most (the least, of measure of a fixed value) competent, where the competence expresses the ability of modification of the workboard.

A collective strategy can be based on the limitation of the communication steps done by the agents, too. An interesting example for collective strategies is the case of temporaly monopolization, which means that until an office agent satisfies the conditions of the strategy then it performs the next action. Note that collective strategies raise such important questions as *fairness* and *efficiency*.

4. A formal model

In the following we present a formal definition of the office society and the office problem solving process.

We first define a notation.

Notation 1.

The elementary office knowledge piece with no information (the empty office knowledge piece) is denoted by ε. For F being a set of elementary office knowledge pieces we denote by $K(F)$ the set of office knowledge pieces built from F. By $K_\varepsilon(F)$ we denote the union of $K(F)$ and $\{\varepsilon\}$. The family of n-tuples ($n \geq 1$) of elements of $K(F)$ ($K_\varepsilon(F)$) is denoted by $K^{(n)}(F)$ ($K_\varepsilon^{(n)}(F)$).

In the following we present the notion of an office agent.

Definition 1.

An office agent (with m communication boards, where $m \geq 1$), is a 4-tuple $A = (N, T, R, m)$, where

(i) N and T are finite disjoint sets of elementary office knowledge pieces, called nonterminal and terminal office knowledge pieces, respectively,

(ii) R is a finite set of rules having the form

r : **if** $cond_r(\underline{w}) = \underline{true}$ **then** change $\underline{w}$ for $modif_r(\underline{w})$, **else** do nothing; where

$cond_r : K_\varepsilon^{(m+1)}(N \cup T) \rightarrow \{\underline{true}, \underline{false}\}$ and

$modif_r : K_\varepsilon^{(m+1)}(N \cup T) \rightarrow K_\varepsilon^{(m+1)}(N \cup T)$.

Note that different rules can have the same condition. Mapping $cond_r$ refers to the state of the office board before performing an action and mapping $modif_r$ defines the modifications. Note that the definition does not mean effective reading and modification, pseudo (empty) actions are assumed to be possible, too.

The notion of an office society is defined as follows.

Definition 2.

An office society with n office agents and m communication boards, $n, m \geq 1$ is a 7-tuple $O = (\mathcal{A}, \sigma, \delta, \pi, \tau, n, m)$, where

(i) $\mathcal{A} = \{A_i : 1 \leq i \leq n\}$, where $A_i = (N_i, T_i, R_i, m)$, is an office agent, where for $i \neq j, 1 \leq i, j \leq n$, $(N_i \cup T_i) \cap N_j = \emptyset$,

(ii) $\sigma : \mathcal{A} \times K_\varepsilon^{(m+1)}(N \cup T) \rightarrow 2^{\mathcal{A}}$, where σ is said to be the social agreement,

(iii) $\delta : \mathcal{A} \times K_\varepsilon^{(m+1)}(N \cup T) \rightarrow 2^{\{1,2,\ldots,m\}}$, where δ is said to be the communication agreement,

(iv) $\pi \subseteq \{\underline{s} : \underline{s} = (s_0, \ldots, s_m),\ s_0 \in K(N \cup T),\ s_i \in K_\varepsilon(N \cup T),\ 1 \leq i \leq m\}$, where π is called the set of problem initializations of O,

(v) $\tau : \pi \rightarrow 2^{K_\varepsilon(T)}$ is the mapping defining the acceptable solutions to the problem initializations.

Note that σ (the social agreement) defines the agent(s) performing the next action and δ defines the set of communication boards which are effectively modified. These notions express dynamic control. Knowledge pieces being in the initial states of the workboard and the communication boards (fixed by π) initialize the problem to be solved. The mapping τ is for selecting problem solutions which are acceptable from the point of view of the society.

In the following we define formally, what the problem solving process means.

Definition 3.

Let $A = (N, T, R, m)$, $m \geq 1$, be an office agent and let $\underline{w}_1, \underline{w}_2 \in K_\varepsilon^{(m+1)}(N \cup T)$. We say $\underline{w}_2$ originates from $\underline{w}_1$ by performing an action by A, denoted by $\underline{w}_1 \Longrightarrow \underline{w}_2$, if A has a rule $r \in R$ such that $cond_r(\underline{w}_1) = \underline{true}$ and $\underline{w}_2 = modif_r(\underline{w}_1)$ holds.

Definition 4.

Let $O = (\mathcal{A}, \sigma, \delta, \pi, \tau, n, m)$, $m, n \geq 1$, be an office society and $\underline{w}_i \in K_\varepsilon^{(m+1)}(N \cup T)$, where $i \geq 0$.

We say $\underline{w}_0 \Longrightarrow \underline{w}_1 \Longrightarrow \ldots \Longrightarrow \underline{w}_r$ is a problem solving process (of r steps) in O iff the following hold:

(i) $\underline{w}_0 \in \pi$,

(ii) $\underline{w}_r \in \tau(\underline{w}_0)$,

(iii) for each $i, 1 \leq i \leq r$, there is an $A_{j_i} \in \mathcal{A}$ such that $\underline{w}_i$ originates from $\underline{w}_{i-1}$ by performing an action by A_{j_i}, $A_{j_i} \in \sigma(A_{j_{i-1}}, \underline{w}_{i-1})$ and for $k \neq 0$ and $k \notin \delta(A_{j_{i-1}}, \underline{w}_{i-1})$ it holds that $\underline{w}_i^{(k)} = \underline{w}_{i-1}^{(k)}$, where $\underline{w}^{(k)}$ denotes the k-th element of $\underline{w}$.

For $\underline{y}, \underline{x} \in K_\varepsilon^{(m+1)}$ with $x \Longrightarrow \ldots \Longrightarrow^{(r)} \underline{y}$ for some $r \geq 0$ we use denotation $\underline{x} \Longrightarrow^* \underline{y}$.

Thus, a problem solving process is a series of actions performed by some agents, in accordance with the social and the communication agreements. It starts in the initial state of the workboard and the communication boards and ends by obtaining a knowledge piece being acceptable solution. Note that we can require communication boards to be empty at the end of the problem solving process.

Using the notion of a problem solving process we can define the problem solution space of an office society.

Definition 5.

Let $O = (\mathcal{A}, \sigma, \delta, \pi, \tau, n, m)$, where $n, m \geq 1$, be an office society. The problem solution space of O is defined as $C(O) = \{w : \underline{w} = (w, \varepsilon, \ldots, \varepsilon) \in K_\varepsilon^{(m+1)}(T), \underline{p} \Longrightarrow^* \underline{w}, \underline{p} \in \pi, \underline{w} \in \tau(\underline{p})\}$.
The problem solution space of the office society is the set of those terminal office knowledge pieces, which are contents of the workboard and are obtained from some problem initializations by a problem solving process and are acceptable solutions to these problem initializations.

Finally, we introduce a notation expressing (size) descriptive complexity.

Definition 6.

The collection of office societies is denoted by O.
We denote by $\mathcal{C}(\mathcal{O})$ the set $\{C(O) : O \in \mathcal{O}\}$.
For n, m, k being natural numbers we denote by $\mathcal{O}(n, m, k)$ the collection of office societies with n office agents, m communication boards and k nonterminal elementary office knowledge pieces.
Let $\mathcal{C}(\mathcal{O}(n, m, k)) = \{C(O) : O \in \mathcal{O}(n, m, k)\}$.

5. Results – a partial case

In [8] some results concerning the cases when office agents do not use communication boards during the problem solving process (the classical blackboard type problem solving) were presented.

In this section we illustrate the previously introduced notions by establishing some results concerning those types of office societies which consist of office agents expressing pure sysntactic aspects.

We first introduce a notion, specializing the notion of an office agent.

Definition 7.

An office agent $A = (N, T, R, m)$, where $m \geq 1$, is said to be syntactic if R consists of a finite number of productions of the form
r : *if* w_0 occurs in the workboard and w_i occurs in the i-th communication board for $1 \leq i \leq m$ *then* replace w_0 by w_0' and add w_i' as suffix to w_i for $1 \leq i \leq m$, *else* do nothing, where $w_0, w_1, \ldots, w_m \in K' \subseteq (N \cup T)^*$.
($(N \cup T)^*$ denotes the set of words over $(N \cup T)$).

A is said to be simple if $w_0 \in N$ and very simple if $w_0' \in (T^*N \cup T^*)$.

Note that syntactic office agents correspond - with some restrictions- generalized conditional grammars [10]. The simple case corresponds to the context-free, the very simple to the regular case.

Notation 2.

The class of office societies consisting of syntactic office agents is denoted by $\mathcal{O}^{syn}$.
The class of office societies consisting of very simple syntactic office agents is denoted as $\mathcal{O}_{vs}^{syn}$.

The following statement expresses that the class of problem solution spaces of societies of syntactic office agents corresponds to the class of recursively enumerable languages, that is, it consists of syntactically arbitrarily complicated elements.

Theorem 1.

$$\mathcal{L}(\mathcal{RE}) = \mathcal{C}(\mathcal{O}^{syn}),$$

where $\mathcal{L}(\mathcal{RE})$ denotes the class of recursively enumerable languages.

The statement can be proven by some technical considerations using some tools of the theory of formal languages (see e.g. [11]).

The next statement establishes that the class of problem solution spaces of the office societies consisting of two very simple syntactic office agents with two communication boards, five open questions during the problem solving process and with rather simple social and communication agreements, is equal to the class of problem solution spaces

of office societies consisting of syntactic office agents in general. Note that this result have been developed on the basis of results obtained in [9,10].

Theorem 2.

$$\mathcal{C}(\mathcal{O}^{syn}_{vs}(2,2,5)) = \mathcal{C}(\mathcal{O}^{syn}).$$

The theorem can be proven by using the following result from [12]. The proof is rather technical, therefore the details are omitted.

Let $T = \{a_1, \ldots, a_n\}$, $n \geq 1$, be an alphabet. An Extended Post Correspondence (shortly, an EPC) is

$$\Pi = (\{(u_1, v_1), \ldots, (u_r, v_r)\}, (z_{a_1}, \ldots, z_{a_n})),$$

where $r \geq 1$, u_j, v_j, $z_{a_i} \in \{0,1\}^*$, $1 \leq j \leq r$, $a_i \in T$, $1 \leq i \leq n$. The language represented by Π in T, denoted by $L(\Pi)$ is the set

$$L(\Pi) = \{x_1 \ldots x_m \in T^* : m \geq 1, s_1, \ldots, s_l \in \{1, \ldots, r\}, l \geq 1,$$

$$\text{such that } v_{s_1} \ldots v_{s_l} = u_{s_1} \ldots u_{s_l} z_{x_1} \ldots z_{x_m}\}.$$

In [12] it was shown that every recursively enumerable language L can be represented by an EPC Π and reversely.

By using the same technique we can interprete a result from [10] obtained for blackboard architectures. There it was shown that by using a finite number of words as recognition conditions, 7 cooperating regular grammar can produce every recursively enumerable languages. Thus, the theorem is as follows:

Theorem 3.

The class of office problem solution spaces of office societies consisting of syntactic office agents is equal to the class of office problem solution spaces of office societies consisting of 7 very simple office agents which use strategy "minimal comment" with temporaly monopolization and during the problem solving process communication tables are of fixed contents.

6. Conclusions

The aim of our research is to introduce a convenient mathematical model for the description of distributed/cooperative activities in offices. We can see that the theory of rewriting systems, which is a highly elaborated field can provide us tools for handling

such problems. Results obtained for the minimal size and structural properties help us to determine the the basic, elementary constituents and properties of software tools to be developed.

References

[1] Office automation. Topics in Information Systems. D. Tsichritzis Ed., Springer Verlag, Berlin, 1985.

[2] ELLIS, C. A. Information control nets: A mathematical model of office information flow. *ACM Proc. of the Conf. on Simulation, Modeling and Management of Computer Systems* (Aug.) ACM, New York, 1979, pp. 225-240.

[3] CHANG, J. M. and CHANG,S.K. Database alerting techniques for office activities management. *IEEE Trans. Commun.* 1 (1982), 74-81.

[4] GIBBS, S. J. Office information models and the representation of office objects. In *SIGOA Conf. on Office Automation Systems*, ACM, New York, 1982, pp. 21-26.

[5] BOOCH, G. Object-oriented development. *IEEE Trans. Softw. Eng.* 2 (1986), 211-221.

[6] BARBER, G. Supporting organizational problem solving with a work station. *ACM Trans. Off. Inf. Syst. 1* 1(Jan.1983), pp. 45-67.

[7] HO, CH-S. and HONG, Y-CH. A society model for office information systems. *ACM Trans. Off. Inf. Syst. 4* 2(Apr.1986), pp. 104-131.

[8] CSABA, L. and CSUHAJ-VARJÚ, E. Cooperative office. In: *Proc. of Beyond Number Chrunching, 3rd Austrian-Hungarian Conf. on Informatics.*, (V. Haase, E. Knuth, eds.), Budapest-Graz, NJSZT-OCG, 1988, pp. 57-68.

[9] CSUHAJ-VARJÚ, E. and DASSOW, J.: Message Handling Grammar Systems. Papers on Automata and Languages, X, DM 88-1, 1988, pp.71-82.

[10] CSUHAJ-VARJÚ, E. and KELEMEN, J.: Cooperating grammar systems: a Syntactical Framework for Blackboard Model of Problem Solving, In: Proc. AI-ICSR'89, (I. Plander, ed.), North Holland, Amsterdam, 1989, pp. 121-129.

[11] SALOMAA, A. Formal languages, Academic Press, 1973.

[12] GEFFERT, V., Context-free-like forms for phrase-structure grammars. In: *MFCS'88,*, LNCS 324, Springer Verlag, 1988, pp. 309–317.

SBS: A Multimedia Service Base System Based on Distributed Objects

Qianshan He and Hidehiko Tanaka
Department of Electrical Engineering, University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo 113, Japan

Abstract

In this paper, we propose a distributed system called service base system (SBS) based on an object-oriented approach. The objective of SBS is to provide a convenient and integrated environment for users to access and combine distributed resources. Distributed resources are abstracted as objects containing the meta-information of the resources. Three layered views (external, conceptual and internal view) about objects are divided. In each layer, a different object management method is used. Service bases in different nodes cooperate with each other to manage these objects. We have constructed an experimental system in an Ethernet-based network environment. This paper presents the concept of SBS and reports its current implementation status.

1 Introduction

Developments in workstations recently provide a new way to access resources: an user can manipulate or display data of different media types (image, text, graphic or voice media) by using a single workstation. Moreover, developments in networks, especially the high bandwidth and high speed networks make the sharing of these resources available and practical. The linking of workstations by a network has become standard practices in organizations or universities. In these distributed environments, the numbers and types of resources available to users are being expanded. Programming, functions provided by computers are becoming more complicated. Users may have troubles to access and manipulate these resources. It is necessary to use an uniform method to integrate these resources to provide some convenient high level services to users.

In this paper, to solve the problems above, we propose a multimedia service base system based on distributed objects. We use the object model to represent resources. In a service base, Three layered views about objects are divided. In each layer, a different object management method is used. With SBS, distributed resources can be used conveniently by users and application programs though an uniform interface without worrying about the different types of media or the heterogeneity of computers. Our design considerations emphasize on providing an integrated, extensible and easy to be constructed distributed system.

We organize the paper as follows. In section 2, the fundamental concept of SBS is given. Section 3 describes the detail structure of a SBS. Section 4 presents an experimental system and discusses the implementation issues. Finally, in section 5, some conclusions are given.

2 The Concept of SBS

2.1 Object Model

In a computer system, processors, memories, I/O, files and networks are usually managed by an operating system which provides higher level functions to users. On the top of the

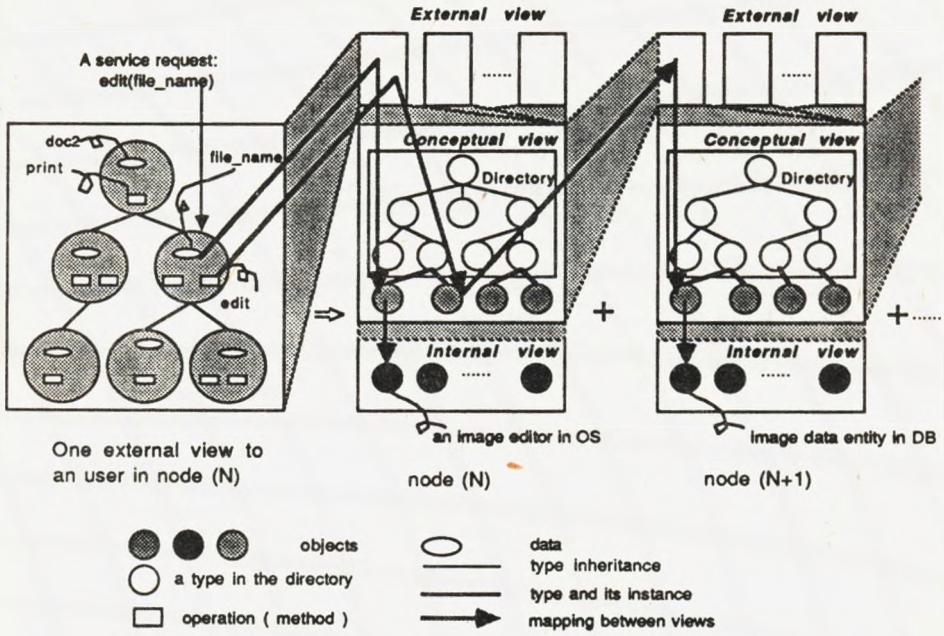


Figure 1: The Three Layered Views and the Mapping of Views

operating system, users can see various programs and data: a program can be a function provided by a computer (e. g., a command of an operating system), or an application program created by an user with a certain language to perform a specific work; data may be expressed in different media types. An application system is constructed by combining these programs and data. By resources in our system, we refer to these high level functions, data and application programs. Usually, using the resources can be described as: "An operation is imposed on data and certain results are produced". The operation may be a function provided by computer (e. g., a compiler or an editor) or an application program. This concept can be extended directly to the object-oriented concept: an object contains its data and operations on the data. Therefore, using the object model to express resources is natural.

On the other hand, In a distributed environment, resources may have their more complicated structures: Where are the data? Where are the programs executed? Who possesses these data and programs? What media types are the data? Are the data or programs in disk or in memory? We call these information meta-information of resources. When abstracting a resource as object, we include these meta-information in the conceptual views of the object. In the external view of a distributed object shown to users, these meta-information is unnecessary. In other words, network transparency should be realized in the user level.

Object model has the characteristic of information encapsulation ability. This characteristic provides us a good method to overcome the different types of media and the heterogeneity. We include the meta-information of a resource inside an object as its attributes. Therefore, only the external interface (specifications) of a resource are shown to the users, the structures or implementation details of the resource are encapsulated inside the object. The attributes of the object will be used by SBS to absorb the different types of media and the heterogeneity. Object model also have good modularity. This characteristic provides a way to extending resources. In a distributed environment where the numbers and types of resources available to users are being expanded, this extensibility is very significant. If the common structure of

SBS has been defined, then adding a new kind of resource is a simple matter by defining a new type in SBS.

2.2 Object Management

In each node of the networks, a service base is built up to manage the local objects and cooperates with other service bases in other nodes for accessing remote objects. Distributed *services* are provided on top of the service bases by combining various local or remote objects. Inside a service base, as shown in Figure 2.1, three layered views about objects (*external view, conceptual view, and internal view*) are divided:

- The external view is the view shown to an user or a remote node. Different users or remote nodes may have their different external views on an object. An object in the external view can be seen as a composite object which contains an object in the conceptual view and an object in the internal view.
- In the conceptual view, meta-information of a resource is abstracted as an object. The differences of media types, heterogeneity and location distribution of resources are absorbed in this layer.
- An object in the internal view is the entity of a resource. The internal view about an object concerns its physical realization in a computer. Objects in the conceptual view are reflected to their respective objects in the internal view.

A service base processes the queries from users or application programs by mapping the objects in three layered views: when a service request (which may contains several objects in the external view) is invoked, the request is mapped to the objects in the conceptual view. According to the location attributes of the objects in the conceptual view, the objects in the conceptual view are mapped either to the objects of the internal view in the local node or to the objects of the external view in a remote node. We will take an example in section 3.2 to show how the mapping of objects in different views works.

In section 3 and section 4, we will discuss the details of object management methods in each layer: object managements in the conceptual view are based on a directory and a database; in the external view, a browser and a library are provided as interface for users and application programs to access objects; object managements in the internal view are based on an existing operating system and a database management system.

2.3 System Configuration

We suppose that the system is built on a long-haul network or a local area network (LAN) environment. In a LAN environment, administrative of machines according to subgroups of an organization will be natural. We call these subgroups *clusters*. In a cluster, there may be some servers such as file servers and print servers which are shared by the machines inside the cluster. Because of the high speeds of the LAN and the uniform administrations, a centralized management configuration will be efficient within a cluster: that is, a service base in each node are constructed to manage local objects and provide objects to other nodes, all objects provided by the cluster are registered centrally in the service base of the cluster server (the cluster server should be a node which has fast responding time and massive secondary storage inside the cluster). The cluster server acts as an overall guidance for the nodes in a cluster. When a request of some objects is invoked in a node of a cluster, the service base in this node first checks if objects are located locally, if not, a further request to the cluster server is invoked. Based on the information about these objects in the cluster server, the objects are fetched from a service base of a node inside the cluster. If the objects are outside the cluster, the cluster server broadcasts the request to the cluster servers in other clusters.

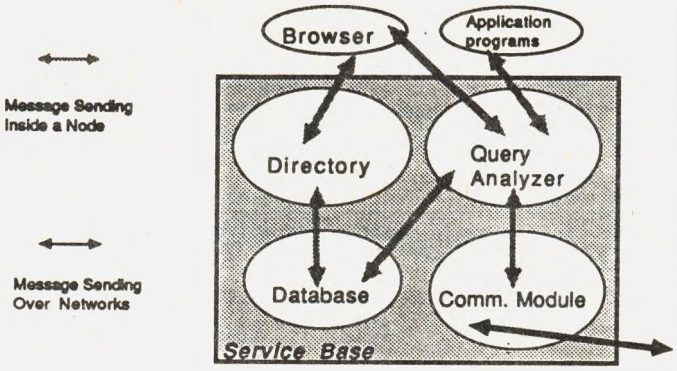


Figure 2: The Structure of a Service Base

Nowadays, most of the long-haul networks are connected by a number of local area networks. In a long-haul network environment, we can consider that the system consists of many centralized clusters. Objects management within one cluster is just like that of LAN environments.

Workstations, personal computers and servers (file servers, cluster servers etc.) may have their different emphases. A workstation usually acts as an access point to manipulate or display multimedia data. This means that the user interface to the service base in a workstation should provide powerful display functions such as multiple windows and graphical displays. On the other hand, servers dedicate to provide objects. Large secondary storage systems and service bases with fast responding time are important in these servers.

3 Structure of a Service Base

The structure of a service base in a workstation node, a personal computer node or a cluster server is about the same. The only difference is that a service base in a workstation or a personal computer contains the information of the local objects, but a cluster server contains the information of objects inside a cluster. In a workstation or a personal computer, if a requested object is not local object, it will send a massage to its cluster server, the cluster server tells which node or cluster the requested object is in.

Figure 2 shows the structure of a service base. We structure the service base (including the interface) as a collection of objects. The *directory* classifies all objects which can be used though this service base. The *query analyzer* object is responsible to handle the queries sent from users or application programs, and to perform mapping of the three layered views. The *communication module* object provides communication interface between objects in different nodes.

3.1 Directory

In the conceptual view of a service base, the meta-information about a resource are abstracted as an object. To classify various kinds of objects, a directory is created in each service base. The directory indicates the characteristics of various resources: for example, what attributes a certain kind of resources has? What operations can be performed on this kind of resources? What is the relationship between this kind of resources and other kind of resources? etc.. The directory contains a collection of types. A type is defined to classify resources which

```
class dirDocument :
      public dirRoot
{
    Attribute name;
    Attribute location;
    Attribute format;
    Attribute mediaType;
    Attribute creatTime;
    Attribute owner;
    ......
public:
    void multimediaMail();
    void multimediaEdit();
    ......
}
```

```
class dirODAdocument :
      public dirDocument
{
    Attribute format = ODA_DOCUMENT;
    Attribute mediaType = MULTIMEDIA;
    Attribute status;
    ......
public:
    void editingProcess();
    void layoutProcess();
    void presentationProcess();
    ......
}
```

Figure 3: Type Examples in the Directory

have the same characteristics. For each type, the operations which can be applied to this type are defined. A type may have its supertype. The attributes and the operations of the supertypes are inherited by its subtypes. In this way, a subtype only need to define the attributes which are different from its supertypes. We structure the directory as a single-inherited type collections. The attributes of a type are expressed by instance variables of the type, and operations are expressed by methods. Figure 3 shows an example of types in the directory (this example will be explained later). In this example, classes in object-oriented language C++ is used to define the types.

A Directory follows a tree structure in which information are hierarchical (Figure 4). We suppose directories in different nodes may have their different tree structures according to their environment. This enriches the extensibility of each node, and make our system applicable to different environments. On the other hand, we define some minimum standards to ensure the whole system can co-operate with each other. This means that we must define some types and attributes which are common to variable objects. In an actual implementation of the system, types and attributes which are not defined in the standard can be added to a directory to make the system suitable to certain applications. For example, if a new type of media or a new type of functions is to be added to the system, then a new type for that specific type should be defined.

In a heterogeneous environment or a multimedia environment, attributes of the objects are very different from each other. How to handle the attributes appropriately is the main point of integrating them as a whole system. In our experimental system, we only define parts of attributes and attribute values. How to include the attributes and attribute values which are necessary to an environment and exclude the attributes which are unnecessary is remained as a problem which should be solved in an actual implementation. The structure of types, attributes and attribute values forms the directory schema.

A type in the directory is only a description of one kind of resources. To register a resource in the service bases and to make it available to network users, a type instance (object) which contains the specific attribute values of the resource should be produced and registered in a database.

We take an example here to show how a directory is structured and used. As shown in Figure 3 and Figure 4, a *dirDocument* type is defined. According to the attribute *format* of type *dirDocument*, the subtypes of *dirDocument* can be type *dirODAdocument*, type *dirText* and type *dirImage* etc.. All attributes and operations of type *dirDocument* can be inherited by its subtypes.

