ISSN: 0002 - 6557

ALTA FREQUENZA

1979 AGOSTO • (AEI) • VOL. XLVIII - N. 8

RIVISTA DI STUDI E RICERCA APPLICATA DELL'ASSOCIAZIONE ELETTROTECNICA ED ELETTRONICA ITALIANA

FASCICOLO IN LINGUA INGLESE
ENGLISH ISSUE N. 5

SPECIAL ISSUE ON EIN AND OTHER EUROPEAN COMPUTER NETWORKS

ALTA FREQUENZA

RIVISTA DI STUDI E RICERCA APPLICATA DELLA

ASSOCIAZIONE ELETTROTECNICA ED ELETTRONICA ITALIANA

SOTTO GLI AUSPICI DEL CONSIGLIO NAZIONALE DELLE RICERCHE

AEI

CONTENTS

COVER:

It shows in enlarged scale the main constructional features of our miniature feed-thrus. On assembling, sliding of the metallic part within the insulator creates a pre-anchorage which suits particularly well modern assembly techniques. Manufacturer FIRIE - Genova (Italy)

Proprietaria ed Editrice: AEI - Associazione Elettrotecnica ed Elettronica Italiana — *Direzione, Redazione, Ufficio Pubblicità e Amministrazione*: « Alta Frequenza » - Viale Monza, 259 - 20126 MILANO. *Telefono*: 2550641 - *Telegrammi*: Asselita, Milano - *Conto Corrente Post.*: 00274209 Milano — *Abbonamento*: I Soci dell'AEI hanno diritto di ricevere gratuitamente « L'Elettrotecnica » o « Alta Frequenza » col solo versamento della quota di associazione annua che varia secondo la Sezione territoriale di appartenenza; i Soci Juniores fino al 26º anno di età versano solo L. 10 500. I Soci possono ricevere ambedue le riviste col supplemento di L. 16 000. L'abbonamento per i non soci è di lire 35 000 per l'Italia e di L. 40 000 per l'Estero. *Fascicoli separati*; ciascuno: in Italia L. 3 500, all'Estero L. 4 000; nel caso di fascicolo speciale il prezzo, stabilito di volta in volta, sarà precisato nel fascicolo stesso — *Fascicoli degli anni precedenti*; ciascuno: in Italia L. 7 000, all'Estero L. 7 000. *Ogni cambio di indirizzo* deve essere accompagnato dall'importo di L. 500. Pubblicazione mensile — Spedizione in abbonamento postale: Gruppo III.

Prezzo del presente fascicolo speciale L. 5 000

Questa NO

Vol. XLVIII **Agosto 1979** **N. 8**

ALTA FREQUENZA

RIVISTA DI STUDI E RICERCA APPLICATA DELLA
ASSOCIAZIONE ELETTROTECNICA ED ELETTRONICA ITALIANA
SOTTO GLI AUSPICI DEL CONSIGLIO NAZIONALE DELLE RICERCHE
FONDATA DA GIANCARLO VALLAURI

AEI

Proprietaria ed Editrice: AEI - ASSOCIAZIONE ELETTROTECNICA ED ELETTRONICA ITALIANA.

Comitato per le pubblicazioni AEI: A. M. Angelini, F. Bianchi di Castelbianco, L. Dadda, G. Dal Monte, N. Faletti, E. Gatti, A. Gigli, P. Lombardi, M. Moretti, P. Peterlongo, P. Regoliosi, G. Ricca, R. Sartori, F. Tedeschi.

Direttore responsabile: G. Ricca.

Segreteria di Redazione: L. Tiné.

La rivista è pubblicata col concorso del Consiglio Nazionale delle Ricerche e della Fondazione Ugo Bordoni.

CONTENTS

Foreword

The most notable features of the development of computers in the next decade will be, on one side the widespread, pervasive use of computing devices at home and in the office (particularly through the microprocessors) and, on the other side the establishment of large systems composed by several large computers and data bases connected by means of complex transmission networks.

The development of such technology is essential for the implementation of future « information utilities » of national or international interest, which will be made accessible by everyone, by means of terminals, connected to the same transmission networks.

The same technology will also be the base for the widespread application of the so called « distributed informatics ».

The European Informatics Network, EIN, is the result of a research project sponsored by the Commission of the European Economic Community in order to promote an European expertise in the field of computer networks through an international cooperation.

This issue of Alta Frequenza is intended to offer a presentation of the main results so far obtained by the European Informatics Network project.

The need of computer networks has already been felt in various application areas. This is shown by the emergence of networks projects intended to satisfy specific needs.

Some of the operational networks developed in Europe are also illustrated in this issue: the SARA network, developed by universities in Lombardy (Italy) and the RPCNET developed by the Italian National Research Council.

Of particular interest is EURONET, promoted by the Commission of the European Economic Community as an operational network linking existing research computing centers and data bases belonging to various European countries, and implemented in accordance with the international standards issued by CCITT.

There is no doubt that computer networks, as are conceived today, are an important step in the evolution of telecommunication and computer technologies toward an increasing degree of integration.

Some of the most relevant problems emerged so far in such a process are discussed in this issue, which shows also in which directions future developments will take place.

Luigi Dadda
Guest Editor

A decade of development in computer communications

D.W. Davies
NATIONAL PHYSICAL LABORATORY, TEDDINGTON MIDDLESEX

Abstract. After a brief look at the early development of packet-switched computer communications the paper considers some critical factors in the development of a satisfactory network - the human interface, reliability and availability of services and adaptability to changes both in technology and in user requirements. Packet switched networks illustrate that the movement of intelligence towards the network periphery assists in all these respects. The paper concludes with a glance at the future of public networks.

1. INTRODUCTION

The European Informatics Network project is the subject of several papers in this issue of Alta Frequenza. This project has been a significant part of the first decade or so of research and development networks which has brought the packet switched computer network from the laboratory to every day use.

The way in which EIN has evolved illustrates the principle that the communication subnet should be simple and the network's intelligence should be at its periphery. In this paper the critical factors in network development are related to this trend towards peripheral intelligence.

2. EARLY DEVELOPMENT OF COMPUTER COMMUNICATIONS

In his introductory paper to the IEEE Proceeding special issue on packet communication networks [1] Larry Roberts has described in some detail the evolution of packet switching. The work of Paul Baran and his team at Rand Corporation in 1962-3 first established the concept of a distributed network with a short, standard message block and adaptive routing. Their main concern was survivability under enemy attack and almost instantaneous recovery of communication. They did not relate this design specifically to computer communications, but they needed to transmit both data and digital voice. Even at this early stage, enciphered, digitised voice communication was described in some detail. The paper by Gitman and Frank in the same issue [2] showed that packet voice is emerging again as an economic possibility.

The special affinity of packet communication with computers and terminals was brought out by two subsequent projects. In my own laboratory, from 1965 onwards, we built a packet switched network [3] for local communications (local networks are also a theme of much new work). We also embarked [4] on an extensive series of analyses of routing and flow control by computer simulation. In a paper of June 1966, which recorded a lecture I gave in March 1966, the word 'packet' first appears - a word which I chose carefully to be translatable but which caused trouble in Russian. The ARPA network project was planned in 1967 and began operation in 1969, giving the first experience of a larger-scale network based on packet switching and introducing a whole generation of researchers to these techniques.

Another important European development began in 1969, the SITA high level network, combining packet and message switching. Shortly afterwards there was the CYCLADES project, the first widespread network to exploit the form of packet switching which became known as a 'datagram' network and to build from it a reliable message transport service.

This was the environment in which the COST Project 11 began its life, and it learnt from the predecessors, adopting the datagram technique and many features of the CYCLADES transport protocol. By this time the centre of research had moved away from the communication subsystem, which was well understood, toward the protocols [5] needed to support network applications.

3. CRITICAL FACTORS IN NETWORK DESIGN

Packet switching can be regarded as a tool for the convenient and economic development of distributed information handling systems. The convenience and economy lies in the whole system, not just the communication component and it stems from the dynamically multiplexed interface, the inherent flow control and speed changing. The success of packet switching lies in the services to the end user.

Some ten years of experience since the first packet switching systems began to operate have taught us that there are three factors above all, which critically affect the quality of the network's services.

The first of these is our ability to design man-machine interfaces which are convenient and natural for most people to use. The second factor is the reliability and availability of the services. They cannot become an integral part of industry and commerce unless they can be utterly reliable. The third factor is the ability of the system to adapt itself to changes in technology and to new user requirements.

When we look at network services such as electronic mail, teleconferencing, and funds transfer we arrive at the conclusion that the goals of a good human interface, availability, reliability, and adaptability are best met by local intelligence. The trends in technology will ensure that this intelligence (and the storage it needs) is incorporated in the terminal. But the need for larger central systems remains. They take such forms as a message registry, a key distributing centre for secrecy and authentication and a "yellow pages" to tell the user where to look for the service he wants. Our proposition is that not only does technology persuade us to build more intelligence into the terminal but it is also a principle of good system design.

For the new era we can redefine the network as "that part of the system which serves a large number of users with common information requirements" and the terminal as "that part of the system which serves one user at a time".

4. THE HUMAN INTERFACE

The chief role of intelligence at the terminal will be to transform the human interface of the system. Network systems are notoriously difficult to operate because, with few exceptions, the human interface has been designed by and for computer specialists. The general standard at present is poor but there are some shining exceptions. They show us what can be done, but we have little idea, in a scientific sense, how to achieve it.

Information processing standards are forging ahead but not even the most optimistic forecast can envisage a network which is much more uniform or better organised than those in existence today. The rate of technology development and the growth of new services will only make this goal more elusive. Consequently the user's own terminal must make up for the deficiency and help to provide a consistent interface.

The present day user of network services suffers from a lack of consistency in the human interface, even in the most trivial details. Having mastered the basic skill of interacting with one service the user should carry over that skill to another quite different service. For example, he should be able to move from a travel booking service to a market service without being conscious of a new language. If we cannot have conformity to comprehensive human interface standards at the user level, at least the terminal can transform the interfaces to make them appear as consistent as possible. This transformational capability will be a principal function of terminal design in the future.

Intelligence in the terminal can transform an information service in many other ways. It can select and store information for a particular "user profile". A high speed data stream from a broadcast satellite can be used to update remote data bases information which is widely used; financial, sporting, weather, travel, news, etc. The storage capacity which will be available will allow people to select the data bases they want to keep and have fully-updated information always available. In this way a broadcast service can be transformed to seem like a personal service.

In the same way, the use of knowledge bases obtained by a network can provide special services on demand. These could offer "conversational" advice of all kinds, medical, taxation, leisure activities, etc. The cost of access to these programs must be low and the updating frequent enough so that most users would prefer to pay again rather than keep an old copy.

Our discussion has focussed on the human interface, but we believe that an analysis of reliability, availability and adaptability would also show the value of using the simplest mechanisms at the centre and building the complex parts (adaption, recovery, encipherment, user languages etc.) close to the periphery.

Packet switching, particularly of the datagram kind, illustrates this movement of intelligence towards the periphery. The communication service is very simple and it leaves all the hard tasks to the communicating processes. The network has the job of reliable delivery of single short messages but what these mean, how they are to be interpreted and what to do if there is a temporary failure are all matters left to the communicating parties. The reliability performance of the network is high but it leaves the ultimate responsibility for recovery to the users equipment, where it must lie, because only here are the goals and requirements of the communicating processes fully understood. In this way, traffic with different requirements of accuracy, security, delay, throughput variation, etc. can be combined on the same network.

5. THE FUTURE OF PUBLIC NETWORKS

This picture of a relatively unintelligent communication network with highly capable terminals prompts a number of questions. Will the network be packet switched and, if so, will the packet switching have changed in any significant way? What new uses of packet switching might there be? Some of the established telecommunications carriers now think of packet switching as a temporary expedient to be replaced by an "integrated network" carrying all kinds of information by fast circuit switching methods.

Two observations throw doubt on this prediction. One is the existence of ambitious plans for public switching in many developed countries, which will result in world-wide packet systems in the early 1980's. Private packet networks are similarly proliferating, demonstrating in many cases that the public networks are coming rather too late for the demand. The complexity of the user's interface to a network and the investment in protocol developments, make a changeover to an "integrated" network unlikely. The counter-argument is that the integrated network will preserve the X.25 interface but implement real circuits instead of virtual ones. But a very fast circuit switch would not be compatible with the cost-conscious design of the telephone system. Furthermore, to achieve fast call establishment an integrated network would need to contain a highly-developed control-signalling system which would be a packet switching network.

A second observation is that if an integrated network was so important a goal the Telephone and Telex networks would have merged already. Their traffic patterns are much closer to each other than are data and voice. Different forms of traffic such as data, voice and facsimile will certainly share common subsystems, much as they now share transmission equipment. There will remain a need for different systems and services. This kind of "integrated network" does not assume

a single type of switching but suits the switching to the characteristic of the traffic.

The packet switched services will offer independent packets (so-called datagrams) for the very brief interactions and virtual circuits for longer sessions. To achieve high availability and reliability, the basic transport mechanism should be as simple as possible, handling independent packets, and enabling virtual circuits to be implemented from end to end at the user level. Because of the high traffic levels and line capacities the transit delay in the main trunks will be quite small, (e.g. at most several milliseconds). To preserve the rapid transit, synchronous satellite links will be avoided in the main trunk . Most of the store and forward delay will be in local access lines, so delay will be almost independent of distance, like the charges for network usage.

6. THE EIN PROJECT

The EIN project has spanned a period of rapid change in network ideas. When it began, packet switching was a controversial new technique and it was opposed by the established telecommunication carriers. As the project ends the era of the public packet switched network begins, with Transpac established and many other such networks close to operation. It is now time for the computer establishment to come to terms with the new situation. The process of adaption to the new ideas by computer systems has been very slow and has proceeded, in general, by a costly overlay to existing teleprocessing software, instead of exploiting the reward offered by better interfaces. Possibly the advent of the Bell System 'Advanced Communications Service' will help to accelerate this adaption. But EIN has been careful not to become 'network dependent' and has used the concept of a transport level interface which is independent of the underlying communication subnet. So they have ensured the adaptability of their work to future changes in communication techniques.

The project has been a rewarding experiment in European cooperation. It has done more than anything else to widen the circle of network and protocol expertise in Europe and given us at least some common ground for the many controversial issues that will arise worldwide in network standards.

There is little doubt that the movement of 'intelligence' outwards to the periphery of a teleprocessing system will continue. Perhaps this will change our basic philosophy of protocol design. Certainly it should be used to enable the central network function of multiplexing, switching and transmission to be simplified. As the EIN project is completed there is a need for similar imaginative European research projects to take its place.

REFERENCES

[1] L G Roberts: The evolution of packet switching, Proc. IEEE, vol.66, No 11, November 1978, pp 1307-1313.

[2] I Gitman and H Frank: Economic analyses of integrated voice and data networks, Proc. IEEE, vol.66, No 11, November 1978, pp 1549-1570.

[3] D W Davies, K A Bartlett, R A Scantlebury and P T Wilkinson: A digital communications network for computers giving rapid response at remote terminals, ACM Symp. Operating Systems Problems, Oct. 1967.

[4] W L Price: Data network simulation: Experiments at the National Physical Laboratory 1968-1976, Computer Networks, Vol.1, No.4, 1977, pp 199-210.

[5] D W Davies, D L A Barber, W L Price and C M Solomonides: Computer networks and their protocols, Wiley 1979.

The paper was received on February, 1979

EIN - an example of cooperative research in Europe

D. L. A. Barber
Director, Executive Body, EUROPEAN INFORMATICS NETWORK
c/o NPL, TEDDINGTON MIDDLESEX

Abstract.

The paper traces the evolution of the COST Project 11 - a European Informatics Network - from its inception to the present day. As a prelude to other papers in this special issue, it introduces the participants who have cooperated together to build and operate an international computer network for research purposes, and outlines their individual roles for developing techniques for its application.

The paper then discusses the part played by the Executive Body in providing a focus for the project as a whole, and concludes by considering some of the lessons that may be learned from this unique international venture.

1. INTRODUCTION

For me, COST Project 11 began on the 16th December 1970, when, as a UK delegate, I attended a meeting in Brussels to discuss a proposal to establish a pilot Informatics Network. The proposal was one of several that had been made by the European Communities CREST Committee (Scientific and Technical Research Policy), which had met under the chairmanship of Mr Aigrain in 1968. These Aigrain proposals were taken up by the COST Group (European Cooperation in the Field of Scientific and Technical Research) during 1969, and a number of study groups were formed to examine the proposals in more detail. It was the initial meeting of one of these study groups that I attended in late 1970. The study group met a further four times and by mid 1971 had prepared a report which was strongly in favour of the establishment of an Informatics Project, based on the construction of a packet switching communicatins network and the conduct of a research program to explore its applications. An international Agreement [1] aimed at bringing such a project into being was formulated, and was signed by nine European Governments, together with Euratom, on 23rd November 1971.

Subsequently, two other countries joined the project and so the present signatories are - France, Germany, Italy, Netherlands, Norway, Portugal, Sweden, Switzerland, the United Kingdom, Yugoslavia and Euratom. The Agreement states that the project shall facilitate research into data processing problems, shall permit the sharing of resources at centres, shall allow the exchange of ideas and the coordination of research programs, shall facilitate the comparison of ideas for national networks, shall promote the agreement of standards and finally, shall be a model for future networks whether for commercial or other purposes.

A technical annex to the Agreement estimated that the project would last five years. In the first two years a communications sub-network would be constructed linking centres nominated by the Signatories. The Computing Systems at these centres would then be joined together to form a Computer Network for advanced research, to be conducted over the remaining three years.

However, it is in the nature of research projects that some tasks prove more difficult than anticipated and EIN is no exception. Accordingly, the Management Committee decided to extend the duration of the project, firstly to the end of 1978 and subsequently for a further year. This required a rearrangement of the sub-network at the beginning of 1979, because the Swiss centre was obliged to discontinue its operations. The initial and present topologies of the sub network are shown in figure 1.

2. PROJECT STRUCTURE

The Agreement provided for the establishment of a Management Committee charged with the overall direction of the project, an Executive Body comprising a Director and three Technical Assistants responsible for day to day operations and a Technical Advisory Group of experts nominated by the signatories.

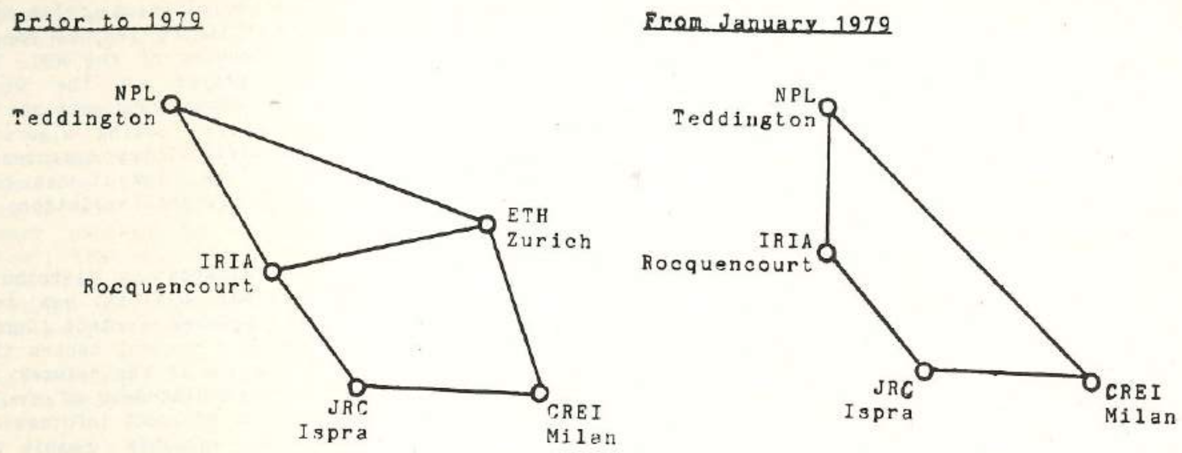


FIGURE 1: TOPOLOGY OF EIN SUBNETWORK

The Director reports to the Project's Management Committee, which is made up of representatives of all the Signatories and an observer from CEPT (the European Posts and Telecommunications Conference). The Commission of the European Communities has custody of the Common Fund, as the agent of the Management Committee, and also supplies the Secretary of that committee.

The Technical Advisory Group is the principal technical committee in the Project, and has representatives of all Signatories. In addition, a Centre Coordination Group comprises representatives of the Centres and discusses their technical work, while other committees and groups, mainly responsible for technical tasks, are set up and discontinued as necessary.

3. THE PACKET SWITCHING SUBNETWORK

At the time when the project was conceived, there were widely differing views on the form that future data networks should take, and indeed, on whether special facilities for data communications would in fact be required, for many doubted then whether data traffic would grow significantly for many years.

The idea of packet switching had been proposed in the early 1960s [2],[3] and research work had been carried out in the USA [4] and United Kingdom [5], but the idea of an international packet switching network was quite new. It was, therefore, something of a bold step for the study group to propose that the basis for a new project should be such a network. Fortunately, events have proved the rightness of the decision, because the majority of public data networks now in service, or being commissioned, are based on the packet switching principle [6].

The development of the original specification for the subnetwork was carried out by experts from the participating centres under the Chairmanship of the Project Director, and the subsequent analysis of tenders was done by another Working Party of experts who marked tenders according to a predetermined marking system. In this way, an independent objective assessment could be made, even though many different countries were involved in the selection of a contractor. As a result, a fixed price contract was awarded in October 1974 to SESA (France) and Logica (UK) as main contractors, with Selenia (Italy) and FIDES (Switzerland) as sub-contractors. The subsequent monitoring of the contractor's progress, and the development of prototypes was carried out on behalf of signatories by the Executive Body, and the subnetwork [7] was handed over to the Centres on schedule in May 1976.

4. THE COMMUNICATIONS SUBNETWORK

The Network Switching Centres (NSCs) that form the nodes of the subnetwork are based on the Mitra 15 from Compagnie Internationale pour l'Informatique (France). Nodes have either 28 or 32 K words of storage, a visual display unit and associated printer, a teletypewriter and paper tape peripherals; they also have specially developed hardware interfaces for the communications links to other NSCs and to Subscriber Computers. These implement the ISO HDLC standard for data transmission that applies a sum check to detect transmission errors but does not constrain the data transmitted. The interface is capable of operating at 48 k bit/S, but the subnetwork communication links operate (in full duplex) at only 9 600 bit/S. They are leased telephone lines, the total rental being about 8 MBF per year.

The subnetwork also has the following features:

Flow control: congestion is avoided by despatching a packet only if it has a serial number lying in a window of a certain size, starting from the serial number of the packet that was last acknowledged;

Transmission error control: errors are avoided by the monitoring of received packets and the automatic requesting of a retransmission when errors are detected by the check sum mechanism;

Adaptive routing: the rerouting of packets to avoid faulty areas of the network is based on the scheme adopted for the ARPA network;

Packet sequencing: the originator of a long message which will be cut up into packets may stipulate that the packets must be delivered to the destination in the correct order for reassembly of the message, since the destination may not have sufficient storage to accumulate and sort the packets itself;

Delivery confirmation: confirmation from the final destination back to the origin is available at the request of the originator;

Provision of diagnostic information on non-delivery, and of "trace" information: information on the route that a packet has taken through the network; both these features are useful for monitoring the performance of the network, as well as in developing the system; they are again optional at the request of the originator;

Facilities for collecting operating statistics;

Flexible arrangements to respond to a Network Control Centre, including the ability to completely reload the software of an NSC over the network from any other NSC;

Other technical features, such as preservation of the originator's reference number attached to a packet, and a set of "Virtual Subscriber Computer Processes" to provide various facilities, for example the anouncement of network time on request, the facility to send back a packet to the origin ("echoing"), etc.

A great deal of experience was gained in the design and development of the EIN subnetwork and this is well described in literature produced in the past five years. At the time of writing, intense activity is being conducted to gain as much information about the operation of the network possible, because the time is rapidly approaching when the network will be phased out and the participating centres will transfer their systems to Euronet, the international public packet switching network that will become operational during 1979 [8].

failures, or other such external events, that have caused corruption of the software serious enough to warrant a complete reloading of the NSC. The NSC hardware also has proved on the whole reliable. Even though the subnetwork consists of only six links, the adaptive routing algorithm helped to overcome some difficulties experienced in the initial phases of use, and it has been proved to react in a stable way to variations in traffic.

The ownership of the subnetwork is distributed amongst five organisations, but it has been accepted that a common Network Control Centre (NCC) is necessary, because a control centre that has information from the whole of the network can perform a much more refined diagnosis of faults than one which has only limited local information. This conclusion is one valuable result of international cooperation in network operations, and an open questions in EIN at present is that of the best administrative arrangements for controlling a distributed network.

In fact, a minimal network control function is possible from the operator's console of any NSC, but the facilities are not extensive, and the operator needs a considerable knowledge of the details of NSC operation to use them. The NPL has therefore developed an NCC system [9] that operates on a small computer, connected to the NSC as an additional Subscriber Computer. Its purpose is to monitor the overall performance of the subnetwork, in order to achieve reasonable levels of service and availability, and to reduce operational manpower by taking automatic corrective action wherever possible.

EIN MAP dd/mm/yy hh.mm

	Z	L	I	P	M	8	9	A	B
Z	-	D		D	D	D	*	*	*
L	D	-		U		U	U	U	U
I			-	U	U	U	.	.	.
P	D	U	U	-		U	.	.	.
M	D		U		-	U	U	.	.

U	up	.	not connected
D	down	*	unknown

Figure 2: Subnetwork Status Reports for Users

5. SUBNETWORK CONTROL

When EIN operations commenced, some faults were found in the software of the NSC; this was only to be expected with such a complicated system which could not be fully tested before it was connected to the Centres. But the majority of the problems appeared to have been solved by the Summer of 1977. Subsequently, it has usually been power

Perhaps the most notable feature of the NCC is its role in providing users with information about the behaviour of the network as a whole. A particular example is the map of the state of the network which can be called up by users from their own terminals, to indicate the condition of the lines, the network switches and the connected subscribers systems. This is shown in Figure 2. It is unfortunate that the services of the NCC will not

Barber D.L.A. - EIN - an example of cooperative research in Europe

be available to users when the subnetwork has been superseded by the use of Euronet.

6. THE PRIMARY CENTRES

The Signatories that have connected computer centres to the communications subnetwork have financed the installation and operation of the network centres by what is called "Concerted Action": that is, each Signatory is responsible for meeting its own costs, plus a share of the subnetwork costs, probably amounting to some 50 M BF for each Primary Centre up to the end of 1977.

The five Primary Centres, nominated by Signatories when the Project began are:-

CREI - Centro Rete Europea di Informatica

CREI was set up in 1975 as a centre of the Politecnico di Milano, sponsored by the Italian Ministry of Research. Networking research in the Politecnico, and subsequently in CREI, started in 1970, on such items as optimisation of the communication network, routing strategies, theory of colloquies, design of protocols, measurements and collection of statistics.

ETH - Eidgenoessische Technische Hochschule

ETH, the Federal Technical University at Zuerich, is one of the largest establisments in Switzerland for higher education and research in technology. It has a strong computer science faculty where the Pascal language was developed, amongst other achievements. The Computer Centre has three CDC mainframes, and until the end of 1978 accommodated the Swiss element of EIN, which was sponsored by the PTT and the Department of the Interior.

IRIA - Institut de Recherche d'Informatique et d'Automatique

IRIA is one of the leading institutions for computer science research in France. It has been involved in computer network research since 1972 with a pilot project called CYCLADES. In this context, a packet switching network (CIGALE) was first designed and implemented to link heterogeneous computer systems, of various scientific institutions. The network has been in full operation since 1975 acting as a focus for network activities in France.

Apart from pure packet switching techniques, IRIA's interests in networking mainly concern end-to-end protocols such as Transport protocol and Virtual Terminal Protocol. Network interconnection problems have been studied in the real case of the CYCLADES-EIN interconnection. Since the very beginning, development and promotion of international standards in systems interworking has also been one of IRIA's main concerns.

JRC - European Communities Joint Research Centre, Ispra Establishment (Computing Centre - CETIS)

The main activity of CETIS is computing support to the research program of the Joint Research Centre of the Commission of the European Communities. At the same time it maintains several research projects particularly in the fields of energy, the environment, reactor safety and computer networks.

The JRC Ispra has been active in the EIN project since its conception, and is now taking part in EURONET in collaboration with the Commission in Luxembourg. The main area of interest is in higher level protocols, and in particular the problems of file transfer.

NPL - National Physical Laboratory

NPL, in addition to its role as the (UK) national standards laboratory, undertakes research in support of industry. Its involvement with the digital computer began in 1946 with the building of the ACE pilot model, one of the earliest stored program machines. In 1966 research in networks began with the building of the local packet switched network which now has more than 200 terminals. Research in the Computer Science Division of NPL includes network protocols, information systems, man-computer interaction (including speech), multiprocessor systems and network job control languages.

The work of the Division associated with EIN has included the design of the Network Control Centre, development of an X25 interface box in conjunction with the Executive Body, and interaction of EIN with the NPL local packet switched network and hence, through another gateway, with the UK Post Office EPSS network.

7. SECONDARY CENTRES

In addition to its Primary Centre a Signatory may nominate any number of Secondary Centres to be connected to the Primary Centre, and hence to the computer network, by national leased lines. Of the several EIN Secondary Centres, those that took part in the 1978 presentation (see below) are:

AERE - Atomic Energy Research Establishment

AERE Harwell is a research and development laboratory within the United Kingdom Atomic Energy Authority. Its interests span many aspects of the natural sciences as well as informatics. Informatics research is carried out in the Computer Science and Systems Division at Harwell. This involves several groups working in such fields as operational research, applied mathematics, information retrieval, real-time systems and networks.

The Networks Group is involved in the development of distributed processing techniques for the effective use of networks. In particular, it is active in areas related to the development of high level protocols for open networking such as protocol structure, verification and application-specific considerations. The group also contributes to national and international standardisation efforts.

CICG - Centre Interuniversitaire de Calcul de Grenoble

CICG, the CNRS (Centre National de la Recherche Scientifique) and the University Laboratory for Informatics and Applied Mathematics (IMAG), have been involved in computer network research since 1970. First studies started on the SOC Network and continued in 1972 on the Cyclades Network. Participation in EIN began in 1976.

This activity has taken two directions: distributed applications, and methods of connecting various systems (such as computers and terminals) to a network. The first covers mainly the areas of Network Job Control Language, and distributed data bases. The second has allowed CICG and IMAG to gain expertise in connecting computers to a variety of networks, using several solutions: internal adaptation, front-end processor or microprogrammed black box. Using these techniques, CICG operates a part of the Cyclades network to allow interconnection of its computers and terminals.

CILEA - Consorzion Interuniversitario Lombardo per Elaborazione Automatica

CILEA is the regional centre whose main activity is supplying computer services to the five universities in Lombardy, including the Politecnico di Milano. It has implemented and manages the SARA computer network for batch users, which connects UNIVAC, IBM, CDC and Honeywell machines. It is at present pursuing applied research and development in the field of data processing terminal concentrators, data bases, and computer system evaluation.

CSATA - Centro Studi e Applicazioni di Tecnologie Avanzate

Since 1969 CSATA has been active in three areas: education, services and applied research, with the main objective of transferring technology and know-how to representative bodies of the "Mezzogiorno" (such as local government, industries etc.). It operates in fact, primarily in Southern Italy, and its activities are mainly related to informatics.

CSATA has been working on EIN as a research project, under CREI's coordination, since 1977. Its main interests in networking are in the investigation of packet switching problems, distributed data bases (high level protocols) and Centre management problems on a network basis.

8. ASSOCIATED CENTRES

Recently some Signatories have nominated Associated Centres, not connected permanently to the network, but capable of access through the public switched telephone network to a number of Primary and Secondary Centres. The Associated Centres that joined in the 1978 presentation were:

GMD - Gesellschaft fuer Mathematik und Datenverarbeitung

GMD, founded in 1968 with headquarters near Bonn, is a large scale research institution financed by the Federal Republic of Germany and by the State of North Rhine-Westphalia. It pursues application-oriented basic research, applied research, and development in the field of data procesing. Long-range basic research, applied research, and development projects constitute a unity within GMD through which new findings in research are put into practice by means of development projects, which themselves indicate further areas of research and development. These functions include advisory activities and contract work, in particular for the public sector.

GMD especially studies application-oriented conceptual problems, thus contributing to further development of computer systems, their improvement, utilisation and application. In this way the research and development activities cover the whole range of hardware, software and applications, and their role in nation, government and society.

In 1973 the former Deutsches Rechenzentrum at Darmstadt became the GMD Institute for Teleprocessing. This is one of the eleven GMD Institutes, and is active in application-oriented research and development in the field of teleprocessing and distributed systems.

QZ - Stockholms Datamaskincentral foer forskning och hoegre utbildning

QZ is the main institution in Stockholm for higher education and research on large mainframes. In addition to its terminal network, it is connected to SCANNET, a Scandinavian packet switched network for distributed databases.

QZ participates in a computer network project on the problem of interfacing a closed user group to a public packet switched network using X25. The purpose of this project is to get a standard structure of the communication software, and especially a neat interface to the human user.

RSS - Raziskovalna Skupnost Slovenije

The Information Centre of the Research Community of Slovenia, RSS, was established in 1974 with the aim of providing information on research activity to all interested parties in the country.

Today the Information Centre is working on establishing computer-based information and documentation services in Slovenia in collaboration with other centres specialising in various fields (mechanical, electrical, medical, agricultural etc). Some applications are being maintained and further developed by the Information Centre itself.

Networks are of special interest due to the fact that on-line data base usage is essential for the efficient information services that are to be provided to the community in the near future. RSS is responsible for coordinating network research in Yugoslavia as far as EIN is concerned.

9. THE ROLE OF THE PARTICIPATING CENTRES

In many cases the experts engaged in the early activities of the project came from the participating centres, although some signatories without centres also provided experts for the various working parties. In parallel with the common activities, the various centres made their own plans for the installation of the network switching centres, the provision of communication links and so on. In addition, they began to consider the problem of interfacing their own

Barber D.L.A. - EIN - an example of cooperative research in Europe

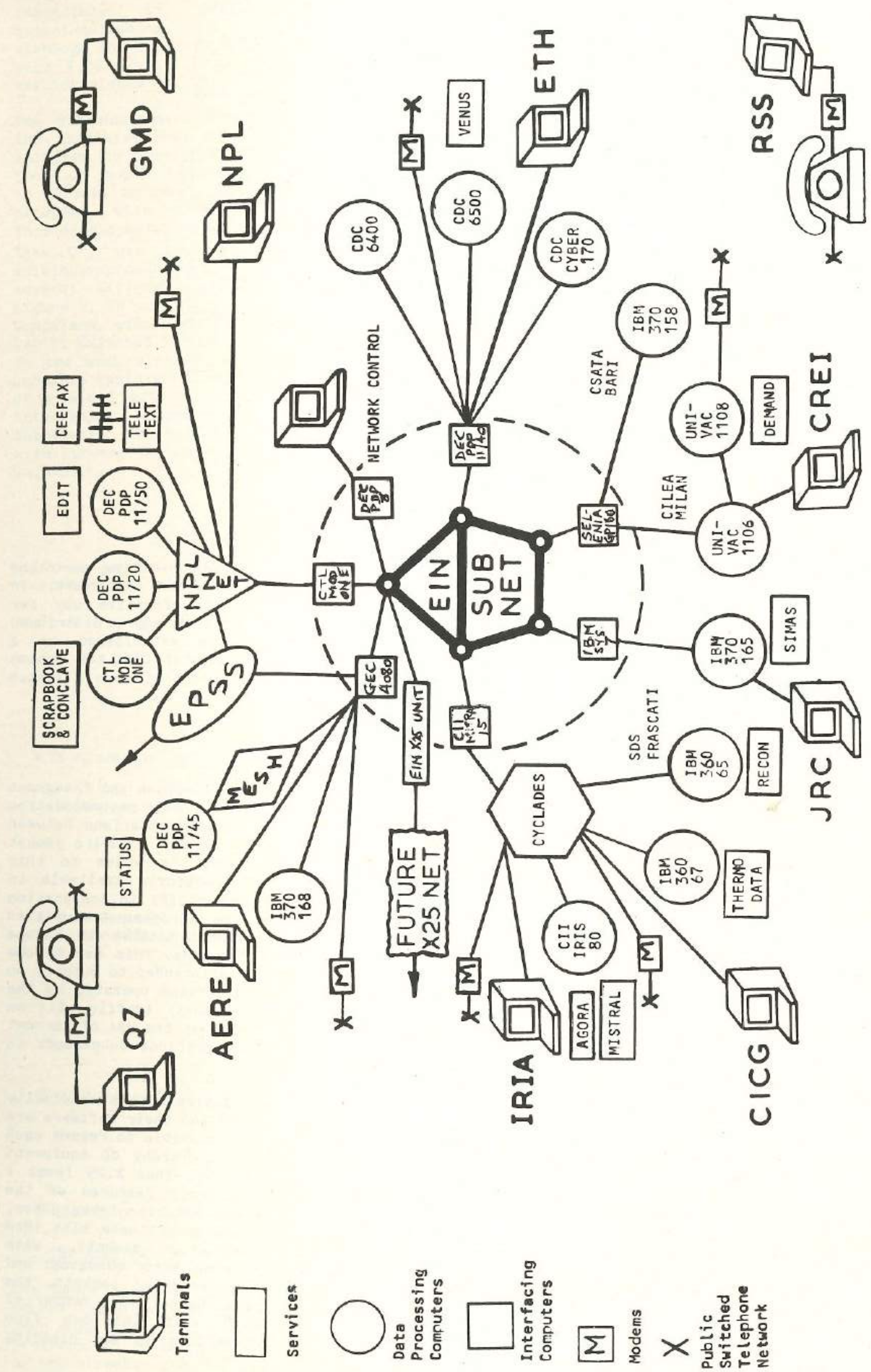


Figure 3: Some Systems Joined to EIN

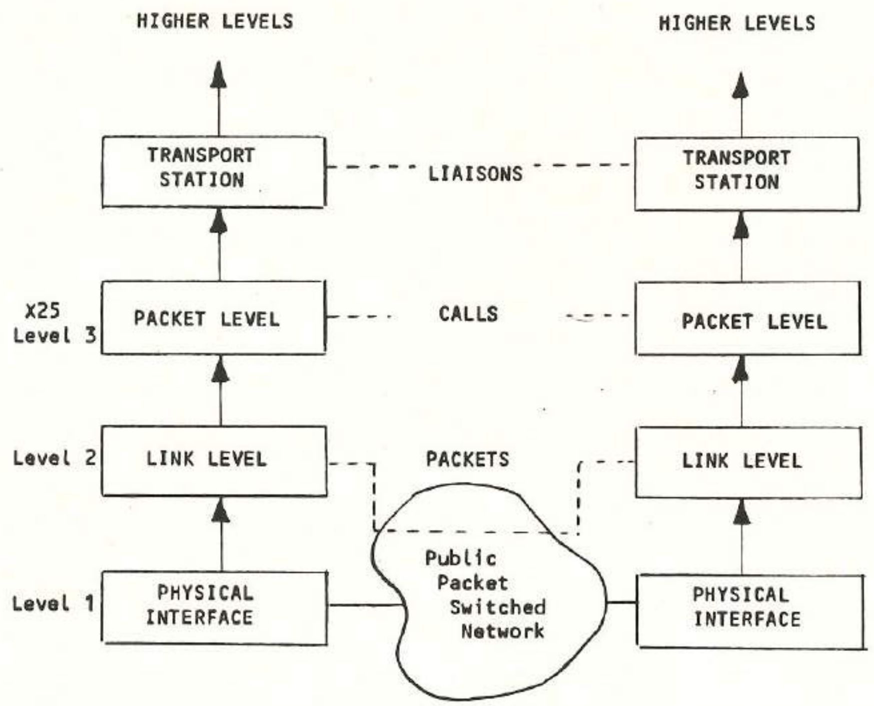


Fig 4: **Structure of the X25 Protocol**

computer systems, and in some cases networks, to the international subnetwork and various different solutions were adopted by the different centres.

At the time these decisions were made it was a strength of the project that different interfacing methods would be adopted, because this enabled a comparison of different methods to be made. As ideas evolved and experience was gained, most of the centres made some changes to their original plans and the present arrangement is that each centre uses a mini computer between the network and its own system. These mini computers form a ring of interfaces matching the centre's systems to the subnetwork. These systems are shown in Figure 3. A wide variety of different computer systems and networks is depicted in the figure and these illustrate the wealth of knowledge that has been gained by the participants during the conduct of the project.

Matching the complex computer systems of EIN to the communications network proved relatively straightforward, but adapting them to each other was, and indeed still is, a much more difficult task. The now generally accepted solution lies in the agreement of a number of levels of protocol or procedure carried out by software in or associated with these systems. This approach, which has been well described in the literature, is being discussed as a basis for international standardisation within CCITT and ISO. The pioneering work in this area conducted by the EIN centres has been a notable contribution to the subject [10].

By early 1978, the development of protocols within EIN had reached such a stage that a reasonable degree of interaction between Centres' systems was possible. The Management Committee therefore decided to give a public presentation of the activities of the participants and this was held on 5th April 1978. The ten centres mentioned above took part in a simultaneous demonstration of the network and its facilities [11]. A wealth of information was gained by those taking part and this led to some reappraisal of the work. In particular, the need for an effective way for communicating between centres through a distribued teleconferencing system was established as a result of using the experimental CONCLAVE scheme provided by NPL. The development of such a system in now taking place within EIN.

10. ADAPTATION TO EURONET

In 1976, the International Telephone and Telegraph Consultative Committee (CCITT) made recommendation X25, which specifies a standard interface between Subscriber's Computer systems and a public packet switching network, the public services to this standard are becoming increasingly available in many countries. In particular CEPT in conjunction with the Commission of the European Communities will provide an international network in Europe from the middle of 1979 onwards. This has become known as Euronet, and it is intended to support an information dissemination service operated by the Commission. However, third party traffic will be carried and it is expected that the use of Euronet will replace the EIN communications subnetwork in due course.

The X.25 specification illustrates the modularity with which modern computers and their software are designed; it has become fashionable to regard each module as a "level" in a hierarchy of equipment and processes (see Figure 4). Thus X.25 level 1 gives the electrical and logic features of the circuit by which an exchange of bits takes place; level 2 specifies the grouping of these bits into "frames" (each carrying one packet), with demarcation between frames and error checking; and level 3 deals with the structure of packets, the multiplexing into one real channel of a number of conceptual channels each with its own flow control, and rules for setting up and clearing virtual calls.

When the EIN subnetwork was specified in 1973, it was impossible to foresee the details of any future CCITT standard, and indeed it was impossible to predict when a public packet-switching service would be available, so although the EIN data link protocol is compatible with X.25 levels 1 and 2, which are based on earlier standards, it differs from level 3.

The EIN Management Committee therefore directed the Executive Body to develop an adaptor box, suitable for interposition between an EIN NSC and the X.25 public network, to allow the changeover to Euronet to take place using permanent virtual circuits, with minimum disturbance to Centres. This development, known for historical reasons as "Box C", was completed early in 1978 using multimicroprocessors, and has furnished ideas for several different adaptor boxes, as shown in Figure 5. In particular, "Box B" allows Subscriber Computers, either directly or via an NSC, to make use of switched virtual circuits, and indeed based on the work already done, it is now seen to be possible rapidly to design an adaptor box for any of a wide variety of practical requirements in interfacing computers to a communications subnetwork. The box which will be ued to connect with Euronet is known as an EMU-B (EIN Matching Unit-B).

have already been mentioned above, as has the part played by the EB in coordination of Centres' activities. Other tasks have been the representation of the project at public conferences, at CCITT and ISO meetings and in interactions with other projects.

