MASTER IN COMPUTER SCIENCE

A contribution to the network interconnection problem

A summary of the current state of art and a description of a practical implementation

Dunon, Paul

Award date:
1986

Awarding institution:
University of Namur

FACULTES
UNIVERSITAIRES
N.D. DE LA PAIX
NAMUR

INSTITUT D'INFORMATIQUE

Année Académique 1985 - 1986

A CONTRIBUTION TO THE NETWORK INTERCONNECTION PROBLEM:

A summary of the current state of art and a description of a practical implementation

Paul Dunon

Directeur : Philippe Van Bastelaer

Mémoire présenté
en vue de l'obtention
du titre de
Licencié et Maître
en Informatique

RUE GRANDGAGNAGE, 21, B - 5000 NAMUR (BELGIUM)

Acknowledgments

I would like to thank all the people who have contributed to the elaboration of this work. First, Mr. Ph. Van Bastelaer, my work director at the Facultés universitaires Notre Dame de la Paix, Namur, who offered me the opportunity of a stage at the European Center for Nuclear Research (CERN), Geneva, during the first semester of the academic year 1985 - 1986.

Then, all the members of the Data Division/Communication Section of the CERN, with whom I spent six months. They are:

Brian Carpenter
Mike Gerard
Kik Piney
Danny Davids
Yves Grandjean
Roch Glitho
Christian Isnard
Marie-Thérèse Monnet
Joop Joosten
Jacques Anthonioz-Blanc
Jacques Rochez
Serge Brobecker

The practical experience gained with them was an invaluable source of reflections and ideas for this work.

I also thank all those I have forgotten and who have helped me in this work.

INTRODUCTION

Within the last few years, the networking scene has evolved rapidly. This evolution is marked by many generations of networking topologies. C. Piney distinguishes three generations, each of which being characterized by offloading a part of the communication processing into a new concept [PINEY86]. In the first generation, the driving of all devices connected to a computer is trusted to intelligent controllers to relieve the central processor. The second generation is characterized by the separation of network control functions such as routing, flow control and addressing from those of general computing. This led to the concept of an autonomous communication subnetwork to which the host computers are connected. The communication subnetwork is composed of node computers connected in a mesh-type fashion. The wish to interconnect many networks was met by using point-to-point links. But this solution combines the problems of address resolution and routing between networks. The third generation solves these problems by using an intermediate (backbone) network between interconnected networks.

Since the beginning of networking, many manufacturers or user communities have developed their own network architectures. These networks are for example SNA from IBM, DECNET from Digital Equipment Corporation, XNS from Xerox Corporation, etc. They are designed by manufacturers for use between their clients and do not allow interconnection of different vendor's computers. They use their own protocols and are thus not compatible one with each others. They are usually spread over a country, a continent or the whole world and are based on a leased or switched telephone line technology.

With the increasing power and the diminishing size of computing equipments, new connection needs arose on relatively small areas, such as a factory, a campus or a single building. New high-speed technologies such as broadband coaxial cables can be used to meet these needs. The Ethernet protocol designed by DEC, Intel and Xerox in 1980 is the best known protocol used for this kind of network.

The growing number of manufacturer's protocols, being sometimes "de facto" standards and the great diversity of technologies used for computer communications incited standardization organizations such as ISO (International Standard Organization), CCITT (Comité Consultatif International pour la Télégraphie et la Téléphonie), ECMA (European Computer Manufacturers Association) or NBS (National Bureau of Standards) to work on protocol standardization.

The most important recommendation is undoubtedly the ISO Reference Model for Open System Interconnection. It sets up the notion of an Open System, a system which permits the interconnection of equipments of many manufacturers in a single network. It provides a layered model to solve the interconnection problem. Whereas the whole computing community agree on the layering principle in networking, some manufacturers still promote their own layered model, which sometimes differs from the ISO one.

Networks are commonly classified in WANs (Wide Area Networks) or LANs (Local Area Networks) following the maximum size the network can have. We agree on this classification but we think that it is preferable to consider the services provided by each layer than to consider a networking problem with two points of view : LAN and WAN.

The future of networking is not in a worldwide agreement on the ISO protocols for many reasons:

- The most important manufacturers have developed their protocols and networks before ISO or CCITT published their recommendations. Some of these have announced their intention to evolve thru ISO standardization. Others did not and it is not our subject to discuss these decisions.

- Some user communities have developed and installed multi-vendor networks. The Arpanet was designed by the American Departement of Defense to meet specific military needs. The experience gained from these "running" networks is very important for standard designing.

- Each new computing technology may create new communication needs that standards do not consider. Manufacturers will not wait future standards to satisfy user's requirements. They will design new protocols which will not necessarily be adopted by standardization organizations.

The interconnection of different networks is absolutely necessary to meet the growing communication needs, using the existing network facilities. More precisely, networks interconnection aims to:

- give users access to services or data (databases) they do not have within a single network.
- increase the connectivity between users of every network.

INTRODUCTION

The starting point of our reflections about internetworking is a work carried on at the CERN (European Center for Nuclear Research), Geneva, between September 1985 and February 1986. This work was concerned with the interconnection of many Ethernet networks via CERNET, a high-speed packet-switching network developed by the CERN for its own use. The first step of this work was the integration to an existing project and the understanding of all its components. The second step consisted of the designing and the writing of a part of the software.

This work is composed of two parts. The first part tries to be a summary of the current situation in network interconnection. This theoretical study aims at clearing up the confusing terminology and ideas about internetworking. The discussion continuously refers to the seven layers of the ISO Reference Model for Open Systems Interconnection(OSI). Each interconnection architecture is studied in a chapter and many illustrations are given.

The second part is a description of the practical experience gained at the CERN. The whole interconnection project is specified and the available tools are described. The design choices are stated and justified. An evaluation of our work in this project ends this part.

It should be noted that some references are in the bibliography even though they are not cited in the text. It has been done because all these articles have contributed to our reflection, even if their ideas are not explicitely developed or used in the text.

PART 1 : SUMMARY OF THE CURRENT STATE OF ART

Chapter 1 : THE INTERNETWORKING CONCEPT

This chapter introduces the basic notions necessary to take up a discussion about internetworking. The ISO Model for Open System Interconnection is briefly resumed. It will be used as a reference for the comparison of the internetworking architectures. The terminology and conventions adopted for the rest of this thesis are stated and explained. The principles and requirements of internetworking are set up and the four approaches to internetworking are defined. The segmentation and reassembly of a packets crossing multiple networks is a problem relevant to each of these architectures. It is studied at the end of the chapter.

1.1 The ISO Reference Model for Open Systems Interconnection

The Reference Model for Open Systems Interconnection is an international standard voted by ISO in 1981. It is fully described in [ISO7498]. This is a Reference Model because it may allow to compare existing standards networks and protocols. It is a general model, so that new computing or communication technologies can fit into it. A System designates a set of computers, peripherals, transmission media, etc, able to process and transfer informations. The expression Open Systems Interconnection (OSI) addresses a set of informations exchange standards to be used by the systems. These systems are said to be open because they use the same set of standards.

Applying the principle "Divide to conquer", the OSI Model is layered. Each layer (N) uses the services of the layer (N-1) and provides some services to the layer (N+1). The seven layers of the Model have been defined applying ten principles [ISO7498]. Zimmerman summarizes the major ones [ZIM83]:

- A layer should be created where a different level of abstraction is needed;
- Each layer should perform a well-defined function;
- The function of each layer should be chosen with an eye towards defining internationally standardized protocols;
- The layer boundaries should be chosen in order to minimize the information flow accross the interfaces;
- The number of layers should be large enough that distinct functions do not need to be thrown together in the same layer out of necessity, and small enough that the architecture does not become unwieldy.

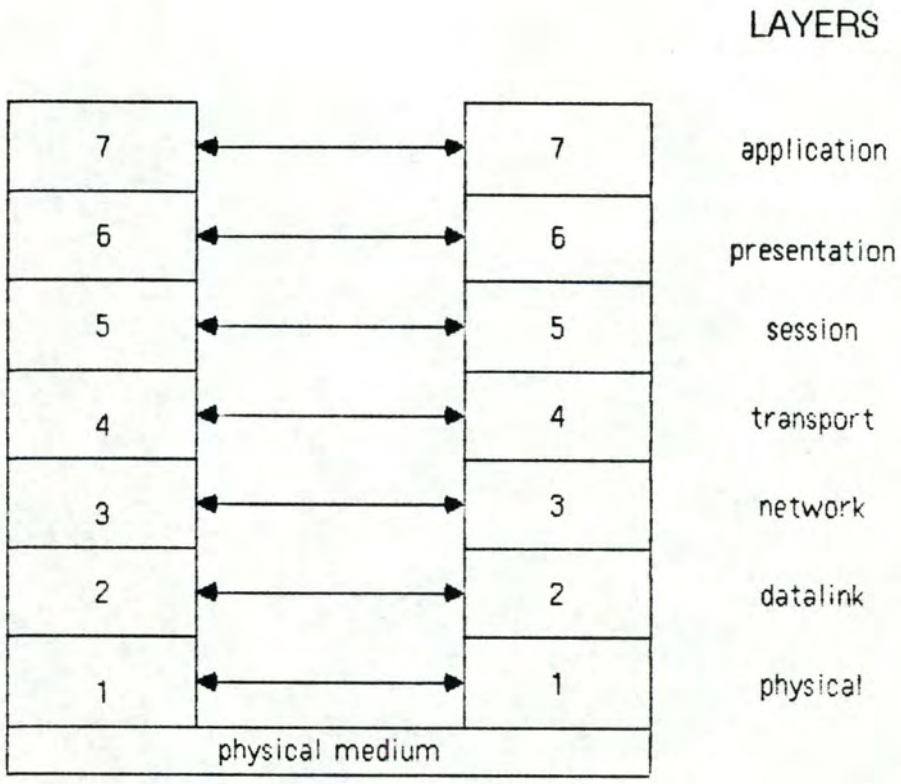


Figure 1.1: The ISO Reference Model for Open Systems Interconnection

Figure 1.1 shows the seven layers of the OSI Model. Each entity of a layer communicates with its peer, the remote entity at the same layer. The seven layers are described briefly below [ISO7498], [ZIM83].

Application Layer

The primary concern of the Application Layer is the semantics of the application. All application processes reside in the Application Layer. Examples of applications are electronic mail, remote access to a database, electronic funds transfer, virtual terminal, etc.

Presentation Layer

The Presentation Layer provides independance to application processes from differences in data representation, i.e., syntax. It means that Application entities may use different syntaxes because the Presentation Layer provides transformation of this syntax to the common syntax to be used between Application entities.