The type instances of *dirODAdocument* are the documents which follow ODA standard of OSI [1]. Besides the attributes in type *dirDocument*, type *dirODAdocument* may con-

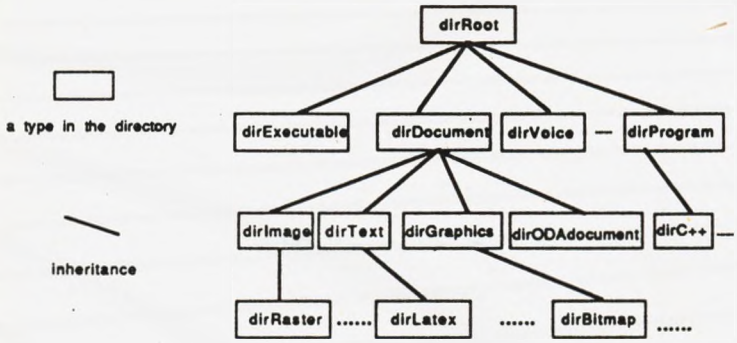


Figure 4: A Directory in a Service Base

tain the attributes like ***status*** which shows the status of the document. The attribute value of *status* (in a type instance) can be *PREVIOUS_CONTENT, EDITED_CONTENT* and *LAYOUT_CONTENT* etc.. [1] The operations in *dirODAdocument* may contain *editingProcess, layoutProcess* and presentationProcess etc.. To a specific document, an object which contains certain attribute values is created and registered in a database. To print this document (that is, to call the *presentationProcess*), the service base first locates and fetches the document (according to the *location* attribute), then checks if this document is printable (attribute *status* is *LAYOUT_CONTENT* or not). If not, say, attribute *status* is *EDITED_CONTENT*, the service base will automatically call the *layoutProcess* service to produce a *LAYOUT_CONTENT* of the document, then print it. In this example, we can see logically that service bases provide a new service that can print a document which is not in printable status by combining some existing registered services (*layoutProcess* in the example). This is only a simple example. When large numbers of services are registered in the service bases, logically more functions can be supplied to users by combining these services.

3.2 Mapping of the Three Layered Views

The relationship of objects in the three layered views has following characteristics: first, objects inside one node are referenced by object pointers. Objects distributed in networks are referenced by remote objects mapping. We do not use remote object pointers to reference remote objects. Second, logically, a global inherited relationship of distributed objects can be seen in the external view of a node.

Our system is being implemented by object-oriented language C++. Using this kind of language to implement remote object pointers is difficult and complicated, in that we must extend the language to a distributed one. Instead, we use the remote objects mapping mechanism to realize remote object references. In the follows, we use a simple example (in Figure 1) to explain the mapping mechanism.

We suppose there is a file registered in the service bases, its object name in the external view of node(N) is called *file_name*. An user in node(N) wants to display this file. We suppose he does not know where the file is actually located and what media type the file is expressed by. Therefore, he is not certain to use what kind of editor to display this file.

[1] ODA distinguishes between a document's logical structure and its layout structure. Before logical structure editing, the content of a document is called previous content. After logical structure editing, the content of a document is called edited content. After layout structure generation, the content of a document is called layout content and can be presented on a medium (paper, screen) through a presentation process.

After *edit(file_name)* is requested to the service base in node(N), the file (which actually is an image file) is displaied in an image editor selected by the service base in node(N).

Inside the service base in node(N), the objects in the external view are first searched. If there is a object with the same name as the *file_name*, this object is selected. An Object in the external view is a composite object containing an object in the conceptual view (connected by a pointer) and an object in the internal view. By using the pointer, the object in the external view of node(N) is mapped to the object in the conceptual view in node(N). The object in the conceptual view (attribute information of the resource) indicates that the file is located in node(N+1) and the media type of the file is image media, and that there is an image editor (object name in the external view is called *edit*) in node(N) which can display this file (in the conceptual view, an object not only contain the attributes of a resource, but also contain the information which indicate what operations can be applied to this resource.). Then, the operation part of the object in the conceptual view is mapped to an object in the internal view in node(N) (the entity of the resource), and a request (object name) is sent to node(N+1) to map an object in the external view in node(N+1). In node(N+1), same processes of objects mapping as in node(N) are done and finally the file is copied to node(N) to be displaied by the editor. In this example, we can see that the remote object mapping is realized by containing the attributes of remote node name and remote object name in the conceptual view, and by searching the object in the external view in the remote node.

From this example we can also see that logically the objects in the external view of node(N) have a global inherited relationship. The global object image is realized by the so-called forward mapping: if objects is not local objects, the objects in the conceptual view of node(N) are mapped to the objects in the external view of node(N+1), and the objects in the conceptual view of node(N+1) are forward mapped to the objects in the external view of node(N+2).

3.3 Query Analyzer

Query analyzer is responsible to handle the queries sent from users or application programs, and to perform mapping of the three layered views. When an user or an application program makes a request (consisting of objects in the external view) to the system, query analyzer first maps their respective objects in the conceptual view, then searches the database to get the attribute information of the objects. Basing on the attributes of the objects, query analyer in a service base works as follows: first, query analyzer checks the location of each object. If the requested objects are local objects, the objects in the conceptual view are mapped to the objects in the internal view. If some of the requested objects are remote objects, messages are sent to the cluster server to locate the objects. Data objects will be fetched to the node where the requested operation is located. Second, before the requested objects are combined, the semantics of each object are checked based on the attributes of the object. This is important in the different media types or heterogeneous cases. For example, to display an image data, a window which can display image rather than text should be used. Third, query analyzer combines necessary objects to provide services. The way to combine objects can be explicitly defined by the user in the request, or automatically made inside the service base. The combinations of objects inside a service base mean that after the semantics checking, query analyzer finds some suitable registered objects to meet the needs of the user's request. Finally, If the operations of an object are local, the query analyzer forks processes to execute the operations locally.

To search or use a object registered in the service bases, we provide two kinds of query methods. The first method is called "query by object name". Users input the name of needed object, SBS searches the object and gives answers. The second method is called "query by object attributes". In some cases, users may want to use or search objects with certain characteristics. In these cases, users can query the service bases by selecting a set of attributes and attribute values to get needed objects. For example, by setting

{ *creatTime* >= "1989.11.1", *Format* = *LATEX* }
the objects which are in LATEX file format and are created after November 1, 1989 will be selected.

4 An Implementation of the System

4.1 Environment

We have implemented an experimental system based the model proposed above. This experimental system is used to detail the problems which occur in an actual implementation of the model and to show the effectiveness of this model. Image, text, graphic and voice data can be manipulated by the system. The network used is an Ethernet-based LAN which is located in the Faculty of Engineering, University of Tokyo. Through a gateway server, this LAN can link to a long-haul network called *junet* in Japan. Our experimental system contains a number of SUN3, SUN4, VAX workstations. The experimental system is written in object-oriented language C++.

4.2 Database

As discussed above, in the conceptual view, a disrectory is used to classify resources. For each resource, a type instance (object) is created and registered as a persistent object in a database.

In an actual implementation, we must make choice to use what kind of databases (for example, relational database or object-oriented database). The external view and the conceptual view of our system are implemented by object-oriented language C++. It will be efficient for object-oriented language to access an object-oriented database. Actually, in an object-oriented database, the language and the database are usually designed as a whole system. But in the early implementation stage of our experimental system, object-oriented database in each node is not available. Therefore, we decided to use a relational database called INGRES [2]. To use a relational database, the main problem is focused on how to map the directory schema to the schema of the database. To represent the directory schema which is a tree structure in a normalized relational form will need a lot of relations (tables). To solve this problem, we use unnormalized relational form to represent the directory. In the database, each type in the directory s represented by one relation, and several relations are used to keep the relationship of the directory tree.

Now, we are using POSTGRES [3] to substitute INGRES. POSTGRES is a successor of INGRES, it provides many functions that an object database supports. For example, POSTGRES provides supports on complex objects, inheritances and procedures. To map the directory schema of SBS to the schema of POSTGRES is more easy than INGRES. Beside, POSTGRES provides variable length arrays for applications that need to store large homogenous sequences of data (e.g. text, image or voice), this feature provides us a way to store multimedia documents (the objects in the internal view) into the database.

4.3 Interface

To manipulate the objects in the external view, we provide a browser for users to manipulate or display objects. Moreover, a standard query language C++ is provided as a standard interface for application programs to access objects managed by the service bases.

The browser consists of shell windows, object registration tools (containing a directory window and many registration windows) and editors for displaying data of different media types. These editors can be seen as system services of SBS: they are the software working in the computer independently and are registered as objects of SBS. Now, we are beginning

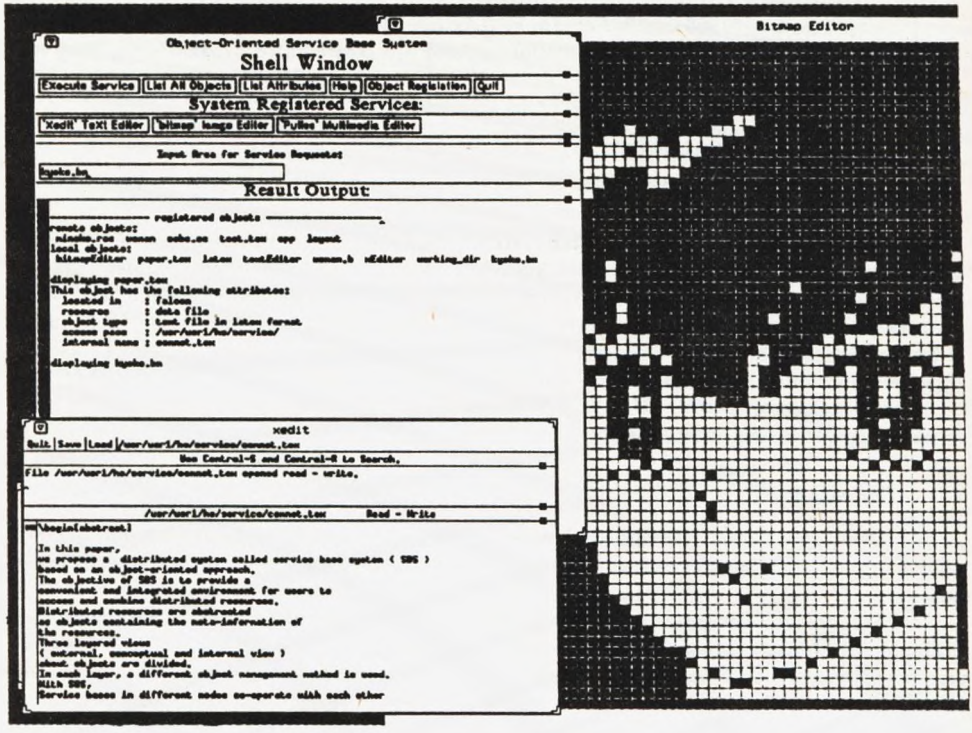


Figure 5: A Shell Window in the Browser

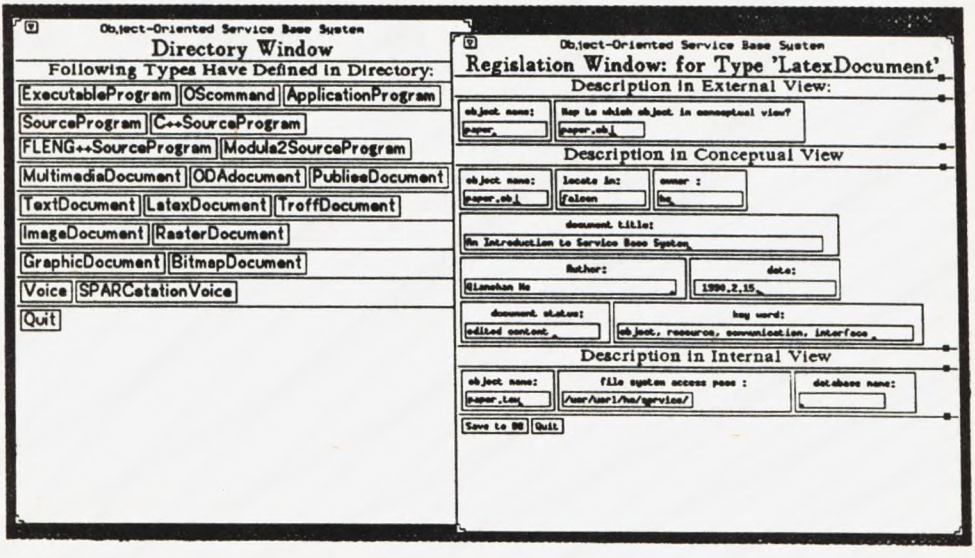


Figure 6: Object Registration Tools

to add an editor called Publiss [2] which can display text, image and graphic data in a some window as a system service of SBS.

The browser in SBS is implemented by using the toolkit of X-window in SUN workstations. An example of shell windows is shown in Figure 5. Shell windows provide functions to invoke service requests, list objects registered in SBS, list attributes of an object etc.. Shell windows also provides accesses to other system services: in Figure 5, a 'xedit' and a 'bitmap' editor is opened through the shell window to display a text object and a bitmap image object registered in SBS. Figure 6 showns an example of object registration tools. Users use these windows to register new objects into SBS. In the directory window, type names defined in the disrectory are shown. Selecting a type in the directory window will produce a registration window for object registration. This registration window contains all attributes the selected type possesses. When resource types increase, more different registration window can be added to the browser by simplely adding subtypes in the directory. As to the workstation without bitmap display, an interpreter which has the same function of a shell window in the browser is created to provided scroll display functions.

For applications written in common languages (Fortran, Modula-2, etc.), a library is provided. The library provides a way for common languages to access the service bases through standard query language C++.

5 Conclusion

In this paper, we have presented the concept and an implementation of service base system. Our experimental system focuses on how to abstract the meta-information from resources and how to manage these meta-information. The object-oriented approach provides us a modeling method. The experimental system is constructed at a relatively high level. The service base in each node is built on an existing operating system, a database management system and communication interface. Through the experiences of the implementation, we can conclude that SBS can be easily constructed. More works are being done on our current experimental system to make it become a practical one. These works include adding an object database and a multimedia document editor to the system, using a parallel object-oriented language FLENG++[4] to implement the concurrent controls in SBS.

SBS is expected to be effective in the environment where new resources are extended frequently and the types of resources are various, for example, in office automation environments or research laboratory environments in universities or organizations.

References

[1] ISO/IEC, "Text and Office System – Office Document Architecture (ODA) and Interchange Formate", DIS 8613, 1988.

[2] M. Stonebraker, E. Wong, P. Kreps and G. Held, "The Design and Implementation of INGRES", ACM TODS, 1, 3, pp. 189 - 222, 1976.

[3] M. Stonebraker and L. A. Rowe, "The Design of POSTGRES", Proc. of ACM SIGMOD, pp. 340 - 355, Washington, 1986.

[4] H. Nakamura and H. Tanaka, "Object-Oriented Programming Language FLENG++ Based on Parallel Logic Language FLENG", WOOC'89, Japan, 1989.

[2] Publiss is a trademark of BRIDGE Ltd.

FRAMES FOR PROTOCOL REPRESENTATION

K. Tarnay*, S. Wagner-Dibuz*, P.A. Wagner**

***Central Research Institute for Physics**
Budapest, H-1525, P.O.B. 49
****Computer Research and Innovation Centre**
Budapest, 1015, Donáti u. 35-45.

ABSTRACT

FAIR, a new frame-based system integrated with MPROLOG is discussed, and it is applied for protocol representation. As an example, FAIR-frames are developed for general description of transport protocol and for its predicates.

1.INTRODUCTION

Our goal is to demonstrate how frames in expert systems [HAR-85] can be applied for protocol representation. This paper is organized as follows: section 1 is an introduction to the protocol engineering and knowledge representation generally, and into protocol field specially; section 2 deals with our new frame-based system, FAIR; section 3 analyses the possibility to use frames for protocol description, section 4 discusses the predicate-frames described in FAIR; finally in section 5 some conclusions are drawn.

Protocol engineering

The reference model defines seven layers with entities in a communication network. The peer entities communicate with each other according to the protocol [BRAN-83]. The protocol description defines the messages which can be sent during the

communication, it determines the semantics of the communication and its time attributes. Protocols have to be standardized so that computers produced by different manufacturers are able to communicate with each other.

The protocol standard is a long description of the protocol (in English). It defines everything about the protocol: the syntax of the messages (the bit-map), what the possible responses are to a given message, the parameters and their values and the protocol mechanisms. It is also a very complicated description, and difficult to understand. The way from the protocol standards to the realized entities which are able to communicate with each other is a long one. This labour-consuming process is called protocol engineering (Fig. 1).

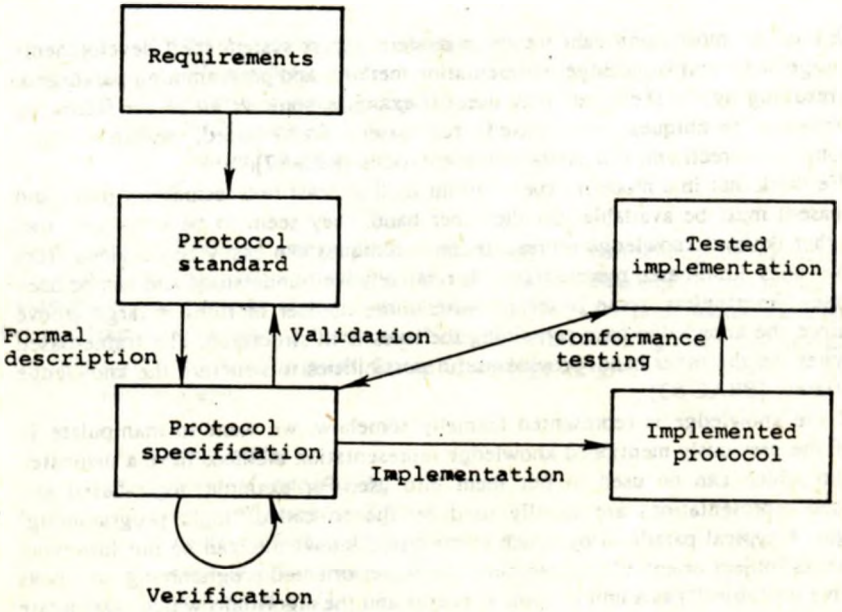


Fig. 1. The protocol engineering process

An exact and clearly arranged description is needed for each protocol, which is called the protocol specification. Some of the most frequently used formal description methods of protocol specifications can be classified into two groups: the graphical methods and the programming language-like methods. The graphical description methods are the finite-state machine models: SDL/GR, Petri nets. The "programming language-like" specification methods are SDL/PR, ESTELLE, LOTOS etc. It is easier to understand the protocol operation from the protocol specification and easier to implement the protocol in a programming language using the protocol specification.

However before implementing the protocol the specification has to be verified which means that the logical correctness of the protocol specification has to be checked, and it can also be validated. Verification and validation reveals deadlocks, unspecified message receptions usually using the finite-state machine model of the protocol. The implemented protocol has to be tested for conformance with the protocol standard, i.e. that it fulfills every requirement of the standard. This process is called conformance testing.

Knowledge representation

One of the most significant trends in modern expert system shell developments is the usage of several knowledge representation methods and programming paradigms. In the resulting hybrid shells one may use, for example, some or all of the following representation techniques: logic-based, rule-based, frame-based, semantic nets, isomorphic (or direct) and procedural representations [RES-87].

We think that in a modern expert system shell at least two techniques (rule- and frame-based) must be available. On the other hand, they seem to be sufficient, too: every other (known) knowledge representation techniques can easily be obtained from these two ones. Rule-based representation is relatively well-understood and can be used with ease. Nevertheless, some problems arise if the number of rules is large (above 1000 since the knowledge base containing the rules is ill-structured. The frame-based approaches, on the other hand, provide useful possibilities to structure the knowledge appropriately [BRAC-83].

If the knowledge is represented formally somehow, we want to manipulate it. Most of the previously mentioned knowledge representation methods have a dedicated paradigm which can be used to put them into use. For example, logic-based and rule-based representations are usually used by the so called "logic programming" paradigm. A typical paradigm by which frame-based knowledge can be put into work is known as "object-oriented" programming. In object oriented programming one looks at a frame (or "object") as a unit of some concepts and the algorithms which manipulate these concepts, and uses a loose or close synchronization among the units.

We would like to describe here a particular system FAIR (Frames in AI Representation of knowledge) [ECS-89a, ECS-89b], which allows the user to merge frames and rules in a flexible manner.

Knowledge representation in protocol engineering

The hierarchical structure of frames provide the best way of representing a protocol. Every piece of information can be put in frames. The most important information

is put in frames on the top of the hierarchy, and as we go lower in the structure we find frames containing more specialized information about protocol messages, states and parameters. Here we show only a small example on the application of frames for protocols, but in section 3 we present a detailed description of the transport protocol with frames.

```
frame:transport_protocol;
    initial-negotiation:yes;
    DT_numbering:normal;
    expedited_data_transfer:yes;
end.

frame:transport_protocolc_lass_0;
    DT_numbering:not;
    expedited_data_transfer:not;
    TPDU_size:128;
    multiplexing:not;
end.
```

In protocol validation frames provides a good way for representing the reachability graph, it is especially useful in storing the information attached to the nodes, the protocol states, the contents of the channels. In this kind of representation each node is put in a frame, and a relation "connected" can be defined among them, which stands for the arcs of the reachability graph. In the followings we give the frames for a node of the reachability graph:

```
frame:  node;
    state-of-machine-1:[];
    state-of-machine-2:[];
    content-of-channel-1:[];
    content-of-channel-2:[];
    connected-to:[];
end.
```

Frames provide a well-structured description of the protocol, which is a good basis for the implementation. The messages and state transitions are stored in frames, the protocol activities are described with demons. On the other hand a frame-based language is very close to a programming language and if it is completed with demons describing the protocol activities gives a runnable protocol description.

In conformance testing the representation of the tester with frames can be given similarly to the description of the protocol with frames. The hierarchical structure provided by frames is also useful in this application. The generic frame representing

a state transition will contain the same slots for the tester as for the protocol, but there will be more transitions, because the tester will have a new state for every expected answer of the implementation under test.

2. FAIR, A FRAME-BASED SYSTEM

The heart of the FAIR system [ECS-89a, ECS-89b] is an integrated representation language. This language makes it possible to use several knowledge representation methods such as frames, semantic nets, scripts, graphs, and logic formulae, production rules, and procedural information (the latter three in MPROLOG). The integrated language is supported by a dedicated user-friendly editor, or can be used in line mode or with menus.

From the "programming paradigm" point of view the FAIR supports only the logic programming. Nevertheless, the language of FAIR is very easily extendible by other means among them by new manipulation methods (i.e. by new paradigms). This flexibility and extendibility are considered to be among the main advantages of the FAIR system.

Information is treated in FAIR on three different levels:

- the level of frames,
- the level of frame structures,
- the level of worlds.

Frames

A frame is a structured abstract model of a concept. This model is usually given by an unordered set of properties. These properties are called slots; each slot represents an aspect of the concept considered important from the point of view of the description. A slot may have values; these values can be names of other frames or expressions of some formal/computer language (in our particular case of MPROLOG) [CLO-81].

We note that some parts of a frame can be omitted as is indicated by the following general syntax:

frame: name;
[slot_1:[[value_11,value_12,...,value_1n$_1$]];]
[slot_2:[[value_21,value_22,...,value_2n$_2$]];]
. . . . . .
[slot_m:[[value_m1,value_m2,...,value_mn$_m$]];]
end.

Observe that the outermost and innermost brackets denote simply the optional parts of the definition, while the brackets inbetween (in boldface) are keywords and denote lists (in MPROLOG).

If a frame has only a name (and no slots or values are present) then we call it "primitive frame". If a frame has a name and some slots but no values, then it is called a "generic frame". Finally, if nothing is omitted, then we say it to be a "full frame".

Meta information

In order to cope with the intrinsic complexity of real world concepts, additional information (called here "meta-information") can be associated to the main parts of frames. Thus, we may associate meta-information to frames themselves, to the slots or to values. Accordingly, meta-information can be separated from the main body of the original frame by the following keywords

meta_frame	*meta_end*
meta_slot	*meta_end*
meta_value	*meta_end*

The meta-information will also be coded by frames. Basically, there are two ways to define frames holding meta-information: first the user can give his/her own frame for this purpose and secondly the user can fill-in a pre-fabricated frame tailored specially for holdingt meta-information. These pre-fabricated frames may contain information about

- demons or other activities concerning frames and slots,
- constraints on the cardinality and range of the values of a slot,
- where a missing value can be obtained from; possibilities include default values, inheritance from other frames or asking prescribed questions from the user.

Relations

As concepts modelled by frames can be connected somehow, we may want to define relations among frames. In FAIR there exists two kinds of relations:

- relations known to the system by default,
- relations defined by the user.

Once a relation has been defined, its name can be used as a slot in the frame to be connected to another, while the value of this slot identifies the frame to which we want to connect our particular one. The FAIR system knows about two relations, namely

the *is_a* and the *instance_of* relations. Actually, these two relations are treated by FAIR in the same way. Intuitively, however, we may use *is_a* as the familiar *subset* relation and *instance_of* as the *membership* relation of set theory.

User can define also relations by filling-in a pre-fabricated generic frame.

Inheritance

Relations defined among frames allow transfer of information from one frame to another. This mechanism makes it possible to store every piece of information at the most appropriate place. In the FAIR system, inheritance [AIT-86] may be controlled in two different ways:

- local control can modify the inheritance of a slot by specifying the way how the slot may inherit information from other places and the way how the slot can be inherited by other places,
- global control can modify the inheritance along a particular relation.

The information needed to control inheritance will be given in prefabricated frames.

Worlds

Concepts of real world can be described from many different points of view. Frame-based systems usually supports this kind of grouping. In fact, FAIR also gives the possibility to partition frames into groups; these groups are called "worlds" in FAIR.

A world is simply a set of frames; all mechanisms (including inheritance) will always work inside a world.

Worlds can be connected to each other; the structure of worlds is a (rooted) tree. After initialisation the root will be automatically created. Then the user may define new worlds and may arrange them into the tree of worlds.

3. A FRAME-BASED APPROACH TO PROTOCOL DESCRIPTION

THE ASM MODEL

In this paper we use the abstract state machine (ASM) model of the protocol [CAS-87]. The model is defined by seven sets:

(S, I, O, V, A, P, T)

where

S: is the finite set of states.

I: is the finite set of the incoming messages of the protocol.

O: is the finite set of the outgoing messages of the protocol.

V: is the finite set of variables, which can be divided into two distinct sets:
- — local variables,
- — global variables.

A: is the finite set of actions, there are local actions concerning local variables, and global actions concerning all the variables.

P: is the finite set of predicates, which map the variables into the values true or false. There are three subsets of predicates:
- — controlled,
- — settable,
- — non-controlled.

T: is the finite set oftransitions, a transition is defined by the following:

(s, i, p, a, o, s')

which means that the automaton it the state "s" receives an input "i", and if "p" predicate is true it executes the "a" action, sends an "o" output and goes into state " s' ".

Fig. 2 shows the simplified state transition graph of the transport protocol class 0. Its ASM model is the following:

S= { CLOSED, WFCC, WFTR, OPEN }
I= { TCONreq, TCONresp, TDISreq, TDTreq, CR, CC, DT, DR }
O= { TCONind, TCONconf, TDISind, TDTind, CR, CC, DT, DR }
V_{local}={ SZEGM,... }
V_{global}={ 0 }
P={pred1, pred2,...}
A={action1, action2,...}
T={ (CLOSED, TCONreq, CR, WFCC),
.
.
.
}

(The elements of the transition which do not have values can be omitted.)