In all these activities, a number of valuable lessons have been learnt about the management of an international cooperative research project, with distributed participants.

For the most part, cooperation between Centres has been satisfactory because, once each particular objective has been agreed, each Centre has been able to work independently to reach it, making its own decisions and using its own resources as required. This is in accord with the provisions of the original Agreement.

But with such a complex project a more detailed control of the work is often desirable, because the success of the whole project relies on the proper interworking of the systems of the individual participants. This has proved hard to achieve through the committee structure adopted for EIN.

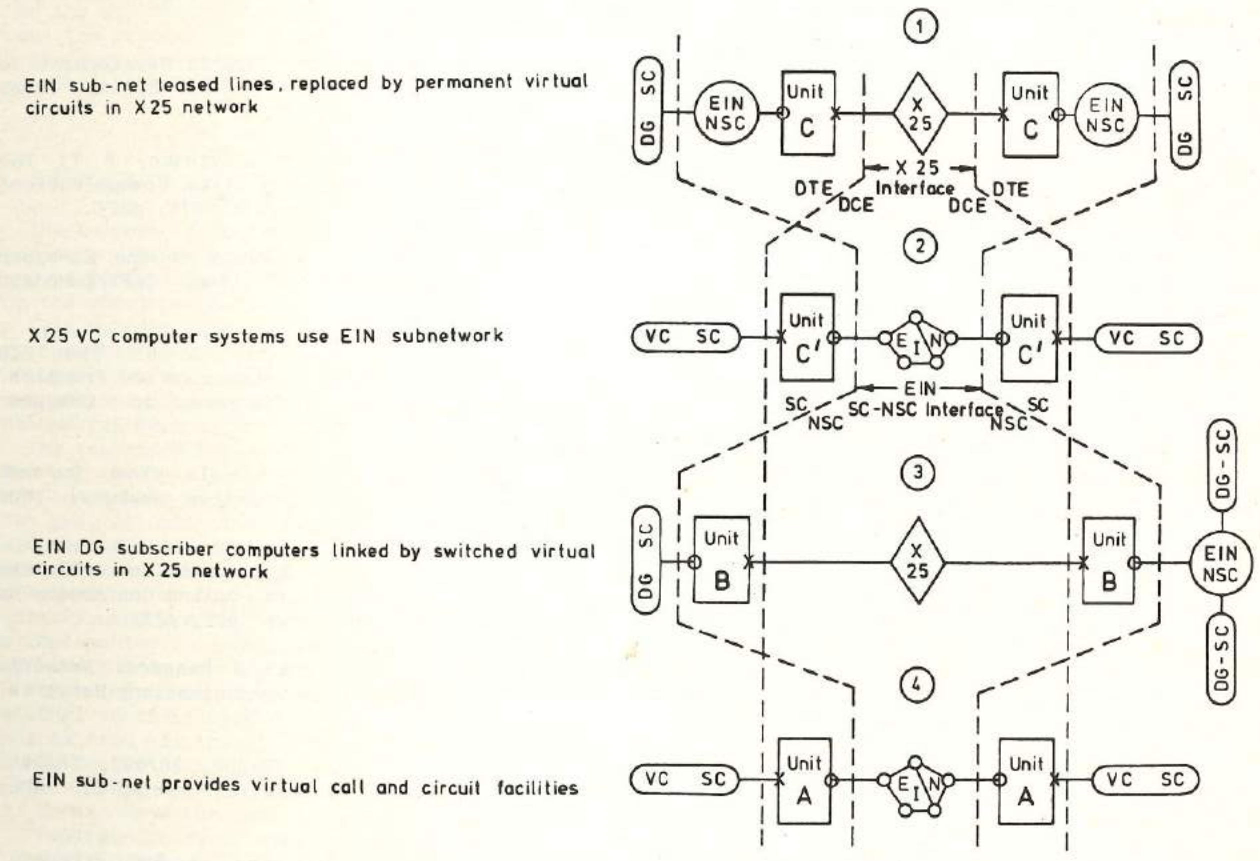


Figure 5 Some possible X25 Adapter Units

NOTE - All units have one datagram interface O and one virtual circuit interface X

11. THE ROLE OF THE EXECUTIVE BODY

The basic tasks of the Executive Body were, of course, laid down in the Agreement. But there have been many problems, that have arisen during the progress of the project that have demanded action by the Director and his assistants. Examples such as monitoring contracts and the development of EMU

As an example, the specification for the Transport Station protocol [12] has been implemented by each centre. But naturally, there are various ways in which this specification can be interpreted, and so separate implementations differ. Effective interworking in EIN proved not to be possible until the situation had been clariried and some changes made to some versions of the Transport

Station. But, even when this had been done secondary problems arose because most protocols may be implemented in a partial way, sufficiently to provide a basic service, but without the more sophisticated features; while all kinds of checking for protocol violations may be included in a comprehensive and robust implementation. With independent designs there is no easy way to assess their relative completeness and it is impossible to be sure that they will interwork under all future circumstances; furthermore, any changes that prove necessary cannot be introduced simultaneously throughout the network.

The Executive Body has endeavoured to coordinate the activities of centres in these kinds of tasks but without a more direct involvement in the work than was provided for by the Agreement this has proved a very taxing and onerous task. Even in non-technical areas the coordination of the actions of centres has been difficult when a really precise objective has been the aim.

A good example has been the organisation of the maintenance of the NSCs. Originally these were procured by centres using their own funds and by separate negotiations with the contractor. This was necessary because no mechanism was provided in the Agreement for the centres to be legally represented by the Management Committee and the Executive Body.

Nevertheless, discussions enabled the contracts to be made similar although they were legally independent. While this was clumsy but workable for the procurement of the NSCs, the same scheme was far less satisfactory for dealing with their maintenance. This is because they all have to interwork together within the framework of the subnetwork, which should preferably be treated as one complete system.

Unfortunately, from a legal point of view, no-one owns the subnetwork so no-one could negotiate a common maintenance contract for all centres. If, for example, the Executive Body had been able to place a simple contract for network maintenance, and then charge each centre accordingly, a much more satisfactory outcome would have been the result.

Problems of this kind have, unfortunately, bedevilled the EIN project since it began, so it is a great tribute to the goodwill and enthusiasm of all the participants that, for the most part, the initial objectives have, nevertheless, been satisfactorily achieved.

12. THE FUTURE

In the few years since EIN began, astonishing changes have occurred in the technological environment, brought about, in part, by the influence of the project itself. Public Packet Switching networks are becoming commonplace; a strong community of informed network users has been created by the signatories, and Europe is now in the forefront of research in this area. A recent development has been the advent of the microprocessor which is revolutionising and accelerating the application of teleinformatics systems. Against this background, the project will draw to a close, as the subnetwork is superseded by the use of Euronet.

The achievements of EIN have been generally acknowledged and there is great goodwill towards the idea of another such project; this time aimed at research into the applications of Teleinformatics Systems, rather than their design, as was the present project. It is too early to say what form a new project would take, or indeed, if one will be possible. But it is certainly the case that experience with the present project has revealed a wealth of problems in the use of networks that need to be solved if the maximum advantage is to be gained from the investment now going into the new public data networks.

The paper was received on May 11, 1979

REFERENCES

[1] Agreement on the Establishment of a European Informatics Network. Her Majesty's Stationery Office, London, Cmnd.4922, 1972

[2] Baran, P: On Distributed Communication Networks. IEE Trans. Communications Systems CS-12, March 1969, pp1-9

[3] Davies, D W: Communication Networks to Serve Rapid Response Computers. IFIP Congress, Invited Papers, 1968, p72

[4] Roberts, L G: Computer Network Development to Achieve Resource Sharing. AFIPS Spring Joint Computer Conference 1970, pp36, 543

[5] Scantlebury, R A and Wilkinson, P T: The National Physical Laboratory Data Communications Network. ICCC Conference August 1974, p223

[6] Public Data Networks, Plans of the European Telecommunications Administration. CEPT/Eurodata Foundation 1978

[7] Poncet, F and Repton, C S: The EIN Communications Subnetwork Principles and Practice. Third International Conference on Computer Communication 1976,p523

[8] Davies, G W P et al: The Euronet Telecommunication and Information Network. ICCC Conference 1978, p189

[9] Wilkinson, K: A Network Control Centre for the European Informatics Network. Online Conference on Data Communications Networks 1977, p271

[10] Deparis, M J F: EIN: A Research Network. Online Conference on Data Communications Networks, 1977, p347

[11] EIN Reports 78/001, 78/002, 78/003, 78/004, 78/005, European Informatics Network, NPL, Teddington 1978

[12] EIN Report 76/003: End to End Protocol. European Informatics Network, NPL, Teddington, 1976

The architecture of the software implemented by CREI for EIN

G.Le Moli[1], G.Affò[1], G.Andreoni[2], L.Baldini[3], A.Belloni[3], D.Boccia[1], M.Bozzetti[4], E.Cellamare[5], C.Consigli[3], E.Crippa[1], A.Faro[6], M.Floro[1], A.Gambaro[7], S.Palazzo[1,6], E.Radassao[3], E.Repossi[8], R.Restelli[7], V.Riccardi[5], P.Rossi[9], G.Scollo[1,6], P.Zupa[5].

Abstract. The paper outlines the characteristics of the software developed by the Italian Centres, coordinated by CREI (Centro Rete Europea Informatica), for the connection of their subscriber computers to the European Informatics Network (EIN).
In the beginning some considerations about the Italian participation to the research project are made and the functional architecture of the whole software is exposed in its broad lines.
Then the implementation of each component part is considered in a deeper detail to focus the problems and give an idea of the solutions that have been adopted.
In the end, some figures are given about, e.g., the effort necessary to develop each part, the CPU time absorbed by each of them, their core occupation, etc.

1. INTRODUCTION

The European Informatics Network (EIN)[(1)] is a computer network resulting from a research project of COST, namely Action 11. The Agreement for the development of this project was signed in 1971 by nine Signatories (The Commission of the European Communities, France, Italy, Norway, Portugal, Sweden, Switzerland, Yugoslavia, United Kingdom) and later on also by the Netherlands and the Federal Republic of Germany.

The technical goal of the project was the design and the development of an informatics network connecting Computer Centres all over Europe. That goal actually implied, on one hand, the development and management of a packet-switching communication network [(2)] (hereafter named subnetwork) and, on the other hand, the development of a distributed processing system architecture to be implemented in each mainframe connected to the subnetwork (computer network).

According to the Agreement, which became operational in 1972, a Technical Committee gathering experts from all the EIN Centres wrote the functional specifications of the communication subnetwork /3/ and tenders were called for a datagram subnetwork, with a packet ordering facility. The subnetwork, a description of which is in /4/, became operational in May 1976.

Meanwhile, as foreseen in the Agreement, the Centres started the definition and implementation of the higher level protocols for the connection of their mainframes (hereafter said Subscriber Computers, SC) to the network.

To this purpose, experts from all the EIN Centres periodically met to produce the specifications for the End-to-End and the higher level protocols.

Each EIN Centre has then implemented the network software in its own way /5/.

In Italy, when the project started, the mainframe of Politecnico di Milano (Univac 1108, now managed by CILEA) had to be connected; however, it was known that other SC's should have been connected to the network at a later time. So, the task of partecipating to the EIN project and implementing the Italian part of EIN was given to an ad hoc Centre, i.e., CREI (Centro Rete Europea Informatica, see Appendix).

CREI does not own any mainframe: it is responsible for the Italian node and for an interface processor, and develops software for the SC's connection to EIN on mainframes managed by other Organizations, by using them as a normal user.

1 Centro Rete Europea di Informatica (C.R.E.I.)- Politecnico di Milano, Piazza Leonardo da Vinci 7, Milano
2 CILEA, Milano
3 Società Italiana Telecomunicazioni Siemens, Milano
4 Olivetti, Ivrea
5 CSATA, Bari
6 Istituto Elettrotecnico della Facoltà di Ingegneria dell'Università di Catania, Catania
7 SIP, Direzione Generale, Roma
8 Now with Olivetti, Milano. At that time, with SIT Siemens, Milano
9 Istituto di Cibernetica della Facoltà di Fisica dell'Università di Milano, Milano

(1) A general description of the EIN network and of its objectives can be found in /1/ and /2/.

(2) Since at that time no public data network was available, EIN designed its own data network, with the purpose to move to a public one when available.

This situation has brought to the following constraints:
- no hardware modification was feasible in any SC;
- no modification to any operating system was allowed; even the use of JCL instructions was to be as close as possible to what each Centre allows to its normal users;
- the whole network software had to run on any SC as a normal user job, possibly with high running priority when available and necessary;
- the network software had to be as portable as possible;
- the network software had to be modular and easily modifiable, because new releases of the software had to be expected, e.g., to meet better definitions of the specifications and of the aims of the implementation, to expand services, to adopt new standards such as the X 25 recommentation, etc.

This approach resulted in three main consequences:

a) The minimization of the implementation costs was considered as a very important goal, while the optimization of the software was rather considered as the main goal of subsequent releases. In the first release provisions were then just kept to make future improvements easy to implement.

b) A minicomputer /6/ or a microcomputer /7/ had to be inserted between the NSC and any SC's to map the lower level of the HDLC procedure into another one which is more suitable for that specific SC.

c) The network software is mainly (say 90%) in FORTRAN IV and only small parts are in Assembler (e.g., real-time parts, handlers, executive request, etc.). The operability of the system that has been built is not affected by new releases of the operating systems and/or hardware modifications of the SC's, what actually happened outside the control of CREI.

This paper contains the general description of the architecture of the software that has been implemented in the Italian SC's of the EIN network. A much more detailed description can be obtained from CREI on request. A report on the way the implementation of the whole project has developed is in /8/.

Presently, the situation of the Italian SC's is as follows:

a) CILEA (Consorzio Interuniversitario Lombardo per l'Elaborazione Automatica, Milano, UNIVAC 1108 and UNIVAC 1106): the whole software described in this paper is fully operational.

b) CSATA (Centro Studi Applicazioni Tecnologie Avanzate, Bari, IBM 370/158): referring to fig. 1 (see also Sect. 2), only I/O-C and a few connections are still being implemented, while the SCM group is not planned, at the moment.

c) Istituto di Cibernetica dell'Università di Milano (Milano, HONEYWELL H 62): I_1 is operational, the TS group is being transferred.

d) CNUCE (Centro Nazionale Universitario di Calcolo Elettronico, Pisa, IBM 370/168): not yet connected. However, the same software as in CSATA will run; its trasfer is just a matter of one to two man-months, and will probably be done in short.

e) CINECA (Centro di Calcolo Elettronico Interuniversitario Italia Nord-Orientale, Bologna, CDC 6600):modalities of connection are now being studied.

The goal of portability of such a complex software has been achieved with a reasonable compromise between minimization of implementation efforts and satisfactory running costs: Tables 5.1 through 5.3 give the sizes of the various modules and an approximate evaluation of the time required for their development. Comments on these Tables follow in Sect. 5: here it is just worth pointing out that the total implementation effort has been of about eight man-years, while the effort required for the first transfer of the network software to a different mainframe has been nearly one fifth of that.

2. THE BASIC FUNCTIONS

The general architecture of the network software is shown in fig. 1. In this Section we describe in broad terms the tasks of the various boxes of fig. 1 and the way they interwork . Each box is actually a group of modules, which will be described in deeper details in Sect. 3.

The main modules are the following ones:

- I1 (I1 stands for Interlocutor 1) provides the network access, by performing the SC-NSC interface protocol.
  The SC-NSC is a Host-Node protocol that is particularly fit for the datagram service provided by the EIN communication subnetwork. For allowing the next connection to EURONET and other public data networks, I1 is going to provide also the standard CCITT X.25 interface.
- TS group (TS stands for Transport Station) performs the EIN End-to-End (EE) protocol and provides services on the top of the EE protocol such as echo, drop, access to the network for local users, access to local computing facilities for remote network users, etc.
- SCM group (SCM stands for Subnetwork Control Module) is a tool particularly fit for experimentation and measurements on the subnetwork. It can generate sequences of packets and is allowed to use all facilities supplied by the subnetwork (on the contrary, TS is only allowed to use Non-Delivery Diagnostic, Delivery Confirmation, Packet Ordering).
- BS group (BS stands for Batch Station) implements the experimental remote job entry service that CREI has developed for the Italian SC's of EIN with the twofold purpose of setting up in the shortest time a significant service for its network users and getting from it the maximum of knowledge to contribute to the EIN community[3].

[3] In Italy most people would use EIN as an access to remote job processing services and a facility like BS was then urgently needed. By the way, BTP (Bulk Transfer Protocol), FTP (File Transfer Protocol) and RJE (Remote Job Entry) are now in discussion within EIN technical groups: as soon as common specifications are agreed upon, CREI will implement these facilities accordingly.

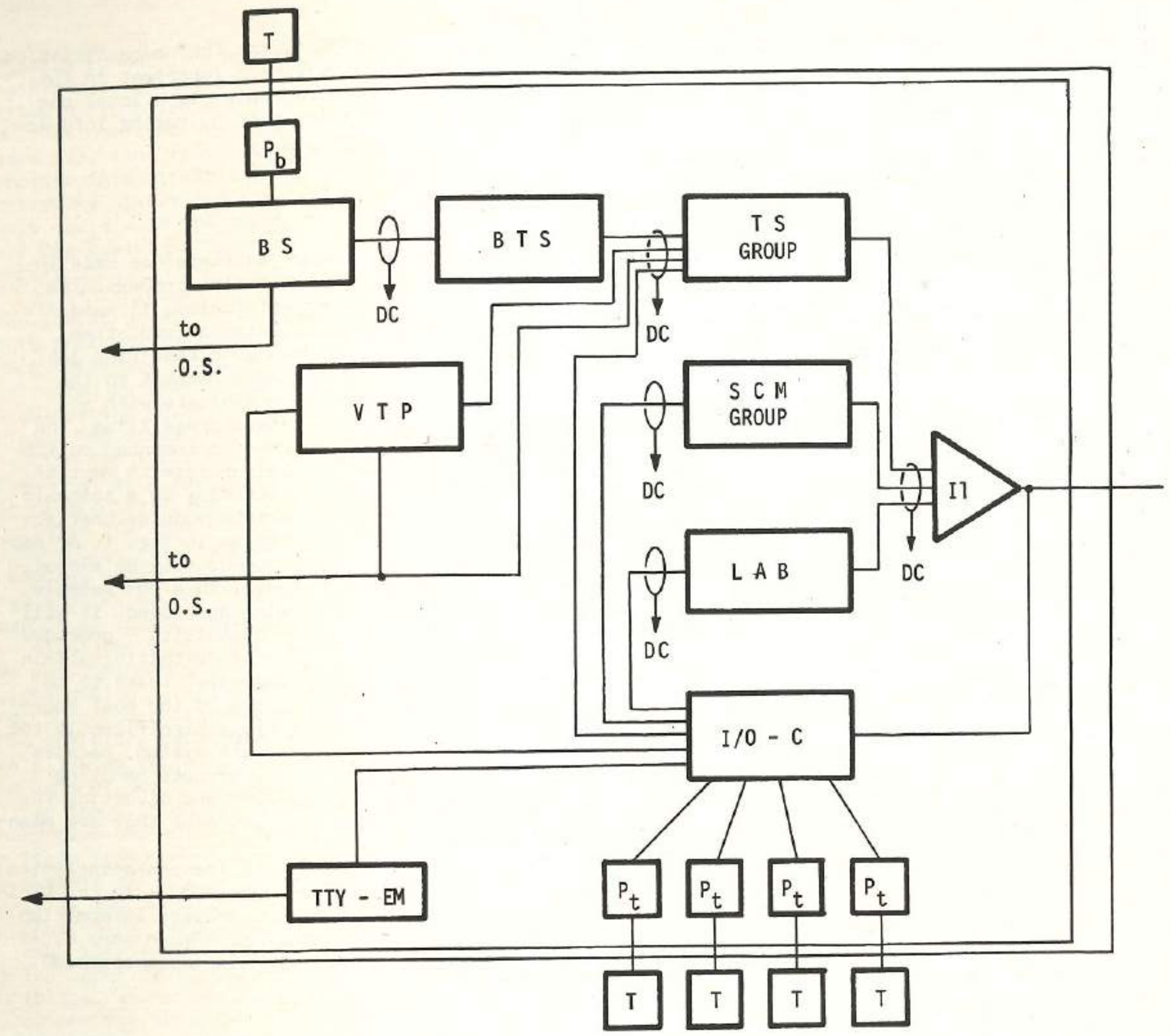


Fig. 1 : the overall architecture

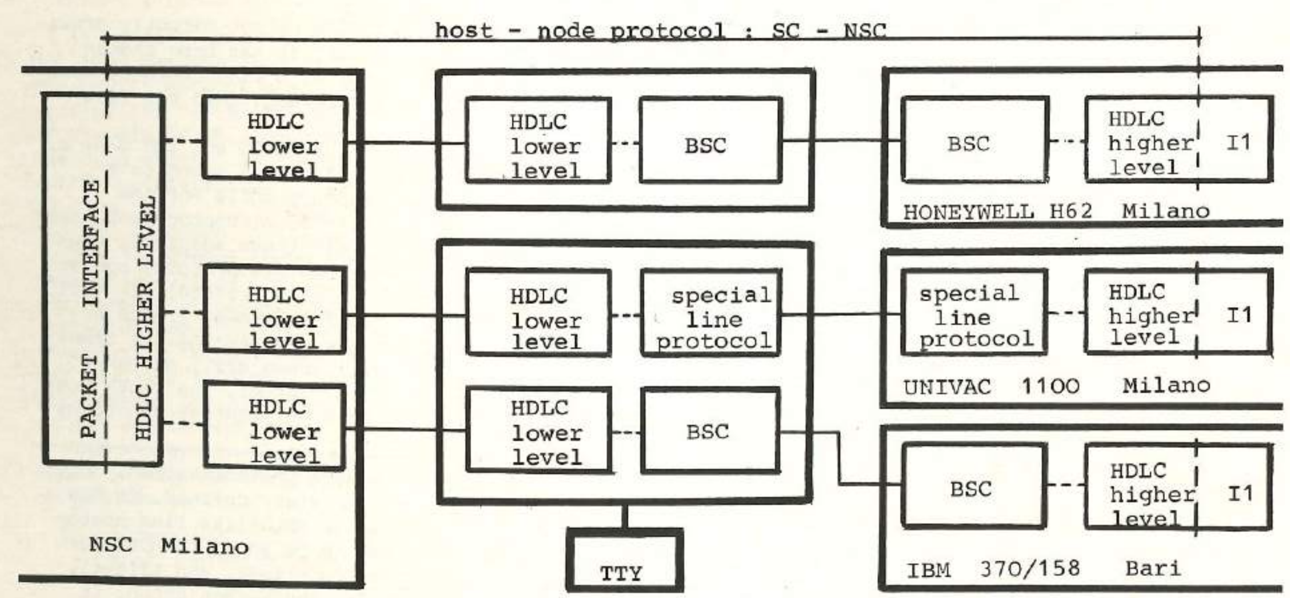


Fig. 2 : the line protocol adapters

As regards the local users, BS allows them either to route their printouts to remote terminals across the network or to get their jobs processed at remote sites (of course, they must implement a BS, too).
As to the remote users, BS instead provides them with a way to submit their jobs to the system and to get back the printouts as they were local users. In particular, BS deals with the transfer of large amounts of sequenced data across the network and implements on its own a bulk transfer facility.

- VT (VT stands for Virtual Terminal) performs the Virtual Terminal Protocol (VTP) and allows any terminal to communicate with any remote application(or terminal) regardless of the type of terminals and applications concerned with.
- TTY Emulator emulates a TTY and allows the SC to connect to a remote mainframe via a telephonic line by emulating any of its TTY - like terminals. The users in the network can thus reach any mainframe that provides a dialled service for TTY - like remote terminals.
- P_t is a simple process that interfaces I/O-C with local interactive terminals of the SC; presently it handles TTY's or TTY-like video terminals.
- P_b is a module that interfaces BS with local batch terminals.
- I/O-C (I/O-C stands for Input/Output Controller) is a multisocket switch that can supply a bidirectional connection between any of its ports. I/O-C is used to establish connections between any software modules, terminals and packet sources in the network.
- LAB is an empty box to which a subnetwork address is assigned.
It is used for debugging new network modules without affecting the normal operation of the network software, which is meanwhile active at a different (logical) SC address associated with the same physical SC.

In fig. 1 the lines connecting the different modules mean that between them full-duplex exchanges of information are possible; the lines do not imply a flow of control from one module to another: one should in fact assume that the modules virtually work in parallel. These full-duplex channels can be implemented in several ways, and it is not required that all of them be implemented in the same way. Presently, two ways are mainly used. Some channels are implemented as COMMON areas in the main store: this is used, e.g., between FORTRAN and/or ASSEMBLER routines belonging to the same job (for example, all SCM modules are subroutines of a unique job). Other channels are implemented as files shared among different jobs (for example, between I/O-C and all modules) In /9/ an approach that uses shared files and could be implemented on several machines is described: at present it has been implemented in the UNIVAC 1100. The choice of either method (and possibly of other ones) depends on the characteristics of the machine on which the system must run: in a machine with virtual storage the whole system can be implemented using the first method only, while in a machine with constraints for the size of real-time jobs the second method can be used, e.g., for connecting I1 to the modules at the upper level, so splitting into different jobs the parts that are strictly real-time and those that are not. Also the availability and the characteristics of segmentation facilities are important in this field. In any case, this problem is a local one and must be solved at any site by taking into account the local conditions.

3. THE IMPLEMENTATION

The whole CREI's network software has been implemented as a multitask parallel process. The strictly real-time parts (typically, I1 and TTY Emulator) have been implemented as real-time tasks (namely, as an independent real-time job) that go on asynchronously with respect to the rest of the software and communicate with it through shared areas in mass storage files. The rest of the software has been implemented on its turn as a multitask parallel process. A monitor (MAIN) either activates, according to a suitable priority hierarchy, the various modules that are comprised within the main boxes in fig. 1, or passivates itself for a few seconds (via an executive request) when running idle. In a new release of the network software being developed, it will be taken advantage of the possibility - provided by some operating systems - of definiting within the same user job many independent tasks to be processed in parallel directly by the host operating system. This would improve significantly the efficiency of the whole network system, because the host operating system interrupts handling would help in suspending tasks and diverting the control to the higher priority ones that are meanwhile ready for operation.

Here follow some details on the characteristics and the implementation of each module in the CREI s network architecture: for a complete information we however refer the reader to a much more detailed report describing the whole implementation, which is being prepared.

3.1 *I1*

The SC-NSC interface /4/ for the connection of a mainframe (SC) to an EIN node (NSC) is made of three layers :

- a link (modem) interface,
- an HDLC interface,
- a packet interface.

Because the Italian SC's do not directly provide the HDLC line protocol, it has been chosen to connect them by means of a line protocol adapter that maps the HDLC protocol into the line protocol used by each SC.

In the case of the UNIVAC 1100 and IBM 370, e.g., the line protocol adapter is based on a SELENIA GP 160 minicomputer, while for the HONEYWELL H 62 an INTEL 8080 microprocessor assembly is used (fig. 2). Both these adapters, however, do not operate a complete protocol conversion. When considering a line protocol, in fact, one can distinguish two independent groups of functions, the former ones concerning the transparent block (character, frame etc.) exchange, the latter ones the flow control, the block acknowledgment and all the other control functions[4].

[4] At the time of the node implementation, HDLC had not yet been completely defined. So EIN designed on its own an HDLC-like line protocol, which resulted to be equal to HDLC for the first groups of functions, and slightly different from it for the second group, in that it uses a channel mechanism instead of the HDLC window mechanism.

For the connection of the UNIVAC 1100 and IBM 370 mainframes the adapter only deals with the first group of functions; in the case of the HONEYWELL H 62 (since some functions of the second group need real-time responses and it is very hard to provide this directly from the H 62 due to its operating system characteristics) the adapter not only deals with the block exchange, but also with all the real-time functions comprised in the second group.

I1 (Interlocutor 1) is the software module at the lowest level in the CREI's network architecture that provides the packet interface and the HDLC interface (that part of course, that is not performed by the line protocol adapters).

Since the HDLC is a full-duplex protocol that claims for simultaneity at transmission and reception and short response times, I1 has been implemented as a real-time task. Moreover, in order to create real-time parallel activities and to optimize the core occupation and the CPU's time consumption, I1 has been written in the Assembler Language of each SC: I1 is typically one of the few modules of the CREI's network software that cannot be transported from a SC to another one.

After the EIN SC-NSC interface had been designed and implemented, the CCITT defined in its Recommendation X.25 a standard Host-Node protocol for SC's connections to public data networks. CREI has therefore developed also an X.25 interface to allow an early connection of the Italian SC's to Euronet and other public data networks.

In the X.25 design the following criteria have been obeyed:

- it is independent from both the SC's operating systems and the users' applications;
- it is modular and strictly respectful of the theoretical model of an interlocutor to allow easy maintenance and testing;
- its upper level interface and its structure follow, as farthest as possible, the interface and the structure of the TS (Sect. 3.2), which had already been implemented at the time of the X.25 design. The X.25 interface indeed covers a large number of the TS functions; in practice, the X.25 and the TS upper level interfaces are at the same level in the network architecture hierarchy, though they manage different information units.

At present, the X.25 interface runs on the UNIVAC 1100 mainframes, but is not yet available in the CREI's network architecture as a parallel service of the EIN SC-NSC interface. On the contrary of the EIN SC-NSC interface, the X.25 interface has been written in FORTRAN IV to ensure its widest portability.

3.2 *The TS group*

The 'TS group' shown in fig. 1 has the main task of providing the SC with the basic end-to-end communication services: however, some simple utilities have been added as built-in services in order to improve the interfacing to higher level modules, to supply simple aids to remote network users, or to collect information pertaining to the end-to-end protocol layer.

The 'TS group' is thus actually split- as shown in fig. 3 - in the following special purpose modules :

- TS (Transport Station) is the interlocutor that implements the EIN End-to-End protocol /10 /on the top of the datagram service provided by the EIN subnetwork. The EIN EE protocol is an extension (e.g., with the lettergram service /11/) of the CYCLADES one /12/, and has been raccomended by IFIP as a standard EE protocol /13/.
This EE protocol carries units of information, named letters, the maximum length of which may range up to 125 packets depending on the implementation. It provides its users with both a single-message-mode service, named 'lettergram', and a connection-mode service, named 'liaison'. It is completely symmetrical and can multiplex up to 2^{16} different ports. Letters longer than the maximum data field length of the subnetwork packets are fragmented into a suitable number of full length packets by the sending TS and are reassembled by the receiving TS; to this purpose a numbering scheme is used. An ACK/NACK scheme refers to whole letters: this means that a whole letter is retransmitted if a single fragment has been lost. TS is used and controlled through a suitable set of commands.
The data structure is a collection of records: each record contains all the information necessary to manage a user command or a message coming from a distant TS.
Records are created to manage ports and liaisons, as well as to control the transmission and the reception of letters (fragmentation and reassembling).
The Transport Station program /14/ consists in two sets of routines: one manages the End-to-End protocol, the other one handles the above mentioned data structure.
The TS program is periodically activated and is basically composed of four subprograms that are called sequentially; every time TS is activated, the program goes through the whole cycle.
The above mentioned subprograms are the following ones :
  - BRXS picks up the messages coming from the network;
  - TOUT scans the internal time outs and takes the proper recovery actions;
  - BCOM picks up the commands coming from the upper level users and starts the proper actions;
  - MRXS supervises the transmission of the letters.
- TS-IF (TS Interface) implements the interface between TS and its upper level users as specified by the EIN technical groups /15/ Through TS-IF several users have access simultaneously and without conflicts to the communication facilities provided by TS. In the CREI's implementation it is possible to distinguish between "internal" and "external" TS users: the former ones are processes that are physically implemented within the same job implementing TS; the latter ones are instead processes implemented in other jobs that share the SC's resources with the job implementing TS and its internal users and, usually, with other local users' jobs.
A same TS user can be connected either as an internal or as an external user without any difficulty other than, at generation, one must specify a suitable selection of the routines implementing the interfacing primitives.
Though physically implemented in different

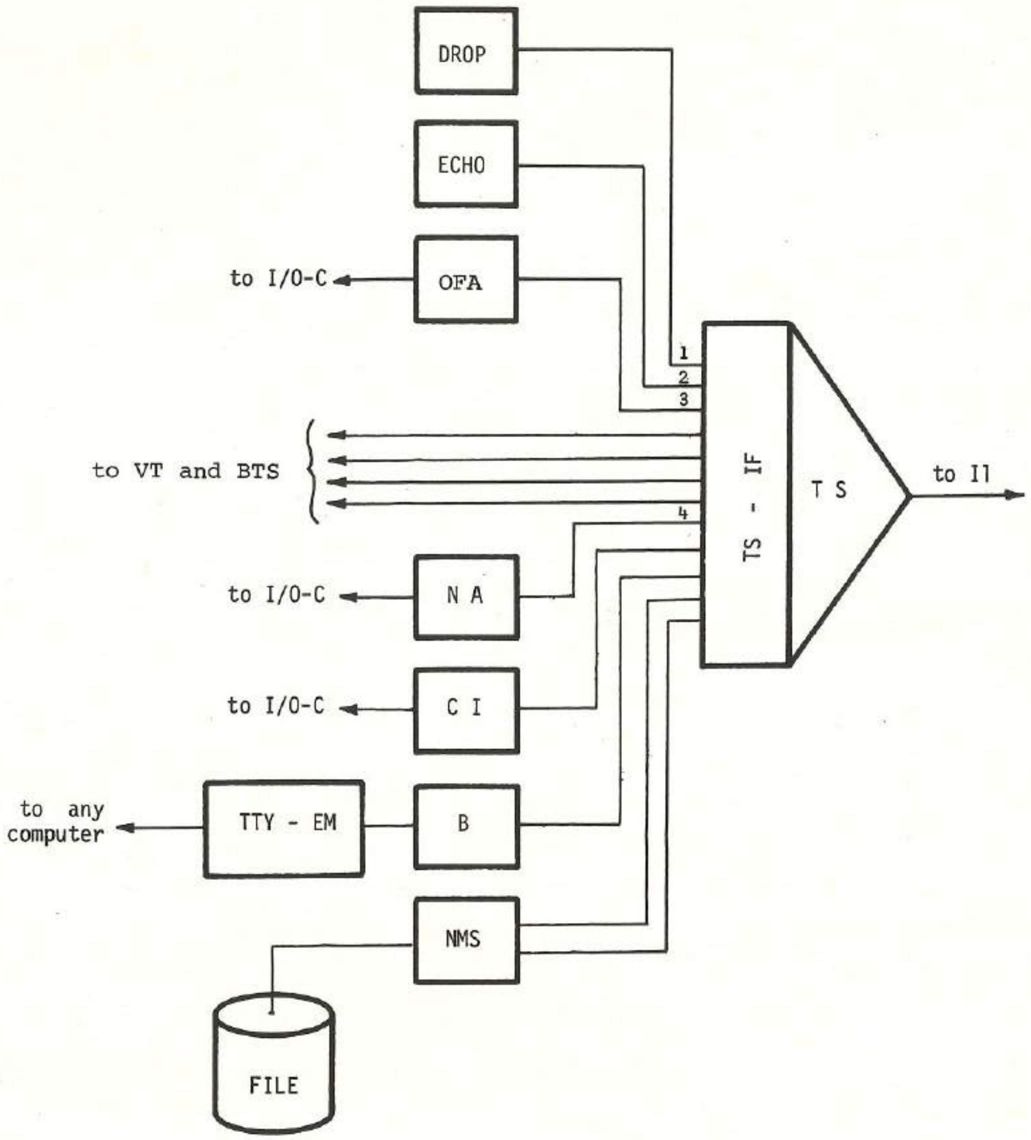


Fig. 3 : the TS group structure

EIN Network Measurement Center	gg/mm/aa	oo/mm					
START TIME :	oo/mm			PERIOD : mm MINUTES			
AERE	: oo/mm	CICG	: oo/mm	CREI/CSATA	: oo/mm		
CYCLADES	: oo/mm	ETH	: oo/mm	IRIA	: oo/mm		
JRC	: oo/mm	NCC	: oo/mm	NPL	: oo/mm		

Legenda:

gg/mm/aa = day/month/year

oo/mm = hour/minute

Fig. 4 : the TS availability prospect dispatched through the network

The architecture of the software implemented by CREI for EIN

ways for either cases, these primitives allow the users to interwork with TS in the same way. Of course, some limitations, typical of the external case, had to be kept for homogeneity also for the internal users (5). The most important of these limitations is that, upon completion of a command, TS-IF cannot awake an external user, interfere with his operation or inform him in any other way than by changing the content of a shared area (either in core of in a mass storage). It follows that, after issuing a command to TS (via the primitive PUTCOM), the user must care to check on its completion (via the primitive GETASW) and take the most appropriate action in function of the command status: typically, while the command is "in action" the user must schedule to check it again later on.

- DROP and ECHO[6] implement on the top of both the lettergram and the liaison services utilities that receive letters from remote sites and respectively discard them or bounce them back to the sender. These services are useful for testing as well as for checking on the availability at the TS level of remote SC's.

- OFA (Operator Facility)[6] allows the operators of the SC's to exchange letters among themselves.

- NA (Network Access) allows any local terminals to dispatch - and to receive - letters through the network, either in liaison or in lettergram mode, in order to have access to remote services.
  In this way many valuable services have been available networkwide while the network was still in its tuning phase.

- CI (Command Interpreter) allows experimenters at local terminals to issue commands to TS and to get back information on their status. This has been particularly useful for debugging TS and TS-IF.
  The module is essentially an improved interface to the human user: it takes command lines from the experimenter, which are written in an easy-to-use language, checks on their formal correctness and translates them into the proper commands to TS-IF; conversely, the answers from TS-IF are printed out in a way that makes them easy to understand.

- BRIDGE connects TS with the TTY emulator and allows remote users to have access via the network (in both lettergram and liaison modes) to the TTY emulator and -through this - to all remote services that are not directly connected to the network[7].

- NMS (Network Measurement Service)[8] cyclically solicits all remote Echo services both in lettergram and in liaison modes. All events are recorded onto a file, which can be processed off-line to produce reports on particular aspects: at present, e.g., the file is systematically processed to produce daily and monthly reports on the availability of the remote centres at the TS level. An online information service is also accomplished by this module: when solicited with an empty-text lettergram, it returns-via lettergram - a prospect on the availability of every TS, as it stems from the latest scan (fig. 4).

3.3 *The SCM group*

The "SCM group" /16/ is basically a tool to perform experiments on the subnetwork and control its behaviour. It is constituted of the three following modules (fig. 5):

- SCM (Subnetwork Control Module) generates (on request of the experimenter) artificial traffic in the shape of sequences of packets with pre-defined features as regards, e.g., their destination addresses, text lengths, interleaving characteristics, requested facilities, etc. Any of these features can be kept fixed through an experiment or varied according to either a function of time or a frequency distribution selected by the user.
  SCM can address, in particular, all the Virtual Subscriber Computer Processes (VSCP) implemented in any of the subnetwork nodes (namely Drop, Echo, Network Time, Internetworking) and use all the facilities of the SC-NSC Protocol (Non-Delivery Diagnostics, Delivery Confirmation, Trace)[9].
  Analogously to TS, SCM interfaces the upper level through ports that can be associated with both local and remote experimenters.

- LC (Local Controller) supervises the use of SCM by remote experimenters. In particular, it aims to assure that they make a fair use of SCM and that their experiments do not clash with others that could be in progress. To this purpose, LC submits the acceptance of remote users' requests for telecommanding SCM to a suitable qualification procedure: if a request is accepted, LC assigns a SCM port to the remote experimenter and notifies him a password.

- SCMI (SCM Interpreter) implements a user-friendly interface for both local and remote experimenters. It takes command lines to SCM written in an easy-to-use language, checks on their correctness and transforms them in-

---

(5) Of course, when necessary to improve the efficiency of the service, a special purpose interfacing to TS can be developed for internal users: this is actually being done together with the development of the second release of BS.

(6) Drop, Echo and Operator Facility have been agreed by the EIN technical groups as basic services to be provided by every implementation just as they were built-in facilities of TS.

(7) The BRIDGE can be used, e.g., to reach via EIN the ESA Documentation Service in Frascati (Roma) through any Italian SC's.

(8) An analogous service for the subnetwork has been developed at NPL-Teddington- as a feature of the Network Control Centre managed there. Co-operation with NPL is effective in correlating measures at different levels and better understanding phenomena.

(9) Most of such possibilities have been inhibited to the End-to-End protocol for obvious reasons.

to the format most suitable to SCM. Conversely, it transforms the information from SCM into a format that one can easily understand.

3.4 *The laboratory (LAB)*
The LAB is an empty address that is used for debugging any piece of software without disturbing the operations of the other ones, for implementing special experiments or testing programs, and so on. LAB is defined as a logical SC: the EIN node, in fact, has four ports for four physical connections to SC's (physical SC's); however, several logical SC's can be defined at the same port, i.e., in the same physical SC. The maximum number of logical SC's the node managed by CREI supports is 8 : octal 32, 33, 34 (presently assigned to the UNIVAC mainframes) 35, 36, 37 (presently assigned to the IBM mainframe), 40, 41 (presently assigned to the HONEYWELL machine) (10).

Let us now suppose that TS and SCM are operational at the addresses 32 and 33 respectively and that a new service, e.g., the rebound service, has to be tested before its integration with the rest of the network software being active at address 32. This will be done by defining a new TS and the rebound service at address 34, and by leaving this logical SC completely at the disposal of people working on the rebound service. The normal operation of the network software being active at addresses 32 and 33 is not disturbed.

Analogously, address 34 has been actually used in parallel with addresses 32 (TS and services) and 33 (SCM) to run ad hoc programs for special tests on the subnetwork /17/. The same address has been used, e.g., for the TTY Emulator at the EIN demonstration at EUROCON (Venice) to connect to EIN the ESA Documentation Service.

3.5 *The Input/Output Controller (I/O-C)*
The I/O-C allows a bidirectional exchange of messages (i.e., lines of characters) between its ports. From a master terminal, defined at generation time, one can command I/O-C to establish connections between any couples of the following elements:
- software components;
- terminals;
- packet sources in the network: they actually address their messages to port 9 of SCM, which in turn forwards them to I/O-C; these messages are processed by I/O-C as if they came from local terminals.

In this way a number of terminals in the network can use the software modules in any SC's, no matter with their physical location. Conflicts that may arise among terminals attempting to control the same piece of software in the same SC are settled by the master. The I/O-C provides two operational modes: command and transparent. In command mode (at present) either messages can be routed to other modules - typically to other local users or to the master terminal - or transparent connections between modules can be established through the master terminal intervention. Commands are checked for correctness and suitable diagnostics are sent to the end terminal. Messages that are not prefixed with a command code are automatically delivered to the master. In transparent mode a bidirectional connection is established and messages are directly exchanged between modules. Nevertheless, one can also route commands to I/O-C without leaving the transparent mode.

3.6 *The Virtual Terminal (VT)*
The VT is a standard terminal which real terminals (RT) are mapped onto and applications are adapted to. By means of this virtual approach, human users can communicate with applications in the network regardless to the particular RT and application being employed. Despite the internal organization and implementation of the VT is strictly related to the particular mainframe the VT is implemented on, the external behaviour of the VT and the way of using it are the same all over the network. The VT approach provides the network with both a "terminal independence" by means of the VT model and a "data independence" by means of a suitable data structure and the VTP.