Session Layer

The Session Layer provides the mechanisms for organizing and structuring the interactions between two Presentation entities. These mechanisms allow for one-way, two-way simultaneous or two-way alternate operation, the establishment of major and minor synchronization points and the definition of special tokens for structuring the exchanges.

Transport Layer

The Transport Layer provides transparent transfer of data between end-systems. It thus relieves upper layers from any concern with providing reliable and cost-effective data transfer. It optimizes the use of Network Services and provides any additional reliability over the one supplied by the Network Service.

Network Layer

The Network Layer provides independance from data transfer technology and independance from relaying and routing considerations. It masks all the peculiarities of the actual transfer medium (optical fiber, packet-switching, satellites or LANs). The Network Layer also handles relaying and routing data through as many concatenated networks as necessary while monitoring the quality of service parameters requested by the Transport Layer.

Datalink Layer

The purpose of the Datalink Layer is to provide the functional and procedural means to transfer data between network entities and to detect and possibly correct the errors which may occur in the Physical Layer. Datalink protocols and services are very sensitive to the physical transfer technology. Typical Datalink Protocols are HDLC for point-to-point and multipoint connections and IEEE 802 family of protocols for LANs.

Physical Layer

The Physical Layer provides the mechanical, electrical, functional and procedural standards to activate, maintain and deactivate physical connections allowing the transfer of bits on a physical medium.

For the three lower layers, protocols are adopted as International Standard (IS) or Draft International Standard (DIS).

1.2 Wide Area Networks and Local Area Networks

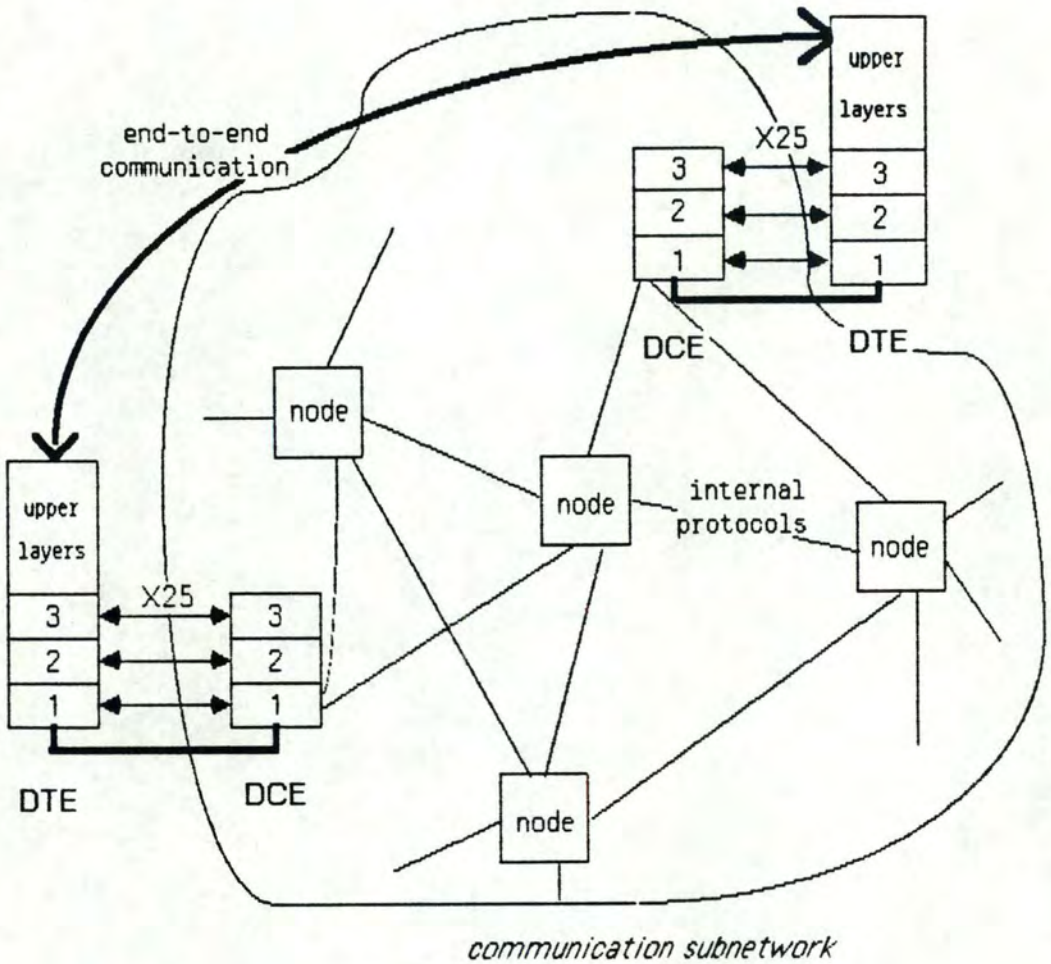


Figure 1.2: Schema of an X25 network

A well known standard mapping the OSI Model is the X25 recommendation published by the CCITT in 1980. This standard is almost universally accepted as the standard for access to the packet-switched public data networks. It provides connection-oriented operation over long-haul communication facilities, such as public telephone networks or Integrated Services Digital Network (ISDN). It is an example of Wide Area Network, also called long-distance or long-haul network. This kind of networks is characterized by [SCHEI83]:

- The covered distance (few to thousands of kilometers);
- A relatively low bit transfer rate (maximum 60000 bits per second) and a relatively high error rate;
- A large delay time due to the distance and the medium.

They are often mesh-type networks with routing, flow control and addressing problems to be solved by the communication subnetwork, consisting of a mesh of node computers. They can also use satellite(s) with high-delay and retransmission problems.

Figure 1.2 schematizes an X25 network. The standard defines the interface between a host computer, called DTE (Data Terminating Equipment) and the node to which it is connected, called DCE (Data Communication Equipment). X25 is not concerned by the protocols internally used in the communication subnetwork, nor by the protocols above layer 3 in the hosts. The protocol is connection-oriented, which means that the layer 3 entity in the DTE communicates with its peer in the DCE on a connection characterized by a virtual circuit number. The nodes of the communication subnetwork operates in a store-and-forward way. Each packet crossing the network passes from the source DTE to its DCE, from the DCE to a node, and so on up to the destination DCE which delivers the packet to the destination DTE. The service provided by X25 allows upper layers to operate on an end-to-end way.

Satellite and packet radio networks are another category of wide area networks. The main difference is that every packet sent is automatically received by every site. Each receiver must select out the packets sent to itself. The communication subnetwork is constituted by radio waves and/or a satellite which acts as a store-and-forward repeater. Only one packet may be in flight at any instant, which means that high-bandwidth medium and sophisticated medium access methods must be used. ALOHA is a satellite network designed at the Hawaii university in the early 1970's [TAN82].

Local Area Networks are often based on a broadcast technology and have been the subject of major research and development activities during the last few years. They result from the trend in computing towards distributed processing, independent workstations, personal computers and intelligent peripherals. A Local Area Network is defined as "a data communication system which allows a number of autonomous devices to communicate with each other" . The characteristics of a LAN are the following [TAN82] [SCHEI83]:

- a range of no more than few kilometers;
- a data rate of 1 to 50 megabits per second and a low error rate;
- very simple network control, addressing and routing scheme, due to the topology.

They are often the property of a single organization, which implies less security, accounting and maintenance operations than X25 public networks.

Local Area Networks may be classified following four criteria [TAN82]:

1. The topology, i.e. the connectivity in terms of mesh, star, ring, tree or linear (bus) configuration;
2. The technology, i.e. the transmission medium: twisted pairs, coaxial cable, optical fibers, microwaves, etc;
3. The medium access method, i.e. the means by which several users share the network: polling, token-passing, message slot, register insertion, random access such as Carrier Sense Multiple Access with Collision Detection (CSMA/CD), Time Division Multiplex (TDM), etc;
4. The signal transmission technique, i.e. the modulation scheme by which the bandwith of the medium is utilized: single-channel baseband or multi-channel (frequency multiplexed) broadband transmission.

The first general-purpose LAN to be commercially developed and widely accepted is Ethernet [ETH82] based on CSMA/CD, bus topology and coaxial cable. Many others are now available, such as token-ring for PCs, token-bus, etc.

Thanks to the work of the IEEE Computer Society (Institute of Electrical & Electronics Engineers), LAN protocols now fit into the OSI Model. The IEEE 802 Project defines two sublayers into the Datalink Layer (see figure 1.3). The Logical Link Control (LLC) Sublayer provides hardware-independent exchange of data between two LLC entities. Two protocols are defined for this sublayer. The first one, LLC1, provides for connectionless unacknowledged service. The second, LLC2, is a connection-oriented protocol with flow control and acknowledgment, nearly equivalent to HDLC, the Datalink Layer Protocol of the X25 standard. The Medium Access Control (MAC) Sublayer is concerned with functions depending of the communication medium and of the technique used to access to it.

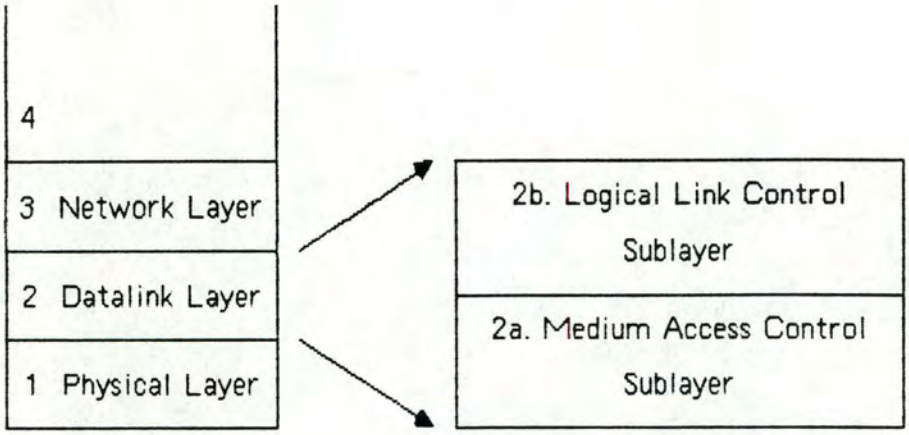


Figure 1.3: The two Sublayers of the Datalink Layer

Up to now, three MAC standards have been developed, each of which specifying a hardware-dependent method of point-to-point data transfer. All MAC protocols provide a similar interface to the LLC Sublayer.