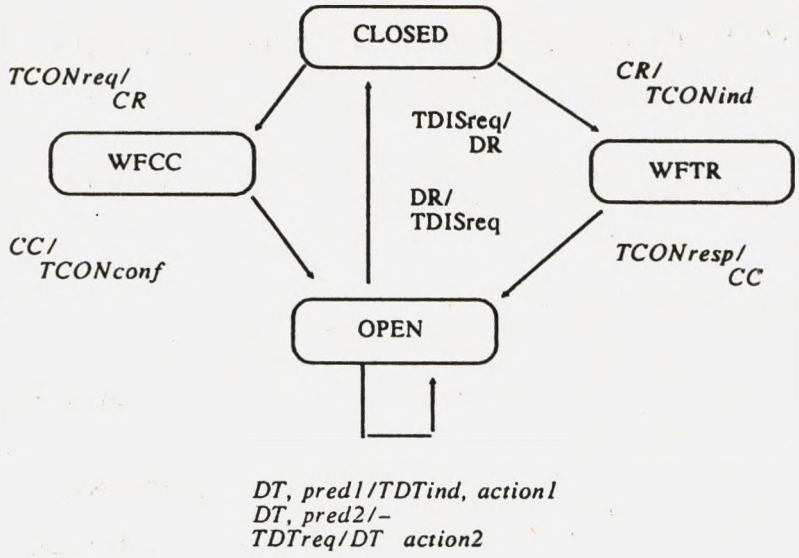


Fig. 2. Simplified state transition graph of transport protocol class 0

This is a very simple protocol, but the ASM model is capable of describing more complex protocols. The ASM model can be the basic model of the protocol in an expert system for protocol engineering. In the next section we demonstrate how this model can be stored in a frame structure.

The model with frames

Frames describing the ASM model can be divided into three groups:

- atomic,
- complex,
- pattern.

An atomic frame contains a state transition. The form of the generic frame for a transition is the following.

```
frame: transition;
    start_state:;
    input:;
    predicate:;
    action:;
    output:;
        meta_slot: output_meta;
            if_accessed_demon: output_demon;
        meta_end
    end_state:;
        meta_slot: end_state_meta;
            if_accessed_demon: end_state_demon;
        meta_end
end.
```

The slots of the meta-frames contain demons which are activated when the value of the slot is accessed during the operation of the program. These demons can be for instance procedures writing out the output and the endstate on the display. The frame of a transport protocol state transition is the following:

```
frame: edge5;
    is_a: transition;
    start_state: open;
    input: dt;
    predicate:;
        meta_slot:;
            if_accessed_demon: pred1;
        meta_end
    action: action1;
    output: tdtind;
    end_state: open;
end.
```

Complex frames store the information about the elements of the sets defining the ASM model. The following is the generic frame for the inputs, and the next frame represents the inputs of our transport protocol example:

```
frame: inputs;
    asp_from_upper_layer:;
    asp_from_lower_layer:;
    pdu:;
end.
```

frame: trans_inputs;
asp_from_upper_layer: tconreq, tconresp, tdisreq, tdtreq;
pdu: cr, cc, dr, dt;
end.

The bit format of the messages, their parameters with their values, can also be represented by complex frames.

Pattern frames store test suits, or subgraphs of the state transition graph. This information is available also in the atomic frames in the description of the transitions. Such redundant information is useful in the procedures of protocol engineering.

frame: pattern;
states:;
inputs:;
outputs:;
transitions:;
represented_actions:;
fixed_variables:;
end.

This is the example of a pattern frame representing the connection establishment phase of the transport protocol:

frame: connection_establishment;
is_a: pattern;
states: closed, wfcc, open;
inputs: tconreq, cc;
outputs: cr, tconconf;
transitions: edge1, edge2;
end.

4. PREDICATE-FRAMES

Frames are very useful for describing the ASM model of the protocol and so frames can describe the FSM model of the protocol similarly. To get the FSM model we have to remove predicates from the ASM model of the protocol, but we have to avoid state explosion.

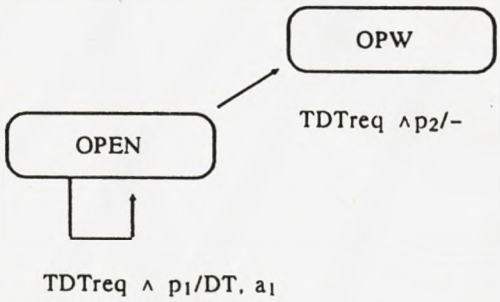
We have three kinds of predicates [TAR-88, TAR-89]:

- controlled
- settable, and
- uncontrolled.

Controlled predicates concern the local variables of the protocol. Local variables define a context.

If the local variables are discrete and bounded we can remove the controlled predicates by creating new states composed of the states of the ASM associated to values of the variable context. So the states of the FSM will be $<s_k, c_k>$ where s_k is the state of the ASM and c_k is a variable context and $<s_k, c_k>$ is the state of the FSM if and only if there exists a transition sequence of the ASM such that brings the FSM from the initial state and context $<s_0, c_0>$ to $<s_k, c_k>$.

The following is a subgraph of the graph model of the abstract state machine:



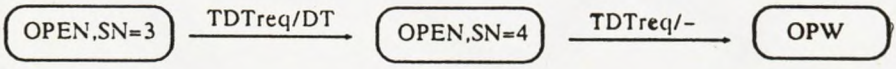
Here we assume that the user data carried by the TDTreq service primitive can be put into one DT protocol data unit.

The frame representing a controllable predicate is the following:

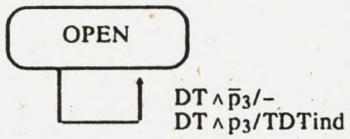
```
frame: p1;
    is_a:    demon;
    condition:   p1_cond;
    activity:    set_true;
end.
```

where p1_cond and set_true are definitions in MPROLOG, p1_cond represents the condition of p1, and if it is true in a given context of the variables the activity slot is activited.

To get the equivalent FSM graph we have to omit the predicates and actions from the graph above. Assume, that SWUE=4 and SN=3:



Settable predicates can be directly set by the peer entity or by the upper layer. They mainly concern the parameter values of protocol data units.



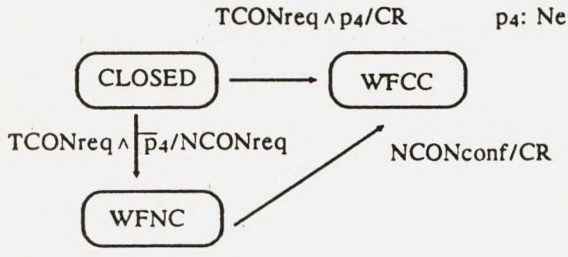
p3: EOT=1 (* end of text bit *)

To remove these predicates we have to treat them as particular inputs so these predicates don't increase the number of the states in the FSM.

Frames representing settable predicates contain the value of the parameters which determine the predicate itself. The frame representing a settable predicate:

```
frame: p3;
    type: settable;
    parameters: eot;
    values: 1;
end.
```

Uncontrolled predicates are not controlled completely by the entity, the peer entity or the upper layer, they usually depend on the communication between the entity and the lower layers. Let's see an example also for the transport protocol class 2:



As we can't change the value of these predicates, only the output message shows which transition fired.

p4 can be represented with the following frame:

```
frame: p4,
    type: uncontrolled,
    condition: "network connection is available",
end.
```

5. CONCLUSION

Some problems of protocol representation can be solved better with frames, others with rules. Therefore the representation space of FAIR with its orthogonality is really suitable for us and has inherent features for protocols. The frame-language can be considered as one axis, the logic-based MPROLOG with its rules as the other axis of the representation space of FAIR. The expressive power of the integrated language is very high and favourable in protocol fields.

A particular advantage is that the FAIR-frames can be built in our PROCONSUL (Protocol Consultant) system [KÖ-87, TAR-88, VAR-89] supporting protocol specification, validation, implementation and conformance testing. The simple and natural syntax, the user-defined relations, the control of inheritance and the structurability of FAIR produce a good and well arranged structure for protocol engineering realized in PROCONSUL.

ACKNOWLEDGEMENTS

The authors wish to express their gratitude to Peter Ecsedi Tóth (Computer Research and Innovation Centre) for the FAIR-system development and for his invaluable advice and illuminating suggestions.

REFERENCES

[AIT-86] Ait-Kaci, H., Nasr, R., "LOGIN: A Logic Programming Language with Built-in Inheritance", J. Log. Prog., 3 (1986), 185-215.

[BRAC-83] Brachman, R.J., Fikes, R.E., Levesque, H.J., "KRYPTON: A Functional Approach to Knowledge Representation", Comput. 16 (10) (1983), 67-74.

[BRAN-83] Brand, D. and Zafiropulo, P., "On communicating finite state machines", J. of ACM, 30 (1983) 323-342.

[CAS-87] Castanet, R., R. Sijelmassi, "Methods and semiautomatic tools for preparing distributed testing", In: Prot. spec., testing and verif. VI. (1987) 177-190.

[CLO-81] Clocksin, W., Mellish, C., "Programming in PROLOG", Springer Verlag, (1981)

[ECS-89a] Ecsedi Tóth,P., Wagner, P., "FAIR, a frame-based system integrated with MPROLOG", In: Proc. of Seminar on Expert Systems, Visegrád, 23-24 Jan. (1989)

[ECS-89b] Ecsedi Tóth, P., Wagner, P., "Backtrackable frame in logic programming", In: Proc. of. First Finnish-Hungarian workshop on Programming Languages and Software Tools, Szeged, 8-11 Aug. (1989)

[HAR-85] Harmon, P., King, D., "Expert Systems", Wiley and Son (1985)

[KÖ-87] König, H., Tarnay, K., "An intelligent protocol workstation", KFKI report 45-M, Budapest, (1987)

[RES-87] Results of Survey on Trends in Expert Systems in Japan, Future Generations Computer Systems 3 (1987), 17-36.

[TAR-88] Tarnay, K., Wagner-Dibuz, S., "A protocol consultant", In: Proc. of DECUS Europe Symposium, Cannes, 5-9 Sept. (1988)

[VAR-89] Varga, L. Zs., Tarnay, K., Wagner-Dibuz, S., "A knowledge base for protocols", In: Proc. of Decus Europe Symposium, The Hague, 18-22 Sept. (1989)

SESSION 7

DATA PROTECTION AND SECURITY

COMPUTER NETWORKS, PRIVACY PROTECTION AND SECURITY

H.P. GASSMANN,

OECD Paris

The issue of privacy protection in computer systems is now over 20 years old. In 1968, in his book "Privacy and Freedom", Prof. Alan Westin from Columbia University in the United States argued that the plans of the US Federal Government to create a super-large data bank containing the names, addresses and personal characteristics of all US citizens, a sort of "Fort Knox of personal data", would amount to a massive invasion of the American citizens' privacy, and would practically make true George Orwell's prediction of "Big brother is watching you" in his now famous fiction book "1984". Although such a centralised, universal data bank was never created, the book sparked off a lively, international discussion on the potential dangers of computerised data banks, especially in public administration, for invading the privacy of the individual.

Interestingly enough, Westin's ideas were taken more seriously in Europe than in the United States. It is difficult to exactly know why - but two factors may have contributed: First, in post-war Europe, in some countries the memory of the use of personal files before and during the war by security police such as the Nazi Gestapo was still vivid; and second, in many European countries public administration was much more precise and person-oriented than in the United States, for example with the obligation made to citizens to always carry personal identity cards, and even using personal identification numbers for administrative purposes, such as social security or driver's licence.

The world's first data protection law was adopted in Germany, in the Land of Hessen, in 1970. It was addressed to the public administration of that State of the Federal Republic and therefore had a rather limited scope. The first country which introduced a full-fledged, comprehensive national privacy protection law, covering both the public as well as the private sector, was Sweden in 1973. During the rest of the '70s various countries followed: in European countries, these laws were sweeping and comprehensive, more or less following the Swedish model (so-called "omnibus laws"), covering both the public and private sectors, and often not only physical persons but also legal persons. This latter approach seemed to many to be stretching human rights considerations too far, and the international business community developed first an awareness and then effective

lobbying, so that new privacy protection laws passed after 1980 no longer covered legal persons. In the United States, in contrast, omnibus laws of this type were not considered useful, no doubt because of the sheer size of that country, but also because of its federal structure. Rather, sector-specific privacy protection laws were introduced for credit reporting, banking, the federal government's data files (as an aftermath to Watergate in 1974), and for some state governments.

At the international level the period of the 1970s also generated the Council of Europe's Convention for the Protection of Individuals with Regard to Automatic Processing of Personal Data as well as the OECD's Guidelines on the Protection of Privacy and Transborder Flows of Personal Data, both signed in 1980. These two international instruments helped in developing a certain international harmonization of common principles on which national privacy protection legislation was built.

The privacy protection (called also data protection, or in French, informatique et liberté) movement of the 1970s reflected the *Zeitgeist*, that on the one hand information technology was regarded as very important for further economic development and growth, but that on the other it needed some social control framework to make sure that its negative impacts were minimised. In retrospect this was an important movement, as it permitted a rather wide public discussion about avoiding possible dangers in computer technology, certainly more so than was the case with nuclear technology. It may well be possible that today many of those involved in civil nuclear technology development regret not having had a similar public discussion in the early stages of nuclear technology development, as this would have permitted a better explanation of the merits, but also the dangers of this technology, and thus may have avoided some of the excessive negative public reactions we have seen in the past years in that field.

During the early-to-mid 1980s there appeared to be a certain decrease of public interest in privacy protection in OECD countries, partly because in those countries which had developed such laws, the public considered that the issue was properly addressed, partly also because the *Zeitgeist* of the early-to-mid 1980s was in favour of liberalisation and deregulation. At the international level this meant that trade in services issues became the focus of attention. This was already perceived and forecast at an OECD conference on transborder data flows held in London in 1983, and was reinforced by the start of the GATT Uruguay Round negotiations in 1986, where for the first time international trade in services was also addressed. These negotiations still continue today and, as part of them, the relatively small, but strategic area of telecommunications services and telecommunications network-based services (TNS) commands special attention.

As of 1988 interest in privacy protection has been revived, especially in those countries not yet possessing such legislation. In 1988 in four OECD Member countries - Ireland, Japan, Australia and the Netherlands - privacy protection legislation was passed. Thus, 16 out of 24 OECD Member countries now have privacy protection legislation, and in all of them the principles contained in Part II of the OECD Guidelines can be found. There are eight principles. They are as follows:

The Collection Limitation Principle;
The Data Quality Principle;
The Purpose Specification Principle;
The Use Limitation Principle;
The Security Safeguards Principle;
The Openness Principle;
The Individual Participation Principle;
The Accountability Principle.

These principles are practically identical to those of the Council of Europe Convention. The OECD Guidelines constitute a good example of international "soft law", because they have led to legislation based on these principles in the abovementioned countries. But in addition, they have also been adopted by private institutions such as trade associations as recommendations to their members; for instance by IATA, the international airlines association, to their members world-wide; by the Canadian Bankers Association to its members (because the Canadian Privacy Protection legislation does not cover the private sector); by FISC, the Japanese association of financial information systems (especially credit information) to its members. Thus, these Guidelines constitute a flexible instrument, which has in its ten years of existence well proven its influence. They had a double effect: first, contributing to some international harmonization of privacy protection, and second, in building international trust in the use of computerised personal information systems.

In the first half of the '80s the growing internationalisation of computer systems linked through telecommunications became known to give effect to so-called Transborder Data Flows, which was a term used quite widely, until it became subsumed in that of international data communications, or more generally in international telecommunications services, or data networks. In 1985, the governments of the 24 OECD Member countries adopted a Declaration on Transborder Data Flows. In it, they agreed to:

1) Promote access to data and information and related services, and avoid the creation of unjustified barriers to the international exchange of data and information;

2) Seek transparency in regulations and policies relating to information, computer and communications services affecting transborder data flows;

3) Develop common approaches to dealing with issues related to transborder data flows, and, when appropriate, develop harmonised solutions;

4) Consider possible implications for other countries when dealing with issues related to transborder data flows.

This declaration can be considered as a precursor to the already mentioned negotiations on trade in services actually taking place in the framework of the GATT Uruguay Round, and already emphasises the access issue, which is all-important in telecommunications in general and in data communications in particular. The Declaration promotes a liberal attitude, and requests transparency in rules.

In the meantime, computer and communication technologies have continued to make great advances. Miniaturisation of equipment, more sophisticated software and digitalisation in telecommunications have contributed to rendering these technologies more and more indispensable for a modern economy. Especially the opening of digital networks (such as ISDN), both public and private, has revived a certain interest in privacy protection.

Interestingly enough, it seems that the telecommunications services providers, especially the public telecommunications operators (PTOs), are keenly aware of their customers' desires to protect their privacy. For example, the German *Bundespost* has on various occasions stressed the importance of data protection in the introduction of ISDN; in the United States some of the new services offered by digital 'intelligent networks', such as identification of the number of an incoming call, have stirred up uneasiness among the telecommunications carriers' customers.

In these instances the dividing line between 'privacy protection' and 'data security' has become thinner. Sometimes the term 'data protection' is also used to cover data security, especially in those countries where data protection is used as a synonym of privacy protection.

Since about 1986 the attention of users of computer/telecommunications services has increasingly turned to the development of mostly private computer 'networks', where it is often difficult to make a distinction between national and international use, especially in large multinational corporations, banks or airlines. While of course also some personal data flow over these networks, the overwhelming majority of the data moved are management data necessary for the daily business of corporations - financial, logistics, orders, data accompanying trade, statistics, etc.

Attention has also increasingly turned towards the physical and logical security of these networks, as stories of unauthorised access to computer systems - often beyond national borders - by 'hackers' have been reported by the press.

This raises the general question of how dependent we are today upon these computer networks, how reliable they are, and what can be done to improve their security and integrity, which is widely believed to be not quite satisfactory.

The first of these questions is easily answered: our dependence upon these systems is great and growing by the day. These systems are indeed marvellous tools for increasing productivity, speeding up operations, making possible new applications unknown before, bringing about new business opportunities. There is no question of limiting this development, as it continues to bring more intelligence to all kinds of economic activities.

It can be said that these systems are, by and large, quite reliable, otherwise they would not be used. To be sure, we all know of cases when the computer system has been momentarily 'down', which means temporary interruption of service with unpleasant consequences, as experienced with airline reservation systems, hotel management systems or even office automation. But when the computer system is operating again, the inconveniences the breakdown has caused are quickly forgotten. Vulnerability to these systems is potentially there, but we may be even more vulnerable to blackouts of electricity supply.

This is not to say that the situation is perfect, and, as the hacker stories have proven, there is a need to improve the security of these computer-communications systems, not only nationally but internationally as well.

Given the rapid development of computer-communication systems and corporation 'networking' with often international scope, what is perhaps needed are some new international rules of how to play this new computer networking game, and to build greater trust of the public in the operation of these systems. 'Public' here means users, which are the enterprises, banks, airlines and other transport operators, but also the public at large, the customers of banks and airlines, for instance.

This may involve agreement on minimum standards for:

- network security
- privacy protection
- system reliability tolerance
- responsibility in case of breakdown

In order to establish such new minimum standards it is necessary to develop a largely based consensus between governments, telecommunications operators, privacy protection authorities, users - as senders and receivers of data, and insurance companies. This could mean that privacy protection aspects, so far treated as a distinct, largely human rights-driven issue, could increasingly become part of a larger set of minimum standards to be observed for safe and reliable computer network operation.

Several developments point to the need for such an international, integrated approach:

1. New technical developments in digital telecommunications networks;
2. Growing internationalisation of economies;
3. Growing use of computer networking;
4. Growing awareness of the vulnerability of such networks to external and even internal attack.

Such an integrated approach may have several advantages:

1) It would constitute a comprehensive, easily understandable framework, which would tell the actors involved in international data transmission what all the rules of their game are.

2) It would avoid a piecemeal approach, which has the danger that it may confuse the users, or that they would pay attention only to those aspects which they understand or know, but not to other issues which might be as important.

3) It may constitute a "level playing field" for all users of data networks. It can indeed be observed that these users do not object to new regulations as such - if they feel that there is some rationale behind them; but they strongly object to important differences in such regulations from country to country. This desire for similar treatment increases with the degree of internationalisation of the users: international users want international uniformity of rules.

So for the next years there is an important goal ahead for ensuring that computer networks develop harmoniously in many countries: to the internationalisation of these networks must correspond an internationalisation of adequate usage rules. It is to be hoped that the international community will agree on such rules rapidly, because the information infrastructures are becoming more important by the day, not only in a particular country, but in all countries.

Note: The opinions expressed here are the author's own and not necessarily those of the organisation to which he belongs.

ISSUES IN COMPUTING SECURITY AND TELECOMMUNICATIONS NETWORK INTEGRITY

Jan Freese

Deputy Director General
Federation of Swedish Industries
Stockholm, Sweden

COMPUTING SECURITY

It is often said that the computer cannot do anything that man cannot do. This might be true - in theory - but definitely not in practice.

One reason is that the computer is time saving. Moreover the computer makes it possible to collect huge data bases and still keep a survey of them. The information can be kept up to date and accessible in a way that was impossible before. It can be kept for a long time without loss of accessibility. Any item stored can be used as key to the denominator for the category of data you want to select from a data base. It has become possible to link data bases in new ways, and the results of linking, or the sums, are often larger than the individual parts. The reason is that usually the information in each linked data base is collected for a different purpose.

The most simple example can be picked from the weather forecast. Man has always been able to find out the weather five days ahead - "manually", but then the weather has passed by long before the calculations were ended. The computer makes it in time.

The computer is an advanced instrument and the limits to its use are still unknown. Its just a question of your CAI (Computer Aided Imagination).

The construction of the computer makes it safer, quicker, and cheaper than any mechanical construction. One of the computer's characteristics is its ability to process information in accordance with a program which, moreover, can change itself according to the results obtained. And still it is no more stimulating to watch a computer in operation than to watch a defrosting refrigerator.

In spite of the computer being safer than any other mechanical construction a Swedish committee on the vulnerability of computer systems stated already in 1978 that the vulnerability of today's computerized society is unacceptable high.

The vulnerability depends partly on traditional problems, partly on completely new possibilities.

All countries are vulnerable, but not always in the same way. A country dependent on export/import and high technology is already vulnerable. The margins of robustness have shrunk. Computerization must not increase

vulnerability, but very often the way in which societies have been computerized has led to growing vulnerability.

Undeveloped or less-developed countries are vulnerable because of their lack of technology. This section, however, will deal with increasing vulnerability caused by methods of computerization.

The developed countries suffer from a number of vulnerability factors, such as criminal acts, misuse for political purposes, acts of war, catastrophes and accidents, information of a confidential nature, functionally sensitive systems, concentration, integration and interdependence, etc.

Particular attention should be paid to criminal acts of a relatively serious kind and of such a nature as to constitute a serious threat to an organization or to a whole country. The criminal offences of main interest are those such as sabotage and industrial and other espionage. Other types of crime may also be of concern, such as various crimes against property. The growing computerization of payment transactions may give rise to crimes against property of entirely new dimensions. The traditional bank robber is slowly losing his job, which is being taken over by skilled persons with a knowledge of computers. A modern computerized bank robbery may concern millions but is often suppressed in order not to hurt confidence in banking. Terrorism with a political aim may in future become a means of aggresion between states, apart from conventional wars. There is no doubt that data bases and computer systems will also be used or misused in the struggle between different political groups within a country.

Computer crimes of various kinds occur today. Such crimes are often extremely difficult to detect, and the crimes that have been detected undoubtedly constitute only a fraction of those committed. Large quantities of data, the possibility of processing them, data communication, etc will increase the risk of different forms of espionage. As computers and information-processing systems have already been the object of terrorist activities in some countries, there is reason to believe that such activities may extend to other countries.

Data traffic from a given subscriber can be tapped in the local network on the line between the subscriber and the nearest exchange. To tap a particular connection in the central parts of the telephone network is complicated and may require advanced equipment. Today, however, very few if any telephone system can guarantee that the network is free from bugs.

Besides, it is not always necessary to use very advanced equipemnt. Already there are electric typewriters which as a side effect transmit signals that can be picked up and recorded. It is usually not very meaningful, of course, to tape what is written on a typewriter. On the other hand terminals and word processors may be of great interest. Often it is possible to pick up the electronic beams from such equipment through the walls of an office. A portable television monitor in the street outside a building may get a better picture -- possible to videotape -- than the ordinary television programme.

There is much more creating a vulnerable computer society. In view of many countries' high degree of dependency on imported computer equipment or services the threat of economic sanctions may be mentioned.

So may the dependency on international data-transmission. It is sometimes not realistic to rely on other safety measures than encryption.

Among other factors that will add to countries' vulnerability in crisis (such as wartime, lack of electricity and limited telecommunications capacity) is the increased interdependence of different data processing systems so that the elemination of one system may have effect on others. Even if it today may go to far to mention nuclear weapons'electromagnetic pulse (EMP) effect, such effects can be the result of a thunderstorm.

Landslides and earthquakes are other dramatic threats which can disturb computer systems. Still there are more common problems.

Populations registers, used as the framework of public and sometimes other kinds of administration, may be an important source of vulnerability. In some developed countries some of the most comprehensive population records, such as those of the national social insurance system, the national registration and taxation authorities, and the automobile and driving-licence agencies, must not be disturbed. Another type of important register is co-ordinate-marked personal tapes, usually kept by authorities responsible for property data.

Other records of a confidential nature are those containing company information and the like. Some government agencies have the right in principle to collect all data they consider they need for their work. In recent years, therefore, growing quantities of company data have been stored in public data processing systems - often far away from the source where the confidentiality of the data were known.

In certain local government ADP systems, furthermore, there may be detailed technical and geographical data concerning, for instance, the electricity, water, sewage, and gas systems.

Many sensitive communal functions are computerized, such as the administration of comprehensive financial security systems; banking, which has increasingly computerized payment transactions; and trading, with computerized distribution, stock-keeping, etc. Computer systems are also used for supervision and control of production processes in iron and steel works, paper mills, and other industries, for traffic control, etc. Disturbances in systems of these kinds may quickly have effects in the form of delayed payment or non-payment, difficulty in getting out orders to production units, with the consequent disruption of production, lack of control over stocks and distribution, with consequent shortage of goods and disruption of production, traffic disturbances, etc.

There are at least two kinds of concentration of ADP operations -- geographical and functional concentration. By geographical concentration is meant a great accumulation of computer capacity in certain areas. By functional concentration is meant large central systems or large service bureaux with centralized operation and many customers. Both geographical and functional concentration occur to a large extent in many countries. Increased -- functional and geographical -- dispersion of computer capacity may have distinct advantages in reducing vulnerability. One reason is that the elimination of a small unit can never have such disastrous consequences as the destruction of a large central system, and the new generation of big units will have a capacity of a million transactions a second.

There is today a considerable flow of information between different systems. Many systems depend on recieving information from one another. Already, much of this information is exchanged on computer-to-computer circuits, and this will become increasingly common, giving rise to system interdependence because, in order to be able to communicate with one another, different systems must use the same procedures and standards in various respects.

In this connection must also be mentioned "Just in time" knitting together the supplier of rawmaterials, subcontractors, factories, transport companies, wholesale dealers, shops, clients, banks etc to a computerized cobweb.