The VT model is composed by :

- a virtual keyboard, which allows the human user to modify the store;
- a virtual presentation unit, which provides the user with an actual image of the store;
- a data store, which contains the data being exchanged and can be accessed by both partners: here the date structure and its contents are maintained;
- a process, which manages the store and provides at one side, the interface with the local user and, at the other side, the VTP for the colloquy with the remote application (or terminal): the process supervises the operation of all the VT modules involved in the colloquy between the user at the terminal and the application in the host mainframe.

Three basic VT classes have been defined, which the user can select to cover a large range of RT's:

a) the "scroll VT", which handles a mono-dimensional data structure, like any TTY-like terminal;
b) the "page VT", which handles a bi-dimensional data structure, like any VDU terminal;
c) the "data entry VT", which handles a bi-dimensional data structure with attributes for each element of the array.

At present, only the scroll VT and the page VT are available in EIN.

The communication between the VT and the application is maintained through a common data structure (the store) that both sides can read and/or modify by means of message units.

The rules for the exchange of message units are defined by the Virtual Terminal Protocol (VTP)

The VTP consists of four phases /18/:

I) the initialisation phase, in which one starts the colloquy and declares the data structure and the dialogue characteristics being used; in this phase the VT class is thus selected;
II) the data phase, in which data are exchanged;
III) the attention phase, which is used for synchronization between the partners and for interrupt and/or abort signalling: this phase

(10) These addresses have been assigned by CREI and will be reassigned when new SC's are connected.

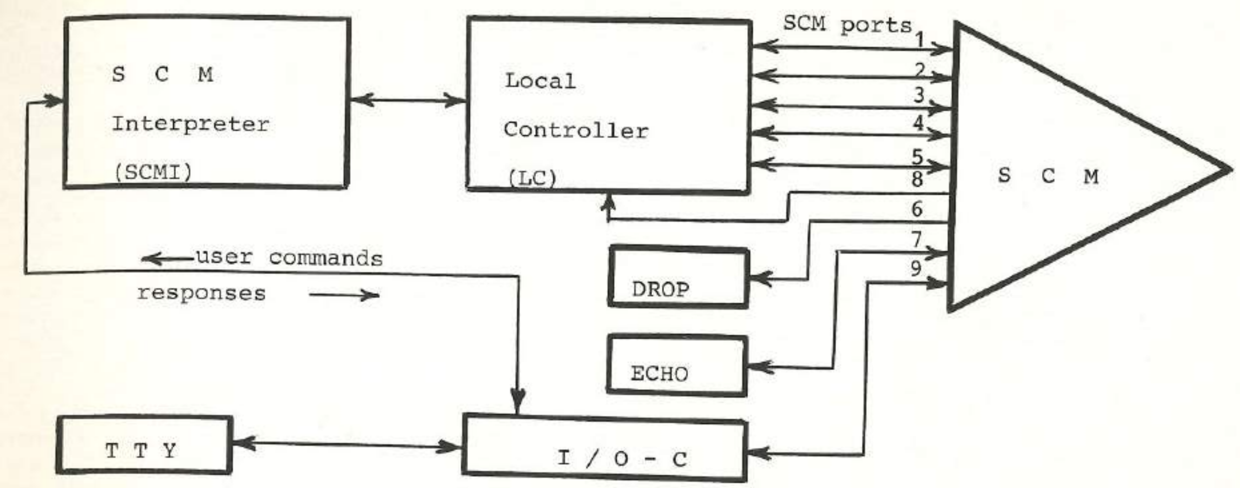


Fig. 5 : the SCM group structure

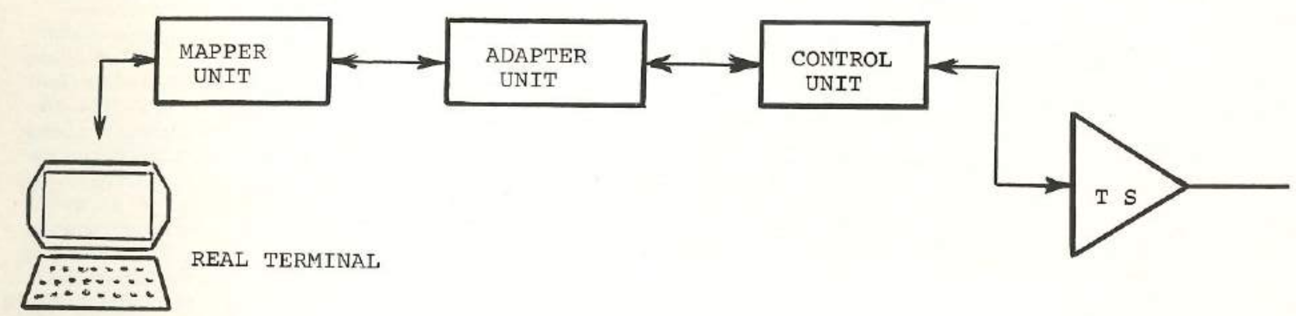


Fig. 6 : the Virtual Terminal structure

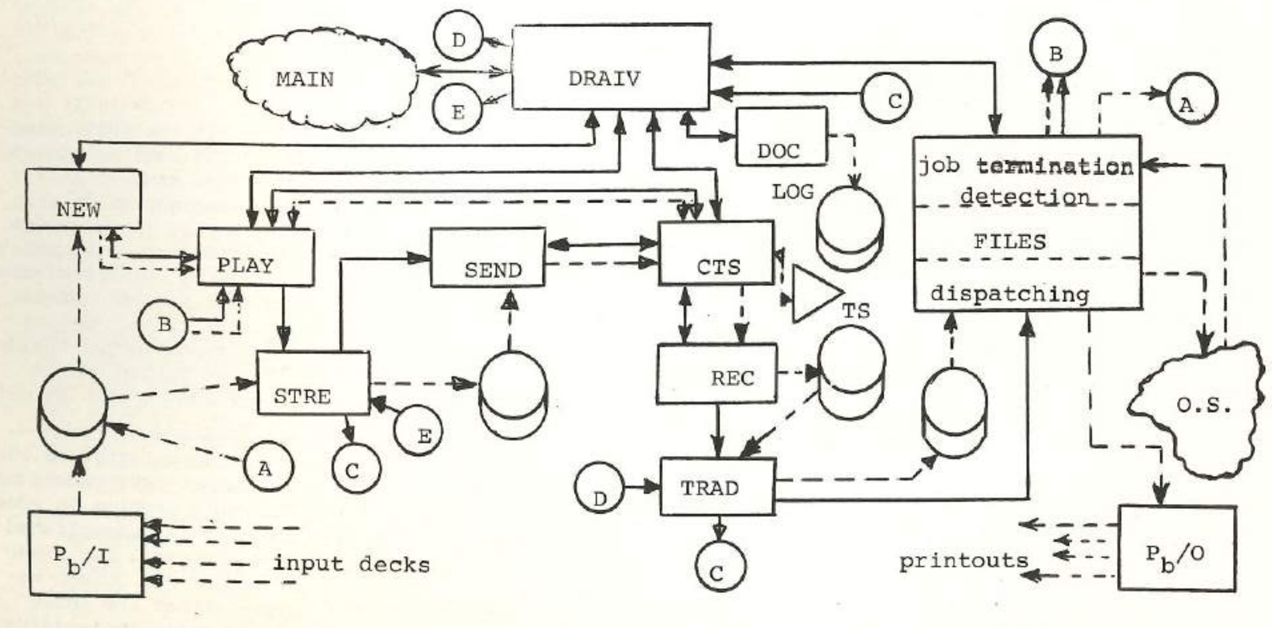


Fig. 7 : the Batch Station structure

consists of a symmetric exchange of attentions and marks;

IV) the end phase, which terminates the colloquy.

The functional structure of the VT Service implemented by CREI is shown in fig. 6. The basic software modules are :

- the Mapper Unit that interpretes the local user's messages and mappes them into messages for the Control Unit and conversely;
- the Adapter Unit that interfaces the Mapper and the Control Unit;
- the Control Unit that performs the Virtual Terminal Protocol and interfaces the Transport Service (TS) at the immediately lower level.

CREI has designed its VT Service to allow the simultaneous access to the network for up to ten terminals. In the Adapter, only the store active at any time is present, and it is used as a working area by the Control Unit. All remaining stores are recorded in mass storage files: for allowing the simultaneous operation of ten terminals, the Adapter manages the files including the non-active stores, and for every transaction it swaps the suitable stores for the Control Unit.

3.7 *The BS group*

The Batch Station has been implemented[11] as a parallel multitask process. In fact, on one hand, the structure strictly sequential of the whole software implementing BS, TS and its services wouldn't work if a routine at the TS user level retained the control while waiting, e.g., for the completion of a command to TS. On the other hand, BS has been designed to handle several jobs at a time; in particular, it can simultaneously transfer across the network up to p_1 decks - jobs or printouts - and process up to p_2 jobs. p_1 and p_2 are actually configuration parameters for each installation: in the UNIVAC 1100, e.g., they are presently both set to 25.

The functional architecture of BS can then be outlined as in fig. 7, where continuous lines stand for control paths and dashed lines stand for data exchange paths.

In practice, the operation of the whole BS is ruled out by a monitor (DRAIV), which is physically a subroutine of the MAIN ruling out TS and the network services, and in turn activates the BS's tasks that are ready for operation, or passivates itself by returning the control to MAIN when either BS is running idle or has consumed its time slot. DRAIV, in particular, cyclically checks on the completion of the BS's commands to TS (CTS), on the termination of the remote users' jobs being processed within the local system (FILES) and on the presence of local users' jobs to be forwarded to remote BS's (NEW)[12]. Local users' decks to remote sites are in general taken from a BS file onto which they have been moved - together with the necessary information as regards, e.g., their destination and nature (i.e., jobs to be remotely processed or simple printouts) - by means of catalogued procedures that are executed within local users' local jobs: these procedures corresponds to the input part of the box P_b in fig. 1.

(11) BS is presently operational in the UNIVAC 1100 mainframes of CILEA (Milano) and in the IBM 370/158 mainframe of CSATA (Bari) under respectively EXEC-8 and OS/VS2 operating systems.

(12) The first time it is activated, DRAIV also accounts for the initiation of the whole BS.

On the contrary, the schedulation of remote users' jobs as independent local jobs, the syntonization on their termination, the retrieval and diversion of their printouts are much more difficult points and require tricky solutions peculiar to each operating system[13]. To this purpose, it has proved useful to confine all of the BS interaction with the local OS within a unique module (FILES) that masks the characteristics of the local OS and makes a virtual operating system - independent of the installation - available to the other BS modules; FILES is then the only BS module that cannot be transported from one machine to another and must be developed for each installation.

The protocol used for communication among BS's /19/ is a simplified version of the bulk transfer protocol proposed in /20/ with the addition of what is necessary to manage a simple remote job entry service. The protocol foresees two logically separated communication channels, namely for the exchange of negotiation items and operation control information (service messages) and for the transfer of a suitable frame (stream) embedding the data - jobs or printouts - and the related control information. The lettergram service has been chosen for the service messages exchanges, while the liaison service -with the flow control option - is used for the streams exchanges. This separation between data and control information exchange channels is helpful when implementing BS, because it allows to keep completely separate - and at different priority levels - the module that deals with the control information (PLAY) and those that deal with the data transfer (REC, SEND, STRE, TRAD). In this way, each bulk transfer can be considered as a low priority task that is created and started when necessary and goes on independently till its end, while the management of control information is a higher priority permanent task that is activated whenever such information is to be processed: both tasks have access to TS-IF without any other switch in between.

In the end, a logbook (LOG) of all events concerning the Batch Station is produced (DOC) during BS operation. LOG can be processed (only off-line at present) to investigate malfunctions of the service and to produce statistics that could be useful to improve it.

4. COLLECTION OF STATISTICS

In a research project like EIN the collection of measures and statistics on the various aspects of the system is a very important aim. To this purpose, the problem has been approached in a systematic and general way: in fig. 1 the arrows labelled as "DC" (Data Collection) show the points where information on the system is collected. One could notice that there are two kinds of points of collection; indeed, two kinds of information are collected: the former one is collected inside a module (e.g. TS, SCM, I1), the latter one is collected at the interface between modules at different levels (e.g., between TS and its users, between I1 and TS, etc.).

The bulk of information that is collected and recorded during the network system normal operation is processed off-line by a set of special-purpose programs to produce significant statistics out of that (typically, distributions, means, standard deviations, tables, correlations, etc.)[14].

Information from the inside of a module leads to a better statistical understanding of the behaviour of the module in terms of queue lengths, delays, repetitions, lost messages, special occurrences (e.g., break downs over a connection), etc. Such statistics are useful in evaluating the adequacy of a protocol and of its implementation, in tuning-up time-outs, in dimensioning queues and buffers, etc.

A second kind of statistics concern what is going on at an interface in terms of traffic rates to/from services and/or ultimate destinations, distributions of messages in length and in time, analysis of favouritisms, if any, and so on. These statistics are helpful in discerning how far the network is adequate to the users' needs.

(13) Routing of printouts from remote sites to local terminals (i.e., the output part of the box P_b in fig. 1) isn't instead a problem in general, though in practice this is accomplished differently in different machines. Our experience, though limited to IBM and UNIVAC machines, furthermore suggests that the above said tricks - though differently implemented - could be logically the same in the different machines, since those goals are accomplished by means of a sequence of operations that could be usually done by a skillful normal user.

(14) Such values can be computed over variable periods of time, e.g., some hours, a day, a week, a month.

5. IMPLEMENTATION AND RUNNING RESOURCES REQUIREMENTS

This Section presents information concerning the size of the software and the implementation effort that has been required. The purpose of this Section is to give an idea of what implies the connection of a mainframe to a network like EIN; however, for a correct use of these data, one must take into account what has already been said in the introduction, namely the constraints and the objectives of the implementation, and particularly its nature of a research project.

For the sake of simplicity, data and tables are given with reference to the only implementation in the UNIVAC 1106. Further information is available on request.

Table 5.1 lists, for each module, the size of the code as it results from compilation, the size of the core required to store data that are local to the module, and the size of the core required to store data that are in common with other modules: in this case, however, the core has been arbitrarily assigned to one of the modules among which it is shared.

We have chosen to produce this information in this way because in our implementation data that are intrinsically necessary to run a module have been normally defined as local data, while data in common have been often used as a way to save time in modules interconnections. All data in all Tables have been derived from information supplied by the host operating system.

Besides the constraints already mentioned in the Introduction, a few more remarks should be considered for Table 5.1:

a) each module includes what is required to keep a complete record of any transaction and to collect statistics on throughput, occurrences of internal events and whatever else could be useful to monitor the behaviour of the module and its interaction with the other ones;

b) each module includes tests on the correctness of both the commands from the users and the messages from the network or other modules;

c) each module includes debugging aids as to the possibility of requesting dynamic printouts to follow what is going on, of requesting historical dumps, of recovering from some abnormal situations, etc.;

d) most of the modules can handle several users simultaneously; for example:
   - VT handles up to 10 real terminals;
   - BS handles up to 25 jobs and up to 25 simultaneous deck transfers across the network;
   - Network Access handles up to 10 users;
   - TS handles up to 100 ports, to which up to 25 liaisons may be connected, with a maximum of 5 liaisons per port; all ports can handle their 5 liaisons and the lettergram service in both directions (from/to the network) simultaneously; up to 50 letters incoming from the network can be simultaneously reassembled;
   - SCM handles up to 5 local and remote users.

e) No special aids from OS have been used to handle files and data banks.

Table 5.2 describes a few possible ways of grouping modules of Table 5.1 in various configurations, in order to measure the use of the CPU. The second Column indicates, for each configuration, the modules that have been grouped (not all the utilities are always included). Data are not anymore distinguished in local data and common data. PCT (Program Control Table) is the core associated by the host operating system to any user's job to keep there the information necessary to the system to run the job in a time-sharing environment. CPU is the ratio between the CPU time and the elapsed (solar) time for that job. I/O + CC/ER is the ratio between the time spent to carry out I/O operations or Executive Requests and the elapsed time: this ratio might be greater than one, since several I/O operations on different devices could be simultaneously in progress. In the end, the amount of core occupied by Fortran and/or system libraries is indicated for each job.

Measures have been made with the mainframe (in the configuration shown in Table 5.4) completely unloaded of other users'jobs and without any traffic from the network. Other measures have proved that this condition is meaningful, because measures can be repeated with the same resuls, what does not usually happen, instead, when the mainframe is busy. This condition is also the severest: in fact, the network software unnecessarily monopolizes the resources of the systems when this is unbusy, while in normal operation conditions the network software is bound by competition with other jobs. For this reasons we have verified a smaller consumption of resources when running the network software in the mainframe loaded with other users' jobs; however, also longer response time has been noticed in this case.

In spite of the language used (Fortran), one can notice that the use of the CPU is quite low, but for the job TS+Services: when comparing this job with the jobs TS+BS and TS+VT it results that the management of the services requires at least as much CPU time as TS alone does.

Finally, Table 5.3 gives an idea of the implementation effort that has been required. It is however to be considered that this software has been written by a group of junior programmers with the part time assistance of a senior analyst. The first column (Table 5.3) contains an approximate estimate of the time spent for the implementation of each module (in man-months): this time is in-

TABLE 5.1

MODULE	CODE	DATA		NOTES
		LOCAL	COMMON	
	WORDS	WORDS	WORDS	
I1	1367	2224	2557	4 users
TS	10618	2462	5531	100 ports, 25 liaisons (5 liaisons and datagram service available at each port)
TS SERVICES AND UTILITIES: DROP	71	27		
ECHO	96	29		
ECHO LIAISON	1033	826		
OP. FACILITY	573	378	90	
NETWORK ACC.	586	177		10 users
N.M.S.	1299	1004		polls all remote centres
BRIDGE	590	140		1 user
(DROP … BRIDGE total)	4248	2581		
TS - IF	1433	304	1951	
UTILITIES	3689	4972	1635	
SCM	7630	2890	3459	5 users
VT	4799	1147	2828	10 users
BS	4993	1255	1410	25 users
I/O-C	2527	3515	1444	25 users
TTY EMULATOR	921	1733	16	

TABLE 5.2

RUN		WORDS			%	
NAME	LOAD	CODE	DATA	PCT	I/O+CC/ER	CPU
TS + SERVICES	TS+TS SERVICES + TS-IF + + UTILITIES FOR-LIB	20121 5328	19106 2411			
		25449	21517	512	64.0	7.9
TS + BS	TS+BS+TS-IF+UTILITIES FOR-LIB	17845 5690	15565 2340			
		23535	17905	5632	113.0	3.4
TS + VT	TS+VT+TS-IF+UTILITIES FOR-LIB	20596 5328	20587 2411			
		25924	22998	512	61.7	4.6
SCM + SERVICES	SCM+SCM-IF+SERVICES FOR-LIB	7630 5178	6349 2146			
		12808	8495	512	10.4	0.6
I1	I1 SYS-LIB	1367 339	4781 147			
		1706	4928	3584	37.2	2.8
I/O-C	I/O-C SYS-LIB	2527 382	4959 162			
		2909	5121	512	19.3	0.3
TTY-EM	TTY EMULATOR SYS-LIB	921 320	1749 161			
		1241	1910	512	17.4	0.2

TABLE 5.3

MODULE	MAN-MONTHS	KWORDS/MAN-MONTH
I1	8	0.17
TS	30	0.35
TS SERVICES AND UTILITIES	12	0.78
SCM	12	0.63
VT	8	0.60
BS	12	0.42
I/O-C	6	0.42
TTY EMUL.	3	0.30
TOTAL	106	-

TABLE 5.4

1	CPU 1106
2	Memory Banks
	131 Kwords each 3 μs access time
8	8414 Disk Drives 65 ms access time
1	8433 Disk Drive 35 ms access time
1	CTMC + CTMIV
Operating System :	
	EXEC-8 V33 R3

clusive of the analysis, the coding and the debugging. The second column is the ratio between the size of the code and the time spent for its implementation: it is expressed in k-words of code per man-month. It can be seen that the experience acquired by the group has been largely useful in reducing the time required for the implementation of new modules of the same kind of those already developed: so, SCM has been implemented at a remarkable rate, thanks to the experience gained in the implementation of TS; analogoulsy, at the upper level, TS Services and VT largely benefit of the experience gained in the implementation of BS. It is also obvious that the global efficiency is increasing and that the analysis of most of the TS Services has been very trivial.

I1, I/O-C and TTY-EM, in the end, have been written directly in Assembler . Since Table 5.1 contains the size of the code produced by the compiler, entries in the second column of Table 5.3 are not homogeneous: entries corresponding to modules written in Assembler should be multiplied by three, which can be assumed as the mean of assembler instructions generated by the Fortran compiler for a Fortran instruction. It stems that, despite its complexity, I1 has been developed with the same productivity as TS. The extraordinarily high productivity corresponding to I/O-C and TTY-EM can be explained with the experience of the senior analyst who took care of their development, besides the fact that some utilities were already available from previous projects.

ACKNOWLEDGMENTS

The system here described is the result of the common work of all the authors. However, a susbset of them, namely A. Belloni, M. Bozzetti and E. Repossi, have moreover been charged with the task of editing this paper.

APPENDIX

In 1971 an Agreement was signed by the Commission of the European Communities and the Governments of Italy, France, Switzerland, United Kingdom, Yugoslavia, Norway, Portugal, Sweden and, more recently, the Netherlands and the Federal Republic of Germany. This Agreement started the execution of the Action 11 (known as COST 11) of the Program of Cost (Cooperation Européenne dans le Domain Scientifique et Technique) concerning the design, the implementation and the experimentation of the European Informatics Network (EIN).

The first five of the eleven Signatories mentioned above take an active part in the experiment and manage a node in their own territory: the five EIN nodes are in fact in Ispra (Euratom), Milan (CREI), Paris (IRIA), Zurich (ETH) and London (NPL). The other six Signatories take part in the project without managing nodes. However, they can install nodes at any time.

In Italy, the Ministry of the Scientific and Technological Research has been charged with the task of following the whole Italian part of the project: to this purpose, and ad hoc Centre has been created in 1976 as the result of a convention between the Ministry and the Politecnico di Milano. The name of this centre is C.R.E.I. (Centro Rete Europea di Informatica) [15]. CREI follows all the aspects of the project and of its implementation, from the technical ones to, according to the guidelines stated by the Ministry , the managerial and political ones.

The COST 11 project has some aims that are clearly stated in the Agreement and can be summarized by saying that the project has to produce know-how about teleinformatics to be spread out in each member State at the disposal of industries, government bodies, telecommunication carriers, research laboratories, for their own enhancement in the field. C.R.E.I. has also this task for Italy: to this purpose, C.R.E.I. has several cooperation agreements with bodies such as Ministries, carriers and industries and several young technicians have been detached from these bodies to work at C.R.E.I. in order to acquire knowledge and experience.

Director of C.R.E.I. is prof. G. Le Moli; the Scientific Committee of C.R.E.I., whose chairman is Prof. Luigi Dadda (Rector of Politecnico di Milano), is composed of members from :

- Ministero per il Coordinamento della Ricerca Scientifica e Tecnologica.
- Ministero delle Poste e Telecomunicazioni;
- Ministero della Pubblica Istruzione;
- Ministero dell'Industria;
- Ministero delle Partecipazioni Statali;
- Politecnico di Milano;

[15] CREI - Politecnico di Milano - Piazza Leonardo da Vinci n. 7, I20133, Milano - Italy tel. 02-296826, Telex 333467

- Consiglio Nazionale delle Ricerche (CNR);
- STET;
- Associazione Nazionale Industrie Elettrotecniche ed Elèttroniche (ANIE).

The paper was received on May 2, 1979

REFERENCES

/1/ D.L.A. Barber: A European Informatics Network: Achievements and Prospects - ICCC 76 Toronto, August 3-6, 1976

/2/ G. Le Moli : La rete Europea di Informatica COST 11 - Convegno AICA 1976, Milano. Also in Tecniche e Sistemi per Trasmissione Dati, Olivetti, 1977.

/3/ EIN: A Specification for a European Informatics Network, January 25, 1974 - Available from Director, D.L.A. Barber, National Physical Laboratory, Teddington, Middx. TW11 OLW, England

/4/ R. Poncet, J.B. Tucker: The Design of the Packet-switching Network for the EIN Project Eurocomp, London, September 23, 1975

/5/ M. Deparis et alii: The Implementation of an End-to-End Protocol by EIN Centres: a Survey and a Comparison - ICCC 76 Toronto, August 3-6, 1976.

/6/ G. Alvisi, G. Carrera, A. Gambaro, G. Le Moli: Structure of the Connection of Subscriber Computers to the EIN Node of Milan - EUROCON Conference, Venice, May 3-6, 1977.

/7/ E. Casero, D. Paleari, R. Polillo, G.P. Rossi: Interfacing a Honeywell Level 62 Host Computer to the EIN Communication Subnetwork with a Microcomputer System - Euromicro Journal, Vol. 4, Nr. 5, September 1978

/8/ G. Le Moli: Implementing the Software for Italian SC's of EIN Network - CONNET '77, CNT, Budapest, October 3-7, 1977

/9/ G. Andreoni: CREI Implementation of EIN Software: Files Structure - EIN/CREI/77/017.

/10/ EIN: End to End Protocol - EIN/76/003

/11/ G. Le Moli: Notes on Adopting the Transport Station for EIN Network - EIN/PM/750407/D1.

/12/ M. Elie, H. Zimmermann: Standard End-to-End Protocol for Heterogeneous Computer Networks - SCH 519, May 2, 1975, IRIA.

/13/ V. Cerf, A. McKenzie, R. Scantlebury, H. Zimmermann: Proposal for an Internetwork Ent-to-End Transport Protocol -IFIP WG 6.1-INWG General Note 96.1, 1975

/14/ A. Gambaro, E. Repossi, R. Restelli: CREI Implementation of EIN Software: The Transport Station - EIN/CREI/77/010.

/15/ J. Laws: Interface between a Subscriber and the Subscriber -to-Subscriber Protocol(TS)-EIN/CCG/JL/75905/D1 - EIN/CCG/JL/751020/D2.

/16/ A. Faro, G. Le Moli, E. Repossi: Specifications of the Subnetwork Control Module - EIN/CREI/77/007.

/17/ G. Kacin: Some Results of VSCP Tests Performed at CREI - EIN/CREI/77/004. Available from CREI.

/18/ P. Schicker, A. Duenki : The Virtual Terminal Definition - Computer Networks, Vol. 2, Nr. 6, December 1978.

/19/ A. Belloni, M. Bozzetti, G. Le Moli: A proposal for a Batch Service - EIN/CREI/76/004-5-6-7

/20/ A. Belloni, M. Bozzetti, G. Le Moli: A Proposal for a Bulk Transfer Protocol - Alta Frequenza, Vol. XLVII, Nr. 9, 417E-428E, September 1978 - IFIP WG6.1 - INWG General Note 196, 1979.

Gateway on higher level protocols

P. Schicker
BELL NORTHERN RESEARCH, OTTAWA
A. Duenki
UNION BANK OF SWITZERLAND, ZURICH

ABSTRACT
The design of the virtual terminal gateway in Zurich, mapping between the Zurich and CCG virtual terminal protocols, is presented, with an ensuing discussion on the experiences and insights gained in this area of network research.

1. INTRODUCTION

The European Informatics Network (EIN), originally known as COST Project 11, is a research community established by an international agreement. The main area of research is the definition and experimentation with network services and their protocols. After establishment of a datagram subnetwork linking several research institutes in Europe, the first achievement was the common definition of a transport service and its protocol.

Parallel to the implementation of this transport service the Center Coordination Group (CCG) set up a working party to define a virtual terminal protocol. This working party finalized a proposal by the end of 1976 [2]. Several centers, not satisfied with the achieved definition, set out to continue work in a smaller working group. This activity led to the definition of a second protocol, the Zurich virtual terminal protocol [1].

The European Informatics Network was then in the unfortunate situation that there existed two different protocols for the same kind of service with a number of centers implementing only one of the two protocols. Although this situation is not without benefit in a research environment where the two different approaches can be compared and experimented with in parallel, it left the research community unable to mutually communicate on the virtual terminal level.

In early 1978, the center at the Swiss Federal Institut of Technology in Zurich (ETH) decided that the attempt to map the two protocols into one another was a worthwile research project and established within a very short time a gateway function that performs the necessary mapping.

2. VIRTUAL TERMINAL PROTOCOLS

With the growth of the terminal market and the accompanying growth in data communication services it is not uncommon to find applications which are accessed by a wide variety of different terminal types, or terminals which are used to access a wide range of applications. The direct handling of a growing number of terminal types by every application becomes costly.

The virtual terminal is the definition of an abstract standardized terminal. The definition is that of the terminal's capabilities and behaviour which is expected by an application who communicates with this terminal. While the internal organization of the virtual terminals may vary, the terminal, as seen by the application, is well-defined and behaves consistently with the definition.

The virtual terminal definition anticipates the use of public data communication networks, especially networks of the packet switching type. To make good use of the new facilities, the virtual terminal is defined as a message terminal rather than a character terminal.

The virtual terminal protocol is the definition of the sequence and interrelation of information that is exchanged via the data communication service between the terminal and the application.

3. MAIN DIFFERENCES BETWEEN THE TWO VIRTUAL TERMINAL DEFINITIONS

The main differences between the two virtual terminal definitions are the following:

- The CCG virtual terminal defines a scroll mode terminal only (roughly equivalent to a Teletype) where the Zurich virtual terminal defines a terminal of the data entry type (roughly equivalent to e.g. an IBM 327Ø). The Zurich virtual terminal allows for subsets of which the smallest is the scroll mode subset (a particular terminal might implement only this subset); it is evident that a gateway can only map the scroll subset of the Zurich virtual terminal into the CCG virtual terminal and vice versa.

- The interrupt mechanism is very different. The CCG virtual terminal knows three different mechanisms with static re-initialization of the turn whereas the Zurich virtual terminal has a uniform mechanism for all kind of interrupts (more details are explained below when discussing the protocol differences).

- The Zurich virtual terminal provides for the possibility to transmit and receive sixteen different function ordinals (invoked e.g. by pressing a function key). In the CCG virtual terminal a concept in this generality is absent; however, the end of a message is qualified with one out of two codes.

The different terminal definitions are reflected in the protocols, as are the designers' different tastes. The main differences between the two virtual terminal protocols are as follows:

- The CCG virtual terminal protocol is heavily based on the EIN transport station and assumes an exchange of blocks. This is manifested in the definition of the protocol which allows the exchange of blocks only. Every block carries a block header with some control information. The Zurich virtual terminal protocol makes not necessarily use of such a structure of an underlying transport service and could be implemented as well on top of a transport service that provides for the transmission of a stream of information only.

- The negotiation for the setting of parameters in the virtual terminal takes two completely different approaches. The CCG virtual terminal protocol assumes a communication between terminal and application only and puts all the burden of parameter selection and setting on the application. The Zurich virtual terminal protocol anticipates the communication not only between terminal and application but also between pairs of terminals or applications. The nature of the parameters are different insofar as the CCG virtual terminal allows the setting of parameters such as 'Backspace' where the Zurich virtual terminal considers parameters of this type not worthy of special consideration especially as the virtual terminals must anyhow dispose of considerable intelligence to cope with the message aspect of the virtual terminals.

- The Zurich virtual terminal dialogue is organized into four different phases: the negotiation phase, the data phase, the attention phase and the end phase. The CCG virtual terminal knows only two phases: negotiation and data. The attention phase of the Zurich virtual terminal dialogue allows the more precise definition of interrupt sequences, the end phase provides for an orderly termination of the dialogue.

- The CCG virtual terminal protocol defines three different type of interrupts: asynchronous signals (on interrupt channel), the clear mechanism (synchronized on interrupt and data channel) and the please mechanism (on data channel only). The Zurich virtual terminal protocol defines only one type of interrupt: the synchronized symmetric attention mechanism (on interrupt and data channel). (The interrupt and data channel are services offered by the transport service; in EIN they are realized by telegrams and letters).

Fortunately, there are many similarities between the two virtual terminals. Without those a gateway between the two protocols would not have been possible.

4. THE MAPPING

This section assumes that the reader is familiar with the two virtual terminal protocols. We suggest that readers not familiar with the protocols skip directly to sections 5 and 6 where we discuss the addressing problem and our experiences.

The gateway is so designed that every assistance is given to provide as complete a mapping as possible. Thus, for example, the gateway resolves most of the cursor/printhead positioning capability conflicts between the protocols by accepting those restrictions in its negotiations with the CCG virtual terminal or application for which it (the gateway) can compensate.

Such intervention is possible only where there is no signifigant logical difference between elements of the two protocols, and where a clear and unambiguous translation can be defined. The gateway cannot compensate for deficiencies in a protocol. This means that only a subset of a protocol can be seen through a gateway; e.g., the attention ordinals 3 to 7 and the functions 1 to 15 of the Zurich protocol have no equivalent in the CCG protocol and thus 'disappear' through the gateway. The user of a gateway must take into account the resulting limitations.

4.1. Negotiation Phase

Because the CCG virtual terminal protocol is non-symmetric, an assumption must be made about the nature of the CCG virtual terminal or application (entity) with which the gateway converses. Therefore, the gateway assumes that a call to a CCG entity is a call to an application, a call from a CCG entity (i.e., a call directed to a Zurich entity) is a call from a CCG virtual terminal. This assumption affects the use of the negotiation items. A CCG application cannot establish a communication through the gateway, nor can a Zurich virtual terminal or application establish a communication through the gateway to a CCG virtual terminal. On the other hand, a CCG virtual terminal can set up a communication to a Zurich virtual terminal or a Zurich application can set up a communication to a CCG application through the gateway.

Negotiation through the gateway will be successful except where:

- one process does not offer Scroll mode. The gateway maps only this mode and forces the negotiations to this option.

- the CCG process demands unlimited values for the x co-ordinate. This concept has no equivalent in the Zurich protocol.

The gateway ignores the overprint option of the CCG protocol.

4.2. Data Phase

<u>Zurich</u>		<u>CCG</u>
Text	<-->	Text
Positioning note 1]	<-->	Positioning [see
F(Ø)+EOM	<-->	EOM(2) [see note 2]
F(1)...F(15)	-->	-
EOM	<-->	EOM(1) [see note 2]
Q-CATTR	<-->	HIDE
Transparent	-->	-

Note 1:
A free hand is used to achieve the desired positioning. E.g., if a CCG process has negotiated CR but no backspace, then a positioning request from the Zurich process which requires backwards positioning is translated by the gateway into a CR and a differential forward.

Note 2:
An F(Ø) [D-FCT item with a zero in the item data] issued by a Zurich process is not immediately forwarded by the gateway; rather, the gateway waits for the subsequent EOM from the Zurich process and then forwards the translated message (which may be empty) with an end of message qualifier of two [EOM(2)] ratherr than one [EOM(1)].

4.3. Attention Phase

<u>Zurich</u>		<u>CCG</u>
A(Ø)	-->	Clear
A(1)	<-->	A(1)
A(2)	<-->	A(2)
A(3)...A(7)	-->	-
Resume	<-->	Please [see note 5]
Purge	<-->	Clear

Note 3:
For all synchronized attentions, that is all Zurich attentions and the CCG clear, the gateway purges incoming information until the synchronizing mark is encountered. This is in keeping with the free interpretation which an application may put on attention ordinals and resolves problems introduced by the dual pipelines.

Note 4:
A CCG initiated attention which results in an attention being forwarded from the gateway to the Zurich entity will, of course, evoke a response attention from the Zurich entity at least as far as the gateway. Response ordinals of A(1) and A(2) are forwarded to the CCG entity. Response ordinals of A(Ø) and Purge will, if the original CCG attention was not a CLEAR, cause the gateway to initiate a CLEAR scenario with the CCG entity.

Note 5:
Resume and Please are translated and forwarded to the gateway only if the mode is alternate and the turn is not with the requestor.

4.4. End Phase

<u>Zurich</u>		<u>CCG</u>
EOD	-->	-

5. ADDRESSING THROUGH THE GATEWAY

The gateway mapping between the two virtual terminal protocols is an intermediate service and not an end service, i.e. dialogues are never done with the gateway but rather through the gateway. A user wishing to converse through the gateway needs to specify two addresses, first the gateway and second the service or terminal he wants to reach through the gateway. This problem can be resolved via the concept of source routing as described in [3]: both addresses are stated by the entity which establishes the communication.

Unfortunately, there exists no possibility in the EIN transport service to specify two destination addresses nor do the two virtual terminal protocols allow for a dialogue to exchange further address information prior to the negotiation phase. On the other hand, the negotiaion must be conducted in both virtual terminal protocols simultaneously such that the gateway can mediate the negotiations.

The solution adapted is to provide at the site of the gateway a range of network addresses each specifying the gateway. Associated with each of these network addresses is a service network address to which the gateway will establish a communication whenever it is called via the corresponding network address. The advantage of this approach is that a user who is for example employing a CCG virtual terminal and wishing to access a service which expects a Zurich virtual terminal needs to know only the specific network address at the gateway site in order to access the service. On the other hand, this solution is very static and services not forseen in the gateway´s tables can not be accessed through the gateway.

To overcome this static and rather unflexible addressing problem the gateway maintains a flexible table for additional address pairs. Anybody in the network can update this table by sending a lettergram to a particular address. This lettergram must specify a source address (i.e. the address of the virtual terminal which wants to establish a dialogue through the gateway) and a destination address (specifying the service to be reached through the gateway). A subsequent establishment of a communication (liaison) between the source address and the gateway address mentioned above will cause the gateway to establish the other conversation to the specified destination address.

6. EXPERIENCE

This gateway mapping two different virtual terminal protocols into each other has only been possible because the two protocols are providing similar services. In addition, the remote ressemblance of the two protocol structures made the implementation of the gateway easier.

The gateway was implemented at the Swiss Federal Institut of Technology in Zurich (ETH), but unfortunately, the ETH ceased operation as a center on the European Informatics Network before the gateway could be used as a routine service by other centers. It was implemented on a PDP-11/40 and ran under the ANNOS operating system [6]. The core requirements were about 2000 (decimal) 16-bit words of code with each instance of the gateway requiring an additional block of 256 words for variable and buffer storage.

The most difficulties in implementing the gateway were found in mapping the interrupt and attention mechanisms. The very different treatment of this concept in the two virtual terminals and their protocols caused some serious problems for the mapping of the two protocols. In addition, the proper handling of the termination of the two dialogues of the gateway also needed some careful analysis. Nevertheless, our and another center´s tests suggested that we had provided an adequate and usable mapping between the two protocols.

Incidentaly, this gateway is the second virtual terminal protocol mapping which we have implemented: the Zurich virtual terminal protocol was also mapped into the VENUS terminal protocol (VENUS is the interactive system on our Cyber machines to which the PDP-11 is the network front-end [7]). It is interesting to observe that the problems of this first virtual terminal protocol mapping implementation also lay in the treatment of interrupts and termination.

We consider these mapping experiences more than worth their effort as it gave us a very deep understanding of the nature and interrelation of protocols. As a side effect, we also learned that the current state of the art of protocol description (English prose) is far from being the optimum and ways must be found in the future to define protocols in a more formal way.

The paper was received on January 22, 1979

REFERENCES

[1] The Virtual Terminal Definition, P. Schicker & A. Duenki, Computer Networks, vol 2 no 6, December 1978

[2] Proposal for a Scroll Mode Virtual Terminal, H. Zimmermann & P. Schicker (Editors), Center Coordination Group of the European Informatics Network (CCG), EIN/CCG/77 02, January 1977

[3] Source Routing in Computer Networks, C. Sunshine, Computer Communication Review, vol 7 no 1, January 1977

[4] Data-Entry Virtual Terminal Protocol for EURONET, Commission of the European Communities, VTP-D/1, January 1977

[5] Symmetry and Attention Handling: Comments on a Virtual Terminal, A. Duenki & P. Schicker, Computer Communication Review, vol 7 no 3, July 1977

[6] Network Interface via a Front-End, A. Duenki & P. Schicker, Minicomputer Forum, Uxbridge, 1976

[7] VENUS, An Interactive Subsystem, P. Schicker (Ph.D. Thesis), Swiss Federal Institute of Technology, May 1976

Host-terminal connection techniques in networks

R. P. J. Winsborrow
Computer Science and Systems Division, AERE HARWELL, Oxfordshire

A. K. Duenki
SWISS FEDERAL INSTITUTE OF TECHNOLOGY (ETH) ZURICH

Abstract. There has been considerable interest in the definition and use of virtual terminal protocols within the European Informatics Network community. This has resulted in the provision of a range of network services and terminal access methods. In this paper the design of terminal concentrator and service access facilities at Harwell and Zurich are presented and compared. Problems encountered in designing terminal protocol gateways are also discussed.

1. INTRODUCTION

Understandably great emphasis has been placed on the development of effective tools enabling interworking between terminals and services on computer networks. Such facilities are a prerequisite in a research network like the European Informatics Network (EIN) for communication between establishments pursuing cooperative programmes of research. In a commercial environment the need for facilities to support remote terminal access to mainframes is also evident. Within the EIN community considerable effort has been devoted to the development of terminal handling protocols for these applications.

The terminal handling protocol developments within EIN have been based on the virtual terminal approach [1, 2]. These protocols assume the presence of a reliable transport service [3] providing end-to-end communications functions. They define a device-independant control language for an abstract or virtual terminal which contain many of the features found in present-day terminals. By providing device independence interworking between terminals is possible without difficulty; however communication between terminals and services must also take into account the data structures with which services interact with terminals. This latter problem is not handled by current protocols.

In this paper, rather than discuss terminal protocols per se, we have chosen to describe techniques which can be used for the implementation of these protocols. The paper is based on our experiences of providing interactive services on EIN. The implication of this experience for the design of terminal handling protocols is described elsewhere [4]. Here we are describing techniques which have been used at ETH Zurich and Harwell for connection of terminals and services to EIN, primarily using the protocol described in [2]. Arising from these experiences we then discuss some of the more general interfacing problems which can occur.

2. ETH TERMINAL SYSTEMS

2.1 *General*

A DEC PDP 11/40 mini connects the ETH-Zurich Control Data multimainframe system to the EIN packet switching subnetwork. This communication processor performs the lower level line protocols, the transport service, interface to the CDC mainframe, as well as its terminal handling and virtual terminal activities. Three virtual terminal related services are provided on the Zurich front-end:

- a terminal handling service, which interfaces individual terminals to the network and provides them with a virtual terminal image.
- a converter service for the central Venus interactive terminal facility, which interfaces individual users to the Venus facility and provides the conversion between network virtual terminal and local Venus terminal representation.
- a gateway service, which interfaces individual terminal or application facilities of one virtual terminal type (Zurich or CCG) with those of the other virtual terminal type.

These services are implemented as subsystems within a specially designed communication system. One process of each subsystem, 'Subsystem Main', is permanent and controls the dynamic creation and destruction of other processes of the subsystem.

While the process configuration of a particular subsystem is special, the process description format itself is identical for any process or coprocess. It is possible for a process to establish a communication channel with any other process of any subsystem for the exchange of data and signals.

2.2 *Terminal Handling Subsystem*

The physical terminal input/output uses a D.E.C. DH11 programmable sixteen-line multiplexer with individually selectable line speeds up to 9.6 Kbits/sec. Output uses direct memory access and is individually programmable for each line, but input uses a common 64-character fifo silo which requires a software character demultiplexer.

The main process of the terminal handler subsystem, 'Terminal Main' (TM), can be considered to 'own' this hardware and it dispenses to the member processes access rights to particular lines. The processes are created either as a result of an 'Open' request from some other process, or as a result of the input from a keyboard of a particular string indicating a desire to use this line.