Figure 1.4 illustrates the family of standards of the IEEE 802 Project(a). The 802.3 standard is nearly similar to Ethernet; 802.4 specifies a MAC by token-passing on baseband or broadband coaxial cable; 802.5 is the standard for token-passing protocol on a ring. These standards also include Physical Layer specifications, due to the close dependence of the medium and the access method.

About the future of these standards, it seems that the token ring on coaxial or optic fiber cable will be adopted in more and more commercial products, whereas 802.3 is now more generally available, due to Ethernet precedence.

The last problem to consider in our LANs and WANs review is the difference between a node and a host. In packet-switching networks, nodes are the components of the communication subnetwork. They provide switching and routing of packets from the source host to the destination host. But, in the case of an Ethernet network, communicating stations or computers are all directly connected to the medium. There is no physical switching

---

(a) The IEEE 802 family of standards is now also designated with the generic ISO number 8802.

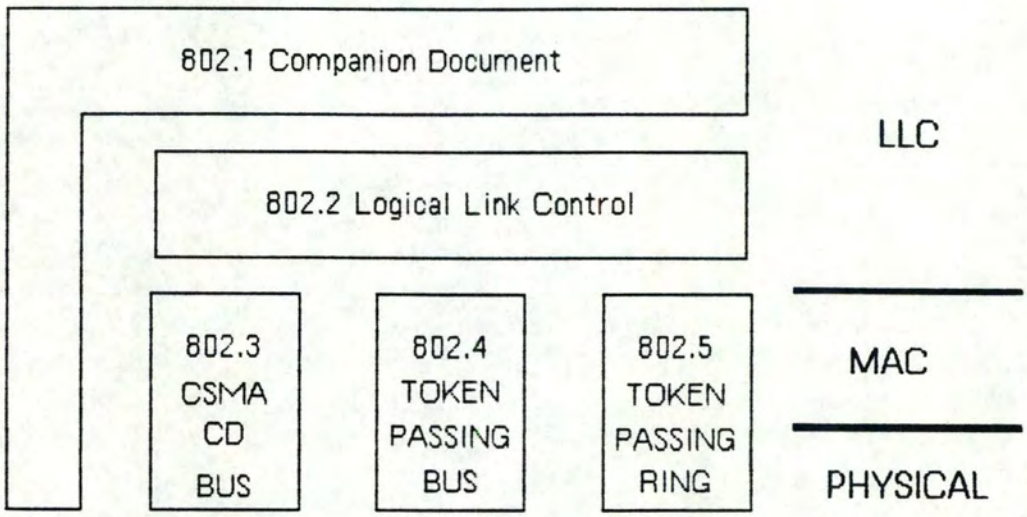


Figure 1.4: The family of standards of the IEEE 802 Project

units because only one way is possible between two stations. This situation is the same for all LANs topologies. We will consider that, in case of LANs, hosts are also nodes because they must select packets which are addressed to themselves, using address recognition mechanisms. In packet-switching networks, the DCE (the node) assumes this function.

1.3 Introductory example

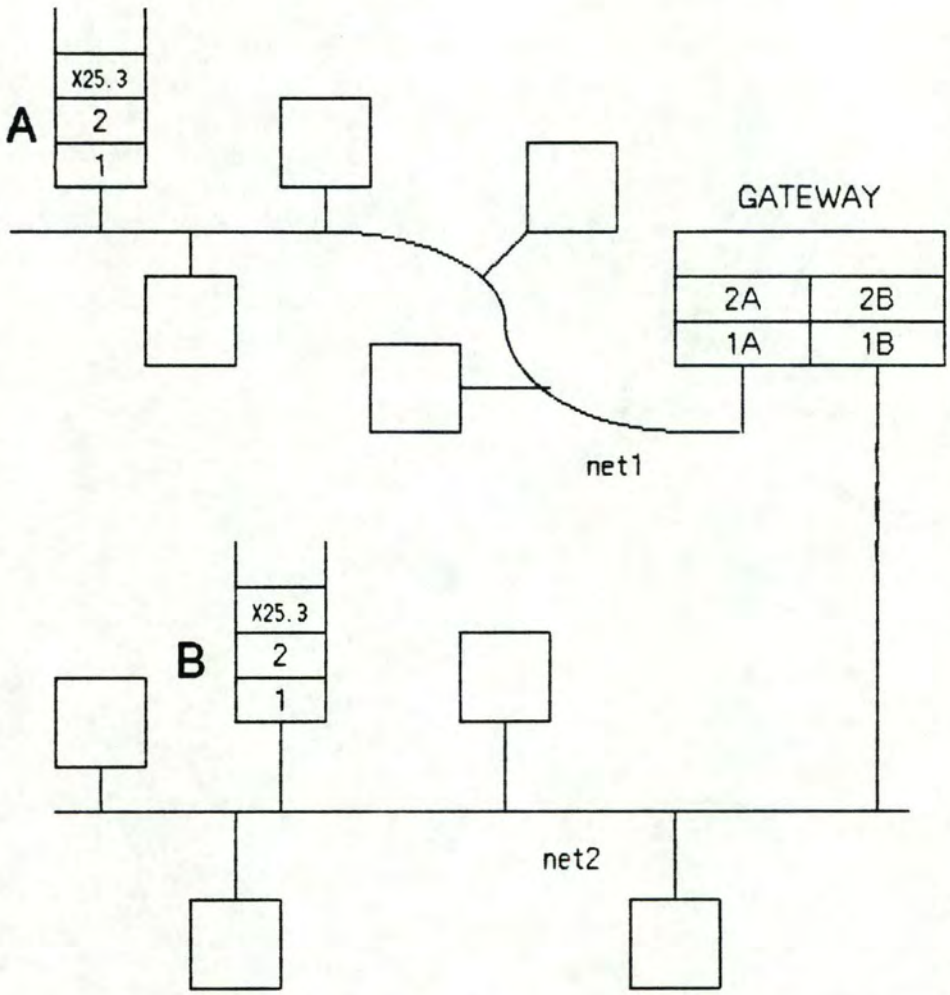


Figure 1.5: Interconnection of two X25 networks based on Ethernet

Let us consider two X25 networks and let us suppose that we want to interconnect them in such a way that users of one network can communicate with users on the other network. We can imagine many ways of realizing the interconnection. In this introductory example, we use the term gateway as a general term to designate the physical unit used to interconnect networks.

First, suppose that the two X25 networks are build on top of Ethernet protocols and topology. To interconnect the two networks, we may use a station whose peculiarity is to be physically connected to the two Ethernet cables, as illustrated in figure 1.5.

What happens when Mr. A wants to communicate with Mr. B ? An upper layer of A asks the establishment of a virtual circuit with B, to whom it gives the X25 address. The Layer 3 of A build a Call Request Packet with this address in the destination address field. It then asks Layer 2 for sending this packet to the Ethernet address of B (The Layer 3 knows or is able to find the Ethernet address of B !). The Layer 2 "encapsulates" the packet into a frame and sends it on the cable. Because the destination address of the frame does not exist on the network, no one station will read the frame, except the gateway which is able to read all frames on the cable. The gateway recognizes the destination address of this frame as existing on the other network to which it is connected. It thus sends this frame on the network 2. Mr. B's Layer 2 receives it , strips off the packet from the data part of the frame and passes it to the Network Layer. If Mr. B accepts the connection demand, the Layer 3 sends a Call Accepted packet which follows the same way to arrive to A, as do all the following packets. This way of interconnecting two networks is called a bridge.

If the two X25 networks are more classicaly based on mesh-type communication subnetworks, one may interconnect the two networks with a node being part of the two communication subnetworks. Figure 1.6 shows this situation. If Mr A. wants to communicate with Mr. B, the A's Layer 3 sends to its DCE a Call Request Packet, with the address of B in the destination address field. The DCE receives this packet and sees that the called address is not in its network. The DCE knows that, in this case, the packet must be routed to a special node, the gateway. The Layer 3A of the gateway, on reception of the Call Request, memorizes the virtual circuit number and passes the packet to the 3B entity for sending on the following network. The B's DCE may eventually change the virtual circuit number of the Call Request before passing it to B. If B accepts the connection request, it sends a Call Accepted packet, which follows the opposite way. The mapping of the two virtual circuit numbers is done by the gateway.