Personnel play a key role. System engeneers and programmers can build up complicated systems which no one other then they themselves can master. If documentation is lacking or defective, the user is in the hands of the system builders.

One often forgotten vulnerability factor is related to the labour market; it is relevant to recall that the word "sabotage" comes from the French word sabot -- the clog thrown into power looms by laid-off workers. There have been cases of dissatisfied, dishonourable, or undependable staff members causing damage by, for example, destroying information in computer files. It must not be forgotten that most illegal computer uses have been inside jobs. At the same time increased education will increase the number of external hackers and crackers.

The fact that so much is dependent on ADP staff has also been used in labour conflicts. A company's or other organization's operations can be very critically affected just by having the ADP staff or even only a small part of the ADP staff go out on strike.

Often systems are made unnecessarily complicated owing to demands for rapid flows of information within and between them despite the absence of any particular need.

For lack of time and economic reasons it often happens that successive changes in ADP systems are not adequately documented. The need for documentation is especially marked when a change is being made to a computer of a new model or a new generation, as well as in conjunction with operational disturbances or change of staff.

The most important point in preparedness for an emergency is that such situations have been planned for. The preparedness must include a check to see that the plans will function and the appointment of a person responsible for the security. The object of emergency planning is that, if serious damage occurs, its consequences will be counteracted and alleviated as far as most feasible use today appears to be the encryption of information to be transmitted on data links, but the encryption of files would in most cases be of greater value than the encryption of transmitted information, though it is more difficult to handle

As regards legitimacy, some suppliers can today offer different forms of complete arrangements built into hardware and software.

Technical and technical-economic developments continue to facilitate the spread of computer capacity. In particular, computers are becoming

smaller, faster and cheaper. Price and technique together, therefore, mean that all users, whether government agencies or others, can acquire computers and systems adapted to their individual needs. This reduces vulnerability in some ways but may also be a source of new vulnerability problems.

Standardization of hardware, software, documentation, etc can help to reduce vulnerability by, for instance, providing better back-up facilities, reduced dependence on staff, and also less dependence on foreign sources. It is therefore important that continuing attention be given to this matter and that standardization questions be considered in purchasing computer systems.

What has been mentioned so far is a part of the catalogue of threats. Lucky enough most of them will not hurt you. Besides there are lots of measures against them.

Swedish committee on the vulnerability of the computerized society, appointed in 1981, proposed a project to develop a simple, practical method, usable both by companies and by government bodies, to assess vulnerability. This method, which later came to be known as vulnerability analysis (SBA), (1) should result in safer computer systems in companies and government agencies, and (2) by improving security in large, important systems, should reduce the total vulnerability of society. The SBA method would quite simply be a step towards a more robust society.

The computerized organization or society has gained a lot of creativity via the marvellous computer. Still it has also become somewhat more boring compared to the "paper and pencil" society. The computerised society demands much more law and order.

Today there is also a legal vulnerability depending on too old legislation. All kinds of laws has to be updated in order to be able to meet e.g. computer crime or EDI and ISDN.

TELECOMMUNICATIONS NETWORK INTEGRITY

Business practices based on the use of information and communications systems are viable only insofar as the users of them are assured of the security of the information held and exchanged within them. However, it is recognised that this information is subject to unlawful and unauthorised access, interception or manipulation, resulting in both financial loss and commercial damage. And whatever the number of cases until now may have been they will increase.

Progress towards the extensive use of communications networks within and between businesses has been inhibited by these threats to security. There is an absence of widely accepted and implemented international standards and a lack of general agreement on the correct technical approach to this problem. This is exacerbated by legal uncertainty over the whole question of unauthorised access to, and theft and manipulation of, data. In many cases, businesses have responded by continuing manual checking and control routines, the existence of which seriously erodes the financial benefits of data networking.

In the international arena, differences in national laws and practice with regard to the protection of data carried on communications networks only serves to compound the problem. Concerted action at an international level in a number of areas offers the best prospect of achieving a more secure environment for the electronic conduct of business.

Telecommunications networks are normally categorised by ownership as public or generally available services provided by the so-called public carriers, private or networks set up by businesses for their own specific purposes or "third party", networks provided by service companies for general or restricted purposes. As a rule, the "transmission infrastructure" are monopoly services provided by the public carriers. The circuits interconnection different locations which typically form private and third party networks are, therefore, furnished from within the public network. Private networks, or elements of them, restricted to a single set of premises may, however, be provided by user companies themselves and do not form part of the public telecommunications infrastructure.

The national public carriers provide similar services in the different countries of the world. Typically these are telephone, telex and data networks. In future most types of communications services will be provided through a single network, known as ISDN. Carriers are also able to provide private companies and other organisations with leased lines.

The types of public data network in widespread use today are the public packet switched network and the circuit switched data network. Private networks can be divided in e.g. PABX, LANS, WANS or hybrids of these.

There are different network topologies e.g. star, ring or bus networks and hybrids of them carrying different kinds of problems. There are also different transmission media from traditional twisted part copper cables, to coaxial cables, fibre-optic and microwave and satellite systems.

In such an environment it is not always easy to recommend security measures, as they must adapt to what choice the user makes.

Besides it must always be remembered that 100 percent security does not exist and should not. It would create an increadible and impossible society. There must be a balance between security measures against for example computer related crime and reasonable access to systems for authorised users.

Societies have become more and more international and so has business. The flows of information has created vulnerabilities. The threats are e.g. unauthorised use of terminals, wiretapping and manipulation, physical damage or communications equipment as well as natural disasters and other accidental events.

An intruder can be anywhere, hitting the network service at will and without any trace or evidence of his presence. The user must therefore consider his network as unguarded and identify the need of protection, the sooner the better. It is always cheaper to buy the security during the specification, tendering, installation and implementation of a system than later on.

The counter measures include both measures to restrict the use of the network and to protect data passing through it.

The restriction of use can be solved by forming closed user groups.

The protection of data covers a lot of measures:

- provision of resilient systems,

- provision of network management systems, allowing fault reporting, diagnosis and prompt clearance,

- the use of proven communications protocols which ensure that transmissions are complete and that orderly recovery from fault conditions occurs,

- error detection and correction techniques which protect data "bits" from corruption due to noise in the transmission systems,

- the use of protected transmission systems,

- the use of message authentication techniques to provide a measure or assurance that the information content of the transmission has not been altered in transit,

- the use of cryptographic techniques to provide various levels of protection against intrusion, electronic eavesdropping, data insertion, deletion and manipulation.

Cryptographic techniques and key management are important counter measures. Encryption is typically used to conceal the contents of an entire data message or fire, to protect the checksum in authentication systems, to conceal password information which has to be sent over the network for verification and to construct electronic signatures.

Until recently, the normal way of using the encryption technique has been to protect vulnerable transmission links. The algorithm and the same secret key are then used at both ends of the link. The management of the keys is a matter for the network operational staff.

For a long time, the problem of key management in symmetrical encryption scenarios imposed constraints on the applications of cryptography. However, experts have devised asymmetrical cryptoalgorithms in which there are matching pairs of keys, one for encryption and another for decryption.

Each user has his own pair of keys. He keeps one (the decryption key) secret and makes the other publicly available to all potential correspondents. Because ot the nature of the algorithm (so-called one-way algorithms) it is not possible either to decrypt a message with the public key or to determine the secret key from a combination of the algorithm and the public key.

In addition to the security problems of key storage, handling keys also presents organisational difficulties. Key management, which is defined as a set of procedures to generate, distribute, store and destroy keys, is

difficult and expensive. If it is not carried out properly, the protection obtained may be illusory.

Dynamic passwords are more secure than static passwords because they change on each occation of logging onto the access control system and, possibly, for every message sent. Dynamic passwords are electronically generated through "smart" cards and other devices built into terminal equipment and software systems.

It has, however, to be mentioned that some countries do not allow the export of products incorporating cryptographic hardware or software developed in that country.

The responsibility for network security is divided between the security administrator, the network provider, the public carrier, the external service provider and the user. This situation creates the need for welldefined interfaces. And don't forget that the final responsibility in any of these parts is placed on the top of each organization.

Many actions are possible in order to improve security and they are more or less the same either we are speaking about data integrity or network security.

There is a need for a security plan and someone in charge of the responsibility for security. The staff must be properly trained to handle equipment and in security routines. Transmission security must be based upon using proper communications protocols. There is of course necessary to use access control, and access rights must frequently be revieved. The physical security must not be forgotten. Proper key administration is vital as well as up-to-date documentation. Finally find out who you hire, especially for sensitive positions. The alternatives you may face can cost you much more, if not, do not overdo it.

It has not been possible to cover the subjects better than a water-spider - sweeping over a large surface without always realizing the depth of the ocean of computerizations.

At last it has, however, to be mentioned that we are all in the same boat. We need international agreements and jurisdiction concerning computer and communication security - up to-date-rules -. So it is in most fields (administrative rules, procedural rules, penal codes, intellectual property rights etc). When we will get them we will feel a wee bet safer.

ASPECTS OF SECURITY, INTEGRITY AND PRIVACY IN THE TELECOMMUNICATION NETWORK

Ornella AMBROIS

CSELT. Centro Studi e Laboratori Telecomunicazioni
Turin, Italy

The paper is written from a telecommunication operating company point of view. It aims at the outlining of the requirements and technical approach in the areas of security, integrity and privacy of the telecommunication network.
These issues were of highest concern from the very beginning of the telecommunication operation, however the penetration of the new software technologies and the availability of new services pose new demands and challenges. In particular the paper focuses on the aspects of network operation and administration taking into consideration the impact of decentralized operations both in terms of the operating company staff and the large business customer that require access and operation capabilities of their private virtual networks.
The last part of the paper deals with some perspective in the area of consensus at the international level. Work is going on both at international standardization organizations, such as the CCITT, and in joint cooperative projects, such as the European Community RACE.

1. TELECOMMUNICATION AND SECURITY ISSUES

As telecommunications networks become more powerful and widespread the needs of security grow. More and more sensitive information is processed and carried over the telecommunications networks and many business sectors are now dependent on the permanent availability, reliability and security of telecommunication services. In the Integrated Broadband Communication (IBC), the integration into a channel system of all telecommunication services could have effect on the vulnerability of single applications.
Security implications differ to a certain extent in public network with respect to computer network, in particular in the telecommunications:
- the services are provided via interconnected public networks, where public refers to accessibility and not to ownership;
- the environment will never be homogeneous and will continually change;
- large amount of software functions and data have to be handled;
- integrate special networks like mobile and satellite;
- the availability has to be granted to a virtually unlimited number of users requiring a powerful and reliable management.

Both telecommunication and computer networks
- are providing access and will allow operation capabilities to customer;
- may require different degrees of security to support different services or customer types.

Five basic sources of Security requirements have been identified. These are:
- Security requirements from PTTs;
- Security requirements from value-added service providers;
- Legislative requirements;
- Quality of Service requirements;
- Conformance/convergence requirements from standards.

The PTT requirements arising from the need to protect objects and data: data related to control, like service registration, routing, configuration, remote accesses, data related to

measurement, audit data, traffic charge and so on. The network provider has also performance quality requirements to support those services with special security and integrity performance.
The **value-added providers** have different requirements strongly depending on the sector of activity: need of Integrity and Authentication in public sectors like Educational and also confidentiality need in Justice, Defense, Government, Finance. The private sector includes banking, finance, retailing, insurance and computer trading, which have high demand for security measures, with financial, commercial and legal risks to contend with. The protection of data, text, voice and images in the office system domain are of importance.
Legal requirements of computer and information technology like propriety rights and individual rights must be considered. These arise from necessity to protect the propriety of information stored electronically (copyright, issue/copy control, charging for use) as well as protecting the privacy of information related to individual.
In the context of the **Quality of Service** certain supplementary security services may be provided to some services which require special performance (acceptable residual bit error rates, connection set-up delay, appropriate cryptographic techniques). I.e. services which make use of mechanisms such as digital signature or data integrity require a nearly error free transmission.
There is still to consider as a requirements the conformance to the other relevant **standards** which already exist in the security area.
In order to provide secure systems three aspects must be considered:
- vulnerabilities of a system and potential threats to systems;
- security services to protect against those threats;
- integrity mechanisms which implement those services.

2. THREATS AND SECURITY SERVICES

The communications networks, due to their distributed nature, are potentially sensitive to a variety of security threats. A short description of principal threats follows.
Accidental threats due to natural failures or network entities operating in an unintentional incorrect manner: these may result in sporadic, partial or total loss of service and data.
- *Operational errors,* due to entering of incorrect data;
- *transmission errors*, due to noise, radiation, diffraction of signals;
- *network resources failures*, due to natural wear and tear, ageing or component failure;
- *network resources overload,* due to the existent network topology or simply to an exceptional simultaneous use of critical network elements;
- *natural calamity* like lightning strikes, floods, earthquake, etc.

Intentional threats come from people getting or manipulating illegitimately information or network resources, or preventing legitimate users from using services.
Inter-communication threats occur during a communication between network entities owing to an external unauthorized entity:
- *masquerade*, an entity claims to be another entity to get restricted information, resources, or intercepts a communication then retransmits it claiming to be the originator;
- *leakage of information*, a breach of confidentiality by unauthorized access or monitoring;
- *loss or corruption of information,* the integrity of data is compromised by unauthorized deletion, insertion, modification, reordering, replay, reflection or delay.

Intra-communication threats by one or more entities who participate in the communication itself:
- *repudiation,* an entity involved in a communication exchange subsequently deny the fact;
forgery, an entity fabricate information and claims that such information was received from another entity;
security label violation, an entity attempts to access or to communicate labelled data in violation of the security policy in force.

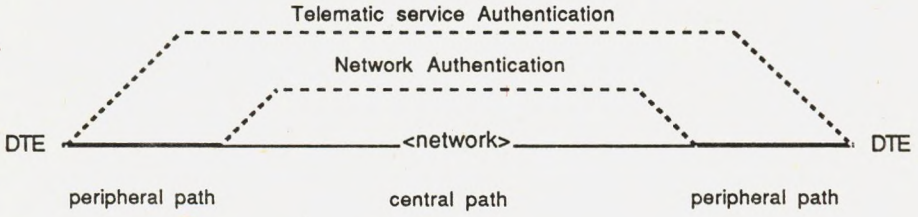
Network resource threats towards those elements of the network which store or process communicated data
- *unauthorized access* from entity guessing, stealing or buying personal authentication data and/or token.

- *Trojan horses*, programmes which contains additional non declared functionalities;
- *trapdoors*, code that allows compromise of software (i.e. operating systems), generally used during the development process by programmers.
- *viruses*, programmes which cause side effects when a particular system state is achieved;
- *covert channels: covert storage channels* are network storage areas which may be written to by a program and read by another entity who is not allowed; *covert timing channels* are the result of a program modulating its usage of the networksystem in some way which can be understood by another entity (e.g. communication rate).
- *electromagnetic emanation* ;
- *denial of service*, a network entity prevents other network entities from having access to their authorized network services and resources;
- *physical attack* via a physical or electrical damage to elements or indirectly via an attack to the hardware or software elements security mechanisms;
- *scavenging*, obtaining information left in or around a computer system after the execution of a job.

Administrative threats when insufficient security administration may lead to violation of the network security policy
- *insufficient security documentation;*
- *insufficient user education;*
- *unnoticed violation of the security policy;*
- *abuse of privileges.*

A well-defined *security policy* must identify which resources are to be protected and how. Given identified subjects and objects, a set of rules are used to determine the manipulation that subjects are allowed to perform on objects, and to enforce such authorizations.
A set of OSI security services has been identified.
Authentication: authentication refers to the proof that subjects or objects are indeed what they claimed to be. In communication it allows to identify a remote user to another entity of the network (User authentication), entities communicating to each other (Peer entity authentication) and, in a connectionless communication, the identity of the entity which has created the data (Data origin authentication).
Access control: access control ensures that objects are accessed by subjects in authorized manner only, it concerns access to the physical system, system software, applications and data. In a network, access control ensures that communication resource are accessed by authorized users in authorized way only.
Data confidentiality: data confidentiality ensures that data is not made available nor disclosed to unauthorized subjects. During the transmission across the network we can have: confidentiality of all data on a connection (connection confidentiality); confidentiality of all data in a connectionless instance of communication (connectionless confidentiality); confidentiality of selective fields within data, either on a connection or in a connectionless communication (selective field confidentiality); protection against traffic analysis (traffic flow confidentiality).
Integrity: integrity ensures that interaction leave data, applications and systems in a correct and available state. Data integrity ensures that data is neither destroyed nor improperly modified; applications integrity ensures that an application continues to operate according to its specifications; system integrity aims at the availability and correct working of the entire system. In a network with sensitive data the detection of accidental errors, intentional errors and manipulations (addition, removal, replay, delay, reorder) of data is of real importance.
Non repudiation: the non repudiation service provide to the recipient of data a proof of origin, to the sender a proof of submission or receipt.
Reliability: reliability means that services are available when needed, and perform correctly when used.
There are some not OSI additional services like **anonymity** and **notarization.**
Basic security and integrity mechanisms such as encipherment, hash, authentication exchange, key management, digital signature, access control, are available and are used, when ever needed, to support the previously described security services.
An example of the Italian Administration (SIP) approach to the security area is the authentication model for the telematic network.

The adoption of authentication techniques has administrative and contractual purpose, in fact to a technical level there are mechanisms which guarantee the integrity of the exchanged data and a reliable routing. The authentication purpose is to guarantee the integrity and authenticity with faults and attacks, for that it must be able to resolve legal questions.
When the communication goes through more logical entities (countries, administrations, networks) each entity has to guarantee its domain at the interfacing points (link to link), otherwise within an entity the authentication modality is end to end, to minimize the enciphering key management cost.
Each communication has three logical paths to define the responsibility domain:
- two peripheral paths under supplementary service provider authority;
- one central path under bearer service provider authority.



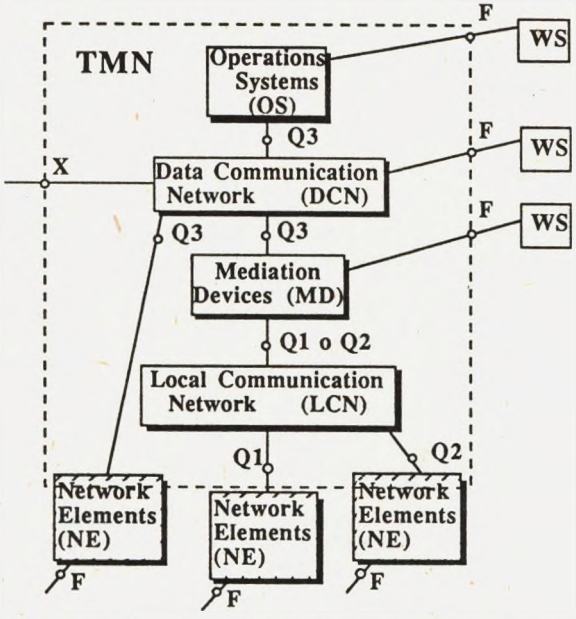
DTE: Data Terminal Equipment

The model defines the overlapping of two different authentication tipologies
- network authentication which is a bearer network service and hence it must be provided by the Network Operator;
- telematic service authentication which is an optional end to end service outside of the network management.

3. PUBLIC NETWORK MANAGEMENT

Some indications on security mechanisms for the protection of the operational network management in public networks follow.
The general situation is quite heterogeneous, there are no standard and not even a "de facto" standard, in an environment with a mix of old and more recent equipments: the new introduced networks have hardware and software management designed and implemented by manufacturers, the studies and solutions adopted by the PTTs are usually specific and "ad hoc".
It is still common to many systems having large part of management done by manual routines and requiring the management personnel to be physically present in the network nodes or special maintenance centers. Many terminals are only locally connected and can not have remote access. For these reasons the security measures are to a large degree in the area of physical protection.
In future years many terminals will operate from remote location managing a wide area network in a much more economical way. A logical access control system based on authentication by passwords grants access through those terminals.
For network security a common way is to use only private, fixed lines which terminate in operating centers, losing flexibility of the system. When the number of managed systems by the same operators increases this solution has to be abandoned.
Private networks can be an alternative: a private switched network may be dedicated to management operations and accessed only by operators. Another way is to simulate private networks, for example using the closed group facility.
If the public telephone network is used for maintenance purposes special network access control must be introduced like dial back system. Today a common approach is the use of an authentication system based on dynamic password or challenge-response handshake.
The type of value added services and the number of users are growing. Among these services a relevant impact on the security threats is expected by the decentralisation to users

of network operation functions.
A number of Administrations and manufactures are actively contributing to the standardization efforts currently undertaken by international bodies like CCITT and ETSI. In particular, significant amount of work has been carried out in the definition of standards for defining an Operation and Maintenance (O&M) network in order
- to centralize and to automatize O&M functionality (with remote control);
- to introduce new distributed services;
- to get a global vision for optimizing the network operations.
CCITT COM*IV.3* defined the so called Telecommunications Management Network (TMN) having the architectural configuration showed in figure. The TMN concept can be implemented using a combination of telecommunication equipment resources and dedicated support systems interconnected via a communication network.
TMN comprises principles, architecture, functions, data, communication services and protocols for the exchange and processing of management data and functions for telecommunication network.



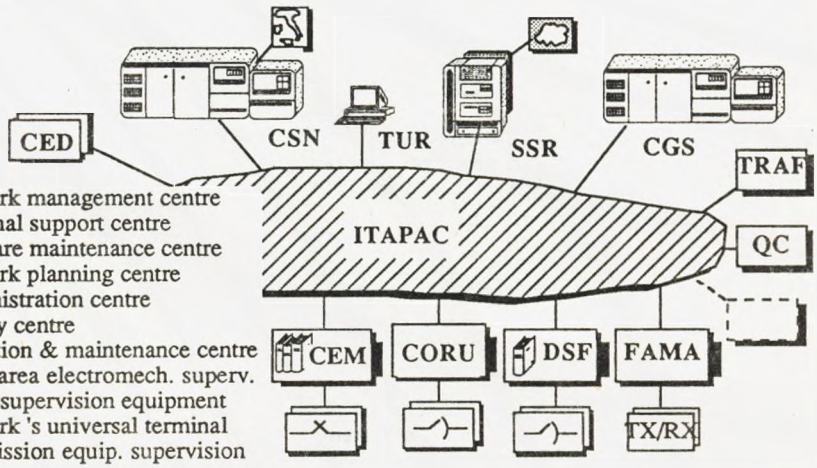
A generalized physical architecture for a Telecommunication Management Network (TMN)

The TMN architecture is based on the following sub-systems:
Network Elements (NE): the sub-systems performing telecommunication functions.
Operation Systems (OS): the subsystems on which are implemented either applications oriented to support O&M functionality, network planning, commercial procedure, etc. (A functions class); or basic functions oriented to the management of the communications and the interfaces with operator terminals, the information log, the protocol conversion, etc. (B functions class).
Mediation Devices (MD) are the media that allow NEs with poor "intelligence" to exchange information with OSs by implementing only class B functions.
Communication Network (DCN) : used to transfer management information between OSs and NEs or between OSs and OSs.
Work Stations (WS) : used by the operators to interact with NE, MD, and OS.
The introduction of the TMN creates a new security situation in the telecommunication area. New and powerful possibilities in network management are introduced: large portion of network can be diagnosed, reconfigured or put out of operation by simple commands through computer terminals in the management network. Such introduction must therefore

include the necessary security protective measures.
Security and data integrity are recognized as fundamental requirements for the definition of a TMN architecture, which in principle shall be *an open system with full connectivity*. Such open network introduces important issues concerning the control of security and integrity. The security issues comprising access control, authentication and audit logs and the integrity issues comprising semantic integrity and transaction integrity will become much more serious. TMN shall follow the principle of centralized security and integrity control. The inclusion of new functions or the modification of existing ones might lead to the definition of security mechanisms inserted at various level (OS,MD,etc.).

4. ITALIAN TELECOMMUNICATION MANAGEMENT NETWORK

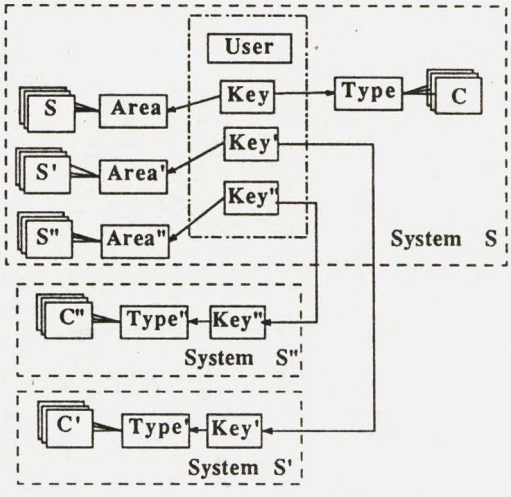
SIP has started a few years ago to centralize some of the operational functions in its telecommunication network.
The present situation is characterized by the presence of several support systems which, in general, have been developed by different organizational units within the company and, therefore, are dedicated and not yet cooperating in the framework of a TMN, being also the TMN concept and the decision to implement it more recent than the various support systems.
The figure below represents the global present situation as far as the various support systems and their location at different SIP organization territorial level are concerned.
The study on security requirements and mechanisms relative to Italian TMN is ongoing.



Italian Telecommunication Management Network

According to the reference architecture agreed, TMN stretches itself from telecommunication equipment to terminal equipped work stations. Its functions can be implemented over different territorial layers, from the local one, coincident with telecommunication systems, up to different centralized layers, based on specialized computer systems.
As far the access procedure is concerned, it seems advisable to recognize as a general philosophy that the network should allow each terminal to set up connections with each system to get reduction of the number of network terminals and reduction of global data flow and global memory.
The possibility of various users and terminals to set up sessions with several TMN systems, needs appropriate tools to assure functional and geographical competences and information privacy. These goals can be reached by associating to terminals and users an appropriate authorization related to the function set allowed and to the systems on which they operate.
The purpose of this procedure is to carry out a mechanism to transfer authorization information through the network and to utilize this information for session authority determination.

The data sets that can be used for authority determination in the TMN are:
Territorial area. A territorial area is a subset of TMN systems grouped by functional or organizational rules (e.g. all switching systems or switching systems of the same technology within a region). This concept can be used to split operation and maintenance competences between the operating personnel.
A user associated with a territorial area is allowed to perform dialogue sessions with the systems belonging to that area.
Functional type. A Functional type is a subset of commands of a TMN system grouped by a logic point of view (e. g. functional types for exchange maintenance, user administration, traffic administration, lines and trunk maintenance).
Within each system functional types are related to corresponding connection keys. When a user opens a dialogue session with a TMN system, a message containing the user connection key is sent to that system.
User authorization, on that system, is the functional type related to that user connection key.
User. A user is a subject allowed to use functions and data available on TMN systems.
The identification is by a password stored in one or more systems, according to the scope of their action field.
The user's privilege is granted, and verified, through a set of keys opening up a geographical area and the set of accessible functionalities. A given user's key may be shared among several systems. A logic diagram that shows the entities for user authorization is shown in figure below.