The terminal process created by TM has two roles to play; on the one hand it behaves like the command process of the main console, providing the user with access to a responsive command interpreter; on the other hand it enhances the terminal with the characteristics of the Virtual terminal and links it, via a communication channel, to some other process and service.

We decided to provide software character echo so full control over key assignment could be offered. Any control key can be defined by the user to be his end-of-message, init, carriage return, tab, etc. The key processing is divided into two parts:

pre-processing:
this occurs as soon as a key is available. The physical key value is mapped into an internal representation of that key's logical function. During pre-processing signals for the command interpreter and attentions are also serviced.

processing:
This occurs only when the presentation device can be assigned to the keyboard. Then, for all preprocessed keys an echo is returned to the presentation device, and the corresponding item or part thereof is added to the current outgoing network buffer. When an end-of-message is encountered the buffer is terminated and sent over the communication channel to the process' communication partner; the presentation device is freed for use and, if in alternate mode, the turn reversed.

Requests for the command interpreter are signalled by the break character, are valid at any time, and for all practical purposes are immediately honoured, although some optimization has been introduced for the event that a message is currently being processed and transmitted to the display when such a request occurs.

Via the command interpreter the following control commands can be issued:

(a) *Terminal Mapping Control*
- PARAM, <x>,<y>

Sets the virtual page size of the terminal.

(b) *Keyboard Mapping Control:*
- LISTKEY

Requests display of current key definitions.
- DEFKEY,<function>,<key>

Defines the function of the named key. Possible designations are:
INIT,NO-FUNCTION,ATTN,TABULATOR,BACKSPACE, ERASE,C.I.CALL.
- DEFEOM,<function>,<key>

Defines the function of the named key to that specified plus EOM.
- DEFTAB,<integer>...<integer>

Defines the horizontal tabulator positions
- KEYSET

Selects a pre-defined set of key definitions.

(c) *Call Control*
- CONNECT<net><host><port>

Requests connection to named destination and facility.
- TERMINATE

Requests termination of terminal process.
- END

Requests termination of dialogue.

(d) *Cassette Control*
- READT

Directs terminal process to read from cassette tape.
- WRITET

Directs terminal process to read to cassette tape.
- ENDT

Directs terminal process to end cassette operation and resume normal correspondence with keyboard or presentation device.

(e) *Miscellaneous*
- VERIFY

Requests re-display of current contents of outgoing message.

2.3 *Converter Subsystem*

The Converter subsystem of the front end provides, in conjunction with a peripheral processor of the CDC mainframe, the linkage and conversion necessary to allow a network virtual terminal to access the Venus facility.

The Venus facility operates on the mainframe central processor and can be viewed as a collection of 'high level modules' (HLM) which perform specific tasks, e.g., editing, copying, job submission/retrieval, plus the organization and control needed to maintain a user identity, file and work space, scheduling, etc. The Venus facility interfaces via 'terminal records' with peripheral processors (PP) which drive the hardware. These 'terminal records' are owned by the drivers and represent individual terminal connections. The Venue facility and the PP-driver exchange lines via these terminal records. The lines are strings of Display (CDC internal code), Ascil, or binary data together with a limited amount of control information, e.g., cr, lf. The Venus facility also recognizes a two-level interrupt signalling; on receipt of an interrupt, the current module is restarted at a 'fall-back' address provided by the module; on receipt of a second interrupt, with no intervening lines, the current module is dropped and control is returned to the Venus command interpreter module.

The network connection is so designed that the central Venus facility is completely unaware of the network. It sees only an additional PP-driver which is fully compatible with other such drivers. The PP-driver and the front-end share the following tasks:
- driving of a high-speed line interface, and the execution of a line protocol for the safe and efficient transmission of information via this line.
- provision of a local 'transport service' to maintain the integrity of individual conversations and to control the flow of information on these conversations.
- provision of a terminal line transmission service in/out of the central Venus system. The burden of this last task is performed primarily by the front-end which utilizes the Venus terminal-record line format.

Within the front-end the converter subsystem is responsible for the latter two tasks, - the TS and terminal levels. A line-driver provides common access to the line interface for the Converter and other possible subsystems.

The main process of the Converter Subsystem, 'Converter Main' (CM), monitors the availability of mainframe facilities as provided by the line driver. Once the Venus facility is reported active, CM attemps a connection with the PP's Venus-Main; if successful, CM starts to consider positively any requests for service. It accepts such a request, negotiates with the PP Venus Main for a partner process in the PP and creates a Converter process. This process verifies its link with its counterpart in the PP and then establishes a communication channel with the requesting process- which might belong to the TS, terminal, or gateway subsystem.

2.4 *Gateway Subsystem*

The Gateway provides a mapping between the Scroll level of two different EIN virtual terminal protocols, the Zurich protocol [2] and the CCG protocol [1]; this latter is operating on the IRIA and Grenoble Hosts. The most significant differences between the two protocols are:
- The Zurich protocol is symmetric, the CCG is not.
- The Zurich protocol assumes basic addressing capabilities, whereas the CCG negotiates them, e.g. carriage return, differential forward, differential backward, and absolute addressing, etc.
- Both protocols negotiate a mode (alternate or free running), and line size, but the CCG protocol allows an infinite line size, the Zurich not.
- The Zurich protocol has only synchronized attentions, and a dynamic reinitialization of the 'turn'; the CCG has some non-synchronized attentions and always a static reinitialization of the turn.
- The Zurich protocol provides for 16 function codes which can occur anywhere within a message, the CCG only two, and these only coupled with the End-of-Message.

Thus, it is the Gateway's problem to:
- maintain addressing tables (static and dynamic) for extending the connection to the actual desired service.
- bind the two liaisons - between caller and gateway and between gateway and callee.
- realize a satisfactory negotiation with both partners such that a reasonable chance exists for the conversation to take place.
- translate and forward data, addressing, control, and attentions, and achieve a proper resynchronization after an attention phase.

To accomplish this task, the main process of the Gateway Subsystem, 'Gateway Main' (GM), maintains static and dynamic address tables, whereby the latter can be updated via lettergram request. GM, in processing a request for a gateway process creates a co-process pair. The first co-process compares the initial request with the addressing tables, thereby determining both the nature of the caller (CCG or Zurich) and the network address of the desired service. The second co-process based upon the destination network address, requests a process from the appropriate subsystem, establishes a private communications channel and reactivates the first co-process which completes the private communications channel with the original requesting process and two 'liaisons' are open.

The negotiation phase has two variants, depending upon whether the caller is a CCG terminal or the callee is a CCG application. If the caller is a CCG terminal, then the CCG co-process sends a READ-PARAMS item to the terminal and receives from it a PARAMS item. This information is used by the Zurich co-process to format a Capability item for the remote partner. If the caller is a Zurich facility then the addressed party must be a CCG application (CCG terminals cannot be called). The CCG co-process thus waits for a SET-PARAMS item from the CCG application process, and this information is used by the Zurich co-process to negotiate with his caller. The gateway remembers addressing limitations of the CCG terminal, but these are not (and cannot be!) reflected in the Zurich negotiation; these differences are easily compensated by the mapping.

If, at the end of the negotiation, neither or both CCG and Zurich have the 'turn', the co-processes resolve the situation by issuing an appropriate 'Please' or End-of-Message to one partner or the other.

During the data phase, the one co-process accepts Zurich items, and maps these to CCG items, compensating for possible address restrictions; e.g. an absolute address for a CCG partner with only differential addressing is mapped into the appropriate offset from the current position. For further details of the exact mapping, attention handling, etc., the reader is referred to [5].

3. HARWELL TERMINAL SYSTEMS

3.1 *General*

A GEC 4080 minicomputer at Harwell is connected to the European Informatics Network (EIN). It is controlled by the OS4000 operating system in conjunction with a network control package providing access to the EIN transport service. The GEC 4080 has connections to an in-house computer network (MESH) and an IBM 370/168. A transport service gateway function between EIN and MESH is supported.

The following virtual terminal services are available at Harwell:
- a terminal handling service
- a network status display service
- access to an interactive service (HUW) provided by the IBM 370/168.
- access to a PDP-11/45 system connected to the MESH network.

The terminal handling service, network status service and HUW access software are implemented within the GEC 4080; access to the PDP-11 system requires the use of a transport service gateway in the GEC machine and virtual terminal service software in the PDP-11/45.

3.2 *Terminal Handling Service*

A terminal handling service is available for terminals connected to the GEC 4080. This service provides terminals connected to the standard timesharing subsystem of the operating system with a virtual terminal image and makes them addressable entities within the transport service. Both scroll and page terminal operation are supported. Terminal users are able to initiate and receive calls via the virtual terminal service.

The terminal handling service maintains a process for each terminal involved in network activity; these processes are created dynamically in response to user requests. A terminal process communicates with its associated terminal through the OS 4000 terminal management system. Terminal handling at the character level is delegated to the underlying operating system; the terminal process interacts with a user on a line-by-line basis. The terminal process accesses EIN via the local transport station. It provides the user with access to a command interpreter and adapts the terminal characteristics to those of the virtual terminal.

The command interface provided for the terminal user is constrained by the line-by-line nature of the interaction with the associated terminal process. However this has been used to advantage in order to design a 'comfortable' user interface. The command process can either be in local or network mode; in local mode user input is treated as a command; in network mode user input is forwarded via the virtual terminal service. It is possible to switch from network mode to local mode by entry of an escape command; the process reverts from local mode to network mode on completion of command processing. A command may be entered at any time in a virtual terminal session; this permits a user to retain control over communications under all circumstances. This facility enables timeouts to be made the responsibility of the user rather than the terminal process.

The more important commands available to the terminal user are:

(a) *Call Control*

- DIAL <net><host><process>

Request set up of a transport service liaison

- BREAK

Request terminal process to enter 'end phase' of protocol

- KILL

Request termination of liaison.

(b) *Terminal Mapping Control*

- LINE [<line length><page length>]

When X and Y are present the command requests the resetting of the virtual page size (Y=0 implies scroll mode working required); otherwise the current parameter settings are returned.

(c) *Status Interrogation*

- WHO

If a liaison has been established the command requests the display of the remote party address. This is particularly useful if an incoming call is being serviced.

- PORT

Requests display of local transport service address of terminal process. The terminal process address is bound to the identity of the user rather than the terminal.

There are a number of other commands for the help of naive users, the definition of escape characters, the generation of seldom-used protocol features (i.e. transparent text, echo control) and the control of a diagnostic tool which maintains a journal of all network activities.

In addition to the commands already described there are a number of more frequently used commands which may be entered while the terminal process remains in network mode. These in-line commands take the form:

<user-defined escape character><command character>[<parameter>].

In-line commands are used for local editing, cursor control, function input, message termination and attention input. If it is desired to use the escape character as data then the usual conventions of character repetition apply.

3.3 *HUW Network Services*

A sub-system is available which provides network access to HUW (Harwell Users Workshop), an interactive program preparation and job submission facility on the IBM 370/168 at Harwell. This subsystem permits users of the virtual terminal service to access HUW, which is primarily designed for teletype - like terminals.

The HUW access sub-system is comprised of two processes. One process handles the connection between the GEC 4080 and HUW. This connection is a full duplex asynchronous communications line. The other process handles network communication and the virtual terminal protocol. Because of the method of connection to HUW the sub-system must emulate a teletypewriter device when viewed from the IBM machine. Scroll mode virtual terminal access is supported; this requires a mapping from message to character-oriented operation within the access sub-system.

Functions of the HUW sub-system include:

- logging-on procedures (included for password protection and authentication)
- automatic logging-off in the event of network failures
- recognition and notification of HUW system shutdown
- alternate dialogue control
- processing of system broadcast messages.

The areas of logging-on and alternate dialogue control are among the more complex features of the sub-system. They both require an intimate knowledge of the expected commands and responses of the HUW system. Alternate dialogue 'turn control' is particularly difficult as it relies on the recognition of context-dependent text strings in conjunction with a timer which permits end of message detection.

The attention handling and function features of the virtual terminal protocol are available for user interaction with HUW. A single attention is used as a system-interrupt. Currently four functions are implemented for purposes such as terminal testing; their long-term usefulness is under review.

3.4 *PDP-11 Access Sub-System*

Access to the PDP-11 system is somewhat more complex than the previous examples as it involves the use of another packet switching network and a transport service gateway. However, from the virtual terminal viewpoint these complexities are transparent as a transport station is provided in the PDP machine with similar interfaces to that in the GEC 4080.

The PDP-11 system accessible via EIN uses the RSX-11M operating system. The virtual terminal access method and transport station execute as separate tasks. A virtual peripheral device driver acts as an interprocess communication link between the access method task and the collection of processes and services normally available in an RSX environment. The pseudo device driver is the only addition required to the operating system to permit network access. Interfacing in this manner resulted in the PDP-11 system presenting an image of a free-running line-oriented terminal handling system to the virtual terminal access task. This permitted straightforward provision of scroll mode terminal access to the PDP-11.

In operation the need to traverse another network when accessing the PDP-11 was initially very noticeable. This was the result of delays coming from end-to-end flow control. Modifications were made to the transport service gateway in the GEC 4080 to enable it to anticipate the requirements of the PDP-11 host. The overlapping of operations obtained by this technique eliminated the delay problem.

3.5 *Network Status Display Service*

A process operating in the GEC 4080 has been designed to poll the status of nodes and subscribers on EIN making use of the echo process available in each of these computers. These status interrogations are made at regular intervals. The information obtained can be accessed over the network by terminals operating in the page virtual terminal mode described in [2]. As the application is self-contained the problems of interfacing are not large. However the application does make some assumptions about the integrity of cursor and display information in virtual terminal emulators which may not always be valid.

4. DISCUSSION

4.1 *Terminal Concentrators*

Both Harwell and Zurich have had little difficulty in designing and implementing terminal concentrators in minicomputers over which they had total control. The main points of interest are:

- for general-purpose use it is necessary to provide a sophisticated operator interface for control over network operations.
- it is necessary to be able to configure the operator interface as regards the definition of keystrokes which have specific actions e.g. attention, in a convenient manner for specific applications.
- the definition of message formats acceptable to a service e.g. line of text preceded by carriage

control, must be possible at the operator interface.

- control of the terminal interface at the character level provides a better opportunity to design a powerful and flexible operator command structure.

One interesting area of difference between the implementations is that of addressing. In the Zurich implementation terminals have network addresses which have a one-to-one correspondence with physical terminals. At Harwell network addresses are owned by users of the terminal concentrator; a terminal has no intrinsic identity. The former approach is more useful where terminal-terminal communications are envisaged; the latter approach is convenient when terminal-service communications are of prime interest.

4.2 *Service Adaptions*

The systems described perform two different types of service adaption:

- external physical device emulation (HUW system)
- internal logical device emulation (VENUS, PDP systems)

When confronted with a large mainframe operating under a manufacturers proprietory software the former method of service adoption is almost always necessary. The latter technique is viable in those circumstances where services and their underlying operating systems are amenable to the addition of extra device handling processes. Although terminal-service dialogues are alternate in nature it proved difficult or impossible to police the dialogue at the level of the terminal adaption. Therefore all our service adaptions make use of the free-running mode of terminal protocol operation. This use of free-running mode may cause flow control problems at the application level of which service users should be aware. In practice we have not experienced any difficulty in this area.

A service always has a particular technique for interacting with a terminal. For example, it may precede all lines with carriage control information on output and expect similar conventions on input. The virtual terminal protocol has no means of conveying this information to a user of a service. Device independence is provided by the protocol, not some higher level of data representation adaption. Information of this type must be known and used by a terminal operator to control the operation of his virtual terminal emulator.

The service adaptions described here give examples of specific and generic addressing of services. Specific transport service addresses are used by the Harwell services, each transport service address enables one remote user to access a service. The Venus system at Zurich has a single generic transport service address. Many concurrent dialogues with Venus using this one address are possible.

5. *Conclusions*

The systems described here represent examples of techniques which can be used for interconnection of services and terminals on networks. One general conclusion can be drawn: there is no best way of performing network interfacing in the current environment. The preferred solution depends critically upon the constraints of a particular system. It has also become evident that terminal protocols do not guarantee compatibility between services and terminals. A similar problem has already been encountered in network file transfer. This is an area which needs to be investigated further. Notwithstanding these reservations our experience has strengthened our belief in the usefulness of virtual terminal protocols.

Acknowledgements

The authors would like to acknowledge the many helpful comments and criticisms offered during the course of the work by their colleagues in the EIN community. Particular thanks are due to P.Schicker and C.R.V. Reed for their active participation in this work. This work was partly funded by the Computer Systems and Electronics Requirements Board of the UK Department of Industry.

REFERENCES

[1] Center Coordination Group (EIN): A Scroll Mode Virtual Terminal, Ed by P. Schicker, H. Zimmermann, EIN/CCG/77/02, 1977.

[2] P. Schicker, A. Duenki: The Virtual Terminal Definition, Computer Networks, December 1978.

[3] M. Deparis et al: The Implementation of an End-to-End Protocol by EIN Centres, Proc. of Third International Conference on Computer Communications, Toronto 1976.

[4] A. Duenki, R.P.J. Winsborrow: A Virtual Terminal Protocol; Experience and Critique, to be published.

[5] P. Schicker, A. Duenki: Page Virtual Terminal, Formal Implementation, EIN/ZHR/72/024, October 1977.

The paper was received on December 13, 1979

The Euronet network: origins, reasons, and possible future applications

M. Mangoni
MINISTERO PT - Direzione Centrale Servizi Telegrafici, ROMA
A. Misino
MINISTERO PT - Azienda Stato Servizi Telefonici, ROMA

ABSTRACT

The article describes the environment in which the idea of implementing the Euronet network was born in the European Economic Community, as well as the environment in which the PTT Administrations of the EEC member countries have been entrusted with its realization and the necessary organizational structure set up for this end.
A short description of the Euronet network is given, its possible utilization as a public data network to meet the general aims of the PTTs is mentioned and the main objectives already achieved are presented, as well.

1. INTRODUCTION

Euronet is an international information network, based on packet switching technology, promoted by the Commission of the European Economic Community which in 1975 assigned the PTT Administrations of the nine member countries with its implementation in order to create a European common market of technical, scientific and socio-economical information.
Euronet will be put into operation probably in October 1979 and about thirty data banks located in the EEC countries are presently foreseen to be gradually connected to it according to the CCITT Recommendation X25.
Euronet has initially been financed by the European Economic Community but afterwards the PTT Administrations decided to contribute the additional necessary funds considering that the PTTs could implement the network also by taking into account their own general aims.
Euronet is potentially in the condition of being upgraded to a public data network.
Euronet is also a first in the realization and in the communitarian management of the EEC PTTs.

2. MOTIVATIONS

The need for implementing an adequate data network, enabling the largest category of user data terminals (DTE) already in business to work together with the various data banks and computer resources, located in the different countries, had already been felt for some time in Europe.
The lack of precise technical standards and of homogeneous and coordinated tariff criteria had hindered even the birth of concrete initiatives by the Administrations in charge of the telecommunication services.
Furthermore, the existing traditional technical environment of manufacturers and managers of telecommunication services on one hand, and of computer manufacturers on the other, belonged to two such different and separate worlds that it was not easy to bring them together.
The lack of solutions supplied by the PTTs gave good reasons for the creation of private networks of different kinds which were generally incompatible from the technical point of view. Such a situation also favoured the generation of confusion between roles and responsability, mixing those of communication -the proper domain of the telecommunication people- with those of data processing -the proper domain of the computers people.
The real object to be pursued was that of configurating a public network not only capable of carrying out the functions of data transmission and switching, but also of enabling a "dialogue" among terminals and computers of different kinds operating at different speeds.
In order to achieve such an aim it was necessary to solve the problem of mutual incompatibility between the equipment already in commerce by fixing the standards for their cooperation, and it was felt desirable to lay down the characteristics of new kinds of terminals to be taken as reference for the future.
The occasion for a decisive push towards this target appeared when the European Economic Community agreed to the setting up of its own computer network for scientific and technical information. It was felt that a significant stimulus towards the solution of the above mentioned problems could be derived from entrusting the achievement and management of such a network to the PTTs of the nine member countries, which would thus have the opportunity of integrating and using the network also for their own specific data transmission purposes.
The importance and validity of these points and the idea of looking for a European solution, made their way within the European Economic Community along with the political significance that such an operation would also represent within the context of developing the European Community.

3. SIGNIFICANT STEPS -- ORGANIZATION AND WORK OF THE EURONET CONSORTIUM

The need for a series of specific initiatives, among which, in particular, the implementation of an adequate computer network within EEC, was indicated, in 1971, in a Ministers Council resolution aimed at coordinating the member countries' activity dealing with scientific and technical information and documentation.
From this starting point the approval was given by the Council for a first three -year plan of actions that gave rise, in December 1975, to the stipulation of an Agreement between the "Commission" of the European Community and the PTT Administrations of the nine member countries.
This Agreement assigned to the nine EEC PTTs - grouped in a Consortium by signing a "Convention" - the task of implementing the European telecommunication and information network, the "Euronet" network.
In 1978 the Council of the European Community approved a second three year plan which would provide for a series of actions in the field of scientific and technical information and documentation to support and develop those actions undertaken under the first plan, and the collateral plan regarding the overcoming of language barriers among the various countries of Community through the adoption of multi-lingual systems.
The fundamental target is always that of developing, within the EEC, an information market which is more and more complete and accessible to the greatest number of users in all member countries, establishing conditions of equal competition.
The Consortium of the PTTs adopted, from the moment of its constitution, the following organizational structure:

- -- Management Committee, with decisional functions; they meet when necessary and each PTT Administration is represented by a member.
- -- Technical Planning and Implementation Committee and Commercial Committee, with consultative functions, respectively, on technical matters and on administrative and tariff matters; they meet when required and each Administration is represented by a member.
- -- Project Team with permanent members coordinated by a Project Director; they work closely with representatives of the Commission to ensure that the network meets with the agreed objectives and is provided within the agreed time table.

At a later stage, in order to solve national installation problems and to face technical-operational matters and equipment maintenance, respectively, a Realisation Group and a Maintenance Group were set up in each country.
Lastly, a Commercial Group with promotional and user assistance functions has been set up in each country.
The choice of the network project was made in December 1976.
The project chosen was the one proposed by a group of European Companies led by the French SESA, which had presented a project derived from TRANSPAC network technology. This project appeared to be more consistent with the view of a possible further upgrading of Euronet to a public data network.
In June 1977 the French Administration, also on behalf of the other PTTs, signed a contract with SESA for the provision of both hardware and software.
During the same year the international tender for modems at 48 Kbit/sec between the network nodes was assigned to the Italian Company ITALTEL with a contract signed in February 1978 by the Italian Administration, also on behalf of the other members.
The contract for supply of modems at 9.6 Kbit/sec for connections between the equipment at some remote access points and their serving node was assigned to the French Company SAT.
Due to the complexity of the problems to be solved (most of them are of a new kind and subject to agreement by the Administrations and by the EEC Commission), the activation of the network, which had been foreseen for the end of 1978, will take place in the second half of 1979.

4. A BRIEF DESCRIPTION OF THE NETWORK

The initial topology of the network is shown in figure 1; four nodes (PSE, packet switching exchange) are located in Frankfurt, London, Paris and Rome, and five remote access points, composed of time division multiplexers, are located in Amsterdam, Brussels, Copenhagen, Dublin and Luxemburg.

The four nodes also perform the functions of assembling the characters in packet and vice versa (PAD, packet assembly/disassembly) in order to allow the interchange of traffic between packet and character mode terminals.
The network will verify the virtual call and permanent virtual circuit facilities.
It will adopt a fixed routing method, that is, all packets of each call will follow the same

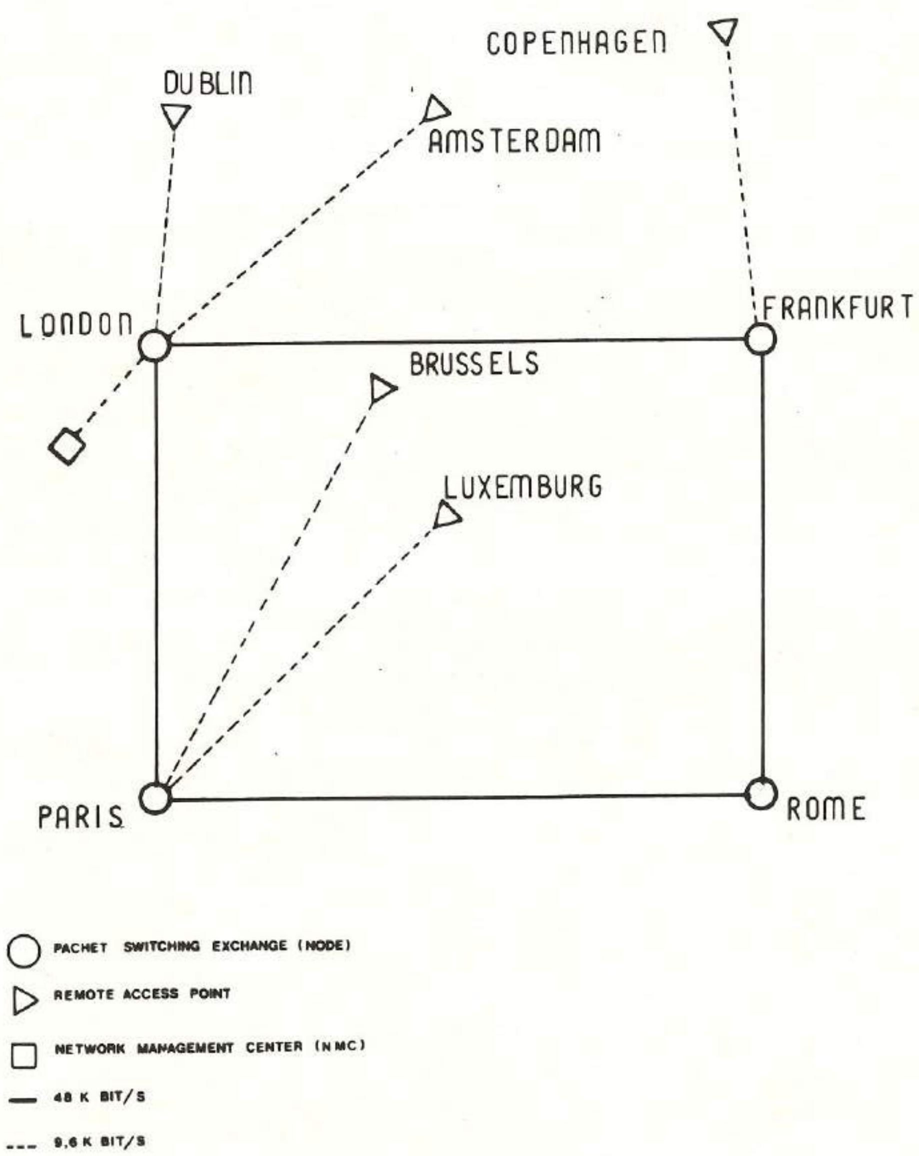


Fig. 1 - Euronet layout at opening date

A network management center (NMC) is located in London; apart from carrying out dynamic working operations on the network, it will provide for the recording of billing and statistical data.
The network, as already mentioned, is a packet switching type derived from the French Transpac network technology.

optimum route assigned to the first packet of the call.
The advantages over the adaptive routing method (that is, independent for each individual packet) are that the packets do not need to be reordered at the arrival node, and the accounting operations for transit calls are easier.

The inter-node protocol, which at the opening date of the network will not conform to the recent international standards, will at later date presumibly be brought up to date to the CCITT protocol X75.
The nodes are the modular type and they are composed of the following basic parts, as indicated in the simplified block diagram in figure 2:

- group unit (GU); it implements the PAD facility and performs the packet switching function; the capacity of the associated memory may be extended up to 240 K words;
- synchronous line adaptors (SLA); they implement the synchronous packet interfaces; at the most 32 SLAs can be added;
- synchronous line unit (SLU); it carries out control and transmission functions between

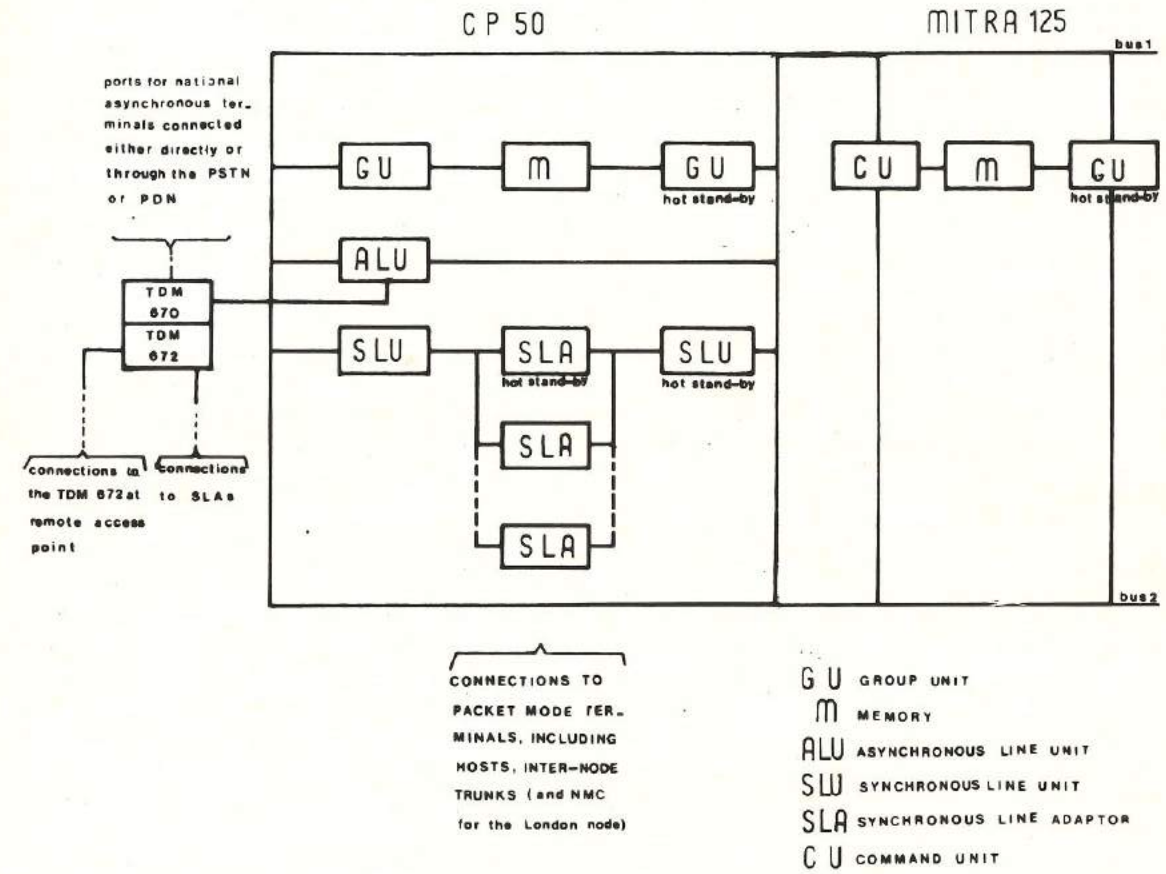


Fig. 2 - Euronet node - Block diagramm

1) CP 50 switch modules; they are made up of a purpose-designed multimicrocomputer developed in France for Transpac network; they support the connection of users and perform the numerous repetitive functions implied in the switching of packets; at the most 32 modules can be added; each of them can support up to 240 synchronous packet interfaces (synchronous packet ports) and 240 asynchronous-character interfaces (asynchronous ports); the synchronuos ones will connect packet mode terminals, including data banks (hosts), operating in accordance with CCITT Recommendation X25, and the asynchronous ones will connect terminals operating in accordance with Recommendation X28; the number of the necessary CP 50 modules depends on the volume of the traffic and the number of the users to be connected. Each CP 50 module is composed of the following main units:

the SLAs and the GU;
- asynchronous line unit (ALU); it carries out control and transmission functions between the asynchronous connections and the GU; the ALU processes the characters coming from the asynchronous lines so that the ALU presents to the GU an interface similar to that presented by the SLU;

2) command unit (CU); it is made up of a general-purpose minicomputer, the SEMS MITRA 125, whose associated memory capacity is 64 K words; the command unit performs the functions that necessitate a more complex treatment; with its own software it forms the most "intelligent" part of the node; the CU controls the operations of the CP 50 and runs the signalling, the construction and the clearing of the calls; at the most 8 CU modules can be added and their necessary number depends closely upon the number of calls per time unit;

3) CASE 670 and 672 time division multiplexers; the 670 ones implement the asynchronous-character interfaces and connect the national asynchronous terminals either by point-to-point circuits or through the public switched telephone or data networks, the 672 ones connect the equipment of the remote access points (these are made up of CASE 672, which connect the synchronous packet mode terminals, including data banks, and of CASE 670 which connect the asynchronous terminals).

The present dimension of each node is made up of two command unit (one main and the other stand-by) and one CP 50 module (with double units).

The distribution of the interfaces for the users of the various countries, foreseen at the moment, is indicated in the following table:

tion codes; after having ascertained the existence of the received code (this is of a secret nature to avoid frauds by one user to another), the node will abilitate the call and will send to the NMC the corresponding address of the caller (NUA, network user address), for charging purposes.

The speeds of 110, 300, 600, 1200 bit/sec have been considered for the asynchronous terminals, and the speeds up to 48 Kbit/sec have been considered for the packet terminals.

With the implementation of the national data networks, user terminals (DTE) as well as the data banks, will foreseeably be transferred from the Euronet network to the national one.

For the extension of the Euronet network and, in particular, the above mentioned interfaces, it is therefore necessary to keep in mind not only the predictable increase in usage, but

Table 1: Euronet terminal and host interfaces (ports)

Country	Character mode Asynchronous	Packet mode Synchronous (included hosts)	Total
Belgium	32	4	36
Denmark	32	4	36
German Federal Republic	64	31	95
France	0 (*)	11	11
Ireland	32	4	36
Italy	48	35	83
Luxemburg	32	4	36
Netherlands	32	4	36
United Kingdom	64	25	89
Total	336	122	458

It is foreseen that in the countries other than France the asynchronous terminals will initially gain access to Euronet through the national switched telephone network; a contention ratio of 3:1 for the relevant ports has been assumed. For the terminals that gain access through the telephone network, the singling out of the calls in debiting the traffic carried out, will be operated by means of an identification code (NUI network user identificator) that will be sent by the caller and will be controlled by the node on its own internal list of the identifica-

also the forecast implementation time of the national data networks of the various countries.

5. PROSPECTS FOR EURONET TO BECOME A PUBLIC DATA NETWORK - MAIN RESULTS ALREADY ACHIEVED

As already mentioned, Euronet was conceived as a potential embryo of a public European data network.

However, the hypothesis of extending, in the various European countries, the Euronet network to a national level through the addition of Euronet nodes has not been carried out and the various PTT Administrations are now oriented to setting up their own national packet networks, in compliance with international standards (including those defined after the Euronet program had already started), so that these national networks will be able to work

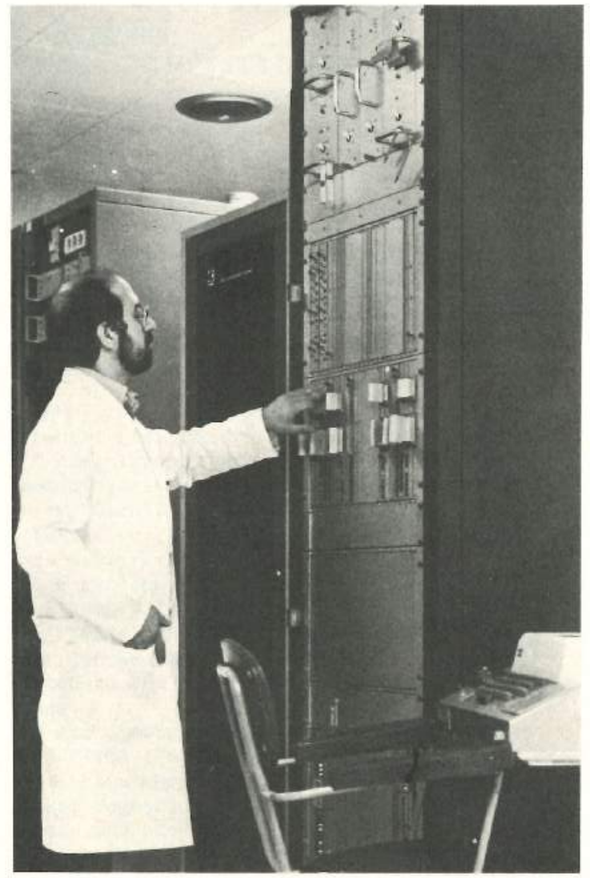
(*)The character mode terminals will be linked up directly to the existing French Traspac network that will be interconnected to the Euronet one by means of an appropriate international transit node.

If also the packet mode terminals will be linked up directly to the Transpac, the eleven interfaces indicated for France will be used as spare parts.

with each other directly.

If Euronet is upgraded to all such international standards, it will be able to work with all national packet networks.

This upgrading appears to be very important because only in such condition will Euronet be able to provide a concrete possibility of operating switching data traffic among all the countries of the Community, independently from the different times of implementation of their respective national networks.



(photo A.Guida)

Fig. 3 - The CP 50 switch module installed at the Rome node

The achievement of this objective would also be in accordance with the aim of estabilishing equal development and access conditions to information resources in all the Community partners.

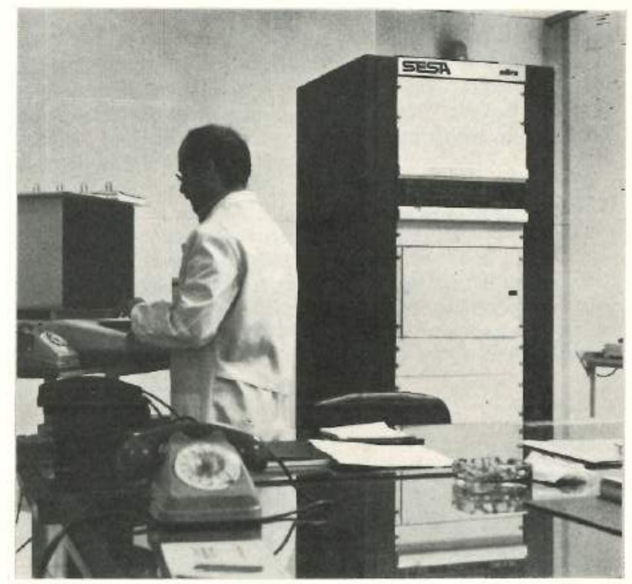
Therefore, the upgrading of Euronet to a public data network seems to acquire also a specific political significance, independent from the real convenience which the single PTT Administrations might gain with such a development.

However it is still unknown what real use is going to be made of Euronet in the field of public data transmission.

Anyway it is foreseeable that, once adapted to all international standards, Euronet will continue to play its significant role even after the realization of all national networks, by providing at least the transit service for overflow or back up traffic in international relationships which would not justify the setting up or increase of bilateral direct connections from country to country.

In any case the Euronet initiative has already achieved some important and concrete results, as for example the following:

- a solid base for the development of cooperation among the member countries in matter of scientific and technical documentation;
- an action of adaptation and organization in an area characterized, up to now, by proliferation of non-coordinated initiatives and private networks incompatible with each other;
- a generalized stimulus for the implementation of national data networks and for packet switching mode;
- a census on, and a sort of publicity for European information resources;
- a concrete incentive to the constitution of new European data banks and to the diffusion of information to benefit the largest level of users;
- an incentive to the definition of international standards for data transmission networks;
- an approach to the problems of the telecommunication and the computer worlds in unity of scopes and interests and looking for an European inclusion in an area of fundamental importance for every advanced progress;
- a promotional incentive for European industry;
- the definition of rational and homogeneus telecommunication tariff criteria, based on the volume of data transmitted without taking into account the lenght of the involved international paths.



(photo A.Guida)

Fig. 4 - The Mitra 125 installed at the Rome node

6. CONCLUSIONS

Euronet has undoubtedly played a fundamental leading role within all the community member countries and has probably promoted in a tangible manner the generalized acceptance of the packet switching technique.
Other European countries (not belonging to the Community), have shown interest in entering the Euronet Consortium. Switzerland has just signed an Agreement to participate in Euronet activities with a Euronet node in Zurich; Spain is waiting for the new protocol X75 to link up its own packet switching network with Euronet, and Norway and Sweden are studing a suitable solution to be connected to Euronet access points. Also Greece, a new member of the EEC, and presumibly Austria are interested in the Euronet connection. Generally speaking, new associations are welcome because on the one hand, they would bring additional traffic due to the new users which will provide a more efficient network utilization, and on the other hand they could add a greater value to the network itself because of the additional information resources made available to all other users.
Until national data networks are implemented all over Europe, Euronet is the only real present possibility of data exchange among all member countries.
Naturally, this network cannot be considered as anything but the beginning of a development process within Europe of an information exchange market, still entirely to be developed, which will require in any case many additional interrelated initiatives.

The paper was received on July 13, 1979

Network management and control in EURONET

A. M. Repichini
SIP SOCIETA ITALIANA PER L'ESERCIZIO TELEFONICO

Abstract. In telecommunication networks design, to obtain better performances in terms of user services, network availability and management simplicity, techniques have been introduced which utilize in switching nodes "intelligent" equipments, based on general purpose or specialized computers. In this context, one of the problems is the control and management of such a system. This problem, already important in experimental private networks, is crucial in public ones, realized and runned by PTT Administrations, which must ensure an high level of quality and continuity of service.
In this paper methodologies and techniques used in EURONET project will be shortly described.

1. NETWORK STRUCTURE

EURONET is an international telecommunication network, realized by a Consortium of European PTT Administrations on behalf of the European Community, in order to create a "Common market" of technical, scientific and social-economical information.
This network - structured on four Packet Switching Exchanges (PSE), mesh interconnected, and five Remote Access Points (RAP), star linked to the PSEs - is controlled by a Network Management Center (NMC), located in London and connected to the London PSE through two 9.6 kb/s lines.
NMC architecture is based on a minicomputer SEMS MITRA 125. The configuration consists of 128 kbytes memory, operator console, 50 Mbytes disks, magnetic tape, 2 VDUs, printer, card reader and special peripherals for visual alarms and buzzer.

2. CONTROL SYSTEM GENERAL ARCHITECTURE

Network operational facilities can be divided into different categories, depending upon whether they are concerned with the quality maintenance of subscriber service or with the network accounting management.
These functions are therefore essentially concerned with processing network control, supervision and management as well as collecting subscriber's billing information.
The main options characterising the implementation of these functions are:
- very high operational reliability;
- automatisation in order to reduce human intervention;
- distinction between distributed functions (e.g. data collection on subscribers or modules status) and centralised ones (e.g. observation of the whole network status, coordination of control and maintenance operations, etc);
- functional independence of various control system components, that are logically distinct from modules which assure the subscriber service;
- operational simplicity, essentially coming from the availability to network operators of visualisation means and powerfull control languages;
- modular design of the intercomponent service communication protocols, to allow future evolution of the system.

These concepts led to define, at a distributed level, PSE programs for local network management, subscribers accounting, management of virtual circuits, failure detection and system reconfiguration, preventive maintenance, etc.
On the other hand, within the NMC functions are defined which perform global network control, subscribers' management, statistics, etc. The NMC is linked to the network as an ordinary user. In order to allow transmission of service information and commands between PSEs and NMC, "virtual subscribers" are implemented within the PSEs, with a logical internal interface, similar to the real subscriber's one. NMC and "virtual subscribers" constitute a particular "closed user group" and can communicate through permanent or switched virtual circuits.
Accounting information collected by the NMC are stored, in a suitable format, on magnetic tape to be then transferred, in a first phase off-line, to the National Network Invoice Centers.
Control information, collected by the NMC on the network status, is shown on local or remote service terminals to the operators, who can interact with the control system by means of a particular network control language.