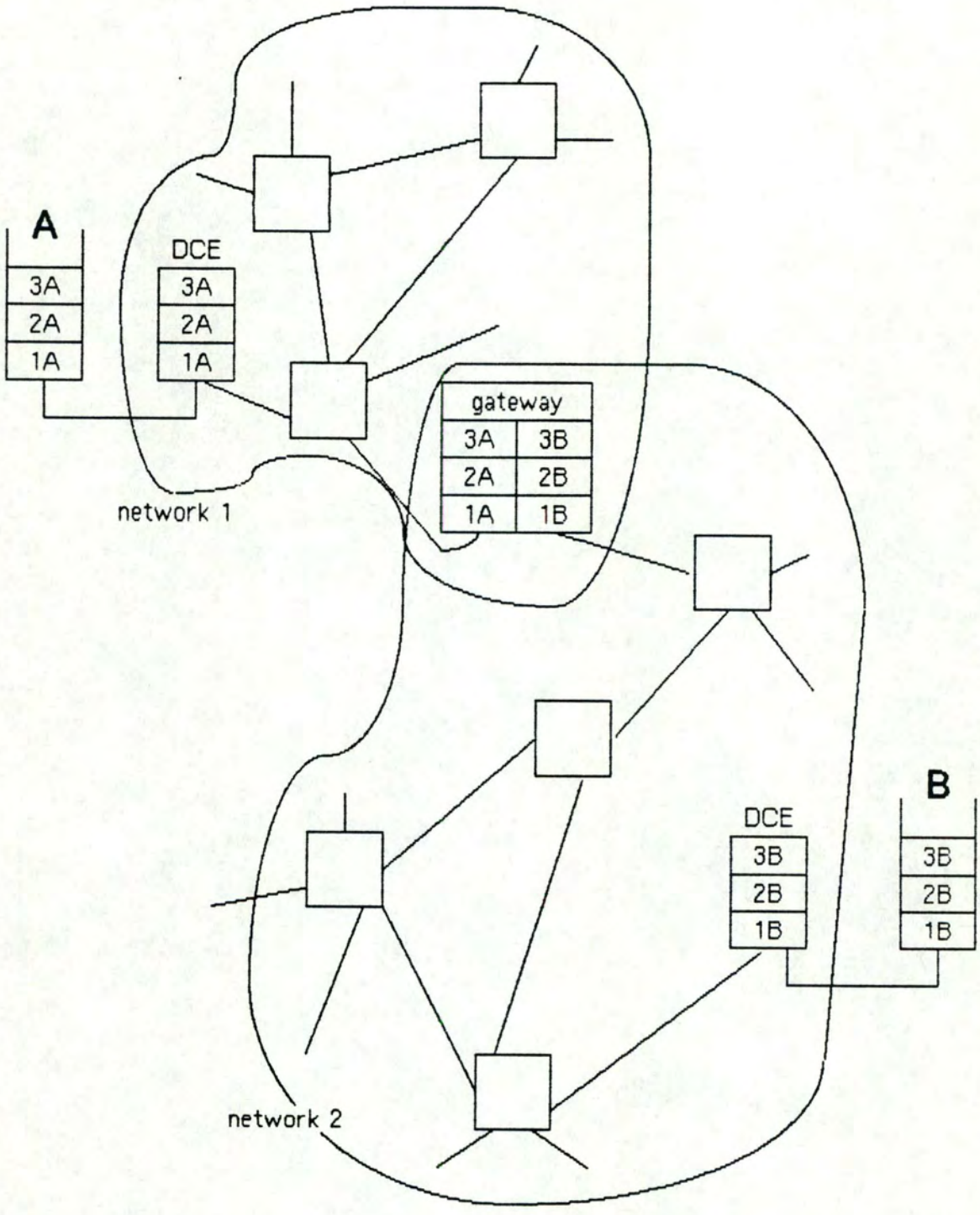


Figure 1.6: Interconnection of two X25 networks with a joint node

Up to now, we have interconnected two X25 networks without changing anything in the existing protocols of the two networks. The interconnection is transparent for the Layer 3 communicating entities, which only sees an enlarged X25 network.

An evolution of this last solution is to split up the two logical parts of the gateway in two physically independent units. These two units may be implemented each in a node of the communication subnetwork and are tied together by a point-to-point link, as illustrated in figure 1.7. A Layer 2 protocol is of course necessary to manage this link.

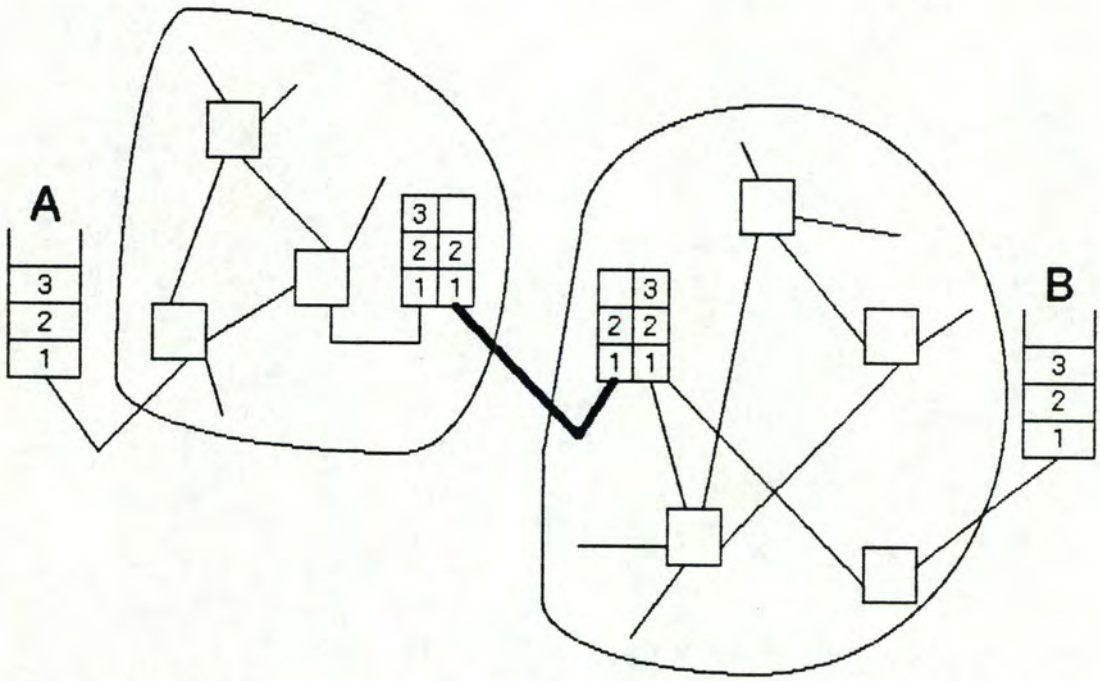


Figure 1.7: Two gateway nodes interconnected by a point-to-point link

A very different way of interconnecting two networks consists of designing a particular host (a DTE), whereas up to now we did the interconnection with a special node. In this case, the Network Layer explicitly dialogs with the gateway station. If Mr. A on network 1 wants to communicate with Mr B. on network 2 (figure 1.8), he has to begin asking to the Layer 3 the establishment of a connection with the station G. Once the virtual circuit is set up, the upper layer of A may send to the gateway an information meaning: "Would you like to send this data to Mr. B". This information is carried through network 1 in a data packet of Layer 3 protocol and is thus interpreted above the

Network Layer, as well in station A as in the gateway. On reception of such information, the gateway opens a connection with station B and sends to B the data it has received from A.

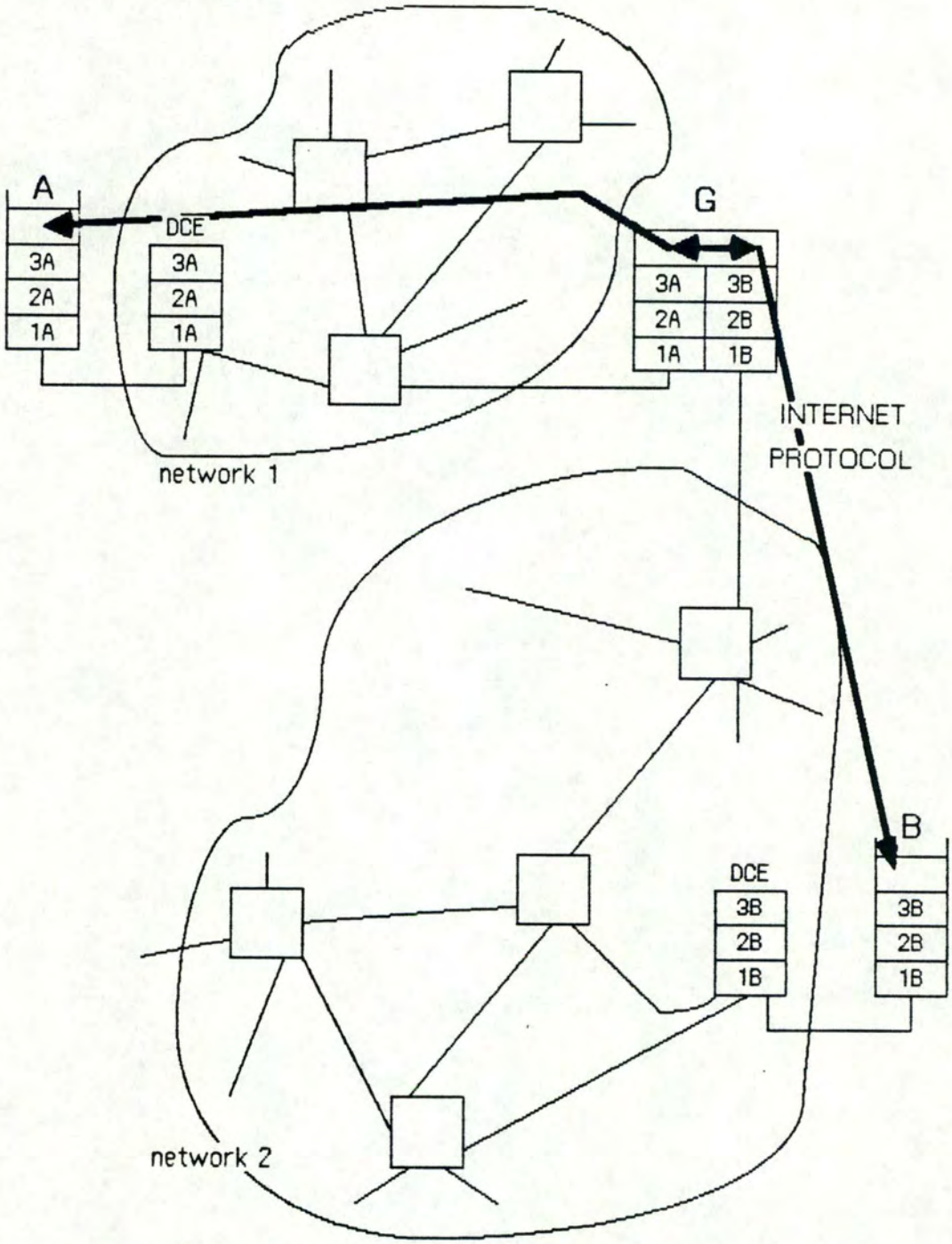


Figure 1.8: Two networks interconnected by specialized station

In summary, this type of internetworking uses the services of the Network Layer to exchange control information and data between the source, the gateway and the destination. A set of conventions, i.e. a protocol, is used above Layer 3 and is to be understood by the stations of the intercommunicating networks and by the gateways. This protocol is implemented in an additional layer, just above Layer 3 and often called Internet Protocol. This internetworking technique also involves an overall internetworking addressing scheme, added to the Network Layer addresses.

The principle of the Internet Protocol may be applied to interconnect networks even if they are not of the same kind. But it involves additional problems if the Network Services of the two networks are not identical. These problems result from a different maximum packet size, a connection-oriented or a connectionless service, etc. In these cases, the gateway has to be more intelligent to deal with these differences and to provide a uniform and performant service to the upper layers.

The Internet Protocol implies some modifications in protocols architecture to be installed in already existing networks.

Finally, two networks may be interconnected at a higher layer than Network Layer. In this case, the interconnection technique is not as general as before but is "application-dependent". It means that the interconnection is made between two Application Layer entities, e.g., two electronic mail systems. The gateway must provide a transformation of mail messages from one network format to the format of the other network. Many other problems may have to be solved because the concepts used at the Application Layer may be a lot more incompatible than these of the Network Layer.

These few examples, from the simplest to the most sophisticated, demonstrate, as needed, that the interconnection problem may be solved in very different ways, depending on the technology of the networks, their layering compatibility, etc.

1.4 Requirements of internetworking

Internetworking may be defined as "a set of hardware and software resources aiming at giving users of a network the access to services and users on another network." The interface between networks must provide the following facilities [STAL83] [STAL85a]:

- A link between networks;
- Routing and delivery of data between processes on different networks;
- Accounting of inter-networks traffic.