Operator Authorization within TMN

Terminal. A TMN terminal is a work station able to support the dialogue between users and systems of the network.
In order to improve the security degree of the access procedures inside the TMN, it should be possible to limit the function set to which such terminal can access.
Such a limitation can be implemented in the same way already described for users, e.g. by the association of terminals to territorial areas and functional types.
A terminal is defined by an identity stored only in one of the TMN systems. The set of commands allowed to a terminal over a generic system of the network is specified by the terminal connection key referred to the territorial area containing that system.
The set of commands allowed during a session of dialogue with a TMN system is called **session authority.** The entities defining session authority are:
- user territorial area;
- terminal territorial area;
- functional type related to user connection key;

- functional type related to terminal connection key;
- time of session set up.

The need for the last parameter can be justified taking into account the different operations performed over the TMN systems during the work-time and during the night.
Session authority can be obtained by an appropriate algorithm; it generally depends on Administration requirements for the management of the TMN.
Only for exemplification purposes, an approach to the determination of the session authority algorithm is described in the following:

- a session over a TMN system is allowed if and only if it belongs to both user territorial area and terminal territorial area.
- each interconnection key can be related to two functional types, the first valid in a time interval T1 and the second in a time interval T2, previously defined; the system will consider active the functional type corresponding to the current session time.
- session authority is defined performing a logic AND function between the operator functional type and the terminal functional type.

5. TRENDS AND EXPECTED EVOLUTIONS

The new developments and the use of digital techniques permit much greater flexibility in the provision of advanced services. According to the sharing of responsabilities between users and provider/operator we can think of many ways in providing a service. Whenever it is decided to defer the control over the manipulation of operation on network data, functions and resources to the end user, validation and control procedures should be embedded within the network to regulate the negotiations between end users and services providers/operating company. This solution, called **customer control** will be common to many new telecommunication services making these more flexible and user-made in their configuration, operation and maintenance.
In particular with the introduction of the Intelligent Network the access directly to Service Management System (SMS) will be available to service customers to modify their owns service parameters. Examples of services with particular needs of flexibility are Private Virtual Network, 800 service, Centrex service (provision of services normally available in PABXs by means of a special equipped exchange).
There are two user-terminal communicating modalities

- the customer uses and is guided by terminal messages and masks : the user may only make a few choices and set some predefined parameters;
- the customer uses a programming language, interactive or not.

6. INTERNATIONAL STANDARDIZATION EFFORT

Most of the technical approaches applicability depends on the achievement of an extended consensus between the various operating companies and telecommunication manufacturers since the network integrity and security can no longer be considered a "local" issue but requires the support of all interconnected parts and functions. Thus a key issue in this area is the establishment of international standards.
About the security approach there are two different philosophic orientation schools, those who think secure only completely secret systems and those who think it is impossible in practice to keep the secret of the adopted security instruments and want public systems which guarantee protection. From the second ones starts a standardization activity.
The adoption of standard techniques, in civil environment, favour *security* : a standard is certificated and, being public, submitted to continuous attacks generally divulged; *generality* : the development is independent of languages and systems; *cheapness* : a standard, with many providers, allows economical criteria and reduce the study, development and certification costs.
In the telecommunication area the achievement of an extended consensus between the various operating companies is essential to support all interconnected parts and functions.
The European Community finance some joint cooperative projects to support the

development of future telecommunication network.
The **Research and Development into Advanced Communications technologies for Europe** (RACE) project has the objective of leading to the development and implementation of advanced communication facilities across Europe before 1995, thanks to the introduction of Integrated Broadband Communications Network. The IBCN main goal is to provide all broadband services, current and future, via homogeneous digital communication networks serving all Europe.
Most advanced applications and many services rely on the cost effective provision of integrity mechanisms as authentication, proof of origin and receipt, data integrity, digital signature, access control, anonymity. These mechanisms have to be provided in a universal and unified manner to all users of IBC.
In the RACE program the security aspects are principally committed to three projects.
The project R1025 Functional Specifications of Security and Privacy in IBC looks at the overall IBC needs by providing the functional specification for the global provision of security. Target objectives are:
- the identification of suitable security services and the security management functionality necessary for the protection and separation of IBC and subscriber resources and processes;
- the specification of relevant security primitives and attributes necessary for the realization of IBC security;
- interface and protocol specification.

The necessary technology bases are provided by other two projects:
R1040 RIPE-RACE Integrity Primitives Evaluation has the goal to put forward a comprehensive ensemble of recommended cryptographic algorithms and suitable modes of use that meet the various security and performance requirements of IBC.
R1047 Techniques for Integrity Mechanisms in IBC concerns with making transmitted data secure and proving their integrity. Procedures for the execution of data exchange transactions must enable the user and the terminal in question to be identified, to test access authorization and to prevent the falsification of data.
A new European project is going to be planned to "enabling the flexible integration of telematic services under user control on a trans-national scale". The requirements for security of the Integrated Service Engineering (ISE) services offered by many different service providers will strongly differ between each other (users, services). As security policies and their enabling services can to a large extent be supported by a basic set of security techniques, it is desirable that security mechanisms and techniques, likely to be of general utility, are provided in a common way to many services, rather that each service providing its own, specific, different solution.
There is a willingness to identify the user requirements for security in the ISE environment and to define an ISE security framework, which is applicable to Open Systems and Open Services environment with user control over service creation and an open market in service provision. In a environment where end-users have a significant level of control over service creation and invocation, the system will have to provide "integrity" to ensure that users cannot call up service primitives or service combinations that may endanger the service layer or other parts of the IBC system or other users.
The standard scope covers a wide range of aspects of data security in both data communications and data processing: ECMA and ISO organization concern with general security standards, CCITT and ETSI concern with telecommunications.
ISO deals with standards for data criptographic techniques (JTC1/SC27); OSI Security Architecture, Framework and Model; Security Management; Upper Layer Security (JTC1/SC21); Banking (JTC1/TC68); Encryption of message handling (JTC1/SC28).
ECMA provides a Framework for the development of security provisions (TR/46) and proposes Data Elements and Security Services (ECMA XXX) for use in the Application Layer in the OSI Reference Model.
CCITT carried out some important standard related to security in communications.
X.400 Message Handling Systems deals with an electronic mail network, which due to its distributed nature could be susceptible to security threats like fraudulent identity claims, breach confidentiality and unauthorized manipulations.
The security services specified are provided on an optional basis and are derived from the OSI Security Architecture. The services are supported by use of security mechanisms, some

of them are remarkable to be exported like algorithm identifier, certificates, labels, tokens.
X.500 the Directory series contains some considerations which can be exported in a more general telecommunication context.
The Directory is a collection of open systems which co-operate to hold a logical database of information about a set of objects. The users of the Directory can, having permission and using specific services, read or modify the information stored by Directory collectively known as the Directory Information Base. Such access shall be in conformance with the security policy of the security domain to which the information is part of.
The Directory Security Model illustrates three security concepts:
security policy composed by definition of an authorization policy to specific access rights, enforce access rights, maintain access rights; definition of an authentication policy to verify identity of users, identify the origin of received information at an access point.
access control determines the access rights in terms of user identifiers or of capabilities.
authentication framework which specifies the form of credentials held by the Directory; describes how these credentials may be obtained from the Directory; states assumptions about how these credentials may be formed and placed in the Directory; defines ways in which applications can use such credentials in order to perform authentication.
Two level of authentication are defined, simple authentication by means of a password and strong authentication by means of cryptographically derived credentials.
The current authentication framework uses public key cryptosystems independent from public key algorithm.

7. CONCLUSIONS

In the telecommunication area the new developments and the use of digital techniques permit much greater flexibility in the provision of advanced services and in network management. Sensitive information is processed and carried over the telecommunication network and many business sectors are now dependent on the permanent availability, reliability and security of telecommunication services. These new aspects make necessary the introduction of the protective measures: security is becoming more and more crucial as telecommunication networks become more and more sophisticated.
Security and data integrity are recognized as fundamental requirements and must be considered as a real problem in the development process of telematic and information systems: the security should be planned and not considered as a solution of particular problems. Nowadays the protection of information seems to be just a argued problem and it is oriented to emergency and not to damages prevention.
Most of the technical approaches applicability depends on the achievement of an extended consensus between the various operating companies and telecommunication manufacturers. Network integrity and security can no longer be considered a "local" issue but requires the support of all interconnected parts and functions: key issue is the establishment of international standards.

BIBLIOGRAPHY

[] R1047 Deliverable 1. Determination of the IBC Integrity Needs and Research for Integrity Mechanisms. January 1989.
[] R1025 Deliverable 1. Basic Assumptions on Security Requirements and Architecture. November 1988.
[] R1025 Deliverable 2. Draft Functional Requirements Model - Security Service in IBC. April 1989.
[] R1025 Deliverable 3. Draft Requirements Model for IBC Management. September 1989.
[] ISO/DIS 7498-2 Information processing system. Open systems interconnection reference model - part 2: Security Architecture.
[] CCITT Rec. X.400 serie. Data Communication Network: Message Handling Systems. Vol.VIII, fas.VIII.7. Red Book. Geneva, 1985.
[] CCITT Rec. X.400 serie. Data Communication Network: Message Handling Systems. Vol.VIII, fas.VIII.7. Blue Book. Geneva, 1989.
[] CCITT Rec. X.500 serie. Data Communication Network: the Directory. Vol.VIII, fas.VIII.8. Blue Book. Geneva, 1989.
[] CCITT Recommendation M.30. Blue Book. Geneva, 1989.

SESSION 8

MAJOR CORPORATION IN NETWORKING

SNA Usability Enhancements for the 90's

Lee M. Rafalow

Architecture and Telecommunications, E96/673
Communications Systems
IBM Corporation
P.O. Box 12195
Research Triangle Park, NC 27709, USA

IBM's Advanced Peer-to-Peer Networking is designed to provide the networking foundation upon which production networks of the future can be built. It takes SNA concepts and improves usability in a world where networking is ubiquitous. Usability features discussed in this paper provide significant reductions in network definition. The concepts and models introduced in APPN position SNA to exploit emerging communications technologies in world-wide networks.

Overview

IBM's System Network Architecture (SNA) was designed to preserve and enhance customer investments in communications hardware and software by providing state-of-the-art, "industrial-strength" function with an eye on future requirements and on current inventories of equipment and programs [1]. This strategy continues today with the S/36 and AS/400 Advanced Peer-to-Peer Networking feature (APPN) [2,3] that builds on the strengths of SNA technology and preserves customer investments in application software.

The initial offerings of SNA, announced in 1974, provided support for the 3600 banking terminals in a single host environment. Those same 3600 banking terminals can be connected to the most current SNA networks. These modern networks are meshed networks using a variety of physical media to provide attachment to a wide variety of terminals, program-to-program communications, interconnection between networks (either via IBM Information Network or through direct connection between customers' SNA networks using SNI [4]), and a rich set of network management functions.

The complexity of a modern SNA network, with its "mix-and-match" set of products, has led to an effort to reduce that complexity. It is reduced partly by network aids such as NetDA [5] that help customers design their networks. It is also reduced by the introduction of a common set of functions and protocols to be implemented by all participating products rather than customizing the functions and protocols based on the application. Special requirements, then, are addressed as additional function above and beyond the base set of commonly implemented function. This "converged" set of functions for SNA network communications can be viewed from the users' perspective and from the network administrators' perspective.

Users' View

There are a number of users of a modern network. There are the terminal operators, both professional and non-professional, and the programmers, both application and system programmers. The ultimate objective of any network provider is to make the network itself disappear and give these users a single system image, a single distributed

operating system.[1] An important step toward providing this distributed operating system is to provide a common set of communications primitives. SNA does this with its Advanced Program-to-Program Communications (APPC), also known as Logical Unit type 6.2 (LU6.2) [6,7].

APPC implements a conversation model for communication. This is a general model in which remote subroutines and coroutines can be implemented taking maximum advantage of the processing capabilities of participating programs while minimizing network costs. Because of its generality, other communications models such as remote procedure calls can be built on top of the APPC primitives. The primitives provided include functions for opening and closing communications, security, sending and receiving data, synchronizing programs, synchronizing databases, and error handling.

These functions are provided such that the application programmer need not know the location of partner programs. The network directory resolves the name associated with the partner program to a system location. The programmer specifies the nature of the conversation via a mode name (e.g., interactive, batch) and the network route selection services determines the path most appropriate for that class of traffic.

Administrators' View

In order to satisfy the users' requirements, the network administrator must design and implement a network with adequate capacity and then manage the use of the network resources. In a production network the smooth operation of the network is central to achieving the organizational objectives. Thus the administrator is responsible for, among other things, network performance and the "fair" allocation of network resources, network security, availability, and problem management. To achieve the level of service demanded by users of production networks, network providers require significant improvements in the ease of installation, and the flexibility and configurability of the network.

Advanced Peer-to-Peer Networking [8,9,10] focuses on these usability aspects of the broader set of networking requirements. The thrust of the initial APPN effort was aimed at networks of smaller systems where the network administration is likely to be distributed among a set of the network users and the network maintenance is more likely to be performed by those same network users rather than a group of systems programmers. Decision-making in this environment is decentralized, changes may occur frequently, and the level of expertise may be relatively low. Thus, minimal system configuration and coordinated definition are key to the success of a network architecture for this environment.

The SNA Type 2.1 (T2.1) Node definition [11] has added a number of capabilities including the ability to negotiate some of the communications parameters rather than relying on coordinated system definition. The T2.1 design is based upon the earlier definition of the T2 node but extends that capability to permit peer-to-peer communications that fully support the capabilities of LU6.2 and do not require the presence of a host (e.g., S/370) system. The T2.1 node architecture is also known as SNA Low Entry Networking because it enables peer, point-to-point connectivity among low end SNA systems. When installed in VTAM and NCP, SNA Low Entry Networking permits applications in attaching T2.1 nodes to communicate with other T2.1 nodes through the SNA backbone network as well as increasing the capabilities of the attaching T2.1 nodes when communicating with host applications.

---

[1] The technology is not to the point yet where this can be done for all users in all networks, but efforts such as IBM's System Application Architecture are making strides in this direction.

To achieve the usability objectives, APPN enhancements extend this SNA T2.1 support with an integrated architecture for automatic network topology and directory support that dramatically simplifies network definition and permits dynamic selection of network routes.

An APPN network is composed of T2.1 nodes with enhancements that classify the nodes as either Network Nodes (NN) or End Nodes (EN). Network Nodes are systems that cooperate with other nodes in the network to provide network services (such as directory and route selection); End Nodes are application systems that do not provide network services to other systems but have access to all of the network facilities available to NNs. A Network Node may also be an application system that uses its own services as well as being a network service provider for other systems.

This classification scheme provides several advantages. First, the networking functions required for ENs form a subset of those required for NNs. Thus, the cost of development is less than a full NN implementation. The system resources required (memory, cycles, and nonvolatile storage) for networking functions in an EN are less than in NN, thus allowing the customer to have systems focussed on their application tasks. Also, the addition of NN function does not require a different set of base functions; products can implement NN functions as an add-on feature to the APPN End Node. Second, since the number of NNs in a network is significantly lower than the number of ENs, the distributed algorithms that are executed only among the NNs are much less subject to constraints due to network size: e.g., a network of 5000 nodes might have as few as 100 NNs--not a difficult number for even the most complex of the APPN algorithms. Finally, when additional networking functions are added, the interface defined between the EN and NN can be kept much more stable than the interface between NNs. Thus, EN implementations, and customer investments in those implementations, can be more easily protected.

Network Join Protocols

The network bootstraps itself during link activation beginning with automatic negotiation of link-level parameters such as the link station roles, the maximum size of messages sent and received, the maximum number of frames sent before acknowledgement, and whether segment generation and reassembly are supported on the link. All of this can be done without prior coordinated network configuration. A configuration screen for this step could be as simple as defining the name of the node, all other parameters can be hardware provided or negotiable default values.

In addition to the link-level parameters, this initial exchange also provides identification information to help bootstrap higher layer functions. For example, the exchange includes information that allows adjacent nodes to dynamically determine the name of the partner node, whether it is an EN or an NN, and whether network services are required between this pair of nodes.

In the event that network services are required between the two adjacent nodes, sessions are established between their Control Points (CPs). The Control Point is the component in the node that provides operational and management control functions for the other node components. The key CP functions discussed in this paper are the topology, directory, route selection and session services. APPN Control Points use LU6.2 formats and protocols to communicate with one another. Control Point sessions are established between adjacent NNs (with some exceptions) and between ENs and their serving NN. Where n is the number of NNs, d is the degree of network node connectivity--the average number of NNs to which an NN is connected--and e is the number of ENs, there are $O(dn+e)$ CP sessions. Typically, network nodes have a relatively low degree (e.g., 2-4) of network connectivity.

Once the CP sessions are established, the capabilities of the control points are exchanged and, if both nodes are NNs, topology database updates are exchanged. The topology database is a fully replicated database that is shared among all NNs in the network; it provides a map of the network nodes and their links to one another. That map is used both for route selection services and management services. The maintenance of that map requires broadcasting updates to all NNs.

In Figure 1 when NN2 joins the network, all NNs must be notified of the changes in the network map: i.e., the addition of NN2, the link connecting it to NN1 (1,2,a) and the link from NN2 to NN3 (2,3,b). The Topology Database Update (TDU) messages contain identifying information, node and link characteristics, and resource sequence numbers to identify the most recent updates for each of the resources described in the TDU. Each logical link (called a TG[2]) is treated in the TDUs as a pair of unidirectional TGs. Thus, each descriptor (known as a TG Vector or TGV) is identified by an ordered triple and it is updated by only one node. So, for example, the TG Vector from NN3 to NN2 is identified by the ordered triple (3,2,b) and is updated only by NN3; the reverse TGV is (2,3,b) and is updated only by NN2.

The Resource Sequence Number (RSN) for a particular resource (a TG or a node) is monotonically increasing. When an NN receives an update, it compares the RSN received with that already stored in its copy of the topology database. If the received RSN is greater than that currently stored, the new information is saved and the update is forwarded to all NN CP session partners; otherwise the received update is ignored. All connected NNs receive the update at least once but only propagate the information once.[3]

NNs that are not connected at the time of an update are given all updates they missed by their neighbors when they rejoin the network. So, in Figure 1, NN2 is given whatever updates it needs to complete its picture of the database. This is done by a pair-wise tracking of the updates sent and received: for an NN first joining the network, the entire topology database is provided, for an NN rejoining the network, the updates missed are provided.

Deleting resources from the topology database is by a simple aging algorithm that removes all inactive TGs and unreachable NNs after a month. Weekly updates are sent on all defined resources to prevent them from being deleted.

Thus NNs are able to maintain the network topology for use in route calculation and network management without administrative assistance.

Connection Networks

A shared access transport facility is a subnetwork in which attached nodes may send and receive data to a number of partners as if they were each attached point-to-point. The best example, and the one for which the Connection Network model was developed, is a local area network; other examples include satellite networks and public networks

---

[2] A Transmission Group (TG) is a set of one or more links between adjacent nodes that are treated as a single logical link for message routing. APPN currently supports only single-link TGs. The TG Number (b in the example) is used to distinguish between the TGs when more than one exists between two adjacent nodes. APPN does support these parallel TGs.

[3] In the event of errors or unstable network situations the network protects itself from inconsistent data and also guarantees termination of the broadcast algorithm.

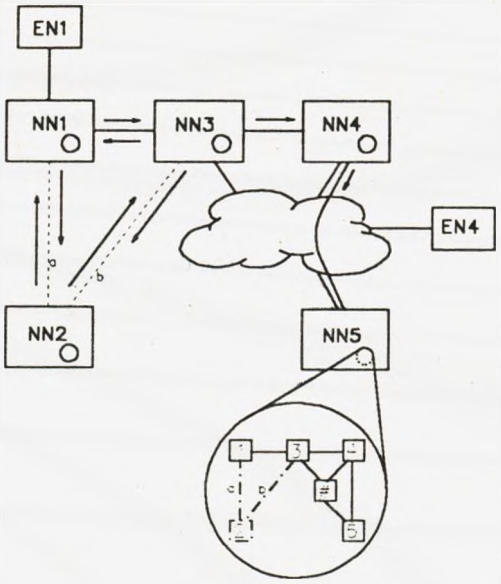


Figure 1. Topology Database Update

accessed using X.21, X.25, or ISDN. When such a subnetwork is used within an APPN network, it may be represented as a Connection Network.

A connection network is a virtual node that represents the subnetwork and a set of virtual TGs that represent the attachments to that subnetwork. In Figure 1, the cloud is a subnetwork represented by the connection network named "#" with four virtual TGs into the connection network. These TGs represent ports into the subnetwork. Routes through these subnetworks can be computed with significantly reduced network definition and a smaller topology database.

Connection networks require some coordinated definition: all nodes using the subnetwork must have the same name for it and a minimum spanning tree of the nodes attached via the subnetwork must be defined so that CP sessions can be established and control information exchanged. Each node on the subnetwork must know its own signalling information (e.g., its LAN adapter address) and, for any of the predefined TGs across the subnetwork in the minimum spanning tree, at least one node must know the signalling information for the partner (e.g., the LAN address for the partner) to be able to establish connections. In Figure 1, the TG connecting NN4 and NN5 is a predefined TG across the subnetwork that joins NN5 to the spanning tree.

This level of definition, however, is much simpler than would be necessary without the connection network model. NN3 has no definition for NN4 or NN5 attachments to the subnetwork, NN4 has none for NN3, etc. Where m is the number of nodes on a subnetwork, the definition required to provide full communications capability with the connection network model is $O(1)$ for each node, $O(m)$ for the entire subnetwork; the definition required without the model is $O(m)$ for each node, $O(m^2)$ for the entire subnetwork. Further reductions in system definition are possible with the addition of a LAN protocol for the discovery of server machines: e.g., query of a group address assigned to all NNs on a token ring. With such a protocol, ENs could discover their

NN servers and NN servers on the same LAN could discover one another without prior coordinated definition.

Topology updates for the virtual node and virtual TGs in both directions are made by the nodes attaching to the subnetwork. For TG vectors describing the connections *into* an attaching node, the TDUs contain the necessary signalling information as well as the other data contained in "normal" TG vectors.

Network Directory

The network must maintain a directory mapping names to locations in order to provide the required location transparency for users. The directory database can be significantly larger than the topology database. It contains names and locations for all the subsystems and terminals in all nodes, not just Network Nodes. Although the names maintained in the APPN directory today are limited, other names, such as user names, can be accommodated in the architecture design. Therefore, the directory database is a distributed, not a fully-replicated database. Each NN is responsible for the network directory for itself and its client End Nodes.

Resources may be added and deleted dynamically at any node in the network. When an EN joins the network, after exchanging CP Capabilities with its serving NN, the EN may register the resources it wishes to make available to the network. When a change is made, the client EN notifies its serving NN with either a Register or a Delete message. This limits the searches that the EN receives to just those for known resources. Alternatively, ENs may prefer to be searched rather than using the registration capability and thus maintain their own part of the distributed network directory; this option might be particularly useful for large application hosts.

The distribution of the directory information among NNs is performed on a demand basis. When a node in the network needs to locate a resource, it searches all NNs. The search is a broadcast, similar to the TDU broadcast, that is propagated to all NN CP partners. The speed of the search is related to the diameter of the network, not to the number of NNs. Each broadcast has a unique search identifier, the Fully-qualified Procedure Correlation Identifier (FQPCID). The algorithm dynamically constructs a search tree, each node in the tree keeping pointers to the parent and child nodes. When a currently unknown (new FQPCID) search is received, it is forwarded to neighboring NNs. When a reply, or a duplicate request, is received from a child in the tree, that child is removed. When all children have been removed, the parent node is notified with a reply. Of course, positive replies are forwarded up the tree immediately thus permitting use of the information as quickly as possible.[4] So, in Figure 2, NN1 removes NN3 from its list of children when it receives a second copy of the original search from NN3. The search tree produced has NN2 at the root with NN1 and NN3-NN4-NN5 as its subtrees.

Although this form of search is extremely fast, it is also costly in its consumption of network cycles and bandwidth. So, the root NN (NN2 in Figure 2) saves the information learned from the search. Future references by NN2 or its client nodes will not require another broadcast. Instead, a simple verification flow is sent directly to the NN last reporting the location. A number of further optimizations to the directory broadcast algorithm are possible. For example, the verification flow can be omitted and the cache entry can be assumed to be correct. A cache server node or set of cache server nodes can be placed in the network (identified by an indicator in the topology database)

[4] Similar to the TDU algorithm, the directory algorithm is complicated by the potential for duplicate FQPCIDs and unstable network situations.

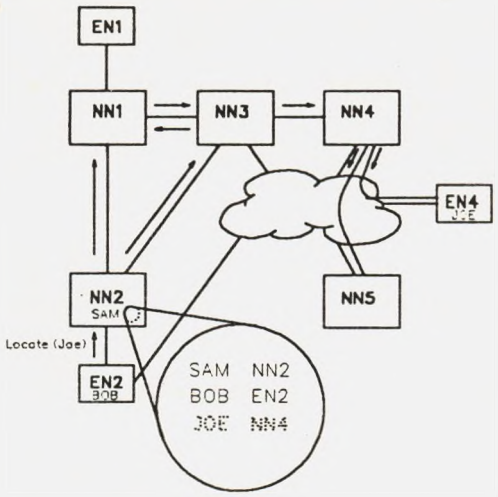


Figure 2. Directory Search

to eliminate broadcasts except when the cache server node does not have an entry for the requested resource name.

This automatic directory function supports an environment where resources are added, deleted, or moved frequently with minimal administrative intervention. Changes are applied locally and propagated through the network as required.

Session Initiation

When the network user requests a conversation, the network may need to initiate a new session.[5] In the most general case, the session partners are in two ENs served by different NNs.

As in Figure 2, the initiating node will send a Locate request to its serving NN, NN2 in the example. In this Locate request, EN2 includes TG Vectors to describe its local topology: its TG to NN2 and its virtual TG to the connection network. NN2 saves this information and conducts the search: a broadcast if there is no cache information or a directed, verification search if the cache data is present. In the reply, in addition to the directory information, EN4 returns its local topology data: its predefined TG from NN4 and its virtual TG from the connection network (with the signalling information).

Upon receipt of the reply, NN2 has accumulated a complete picture of the necessary parts of the network topology; it has the global NN topology database, it has the local topology for EN2, and it has the local topology for EN4.

To compute a route appropriate for the requested session requires knowledge of the nature of the session traffic as well as the characteristics of the links and nodes in the network. In SNA, the nature of the session traffic is embodied in a parameter specified by the requesting user, the mode name. In APPC, the mode name can be a reasonably

[5] SNA is a connection-oriented architecture in which connections (sessions) are established to maintain information for the management and control of data traffic.

user-friendly name for the required session characteristics; the architecturally-defined defaults are INTER, BATCH, INTERSC, and BATCHSC which specify interactive, batch, interactive-secure, and batch-secure, respectively.