3. SERVICE TRAFFIC

Service information traffic between PSE, NMC and operators consists of:
a) Data systematically emitted by a switch. These data can originate from abnormal events (anomaly or failure reports), periodic events (statistical reports on switch status; "traffic tickets" describing the degree of utilisation of virtual circuits) or call events (set-up or clear-down of a virtual call)
b) Network control language commands sent by the NMC to a switch. These commands can

cause:
- activation of memory resident switch programs for failure localisation or hardware monitoring;
- looping of lines, modems, link equipments;
- isolation or integration of interchangeable elements in a switch;
- reconfiguration of the functional context of a switch (parameters modification);
- updating of program modules and tables (routing tables, users data tables, closed user group tables, etc);
- remote loading of programs for preventive diagnostic;
- remote loading of tables or programs;
- dumping of switch programs.

c) Data sent by a switch on NMC request. In general each command of the network control language causes a report from the switch about the command execution or, in any case, its reception.

d) Data sent by the NMC to operators. These information are constituted by alarms, indicating that a network component has failed and operator intervention is needed, or failure reports, concerning failures of elements that the system has automatically circumvented.
These information are presented using visual or audible alarms, listings, displays, teletype.

4. NETWORK OPERATION

By the mechanism above described, network operators are kept informed about the status of the various network elements as well as the traffic load, transmission problems, etc.
Moreover, in case of failure and depending on his nature, the operating personal can know:
- the exact location of the failure;
- information from different sources, whose analysis ehables localisation of the failed component;
- general diagnostic information, needing execution of more precise test to localise the failure.

As consequence of this information, several kinds of actions are possible.
On alarm and failure indication, the operator can restart, by a reconfiguration or a remote loading, a system completely failed, or start maintenance operations by giving location of the failure or, finally, cooperate in the maintenance operations by loading test programs, performing remote loops, etc.
Moreover, the operation staff can put under surveillance a system element or assist a subscriber in debugging his interface, using the facilities for remote looping and error counting.
Finally, network operators can perform system modification, related to:
- introduction of new hardware and/or software;
- connection of new subscribers. Even though no new hardware is to be added, nevertheless user configuration tables have to be changed on line and tests have to be performed in order to verify functionality of the new subscribers' transmission chain and interface.

5. CONCLUSION

The general structure of the supervision, measurement and accounting mechanisms in EURONET has been described.
Switches record periodically and on alarms measurements, billing information and data collected on modules status.
The information are processed by the NMC, stored into files and presented to operators by means of alarms, displays, listings.
This enables the operators to have a complete picture of the network.
In addition to this systematic data collection, operators may request special information or can interact with the network changing parameters, programs, tables, connections, etc.

The paper was received on July 13, 1979

A formal description of the DTE packet level in the X.25 Recommendation

S. Alfonzetti, S. Casale, A. Faro

ISTITUTO ELETTROTECNICO, UNIVERSITA' DI CATANIA — CREI (POLITECNICO DI MILANO)

Abstract. This paper proposes a formal description of the DTE packet level (level 3) in the CCITT Recommendation X.25. The formalization is based on the theory of colloquies and uses the logical matrix method. Commands exchanged between level 3 and its upper level are selected. Some uncertainties of the procedure of the DTE level 3 are pointed out and are overcome by introducing suitable hypotheses.

1. INTRODUCTION

In the last years, computer network study has been greatly developed. A computer network is made up of a communication subnetwork to which terminals and computers are connected. Both these users of the subnetwork are called DTE (Data Terminal Equipment). DCE (Data Circuit-terminating Equipment) is the interface of the communication subnetwork with the DTE.

The interface between DTE and DCE is the argument of the Recommendation X.25 of the CCITT [1] for terminals operating in the packet mode on public data networks. Recommendation X.25 independently defines:
- level 1, regarding the physical link between the DTE and the DCE;
- level 2, regarding the link access procedure (LAP)
- level 3, regarding the packet format and the control procedures for the exchange of packets between the DTE and the DCE.

Recommendation X.25 has caused several perplexities and discussions, specially on level 3 [2] [3] [4] [5], also because the levels 1 and 2 are consistent with some old recommendations. As to the level 3, the DCE is related in a more detailed way than DTE (1) and therefore it is easy to realize as several questions regard the behaviour of the DTE. Some of these questions are justly not resolved by Recommendation X.25 to leave a margin to the user, some others less justly. But, without making such distinctions, these questions crop up to the software designer who has to implement the DTE level 3.

The aim of this paper is to give to the software designer an useful tool for the implementation by formally describing the DTE level 3. In fact a formal description is a more convenient reference point than a word description.

In this work we have formally described the DTE level 3 as an "interlocutor" of the "theory of colloquies" [6]. The description technique used is based on logical matrices, state vectors and context procedures [7] [8]. For the completeness and non ambiguity of such a description technique, it has been possible to point out all the procedural questions of the DTE level 3. To formalize the procedure it has been necessary to overcome such questions by introducing suitable hypotheses, some of which according to [4] and [5]. Moreover the interrelationships between level 3 and its upper level (level 4) have been defined by introducing a set of commands as requested by the description technique.

Section 2 gives the reader a brief recall of the "theory of colloquies" and of the logical matrix description technique. Section 3 presents the hy-

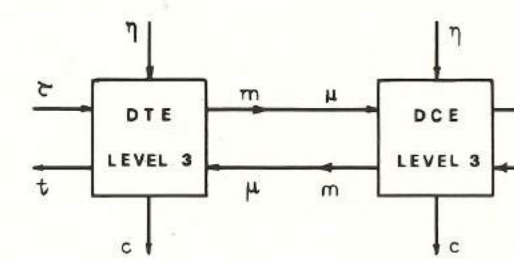


Fig. 1 - Schema of colloquy between the two interlocutors DTE and DCE levels 3

(1) In order to protect oneself from all abnormal DTE behaviours, the actions taken by the DCE on receipt of packets from the DTE are specified in all possible cases.

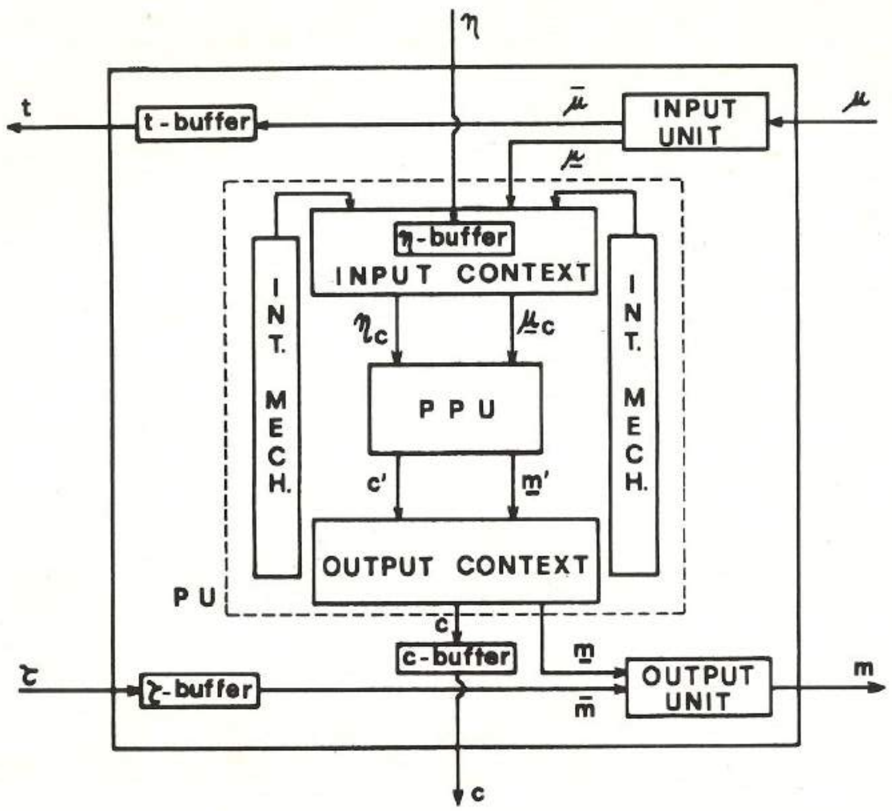


Fig. 2 - Internal structure of an interlocutor.

potheses on the DTE level 3. In section 4 a set of commands exchanged between level 3 and level 4 is introduced. Finally, section 5 is concerned with the formal description of the DTE level 3.

2. THEORY OF COLLOQUIES

The DTE and DCE packet levels can be represented by means of two machines called interlocutors in the theory of colloquies [6]. Each machine has 3 inputs, 3 outputs and a set of internal states. In fig. 1 the communication schema of the colloquy between the two interlocutors is shown.

The inputs of an interlocutor are :

- commands η, coming from the user of the interlocutor ;
- messages μ, coming from the other interlocutor;
- texts τ, coming from the user of the interlocutor and to be embodied in messages m.

The output of an interlocutor are :

- commands c, for the user of the interlocutor;
- messages m, for the other interlocutor;
- texts t, extracted from messages μ and for the user of the interlocutor.

Fig. 2 shows the internal structure of an interlocutor, which consists of the :

- buffers τ, t, η, c which allow the respective channels to be asynchronous;
- input and output units, which respectively divide and assemble the envelopes and the texts of the messages;
- processing unit (PU), which, following suitable rules, produces the output envelopes and the output commands owing to the input commands or the input envelopes.

Messages can be subdivided into two parts: envelope and text (see fig. 3). The envelope is that part of the message which contains information of PU. The text is the remaining part, to which PU is transparent; it is directly transfered from the input unit to buffer t, or from buffer τ to the input unit.

Envelopes (and commands) can contain not only of a op-code, but also of parameters (i. e. the packet send sequence number P(S) in data packets). In this case the number of the possible (input and output) envelopes (and commands) becomes high, and

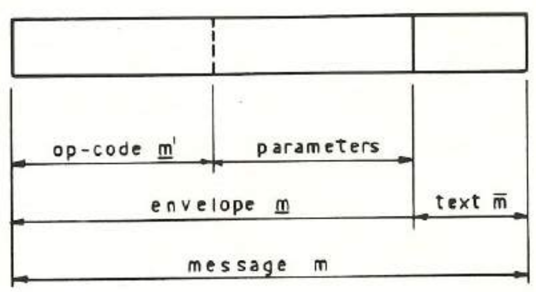


Fig. 3 - Message format.

consequently the description too complex. Therefore it is advisable to split the PU into two parts, in such a way that one processes the possible parameters of the envelopes (or commands) and the other the op-codes [7] [8].

For this reason PU contains (see fig. 2) :

- input context (IC), which processes the parameters of the input envelopes and commands and transmits the IC envelopes $\underline{\mu}_c$ or the IC commands η_c to PPU;
- proper processing unit (PPU), which produces the output op-codes owing to $\underline{\mu}_c$ or η_c coming from IC;
- output context (OC), which adds the parameters to the op-codes coming from PPU.

Moreover to avoid the interlocutor PU remains indefinitely in a given state, it is convenient that PU contains internal mechanisms (time-outs) which emulate input commands or input envelopes (see fig. 2).

From the above, the interlocutor procedure is completely specified if one formally describes the various components of PU.

The formal description technique with logical matrices is briefly recalled in the following.

According to this technique, the PU internal states are defined by means of two state variable sets :

- the first set is constituted of binary variables which can be modified by the PPU inputs; such a variable is shown with s, if it depends only on $\underline{\mu}_c$, or with σ, if it also depends on η_c;
- the second set V is constituded of generally non binary variables which can be modified by the inputs of the two contexts IC and OC; these variables are called context variables.

IC and OC can utilize the PPU binary variables while PPU cannot directly utilize the context variables; for this reason a set of binary variables Z has been introduced [8], which is a function of the context variables v and which can be directly used by PPU apart from by the contexts; these variables have been called global states (²).

In order to describe IC and OC, it is suitable to utilize programming languages (Algol, Pascal, etc.), while PPU can be axiomatically described as follows :

- at the reception of a command η_c, PPU produces:
  - an output message op-code: $\underline{m}' = Q(\sigma,s,z)\eta_c$
  - an output command op-code: $c' = H(\sigma,s,z)\eta_c$
  - a state transition : $\sigma = h(\sigma,s,z,\eta_c,\underline{0}_\mu)$
  - a text transmission : $\bar{m} = f_1(\sigma,s,z,\eta_c)\tau$
- at the reception of a envelope μ_c, PPU produces:
  - an output message op-code: $\underline{m}' = P(\sigma,s,z)\underline{\mu}_c$
  - an output command op-code: $c' = G(\sigma,s,z)\underline{\mu}_c$
  - a state transition : $\sigma = h(\sigma,s,z,0_\eta,\underline{\mu}_c)$, $s = g(\sigma,s,z,\underline{\mu}_c)$
  - a text transmission : $t = f_2(\sigma,s,z,\underline{\mu}_c)\bar{\mu}$

where :

Q,H,P,G	are matrices of logical functions;
h,g	are vectors of logical functions;
f_1, f_2	are logical functions;
$\underline{m}', \underline{\mu}_c, \eta_c, c'$	are vectors with only one entry different to zero (codification);
σ, s, z	are vectors of the binary state variables;
$\bar{m}, \bar{\mu}, t, \tau$	are texts;
$\underline{0}_\mu, 0_\eta$	are null vectors of suitable dimensions.

(²) Global states z can be defined also independently by v ; in such cases they take into account secundary states.

3. DTE PACKET LEVEL HYPOTHESES

The virtual connections between the DTE and the DCE are realized by means of logical channels and are of two types :

- permanent virtual circuit (PVC): established on permanent basis without selection;
- virtual call (VC) established on a temporary basis (and therefore needs set-up and clearing phases).

In this paper we consider only the last one because the description of the PVC can be obtained from the one of the VC, by eliminating the set-up and clearing phases and considering the restart procedure as the reset procedure of the VC.

Since several VCs may be active simultaneously we can suppose the DTE is composed of several interlocutors, one for each logical channel [8]. In the following we consider only one of such interlocutors and, for sake of generality, we suppose its channel to be mixed (that is utilizable for input and output calls) and available for the restart procedure (³). The various phases of a VC as foreseen by Recommendation X.25 are now recalled and for each phase the hypotheses which we must introduce during the formalization are presented. Input message not foreseen by the procedure (errors) are considered. The actions taken by the DTE in this case have been determined to eliminate the most critical situations (considering the time-outs of section 5).

Ready state

"If there is no call in existence, a logical channel is in the ready state"[1, sect. 3.1.1].

Hyp. 1: We suppose that the DTE, receiving a CALL CONNECTED packet on a logical channel in the ready state, transmits a CLEAR INDICATION packet to the DCE, but ignores data, interrupt, flow control, reset and DCE CLEAR CONFIRMATION packets.

Call set-up phase

"The calling DTE shall indicate a call request by transfering a CALL REQUEST packet across the DTE /DCE interface"[1, sect. 3.1.2]. The DTE is waiting for the CALL CONNECTED packet from DCE before the data transfer.

Hyp. 2: If the calling DTE, before the CALL CONNECTED packet reception, receives other types of packets (except for CLEAR INDICATION and restart packets), we suppose they will be ig-

(³) Recommendation X.25 specifies that restart packets must be transmitted on the logical channel 0. When the logical channel is not mixed and/or when the restart procedure cannot be utilized on it, the formal descriptions are easily deductible.

nored ([4]).

"The DCE will indicate that there is an incoming call by transfering across the DTE/DCE interface an INCOMING CALL packet"[1, sect. 3.1.3]. If the call is accepted, the called DTE will answer transfering a CALL ACCEPTED packet, permitting the data transfer.

Hyp. 3: We suppose the DTE transfers a CLEAR REQUEST packet if it does not accept the call ([5]).

Call clearing phase

"The DTE may indicate clearing by transfering across the DTE/DCE interface a CLEAR REQUEST packet"[1, sect. 3.1.7]. The DTE is waiting for the DCE CLEAR CONFIRMATION packet from DCE before considering the logical channel ready for a next call. In this waiting period the DTE can receive all the types of packets from DCE and may treat them, as foreseen by Recommendation X.25, but cannot transmit packets to DCE (except for clear and restart packets) because the DCE will consider the reception of these packets as an error.

Hyp. 4: We suppose that the DTE, receiving a DCE CLEAR CONFIRMATION packet in data transfer phase, transmits a CLEAR REQUEST packet.

"The DCE will indicate clearing by transfering across the DTE/DCE interface a CLEAR INDICATION packet"[1, sect. 3.1.8]. At the reception of this packet the DTE will answer with a DTE CLEAR CONFIRMATION packet.

Data transfer phase

In this phase, data, interrupt, flow control and reset packets may be transmitted and received by DTE.

Hyp. 5: In the data transfer phase, we suppose the DTE ignores the received call packets.

The flow control is used to avoid the transmitter sending more data packets than the receiver can accept. The flow control is realized by the window mechanism, which uses the packet send sequence number P(S) and the packet (correctly) receive sequence number P(R). "The packet receive sequence number, P(R), is conveyed in data, RECEIVE READY (RR) and RECEIVE NOT READY (RNR) packets"[1, sect. 3.4.1.2], whereas P(S) is conveyed only in data packets. The significance of P(R) in a RNR packet is unclear [2][3].

Hyp. 6: We suppose the significance of P(R) in RNR packet is the normal one, that is P(R) becomes the lower window edge.

"A DTE or DCE receiving a RNR packet shall stop transmitting data packets on the indicated logical channel"[1, sect. 3.4.1.4]. It is unclear when exactly a DTE receiving a DCE RNR packet should stop transmitting data packets [2].

Hyp. 7: We suppose that the DTE, receiving a DCE RNR packet, immediately stops transmitting data packets.

Recommendation X.25 does not specify what algorithm must be used by DTE to advance the DCE transmit window [2].

Hyp. 8: We suppose that the DTE, to advance the DCE transmit window, uses the algorithm reported in appendix.

"The interrupt procedure allows a DTE to transmit data to the remote DTE, without following the flow control procedure applying to data packets"[1, sect. 3.3.5].

Hyp. 9: We suppose that the DCE INTERRUPT CONFIRMATION packet has an end-to-end significance, that is the remote DTE has received the interrupt packet.

"The reset procedure is used to reinizialize the virtual call or permanent virtual circuit and in so doing removes in each direction all data and interrupt packets which may be in the network"[1, sect. 3.4.2].

Hyp. 10: We suppose that the DTE receiving an unexpected DCE RESET CONFIRMATION packet transmits a RESET REQUEST packet to DCE.

Restart phase

"The restart procedure is used to simultaneously clear all the virtual calls and reset all the permanent virtual circuits at the DTE/DCE interface"[1, sect. 3.5].

Hyp. 11: We suppose that the DTE receiving an unexpected DCE RESTART CONFIRMATION packet transmits a RESTART REQUEST packet to DCE.

Optional user facilities

For sake of simplicity, we have not considered the optional user facilities foreseen in the section 5 of the Recommendation X.25. For this reason we have not inserted the DTE REJECT packet in the output message set of the DTE.

4. INTERRELATIONSHIPS BETWEEN THE LEVEL 3 AND THE LEVEL 4 OF THE DTE

To describe an interlocutor (of level n) it is necessary to foresee all the commands η and c, which are exchanged with the adjacent levels (n-1 and n+1) and with the local controller ([6]). Nevertheless in the case of the level 3 of DTE, the commands exchanged with level 2 have only a local effect and therefore are not relevant to the purpose of this paper. We have foreseen the following commands :

- CALL REQUEST : level 4 requests to open a VC;
- WAIT FOR CALL : level 4 is waiting for a call from another DTE ([7]);
- CLEAR REQUEST : level 4 requests to clear the VC;
- RECEIVE DATA : level 4 is ready to accept a data packet coming from DCE;

([4]) Including INCOMING CALL packet (call collision).

([5]) Recommendation X.25 specifies only that in case of unsuccessful call the calling DTE will receive a CLEAR INDICATION packet [1, sect. 3.1.10]; than can occur because of a time-out in the network.

([6]) The local controller is a device which allows the user to interact with an interlocutor of a given level, without involving the intermediate levels.

([7]) Of course, in WAIT FOR CALL command, level 4 can specify the addresses of the calling DTEs. The absence of these addresses means that any INCOMING CALL is accepted. To semplify we have considered only the last case.

- SEND DATA : level 4 requests to send a data packet ([8]);
- RECEIVE INTERRUPT : level 4 is ready to accept an interrupt packet coming from DCE;
- SEND INTERRUPT : level 4 requests to send an interrupt packet;
- RESET REQUEST : level 4 requests to reset the VC in order to reinitialize a new data transfer or to retransmit immediately an interrupt not confirmed by DCE [4];
- RESTART REQUEST : the local controller requests to send a restart packet.

Moreover we have foreseen the following commands c :

- INCOMING CALL : level 3 notifies level 4 the arrival of a call;
- CALL CONNECTED : level 3 notifies level 4 that the VC has been established;
- CLEAR : level 3 notifies level 4 that the clearing of the VC is completed;
- CLEAR REQUEST : level 3 notifies level 4 the emission of a CLEAR REQUEST packet;
- DCE DATA : level 3 notifies level 4 that the text of a data packet is available in buffer t;
- DCE INTERRUPT : level 3 notifies level 4 that the text of an interrupt packet is available in buffer t;
- DCE INTERRUPT CONFIRMATION : level 3 notifies level 4 that the interrupt packet has been delivered to the remote DTE;
- RESET : level 3 notifies level 4 that the reset is completed;
- RESET REQUEST : level 3 notifies level 4 the emission of a RESET REQUEST packet;
- RESTART : level 3 notifies level 4 the clearing of the VC because of a restart.

5. FORMAL REPRESENTATION OF THE DTE PACKET LEVEL

DTE state variables

The states defined in the Recommendation X.25 are logical channel states. We define DTE states as logical channel states "seen" by DTE. As we suppose that the answers of the PU of the DTE to the envelopes coming from the context are istantaneous, some logical channel states foreseen in tbo X.25 Recommendation (p_3, p_5, p_7, d_3 and r_3) have no correspondents in the PU of the DTE.

To formalize the behaviour of the DTE packet level we introduce the following state variables :

- state variable subset Σ :

$\sigma_1=\{$ 1 if the DTE has emitted a CALL REQUEST packet; 0 otherwise

$\sigma_2=\{$ 1 if the DTE has emitted a CLEAR REQUEST packet; 0 otherwise

$\sigma_3=\{$ 1 if the DTE has emitted a RESET REQUEST packet; 0 otherwise

$\sigma_4=\{$ 1 if the DTE has emitted a RESTART REQUEST packet; 0 otherwise

- state variable subset S :

$s_1=\{$ 1 if the DTE has received an INCOMING CALL packet; 0 otherwise

Tab. 1 - X.25 DTE states ([9]).

DTE state	σ_1	σ_2	σ_3	σ_4	s_1	s_2	Logical function
Ready p_1	0	0	0	0	0	0	$\bar{\sigma}_1\bar{s}_1\bar{\sigma}_2\bar{\sigma}_4$
DTE waiting p_2	1	0	0	0	0	0	$\sigma_1\bar{\sigma}_2\bar{s}_2$
Flow control ready d_1	1 0	0 0	0 0	0 0	0 1	1 0	$(\sigma_1 s_2+s_1)\bar{\sigma}_2\bar{\sigma}_3\bar{\sigma}_4$
DTE reset request d_2	1 0	0 0	1 1	0 0	0 1	1 0	$\bar{\sigma}_2\sigma_3$
DTE clear request p_6	-	1	-	0	-	-	σ_2
DTE restart request r_1	0	0	0	1	-	-	σ_4

$s_2=\{$ 1 if the DTE has received a CALL CONNECTED packet; 0 otherwise

- context variable set V :

v_1: lower edge of the receive window of the DTE

v_2: P(S) of the next data packet to be sent

v_3: lower edge of the transmit window of the DTE

v_4: last P(R) emitted by DTE

v_5: last P(R) received in a DCE RNR packet; if no DCE RNR packet has been received, $v_5=8$ ([10])

v_6: number of SEND DATA commands in buffer η

v_7: number of RECEIVE DATA commands in buffer η

- global state set Z :

$z_1=\{$ 1 if $v_2-v_3<W$, where W is the window size; 0 otherwise

$z_2=\{$ 1 if $v_5=8$; 0 otherwise

$z_3=\{$ 1 if $v_6>0$; 0 otherwise

$z_4=\{$ 1 if $v_7>0$; 0 otherwise

$z_5=\{$ 1 if $v_7>W$; 0 otherwise

$z_6=\{$ 1 if $v_1>v_4$; 0 otherwise

$z_7=\{$ 1 if PU has received a WAIT FOR CALL command; 0 otherwise ([11])

$z_8=\{$ 1 if PU has received a SEND INTERRUPT command; 0 otherwise

$z_9=\{$ 1 if PU has received a RECEIVE INTERRUPT command; 0 otherwise

It is possible to represent the DTE states by means of logical functions of the state variables previously defined. Since these are not indipendent, the logical functions can have a simplified form. In table 1 the logical functions of the DTE states are shown (indicated with the same names of the corresponding logical channel states of Recommendation X.25). Let us note that the sets Σ and S are enough to describe the X.25 DTE states. Moreover the following logical functions have interest :

p_{10} (ready without waiting) : $\bar{\sigma}_1\bar{s}_1\bar{\sigma}_2\bar{\sigma}_4\bar{z}_7$

p_{11} (ready with waiting) : $\bar{\sigma}_1\bar{s}_1\bar{\sigma}_2\bar{\sigma}_4 z_7$

$p_4=d_1 \cup d_2$ (data transfer) : $D\bar{\sigma}_2\bar{\sigma}_4$

([8]) More data and data qualifier bits can be included in the text.

([9]) Hyphens in table 1 mean that the states variables can be equal to 0 or 1 according to the previous state (see vectors h and g in app. B)

([10]) That because we have considered P(R) and P(S) to be modulo 8.

([11]) Let us note that z_7, z_8 and z_9 are examples of secundary state variables.

$p_2Up_4 : C\bar{\sigma}_2\bar{\sigma}_4$ $\qquad p_1Up_2Up_4 : \bar{\sigma}_2\bar{\sigma}_4$

$d_1Up_6 : D\bar{\sigma}_3\bar{\sigma}_4$ $\qquad p_1Up_2Up_4Up_6 : \bar{\sigma}_4$

$p_4Up_6 : D\bar{\sigma}_4$ $\qquad p_2Up_4Up_6 : C\bar{\sigma}_2\bar{\sigma}_4+\sigma_2\bar{\sigma}_4$

where $D=\sigma_1 s_2+s_1$ and $C=\sigma_1+s_1$.

Input context

There are two IC procedures, one for input envelopes μ and the other one for input command η. These procedures are reported in app. A using an Algol-like language.

The first one compares the P(S) and/or P(R) in the input envelopes (DCE DATA, DCE RR and DCE RNR) with the current values of the context variables. If P(S) and/or P(R) are within suitable ranges, IC transmits to PPU the relative op-code with /0, otherwise with /1. In this last case, if the state of DTE is flow control ready, a reset procedure occurs [1, sect. 3.4.1.2]. At the reception of DCE DATA/1, DCE RR/1 and DCE RNR/1, PPU performs the same action and therefore we have summarized these IC envelopes in table 1 under the name DCE/1. The other envelopes have no parameters and therefore are transmitted unchanged to PPU.

The second procedure does not transmit to PPU WAIT FOR CALL and RECEIVE INTERRUPT commands (state commands) [12], it transmits SEND DATA and RECEIVE DATA if the PU state is suitable, whereas it always transmits the other commands (operate commands). In table 2 the IC command set is shown; of course the

([12]) The state commands are not transmitted to PPU because in this case they act only on global states and/or context variables.

Tab. 2 - Inputs and outputs of PPU

IC envelopes μ_c	*Output op-codes* $\underline{m}'$
μ_{c1}=INCOMING CALL	$\underline{m}'_1$=CALL REQUEST
μ_{c2}=CALL CONNECTED	$\underline{m}'_2$=CALL ACCEPTED
μ_{c3}=CLEAR INDICATION	$\underline{m}'_3$=CLEAR REQUEST
μ_{c4}=DCE CLEAR CONFIRMATION	$\underline{m}'_4$=DTE CLEAR CONFIRMATION
μ_{c5}=DCE DATA/0	$\underline{m}'_5$=DTE DATA
μ_{c6}=DCE INTERRUPT	$\underline{m}'_6$=DTE INTERRUPT
μ_{c7}=DCE INTERRUPT CONFIRMATION	$\underline{m}'_7$=DTE INTERRUPT CONFIRMATION
μ_{c8}=DCE RECEIVE READY/0	$\underline{m}'_8$=DTE RECEIVE READY
μ_{c9}=DCE RECEIVE NOT READY/0	$\underline{m}'_9$=DTE RECEIVE NOT READY
μ_{c10}=RESET INDICATION	$\underline{m}'_{10}$=RESET REQUEST
μ_{c11}=DCE RESET CONFIRMATION	$\underline{m}'_{11}$=DTE RESET CONFIRMATION
μ_{c12}=RESTART INDICATION	$\underline{m}'_{12}$=RESTART REQUEST
μ_{c13}=DCE RESTART CONFIRMATION	$\underline{m}'_{13}$=DTE RESTART CONFIRMATION
μ_{c14}=DCE/1	
IC commands η_c	*Output op-codes* c'
	c'_1=INCOMING CALL
	c'_2=CALL CONNECTED
	c'_3=CLEAR
η_{c1}=CALL REQUEST	c'_4=CLEAR REQUEST
η_{c2}=CLEAR REQUEST	c'_5=DCE DATA
η_{c3}=SEND DATA	c'_6=DCE INTERRUPT
η_{c4}=RECEIVE DATA	c'_7=DCE INTERRUPT CONFIRMATION
η_{c5}=SEND INTERRUPT	c'_8=RESET
η_{c6}=RESET REQUEST	c'_9=RESET REQUEST
η_{c7}=RESTART REQUEST	c'_{10}=RESTART

Tab. 3 - Actions of PPU at the reception of IC envelopes μ_c.

DTE states \ IC envelopes	1 INCOM CALL	2 CALL CONN.	3 CLEAR INDIC.	4 DCE CLEAR CONF.	5 DCE DATA /0	6 DCE INTER.	7 DCE INTER. CONF.	8 DCE RR/0	9 DCE RNR/0	10 RESET INDIC.	11 DCE RESET CONF.	12 REST. INDIC.	13 DCE REST. CONF.	14 DCE/1
Ready p_1 — p_{10}	$\underline{N}$ (3) $\underline{m}'_3$ c'_1 $\rightarrow p_6$	$\underline{E}$ (1) $\underline{m}'_3$ $\rightarrow p_6$	N $\underline{m}'_4$	$\underline{E}$ (1)	WM	$\underline{E}$ (1)	$\underline{E}$ (1)	WM	WM	$\underline{E}$ (1)	$\underline{E}$ (1)	N $\underline{m}'_{13}$ c'_{10} $\rightarrow p_1$	$\underline{E}$ (11) $\underline{m}'_{12}$ c'_{10} $\rightarrow r_1$	$\underline{E}$ (1)
Ready p_1 — p_{11}	N $\underline{m}'_2$ c'_1 $\rightarrow d_1$													
DTE waiting p_2	N	N c'_2 $\rightarrow d_1$	N $\underline{m}'_4$ c'_3 $\rightarrow p_1$	$\underline{E}$ (2)	WM	$\underline{E}$ (2)	$\underline{E}$ (2)	WM	WM	$\underline{E}$ (2)	$\underline{E}$ (2)	N $\underline{m}'_{13}$ c'_{10} $\rightarrow p_1$	$\underline{E}$ (11) $\underline{m}'_{12}$ c'_{10} $\rightarrow r_1$	$\underline{E}$ (2)
Data transfer p_4 — d_1	$\underline{E}$ (5)	$\underline{E}$ (5)	N $\underline{m}'_4$ c'_3 $\rightarrow p_1$	$\underline{E}$ (4) $\underline{m}'_3$ c'_4 $\rightarrow p_6$	$\underline{N}$ + +	N + +	N +	N +	N	N $\underline{m}'_{11}$ c'_8	$\underline{E}$ (10) $\underline{m}'_{10}$ c'_9 $\rightarrow d_2$	N $\underline{m}'_{13}$ c'_{10} $\rightarrow p_1$	$\underline{E}$ (11) $\underline{m}'_{12}$ c'_{10} $\rightarrow r_1$	N $\underline{m}'_{10}$ c'_9 $\rightarrow d_2$
Data transfer p_4 — d_2					WM	N +		WM	WM	N c'_8 $\rightarrow d_1$	N c'_8 $\rightarrow d_1$			N
DTE clear request p_6	N-E	N-E	N c'_3 $\rightarrow p_1$	N c'_3 $\rightarrow p_1$	N +	N-E	N-E +	N	N	N-E	N-E	N $\underline{m}'_{13}$ c'_{10} $\rightarrow p_1$	$\underline{E}$ (11) $\underline{m}'_{12}$ c'_{10} $\rightarrow r_1$	N-E
DTE restart request r_1	N-E	N-E	N	N-E	WM	N-E	N-E	WM	WM	N-E	N-E	N c'_{10} $\rightarrow p_1$	N c'_{10} $\rightarrow p_1$	N-E

N : event which can occur also without errors
E : event caused by an error
N-E: N or E according to the previous DTE state
WM : without meaning
_ : actions not explicitly foreseen by X.25
(.): the figures in the round brackets refer to the corresponding hypothesis in the text
\+ : see matrices P or G

commands which are not transmitted to PPU do not appear.

Proper processing unit

In appendix B the matrices Q,H,P,G and the vectors h and g are shown.

Matrix Q (table 4) gives the output message op-code at the reception of an IC command η_c. Let us note that the RECEIVE DATA command can produce a DTE RR op-code. This occurs when in absence of SEND DATA commands in buffer η the number v_7 of RECEIVE DATA commands in the same buffer is less than the window size W.

In table 3 the actions of the PPU at the reception of an IC envelope μ_c are shown. From it it is possible to deduce:

- whether a given reception has been produced by an error;
- whether the actions taken by PPU are foreseen by Recommendation X.25.

A great many of the entries of matrices P and G (tables 6 and 7) are deductible from table 3 ([13]). There are some other entries which cannot be deduced from table 3. This occurs because the global states do not appear in this table.

Let us note that DTE RNR op-codes are never transmitted (see matrices Q and P) because, in accordance with [4], the DTE does not have the option reject. In fact the RNR packets are no more effective than a non progression of the DCE window.

Output context

OC adds the parameters P(S) and/or P(R) to DTE DATA, DTE RR and DTE RNR output message op-codes coming from PPU (see app. C). Moreover OC, as IC, can update the context variables and/or global states.

Internal mechanisms

To avoid the interlocutor PU remains indefinitely in a given state, it is convenient to associate a time-out to each pricipal state of the DTE. We have foreseen the following time-outs:

TO1 : is related to the DTE waiting state (p_2) and emulates the CLEAR REQUEST command;

TO2 : is related to the DTE clear request state (p_6) and emulates the CLEAR REQUEST command; after N2 repetitions a command is given to local controller to notify this situation (DCE out-order);

TO3 : is related to the flow control ready state (d_1) if the lower edges of the transmit and receive windows are not proceeding and it emulates the RESET REQUEST command;

TO4 : is related to the DTE reset request state (d_2) and emulates the RESET REQUEST command; after this it emulates a CLEAR REQUEST command [4];

TO5 : is related to the DTE restart request state (r_1) and emulates the RESTART REQUEST command; after N5 repetitions a command is given to local controller to notify this situation (DCE out-order).

([13]) For example at the reception of a CLEAR INDICATION, PPU answer with a DTE CLEAR CONFIRMATION op-code if the DTE state is p_1 or p_2 or p_4. Then the corresponding entry of the matrix P is the logical function of $p_1Up_2Up_4 : \bar{\sigma}_2\bar{\sigma}_4$.

6. CONCLUSIONS

In this paper we have proposed a formalization of the DTE packet level in the CCITT X.25 Recommendation, as an interlocutor of the theory of colloquies. This formal description has allowed us to point out some ambiguities of the DTE procedure, to overcome which suitable hypotheses have need to be introduced. Moreover this formalization is an useful tool for the software implementation because it not only describes the procedure, but also takes into account the commands exchanged with the upper and lower levels. For example, it has been used as starting-point for the CREI implementation of the DTE level 3 [9].

APPENDIX A

IC procedure for input envelopes

Begin if DCE DATA
 then if $v_3\leqslant P(R)\leqslant v_2$ *and* $v_1\leqslant P(S)\leqslant v_1+W-1$ *and* $D\bar{\sigma}_3\bar{\sigma}_4=1$
  then transmit(DCE DATA/0);
   $v_3=P(R)$;
   if $P(S)=v_1$
    then $v_1=P(S)+1$;
    else if there is a $P(S)_m$ ([14])
     then $v_1=P(S)_m+1$;
     else store(P(S) ;
  else transmit(DCE/1);
if DCE RR
 then if $v_3\leqslant P(R)\leqslant v_2$ *and* $D\bar{\sigma}_3\bar{\sigma}_4=1$
  then transmit(DCE RR/0);
   $v_3=P(R)$, $v_5=8$;
  else transmit(DCE/1);
if DCE RNR
 then if $v_3\leqslant P(R)\leqslant v_2$ *and* $D\bar{\sigma}_3\bar{\sigma}_4=1$
  then transmit(DCE RNR/0);
   $v_3=P(R)$, $v_5=P(R)$; ([15])
  else transmit(DCE/1);
if INCOMING CALL *and* $\bar{\sigma}_1\bar{s}_1\bar{\sigma}_2\bar{\sigma}_4 z_7=1$ *or*
 CALL CONNECTED *and* $\sigma_1\bar{\sigma}_2\bar{s}_2=1$ *or*
 DCE RESET CONFIRMATION *and* $\bar{\sigma}_2\sigma_3=1$
 then $v_7=R$; ([16])
if RESET INDICATION *and* $\bar{\sigma}_2\sigma_3=1$
 then $v_1=0$, $v_2=0$, $v_3=0$, $v_4=0$, $v_6=0$;
  $v_7=R$, $z_8=0$, $v_5=8$;
if CLEAR INDICATION *or* RESTART INDICATION *or*
 DCE RESTART CONFIRMATION
 then $v_1=0$, $v_2=0$, $v_3=0$, $v_4=0$;
  $v_6=0$, $v_7=0$, $v_5=8$, $z_8=0$;
if other envelope
 then transmit(envelope);
end

IC procedure for input commands

Begin if WAIT FOR CALL *and* $\bar{\sigma}_1\bar{s}_1\bar{\sigma}_2\bar{\sigma}_4=1$
 then $z_1=1$;
 if RECEIVE INTERRUPT *and* $\sigma_2\bar{\sigma}_4+C\bar{\sigma}_2\bar{\sigma}_4=1$
 then $z_9=1$;

([14]) $P(S)_m$ is the maximum P(S) received by DTE in a continuous sequence of data packets; note that $P(S)_m$ can be different from the last P(S) received

([15]) That because of hypothesis 6.

([16]) R is a number greater than W.

```
if RECEIVE DATA and Dσ̄2σ̄3σ̄4=1
   then v7=v7+1;
      if z̄5z6=1
         then v4=v4+1;
            if z3=0
               then transmit(RECEIVE DATA);
if SEND DATA and Cσ̄2σ̄4=1
   then v6=v6+1;
      if v6=1
         then transmit(SEND DATA);
if other command
   then transmit(command);
end
```

APPENDIX B

Vector h

$$h_1 = (\eta_{c1}\bar\sigma_1\bar s_1\bar\sigma_2\bar\sigma_4 + \sigma_1)\cdot[\bar\mu_{c3}(\overline{\mu_{c4}\sigma_2})\bar\mu_{c12}\,\bar\mu_{c13}\,\bar\eta_{c7}]$$

$$h_2 = (\eta_{c2}\bar\sigma_2\bar\sigma_4 + \mu_{c1}\bar\sigma_1\bar s_1\bar\sigma_2\bar\sigma_4\bar z_7 + \mu_{c2}\bar\sigma_1\bar s_1\bar\sigma_2\bar\sigma_4 + \mu_{c4}D\bar\sigma_2\bar\sigma_4 + \sigma_2)\cdot[\bar\mu_{c3}(\overline{\mu_{c4}\sigma_2})\bar\mu_{c12}\,\bar\mu_{c13}\,\bar\eta_{c7}]$$

$$h_3 = (\eta_{c6}D\bar\sigma_2\bar\sigma_3\bar\sigma_4 + \mu_{c11}\,D\bar\sigma_2\bar\sigma_3\bar\sigma_4 + \mu_{c14}\,D\bar\sigma_2\bar\sigma_3\bar\sigma_4 + \sigma_3)\cdot[\bar\mu_{c3}(\overline{\mu_{c4}\sigma_2})\bar\mu_{c10}\,(\overline{\mu_{c11}\,\sigma_3})\bar\mu_{c12}\,\bar\mu_{c13}\,\bar\eta_{c7}]$$

$$h_4 = (\eta_{c7}\bar\sigma_4 + \mu_{c13}\,\bar\sigma_4 + \sigma_4)\cdot[\bar\mu_{c12}\,(\overline{\mu_{c13}\,\sigma_4})]$$

Vector g

$$g_1 = (\mu_{c1}\bar\sigma_1\bar s_1\bar\sigma_2\bar\sigma_4 z_7 + s_1)\cdot[\bar\mu_{c3}(\overline{\mu_{c4}\sigma_2})\bar\mu_{c12}\,\bar\mu_{c13}]$$

$$g_2 = (\mu_{c2}\sigma_1\bar\sigma_2\bar s_2 + s_2)\cdot[\bar\mu_{c3}(\overline{\mu_{c4}\sigma_2})\bar\mu_{c12}\,\bar\mu_{c13}]$$

Tab. 4 - Matrix Q.

IC commands / Output op-codes m'	1 CALL REQUEST	2 CLEAR REQUEST	3 SEND DATA	4 RECEIVE DATA	5 SEND INTERRUPT	6 RESET REQUEST	7 RESTART REQUEST
1 CALL REQUEST	$\bar\sigma_1\bar s_1\bar\sigma_2\bar\sigma_4$	–	–	–	–	–	–
2 CALL ACCEPTED	–	–	–	–	–	–	–
3 CLEAR REQUEST	–	$\bar\sigma_4$	–	–	–	–	–
4 DTE CL. CONF.	–	–	–	–	–	–	–
5 DTE DATA	–	–	$D\bar\sigma_2\bar\sigma_3\bar\sigma_4 z_1 z_2$	–	–	–	–
6 DTE INTERRUPT	–	–	–	–	$D\bar\sigma_2\bar\sigma_3\bar\sigma_4 z_8$	–	–
7 DTE INT. CONF.	–	–	–	–	–	–	–
8 DTE RR	–	–	–	$D\bar\sigma_2\bar\sigma_3\bar\sigma_4\bar z_3\bar z_5$	–	–	–
9 DTE RNR	–	–	–	–	–	–	–
10 RESET REQ.	–	–	–	–	–	$D\bar\sigma_2\bar\sigma_4$	–
11 DTE RESET C.	–	–	–	–	–	–	–
12 RESTART REQ.	–	–	–	–	–	–	1
13 DTE RESTART C.	–	–	–	–	–	–	–

Tab. 5 - Matrix H.