To provide these overall requirements, the interface must accomodate differences between interconnected networks:

1. Addressing schemes: An Ethernet address is different from an X25 one.

2. Maximum packet size: When a packet of 1520 bytes length must cross a network for which the maximum packet size is 256 bytes, it has to be broken into smaller packets. This process is referred to as segmentation in ISO terminology and is studied at the end of this chapter.

3. Network access mechanism: the methods used to access the network are closely dependent of the medium: Time Division Multiplex in a satellite network, token-passing in a bus or a ring cable, random access in a point-to-point mesh-type network.

4. Network speed: from 2400 bits to 50 Megabits per second. It implies special timing procedures to accomodate very different timeouts and to avoid congestion at the interface between two networks.

5. Network services: from unacknowledged datagram service to acknowledged connection-oriented service.

All these differences are relevant to the Network Layer or the Datalink Layer. It is obvious because we consider the differences between networks, without any regards to the final use of the networks, i.e. the Application, Presentation and Session Layers. The Network Layer is the "last" layer affected by the technology of the network. But it does not mean that the interconnection has to be realized at the Network Layer ! Interconnection at higher layer is more application-dependent, thus less general or wide-range, but every layer interconnection is "a priori" feasible.

1.5 Internetwork architectures

To define and classify many internetworking architectures, we need well-defined criteria to do this. C. Piney uses the layer at which interconnection is made as the main criteria [PINEY86]. Other authors build their interconnection techniques on the differences between LANs and WANs. Norman F. Schneidewind applies the following principle in designing the interface between two networks [SCHNEI83]:

> "The more a local network is designed to increase the effectiveness of intra-local network communication, the more the cost of the interface to a long-distance network increases and the more the effectiveness of inter-local network communication decreases."

But he also recognizes the importance of layering: "The extent to which local and long-distance networks adhere to the ISO model has an important bearing on the nature and complexity of interconnection".

V.G. Cerf. and P.T. Kirstein define a set of issues which "must be resolved before a coherent network interconnection strategy can be defined" [CERF78]. These issues are:

- level of interconnection;
- naming, addressing and routing;
- flow and congestion control;
- accounting;
- access control;
- internet services.

But the main one of these seems to be the level of interconnection, because:

> "The level at which networks are interconnected can be determined by the protocol layers terminated by the gateway. (...) Those layers which are terminated by the gateways could be different in each net, while those which are passed transparently through the gateway are assumed to be common in both networks." [CERF78]

William Stallings determines a gateway architecture by [STAL85a]:

- The nature of the interface.
- The nature of the transmission service.

"The interface is the physical attachment between networks. These can be linked at the node level (DCE) or at the host level (station, DTE). The transmission service can be either end-to-end or network-by-network. The end-to-end approach assumes only that all networks offer at least an unreliable datagram service; that is, if a sequence of packets is sent from one station to another on the same network, some but not necessarily all will get through, and there may be duplications and reordering of sequence. The transmission across multiple networks requires a common end-to-end protocol for providing reliable end-to-end service. In the network-by network transmission service, the technique is to provide reliable service within each network and to splice together individual connections across multiple networks. The combination of interface and transmission levels results in four possible gateway architectures, as illustrated in figure 1.9 [STAL85a]."

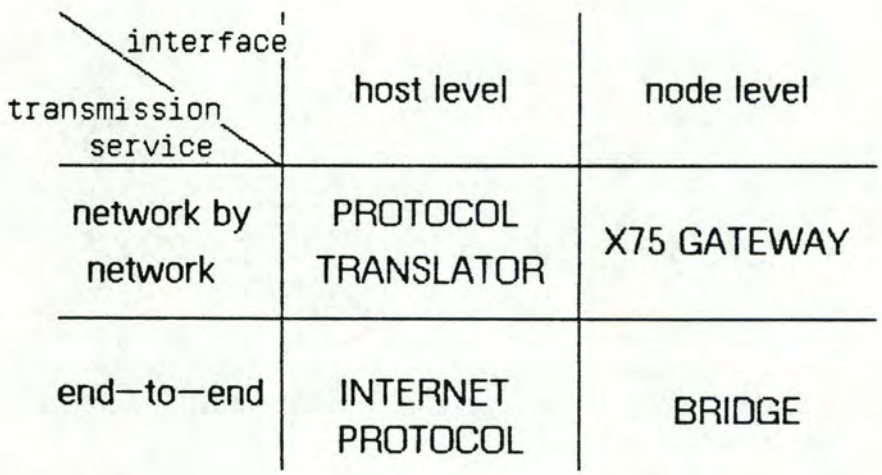


Figure 1.9: The interconnection architectures of W. Stallings

We understand the opinion of W. Stallings in the following way. The distinction between the node level and the host level is quite clear for any packet switching network. It is less obvious for networks not based on a packet-switching communication subnetwork, such as LANs defined by the IEEE 802 Project. But we have already stated our opinion about this problem (see the end of section 1.2.). We consider that hosts connected to an Ethernet can be functionally viewed as hosts and nodes, because they are both end-users of the network, and also perform at least one function of a node: selection of packets addressed to a host, i.e., themselves.

The transmission service is to be end-to-end when it allows two end-users to communicate, whatever the medium (or media) by which the two users are connected. For example, in the context of a national public data network, the CCITT X25 protocol provides an end-to-end transmission service. When two X25 networks are interconnected, some features are necessary to map the two X25 protocols. The transmission service is thus said to be network-by-network because a device performs some relaying functions between the two protocols. The X75 protocol performs such functions. The transmission service is end-to-end when a single protocol allows two users to communicate across multiple networks. The IP protocol of the American Department of Defense (DoD) is an example of such a protocol.

A consequence of this transmission service difference is that, in the end-to-end transmission service, any control information defined by the protocol has still end-to-end significance. The transmission service provided by an interconnection technique is said to be network-by-network when control information has only local significance to the originating network.

Our classification is not as strict as the one of Stallings, not as technical as the one of Schneidewind and more precise than the one of Cerf and Kirstein. The two criteria are the layer of interconnection and the level of interconnection (what Stallings calls the nature of the interface).

The level can be the node or the host. We have explained in the introductory example the problem of two X25 networks interconnected by a gateway node or by a gateway host. The gateway node is not addressed by the two communicating Network Layers (nodes are never addressed at Network Layer !). The gateway host is addressed at Network Layer by communicating hosts. They explicitly dialog with the gateway host to ask it to forward data to the destination host.

The second criterion is the Layer of interconnection in terms of the OSI Model. Two networks may be, in theory, interconnected at any layer of the Reference Model. The choice of a solution depends on technologies and topologies of networks, and the compatibility between protocols. The common principle is that, above the interconnection layer, the presence of multiple networks is invisible because this is the function of the interconnection service, whereas at lower layers or sublayers, many individual services or connections are "concatenated" by the gateways. This principle has as consequence that all layers above the interconnection layer have to be identical in order to intercommunicate, i.e., have to share unique protocols at each layer, to allow end-users to communicate.

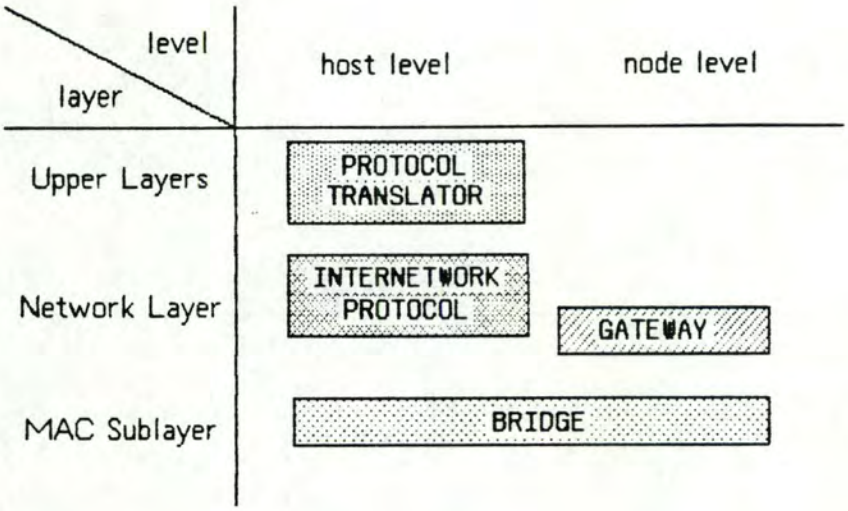


Figure 1.10: Classification of interconnection architectures

Given these two criteria, we establish our classification in figure 1.10. The Bridge is the device performing the interconnection of two networks at the MAC Sublayer of the Datalink Layer. This technique is designed for use between Local Area Networks. It covers the host level as well as the node level because we consider that the host or node level notion is not relevant in case of LANs. The Gateway interconnects two networks at Network Layer and at node level. The CCITT X75 standard is an example of such a technique. The Internetwork Protocol operates at the same layer but at the host level. It consists, in fact, of the adding of an additional Network Layer common to all interconnected networks. This internetwork Layer is put on top of the Network Layers of networks to interconnect. In this technique, we call the host performing the physical connection between networks an IN Gateway (IN = Internetwork). The last technique is the Protocol Translator. It designates all interconnections possible at layers above the Network Layer. It is necessarily made at host level because the "node" notion does not exist above the Network Layer.

These four interconnection techniques will be studied in detail in the following chapters. From now on, the terminology used in this thesis is the one defined for these techniques in the previous paragraph.

1.6 The segmentation/reassembly problem

A problem common to many internetworking architectures is the segmentation and reasssembly of a packet (or Protocol Data Unit, whatever the layer you consider) crossing a network with a small maximum packet size.

In the literature, this problem is often called "fragmentation" and is usually studied in an architecture of Internetwork Protocol. We prefer the segmentation term because it is the one adopted by ISO. We introduce the segmentation problem in this chapter because it may occur in many internetwork architecture, whereas it has been more illustrated and developed in the case of Internetwork Protocols.

The segmentation problem occurs at a gateway when it is necessary to forward a packet to the next gateway or to the destination, through a network which can not directly handle a packet of the same size. A first solution is to discard the packet, which is not always acceptable, regarding to the service provided. Another one could be to avoid this network, using another way to reach the destination, but it is not always possible and it is not a real solution. There exists two ways of providing segmentation and reassembly of segments [SHO79].