The mode name--whether architecturally-defined, product-defined, or defined by the network administrator--resolves to a set of parameters for the session including message size (Maximum RU Size), flow control, session limits (the number of sessions to automatically activate and the maximum number of sessions), transmission priority (network, high, medium, or low), and a set of characteristics that are referred to collectively as class of service.

The class of service (COS) is derived from the mode name; for the architecturally-defined mode names, the COS Name is the same as the mode name. The COS Name indirectly specifies the relative significance of factors such as acceptable levels of security, cost/packet, cost/connect-time, propagation delay, and effective capacity. For example, the architecture-defined class of service for BATCH prefers routes with low cost/packet and large effective capacity; INTER prefers routes with low cost/connect-time and small propagation delays; the secure varieties of these COS Names require secure nodes and links along the path.

As one might expect, these are the kind of characteristics that are found in the TG and node characteristics in the topology database for NNs and their TGs and in the TG vectors provided by ENs at session initiation-time. Additional parameters reflected in the topology data include TG and node status (operational, quiescing, or inoperative), congestion indicators, and, to provide additional flexibility, there are user-definable fields.

In order to provide users with appropriate paths through the network, the algorithm used to select routes first weights TGs and nodes. The weighting algorithm computes a scalar value for each node and TG based on the relative significance of the characteristics for the request class of service. The current weighting algorithm uses an ordered table look-up to find the first row in the COS Name preference table that matches the TG or node characteristics and assigns the weight associated with that row to the resource. Again, default COS tables are defined by the architecture.

Once the weights are determined, Dijkstra's shortest path algorithm [12] is used to determine the least-weight path through the network for the requested class of service. Where two paths have identical weight, a random choice is made.

The selected route information is provided to the requesting node so a session can be established. When the selected route is through a connection network, as is the case in the example for the interactive request in Figure 3, the signalling information is provided as well.

APPN selects an appropriate route based upon the desired class of service and the current network configuration and characteristics.

The protocols to establish a session connection include the capability to negotiate session parameters such as the maximum send and receive message sizes. Where intermediate nodes are able to do so, mismatches in maximum frame sizes supported from link to link are addressed by automatically invoking segment generation and reassembly logic. Where this is not appropriate, end-to-end maximum send and receive message sizes are adjusted, again, automatically to allow for the minimum maximum frame size.

Flow control in APPN is done completely automatically. To prevent deadlocks in the network, the architecture specifies that all end-point and intermediate buffers are pre-

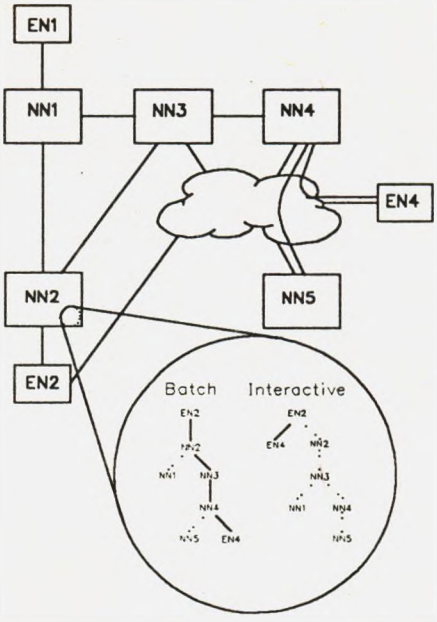


Figure 3. Route Selection Services

committed.[6] However, to make efficient use of these buffers, products must be able to dynamically adjust the number of committed buffers. During session setup, one send and one receive buffer is allocated at each affected node. Thereafter, the send and receive window sizes are adjusted via protocol to accommodate the traffic demand. The architecture permits products to ensure fair allocation of network bandwidth based upon offered load and parameters such as the transmission priority and the pacing window size values specified during session establishment.

Sessions with these self-adjusting characteristics provide efficient and fair utilization of network resources with minimal administrative support.

Application to Emerging Technologies

Two very significant networking phenomenon became quite clear as the last decade came to a close: the emergence of distributed operating system technologies in the very high speed local area network environment and the interconnection of independent networks. As the current decade unfolds, it is reasonable to expect the commercial exploitation of very high speed media in metropolitan-area and wide-area networks as well. The LAN operating systems that are emerging in the local area network environments will be strained to extend into these larger and larger networks. Further, the interconnection of independent networks, both SNA and non-SNA, will continue to progress toward a world-wide network of networks. The users' expectations in this environment

[6] Products are permitted to commit buffers by statistically managing a smaller, high speed buffer pool with sufficient other storage allocated to cover the entire commitment under heavy load conditions.

will be based on their reliance on these distributed systems. These networks will support our daily lives at home and at work.

The usability features provided in architectures like APPN will be necessary ingredients in any viable solution for these large, distributed operating systems. APPN network dynamics will need to be enhanced to allow for network growth without algorithmic complexity as an inhibitor. The availability improvements achieved in APPN by automatically adjusting to the changing network topology will need to be made completely nondisruptive. As high speed switched public facilities become more available, the connection network model will require further enhancements to optimize the use of these facilities without compromising administrative overhead.

Most importantly, all of this must be accomplished with the level of reliability required in a production network. Performance must be predictably good; that is, network planning tools must be available, network administration must be simple but fair in its allocation of resources, stress conditions must be handled gracefully, and problem management functions must be efficient.

References

[1] ***Systems Network Architecture-Concepts and Products***, IBM Corporation, SC30-3112 (available through IBM branch offices)

[2] R. A. Sultan, P. Kermani, G. A. Grover, T. P. Barzilai, and A. E. Baratz, "Implementing System/36 Advanced Peer-to-Peer Networking," ***IBM Systems Journal***, v.26 n.4 pp.429-452, 1987

[3] R. K. Harney and C. H. Jones, "Advanced Peer-to-Peer Networking," ***IBM Application System/400 Technology***, pp.50-58 IBM Corporation, SC21-9540 (available through IBM branch offices)

[4] ***SNA Format and Protocol Reference Manual: SNA Network Interconnect***, IBM Corporation, SC30-3339 (available through IBM branch offices)

[5] ***Network Design and Analysis User's Guide (VM)***, IBM Corporation, SC34-4061 (available through IBM branch offices)

[6] J. P. Gray, P. Homan, P. J. Hansen, M. A. Lerner, and M. Pozefsky, "Advanced Program-to-Program Communication in SNA," ***IBM Systems Journal***, v.22 n.4 pp.298-318, 1983

[7] ***SNA LU6.2 Reference: Peer Protocols***, IBM Corporation, SC31-6808 (available through IBM branch offices)

[8] A. E. Baratz, J. P. Gray, P. E. Green, J. M. Jaffe, and D. P. Pozefsky, "SNA Networks of Small Systems," ***IEEE Journal on Selected Areas in Communications***, pp.416-426, May 1985

[9] P. E. Green, R. J. Chappuis, J. D. Fisher, P. S. Frosch, and C. E. Wood, "A Perspective on Advanced-Peer-to-Peer Networking," ***IBM Systems Journal***, v.26 n.4 pp.414-428, 1987

[10] D. P. Pozefsky, D. A. Pitt, and J. P. Gray, "IBM's SNA Architecture," ***Computer Network Architectures and Protocols***, 2nd edition, C. A. Sunshine, ed., pp.449-509, Plenum Press, NY, 1989

[11] ***SNA Type 2.1 Reference***, IBM Corporation, SC30-3422 (available through IBM branch offices)

[12] E. W. Dijkstra, "A Note on Two Problems in Connexion with Graphs," ***Numerische Mathematik***, 1, pp.269-271, 1959

FOUNDATION FOR FUTURE NETWORKING

David A. Twyver

Vice President
Product Line Management
Northern Telecom World Trade

Northern Telecom was a pioneer in the standardisation of X.25 and the commercialisation of packet switching technology in the mid 1970's. Fifteen years later, Northern Telecom is the leading supplier of packet networks and has a new packet switching product line that is capable of serving the networks of the next decade.

Worldwide, data communications is one of the fastest growing sectors of the telecommunications business. Just a few years ago, virtually 100 percent of infrastructure spending was accounted for by voice services. But today this has dropped to 75 percent, with data communications taking a quarter of the total spending. By the turn of the century, the data portion is expected to rise to at least one-third of total telecommunications spending.

Within data communications, packet switching, using X.25, has become the world standard for both public and private data networking, and is fully defined by the international telecommunications authorities. Northern Telecom was among the originators of this X.25 standard, and has maintained a continuous involvement in international standards activities. To date, Northern Telecom's total R&D investment in packet switching is well over 100 million dollars. It includes three generations of hardware designs and three million lines of software code for systems and tools.

The benefits of packet switching systems are being enjoyed by an increasing number of organisations around the world. The resource sharing techniques of packet switching ensure that cost effective systems can be developed, offering real savings to customers. High availability, high capacity systems, like the Northern Telecom DPN-100, enable organisations to support critical applications across a wide area, thus delivering a full range of services to all their customers. The open environment of a packet network allows the user a wide range of choice in computer and terminal systems.

Powerful switching capabilities, combined with this open architecture, enable multiple applications to be supported on one common network, thus rationalising an organisation's communications strategy. Through utilisation of a packet communications system within a well developed evolutionary strategy, an organisation can rationalise and protect its communications investment, which is becoming a significant element of its budget.

Northern Telecom has been a leader in the field of packet switching since the introduction of the first services in the 1970's. Since those early days, the market has experienced a growth unforeseen by all but the most visionary of our forecasting specialists. Current predictions are that this growth will continue well into the next decade, and network operators across the world confirm this ongoing trend.

The packet switching market has been growing consistently at about 40 per cent per year. This has caused manufacturers to constantly upgrade their products to meet the ever increasing demands of the marketplace for high performance, cost effective systems. In Northern Telecom we place an additional challenge on our designers, as every new product development must be fully compatible with our current products and network management systems, such that our users can protect their investment in our technology. The DPN-100 product line is our most recent introduction.

The new product is fully compatible with all of the Northern Telecom packet switching products introduced since the mid 1970's, while taking full advantage of the latest technology available today. It is suitable for large and small networks in both the private and public sectors of the marketplace.

Northern Telecom developed an experimental packet switching system in the early 1970's and introduced its first commercial packet product, called SL-10, in 1976. This product, which was based on the technology of the highly successful SL-1 PABX, went into service in the Telecom Canada Datapac network in 1977.

The first European public sector network was provided to the Deutsche Bundespost in 1978. In the private sector, we provided a network in 1979 to Generale de Banque in Belgium. Thus our world wide penetration of the packet switching market began right here in Europe, and Europe has continued to play a dominant role in the success of the product line.

Our major customers in the public sector include ten administrations in Europe. This makes us the largest supplier to the European PTT market. We are also the leading supplier to the North American public sector market, with sales to six of the Bell Operating Companies. And in Asia, Northern is supplying the new packet networks to Telecom Australia, and NTT in Japan.

Northern Telecom has been equally successful in the private network sector, having achieved a significant penetration of the financial community with our packet switching products. A decade of successes including the Generale de Banque in Belgium, the Federal Reserve Board in the US, Barclays Bank in the UK, SPADAB in Sweden and Den Norske Creditbank in Norway, led to our recent award of the global SWIFT network, with nodes in over 70 countries.

In all, Northern Telecom has fifty networks in service around the world, with over 150,000 ports on 1,500 nodes.

All these customers, and the trends of the marketplace, have defined a number of requirements for the product to take us forward into the next decade. The key requirement is for a high capacity system to meet that 40 percent annual growth demand. We foresee an increasing demand for capacity, both in the size of individual switches and the overall size of the networks. As the market matures, users are expecting improved levels of reliability and availability, especially as their business operations become more dependant on the packet switching network. Enhanced access for Local Area Networks and other high speed systems is required. Additionally, further expansion and evolution of the networks must provide full compatibility with existing systems, while providing a path to the ISDN world of the 1990's. Below are examples of two network operators who are experiencing this increasing demand for service in their marketplace. This provides an excellent illustration of how success has led to increasing demand for network capacity.

The Datapac network in Canada, established in 1977, demonstrated slow growth initially as the market developed an awareness of the benefits of packet switching. Since 1980, however, the network has grown tremendously and expanded at over forty percent per year for most of the last decade. The situation on the Datex-P network in Germany is very similar. This network, which has been in service for over ten years, is also growing at a rate of forty per cent per year. Both Datapac and Datex-P currently have approximately 50,000 ports in their networks. This characteristic of slow start up followed by a sustained period of high growth is being experienced by most public network operators in Europe.

Today, the initial start up period is much shorter, due to the general market awareness of the benefits of packet switching. A network planner, faced with such a growth profile, has only one important question: will this high market demand for packet switching continue.

The answer from around the packet switching world, is that market demand for packet switching services is forecasted to continue to maintain the current high growth rates until well into the middle of the decade. Faced with a 40 percent annual expansion profile, Northern Telecom realised that a minor adjustment in product capabilities would not offer a long term solution to our customers. A quantum leap in product capacity was required.

The SL-10 packet switch introduced in the mid 1970's could support about 100 packets per second, which seemed a high number at the time. By 1979 SL-10 capacity had grown to about 1,000 packets per second, and this evolution continued with 3,000 packets per second being achieved in 1985 on the basic DPN switch. The current product line has ten times that capacity.

DPN-100 extends the top of the single switch range to over 30,000 packets per second, with the capacity to handle 30,000 lines. Equally significant, it is also cost effective for a single switch of just eight lines with a capacity of 100 packets per second. As will be seen from the description of the technology, growth from these small systems to the largest is accomplished with the minimum of hardware changes, in a smooth transition with no discontinuities. The new network switches can be fully integrated into existing networks, using the same software base and, more importantly, the same network management system.

To understand the DPN-100 architecture, let's look first of all at the original product design. Northern Telecom packet switching products have an architecture based on functional layers, Figure 1, which allows the development of one layer independently from the other. The network layer is connected to other switches in the network across the datagram subnetwork which, unlike a virtual circuit subnet, provides a very efficient and resilient connection-less media. The access layer is used to terminate all user access lines, and identical systems are used for both remote and local access. The access layer implements virtual circuits, which operate over the datagram subnetwork.

The DPN-100 system architecture has envolved from the basic DPN architecture, which consists of two major building blocks: the access module and the network module. The access module terminates user access lines and supports such access protocols and X.25.

The network module provides overall switch and network functions, such as packet routing, switch management, and control. In addition, during call setup, the network module, which is linked by trunks to network modules in other packet switches, supplies call-routing functions for all attached access modules.

What differentiates the DPN-100 architecture from the basic DPN architecture is a new switch core, which performs the functions of the network module. This core consists of three new modules - a resource module, a trunk module, and an optional capacity module, called the DMS-Bus and derived from the successful DMS PSTN Exchange. These core modules, together with the recently enhanced access module, give the DPN-100 the flexibility to economically satisfy a wide range of capacity requirements.

For example, the smallest configurations of a DPN-100 switch, Figure 2, without the optional DMS-Bus, support a total switch throughput of 3,000 data packets per second (DPPS) and have a capacity of more than 3,000 access lines. (The smallest configurations include the DPN-100/20, with 32 to 176 access lines; the DPN-100/40, with 200 to 800 lines; and the DPN-100/50 with 300 to 3,000 lines). The largest configuration, Figure 3, called the DPN-100/500, includes the DMS-Bus and multiple resource modules. It supports 2,000 to 30,000 access lines and offers a throughput in excess of 30,000 DPPS.

Regardless of switch size, every configuration of the DPN-100 is guaranteed identical access services because all configurations use the same access module.

Access module

The DPN-100 access module is a distributed processing system that consists of numerous processing elements (or computing engines), a duplicated common bus, and two independent common memories. The common buses interconnect the processing elements, allowing them to communicate with each other by sending messages to the common memories.

Every processing element in the DPN-100 access module is a single printed circuit pack with an on-board Intel 80286 microprocessor and one megabyte of memory. Each processing element may have associated with it one or more peripheral interface cards, which terminate access lines, network links, and local disks. A single processing element can control as many as 32 access lines.

Processing elements are reserved for common access module functions, such as call routing, administration, and control; they also terminate loadshared network links. Among these processing elements, one is designated the access module master and another the back-up for common functions. Each of the remaining processing elements can be assigned to support any one of the many available access protocols, allowing a flexible mix of service offerings on the access module. These protocols include:

- the 1980 and 1984 versions of X.25, which include Link Access Protocol-Balanced (LAPB), as well as LAPD, the ISDN (Integrated Services Digital Network) version of LAPB;

- the CCITT X.3, X.28, and X.29 asynchronous protocols, which collectively define how data is transported to and from asynchronous terminals across a packet-switched network. Through the built-in X.3 packet assembler/disassembler (PAD) in the DPN-100, asynchronous terminals not designed for packet switching can transmit and receive data on packet networks. Where the PAD is used, X.28 and X.29 standards define the protocol for the terminal and the host computer, respectively;

- the display station protocol (DSP), devised by Telecom Canada, Telenet, and Tymnet for the IBM 3270 terminal PAD and host PAD;

- the IBM synchronous data link control (SDLC) terminal PAD protocol.

- the IBM System Network Architecture (SNA) Multihost terminal PAD protocol. This service allows SNA terminals connecting through one controller to independently access different SNA host computers. With DPN SNA Multihost, a call placed from a terminal establishes a virtual circuit to a particular destination host computer; and

- the asynchronous polled interface protocol for various point-of-sale terminals.

In addition to supporting a variety of access protocols, each access module contains duplicated hard disks for local storage of software and service parameters, updates of which are downloaded from the DPN-100 network administration system. New service data files can be loaded into a processing element at a prescribed time without any service disruption.

Apart from the access module, the other key building block in the DPN-100 is the new switch core, which comprises the resource module and, for the largest DPN-100 configuration, the DMS-Bus and trunk module.

Resource module

In addition to performing core processing functions, the resource module, which is common to all DPN-100 configurations, provides essential resources for all access modules connected to it. These resources include source and destination call routers, switch management and control functions, and access to servers that provide such network features as call-redirection and the Network User Identification (NUI). (For billing purposes, the NUI identifies a caller to the network. The NUI is equivalent to the credit card number used in voice networks).

In large DPN-100 configurations, at least two resource modules are dedicated to tasks of switch management, network operations, and server functions.

One resource module is designated the prime for those switch functions where there can be only one active instance of the process, such as switch control. For these unitary processes, backing up the prime resource module in standby mode is a designated alternative resource module. All other processes can be performed by any resource module and are dynamically loadshared, first across all resource modules and then, if there is more than one instance of the process, within a resource module. The switch management and server functions are the same in all sizes of the DPN-100 and are compatible with existing networks.

The internal architecture of the resource module is similar to that of the access module, although the common memories and the dual bus in the resource module have been enhanced to provide the additional packet throughput needed in the switch core. The resource module employs the same Intel 80286-based processing elements as the access module. Each processing element is assigned one of the three basic processor functions: switch master, trunk or network link controller, or call server.

Switch master: While a resource module may have many trunk or link controllers and call servers, only one processing element is requried to control a resource module. Another processing element, however, is designated as a backup. Should the master processing element encounter difficulty, the backup assumes control of the resource module. Recovery by the backup is fullty automatic and proceeds without clearing calls in progress.

The master processing element serves as the resource module's centralized control point, ensuring efficiency and consistency of operation in a multiprocessor environment. For example, the master processing element is responsibe for loading the system and applications software.

As well, it creates the switch operating environment, an activity that includes initializing routing tables, establishing various subprocesses, and loading and activating service parameters. The master processing element also monitors the switch's health by detecting and reporting failures; initiates recovery; and manages the resource module's disks and life system.

As well, the master processing element coordinates the activity of many distributed processes in other processing elements or modules. For example, to coordinate the pack-routing process, the master dynamically determines the optimal path to all other switches in the network by exchanging updated routing information with these neighboring switches. This information is stored in common memory tables for use by other processing elements. Centralizing the optimal route paths in common memory tables ensures a consistent switch view. Trunk controllers within a resource module have instantaneous access to these optimal routing tables and can transmit packets without involving the master processing element. This autonomous operation of individual processors in the key to high switch throughput.

Trunk controller: Using patented DPN algorithms, trunk controller processing elements handle all interswitch packet communication. These algorithms are designed for high-speed packet transport over both terrestrial and satellite trunks. The trunk controller processing element can support either two trunks, each with a maximum transmission capacity of 128 kilobits per second (kbit/s), or one trunk with a transmission capacity of 192 kbit/s. A high-speed version of the trunk controller processing element is used for trunks running at T1 speeds (1.544 megabits per second (Mbit/s) in North America and 2.048 Mbit/s in Europe).

As packets flow through the network, trunk controller processing at each step is minimized by the use of the connectionless subnet and an associated hierarchical routing strategy. Because the subnet is connectionless, intermediate tandem nodes in the network are not required to reserve resources for a particular call. By routing packets along optimal paths, the number of switching points is minimized, resulting in low network transit delay and no unnecessary processing for tandem traffic.

The DPN-100 switch architecture supports the flexible configuration of many trunk controllers. Where traffic loads require, several trunks can directly connect two switches and can dynamically loadshare the traffic over the trunk group. The trunk controllers are responsible for maintaining packet priority on the links and, when necessary, for invoking congestion control.

They also continually monitor the quality of their links and periodically report link and throughput statistics to the network control system. Any trunk or network link can be tested at several loop points and can be removed from service without affecting customers.

Network link controller: Network link controllers perform a function similar to that of trunk controllers, but rather than interconnecting switches they interconnect access modules to resource modules. These processing elements are architecturally similar to the trunk controllers and support the same high level of link performance.

Call server: For those access modules connected directly to a resource module, and optionally for other switches in the network, call-server processing elements support various functions related to call establishmenrt. Each server contains a database and is located in the resourcer module, principally to allow effective management and administration of these databases.

The primary functions of the call server are source and destination call routing, network user identification, and call redirection. Because the real-time performance requirements for a server function can be high, the switch architecture allows replication and dynamic load-sharing of servers and their databases. The number of servers in a particular resource module can be engineered as required.

DMS-Bus

Unlike the resource module, which is common to the switch core of both large and small DPN-100 configurations, the DMS-Bus is unique to the large configuration. The addition of the DMS-Bus, plus some distribution of process communication and switch control, represents the major differences between the large and small DPN-100 configurations.

The DMS-Bus consists of two DMS-Bus planes, which operate in a loadshared redundant mode. Each plane has the capacity to support the full switch throughput of 30,000 data packets per second and therefore can handle the total traffic throughput in the event that one plane should fail. (DMS-Bus is also used in DMS SuperNode - a major architectural enhancement to the DMS-100 Family of digital PSTN switches).

All packet traffic flowing between the resource modules, access modules, and trunk modules within the switch passes through the DMS-Bus.

These modules are connected to both DMS-Bus planes by 2.56-Mbit/s links. Messages are not stored in the DMS-Bus, but are forwarded directly from the inbound 2.56-Mbit/s link to a specified outbound link.

Message routing within the DMS-Bus is determined by a header at the front of every message. This header specifies either a physical link or a logical destination name, which is mapped onto a physical link by DMS-Bus. This mapping capability allows each module to route messages by destination module name, process name (for example, the source call router), or destination switch name.

The flexible naming in the header is used by the message sender to implement loadsharing among destination modules. Loadsharing among modules is implemented locally by each message sender. Each module can route messages independently; only the DMS-Bus mapping function requires updating to reflect the current module or process availability.

This mapping function also enables the switch to be reconfigured quickly to respond to failures, recovery, and the addition of more processing elements and modules.

Trunk modules also distinguish the large DPN-100 configuration from the smaller one. These modules contain trunk controllers that operate at speeds up to 2 Mbit/s to interconnect a large DPN-100 switch to other switches in the network.

The internal architecture of the trunk module, similar to the resource module, consists of trunk controller processors, DMS-Bus link controllers, and two master processing elements. The master processing elements handle software loading, trunk module control, and routing tables maintenance. Low-speed trunks up to 192 kbit/s may still be configured in resource modules as well as in trunk modules.

DPN-100 network

In order for trunk modules to effectively connect large DPN-100 switches to other switches in the network, a hierarchical approach to packet routing is necessary. Hierarchical routing is achieved in the DPN-100 layering the network architecture into three logical largers: packets are first routed over trunks to a switch; within the switch they are forwarded to a local or remote access module; and finally, within the access module they are routed to the destination software entity that controls an access line.

Within this architectural hierarchy, a single large DPN-100 switch can support more than 200 access modules, which can be flexibly connected to form a subtopology around the switch core.

An access module can connect to other access modules above it in the hierarchy and, where geographic redundancy is important, the access module can also be connected in a dual-homed manner to two DPN-100 switches. The high traffic load generated by large DPN-100 switches is carried on trunks at rates up to 2 Mbit/s speeds to other switch sites at the top level of the routing hierarchy.

The DPN-100 architecture offers three major benefits to private and public packet-switching networks. One, the modularity of the architecture permits incremental, nondisruptive growth through the addition of modules and processing elements. Two, dynamic loadsharing across all applicable switch elements simplifies engineering. And three, such architectural attributes as replicated common elements, distributed processing with modular software, dynamic loadsharing, dual-homed modules, and data switching on all hardware modules ensure high levels of availability and reliability.

Perhaps most significantly, however, the DPN-100 architecture represents a natural evolution of the Northern Telecom SL-10 and DPN packet-switching family. It provides an order-of-magnitude increase in capacity over previous switches and represents a foundation for satisfying the future growth requirements of even the largest packet networks.