IC commands / Output op-codes c'	1 CALL REQUEST	2 CLEAR REQUEST	3 SEND DATA	4 RECEIVE DATA	5 SEND INTERRUPT	6 RESET REQUEST	7 RESTART REQUEST
1 INCOMING CALL	–	–	–	–	–	–	–
2 CALL CONN.	–	–	–	–	–	–	–
3 CLEAR	–	–	–	–	–	–	–
4 CLEAR REQUEST	–	$\bar\sigma_2\bar\sigma_4$	–	–	–	–	–
5 DCE DATA	–	–	–	–	–	–	–
6 DCE INTERRUPT	–	–	–	–	–	–	–
7 DCE INT. CONF.	–	–	–	–	–	–	–
8 RESET	–	–	–	–	–	–	–
9 RESET REQ.	–	–	–	–	–	$D\bar\sigma_2\bar\sigma_3\bar\sigma_4$	–
10 RESTART	–	–	–	–	–	–	$\bar\sigma_4$

Tab. 6 - Matrix P.

IC envelopes / Output op-codes m'	1 INC. CALL	2 CALL CONN.	3 CLEAR IND.	4 DCE CLEAR CONF.	5 DCE DATA/O	6 DCE INTER.	7 DCE INTEP. CONF.	8 DCE RR/O	9 DCE RNR /O	10 RESET IND.	11 DCE RESET CONF.	12 REST. IND.	13 DCE REST. CONF.	14 DCE/I
1 CALL REQUEST	–	–	–	–	–	–	–	–	–	–	–	–	–	–
2 CALL ACCEPTED	$\bar\sigma_1\bar s_1\bar\sigma_2\bar\sigma_4\cdot z_7$	–	–	–	–	–	–	–	–	–	–	–	–	–
3 CLEAR REQUEST	$\bar\sigma_1\bar s_1\bar\sigma_2\bar\sigma_4\cdot\bar z_7$	$\bar\sigma_1\bar s_1\bar\sigma_2\bar\sigma_4$	–	$D\bar\sigma_2\bar\sigma_4$	–	–	–	–	–	–	–	–	–	–
4 DTE CLEAR CONFIRMATION	–	–	$\bar\sigma_2\bar\sigma_4$	–	–	–	–	–	–	–	–	–	–	–
5 DTE DATA	–	$\sigma_1\bar\sigma_2\bar s_2 z_3$	–	–	$D\bar\sigma_2\bar\sigma_3\bar\sigma_4\cdot z_1 z_2 z_3$	–	–	$D\bar\sigma_2\bar\sigma_3\bar\sigma_4 z_1 z_2 z_3$	–	–	$\bar\sigma_2\sigma_3 z_3$	–	–	–
6 DTE INTERRUPT	–	–	–	–	–	–	–	–	–	–	–	–	–	–
7 DTE INTERRUPT CONFIRMATION	–	–	–	–	–	$D\bar\sigma_2\bar\sigma_3\bar\sigma_4\cdot z_9$	–	–	–	–	–	–	–	–
8 DTE RR	–	–	–	–	$D\bar\sigma_2\bar\sigma_3\bar\sigma_4\cdot\overline{z_1 z_2 z_3} z_5$	–	–	–	–	–	–	–	–	–
9 DTE RNR	–	–	–	–	–	–	–	–	–	–	–	–	–	–
10 RESET REQUEST	–	–	–	–	–	–	–	–	–	–	$D\bar\sigma_2\bar\sigma_3\bar\sigma_4$	–	–	$D\bar\sigma_2\bar\sigma_3\bar\sigma_4$
11 DTE RESET CONFIRMATION	–	–	–	–	–	–	–	–	–	$D\bar\sigma_2\bar\sigma_3\bar\sigma_4$	–	–	–	–
12 RESTART REQUEST	–	–	–	–	–	–	–	–	–	–	–	–	$\bar\sigma_4$	–
13 DTE RESTART CONFIRMATION	–	–	–	–	–	–	–	–	–	–	–	$\bar\sigma_4$	–	–

Tab. 7 - Matrix G.

IC envelopes / Output op-codes c'	1 INC. CALL	2 CALL CONN.	3 CLEAR IND.	4 DCE CLEAR CONF	5 DCE DATA/O	6 DCE INTER.	7 DCE INTER. CONF.	8 DCE RR/O	9 DCE RNR/O	10 RESET IND.	11 DCE RESET CONF.	12 REST. IND.	13 DCE REST. CONF.	14 DCE/I
1 INCOMING CALL	$\bar\sigma_1\bar s_1\bar\sigma_2\bar\sigma_4$	–	–	–	–	–	–	–	–	–	–	–	–	–
2 CALL CONNECTED	–	$\sigma_1\bar\sigma_2\bar s_2$	–	–	–	–	–	–	–	–	–	–	–	–
3 CLEAR	–	–	$C\,\bar\sigma_2\bar\sigma_4 + \sigma_2\bar\sigma_4$	σ_2	–	–	–	–	–	–	–	–	–	–
4 CLEAR REQUEST	–	–	–	$D\bar\sigma_2\bar\sigma_4$	–	–	–	–	–	–	–	–	–	–
5 DCE DATA	–	–	–	–	$D\bar\sigma_3\bar\sigma_4 z_4$	–	–	–	–	–	–	–	–	–
6 DCE INTERRUPT	–	–	–	–	–	$D\bar\sigma_4 z_9$	–	–	–	–	–	–	–	–
7 DCE INTERRUPT CONFIRMATION	–	–	–	–	–	–	$D\bar\sigma_4 z_8$	–	–	–	–	–	–	–
8 RESET	–	–	–	–	–	–	–	–	–	$D\bar\sigma_2\bar\sigma_4$	$\bar\sigma_2\sigma_3$	–	–	–
9 RESET REQUEST	–	–	–	–	–	–	–	–	–	–	$D\bar\sigma_2\bar\sigma_3\bar\sigma_4$	–	–	$D\bar\sigma_2\bar\sigma_3\bar\sigma_4$
10 RESTART	–	–	–	–	–	–	–	–	–	–	–	1	1	–

APPENDIX C

OC procedure to compute P(S) in DTE DATA envelopes
Begin $P(S)=v_2$;
    $v_2=v_2+1$;
    $v_6=v_6-1$;
end

OC procedure to compute P(R) in output envelopes
Begin if $z_5=1$
    then $P(R)=v_1$;
        $v_4=v_1$;
    else $P(R)=v_4$;
end

OC procedure for output op-codes *m'*
Begin if DTE INTERRUPT
    then $z_8=1$;
  if RESET REQUEST
    then $v_1=0$, $v_2=0$, $v_3=0$, $v_4=0$, $v_6=0$;
        $v_7=0$, $v_5=8$, $z_8=0$;
end

OC procedure for output op-codes c'
Begin if DCE DATA
    then $v_7=v_7-1$;
  if DCE INTERRUPT
    then $z_9=0$;
  if DCE INTERRUPT CONFIRMATION
    then $z_8=0$;
end

ACKNOWLEDGEMENTS

The authors wish to thank Prof. G. Le Moli for the helpful discussions and suggestions.

The paper was first received on February 24, 1978.

REFERENCES

[1] CCITT: *Recommendation X.25*. Orange Book, vol. VII-2, pp. 70-108, 1977.

[2] IBM Europe: *Technical comments regarding CCITT Recommendation X.25*.Com. VII-N.39, November 1976.

[3] A. M. Rybczynski: *Questions on X25 and on characteristics of virtual circuits*. December 1976.

[4] CCETT: *Position des PTT sur la contribution IBM au CCITT (Com. VII-N. 39)*. TPC/R/DT/22/77/MM.

[5] CCITT: *Recommendation X.25-Extensions, points requiring clarification and points for further study proposed by the editor's group*. Com. VII N. 54, part III, annex 1.

[6] G. Le Moli: *A Theory of Colloquies*. First European Workshop on Computer Networks, Arles, April-May 1973, pp. 153-173. Also:"Alta Frequenza", vol.XLII, N.10, October 1973, pp. 493-500.

[7] A. Danthine, J. Bremer: *Définition, représentation et simulation des protocoles dans un contexte réseaux*. Journ. Itern. Mini-ordinateurs et Transm. de Données, AIM, Liège, January 1975, pp. 115-126.

[8] A. Danthine, J. Bremer: *An axiomatic description of the transport protocol of Cyclades*. Professional Conference on Computer Networks and Teleprocessing, Aachen, March 31- April 2, 1976

[9] G. Le Moli et alii: *The architecture of the software implemented by CREI for EIN*. This issue.

RPCNET: status and trends

F. Caneschi, E. Ferro, L. Lenzini, M. Martelli, C. Menchi, M. Sommani, F. Tarini

CNUCE, Institute of the Italian National Research Council (CNR) PISA

ABSTRACT

Status and trends of the REEL Project Computer NETwork (RPCNET) are described. The project partners are first introduced and a brief description of the architecture follows. The specific hardware and software to which the network is committed is also described. The main part of the paper is dedicated to the services which are at the moment available to the RPCNET user community. An outline of the RPCNET trends follows.

1. INTRODUCTION

REEL (REte di ELaboratori), a project to investigate concepts and experimental solutions for distributed processing problems, was formally established in June 1974 (Lenzini [1], Caneschi [2]), as a collaboration among the following Italian institutions:

- CNEN, Division for the Management of Information Systems, Bologna.
- CNR, CNUCE Institute, Pisa.
- CSATA, Center of Studies for Advanced Technological Applications, Bari.
- IBM, Scientific Center, Pisa.
- University of Padua, Computing Center.
- University of Turin, Computing Center.

The result of this project, which lasted four years, was the design and the implementation of a packet switching distributed control network named RPCNET (REEL Project Computer NETwork).

The RPCNET design was determined by the principal requirements of the REEL project partners:

a) Minimum of additional hardware

The minimum hardware required to get into the network constituted by the appropriate features to drive telecommunication lines at the convenient speed (full duplex binary synchronous from 1200 to 9600 bps) and by a (single) intelligent processor on which to implement the RPCNET functions. The utilization of a separate processor (a Front End Processor or a Telecommunication Processor) is not mandatory.

b) Partner independence

Each center should retain control of any resources placed at the disposal of the other network partners. Thus, each center can dynamically attach to/or detach from the network. The network should thus be able to reconfigure itself in a highly automatic way.

c) Minimum impact on the existing operating systems

The introduction of the new network capabilities into the operating system of each participating center should neither disrupt the user community nor the system maintenance but should act as a straightforward extension of the centers' existing concepts and facilities.

d) The network should not be limited only to terminal handling and/or to Remote Job Entry traffic

The network architecture should be based on the concept that the fundamental communication across the network is a process-to-process information exchange. The terminal-to-application and job transfer data communication, also provided by RPCNET, should be mapped on this symmetric process-to-process capability.

From the above requirements the following architectural characteristics were derived:
-Packet switching, distributed control communication Subnetwork
-Availability of a Communication System that can be accessed via a generalized Access Method
-Functional Layering allowing a variety of physical node configurations
In view of the RPCNET characteristics outlined above, REEL project activities can be grouped into three fundamental areas:

1) Study and implementation of a generalized Communication System. This system results from the union of the software subsystems and hardware components distributed in all the network Nodes. These subsystems and components interact and cooperate through Connections represented by Data Links or Local Attachments.

2) Definition and implementation of an access method to the Communication System, called RNAM (REEL Network Access Method), with the objective of broadening the spectrum of Applications accessing the network.

3) Implementation of some Applications with the aim of both providing friendly access to the network services for inexperienced users and of testing the performance characteristics of RPCNET, thus validating its functional design.

It is worthwhile underlining that RPCNET was sponsored by the above mentioned Italian institutions because they strongly felt the urgency and importance of promoting, within an operational environment, an Italian experiment in the networking area.

2. NETWORK ARCHITECTURE

RPCNET is conceptually divided into three Layers as shown in Fig.1:

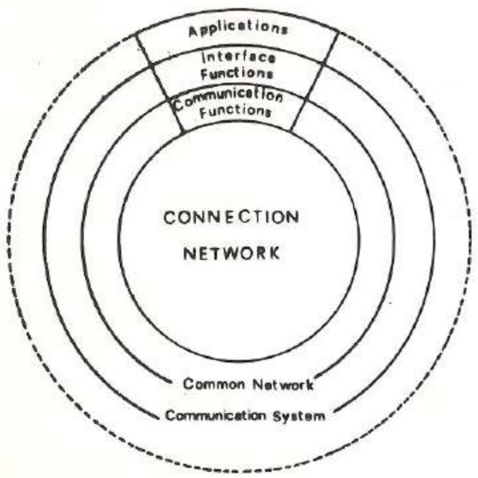


Fig. 1

2.1 Common Network

This Layer provides the packet switching capability by using a physical communication medium together with routing and reconfiguration strategies and a well defined packet format. The Common Network does not guarantee that packets will be delivered in the same order as that in which they were sent, nor does it guarantee that there will be no loss or duplication of packets. These can be caused by various factors such as the use of alternative routes between Nodes for packets in case of failure, and their retransmission on data links when errors are detected. As shown in Fig.1, the Common Network is structured into a Connections Network and a Communication Functions sub-Layer. The Connections Network provides the physical communication medium to transport data from one processor to an adjacent one. The Connections Network is made up of full duplex circuits called Connections. These can be point-to-point synchronous data links or channel attachment subchannels. The Communication Functions sub-Layer has the internal structure shown in Fig.2. The Network Connection Handler (NCH) is in charge of sending and receiving packets between two adjacent Nodes. All NCHs have three control phases: a)-the Connection making phase for establishing a usable Connection; b)-the data transfer phase, whose protocol details are hidden from the NCH user; and c)-the Connection release phase. The control procedures for the data transfer phase are independent of the other two phases. For synchronous data links, NCH employs a logical protocol which is designed to detect transmission errors and to invoke retransmission of packets when any errors are detected. NCH supports an input queue for each Connection from which packets are taken for transmission. The Packet Switcher routes packets towards their destinations by using a routing table that contains the corresponding NCH input queue for each destination. When a packet reaches its destination Node, it is passed to the Interface Function Layer for further analysis. The Common Network Manager (CNM) manages and updates the routing table for the Node. For this purpose, packets are sent among CNMs in order to propagate topology changes through the network. Topology changes take place when activation and de-activation of Connections or Nodes occur. Obviously this is either a consequence of a network operator intervention or of failure conditions.

2.2 Interface Functions Layer

This Layer provides the interface between the Common Network and the Application Layer (the Layer in which the network users reside). The Interface Functions basically provide:

a)-the defining and undefining of ports (or Logical Units/LUs) on the Communication System;

Caneschi F. et al. — RPCNET: Status and trends

b)-extemporaneous services such as query and mailing;

c)-setting up and closing down of Logical Channels between Applications.

These functions are performed by the Network Services Manager (NSM) (Fig.2). The QMSG routine (Fig.2) controls the NSM communication protocol. Once established, Logical Channels are maintained by a software component, the Session Handler (SH) (Fig.2) which is in turn made up of three modules; Presentation Services (PR), Data Length Adapter (DLA) and Data Flow Control (DFC). The DFC provides a full duplex packet oriented protocol which is associated with an LU, in session with another LU in the network.

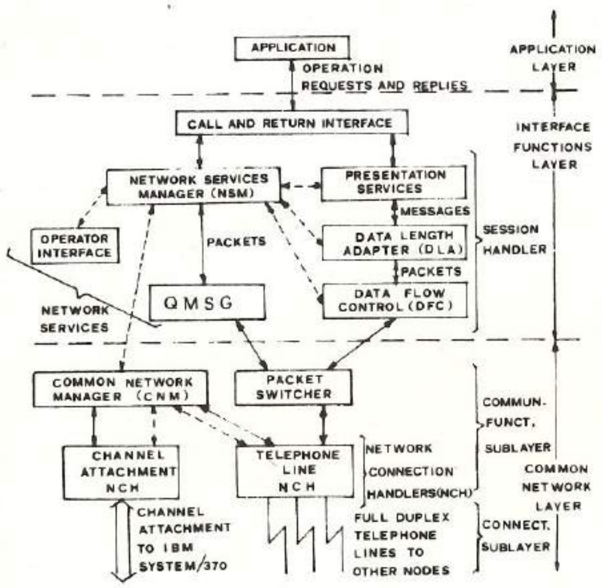


Fig. 2

The DFC has a window oriented protocol that filters out duplicates, puts packets into sequence for the DLA, detects packet loss and reports this to the DLA and provides for pacing. The DLA presents a full duplex message interface to the PR module. The function of the DLA is to mask the packeting activity of the lower levels and to report message loss to the PR. The design of the Session Handler is intended to optimize the transmission of message sequences. Messages are normally transmitted in one direction at a time, but there is the possibility of changing direction when requested by the sender, and of sending, at any time, packet size messages in the direction opposite to the "normal" direction. The PR provides additional protocol facilities, such as the "chaining" of sequences of messages. The only errors passed on to an Application are: a)-lack of local free storage for performing an operation; b)-loss of contact with the Application at the other end of the Logical Channel; c)-message loss; and d)-Application violation of the chaining protocol. It is worth noting, at this point, that message loss can only occur if a Node somewhere in the network crashes while holding the only copy in the network of a packet of the message. The Session Handler provides for error detection in the case of message loss as well as reordering incoming messages in the correct sequence. Applications must accomplish error recovery when a message is lost. It is normally easier for the sender, rather than the receiver, to recover lost messages. The packets that cross the boundary between the Interface Functions and the Common Network are sent and received from specific addresses in a network-wide address space. They are either travelling between NSM's or they are travelling between LU's which are joined by a Logical Channel. Along the Logical Channel, Applications exchange units of information called BIUs (Basic Information Units), which consist of two parts: the RU (Request/Response Unit) containing that part of the information which is the object of the communication and is transparent to the Communication System and the RH (Request/Response Header) which contains indications on the modalities for use and status of the Logical Channel. It should be noted that:

a)-RU is limited in size and

b)-RU data integrity and sequentiality are maintained.

The combination of the Interface Functions Layer and the Common Network constitutes the Communication System. The characteristics of a Logical Channel can be summarized as follows:

a)-it is driven with a half-duplex technique;

b)-it does not provide an error-free Connection;

c)-error (loss of RU) is detected and signalled at both ends of the Logical Channel.

2.3 Applications Layer

This most external Layer of the network contains the original sources and ultimate destinations of the information. The term Application is used here to indicate the generic network end user. More precisely, an Application is defined as any process or set of coordinated processes that access the Communication System in order to obtain network services. Towards the external boundary, Applications can be directly or indirectly attached to one or more end user physical devices which represent the original sources or final destinations of the information. These devices, if present, are also included in the concept of the term "Application".

3. NETWORK CONTROL AND RNAM

Two software components, already defined above, provide the control in each Node:

the Network Services Manager (NSM), within the Interface Functions, and the Common Network Manager (CNM) within the Communication Functions. Each shares control of its respective Layer in an equihierarchical way with the corresponding components in all other Nodes and represents an address in the network address space (Franchi [3]). This separation of control gives rise to a logical Node configuration that allows for a variety of physical Node configurations. The logical Node configuration is defined as follows: in a Node, there is one and only one Common Network Manager (CNM) and one, several, or even no Network Services Managers (NSM). By distributing the software Layers defined by the architecture on one or two processors, the physical Node configurations of Fig.3 are possible.

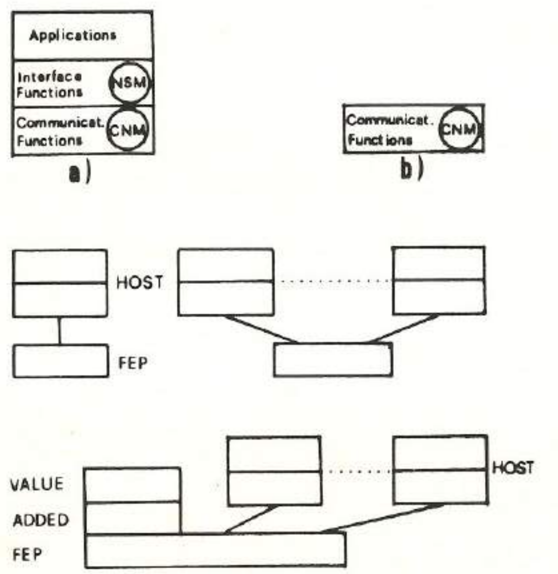


Fig. 3

According to the RPCNET terminology a network Connection is defined as a communication channel connecting any two processors both of which contain a CNM component, i.e. two network Nodes. Communication channels physically connecting distinct processors within the same Node are called Internal Connections. Telecommunication lines (or simply Lines) are instead communication channels used to attach devices to processors. Given the topological layout of the network hardware components, the Host role implies that a processor has a single Connection (Internal Connection) to a FEP. Data coming from many Applications, possibly associated with device Lines, are multiplexed/demultiplexed on this Connection. The routing, the Host multiplexing/demultiplexing and the Application multiplexing/demultiplexing stages correspond to the three-level address scheme of RPCNET. A zero value of the second and third address fields (Fig.4) identifies the addressable unit associated with the Common Network Manager (CNM). A zero value of the third address field identifies an NSM. A non-zero address is used to identify a network addressable unit associated with an Application.

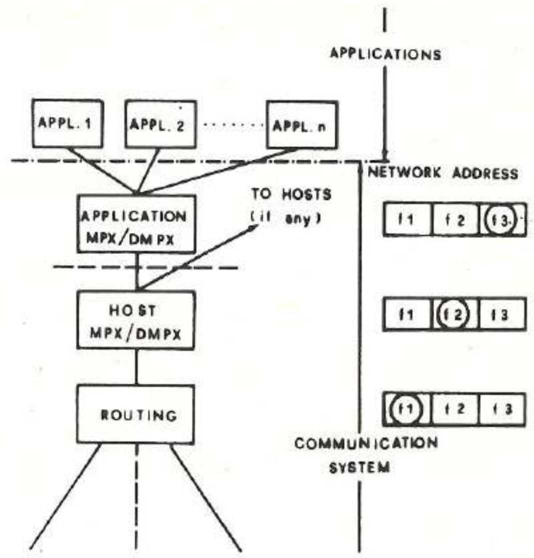


Fig. 4

This type of unit is called LCT (Logical Channel Termination). Before starting any network activity, an Application asks the Communication System for an LU (Logical Unit), using the RNAM macro instruction OPENLU. This allows the Application to contact the NSM component and via this component to send and receive messages and/or inquiries (MAIL and INQUIRE macros). When one Application wishes to connect to another, it issues a BIND operation. An Application wishing to be connected by a BIND request, must issue an INVITE operation. If the Application requested by the BIND exists and has issued an INVITE, the two Applications become addressable without the intervention of their respective NSMs. The SH (Session Handler) component carries out the in-Session information exchange task by executing the SEND, RECEIVE and BREAK macros issued by the Applications. The RPCNET Logical Channel allows only one operation to be specified at a time. SEND is used to send a message, which will be buffered at the receiving side (given sufficient storage) until a matching RECEIVE is issued by the Application at this side. Due to the nature of the half duplex Logical Channel, it is necessary to "change direction" before a receiver can send and a sender can receive. The sender is in charge of this direction change task. A BREAK can send a message (restricted in length to fit within a Common Network packet) from an Application in receive state to one that is in send state. The break message is received in an asynchronous fashion, and no special operation issued by the receiver is necessary. The TESTLC operation is used by an Application to control certain aspects of the Logical Channel error detection mechanism. In effect, it verifies whether the messages sent have arrived at the other end of the Logical Channel. The UNBIND and CLOSELU macros

are used respectively to release a Logical Channel and a Logical Unit.

4. EXPERIMENTAL IMPLEMENTATION

The network configuration on which the RPCNET software prototype is operating, as of July '79, is shown in Fig.5.

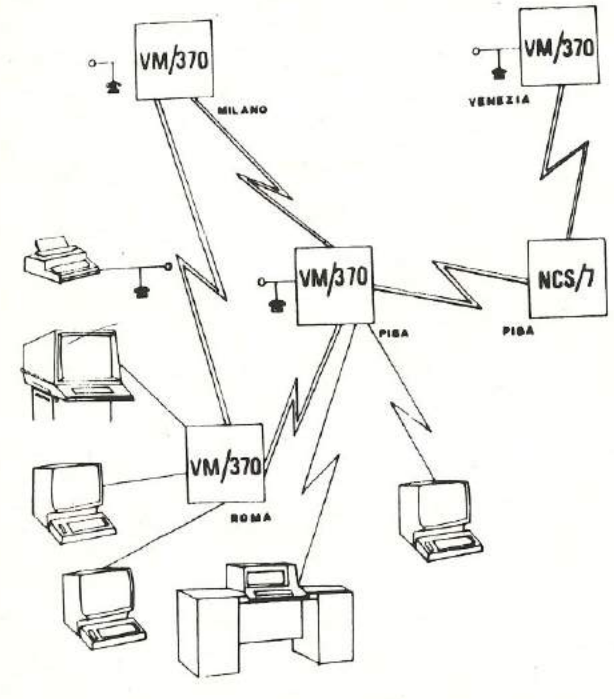


Fig. 5

The following institutions:

-IBM Scientific Center in Venice (1975);

-SIAM Laboratory of CNR in Milan (1977);

-SELTE; CNR Central Administration in Rome (1978);

have joined the project as users of the network facilities.

The central processors available at the Node locations are IBM System/370s, model 158 and 168, and IBM System/7 as Front End Processor (or FEP). The Operating System software on the System/370s is VM/370 in some Nodes and OS-VS (SVS/HASP4 or MVS/JES2) in the others. Connections between these processors are leased telephone lines. The System/370 Channel Attachment feature (Local Attachment) can be used between a System/7 and a System/370, when these are located back to back. This type of Connection allows a faster, parallel by byte, data transfer between processors. The software providing the Communication System services and the RNAM access facilities is CNS/VM under VM/370 (Fusi [4]) and CNS/VS under OS/VS (Gori [5]) (CNS stands for Computer Network Subsystem). Under VM/370, CNS is an Operating System running on a specialized virtual machine. The System/7 network software is an independent stand-alone system called NCS/7 (Network Control System for System/7). It is written in System/7 assembler language and is compatible with MSP/7, the IBM standard support. NCS/7 has been provided with its own Task, Storage, I/O and Command Management (Springer [6]). Each processor in the configuration of Fig.5 acts as Full Node (see Fig.3a). The Communication System software and RNAM are operational under CNS/VM, CNS/VS and NCS/7. This means that Applications running under the control of OS-VS, VM or NCS/7 can communicate with each other through Logical Channels by using RNAM. At the moment, the implementation of the RPCNET architecture or any improvement on it, is not available on the IBM Series/1 minicomputer.

Because the RPCNET network architecture is fully independent of the hardware and software on which it is implemented, various computing centers in Italy, using non-IBM computers, expressed their wish to implement RPCNET architecture on their machines. We were, however, reluctant to encourage this, because EURONET will become operational in 1979 and because of the position of the European PTTs with respect to non-public networks.

5. RPCNET SERVICES

The following services , which are at present all operational, were included with priority in the REEL project objectives:

a)-Interactive Terminal Access;
b)-Spool File Transfer;
c)-Remote File Access.

5.1 Interactive Terminal Access

For all interconnections of terminals to interactive Operating Systems through RPCNET, there will be a Terminal Application at the terminal side and a Host Application at the interactive system side. In RPCNET, two different approaches were considered to implement the Interactive Terminal Access. The first approach does not involve any modification of the Operating System code; the second, however, normally requires some modifications. In the first approach, an IBM System/7 Host Application drives a System/7 emulator of an IBM Start/Stop 370X/EP Communication Control Unit. The System/7 thus appears, to the interactive computer, to be a 370X/EP with start-stop terminals (Lazzeri [8]). The Operating System can support terminals via the network if the Operating System itself has software to access terminals via the 370X/EP. On the contrary, the second approach has Host and Terminal Applications residing in the VM/370 control program which are essentially additions to the VM/370 terminal access method. Fig.6 illustrates the logical separation of the various Application components and their interconnection via RPCNET. The Terminal Application is made up of a line driver that maps a real terminal into a Virtual Terminal (or VT), a network command processor, and a module for communicating with the Host Application via the RPCNET Logical Channel. The protocol for this communication is the Virtual Terminal Protocol, or VTP (Lazzeri [7]). The VM/370 Terminal Application could use the access methods already provided for the terminals supported by VM/370. The Host Application maps the Virtual Terminal Protocol into computer significant operations. For the System/7, the Application maps the Virtual Terminal Protocol into 370X/EP emulator operations.

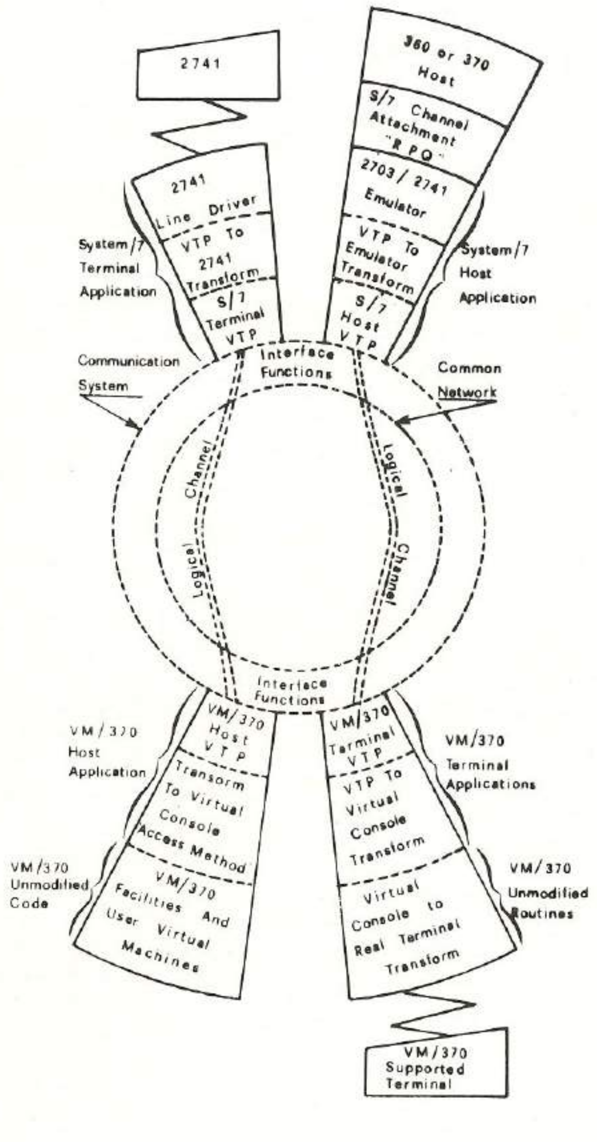


Fig. 6

For VM/370, the mapping is made into a virtual console for the virtual machines, at a level where both the VM control program and the user virtual machine access the virtual console. Codes have thus been added to the terminal access method of VM/370 so that the RPCNET Virtual Terminal is inserted into the system. Terminals, attached to the VM/370 system as virtual machine consoles, can connect to any other VM in the network by issuing a REELON command, followed by the specification of the destination system. After giving this command, users can LOGON into any virtual machine of the destination system. A similar capability is available for terminals attached to System/7. By issuing a REELON command users can reach any VM/370, either directly or via a System/7 Front End, or via any other IBM Operating System (supporting a 370X Communication Control Unit) via a System/7 Front End.

5.2 Spool File Exchange

The Spool File Exchange (Bertaina [9]) under VM is provided as an extension of the SPOOL and TAG console functions of the VM/370 Control Program. By using this extension, the VM/370 user at the virtual machine console can consider any spool file destination as a virtual machine running under the processor to which he is attached.

Under CMS, the VM/370 Conversational Monitor System, spool information is not limited to being logically associated to input card deck or output print lines, but a CMS user can put on the spool areas any information available on his private direct access space. CMS users have at their disposal a special command, SEND, which is expanded in the equivalent SPOOL and TAG console functions. VM users accessing local or remote card input devices can address any destination processor in the network by putting a TAG card in front of their card decks.

Under OS-VS operating systems, the HASP4 facilities were extended to allow the exchange of spool files, not only with other OS-VS systems but also with VM/370 Nodes.

A /*SEND card put in front of input card decks provides facilities analogous to those provided by a TAG card under VM, while a /*ROUTE card allows the routing of job outputs anywhere in the network. In particular, jobs to be executed under OS-VS can be acquired both from the network and from the local, internal or RJE workstation card readers. In the same way, spool files to be printed or punched or to be sent elsewhere through the network can be acquired without OS-VS execution or scheduling.

The end-to-end protocol adopted provides translation from VM to OS-VS spool format, and viceversa, for card image or print image data.

5.3 Remote File Access

The CMS user has at his disposal a set of commands by which he can attach to his virtual machine one or more minidisks defined in the directories of one or more virtual machines, which reside in remote hosts. When a remote device is attached, by issuing the NETATT command, it can be accessed and used just like a local one, i.e., a list of the files in that disk may be obtained, files may be read and edited,

renamed and erased, simply by issuing standard CMS commands. The approach allows different users to read and write in the same device simultaneously, without destroying or modifying the master file directory of the disk. A user may also access a disk in exclusive mode: in this case, other users cannot access the same device. If a user wants to reserve some files, and not the entire disk, he can do so by simply issuing the NETRES command from his virtual machine. A file reserved may be accessed by other users read/only, read/write, or cannot be accessed, depending on the reservation mode.

When the reservation of a file is no longer necessary, the user can issue the command NETREL.

Finally, when a user wants to detach the remote device, he can release it by a CMS command, and then issue the command NETDET, otherwise he issues NETDET directly: in both cases, the effect is the same.

6. CONCLUSIONS

RPCNET has been implemented almost on schedule and in accordance with the initial design (Lenzini [1]). The objectives and requirements formulated by the project partners have largely been met.

The Communication System, RNAM and the following services: Interactive Terminal Access, Spool File Transfer and Remote File Access are all operational. Other services, such as electronic mailing are now being implemented. The present network prototype can be used, either in its actual configuration or with certain extensions. In addition, independent replicas of RPCNET can be developed for the internal use of the project partners' organizations.

CNUCE's main objective for RPCNET in terms of network usage, is to turn the RPCNET prototype implementation into a network of Research Institutes of the Italian National Research Council (CNR) and possibly for the benefit of other scientific organizations. At the present SIAM (CNR Laboratory) and SELTE (CNR Central Administration) constitute, together with CNUCE the nucleus for such a network.

RPCNET facilitates research work providing cheap and effective computing services. It mainly offers a service bureau type of operation for unsophisticated users who want to use the RPCNET services as a tool in their research work. On the other hand, system programmers can use the RNAM access method in developing new network Applications.

Quite a large amount of data is already being exchanged between the network Nodes, because the CNR and IBM researchers located in the above mentioned centers are collaborating on common projects. Statistics and accounting facilities are also provided. In order to allow researchers to also access to facilities belonging to other networks, software which allows an interactive RPCNET user to connect to any other network supporting TTY terminals has been developed. This software was realized, in particular, as we had hoped to establish a connection between RPCNET and EURONET. Unfortunately, it is not yet clear whether the Italian PTT would permit this, even if RPCNET can continue to operate. We do believe, however, that RPCNET is now playing a very important role in the Italian scientific and technical community, by providing the user with the opportunity for practical computing network experience. In the future, it will be easy for such users to move into networks of the EURONET type as soon as they become operational.

In order to help traditional users to become familiar with RPCNET, courses have been organized at all of the computing centers which act as RPCNET Nodes. RPCNET user oriented documentation has also been published and widely distributed among the RPCNET user community.

A group of CNUCE System Programmers has been set up to perform these and other tasks e.g. consulting services, standardisation of new procedures, etc. Software bugs in the initial RPCNET prototype have already been fixed by this group. In addition, because the RPCNET software heavily stresses the Telecommunication section of the Operating System in which it is implemented, bugs in this section have been discovered and fixed by the manufacturer's system engineers.

Improvements in the original prototype software have already been made and others are foreseen in the near future. These improvements usually apply to the interaction between CNS and the corresponding Operating System, as timing effects become increasingly critical as the network traffic increases.

A number of other educational or research institutions in Italy, running VM/370, have expressed the wish to join RPCNET. At the moment, however, we do not want to increase the size of the network until we are sure RPCNET runs perfectly, even in heavy traffic conditions. We will then prefer to allow just one center at a time join RPCNET so that we can observe network behavior with the addition of a new Node and correct any bugs if necessary before considering the possibility of adding another center. It is also obvious that, as the network size augments, its management becomes increasingly complex.

Before concluding, it is worthwhile pointing out that the experience gained in the design and implementation of RPCNET has allowed CNUCE and the other partners in this project to give valuable contributions within national and international committees dealing with technical matters in the area of computer networking.

ACKNOWLEDGEMENTS

RPCNET has been a team project and the authors would like to thank all the other partners in the project who have contributed to the ideas and the work presented in this paper.

REFERENCES

[1] L. Lenzini, G. Sommi: Architecture and Implementation of RPCNET. Proceedings of the 3rd ICCC Conference, Toronto, August 1976, p. 605-611.
[2] F. Caneschi, E. Ferro, L. Lenzini, M. Martelli, C. Menchi, M. Sommani, F. Tarini: Architecture of and the Service Facilities Provided by RPCNET - The Italian Computer Network for Education and Research Institutions. Proceedings of the ICCC/78 Conference, Kyoto, September 1978, p. 695-701.
[3] P. Franchi: Distribution of Functions and Control in RPCNET. Proceedings of the 3rd Annual Symposium on Computer Architecture, Clearwater Fl., January 1976, p. 130-135.
[4] A. Fusi: REEL Project: CNS/VM, The Virtual Machine Environment Computer Network Subsystem. Proceedings IEEE Computer Networks: Trends and Applications, Gaithersburg, November 1976, p. 128-134.
[5] G. Gori, M. Maier: Design and Implementation of Software for a Distributed Control Computer Network. Proceedings of the International Symposium on Technology for Selective Dissemination of Information, Republic of S.Marino, September 1976, p. 89-93.
[6] A. Springer, L. Lazzeri, L. Lenzini: The Implementation of RPCNET on a Minicomputer. ACM Computer Communication Review, January 1978, Volume 8, Number 1, p. 5-14.
[7] L. Lazzeri, L. Lenzini, A. Springer: Terminal Access to Host Computer through RPCNET. Proceedings of the ICS77/ACM Conference, Liege, April 1977, p. 335-344.
[8] L. Lazzeri, L. Lenzini: EPS7: 2703 Emulation Program for the System/7. Proceedings of the SEAS (SHARE European Association) Anniversary Meeting, West Berlin, September 1976, p. 202-210.
[9] P. Bertaina, M. Magini, C. Paoli, F. Tarini: Spool to Spool Protocol: Initial Design. RPCNET Internal Document IS007-01, October 1974.

The paper was received on January 29, 1979

SARA: a network between non-homogeneous batch computers

C. Di Filippo
SPERRY UNIVAC ITALIA
P. Mapelli, A. Mattasoglio, M. Zagolin
C.I.L.E.A. SEGRATE (Milano)
G. Meloni
POLITECNICO DI MILANO - C.I.L.E.A. SEGRATE

Abstract. in this paper is described a network for the trasmission of batch jobs from the terminals of non-homogeneus computers using a Univac computer as node of the network.

1. INTRODUCTION

The first tests to link up two computers were carried out in 1971 between the 1100's of the Computer Centre of Milan's Polytechnic and Rome University.
When the Polytechnic's computer was transferred to the S. Giuliano Milanese area, and the CILEA was set up, uninterrupted tests were resumed in 1975, due to the need to transfer the workload weighing on the Milan State University's 1106 onto the quicker 1108 without having to install a large number of lines.
In January 1976, the link-up between the 1108 and the 1106 became operative, enabling the batchload to be transferred to the more powerful computer.
The success of this exeperiment led to the decision to effect a link up along the same lines with CDC CYBER of the CINECA of Bologna, a machine which had been used by a large number of researchers in the University area.
This link-up began as an experiment in the Summer of 1976 and became official at the beginning of 1971. In the Summer of 1977, tests began on the link-up between the 370/68 of the CNUCE of Pisa and connection was fully effected in 1978.
At the present time, tests are under way on link-up with a Honeywell 6600 of the Computer Centre of the University of Pavia.
In this way, an organic system is created for the link-up of non-homogenous computer intended to sort batch loads according to CILEA equipment user requirements.

2. STRUCTURE OF THE SARA SYSTEM

The problem which the SARA system solved was that of providing all the users of a certain computer network with access to the calculation capacities of widely differing machines without making any hardware/software variations to the interconnected systems. The need to avoid at all costs changes to the operating systems of the interconnected centres led to research for a system which, although simple, was easily adaptable to any computer, and made it possible to develop a computer network at a time, 1975, when the market offered non hardware instruments dedicated esclusively to the interconnection of non-homogenous computers.
This solution, based exclusively on software and developed by the CILEA of Milan, although designed according to the specific requirements of technical /scientific users, besides allowing a high degree of generality, offers the possibility of easily managing and controlling a software product resident on a single machine which also acts as a concentrator.
SARA is a system which manages simulated terminals This means that in the node computers there are software boxes that are for the connected computers standard terminals.
These terminals receive the input queue, (fig. 1) transmit it for processing to the pre-selected computer and obtain the printouts which they insert in the output queue (fig. 2). The users who wishes to send jobs via SARA simply activates execution of a batch program whose purpose is to inform SARA of the existence of a job which has to be sent.
The general SARA system is divided into three distinct parts:
- the real-time part RT
- the batch-part RETEA
- the batch part SMISTA

First of all, the RT programs simulate as many hardware terminals recognized as their own by the

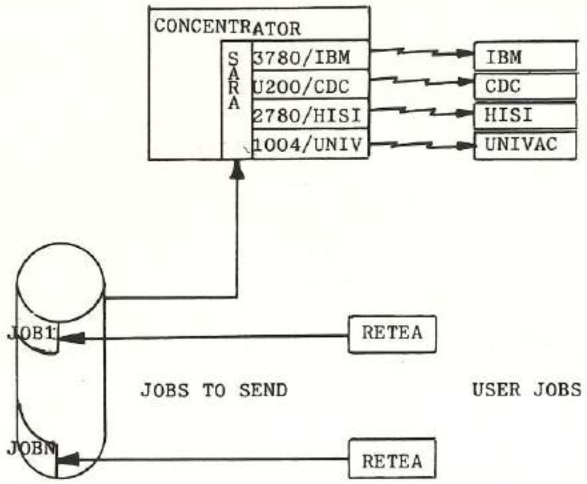


FIG. 1 JOBS TRANSMISSION

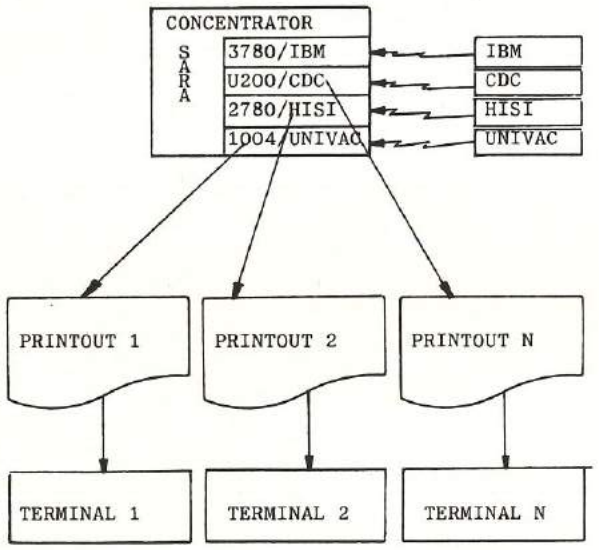


FIG. 2 PRINTOUTS RECEPTION

computers connected to the concentrator. As well as incorporating real trasmission/reception capabilities, these programs perform intelligent-type programs which enable them to recognize the final destination of the printouts arriving. In fact these can be destined to printers normally connected to the concentrator or to other centre connected via SARA.