1.6.1 Intra-network segmentation

The entrance gateway can do a network-specific segmentation, breaking up the oversize packet into smaller pieces, called segments, and allowing the exit gateway at the other end of this network to do the reassembly. This segmentation is invisible to the other networks, but it has some disadvantages :

- processing and headers overhead;
- all segments must reach the same exit gateway;
- a fair amount of duplication as the same large packet is repeadly segmented and reassembled;

Additionally, the destination host may have to reassemble, when the maximum packet size of the destination network is smaller than the size of the packet to receive.

1.6.2 Inter-network segmentation

In this case the gateway may divide the packet into smaller pieces which are themselves addressed to the ultimate destination, and must be reassembled there. The so created segments need not all depart the network through the same exit gateway but the destination must always be able to properly reassemble the segments. It implies a suitable means for identifying the segments: sequence numbers, hierarchical subdivision, etc. Additionally, the header must be replicated and this overhead may be not negligible. A special technique is to be used to avoid an indefinite waiting of a corrupted segment.

We will see in the following chapter in which cases the segmentation/reassembly problem is relevant and which solution is chosen.

Chapter 2 : THE BRIDGE

Bridge commonly designates the interconnection of two or more Local Area Networks. After having explained differences between a bridge and a repeater, we give our definition and compare it with other authors' opinions. Bridge functions are illustrated by an example of a frame transferred by a bridge. Many architectural uses of bridges are possible, due to the pairwise or multi-network characteristic of the bridge. Finally, advantages and disadvantages of bridges are stated. The work carried on MAC bridges by ISO is resumed and explained at the end of this chapter.

2.1 Definition and Bridge functions

A clear distinction is to be made between a bridge and a repeater. A repeater -also called a bit repeater- is not a real interconnection technique. It simply amplifies or regenerates the physical signal on a link or bus-type network. It is often used to overcome distance limitations due to signal degradation on a coaxial cable. The repeater is only concerned with the Physical Layer of the OSI model. A repeater between two Ethernet coaxial segments retransmits the whole electrical signal, including collisions, noise, etc.

A bridge is a more intelligent technique. E. Benhamou and J. Estrin give a good comparison between a bridge and a repeater [BENH83]:

> A Bridge is different from a repeater in that the bridge is a store-and-forward device that receives full packets from networks, stores them, and then retransmits them to the other network. A (bit) repeater is not a store-and-forward device.

David Flint defines a bridge -that he calls a filter- as being the way of interconnecting several LANs on one site. It must select packets received from one LAN for retransmission on the other. This selection is based on the Datalink Address and four approaches are considered [FLINT83]:

1. The source indicates that the packet should be filtered;
2. The bridge knows the locations of individual stations and passes packets accordingly;
3. The bridge recognizes one part of the Datalink Address as identifying the destination LAN;
4. The bridge implements Network Layer routing functions.

All this address interpretation and routing implies some buffering capabilities in the bridge due to the high throughput of the LANs. But whatever provision is made for buffer storage, there is always some possibility of its overflowing. Flint considers that, in this case, the bridge may discard packets at random or according priorities. This loss of packets have to be reported to sources. But this possibility increases the complexity of the bridge and adds to the network load when that load is the greatest.

V.E. Cheong and R.A. Hirschheim see the bridge as a technique which allows the interconnection of many subnetworks of different technologies and transmission rates, but which uses identical software protocols and a single overall homogeneous address space [CHE083]. The bridge must know the location of every station of the interconnected subnetworks to do the selection of packets to transmit.

We agree with these approaches but our view is more "ISO-referred" and based on the service provided by the bridge. The bridge provides an interconnection between two networks using the same address space at the Datalink Layer. The bridge is not involved in any Network Layer protocol. For LANs, the Datalink Layer is divided into the Logical Link Control (LLC) and the Medium Access Control (MAC) Sublayers. A bridge provides the forwarding of MAC frames in a transparent way for the sender and the receiver (on two interconnected networks). It is thus called a MAC-Sublayer bridge because frame forwarding actions do not involve LLC protocols [ISO3751]. Figure 2.1 illustrates the operational model of a bridge. The transparency of the bridge implies no special features in the MAC-protocols to provide the internetworking. It also means that a bridge can be easily installed between two already existing LANs.

The given definition does not specify that a bridge interconnects two networks that have identical Physical Layers and MAC Sublayers. We may conceive of an Ethernet bus bridged with a token-passing ring, as they use the same address space. The ISO and IEEE work on MAC protocols is done towards this aim.

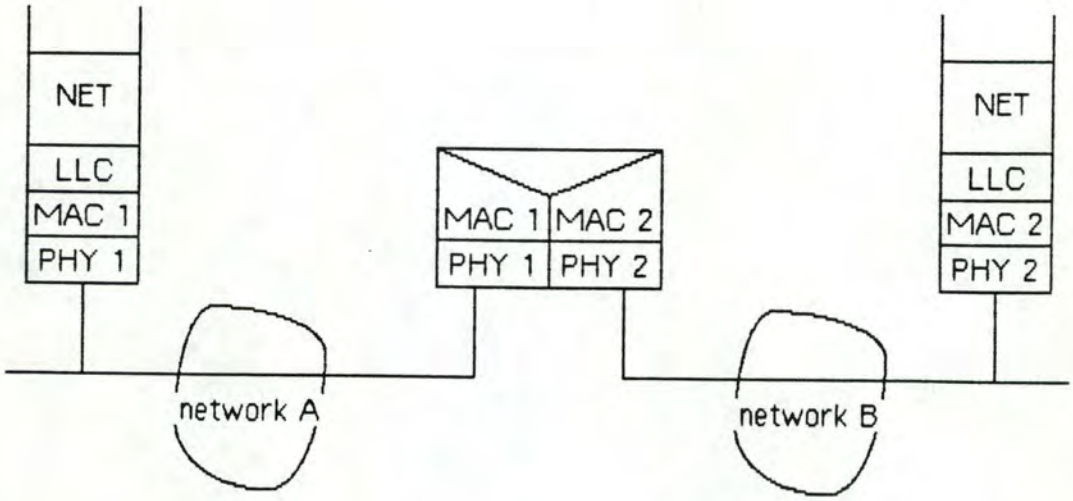


Figure 2.1: Operational model of a bridge

In the same idea, a bridge does not necessarily consists of one physical unit to which two or more networks are physically connected. It may use internally a point-to-point link or the services of another network to do the routing between interconnected networks. An example of such a configuration is developed in the second part of this thesis.

The functions of a bridge are the following [ISO3751] [STAL85]:

- filtering: The bridge must select the frames from one net which are addressed to another net. Because of transparency, the bridge must have a look-up table giving the location of every known address.

- switching and routing: When the bridge interconnects more than two networks, frames have to be routed to the appropriate network.

- buffering: Input buffering must deal with peak incoming traffic or with a full output buffer. An output frame may need to be buffered at the threshold of the destination network, waiting for an opportunity to be sent.

- interfacing: with the interconnected networks, depending on the medium and the medium access method. The interface must be able to receive all frames generated on the network. It implies that the physical interface operates in a promiscuous mode. This mode of operation is not the same as the one in which end-stations access to the medium. They "listen" to all frames but receive only these addressed to themselves.

A bridge interconnects networks at node level, due to its transparency for end-users - We have already explained the relevancy of the "node" notion in case of LANs (see 1.2). This node level interconnection makes the bridge fit into the classification of W. Stallings, as well as in our criteria of architectures definition.

2.2 Bridge operation : an example

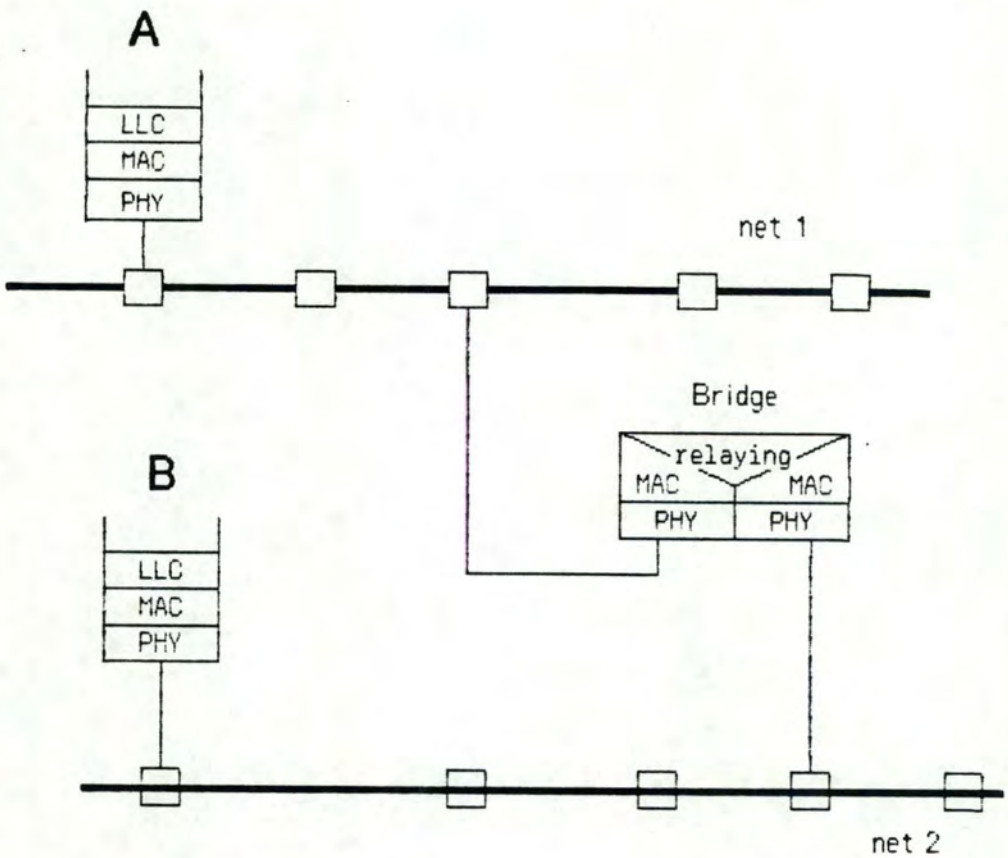


Figure 2.2: Two CSMA/CD bridged buses

Let us consider a CSMA/CD (IEEE 802.3) bus network bridged with another CSMA/CD bus network, as illustrated in figure 2.2. These two networks use the same 48-bit MAC Addresses. Let us suppose that station A wants to send a frame to station B.

The frame sent by A on the cable has the format illustrated in figure 2.3 [IEEE802.3]. The Destination Address field of this frame contains the MAC Address of station B. Once sent on the cable, this frame propagates to all the stations connected on the bus. The bridge thus receives this frame and, because it works in promiscuous mode, reads it.