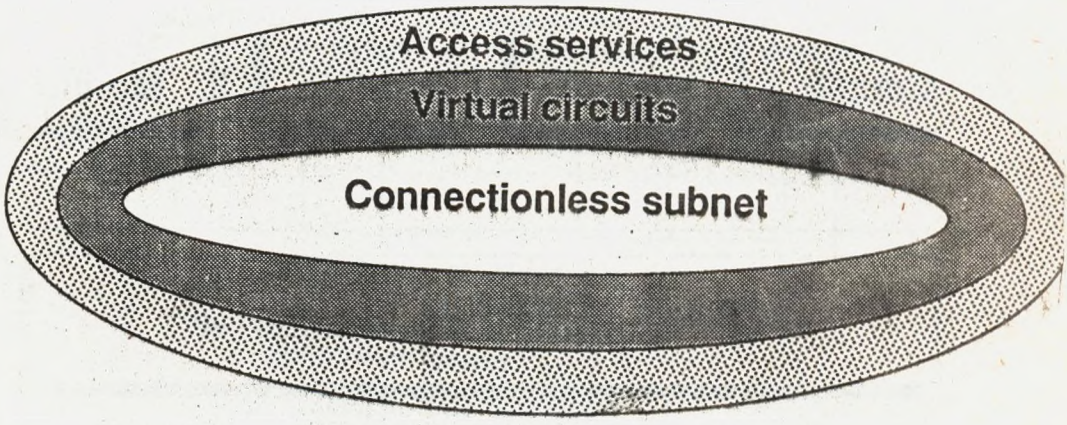


Figure 1 DPN Network Layers

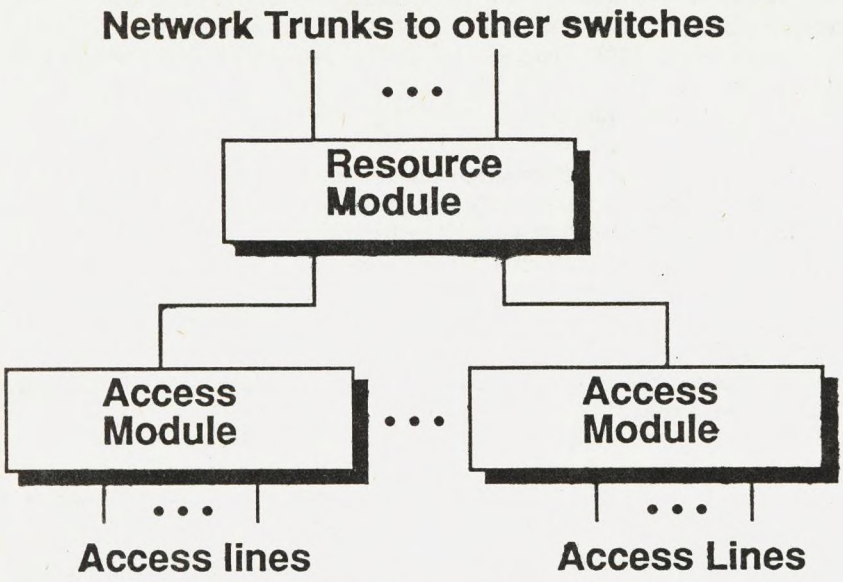


Figure 2 Small DPN - 100 Configuration

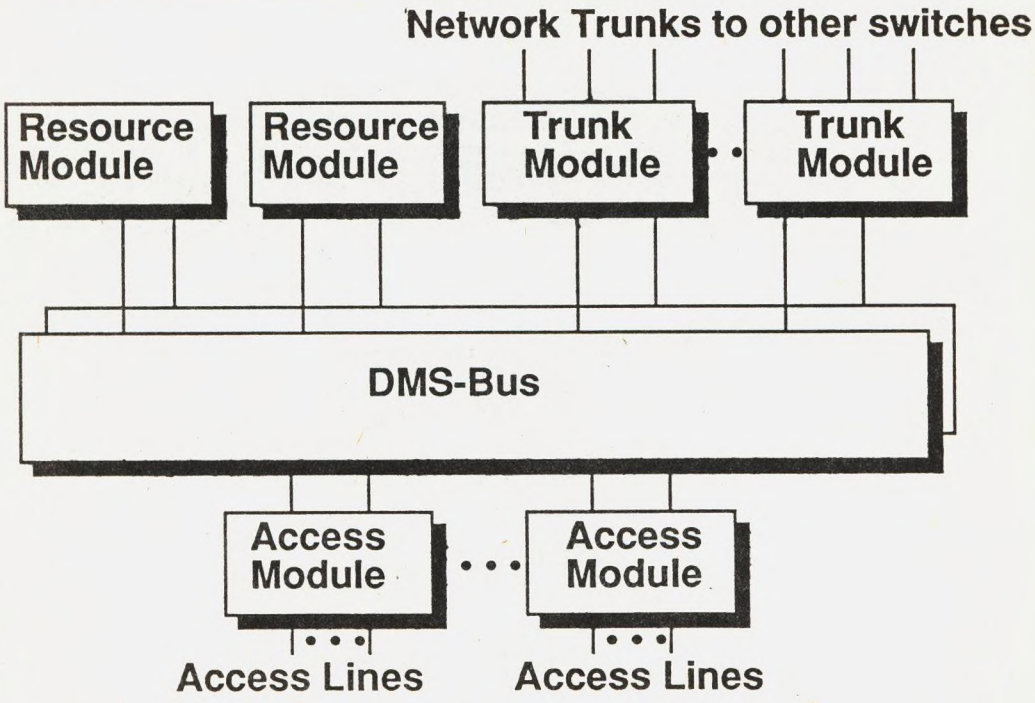


Figure 3 Large DPN - 100 Configuration

FIFTH GENERATION PRIVATE WIDE AREA NETWORKING

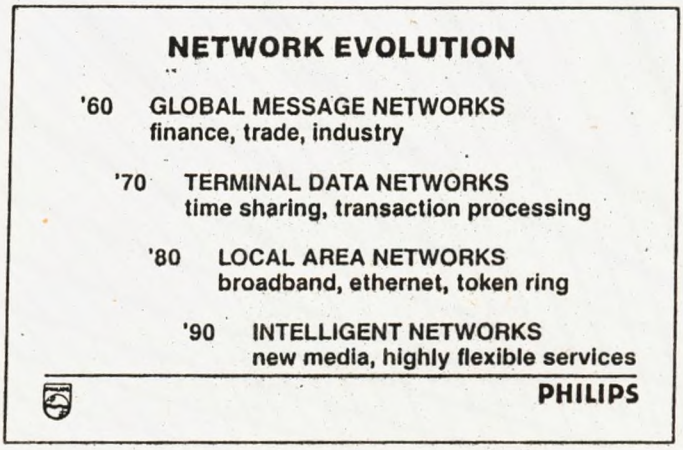
Ir. V.H.C.M. Evers

Philips Communications & Processing Services

A global communications infrastructure for data communications services is about to emerge. A number of new technologies are competing to attract user demand. Philips Communications & Processing Services (C&P) is well positioned to take advantage of these new opportunities on behalf of its customers. Some typical examples of network applications using various modern technologies illustrate this point. It is said that ways have to be explored to allow cooperative exploitation of new information technology options.

1. INTRODUCTION

There is a noticable acceleration of developments in wide area private networking. In the sixties several private global message switching networks were the precursors of the private data networks with integrated electronic mail capabilities of today. The seventies saw the emergence of public data networks in the midst of a debate among proponents of circuit and packet switching. Now we know packet switching is going to stay, but perhaps not in quite as dominant position as some might have guessed.



In the eighties two major standardization efforts were launched successfully and subsequently influence the introduction of new facilities and services.

Local area networks not only offered a standard way of cabling, but also pioneered high-speed virtual connections in-house. Today they represent the fastest growing segment in the data communications equipment market. The motor for this growth is undoubtedly the personal computer.

The public telephone network finally is undergoing a redefinition with the introduction of the ISDN standard. Here the driving force is much more the power of intelligent networking solutions combined with enhanced telephone function features.

Whether public ISDN will be attractive to private organizations for the exchange of data is debatable. The availability of new facilities and media will influence the usage patterns and the requirements for business data network services at the same time.

With so much change going on, only a balanced view of what may constitute the elements of fifth generation private data networking is appropriate. The infinite potential of data broadcasting, satellite communications and high-speed optical fibre links has to be recognized. However, in all this horizon gazing we should still find time to link the potential of new developments back to the day-to-day business communication requirements. If only to find the economic justification for these developments.

In this paper experiences with data communication within the world-wide Philips organization are taken as a starting point for discussing the potential of fifth generation networking.

2. THE GLOBAL COMMUNICATIONS INFRASTRUCTURE

First of all there is no such thing as a global infrastructure. The world of communications consists of highly heterogeneous national infrastructures with some forms of regional cooperation.

Despite this situation there has been a marked increase in the interconnectivity between national operators and a constant improvement in quality of service of the telephone network over the last few decades.

Unfortunately, less energy has been devoted to the more specific business communications services such as telex, packet data, and videotex, while room has been created for facsimile which now competes with all the more integrated forms of message communications including computer based electronic mail.

A whole new area of data communications is being opened up by satellite broadcasting in various forms. The consumer will very soon have selective access to data sent via the new D2MAC digital broadcast standard for direct reception by TV viewers in large areas of Europe and the United States.

Intelligent schemes for multiplexing upstream data from small antenna

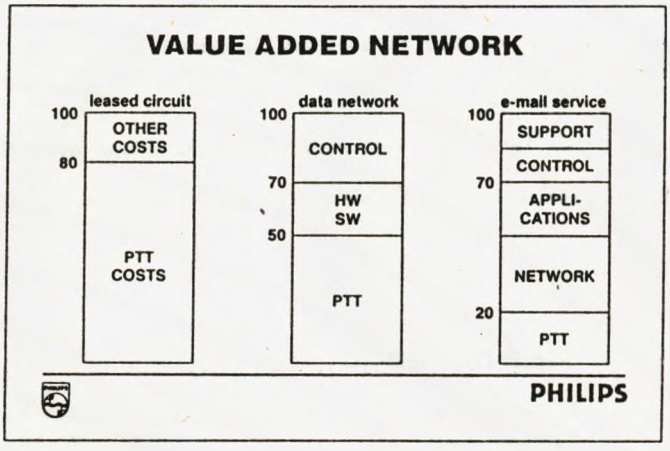
dishes at business premises allow two-way data communications to be established in a very cost-effective way. For this reason the Government of India has adopted this so-called VSAT (Very Small Aperture Antenna Dish) technology for linking-up remote computer systems to central offices in New Delhi.

Although largely oriented towards application of dedicated leased circuits, data network suppliers are gradually opening-up their network architectures to allow exploitation of a wider range of basic telecommunication transport facilities such as:

- switched connection via the public telephone network
- switched digital services as offered by ISDN in the public area and by the PABX in private telephone networks
- high speed fixed services (multiple of 56 and 64 Kbps)

Generally speaking these data networks are promoted exclusively for the needs of private enterprise and are provided on a world-wide basis. This simplifies uniform implementation in a multinational corporation such as Philips.

This contrasts greatly with the low level of service still experienced in international packet switching destined for a much wider range of users, both small business and private.



These imbalances in the market for data communications equipment and services create room for deployment of global network services, especially now that end-user demand is expected to create high traffic volume growth in the nineties.

Not only have computer manufacturers entered this market (e.g. IBM's IIS network) but telecommunication agencies are also reaching out to reinforce their position (BT's take-over of Tymnet; ATT's acquisition of ISTEL, joint PTT ownership of Infonet).

3. USER REQUIREMENTS FOR DATACOMMUNICATION

Data communication supports very diverse forms of applications such as file transfer, database access, electronic mail and electronic data interchange. These all differ in basic requirements for throughput time, connectivity, response time, interoperability and availability.

End-users may also want to use very different methods of local connections, have access to remote processing equipment and need various international communications facilities.

However within Philips we still aim to support clearly defined service levels for data communications. These are defined in functional terms, quite independently of a specific networking solution and its performance and availability.

The following uniform set of user requirements applies in virtually all situations:

I Simple user command interface

II Straight forward and uniform definition of network addresses, application and end-user names; a minimum of address conversions

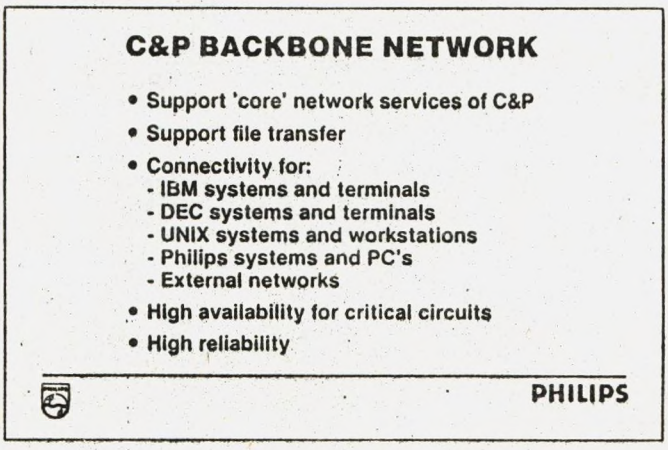
III Transparent data exchange; minimum of protocol/format conversions

IV Support both by instruction, a 24-hour "helpdesk", built-in features for user status tracking and, in the case of electronic mail, an "address book" facility

V Guaranteed service levels in terms of performance and availability

VI General and optional security facilities providing basic privacy and protection against data manipulation

VII Methods and means for performance measurement and cost accounting

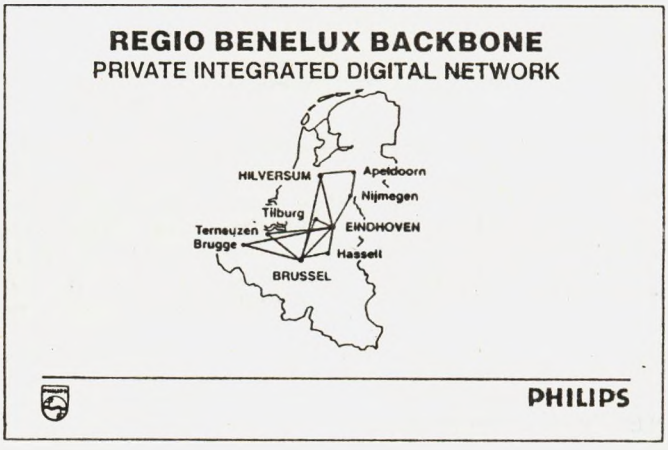


4. THE PHILIPS DATACOMMUNICATIONS INFRASTRUCTURE

Philips operates its internal communications and processing facilities, including postal mail and private telephony systems in eight internal service regions world-wide.

The Philips Communications & Processing Services organization has its regional headquarters in:

- Eindhoven, The Netherlands
- Hamburg, Germany
- Suresnes, France
- Mitcham, UK
- Milano, Italy
- Taipeh, Taiwan
- New York, USA
- Sao Paulo, Brazil



Within each region highly different networking approaches exist, mostly geared towards cost-effectiveness at the national level. This favours the use of public packet switching in France, Canada and Italy, sharing of voice and data circuits in the USA and separate leased data circuits in a number of other countries.

Four basic network architectures are available for inter-regional use:

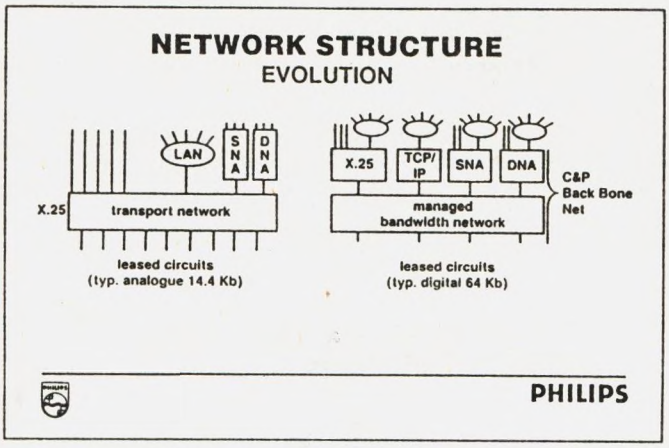
- SNA to connect the central mainframe sites for business data processing via MSNF/SDLC in a highly efficient way both for interactive and batch traffic.

- DNA to connect distributed minicomputers both within a plant and between different plants. File exchange between technical applications in product development, engineering and manufacturing is usually supported.

- Private X.25 to provide efficient remote terminal access in-house and via gateways to external systems and network services.

- TCP/IP to interconnect LANs for direct bulk data transfers of CADCAM data.

In order to provide high availability and maximum resilience most regions are connected via different geograhic routes to at least two other regions. Also back-up processing facilities within regions are also connected in this way.



To optimize costs, "bandwidth" is shared by multiplexing different datastreams over the same physical circuit. At this point voice/data integration is also envisaged at an international level. At the same time opportunities are investigated to channel facsimile (64 Kbps), picture telephone (128 Kbps) and Video Conferencing (384 Kbps) datastreams via the internal network.

5. NETWORK APPLICATIONS WITHIN PHILIPS

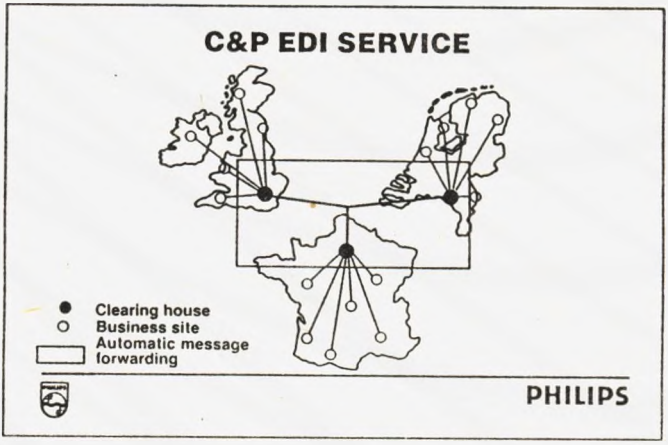
The need for communications is increasing both within the Philips corporation, between Philips and its business partners and distributors and most importantly with its clients.

In defining a new structure for business units with global product and marketing responsibility, Philips is in an excellent position to take advantage of new opportunities arising in the networking area.

Distribution channels both in professional and consumer goods need up-to-date technical, price and service information. Advertising and promotion plays an ever increasing role in preserving current market positions and creating new ones.

Partnerships with public telecommunications and private broadcasting agencies have proved to be very succesful in exploiting new opportunities.

In France, Philips Consumer Service was one of the early users of Minitel enabling up-to-date product information to be distributed. Now it is pioneering image transmission for service documentation via ISDN and a similar approach may be adopted in Germany.



Electronic data interchange is revolutionizing the way business transactions between different companies are recorded and exchanged. Philips C&P operates international EDI services which comprise message clearing house facilities, authentication, integrity and security control. Translation facilities convert between the internal Philips standard (COPS) and the now world-wide EDIFACT standard. In each region, EDI service centres provide the consultany, application software skills and basic services both to internal and external customers.

Electronic mail is world-wide available as an in-house facility integrated with gateways to access the public telex network in a large number of countries. Starting-up is interconnection to public electronic mail facilities (X.400) from various operators. At the same time internal interconnectivity between dissimilar E-Mail environments is improved using standard products both from IBM, DEC and Philips. Procedures for exchange of directory information between these systems are implemented.

Business applications for planning and logistics are the most demanding today. They support interactive access on a world-wide basis and support ways of information consolidation in central order and dispatch files which are input for production and delivery schedules. As a consequence, in this area service level agreements have to cover:

- world-wide accessability
- availability of 24 hrs a day, 365 days a year
- reliability of network software and hardware; error control and change management
- fast response times (typically less than 2 sec.)
- customer support both on application and network level (central and local)
- technical support tied into strict problem and change procedures.

Fast exchange of large files between design centres of integrated circuits in various plants and the IC production facilities is extremely important to reduce the time lag between development and actual production. We have taken advantage of the new digital 64 Kbps fixed leased circuits offered by the European PTT administrations and of new technology for interconnection of Ethernet LAN'S using the TCP/IP protocol.

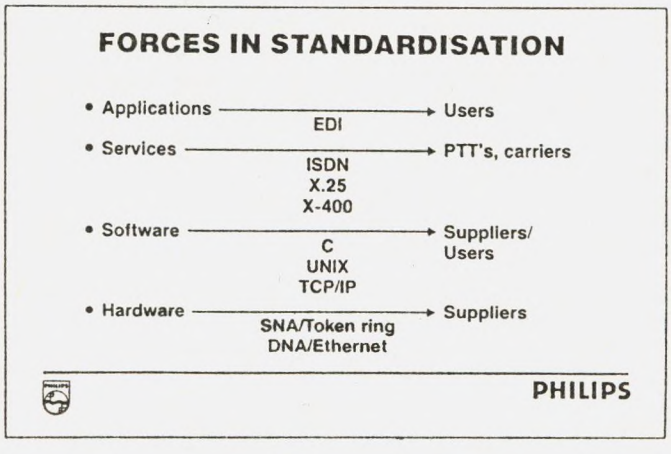
The education of specialists in various fields is very difficult to achieve. Consequently Philips participates in a European initiative for distant education called Euro Pace. This uses overnight satellite TV transmission to distribute taped presentations by highly professional lecturers for viewing by small audiences in universities and industry. An electronic mail and conferencing facility backs up the presentation material to provide a feedback channel with the lecturer and a method of discussion between participants.

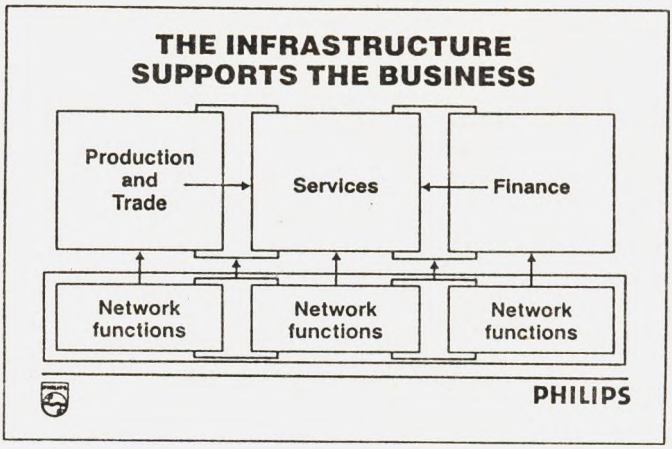
With the arrival of suitable equipment on the market, this year will see the large scale introduction for institutional applications of D2MAC broadcasting in Europe. This also offers the possibility to address restricted audiences directly, e.g. all the Philips distributors in the Netherlands and Europe with product information, the latest Philips background stories and any other information to stimulate motivation and enhance knowledge at the same time.

6. CONCLUSIONS

Communications and data processing technologies support entirely new ways of doing business.

A basic public network infrastructure is essential to allow various forms of value added applications to blossom.





Large benefits will be created for consumers, public and private organizations alike, if partnerships between the various players create the economic platform for launching new applications.

In this respect, structural changes which would allow telecommunications agencies to engage in such partnerships, are about to be realized in Europe. It would not be wise of the telecommunications agencies to continue to focus so much on telephony alone. In Europe in particular non-voice services, cable and satellite options need to be given a much better chance and should not be left to largely diverging national approaches.

Within Philips, C&P is a natural partner both for providing communications and processing services to the Philips business units and their trade relations.

By exploiting highly standardized processing and communications platforms, Philips Communications & Processing Services provides economies of scale not attainable by individual business units. This also helps to reduce application software costs to a large extent and, at the same time, creates more opportunities to support flexible customer oriented solutions for information and communications requirements.

Both end-user requirements and service levels based on business requirements have been identified to complement the public infrastructure with an adequate in-house capability.

AUTHOR INDEX

AHOOJA, R.	FRG	4-20
AHTIAINEN, A.	FINNLAND	5-48
AMBROIS, O.	ITALY	7-15
BAHK, S.	USA	1-15
BARAKA, H.	EGYPT	1-75
BARRETTO, M.L.	FRANCE	4-51
BLOME, F.	FRG	1-137
BOCHMANN, v. G.	CANADA	4-1
BUHRANDT, U.	GDR	5-11
CARL, D.	GDR	5-76
CASTANET, R.	FRANCE	5-1
COURTIAT, J.-P.	FRANCE	4-51
CSABA, L.	HUNGARY	6-12
CSOPAKI, GY.	HUNGARY	3-1
CSUHAJ-VARJÚ, E.	HUNGARY	6-12
el DESSOUKI, O.	EGYPT	1-75
DUBIELEWICZ, A.	POLAND	5-33
DUBIELEWICZ, I.	POLAND	5-33, 5-42
EVERS, V.H.C.M.	HOLLAND	8-23
FARO, A.	ITALY	5-21
FILIPIAK, J.	POLAND	1-56
FOERSTERLING, F	GDR	2-31
FREESE, J.	SWEDEN	7-7
GABLER, H.	FRG	P-20
GARBE, K.	GDR	3-20
GASSMANN, H.P.	FRANCE	7-1
GOLLNICK, d.	GDR	3-9
HAMICH, A.	FRG	1-137
HARANGOZÓ, J.	HUNGARY	1-65
HE, Q.	JAPAN	6-22
HEYMER, V.	GDR	5-11
HINKELMANN, J.	GDR	5-11
HO, W.P.C.	USA	6-1
HOLZINGER, K.	FRG	3-28
JAKOBS, K.	FRG	2-53, 2-63
JACOBS, S.	FRG	1-25
KESKINEN, J.	FINNLAND	5-48
KLEHN, N.	GDR	2-31
KOLESNIK, K.	POLAND	5-42
KOMOROWSKI, W.	POLAND	5-33

KOENIG, H.	GDR	5-90
LAMPRECHT, H.	GDR	3-9
LEAKEY, D.M.	UK	P-1
LOGRIPPO, L.	CANADA	4-1
MARTINI, P.	FRG	1-45
MEANDZIJA, B.	USA	6-1
de MEER, J.	FRG	4-20
MEULEMANS, N.	BELGIUM	2-41
MOLISZ, W.	POLAND	1-85
MOLNÁR, T.	HUNGARY	3-1
NAJM, E.	FRANCE	5-59
PAPP, G.	HUNGARY	1-65
PETRENKO, A.	USSR	5-21
PISTELAK, J.	CZECHOSL.	5-100
PRIBYL, J.	CZECHOSL.	1-95
PROBERT, R.	CANADA	4-30
QUERNHEIM, U.	FRG	1-25
QUEIROZ, J.	FRANCE	5-59
RAFALOW, L.M.	USA	8-1
RAFIQ, O.	FRANCE	5-1
RATAJCZAK, J.	POLAND	5-33
REICHERT, F.	FRG	2-63
SABEV, V.	BULGARIA	2-31
SAITO, Y.	JAPAN	2-21
SALEH, K.	CANADA	4-30
SAREK, M.	CZECHOSL.	5-100
SARIKAYA, B.	CANADA	4-1
de SAQUIS-SANNES, P.	FRANCE	4-51
SCHNEIDER, G.M.	USA	1-104
SCHNEIDER, P.	FRG	1-1
SERHROUCHNI, A.	FRANCE	5-59
SIMOLIN, J.	FINNLAND	5-48
SPANIOL, O.	FRG	P-10
SZABÓ, CS.	HUNGARY	3-1
SZANDI, L.	HUNGARY	3-1
TANAKA, H.	JAPAN	6-22
TARNAY, K.	HUNGARY	6-32
TARPILA, K.	FINNLAND	5-48
TERAMOTO, M.	JAPAN	2-21
TIETZ, W.	FRG	2-1
TOLLKÜHN, P.	GDR	5-11
TRIFONOV, R.T.	BULGARIA	1-128
TRIFONOVA, S.G.	BULGARIA	1-128
TURUNEN, I.	FINNLAND	5-48
TWYVER, D.A.	CANADA	8-11
TZANOV, V.A.	BULGARIA	1-128

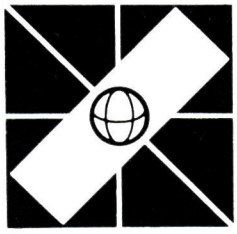
UEGAKI, K.	JAPAN	2-21
VAJDA, I.	CZECHOSL.	1-95
VARGA, L.ZS.	HUNGARY	4-41
WAGNER, P.A.	HUNGARY	6-32
WAGNER-DIBUZ, S.	HUNGARY	6-32
WOZNIAK, J.	POLAND	1-35
el ZARKI, M.	USA	1-15
ZASTROW, M.N.	USA	1-104

LIST OF REVIEWERS

AZEMA, A.
BERBEAU, M.
BOCHMANN, v.G.
BRADY, S.
BRINKSMA, E.
CHANSON, S.
CSABA, L.
CSOPAKI, GY.
DALLOS, GY.
DSSOULI, R.
ERCSÉNYI, A.
FILP, A.
FOURNIER, R.
FÖLDVÁRI, R.
FUJIWARA, S.
GANTENBEIN, D.
HARANGOZÓ, J.
HOLZMANN, G.I.
HORVÁTH, P.
JAKOBS, K.
KEREKES, I.
KOENIG, H.
LAJTHA, GY.
LECOMTE, L.
LÖFFLER, H.
MAKINO, Y.
MAZGON, S.
MEANDZIJA, B.
MONDAIN-MONVAL, P.
OUIMET, D.
PETRENKO, A.
PRANEVICIUS, H.
PUJOLLE, P.
QUEMADA, J.
RAYNER, D.
RICO, N.
ROSENBERG, C.
RUDIN, H.
SARIKAYA, B.
SHIRATORI, N.
SIDHU, D.
SZABÓ, CS.
TENNEY, R.L.
TELBISZ, F.
TÓTH, P.
ULLMANN, K.
VARGA, L.ZS.