Side by side with the functions seen above, RT takes care of gathering a series of management data on traffic, continually updating a special historical file, which is a necessary requisite for the network's efficient information system. Essentially, RETEA programs take care of placing in a situable queue the jobs to be trasmitted to the hosts to which they are destined; the information regarding the jobs placed in the queue is memorized in an area called TABCO, to which RETEA and RT have access.

Fig. 3 sums up the logical function of the system: firstly, we note that the SARA system, in the part relative to the real time program, can be activated at any point in time, since it is the RETEA programs' function to accumulate the jobs to be sent. On the other hand, it is easy to check the situation of the jobs to be sent, since it is sufficient to read the TABCO file: a similar observation is valid if we wish to save the contents of the queue of the jobs to be sent.

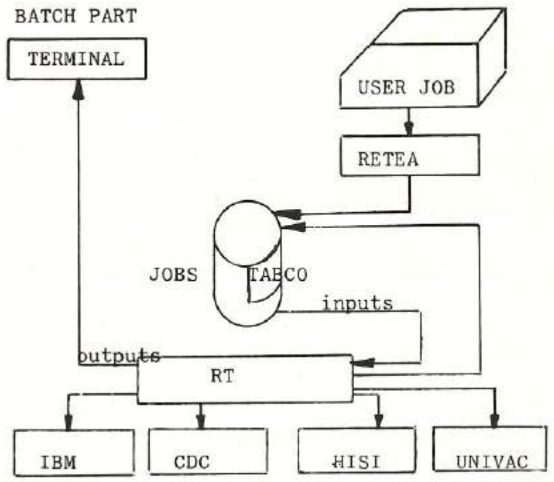


FIG. 3 FUNCTION OF SARA SYSTEM

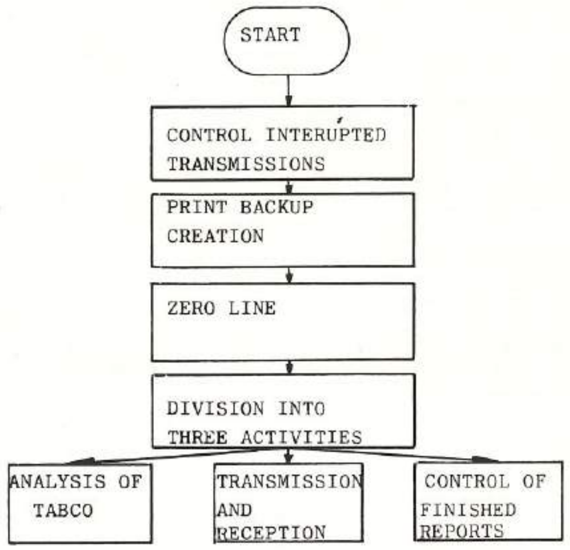


FIG. 4 ZEROING OF RT AND CREATION OF PROGRAM ACTIVITIES

Fig. 4 is a diagram of the RT program flow. After zeroing lines and the initial RT control functions, the 3 main activities are begun.

Let us now examine in detail the components of SARA by analysing the TABCO file.

TABCO

In the present version, this file consist of a variable number of entries, each one of five words associated to each job entering the network.

The data memorized are:

- date and time of placing the job in the queue, i.e. execution of RETEA
- name of the job entering the network
- name of the program which executed RETEA
- address of the entry which contains the management data relative to the transmitted jobs. This entry is created the moment the job enters the network and maintained until it finally leaves the system.
- name of the file containing the deck to be sent.

The mass memory occupation of TABCO depends on the number of jobs waiting to be sent: when this has been completed, the relative entry is freed.

RETEA

The RETEA program reads the group of cards which make up the job to be sent, and performs the following functions:

- updating of TABCO
- creation inside the job of two control comment cards containing the management data on the same
- translation of the cards into trasmittable format, which involves firstly converting the characters into the coding accepted by the host.

The two comment cards introduced by RETEA are extremely important.

These contain the following data:

- name of the print processor
- name of the on-site printer connected to the latter
- debit code for printout
- date and time of entry into network
- address of the entry which temporarily contains the updated management data relative to the job.

RT

This is a real-time program, and therefore always resident in memory and with operative priority over the other jobs present in the concentrator. Structurally, it is divided into three activities, where this term is used to indicate a program component independent of the others and operating in its own right.

The functions relative to the three parts mentioned above are:

- analysis of TABCO
- trasmission of the cards and reception of the printout lines
- analysis of the completed printouts

These activities perform their functions independently of each other. Variables for common use are accessible in controlled mode and at a certain time from one to only one of these.

Analysis of TABCO is extremely simple. The file is read and the job loaded with the latest date is searched for: if the search is positive, a variable used by the trasmission activity is set. Otherwise, the analysis is interrupted for approximately 30 seconds before restarting the cycle. The search activity for the first job to send is illustraded in the flowchart shown in fig. 5.

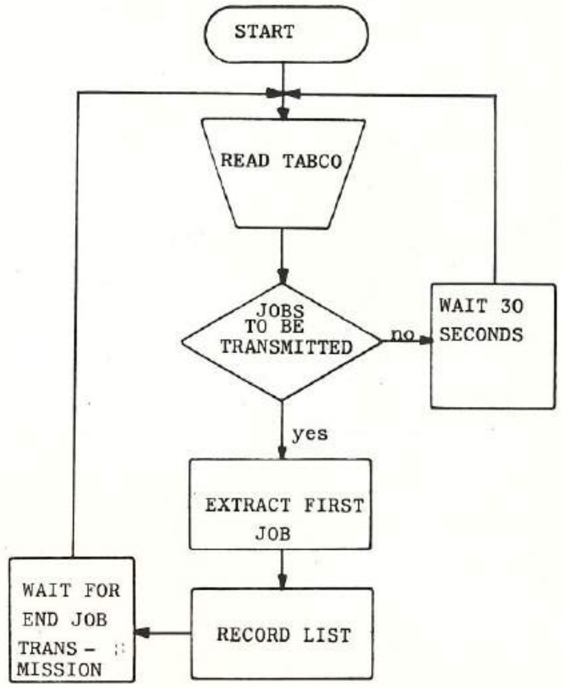


FIG. 5 ANALYSIS OF TABCO

The activity regarding transmission of the cards reception of printouts is carried out by the simulator of the terminal connected to the host. As far as sending thecards is concerned, it should be noted that RT reads from the disk the next blocks to be sent, prepared beforehand by RETEA and complete in all their parts, such as synchronism, characters, control etc.

Reception of printouts is rather completex, since the various reports have to be recognized and sorted at the terminals requested by the user.

RT recognizes two types of reports:

- real printouts, destined to the concentrator terminals or to other hosts
- simulated printouts containing card images, which we shall mention in the section on the SMISTA program.

If the process involves outputs destined to terminals of the computer containing SARA, the concentrator, RT orders printout by using the normal functions of the operating system; if, on the contrary, printouts destined to other hosts are involved, it activates a batch program called SMISTA which inserts printout into the queue of SARA.

Fig. 6 illustrates the print reception activity.

The third RT activity regards analysis of completed printouts which indicate whether the output relative to each job has been physically concluded at the requested terminal.

Fig. 7. Shows control of the concluded printout.

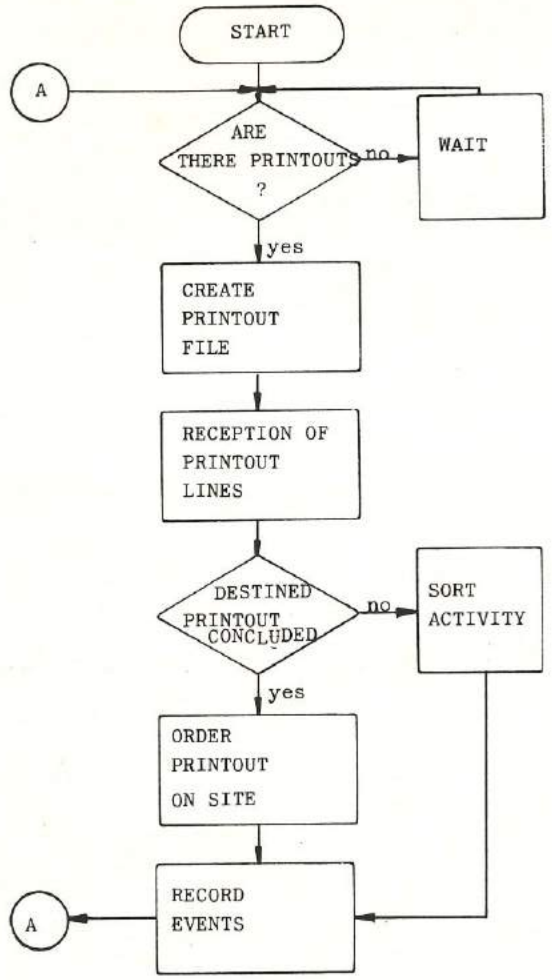


FIG. 6 RECEPTION OF PRINTOUT

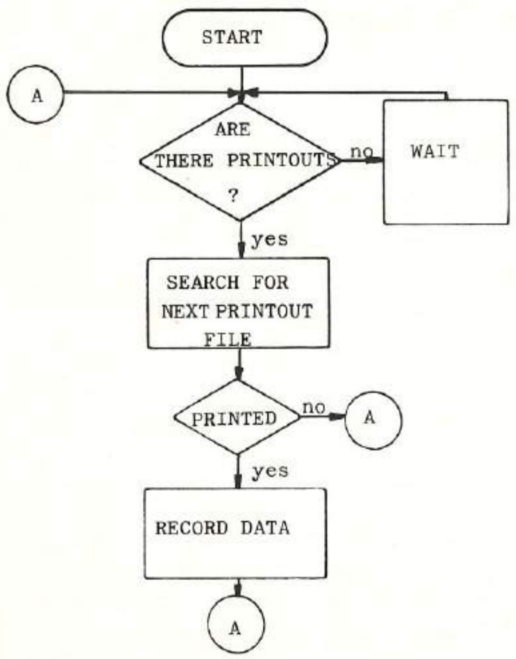


FIG. 7 CONCLUDED PRINTOUTS CONTROL

As well as the functions described above, the RT, the moment it is started:
- makes an analysis as to whether there exists a job, the sending of which has been interrupted for any reason (line interruption, system stop, etc.) If this is the case, it sends a message to the console operator and automatically activates a warning job the user, which will be printed out at the site to which the printout regarding the lost job was directed.
- catalogues a mass memory area, on which in all cases, all the print lines received from the host (print back-up) will be saved.

SMISTA

The following may all travel along the same line of communication which interconnects the computer containing SARA and a host:
- blocks of cards from the concentrator to the host (job to process).
- real reports destined to concentrator terminals.
- real reports not destined to concentrator terminals
- simulated reports containing card images.

In fact, in order to use the same line of communication in two directions, the input cards of job sent from a host for execution on the concentrator or on another host are transmitted like print lines in arrival, whilst printouts directed from the concentrator to a host are split into an 80-column format.

The above-mentioned sorting actions are performed by the SMISTA program, activated by the RT which carries out the following jobs:
- if the report has to be sent to a host, it transforms the print lines into cards. It then creates suitable control cards relative to a 'DECOD' program resident on the host, which will then carry out the inverse operation and will supervise sorting at the site requested by the user. The relative RETEA is then executed.
- if a job is involved, it executes RETEA immediately.

On each host connected therefore, there are two batch programs:
- DECOD
- INVIA

The first reads the input cards and changes them to print lines.

The second reads the job to be transmitted, prints it and orders the report to be sent to the simulated terminal in the concentrator.

Fig. 8. Shows the various steps (numbered from 1 to 11) which a job must take if it enters a host connected to the concentrator.

The step is relative to a job sent to the concentrator in report form, esecution of the INVIA program.

RT activates SMISTA (2) which changes the lines into cards (3), loads everthing into the queue of SARA (4,5).

At this point, the job follows the normal path; i.e. it is transmitted for execution to the host requested (7).

The report relative to the job executed at the host is received by SARA (8) which once more activates SMISTA and re-runs through steps 2,3,4, 5.

At this point, RT must send a job to the host which includes the execution of the DECOD program, which reads the cards, prints them, and sends everything to the terminal requested by the user in step 1.

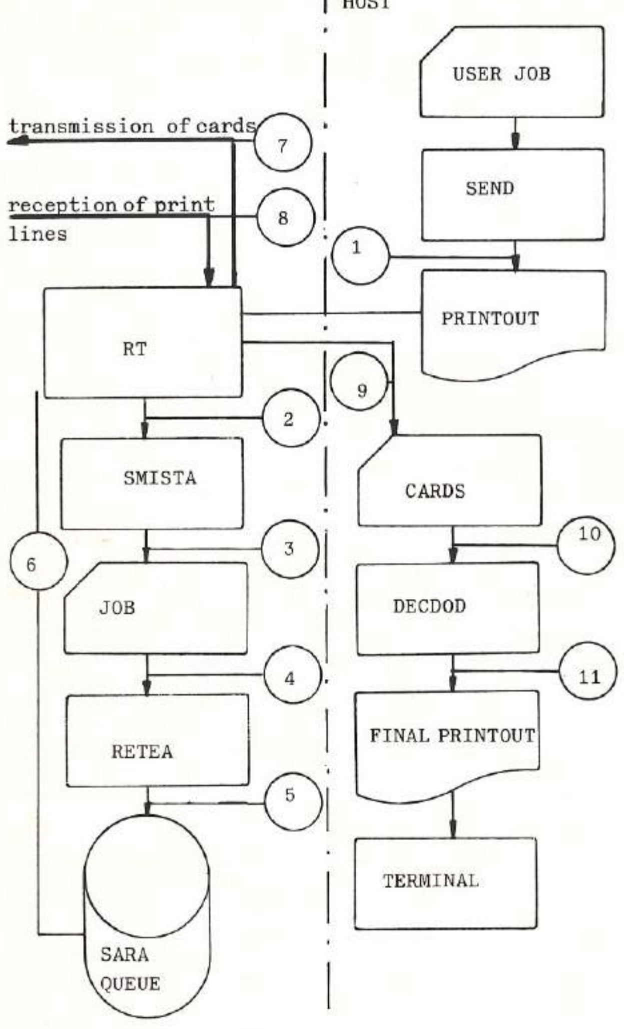


FIG. 8 RECIPROCITY OF SARA
SORTING A JOB AND RELATIVE PRINTOUT

3. TYPES OF CONNECTION EFFECTED

On the basis of the algorithm described in general terms in the previous chapter, three types of connections with as many University centers in northern Italy have been developed by the CILEA of Milan, (The Inter-university Consortium of Automatic Calculation). These are:
- CINECA of Bologna, in particular with the CYB76 of the CDC;
- CNUCE of Pisa, in particular with the IBM 370/68
- Centro di Calcolo of the University of Pavia, and in particular with the Honeywell 6600.

As well as the three link-ups mentioned above, there esists a connection with the Univac 1108 of the Cilea.

The computer which acts as a concentrator, as well as providing users with normal services, is a Univac 1106, which also belongs to the CILEA. For each center connected, the SARA system offers as many versions of the RETEA and RT programs, each one suitable for a particular type of dialogue to be handled, whith the other parts of the program remaining practically unchanged. As far as the lines of communication are concerned, the same number of lines have been exploited as those previously used by hardware terminals connacted to hosts; in other words, the SARA system has pratically replaced, as far as dialogue is concerned a pre-existing machine accepted by the host itself.

Now let us examine in detail the various connections which can be effected.

Univac 1106 - 1108

The dialogue module reflects the standard one used by the 1104 teminal. During the development phase, no real problems arose, since tests were carried out on two physically separated machines in the same room.

Univac 1106 - CYB76

In this case a USER 200 emulator was written. Here, the development phase was rather difficult, since, apart from the complexity of the dialogue itself, other problems arose concerning the identification of the beginning and end points of each printout received by SARA.

The reciprocity module was developed in collaboration with the software staff of the Cineca EDP Center.

Univac 1106 - IBM 370/68

After having emulated a standard 3780, there were no problems, thanks to the collaboration provided by the software group of the Cnuce.

Univac 1106 - Honeywell 6600

This phase is in its final stages. The dialogue module is the one used on the 2780 terminal.

4. MANAGEMENT AND USE OF THE NETWORK

The SARA system is designed as a series of modules which, at least in general terms, are not affected by the variations which the operative systems of connected machines undergo in their logical evolution. It should be sufficient to consider the dialogue protocol which reflects the principles of as many hardware terminals which the network's hosts normally accept as their own. SARA is also subject to changes, but since its relative software is entirely resident on a single center, it is easily controlled: the INVIA and DECOD programs belonging to the libraries of the connected computers, due to their extreme simplicity, do not present any problems whatsoever. As far as the management of SARA is concerned, we should single out two aspects:
- RT programs
- RETEA programs.

which communicate with each other using TABCO. In order to provide users always with the best service, it is important for the elements which effect link-up to be continuously active: in SARA's philosophy, however, only the RETEA programs need to be permanently resident in the library and at the disposal of those users who wish to exploit the services offered.

On the contrary, the RT programs can be activated by the Center manager at any point in time, also at set times during the day, since, in addition, it is possible to know the consistency

of jobs waiting to be sent to the host; these require no preliminary operator intervention, since all their functions have been automized, such as, for example, assigning of the line of communication, cataloguing of the mass memory areas to be used,
etc.
The modality of use of the RETEA programs which interface the user with SARA highlight the "external" aspect of this system and show how simple it is to use.
The RETEA programs, each one of which is relative to a certain host, enable a job-stream to be attached to the queue of SARA:
these provide SARA with data on:
- print processor
- print site connected to the latter

deducing them from a parameter card provided by the user.
Use of the system is simple, and has a pre-established plan formed by five controlcards of the EXEC 8 system of the concentrator.
We illustrate an example of usage in fig. 9 which shows a job-stream a batch terminal or conversational user must execute before sending his own job to the host.
Instructions 'B' contain the job to be executed by the host connected via SARA whilst the part A is relative to the execution of RETEA. In particular, we shall have:
- card 1 begins execution of a job on the Univac 1106
- cards 2 and 3 fixed.
- parameter card 4 indicates the name of the printer on which the user wishes to have his results. Since only parameter PR1 is specified, the system assumes as a printer the concentrator itself.
- card 5 is fixed
- card 6 indicates the type of RETEA program to be executed. In other words, it specifies in which queue the job to be sent is to be memorized.

If, on the contrary, the user wishes to obtain printout of his job on a terminal connected to a computer different from the concentrator, the parameter card takes on the format:
SITO, ELSTA, ACCT.
where:
SITO is the name of the terminal
ELSTA is the name of the host processor of the network in which printout is required, e.g. CDC.
ACCST is the debit code with which to execute the DECOD program by the host.

```
1   "RUN.......
2   "ASG,T 4,F2
3   "DATA,ID A.
4    PR1
5   "END
6   "XQT  RETE$*U8.IBM
7   "FIN
```

```
      //JOB.............
      //
A   B :
      :
```

Fig. 9. Example of a job to transmit to the host.

5. THE INFORMATION SYSTEM

As well as managing the traffic of the link-up with the hosts, the SARA system performs the important function of gathering and filing the management data relative to jobs which are under its direct control.
A certain job has two possibilities of entering the network:
- execution, in the concentrator, of RETEA, which directly memorizes the job in the queue of SARA
- execution of INVIA, by the host, which prints card decks at the terminal simulated by SARA.

From a management and control point of view, the significant events which reflect as many important points in the journey of a certain job are:
- execution of RETEA; the job is placed in a queue and waits to be transmitted
- beginning of transmission
- end of transmission

At this point, the job has been accepted by the host and is processed.
At the end of processing, the relative printout is sent to the
concentrator and recognized by one of the RT activities which sorts it.
Management data travel together with the job in the form of a comment card. In this phase, the following data are taken into consideration and recorded:
- beginning of reception of printout
- end or reception of printout

The third RT activity also records the moment at which it phisycally ends and the destination to which it was addressed.
On the contrary, in cases where a job is sent by a host to be executed by another host, the significant moments are:
- beginning of reception of the card deck in the form of a report, (first contact with the network).
- end of reception
- execution of SMISTA with queueing in SARA's entry queue
- beginning of transmission of job
- end of transmission of job
- beginning of reception of printout
- end of reception of printout
- execution of SMISTA which places the report, in card form in SARA's queue.
- beginning of transmission
- end of transmission

The job finally leaves the network. For each of the moments now described, SARA creates an entry which is placed sequentially, i.e. in order of date, in the queue in a special historical file.
The main data contained in this queue are:
- type of event (beginning of transmission, end of reception etc.)
- date and time of the entry
- name of the job in the network

- name of the job which executes RETEA
- debit code of the transmitted job which executes RETEA
- number of the cards transmitted
- number of printout pages produced
- identification of the entry (only name of the file).

The management data gathered by SARA provide the system manager with an effective control istrument which enable him to know the exact situation of the traffic passing through the concentrator and to handle it accordingly with the highest degrees of performance and reliability.

6. LOAD MEASUREMENTS

Load measurements were carried out the batch job traffic transmitted by SARA between the 1106 and the 1108 of CILEA, and between the 1106 and the CYBER 76 of the CINECA.
The period considerad is February 1979.
During this period, the duration of the 1106's operation was 327 hours, during which the R/T programs for link-up with the 1108 and the CYBER 76 remained active respectively for 248 and 158 hours, transmitting a total of 451 jobs.
Bearing in mind that the operation lasted for 19 days, we have an average of 23,7 transmission per day, with peaks of 109.
The ratio between the CPU time consumed by R/T program and the link-up time with the 1108 is:

$$R2 = \frac{Tcpu\ rt1108}{Tcoll\ rt1108} = 0.0028$$

whilst for link-up with the CYBER 76, we are:

$$R3 = \frac{Tcpu\ rtcdc}{Tcoll\ rtcdc} = 0.022$$

The ratio between CPU time consumption of the RT1108 and the time for all programs processed by the 1106 is:

$$R4 = \frac{Tcpu\ rt1108}{\sum cpu\ 1106} = 0.008$$

whilst for RTCDC we have:

$$R5 = \frac{Tcpu\ rtcdc}{\sum Tcpu\ 1106} = 0.04$$

The ratio of link-up time of the RT1108 and the operation time of th 1106 is:

$$R6 = \frac{Trt\ 1108}{T1106} = 0.75$$

whilst for RTCDC it is:

$$R7 = \frac{Trtcdc}{T1106} = 0.48$$

Fig. 10 shows the progress of the traffic in the period taken into consideration.
Memory occupation of RT 1108 and RT CDC is 12 K words each.

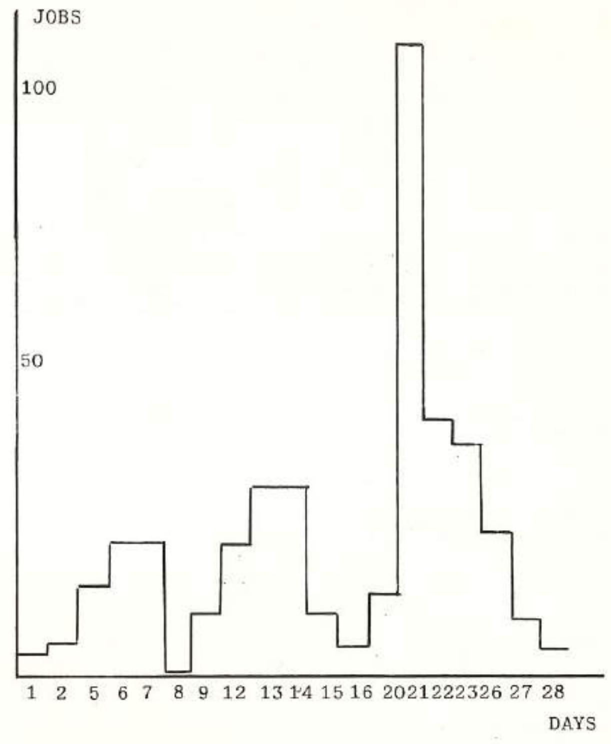


FIG. 10

The paper was received on April 18, 1979

REFERENCES

[1] Collegamenti tra i sistemi di elaborazione, INFORMATICA E DOCUMENTAZIONE, July 1976.

[2] P.L. Monti, Il Sistema SARA, Sistemi ed Automazione no. 183, 1979.

[3] Corriere della Sera Anno 104 no. 148 inserto economico (VIII) 5 luglio 1979.

From R & D Laboratories

Some statistical measurements on the European Informatics Network (EIN)

S. Alfonzetti, S. Casale, A. Faro, S. Palazzo, V. Saletti, G. Scollo
ISTITUTO ELETTROTECNICO, UNIVERSITA' DI CATANIA — CREI (POLITECNICO DI MILANO)

Abstract. This paper deals briefly with some measurements on EIN obtained by processing the statistical data collected at the Network Control Centre. The data concern July 1978 and the results obtained allow us to characterize some aspects of the communication subnetwork performance, of the traffic on it and of the user activity.

1. INTRODUCTION

The EIN Executive Body [1] is responsible for collecting data on user and service packet traffic in the network through the Network Control Centre (NCC). The measurements described here are taken daily (five days a week) as follows :

- in the morning (at Core Time Start (CTS)) the NCC sends to each Network Switching Centre (NSC) a packet to reset the statistical counters inside it; the NSC polling sequence is always the same, that is :
  LONDON - PARIS - ZURICH - ISPRA - MILAN
- at the end of the day (Core Time End (CTE)) the NCC asks each NSC for a packet containing the final counter values; here also the NSC polling sequence is as above [2].

In each NSC the counters refer separately to the activity of each afferent line. These lines are of two types : lines NSC-NSC, which connect the NSC with another NSC, and lines NSC-SC, which connect the NSC with a Subscriber Computer (SC).

For a NSC-NSC line the counters are as follows:

CPPO : number of user packets sent [3] [4];
CPSO : number of local service frames and NIPs sent [4];
CBTO : total bytes sent (including retransmissions);
CPPI : number of user packets received [3] [4];
CPSI : number of local service frames and NIPs received [4];
CBTI : total bytes received (including retransmissions and incorrect frames received at the hardware level);
CHCFI : number of incorrect frames at the hardware level given by CRC error;
CSCFI : number of user packets, routing updates and NIPs received with software checksum error [5];
CPXDI : number of frames discarded at line level for one of the following reasons: 1) duplicated, 2) congestion of the node, 3) parallel activity by DEPAN, 4) abnormal hardware end without CRC error.

For a NSC-NSC line, besides the above ones, in the NSC there are other counters which provide the histogram of text length of the received user packets.

Some elaborations of such data referring to July 1978 are shown in the next section.

2. ELABORATIONS CARRIED OUT AND THEIR RESULTS

The data were elaborated in order to statistically characterize the performance of and the traffic on the communication subnetwork and the activity of each SC.

2.1 Communication subnetwork

The behaviour of the communication subnetwork can be analyzed from two different points of view; the first one refers to the subnetwork components (lines and nodes); the second one refers to the subnetwork as a single system interacting with its users, i. e. the SCs.

In order to evaluate the line performance we introduce a parameter defined as the percent ratio of the number of frames not affected by CRC error to the total number of frames received, i. e. :

$$q_{line} = \frac{CPPI + CPSI + CPXDI}{CPPI + CPSI + CPXDI + CHCFI} \cdot 100$$

Such a relationship is used to characterize the NSC-NSC lines in each of the two directions and the NSC-SC lines only in the SC→NSC direction [6]. For each line the average and r.m.s. values of this ratio are shown in table 1.Still referring to the NSC-NSC lines, tables 2 and 3 respectively show the average number of user packets per day and the average number of service packets for each subnetwork line (oriented). The substantial uniformity of the service traffic can be pointed out.

The subnetwork nodes will now be considered. In order to evaluate the NSC's behaviour, the following parameter is introduced for each NSC :

$$d_{node} = \frac{\Sigma\ CPXDI}{\Sigma\ (\ CPPI + CPSI + CPXDI\ )} \cdot 100$$

where the summations are extended to all the NSC-NSC and NSC-SC lines connected to the node. This parameter gives the percent ratio of frames discarded by the node to the total number of frames correctly received. Table 4 gives the average and the r.m.s. values of this ratio for each NSC.

Fig. 1 shows the average number of user packets per day in input and output on each NSC. The difference between the input and output values is due to the fact that the NSC can behave as a "well" or

Tab. 1 - Average and r.m.s. values of the ratio q_{line}

RECEIVER		SENDER Z	L	I	P	M	SCs			
Z	$\bar{x}$	-	76.112	-	95.664	99.294	ETH 99.904	-	-	-
	σ		14.637		12.850	.974	.269			
L	$\bar{x}$	97.004	-	-	98.312	-	EMU 98.041	AERE 99.992	NPL 99.998	NCC 100.000
	σ	1.003			2.284		2.749	.009	.002	.001
I	$\bar{x}$	-	-	-	99.377	99.952	JRC --	-	-	-
	σ				.680	.081	--			
P	$\bar{x}$	98.971	94.378	99.652	-	-	IRIA 100.000	-	-	-
	σ	1.057	6.256	.564			.000			
M	$\bar{x}$	99.788	-	97.349	-	-	CILEA 99.778	CSATA 99.714	-	-
	σ	.140		.907			.338	.173		

Tab. 2 - User packets per day

RECEIVER	SENDER Z	L	I	P	M
Z	-	2152	-	329	805
L	1060	-	-	2216	-
I	-	-	-	1967	476
P	330	4389	861	-	-
M	1469	-	814	-	-

Tab. 3 - Service packets per day

RECEIVER	SENDER Z	L	I	P	M
Z	-	55252	-	58279	56685
L	58244	-	-	59619	-
I	-	-	-	59474	57464
P	58098	57306	55629	-	-
M	58408	-	59240	-	-

Tab. 4 - Average and r.m.s. values of the number of frames discarded by each NSC

	Z	L	I	P	M
$\bar{x}$	32.7	18.8	18.1	25.7	22.2
σ	4.4	2.3	3.8	2.8	5.8

[1] The Executive Body is a committee charged by the technical management of the EIN project.
[2] This way the time intervals between the CTS and the CTE of each NSC have pratically the same size even though a little bit shifted.
[3] The NON DELIVERY DIAGNOSTIC (NDD), DELIVERY CONFIRMATION and TRACE packets are included.
[4] Retransmissions not included.
[5] This counter is meaningful only with the version "protocol" of the NSC software. Since this version was not running in the period the data refer to, CSCFI has not been taken into consideration in our elaborations.
[6] This is due to the lack of data on the traffic entering the SCs.

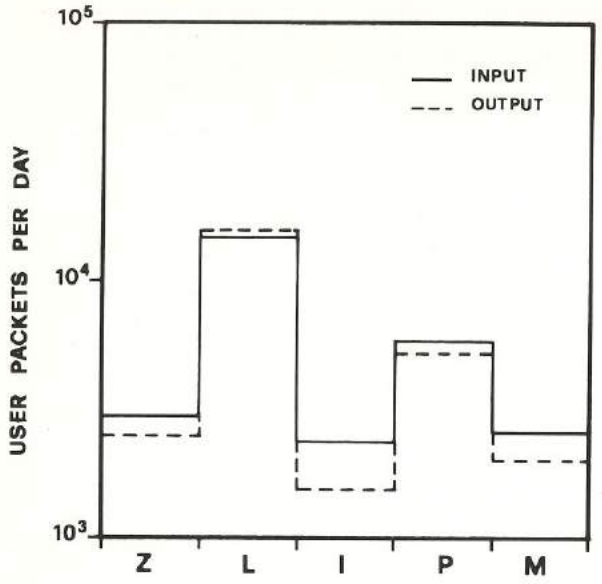


Fig. 1 - Average number of user packets per day in input and output for each NSC.

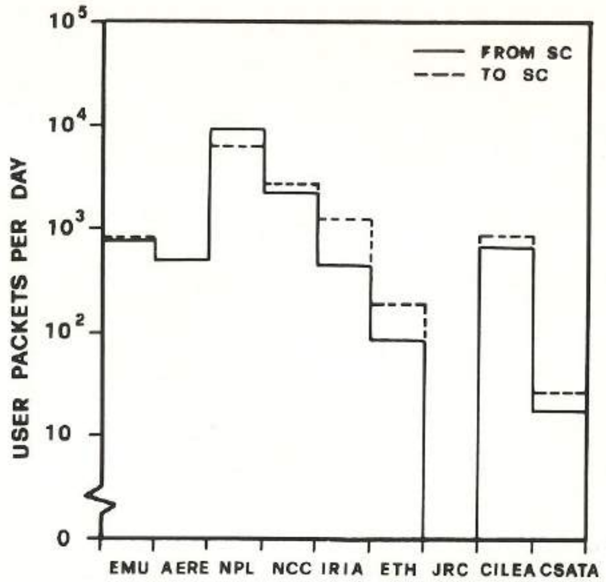


Fig. 3 - Average number of user packets per day in input and output for each SC.

a "source" of user packets. An example of the former behaviour is the discarding of the user packets addressed to the nodal process DROP; an example of the latter one is the generation of a TRACE packet at the switching of a user packet requesting this subnetwork facility.

Finally the overall behaviour of the subnetwork is characterized. In fig. 2 the daily values of the total user traffic in input to, in output from and in the subnetwork (as the summation extended to all the NSC-NSC lines) are plotted. The ratio of the total traffic in the subnetwork to the total input traffic in the month is 300341/248904 ≃ 1.21; this value, increased by one gives the average number of NSCs that a user packet passes through.

2.2 Users

In order to statistically describe the SC activity in EIN the histogram of the average number of user packets in input and output per day is shown in fig. 3. Furthermore fig. 4 shows the histogram of the total number of user packets sent during the month by each SC and all together, separated into three classes according to the byte length of their texts:

small	0 ÷ 63
medium	64 ÷ 159
large	160 ÷ 255

Let us note that for some SCs (AERE, IRIA, ETH, CILEA) the histogram is characterized by a predominant amount of small packets and this is indicative of a mainly demand user activity; for some others (f. e. NPL) the traffic, produced by a text fragmentation process and therefore characterized by a large number of maximum length packets, is added to that of the demand activity; for other ones the distribution is influenced by the services they offer (f.e. NCC, which gives the EIN map service, produces a great number both of small packets asking the NSCs to update the map and of large packets to send the map to the users asking for it).

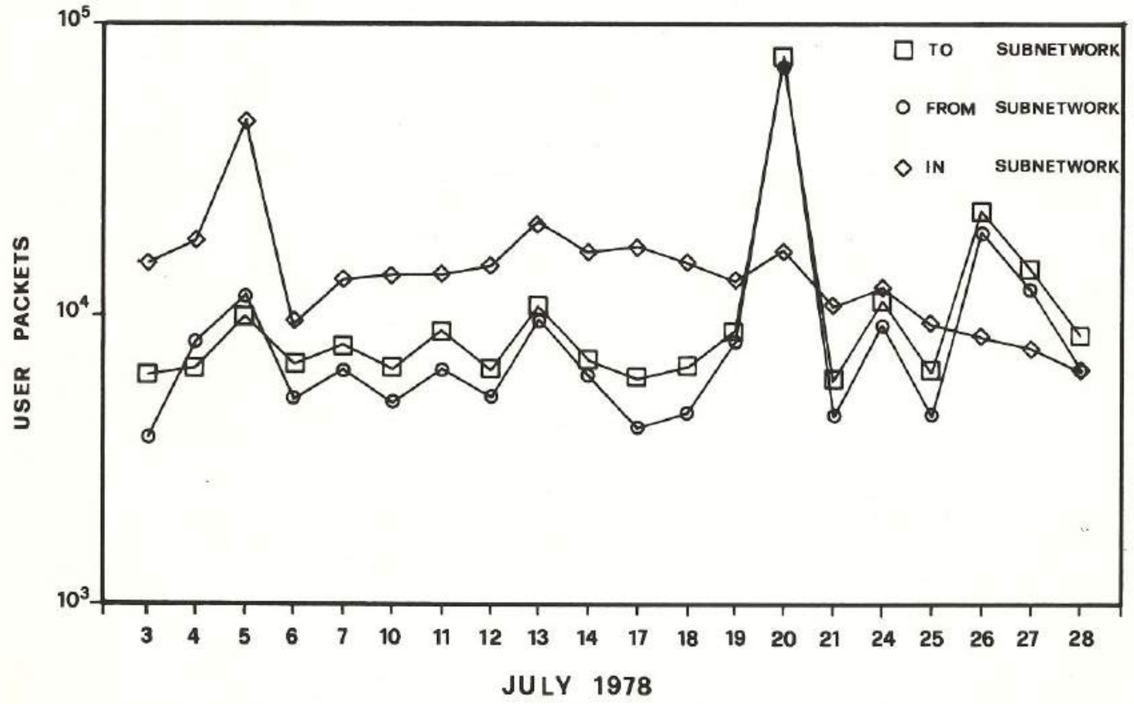


Fig. 2 - Daily user packet traffic in July 1978.

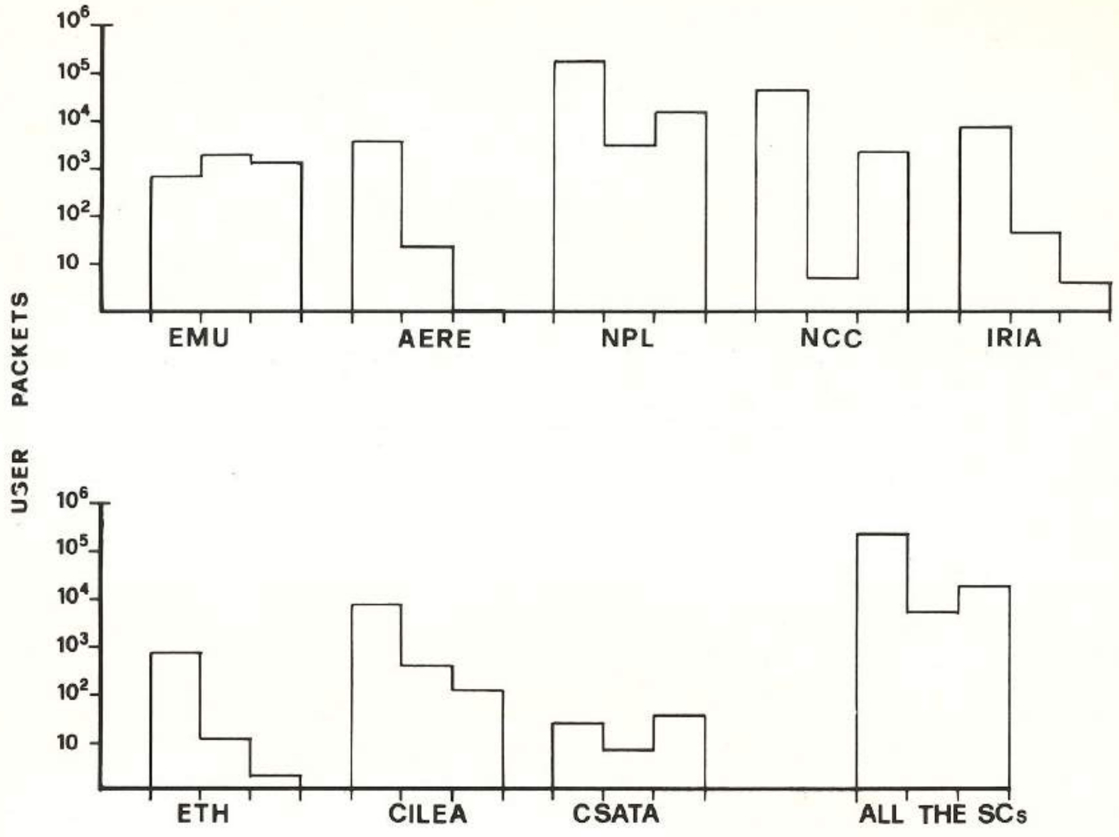


Fig. 4 - Histograms of text length in the user packets sent by each SC.

3. CONCLUSIONS

The purpose of this paper was to show some first measurements on EIN obtained by processing the statistical data collected at NCC. These measurements allow one to characterize some aspects of the subnetwork performance, of the traffic on it and of the activity of its users. Of course the results obtained are influenced by the period (July 1978) in which the data were collected. For example the decreasing traffic in the subnetwork (fig. 2) is an obvious consequence of the lack of coordination between the users, due to the approach of the summer holidays.

A deeper analysis of the network behaviour will be possible only after elaborations are made over more and longer periods. The results of these elaborations will be shown in further papers.

Measurements concerning other aspects of the EIN behaviour are shown in [1] [2] [3] [4].

ACKNOWLEDGEMENTS

The authors wish to thank Mr. G. Kacin for his helpful suggestions.

This work was partially supported by CNR (Consiglio Nazionale delle Ricerche) Contract N. 78/00767.

The paper was received on March 29, 1979

REFERENCES

[1] S. Sedillot: EIN subnetwork performance measurements from the EIN/CIGALE gateway. EIN/IRIA/77/5.

[2] J. Laws: EIN subnetwork performance measurements from the EIN/NPL Gateway. EIN/NPL/77/005.

[3] A. Faro, G. Scollo, F. Valora: Some measurements on the EIN Computer Network performed at CREI by means of the Subnetwork Control Module. EIN/CREI/79/2.

[4] G. Scollo: Realizzazione di misure su alcuni aspetti della rete di calcolatori EIN. Facoltà di Ingegneria, Università di Catania, degree thesis, 1977.

Some measurements on the EIN computer network performed at CREI by means of the subnetwork control module

A. Faro, G. Scollo, F. Valora
ISTITUTO ELETTROTECNICO, UNIVERSITA'DI CATANIA — CREI (POLITECNICO DI MILANO)

Abstract. In this paper we present and discuss some measurements on the EIN and CIGALE interconnected computer networks, performed over a five weeks period during summer 1978 at CREI (Centro Rete Europea di Informatica) in Milan by means of the Subnetwork Control Module. The measured behaviour regards the subnetwork facilities implemented in all the nodes of the computer network. The obtained results allow us to give an estimation of some network parameters and useful information also about the connection of the Italian SC's to the computer network.

1. INTRODUCTION

In this paper we show and discuss the results of some measurements on the EIN and CIGALE computer networks, performed over a five weeks period during summer 1978.

The measurements, performed at CREI by means of the Subnetwork Control Module (SCM) [1], [2], have been made at the packet level and regard the addressable and non-addressable facilities implemented in all the nodes of the subnetwork.

Suitable commands have been given to SCM in order to produce sequences of packets directed to the addressable facilities (Virtual Subscriber Computer Processes - VSCP's -) : each command causes SCM to transmit a sequence of packets to a VSCP in a node with DELIVERY CONFIRMATION (DC), NON DELIVERY DIAGNOSTIC (NDD) or TRACE request; the time interval between the expected departures of two subsequent packets of the sequence was selected constant.

The responses coming from the subnetwork to SCM were recorded onto a file which was processed in order to obtain a statistical description of the subnetwork behaviour as seen from the SC's implemented in the UNIVAC 1106 Host-computer in Milan.

The obtained results allow us to give an estimation of some important network parameters so as the response time, the loss and the duplication frequency of the responses coming from the subnetwork, the network time precision for each EIN node, the routes of the packets in the subnetwork and useful information also about the connection of the Italian SC's to the computer network.

2. MEASUREMENT SYSTEM

The system available for performing the measurements on the computer network is schematized in fig.1 . This system is constituted essentially by the Subnetwork Control Module (SCM) which is a tool studied by CREI for testing and measuring the computer network and realized by means of a piece of software running at the UNIVAC 1106 Host-computer in Milan.