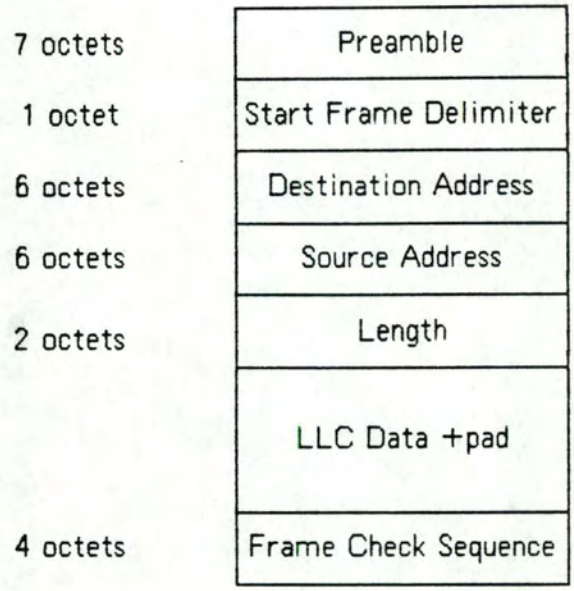


Figure 2.3: 802.3 MAC frame format

The bridge checks the Destination Address field and sees in its look-up tables that this address is located on network 2. Without changing anything in the format, it sends the frame on network 2.

If the destination network had been a token-passing ring (IEEE 802.5), some modifications would have been made to the frame format. The most important is the re-calculation of the Frame Check Sequence, because this check does not cover the same fields in 802.3 than in 802.5 standards. In such a case, the transparency of the service provided by the bridge is not complete because FCS errors generated on one network will not be transmitted on the other network.

2.3 Architectural utilizations of bridges

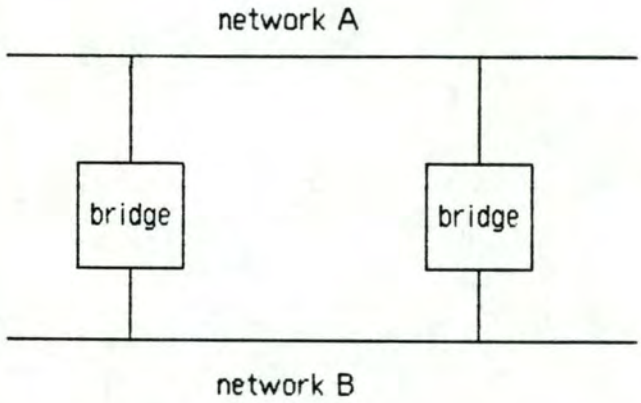


Figure 2.4: Two networks interconnected by two bridges

Bridges can be used in several ways to build a Bridged Area Network (BAN). The bridges characteristics that define the topology of the BAN are the following:

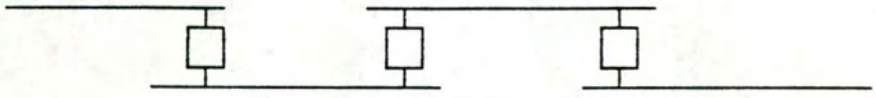
1. The number of networks that a bridge may interconnect; two (pairwise bridge) or more (multi-network bridge);

2. The fact that two networks may be interconnected by more than one bridge (see figure 2.4). Such a bridge has to be able to detect the presence of another identical bridge interconnecting the two same networks. If no special care is taken, any internetwork frame will be transmitted twice.

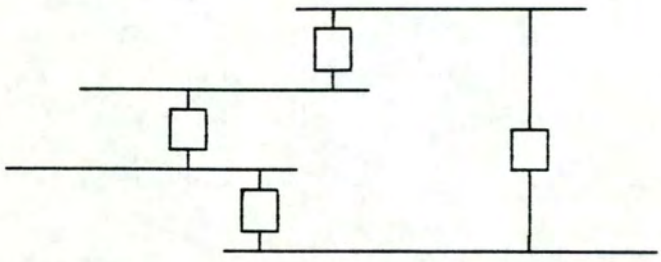
The bridge may also use internally communication links. This characteristic does not affect the architecture of the resulting BAN, but only the maximum area that the BAN will be able to span.

Figure 2.5 illustrates the topologies that a pairwise bridge can support (networks are drawn with a line for simplicity reason). These are the following:

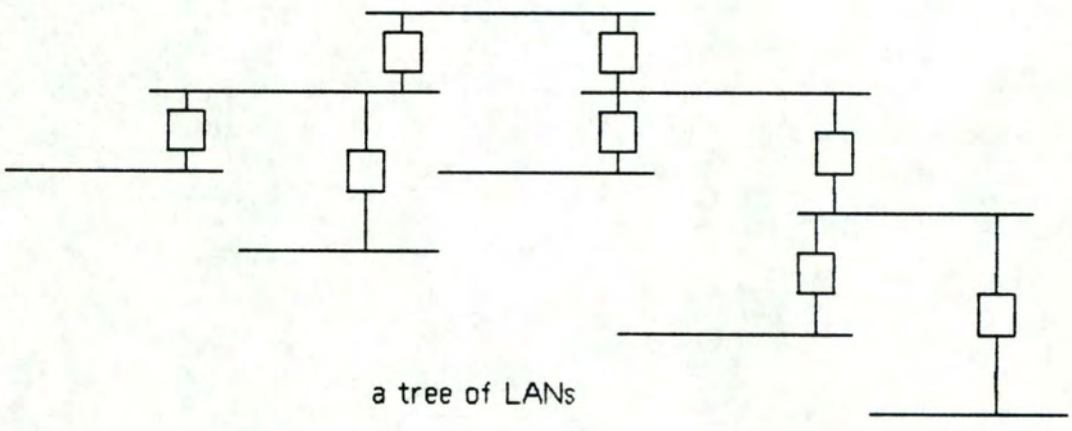
- The chain of LANs where the networks are connected end-to-end;
- The stair of LANs when they are interconnected part of the way along;
- A tree of LANs.



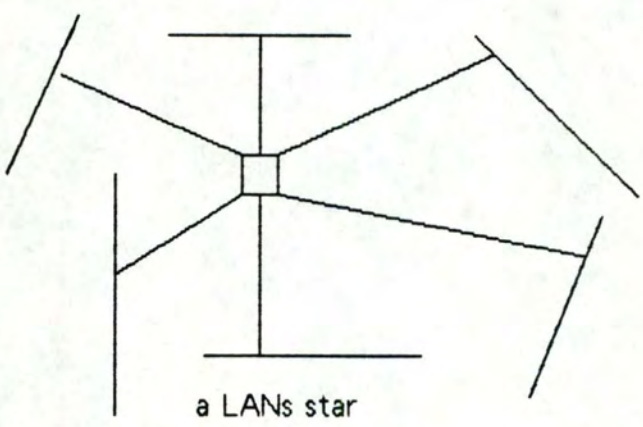
a chain of LANs



a stair of LANs



a tree of LANs



a LANs star

Figure 2.5: Many Bridged Area Network topologies

The multi-network bridge can be used in similar configurations to those outlined above, but with a greater multiplicity of networks at each bridge, leading, for example, to LANs stars. (figure 2.5). All these topologies can also be combined at will.

Depending on the topology, the bridge has to provide an additional function: the routing of the frames to the appropriate destination network. To do this, the bridge has to know the location of the users of every network of the BAN, including these that the bridge does not interconnect.

2.4 Advantages and disadvantages of bridges

The use of a set of bridged LANs rather than one LAN have the following advantages [STALL83], [PINEY86], [FLINT83]:

1. Reliability:

   The danger in connecting all data processing devices in an organization to one network is that a fault on the network disables all communications. Partitionning the network with bridges sets limits to the probability of loss of service for each user.

2. Performance:

   In general, performance on a LAN declines with the increase of the number of stations or of the length of the medium (i.e.: Ethernet). Some LANs have disadvantages linked to the increased distance (i.e.:increased latency on token ring). Additionally, networks users with a large amount of mutual intercommunication can be grouped on a separate LAN. In this way, users can be clustered so that intranetwork traffic significantly exceeds internetwork traffic.

3. Maintenance:

   Localisation of faulty elements is more easily performed on a network of limited size.

4. Economy:

   Use and type of LAN can be optimized by separating the problem of local network service from the overall geographic coverage. For example, if a broadband local network is to be installed on two buildings separated by a long distance, it may be easier to use two LANs interconnected by a bridge using a microwave point-to-point link. In the same idea, a more expensive and more reliable LAN may be installed in the area where traffic is most important.

5. Integration:

   It is possible between bridged LANs to use existing communications facilities. For example, in the implementation described in the second part, an already existing packet-switching network is used to provide routing between bridged LANs.

6. Adaptability:

   Inclusion of additional LANs in a BAN can be carried out with a minimum disruption to the existing service.

The main problems that can be encountered in a Bridged Area Network fit into five main classes [PINEY86]:

- Increased delay:

  The bridges impose a store-and-forward delay.

- Congestion:

  Congestion in the bridge can lead to several symptoms such as lost frames, irregular traffic, etc.

- Increased security risks:

  The transparency of the MAC Sublayer bridge service may allow remote access to unauthorized LANs.

- Degradation of some service characterisics:

  The overall service relies on the individual services available along the traversed path. This can lead, for example, to reductions in the maximum frame length compared to that available on each of the LANs.

- Management complexity:

  Analysis of performance and diagnosis of end-to-end problems become even more complicated than in a single network.

2.5 ISO work on MAC Sublayer bridges

ISO and IEEE work on 802 Project aims at allowing an easy interconnection of Local Area Networks using any 802 MAC Protocol. The primary condition is the use of a common address format, which is the case in all 802 standards.

A Bridged Area Network has the following characteristic features [ISO3751]:

- The address space by which the MAC stations are identified must encompass the entire BAN. Two stations within the BAN can not use the same MAC Address.
- Each individual LAN has its own independent MAC. For example, the token will never leave its bus or its ring to circulate on any other LAN.
- The independence of MACs also dictates that no control frames that govern the operation of an individual LAN are forwarded off the LAN of their origin.
- The bridge may have to calculate a new Frame Check Sequence, because the fields covered by the FCS may change from one MAC Protocol to another. Therefore the service provided by the FCS in a BAN is different from the one of a single LAN.

These two last characteristics prevent the bridge service from being fully transparent.