MTESZ - egyesületi használatra !

Kiadja: Neumann János Számitógéptudományi Társaság
Készült: 300 példányban
90/496 MTESZ Házinyomda, Budapest.
Felelős vezető: Boncza Gábor



ICCC — IFIP — ITU — UNESCO

COMNET'90

"Forward into the second quarter-century in networking"

preliminary program

8—10 May 1990

Budapest, Hungary

The John von Neumann Society for Computing Sciences

iTA/319/mell.

CONTENTS

Message from Miklós Havass 1

General information 2

Detailed conference program 4

Technical information 17

Optional programs 19

At-a-glance timetable 20

319/meet.

Message from the President of the John v. Neumann Society

Ladies and Gentlemen,

It is with great pleasure that I extend a welcome to all of you coming to COMNET' 90.

Being staged for the fourth time within fifteen years by our Society, this event is of outstanding importance for us. Not only has it made tradition over the years past but has indeed established itself as the most significant series of conferences in it's field in Eastern-Central Europe. Due to the position of the host country it provides an excellent opportunity for delegates coming from all over the world to exchange ideas, learn up-to-date facts, establish new personal contacts and build a bridge between experts from East and West.

The theme of our present meeting, Computer Networks, spans a wide scope of relevant topics ranging from telecommunications to computer applications and from information systems to the application of international law. Because of the appearance of ever renewing and diverse equipment there are always new problems to be solved, most of which fall into the domain of standardization but besides technical questions requiring adequate solution a lot of new application tasks too have to be performed again.

COMNET' 90 is the result of the enthusiastic efforts of a large number of volunteers, members of the program committee and of the team of referees, similarly that of the local organizing committee and of the international advisory group. The kind efforts of all of these colleagues paved the way to the present day and hopefully to the success of the conference.

Many international institutions contributed to COMNET' 90, above all the IFIP TC-6 committee. Hearty thanks to all of them.

Hoping the speakers and delegates will find the conference useful and will yet again return to their research bases with a good deal of new experience made I wish every success to all participants.

Miklós Havass

C O M N E T '90

"FORWARD INTO THE SECOND QUARTER-CENTURY IN NETWORKING"

SPONSORED BY : IFIP (as a major event of TC-6)

CO-SPONSORED BY : ICCC, ITU and UNESCO

HOSTED BY : The John v. Neumann Society for Computing Sciences

SUPPORTED BY : The Hungarian Academy of Sciences

SCOPE

As has been the case at earlier COMNET conferences, this time again the state-of-the-art in networking will be thoroughly discussed, with an overview of results, bottlenecks and development trends in networks of various dimensions, topologics and destinations.

Contributions will present research achievements by the academic world as well as development strategies of some companies in the forefront of networking.

TOPIC AREAS

1) LANs, MANs and ISDN.

2) Message handling systems and directories.

3) Management and monitoring of networks.

4) Protocol specification and verification.

5) Protocol implementation and testing.

6) Knowledge based systems in computer communications.

7) Data protection, privacy, authentication and security.

8) Fundamental contributions by major corporations affecting networking.

C O M N E T '90

CONFERENCE COMMITTEES AND OFFICIALS

Honorary Chairman: Valter, F. (H)
Conference Governor: Szentiványi, T. (H)

PROGRAM COMMITTEE

Tarnay, K. (H), Chair
Bailes, P. (UK)
Bakonyi, P. (H)
Bochmann, G. v. (CDN)
Danthine, A. (B)
Ercsényi, A. (H)
Gabler, H. (FRG)
Gagliardi, D., ETSI (I)
Handler, G. J. (USA)
Harangozó, J. (H)
Horváth, P. (H)
Ivanyos, L. (H)
Kappel, K. W. (USA)
Khakhar, D. (S)
Könyves Tóth, P. (H)
Lajta, Gy. (H)
Makino, Y. (J)
Mazgon, S. (H)
Meandzija, B. (USA)
Pouzin, L. (F)
Radford, P. (UK)
Rudin, H. (CH)
Schicker, P. (CH)
Spaniol, O. (FRG)
Szabó, Cs. (H)
Telbisz, F. (H)
Tóth, P. (H)
Uhlig, R. (USA)
Ullmann, K. (FRG)
Westcott, J. (USA)

INTERNATIONAL ADVISORY BOARD

Basu, D. (IND)
Bazewicz, M. (PL)
Boutmy, E. J. (NL)
Boyanov, K. (BG)
Casaca, A. (P)
Ekberg, J. (SF)
Eliassen, F. (N)
Housley, T. (AUS)
Königshofer, F. (A)
Ono, K. (J)
Owolabi, A. (UNESCO)
Puzman, J. (CS)
Riera, J. (E)
Samoylenko, S. (SU)
Sattler, S. (GDR)
Sobczak, W. (PL)
Twyver, D. A. (CDN)

ORGANIZING COMMITTEE

Chair: Gergely, Cs.
Co-chair: Bagonyi, L.
Public relations: Brückner, H.
Proceedings: Csaba, L.
Secretary: Hetthéssy, G. (Ms.)
Treasurer: Tóth, M. (Ms.)
Industry liaison: Vadász, P.

DETAILED CONFERENCE PROGRAMME

TUESDAY, MAY 8

OPENING SESSION

Chairman: T.Szentiványi

10.00 Opening of the Conference on behalf of the Hungarian Government

Greetings, addresses on behalf of the sponsoring and host organisations
R.UHLIG, Chairman of TC6, IFIP
M.HAVASS, President, John v. Neumann Society

PLENARY SESSION

10.30-11.15 High Speed Local Area Networks: What, Why, When and How: Planning, Installation and First Experiences of a HSLAN in a Heterogeneous Environment (P-1)
O.SPANIOL, FRG

11.15-12.00 Formal Specifications for Protocols: Issues and Experiences (4-1)
G.v.BOCHMANN, L.LOGRIPPO, B.SARIKAYA, CANADA

LUNCH BREAK

TUESDAY, MAY 8

TRACK A

PLENARY SESSION

Chairman:R.Uhlig,IFIP
Co-chairman:L.Csaba, Hungary

14.00-14.45 Commercial development of OSI management (3-1)
K.KAPPEL, USA

14.45-15.30 Meta-Management of Communications Networks (6-1)
B.MEANDZIJA, W.P.C.HO, USA

BREAK

15.45-16.15 Experience with the Astride Testing Approach (5-1)
O.RAFIQ, R.CASTANET, FRANCE

16.15-17.00 Issues in Computing Security and Telecommunications Network Integrity (7-1)
J.FREESE, SWEDEN

TUESDAY, MAY 8

TRACK B

PLENARY SESSION

Chairman: D.A.Twyver, Canada
Co-chairman: Gy.Lajtha, Hungary

14.00-14.45 Telecommunication Standardization in Western Europe (P-2)
H.GABLER, FRG

TOPIC 1 LANS, MANS AND ISDN

14.45-15.30 A Concept Becomes Reality: First Experiences with Commercial ISDN Services (1-1)
P.SCHNEIDER, FRG

BREAK

15.45-16.15 A Routing Scheme for a Non-Strict Hierarchical Networking Environment (1-2)
S.BAHK, M.el ZARKI, USA

16.15-16.45 ARQ Error Control for Multipoint Satellite Communication (1-3)
S.JACOBS, U.QUERNHEIM, FRG

16.45-17.15 An Approximate Analysis of Multi-Hop S-ALOHA Networks with or without Immediate Acknowledgements (1-4)
J.WOZNIAK, POLAND

WEDNESDAY, MAY 9

TRACK A

MORNING SESSION

Chairman: H.Gabler, FRG
Co-chairman:S.Mazgon, Hungary

TOPIC 1 LANS,MANS AND ISDN

9.00- 9.30	Towards High Speed Networking in Local and Metropolitan Areas (1-5) P.MARTINI, FRG
9.30-10.00	Performance Evaluation of Token Passing Bus LANs (1-6) G.PAPP, J.HARANGOZO, HUNGARY
10.00-10.30	Bandwidth Allocation and Flow Control in ATM Networks (1-7) J.~~PILIPIAK~~, POLAND SALZBORN
	BREAK
10.45-11.15	Performance Models of Access Integration Policies in ISDN Channels (1-8) O.el DESSOUKI, H.BARAKA, EGYPT
11.15-11.45	A Survey of Fault-Tolerant LANs (1-9) W.MOLISZ, POLAND
11.45-12.15	An Algorithm for the Design of Multilevel Concentrator Networks (1-10) G.M.SCHNEIDER, USA
	LUNCH BREAK

WEDNESDAY, MAY 9

TRACK B

MORNING SESSION

Chairman: G.v.Bochmann, Canada
Co-chairman: J.Harangozo, Hungary

TOPIC 4 PROTOCOL SPECIFICATION AND VERIFICATION

9.00-9.30 Liveness Verification as a Tool for Protocol Validation (4-2)
L.Zs.VARGA, HUNGARY

9.30-10.00 A Synthetic Localized Approach to Protocol Validation (4-3)
K.SALEH, R.PROBERT, CANADA

10.00-10.30 Protocol Validation Using Process Algebraic Recursive Equations (4-4)
J.de MEER, R.AHOOJA, FRG

BREAK

10.45-11.15 Experience in Using ESTELLE for the Specification and Verification of a Fieldbus Protocol: LIP (4-5)
L.BARETTO, FRANCE

11.15-11.45 Experience in Conformance Testing with the Test System TEKOS (5-2)
U.BUHRANDT, V.HEYMER, J.HINKELMANN, P.TOLLKÜHN, GDR

11.45-12.15 Sequence Generation from EFSMs for Protocol Testing (5-3)
A.FARO, A.PETRENKO, ITALY-USSR

LUNCH BREAK

WEDNESDAY, MAY 9

TRACK A

AFTERNOON SESSION

Chairman: O.Spaniol, FRG
Co-chairman: Cs.Szabo, Hungary

14.00-14.30 O&M Experience Gained in the Introduction of ISDN (1-11)
F.BLOMER, A.HAMICH, FRG

14.30-15.00 Error Statistics in Data Networks (1-12)
J.PRIBYL, I.VAJDA, CZECHOSLOVAKIA

15.00-15.30 An Approach to Optimal Reliability Assurance in Distributed Information System (1-13)
S.TRIFONOVA, R. TRIFONOV, BULGARIA

15.30-16.00 Directory Names and Schema- An Evaluation (2-5)
K.JAKOBS, FRG

BREAK

16.30-18.00 PANEL DISCUSSION
Chairman: K.Ullmann, FRG
Host: P.Bakonyi, Hungary
Panelists: D. Basu, India
G. v. Bochmann, CDN
H. P. Gassmann, OECD, FRG
T. Housley, Australia
K. Kappel, USA
Y. Saito, Japan
R. Uhlig, USA
K. Ullmann, FRG
Theme: Problems in networking an how to overcome them

WEDNESDAY, MAY 9

TRACK B

AFTERNOON SESSION

Chairman: A.Danthine, Belgium
Co-chair: K.Tarnay, Hungary

14.00-14.30 Protocol Software Engineering Tools for Implementation of a General Purpose OSI Stack (5-4)
A.P.AHTIAINEN, J.KESKINEN, J.SIMOLIN, K.TARPILA, I.TURUNEN, FINLAND

14.30-15.00 Experience in Computer-Aided Protocol Implementation (5-5)
H.KOENIG, GDR

15.00-15.30 The Pre-Implementation and Verification of LOTOS (5-6)
E.NAJM, J.QOUIROZ, A.SERHROUCHNI, FRANCE

15.30-16.00 Network LOPOS-Performance Analysis (5-7)
J.PISTELAK, M.SAREK, CZECHOSLOVAKIA

BREAK

16.30-18.00 PANEL DISCUSSION
Chairman: K.Ullmann, FRG
Host: P.Bakonyi, Hungary
Panelists: D. Basu, India
G. v. Bochmann, CDN
H. P. Gassmann, OECD, FRG
T. Housley, Australia
K. Kappel, USA
Y. Saito, Japan
R. Uhlig, USA
K. Ullmann, FRG
Theme: Problems in networking an how to overcome them

WEDNESDAY, MAY 9

TRACK C
AFTERNOON SESSION

Chair:K.Kappel, USA
Co-chairman: F.Telbisz, Hungary

TOPIC 3 MANAGEMENT AND MONITORING OF NETWORKS

14.00-14.30 A Traffic Generating and Monitoring Technique for Standard and Integrated Voice/Data Ethernet LANs (3-2)
Gy.CSOPAKI,T.MOLNAR, CS.SZABO, L.SZANDI, HUNGARY

14.30-15.00 Networking Management in Heterogeneous Computer Centre (3-3)
D.GOLLNICK, H.LAMPRECHT, GDR

15.00-15.30 An Experimental OSI Network Management Implementation for a PC Network (3-4)
K.GARBE, GDR

BREAK

15.30-16.00 Central System Support for Digital Switching Systems (CCS) (3-5)
K.HOLZINGER, FRG

16.30-18.00 PANEL DISCUSSION
Chairman: K.Ullmann, FRG
Host: P.Bakonyi, Hungary
Panelists: D. Basu, India
G. v. Bochmann, CDN
H. P. Gassmann, OECD, FRG
T. Housley, Australia
K. Kappel, USA
Y. Saito, Japan
R. Uhlig, USA
K. Ullmann, FRG
Theme: Problems in networking an how to overcome them

THURSDAY, MAY 10

TRACK A

MORNING SESSION

Chairman: K.Ullmann, FRG
Co-chairman: A.Ercsényi, Hungary

TOPIC 2 MESSAGE HANDLING SYSTEMS AND DIRECTORIES

9.00-9.45 Standards for Message Handling and their Impact (2-2)
W.TIETZ, FRG

9.45-10.15 Extensions and Evaluation of Method for MHS Interwork (2-3)
Y.SAITO, M.TERAMOTO, K.UEGAKI, JAPAN

10.15-10.45 Approaches to the MHS Message Store Structures (2-4)
V.SABEV, F.FOERSTERLING, W.KLEHN, BULGARIA

BREAK

11.00-11.30 A Simple Multi-Purpose Internet Directory (2-5)
N.MEULAMANS, BELGIUM

11.30-12.00 Using Directory Services in Mobile Communication (2-6)
K.JAKOBS, F.REICHERT, FRG

LUNCH BREAK

THURSDAY, MAY 10

TRACK B

MORNING SESSION

Chairman: B.Meandzija, USA
Co-chairman: P.Toth, HUNGARY

TOPIC 5 PROTOCOL IMPLEMENTAION AND TESTING

9.00-9.45 Conformance Testing System for the KASK Network: Some Implementation Problems (5-8)
A.DUBIELEWICZ, I.DUBIELEWICZ, W.KOMORSKI, J.RATAJCZAK, POLAND
Comments on TTCN Implementaion in the KASK Network (5-9)
I.DUBIELEWICZ, K.KOLESNIK, POLAND

9.45-10.15 Functional and Performance Prototyping Concepts Based on SDL (5-10)
C.CARL, GDR

BREAK

TOPIC 6 KNOWLEDGE BASED SYSTEMS IN COMPUTER COMMUNICATION

10.30-11.00 Distributed/Cooperative Office (6-2)
L.CSABA, E.CSUHAJ-VARJU, HUNGARY

11.00-11.30 SBS: A Multimedia Service Base System Based on Distributed Objects (6-3)
Q.HE, H.TANAKA, JAPAN

11.30-12.00 Frames for Protocol Representation (6-4)
K.TARNAY, S.WAGNER-DIBUZ, P.A.WAGNER, HUNGARY

LUNCH BREAK

THURSDAY, MAY 10

TRACK A

AFTERNOON SESSION

Chairman: L.Pouzin, France
Co-chairman: P.Könyves-Toth, Hungary

TOPIC 7 DATA PROTECTION AND SECURITY

14.00-14.45 Computer Networks, Privacy, Protection and Security (7-2)
H.P.GASSMANN, FRANCE

14.45-15.30 Aspects of Security, Integrity and Privacy in the Telecommunication Network (7-3)
O.AMBROIS, ITALY

15.30-16.00 Transborder Aspects of Networking (7-4)
R.PIPE, UK

BREAK

CLOSING PLENARY

Chairman: Cs.Gergely, Hungary

16.15 Address: O.Spaniol, IFIP,TC6

THURSDAY, MAY 10

TRACK B

AFTERNOON SESSION

Chairman: J.Koen, South Africa
Co-chairman: P.Horvath, Hungary

TOPIC 9 MAJOR CORPORATIONS

14.00-14.30 SNA Usability Enhancement for the 90's (8-1)
L.M.RAFALOW, USA

14.30-15.00 Fifth Generation Private Wide Area Networking (8-2)
V.H.C.M.EVERS, UK

15.00-15.30 Foundation for Future Networking (8-3)
D.A.TWYVER, CANADA

15.30-16.00 Network Planning within a Large Corporation (8-4)
J.DIPPE, FRG

BREAK

CLOSING PLENARY

Chairman: Cs.Gergely, Hungary

16.15 Address: O.Spaniol, IFIP,TC6

THURSDAY, MAY 10

TRACK C

AFTERNOON SESSION

Chairman: D.Kakhar, Sweden
Co-chairman: L.Ivanyos, Hungary

LATE CONTRIBUTIONS

14.00-14.15 Distributed Systems based on Open System Interconnection
G.M.CHILD, UK

2. 14.15-14.30 Knowledge-based Tools for Networking
M.BAZEWICZ, POLAND

3. 14.30-14.45 S-CORE: an Operating System for LANs
J.NEMETH, HUNGARY

14.45-16.00 Poster presentations

BREAK

CLOSING PLENARY

Chairman: Cs.Gergely, Hungary

16.15 Address: O.Spaniol, IFIP,TC6

1. Lea Hubert: Telecommunication Standards

4. Gallnick

5. Sdaa : New approach for Conformance Testing

COMNET '90 TECHNICAL INFORMATION

WORKING LANGUAGE

English

ADDRESSES OF IMPORTANCE

Venue of conference, get-together party, TC-6 meeting, accomodation

Hotel T. O. T.
H-1121 Budapest, Normafa út 54.

COMNET '90 Secretariat
John v. Neumann Society for Computing Sciences
H-1368 Budapest, P. O. B. 240
Telephone: (36-1) 132-9349
Telex: 225792
Telefax: (36-1) 156-1215

JOINING INSTRUCTIONS

Participants should check in at the COMNET '90 conference desk in the lobby of Hotel T. O. T. The conference desk will be open daily from 8.00 am to 16.00 pm, from Sunday, May 6 to Thursday, May 10.

Upon registration participants will receive their registration kits including the preprint of the conference, the identification badge, invitations to the get-together party and the banquet and other related materials.

Registrants are requested to wear their badges throughout the conference.

Those participants who haven't transferred conference fees in advance can effect payment on the spot.

COMNET '90 TECHNICAL INFORMATION (continued)

HUNGARIAN ENTRY VISA

A Hungarian visa is issued by the Hungarian diplomatic missions abroad. Visas are also available at the appointed highway frontier stations and at Budapest Ferihegy Airport. It is advisable, however, to have the visa issued in advance to avoid long waiting at the frontier stations. Children under 14 and registered in the passport of the parents need no visa. The visa permits a stay of 30 days in Hungary and is valid for 3 months. Citizens of Sweden, Finland, Italy, Austria and Malta need no visa.

Attention! No visa can be obtained at the railway frontier station if you arrive by train.

To have the Hungarian visa issued, the following documents are necessary:

a) Passport valid to Hungary
b) Visa application form filled out in triplicate
c) 2 photos (not older than six months)

TRANSPORT

The venue of the conference can be reached by red line bus No 21 starting from Moszkva tér (Moscow Square, also a Metro station on Line 2). Get off at the Olympia Hotel stop.

There is an airport bus service connecting Ferihegy airport with Deák tér in dowtown Budapest. Metro Line 2 has also a station on Deák tér providing a connection to Moszkva tér. All railway stations have metro connections, too.

Taxi fare from Ferihegy airport to Hotel T.O.T. is expected to be in the range of five to six hundred forints.

CATERING

Participants will be served refreshments in the breaks between sessions.
Lunch is available at a price of 10 USD/day at the restaurant of Hotel T.O.T.

COMNET '90 OPTIONAL PROGRAMMES

PRE-CONFERENCE TOUR

This all day programme is offered to participants and accompanying persons wishing to get acquainted with the South-Hungarian countryside. Route: Budapest-Szeged-Kecskemét-Gerébi puszta-Budapest. The programme includes: Sightseeing in Szeged, which is a beautiful city considered to be the cultural centre of South-Hungary, 180 km from Budapest. Special fish-soup will be served in an old-fashioned restaurant during the trip. A pleasant walk in the downtown of Kecskemét offers unique opportunities to visit the City Hall, Synagogue, Kodály Institute, Toy museum, and to have a welcome drink with the mayor. Further programmes in Gerébi puszta include horse show, horse riding, carting in the farm of the castle Gerébi.

6th May Price: USD 55

A TOUR TO THE CASTLE DISTRICT

The Castle Hill is a beautiful picture as viewed from the city, while the panorama of the city offered from the Castle is also arresting. The Castle District is a quiet area with well-preserved old houses, public collections, museums scientific institutes and churches. A visit is a must for all who feel like learning about the past of Hungary and her capital. The guided tour includes the Royal Palace, Matthias Church, Fishermen's Bastion, the Casemates, and an unforgettable walk in the Castle District.

7th May, morning Price: USD 10

CHURCHES IN HUNGARY

This special programme includes visits to churches of different denominations in Budapest with ecclesiastical and church historical reviews. (Roman Catholic Basilica, Evangelical Church in Deák Square, Presbyterian Church in Calvin Square, Synagogue, Serbian Church).

7th May, afternoon Price: USD 10

POST-CONFERENCE TOUR

The route of this all-day tour is: Budapest-Székesfehérvár-Veszprém-Balaton-Budapest.
Székesfehérvár: a walk in downtown Székesfehérvár, the thousand years old royal capital.
Veszprém: sightseeing in this picturesquely situated ancient episcopal seat with a number of beautiful art memorials.
Balatonfüred: a short walk in the first health resort city in Hungary situated on the north shore of Lake Balaton.
Csopak: lunch in a typical Hungarian folk-restaurant.
Tihany: sightseeing on this marvellous peninsula of the Lake Balaton, visit to the Abbey built in the 11th century, a review of the exhibitions of the peasants' art and craft products.
Szántódpuszta: visit to a nobiliary countryhouse built in the 19th century and a specimen farm.
Gorsium: on the way back a short look at the relics of the Roman settlement.
11th May Price: USD 55

LADIES' PROGRAMMES

Cooking course in a famous Hungarian restaurant
8th May, morning Price: USD 8

City of spas: Briefing on the thermal waters of Budapest, on their effect and the balneoculture dating back to Roman times. Bathing in the Gellért Bath.
9th May, morning Price: USD 8

Shopping tour in downtown Budapest
10th May, morning Price: USD 6

COMNET'90 AT-A-GLANCE TIMETABLE

	TUESDAY MAY 8		WEDNESDAY MAY 9			THURSDAY MAY 10		
			TRACK A	TRACK B		TRACK A	TRACK B	
9.00			P. Martini 1-5	L. Zs. Varga 4-2		W. Tietz 2-2	A. Dubielewicz et al. 5-8	9.00
9.15							5-9	9.15
9.30			G. Papp et al. 1-6	K. Saleh et al. 4-3		Y. Saito et al. 2-3	C. Carl 5-10	9.30
9.45								9.45
10.00	OPENING SESSION		J. Filipiak 1-7	J. de Meer et al. 4-4		V. Sabev et al. 2-4	BREAK	10.00
10.15	Addresses/Greetings							10.15
10.30			BREAK			BREAK	L. Csaba et al. 6-2	10.30
10.45	PLENARY SESSION							10.45
11.00	O. Spaniol P-1		O. el Dessouki et al. 1-8	L. Baretto 4-5		N. Meulamans 2-5	Q. He et al. 6-3	11.00
11.15								11.15
11.30	G.v. Bochmann et al. 4-1		W. Molisz 1-9	U. Buhrandt et al. 5-2		K. Jakobs et al. 2-6	K. Tarnay et al. 6-4	11.30
11.45								11.45
12.00			G. M. Schneider 1-10	A. Faro et al. 5-3				12.00
12.15								12.15
	TRACK A	TRACK B	TRACK A	TRACK B	TRACK C	TRACK A	TRACK B	TRACK C
14.00		PLENARY SESSION	F. Blome et al. 1-11	A.P.Ahtiainen 5-4	Gy. Csopaki et al. 3-2	P. Gassmann 7-2	L.M. Rafalow 8-1		14.00
14.15	K. Kappel 3-1	H. Gabler P-2							14.15
14.30			J. Pribyl et al. 1-12	H. Koenig 5-5	D. Gollnick et al. 3-3		V.H.C.M.Evers 8-2	LATE CONTRIBUTIONS	14.30
14.45									14.45
15.00	B. Meandzija et al. 6-1	P. Schneider 1-1	S. Trifonova et al. 1-13	E. Najm et al. 5-6	K. Garbe 3-4	O. Ambrois 7-3	D.A. Twyver 8-3		15.00
15.15									15.15
15.30	BREAK		K. Jakobs 2-1	J. Pistelak et al. 5-7	K. Holzinger 3-5	R. Pipe 7-4	J. Dippe 8-4		15.30
15.45	O. Castanet et al. 5-1	S. Bahk et al. 1-2							15.45
16.00						BREAK			16.00
16.15		S. Jacobs et al. 1-3							16.15
16.30	J. Freese 7-1		PANEL DISCUSSION (16.30-18.00)			CLOSING SESSION Address: O. Spaniol, IFIP, TC 6			16.30
16.45		J. Wozniak 1-4							16.45
17.00									17.00
17.15									17.15
17.30									17.30
17.45						BANQUET (19.00) Host: M. Havass President J.v.Neumann Society			17.45
18.00	CONCERT (19.30)								18.00

Megnyitó:

T. Szentiváry

R. Uhlig

M. Havass

P. Horváth

Á. Csurgay

MTESZ HÁZINYOMDA 90.0527
Felelős vez.: BONCZA GÁBOR