SCM is an interlocutor at the packet level connected to the interlocutor I1 which performs the line protocol. Also I1 is realized by means of a piece of software running on the UNIVAC 1106.

A procedure adaptor realized by the SELENIA GP 160 minicomputer is put between the UNIVAC 1106 and the Network Switching Centre (NSC) because the UNIVAC 1106 has not a line procedure compatible with that of the EIN NSC. The GP 160 minicomputer receives frames with fixed length from the UNIVAC 1106 and maps them in frames at HDLC level directed to the NSC [3].

The pattern of the travelling packets between SCM and NSC is schematized in fig.2 . Let us note that a file exists between SCM (working in demand) and I1 (working in real time) in order to make them asynchronous.

The queueing structure between SCM and NSC increases the response time from the subnetwork. In particular the delay due to the connection SCM-NSC is constituted essentially by the following two components :

- the time for reading and writing on the files between SCM and I1 (about 1.0 - 1.5 s);

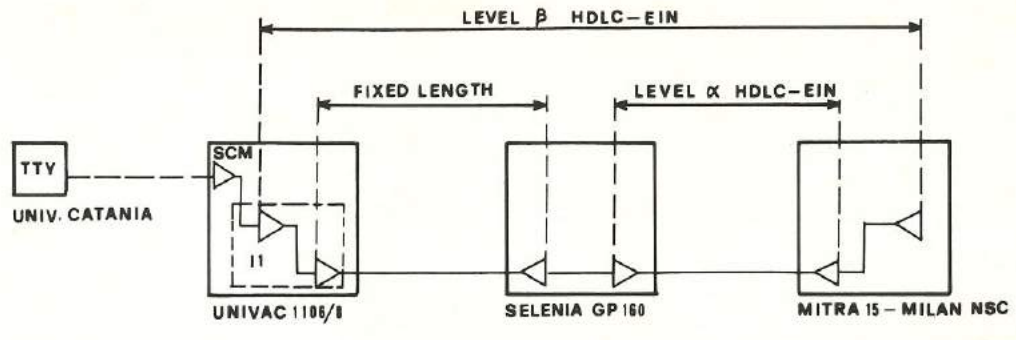


Fig. 1 - Connection of SCM to EIN

- the transmission time on the lines between the UNIVAC 1106 and the NSC, and the time for the mapping process performed by the procedure adaptor (about 1.5 - 2.0 s).

During the experiments which are of concern to this paper, the local terminal of SCM (called Master TTY) was a TTY at the University of Catania. From this TTY we controlled the way in which SCM produces traffic of packets directed to the subnetwork. The experimental data coming from the subnetwork were recorded at first onto a file on the UNIVAC 1106 in Milan, then they were transfered from the UNIVAC 1106 to the CDC 6600 computer in Casalecchio (Bologna), through the line connecting those computers, in order to be processed off line by suitable programs arranged from another terminal of the University of Catania connected to the CDC 6600.

3. EXPERIMENTS

The purpose of the experiments has been to measure some network parameters dealing with the subnetwork facilities implemented in the nodes of the EIN and CIGALE interconnected networks (fig. 3).

The traffic to the subnetwork facilities has been produced by commands emitted at the user interface of SCM which cause SCM to transmit suitable sequences of packets.

In these user commands it has been specified :

- the destination VSCP to which SCM has to transmit the sequence of packets;
- the non addressable subnetwork facilities requested for the packets of the sequence;
- the number of the packets of the sequence;
- the time interval between the departures of two subsequent packets of the sequence.

When the packets were delivered by SCM to I1 or received by SCM from I1, then they were recorded onto a file together with a time mark. As already said, this file has been processed off line in order to obtain the statistical description of the subnetwork shown in the next section.

The time interval between the departures of two subsequent packets of a sequence has been selected

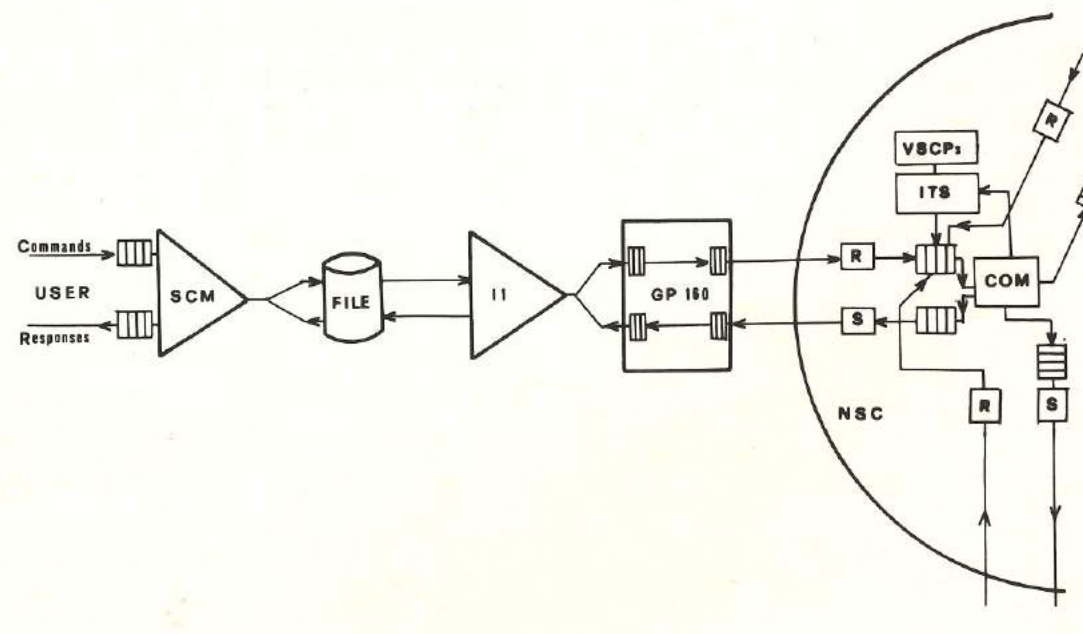


Fig. 2 - Queueing structure of the connection of SCM to EIN

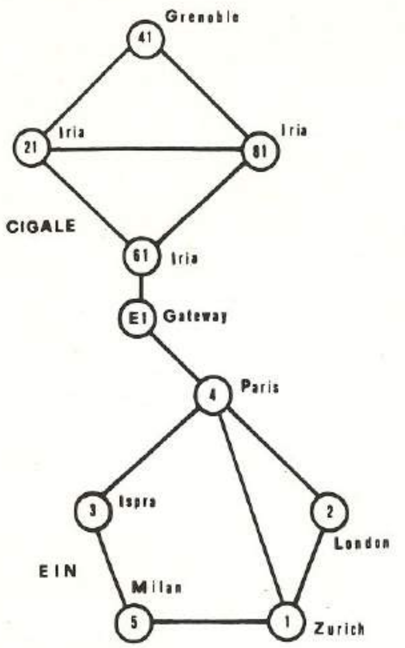


Fig. 3 - EIN and CIGALE communication subnetworks

in order to avoid congestion in the procedure adaptor which produces a slowing down in the transmission of packets between the UNIVAC 1106 and the Milan NSC for the reasons specified in section 2.

After some experiments the following time intervals between the departures of two subsequent packets have been selected :

- 15 s between the departures of two subsequent packets of a sequence directed to the VSCP DROP (1) and requesting TRACE in EIN;
- 20 s between the departures of two subsequent packets of a sequence directed to the VSCP DROP and requesting TRACE in EIN and CIGALE;
- 10 s between the departures of two subsequent packets of a sequence directed to the VSCP NT (2) in EIN or to the VSCP ECHO (2) in EIN and CIGALE.

The time interval between the emission of two subsequent commands has been selected in order to allow the reception of all the responses of a command before the emission of the next command.

3.1. *TRACE*

This experiment has been executed by means of commands which cause SCM to transmit sequences of packets directed to the VSCP DROP for all the EIN and CIGALE nodes with request of TRACE facility.

The nodes, which the packets passe through, send TRACE packets to SCM containing in the text the network time in that node and the destination of the packets.

3.2. *NETWORK TIME, ECHO (with request of DC and NDD facilities)*

This experiment has been executed by means of commands which cause SCM to transmit sequences of packets directed to the VSCP NT for all the EIN nodes and to the VSCP ECHO for all the EIN and CIGALE nodes, with request of DC-NDD facilities for EIN and only NDD for CIGALE (3). The destination nodes of the packets send ECHO or NT packets to SCM and also DC or NDD packets to notify to SCM the delivery confirmation or the cause for which the delivery could not be achieved.

4. RESULTS

At first let us show how the subnetwork reacts in terms of interarrival time intervals in the experiments dealing with the TRACE and NT facilities.

Fig. 4 shows the normalized distribution (4) of the interdeparture time and interarrival time intervals for the TRACE experiments. Let us note that the interdeparture distributions are impulsive like around the nominal time interval (r.m.s. of 2s)(5), whereas the interarrival distributions are much more spreaded (r.m.s. of 5s), due to the internal processes of the subnetwork and to the mapping process made by the procedure adaptor minicomputer. In these experiments the areas of the interdeparture distributions are roughly the same, but not those of the interarrival distributions because they depend on the routes of the packets in the subnetwork. The interarrival distribution for the Milan, Zurich and Paris nodes show two peaks : the first (around 15s) due to the TRACE experiments in EIN and the second (around 20s) due to the TRACE experiments in EIN and CIGALE (see section 3).

Fig. 5 shows a tipical normalized distribution of the interdeparture time and interarrival time intervals in the NT experiments with DC-NDD request. Also in this case the interarrival distribution is more spreaded; moreover it shows two peaks : the first (around 2s) due to the interarrival time interval of a NT packet after a DC packet, the second (around 10s) due to the interarrival of DC (or NT) packets.

4.1. *RESPONSE TIME*

The response time for both the experiments is shown in figg. 6,7 . Let us note that the delay normalized distribution for the TRACE responses shifts on greater values as the responses come from more distant nodes (fig. 6); in particular, in EIN the TRACE response average delay increases about 300

(1) VSCP DROP discards the packets on receipt.
(2) VSCP NT returns a packet containing the network time embedded in the text; VSCP ECHO returns the complete packet it received to the sender.
(3) NT and DC facilities are not implemented in CIGALE.

(4) These distributions are normalized to the maximum frequency of all of them.
(5) A little spreading around the nominal interdeparture interval is unavoidable, because SCM "lives" into the UNIVAC 1106 like a user job : therefore it has not a complete control of the CPU, but it is subjected to a priority assignment process.

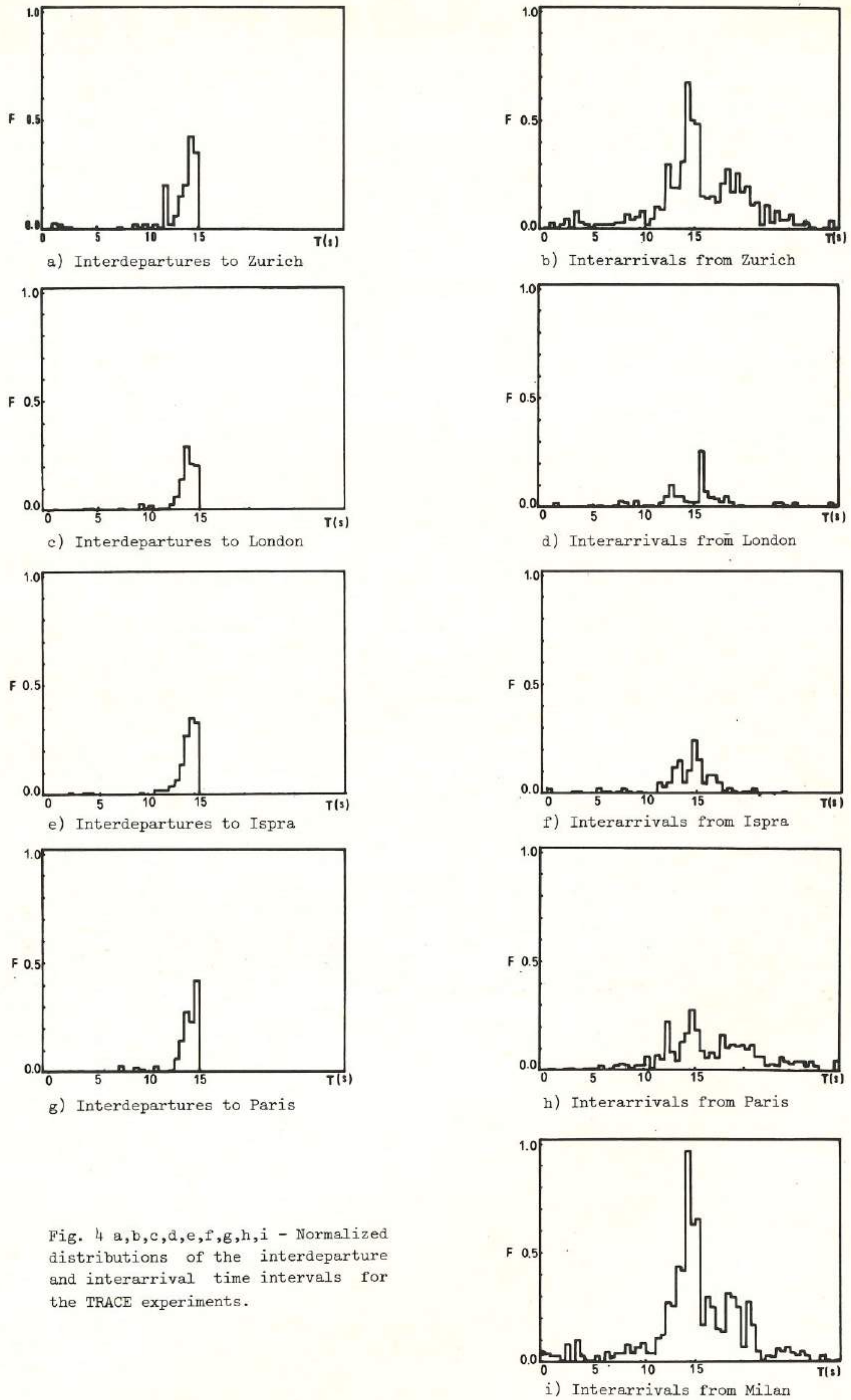


Fig. 4 a,b,c,d,e,f,g,h,i - Normalized distributions of the interdeparture and interarrival time intervals for the TRACE experiments.

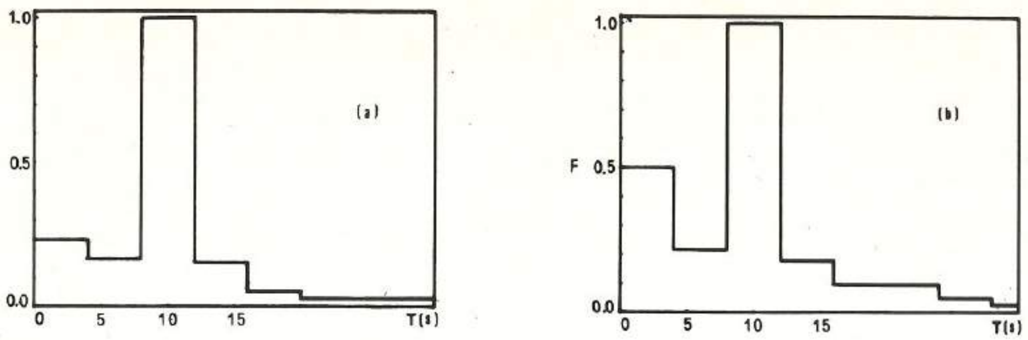


Fig. 5 - Normalized distributions of the interdeparture (a) and interarrival (b) time intervals for a NT experiment (Zurich).

msec for each crossed node (tab. 1) [4],[5].

All the NDD's have been received from the Milan node : the NDD response time behaves in the same way of the TRACE response time (fig. 7); whereas the DC and NT response time shifts on greater values than the NDD response time (fig. 7 and tab.2); moreover, the NT response time is greater than the DC response time (fig. 7) because the DC process has an higher priority than the NT process.

4.2. *LOSS AND DUPLICATION*

The percentage of incomplete TRACE's due to the loss of TRACE packets is shown in tab. 3 ; it increases for the more distant nodes and it becomes very high in CIGALE.

The duplication is a very rare event which has been observed only once in an experiment on EIN : the duplicated packet was generated by the Paris node.

The percentage of the DC, NT and NDD responses in all the experiments excepted the last is shown in tab. 4a ; the last experiment was mostly performed with the subnetwork partitioned : the percentage of the DC, NT and NDD responses in this experiment is shown in tab. 4b .

Let us note that the DC packets have been always received when the relative NT packets have been received, but not the contrary.

For all the experiments excepted the last, the percentage of the incomplete transactions dealing with the NT facility, because no answer or not complete answer has been received, is shown in tab.5.

The sample amplitude (i.e. the number of packets) is 787 for all the experiments excepted the last, and it is 250 for the last experiment. In the first case the following parameters can be well estimated [6] :

- p : probability that a packet sent to a VSCP is lost;
- d : probability that a DC is lost;
- r : probability that a packet created by a VSCP (that is a NT or ECHO packet) is lost.

In the second case the following parameter can be well estimated [6] :

- n : probability that a NDD packet is lost.

Tab. 6 shows the estimated values with confidence levels of 99% and 95%. Let us note that the probabilities d and n could be also equal to zero, whereas p and r are certainly different than zero.

The way in which the experiments have been executed does not allow us to evaluate if the packet loss occurs in the subnetwork or in the connection between SCM and NSC.

4.3. *ROUTES*

The percentage of the routes followed by the packets in EIN is shown in tab. 7 . The routes of the packets from Milan to other EIN nodes are nearly the same. Usually the line Milan-Zurich is favoured to the line Milan-Ispra by the packets directed to more distant nodes and in some cases also by the packets directed to the Ispra node.

The percentage of the routes followed by the packets in CIGALE is shown in tab. 8 . The routes of the packets in CIGALE are always the same. Let us note that the line 61-21 is favoured to the line 61-81 by the packets directed to 41 node (that is Grenoble).

4.4. *NETWORK TIME PRECISION*

The network time is expressed in units of 400 msec with a precision of about 7%. Each EIN NSC periodically sends service packets to the neighboring NSC's in order to synchronize the clocks : the synchronization takes place to the higher values. For this reason, it can be expected that the real rate should be a bit lower than the nominal one.

Tab. 9 shows the measured rates for each EIN node in the three longer experiments. The accuracy of the measured rates depends on the delay between the transmission of a packet to the VSCP NT and the reception of its response [6].

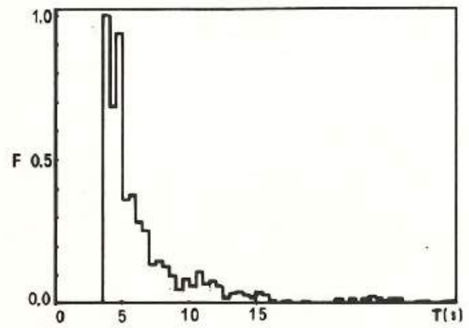
5. CONCLUSIONS

The purpose of this paper was to present the results of some measurements on the subnetwork facilities of the EIN and CIGALE computer interconnected networks.

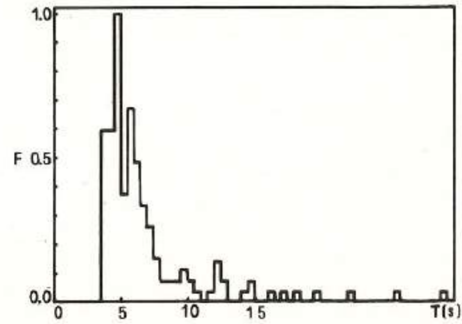
The measurements were taken at CREI in Milan by means of the Subnetwork Control Module managed by a TTY at the University of Catania. The collected data characterize the network behaviour as seen from the UNIVAC Host-computer in Milan; in fact the meas

Tab. 1 - Average response time (in seconds) of the TRACE facility : with 63% (a) and with 91% (b) of the responses (the lower ones).

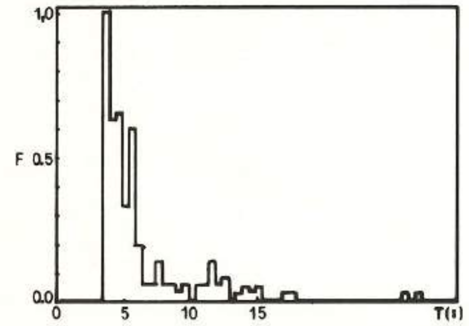
	M	Z	I	L	P	El	61	21	81	41
(a)	4.20	4.51	4.43	4.80	4.91	5.20	7.02	6.93	6.82	7.14
(b)	5.30	5.66	5.53	5.90	6.40	6.66	8.98	8.38	8.89	9.04



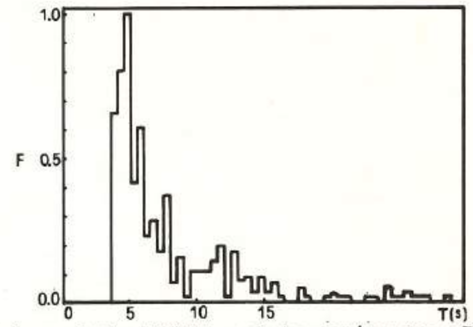
a) Delay of the TRACE packets coming from Zurich



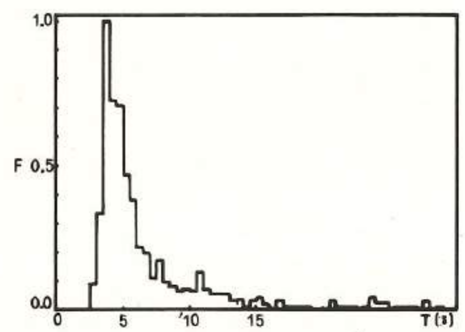
b) Delay of the TRACE packets coming from London



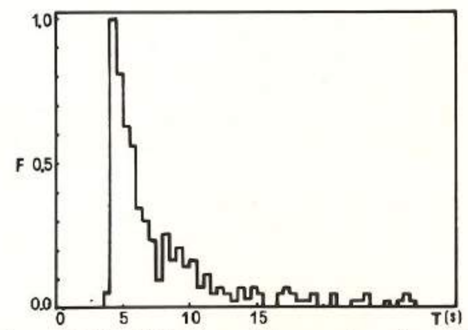
c) Delay of the TRACE packets coming from Ispra



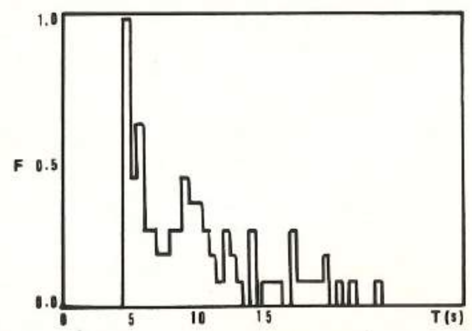
d) Delay of the TRACE packets coming from Paris



e) Delay of the TRACE packets coming from Milan



f) Delay of the TRACE packets coming from El

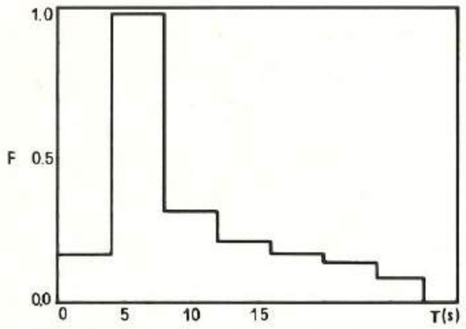


g) Delay of the TRACE packets coming from 61

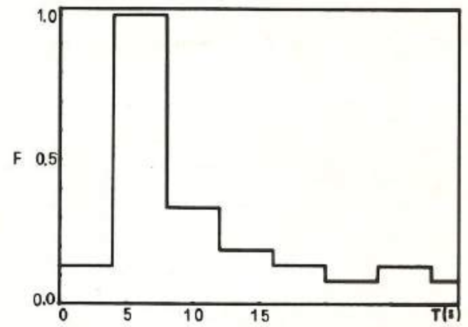
Fig. 6 a,b,c,d,e,f,g - Normalized distributions of the delay of the TRACE packets.

Tab. 2 - Delay (in seconds) of the DC, NT and NDD packets : average value m and r.m.s. σ .

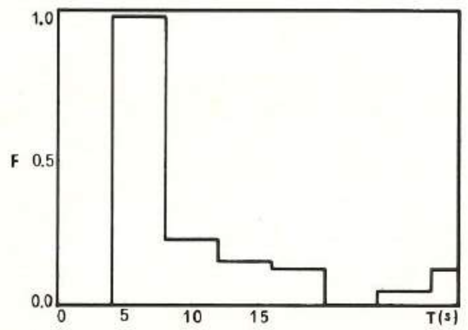
	Z	L	I	P	M	EIN
m_{DC}	9.82	10.29	9.77	11.31	9.75	9.78
σ_{DC}	6.27	6.95	6.30	6.54	6.62	6.19
m_{NT}	11.37	11.32	11.26	12.31	15.72	12.33
σ_{NT}	6.83	6.52	6.11	6.71	5.37	6.59
m_{NDD}	-	-	-	-	4.40	4.40
σ_{NDD}	-	-	-	-	2.70	2.70



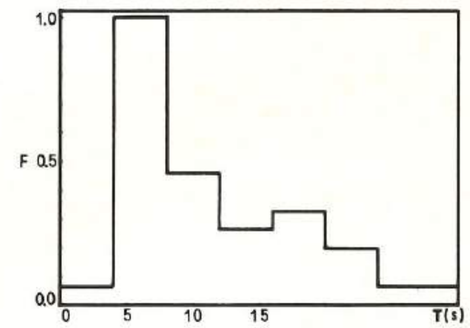
a) Delay of the DC packets from Zurich



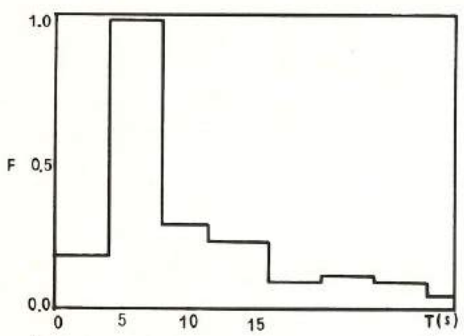
b) Delay of the DC packets from London



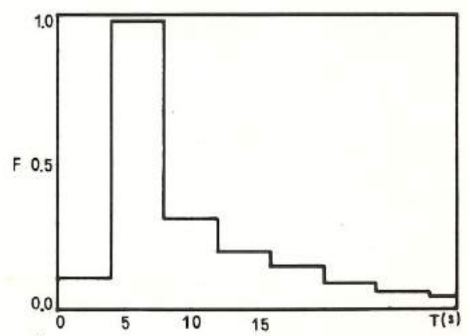
c) Delay of the DC packets from Ispra



d) Delay of the DC packets from Paris

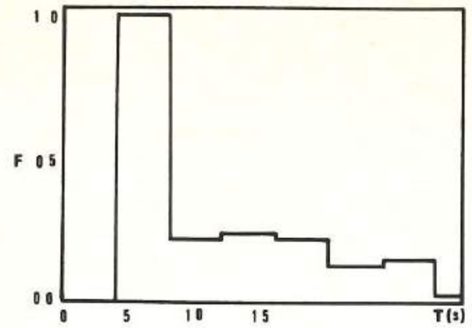


e) Delay of the DC packets from Milan

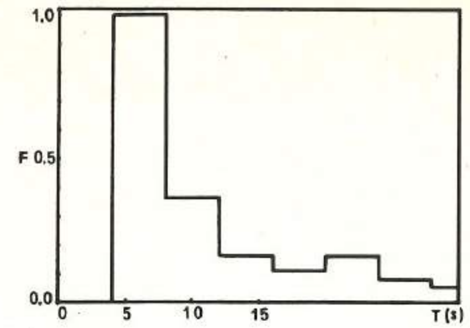


f) Delay of the DC packets from EIN

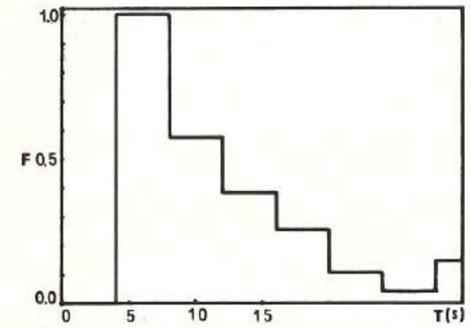
Fig. 7.1. a,b,c,d,e,f - Normalized distributions of the DC packet delay.



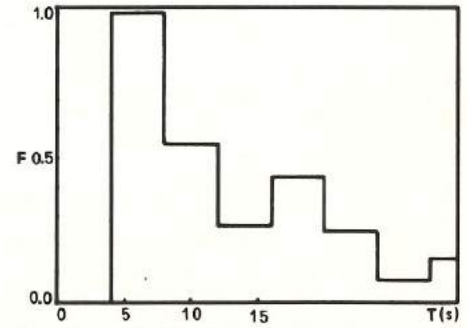
a) Delay of the NT packets from Zurich



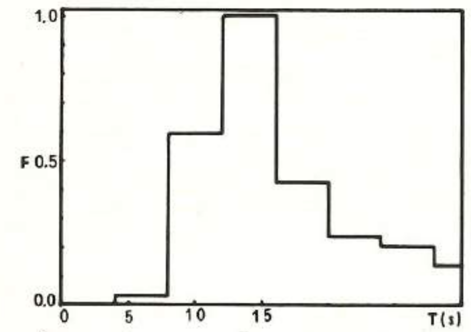
b) Delay of the NT packets from London



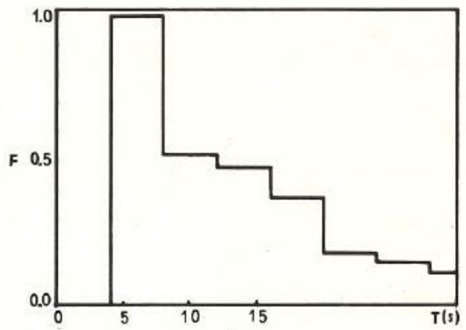
c) Delay of the NT packets from Ispra



d) Delay of the NT packets from Paris



e) Delay of the NT packets from Milan



f) Delay of the NT packets from EIN

Fig. 7.2. a,b,c,d,e,f - Normalized distributions of the NT packet delay.

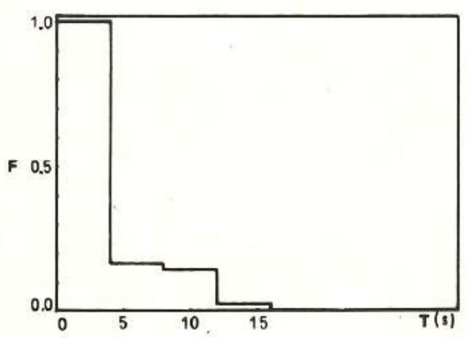


Fig. 7.3. - Delay of the NT packet after the DC packet from EIN.

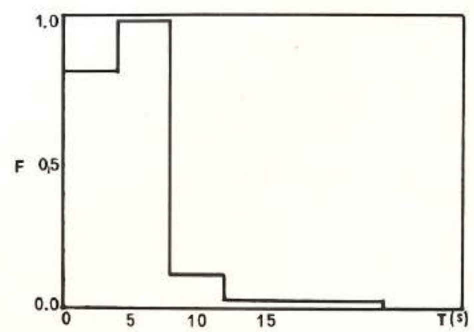


Fig. 7.4. - Delay of the NDD packets from Milan.

Tab. 3 - Percenatge of inclompete TRACE's

Z	L	I	P	E1	61	21	81	41
1.9	5.9	0.0	3.9	26.25	67.14	67.53	68.57	68.35

Tab. 4a - Percentage of the NT (a), DC (b) and NDD (c) responses for all the experiments excepted the last

	Z	L	I	P	M	EIN
(a)	96.43	94.41	92.36	95.54	92.36	94.92
(b)	97.62	98.14	92.99	96.18	98.61	96.70
(c)	0.0	1.24	5.10	0.0	-	1.27

Tab. 4b - Percentage of the NT (a), DC (b) and NDD (c) responses in the last experiment.

	Z	L	I	P	M	EIN
(a)	94.00	50.00	0.0	0.0	100.	47.60
(b)	94.00	50.00	0.0	0.0	100.	47.60
(c)	6.00	50.00	100.	100.	-	52.40

Tab. 5 - Percentage of no answer (a) or incomplete answer (b) for all the experiments excepted the last.

	Z	L	I	P	M	EIN
(a)	2.38	0.62	1.91	3.82	1.39	2.03
(b)	3.57	2.48	2.55	4.46	6.25	3.81

Tab. 6 - Estimated probabilities of loss of packets with 99% (a) and with 95% (b) confidence levels.

	p,(%)	d,(%)	r,(%)	n,(%)
(a)	4.09±2.47	0.45±0.45	2.30±1.39	2.91±2.91
(b)	3.76±1.84	0.26±0.26	2.12±1.04	1.64±1.64

Tab. 7 - Percentage of routes followed by the packets in EIN.

Destination	Routes	Percentage
Zurich	M-Z	100.00
London	M-Z-L	99.50
	M-I-P-L	0.50
Ispra	M-I	96.60
	M-Z-P-I	3.40
Paris	M-Z-P	88.74
	M-I-P	11.26

Tab. 8 - Percentage of routes followed by the packets in CIGALE.

Destination	Routes	Percentage
21	E1-61-21	100.00
41	E1-61-21-41	100.00
61	E1-61	100.00
81	E1-61-81	100.00

Tab. 9 - Measured clock rates of the network time (in msec).

Date of the experiment	Z		L		I		P		M	
	from	to	from	to	from	to	from	to	from	to
780907	374	377	371	375	370	375	372	377	371	380
780913	371	385	373	378	373	378	371	376	370	378
780914	384	387	385	388	382	394	387	389	382	388

urement system has certainly influenced the histograms of the response time owing to the introduced delay (about 2.5 - 3.5 s) and the probabilities regarding the loss and the duplication of the packets.

Nevertheless not all the results have been directly influenced by the modalities of the collected data; indeed many elaborations of the results allow us to evaluate some important network parameters and also give useful information on the subnetwork behaviour as the trend of the response time for the TRACE packets as a function of the source-destination node distance, the influence of the synchronization technique on the clock rates for the NT in the nodes, the routes and the most favoured lines followed by the packets in EIN and CIGALE and the dependence of the response time histograms on the priority of the processes implemented in the nodes.

Further implementations of SCM in other Italian SC's are envisaged so that the present measurements integrated with those obtained from other possible experiments can lead us to characterize better the influence of the measurement system on the collected data.

This work was partially supported by CNR (Consiglio Nazionale delle Ricerche) Contract n. 78/00767.

The paper was received on April 3, 1979.

REFERENCES

[1] A. Faro, G. Le Moli, E. Repossi : Specifications of the Subnetwork Control Module for EIN. EIN/CREI/77/007

[2] A. Faro : CREI implementation of the EIN software:the Subnetwork Control Module. EIN/CREI/77/008

[3] A. Gambaro : Protocolli di linea:BSC,HDLC, EIN-HDLC. Corso di aggiornamento sulle reti di calcolatori,Politecnico di Milano,Nov.77,report n.19

[4] L. Kleinrock : Queueing systems. Vol.II, Wiley Sons, New York 1976, p.457

[5] M. Gien, J.L. Grangé : Performance evaluations in CYCLADES. Proc. of the fourth international conference on computer communication, Kyoto 26-29 September 1978, p.23-31

[6] G. Scollo : Realizzazione di misure su alcuni aspetti della rete di calcolatori EIN. Facoltà di Ingegneria, Università di Catania,degree thesis 77

Contributors

Derek L. A. Barber, at present director of the European Informatics Network Project, from October 1975 to June 1979 was chairman of IFIP WG 6.1. He is co-author of three books and has had around 100 papers published over the last ten years. Before joining EIN Mr. Barber was head of Information Systems Branch - NPL, with responsibility for managing research in computer networks and information systems.

Fausto Caneschi was born in Arezzo in 1950. He obtained his degree in Electronic Engineering at the University of Pisa. He joined CNUCE in 1977 and started working on the RPCNET project. In 1978 he joined IIASA (International Institute for Applied Systems Analysis in order to study problems concerning high-level protocols.

Since February 1979 Dr. Caneschi is back at CNUCE, where he has the responsibility of the RPCNET upgrading, and also works in the theoretical field of high-level protocols. Dr. Caneschi is a member of AEI.

Donald W. Davies, graduated at Imperial College, joined the team which developed the ACE pilot model and then the ACE computer; he applied these early computers to traffic simulation, then led research in machine translation and other subjects. Since 1965 he worked on packet switching and since 1978 on data security in networks.

He is author and co-author of 3 books (the most recent was: Computer Networks and their Protocols) just published by John Wiley.

Carlo Di Filippo received the Dr. degree in Statistics from the University of Roma and then got the specialization in Calcolo Automatico at the CNUCE of Pisa. He is currently working in the fields of information retrieval, computer networks and front-ending with the Sperry Univac E.D.P. consultant.

A. Dunki studied theoretical Mathematics, receiving a B.A. in 1966 from St. Joseph's College, N.Y., and an M.A. in 1968 from Boston College. From 1968-1971 she worked with a medical research group at Roosevelt Hospital, N.Y., developing an on-line monitoring system for the critically ill. She then worked with Control Data Corp., Zurich, developing communications software. In 1973 she joined a research project in Switzerland participating in the European Informatics Network. Here she assumed responsibility for a front-end connection of the ETH's multi-mainframe system to the network, developing a specialized communications system for this task. She has used this system to back up her work in protocol design with implementations of a Terminal Service and Virtual Terminal. She is now working on the communication problems of an on-line private banking network for the Union Bank of Switzerland.

Erina Ferro was born in Pisa in 1951. She obtained her degree in Computer Science at the University of Pisa in 1974.

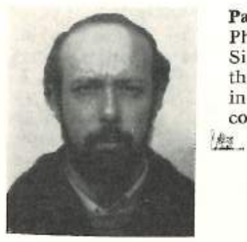
After some teaching at the University, she joined CNUCE in 1976. She participated in the RPCNET project, and now is part of the team of the STELLA (Satellite Transmission Experiment Linking LAboratories) project.

Luciano Lenzini was born in Lucca in 1944. Obtained degree in Physics at the University of Pisa in 1969 and for the whole of that year was assistant professor of Physics at the same University. In 1973, spent a year in the United States, at the IBM Scientific Center, Cambridge, Mass. where he worked on computer communications network design and techniques for distributed computing with reference to the RPCNET design.

Since 1978 he has been leading the Italian part of the international STELLA (Satellite Transmission Experiment Linking Laboratories) project. Since 1974, he has been the manager of the Distributed Systems Division of CNUCE.

Mario Mangoni was born in Arezzo (Italy). He received the Dr. Ing. degree in Electrical Engineering from the University of Pisa in 1957.

After graduating he joined the Telecommunication Services of the PTT Ministry where he has been active in the Telegraph Department and where he now hold the position of head of the technical division dealing with special telecommunication and computer networks.

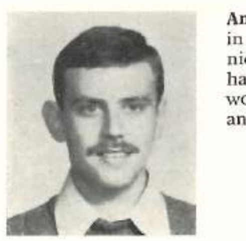


Patrizio Mapelli received the Dr. degree in Physics from the University of Milano. Since his graduation he has been with the CILEA. At the present he his working in the fields of computer networks and computer evaluation performance.

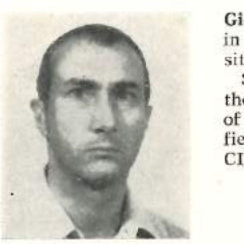
Maurizio Martelli was born in 1951 and graduated at the University of Pisa in Computer Science in 1974.

After teaching for a brief period at the University and a one year post graduate course, he joined CNUCE, an Institute of the Italian Research Council, in 1976.

Since then his interests have been computer networks and functionally distrbuted systems.



Andrea Mattasoglio received the Dr. degree in Electronic Engineering from the Politecnico of Milano. Since his graduation he has been with the CILEA, where he is working on data communication, terminal« and computer network.

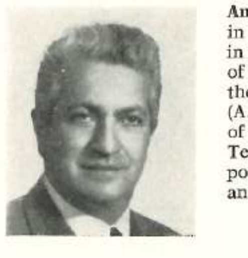


Giovanni Meloni received the Dr. degree in Electrical Engineering from the University of Pavia.

Since his graduation he has been at the Centro di Calcolo of the Politecnico of Milano and is currently working in the fields of computer management with the CILEA.

Claudio Menchi was born in Sorengo (Switzerland) in 1947. He joined CNUCE in 1970 as system analyst and worked initially in the field of computational linguistics for the D.M.I. project (Italian Machine Dictionary).

Since 1975 he has been working on the RPCNET project.



Angelo Misino was born in Bisceglie (Bari) in 1918. He received the Dr. Ing. degree in Electrical Engineering at the University of Roma in 1943. After graduation he joined the Azienda di Stato per i Servizi Telefonici (A.S.S.T.) in 1945 where he was responsible of the Maintenance Service of National Telephone Network and holds now the position of Deputy Manager of Commercial and Traffic Department.

Alberto M. Repichini, born in Roma in 1945, received the degree in Electronic Engineering from the University of Roma in 1969. From 1969 to 1971 he was assistant with the Istituto di Automatica of Roma, where he was concerned with system theory problems.

In 1972 he joined SIP - Direzione Generale, here ha has been associated with the Data Transmission Department. Now he is responsible of design of dedicated packet and/or message switched networks.

Mr. Repichini is member of CCITT and CEPT groups for the packet service standardisation in public data network and is also involved in EIN project, as member of TAG, and in EURONET project, as consultant of the Equipe de Projet.

P. Schicker received his Diploma in Mathematics in 1967 from the Swiss Federal Institute of Technology (ETH - Zurich). He spent one year at M.I.T., Boston, on the MULTICS project MAC, returning to the ETH-Z to head the systems software group of the computing center. His design of the terminal subsystem VENUS for the CDC 6000 earned him a PH.D. in Mathematics in 1976. In 1973 he became project leader of a Swiss research group participating in the European Informatics Network. Since then he has been active in the area of protocol design and network architecture, coupling numerous articles with a high degree of experimentation. He has directed the ETH-network connection which today provides daily production service based on the Zurich virtual Terminal design. He is now on leave of absence from ETH to work on higher protocols designs with Bell Northern Research in Ottawa.

Marco Sommani was born in Roma in 1949. He obtained his degree in Mathematics at the University of Pisa in 1971. In 1972, he joined CNUCE as system engineer.

Since 1975 he has been working on RPCNET and, in particular, in the applications area.

Fabio Tarini obtained his degree in Physics at the University of Pisa in 1971; after teaching for a wile at the high school, he joined CNUCE, an Institute of the Italian Research Council, in 1972, as system engineer.

Since 1974, he has been manager of the Computer Network Section of CNUCE. His present interests are high level protocols and functionally distributed system design.

Richard Winsborrow received a D.Phil from the University of Oxford in 1971. Since that time he has been engaged in the design of reliable multiprocessor systems, process control systems and the development of high level protocols for use on data communications networks.

Marco Zagolin received the Dr. degree in Mathematics from the University of Milano. Since his graduation he has been with the CILEA, where he works in the fields of computer networks and computer front-ending.

Autorizz. Tribunale di Milano 28 *agosto* 1948, *N.* 394 *del Registro*
Direttore responsabile: Dott. GIOVANNI RICCA
Proprietario: Associazione Elettrotecnica ed Elettronica Italiana

Stampa: ASP - ARTI GRAFICHE STEFANO PINELLI - Via Farneti, 8
20129 Milano - 1979