As for any other layer, ISO has defined the service provided by the MAC Sublayer, without reference to the specific protocol used to provide this service (802.2, 3, 4, 5). They have begun to study the implications of this service definition on a MAC bridge. The following list represents the main elements used to define the MAC Sublayer service. For each of these, the implications on MAC bridge architecture are given [ISO3751]:

1. The service provided by the MAC Sublayer of frame relay is independent of the topology.
   Implications :
   - Frames transmitted between end stations carry the MAC Address of the end station in their Destination Address Field, not the MAC Address of the bridge, if any.
   - The architecture should allow for the interconnection of all IEEE 802 LAN types, in any combination.
   - The architecture should impose as little overhead as possible on the end-stations.

2. The MAC/LLC interface used the MA_DATA primitives (request, indication, confirm) and the MA_STATUS primitives (request, indication).
   Implications :
   - The architecture should not require modifications to the MAC/LLC interface.
   - The architecture should introduce as few new management burdens as possible.
   - The user should have as much freedom as possible in the choice of MAC station address types (local, global, hierarchical, flat) with the restriction that all MAC Addresses must be unique in the BAN.

3. The MAC service exhibits a negligible rate of residual bit-errors.
   Implication:
   - Bridges should not cause the end stations to experience residual bit-errors significantly higher than those which it would normally experience on a single LAN.

4. The MAC Service exhibits a negligible rate of misordered frames.
   Implication:
   - Bridges should not permute the frame order as they forward frames except, perhaps, under failure conditions.

5. The MAC Service might exhibit a non-negligible rate of frame duplication.
   Implications:
   - A bridge should not permit the generation of a duplicate frame.
   - When redundant bridges exist between pairs of end stations and are active (causing multiple possible routes), the probability of duplicate frame generation by the MAC Service should be kept as low as possible.

6. A maximum MAC Service Data Unit size shall be no less than 263 bytes.
   Implication:
   - Since MAC bridges do not perform segmenting even when LANs with differing maximum frame size are mixed in a BAN, care must be taken to ensure that the MAC Protocol Data Unit emitted from end stations can be accomodated by all LANs in the BAN.

7. The delay provided by the MAC Service should be limited to a reasonable value (five seconds) with the probability of the delay exceeding that limit sufficiently small.
   Implication:
   - The MAC Service delay should not be limited by the use of bridges.

Chapter 3 : THE GATEWAY

This chapter discusses an interconnection technique operating at Network Layer and at node level. The definition and the functions of this technique are explained and then illustrated by the X75 Recommendation voted by the CCITT and allowing the interconnection of two X25 networks.

3.1 Definition and gateway functions

Let us first note that the Gateway term used in this chapter designates an interconnection technique and is not a generic term designating any physical device interconnecting two networks.

As introduced at the end of the section 1.5, the Gateway is an interconnection technique operating at the Network Layer and at node level. The two networks may have identical or different Network Layer Protocols. We have already explained the interconnection of two X25 networks with a joint node (see 1.3.). In this case, internetwork packets are routed by the communication subnetwork to the gateway. The existence of two or more networks is thus invisible for the two communicating Network Layers, if the two networks are identical. If there are not, the address structure must allow the Network Protocol to specify the name or address of the destination network. This can be done, for example, by a hierarchical address structure. It is the only way by which the Network Protocol "sees" that it is sending an internetwork packet, but it does not mean that internetwork packets are addressed to the gateway by the Network Layer. This is the reason why we say that the gateway is invisible for communicating Network Layers. It is the task of the communication subnetwork to forward the packet just to the gateway.

Figure 3.1 shows the functional scheme of a gateway. The gateway must provide functions of mapping between two Network Protocols. These functions are very simple when the networks are identical (e.g., two X25 networks). In this case, the main function is to provide the mapping between virtual circuit numbers and the flow control between the two communication subnetworks. If the two Network Layer Protocols have only low importance differences, such as packet format, sequence number mechanisms or resetting conditions, the work of the gateway is more complicated because more differences are to be mapped. But when services provided by the Network Layers of the two interconnected networks are really incompatible (e.g., connection-oriented vs connectionless transmission), the task of mapping these differences at node level is too difficult for the Gateway to be an efficient interconnection technique. The

Gateway would have to map a virtual circuit service to a datagram service, in an invisible way for the two communicating Network Layers. It is more efficient to interconnect two such networks at host level, using the interconnection technique described in the following chapter (Internetwork Protocol).

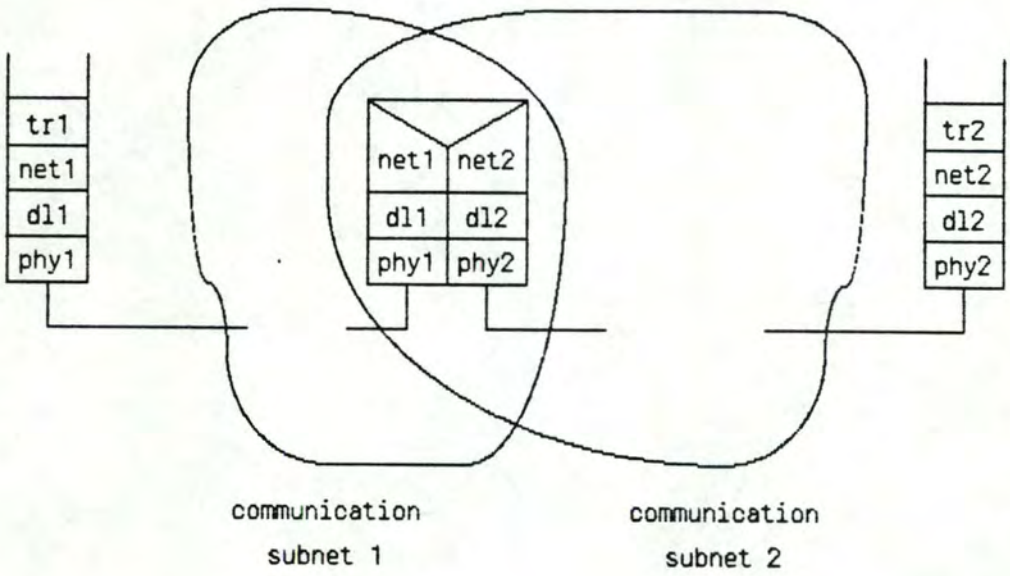


Figure 3.1: Functional scheme of a gateway

The physical architecture of a gateway may consist of a node being part of the two communication subnetworks, as illustrated in figure 3.1, or of two half-gateways tied together by one or more communication links. A protocol is thus necessary to manage this link. The Recommendation X75 voted by CCITT in 1984 is a well-known example of a protocol designed for transmission between two half-gateways interconnecting two X25 public data networks. This protocol is studied further on in this chapter.

3.2 The X75 Principle and Operation

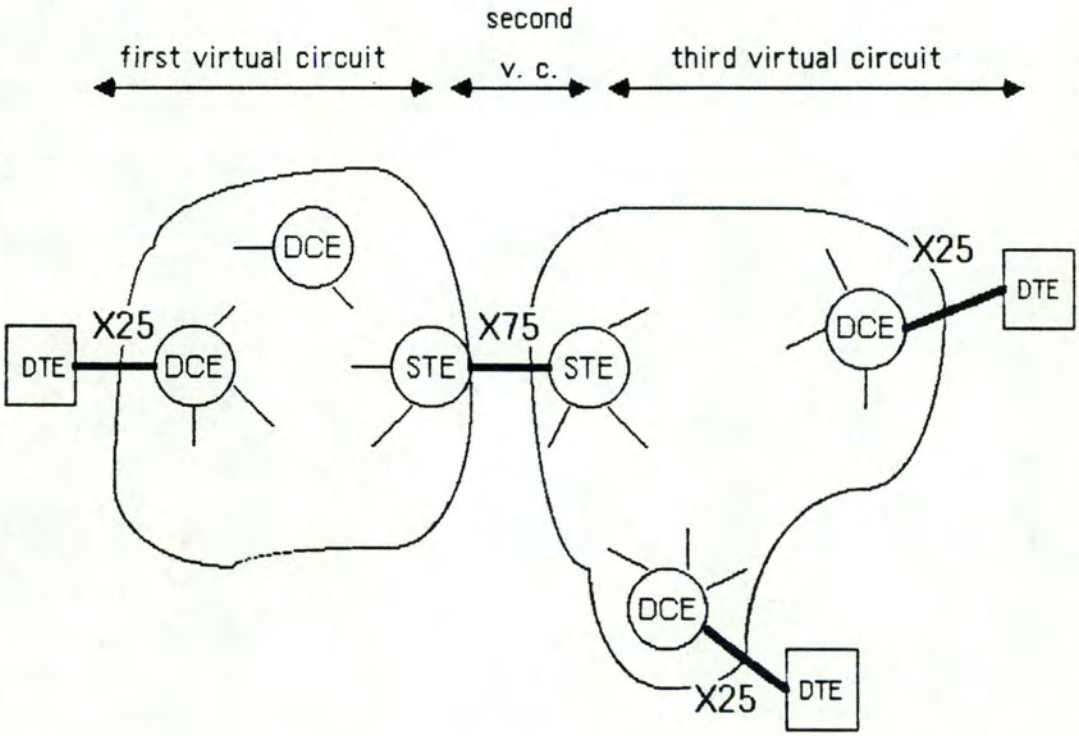


Figure 3.2: Two X25 networks interconnected by a X75 gateway

Let us first note that X25 specifies the interface between host equipment (called DTE) and network equipment (called DCE) that encompasses layers 1 through 3 and permits the set up, maintenance and termination of virtual circuits between two DTEs. The X75 Recommendation specifies a Signal Terminating Equipment (STE) that acts as DCE-level gateway between two X25 networks, as illustrated in figure 3.2. The STE is thus a special node (or a part of a node) being part of the communication subnetwork. The two STEs and the link between them make up the Gateway as defined above. This gateway interconnects the two networks in plugging together two X25 virtual circuits in an invisible way for host's Network Layers. These Layers only "see" a greater number of users, a greater number of Network Addresses, an enlarged X25 network.

Figure 3.3 shows the functional model of a X75 gateway. Each of the communicating hosts dialogs at the Network Layer on a X25 virtual circuit, without knowing the existence of the X75 gateway. The DTE-DTE virtual circuit service provided is in fact made of a series of virtual circuits:

1. From the source DTE to the STE of the same network;

2. From this STE to another STE (one or more);

3. From the last STE to the destination DTE.

Each of these sections is a distinct entity with a separate virtual circuit, flow control and error control. There can be, of course, more than one X75 gateway in the resulting virtual circuit.

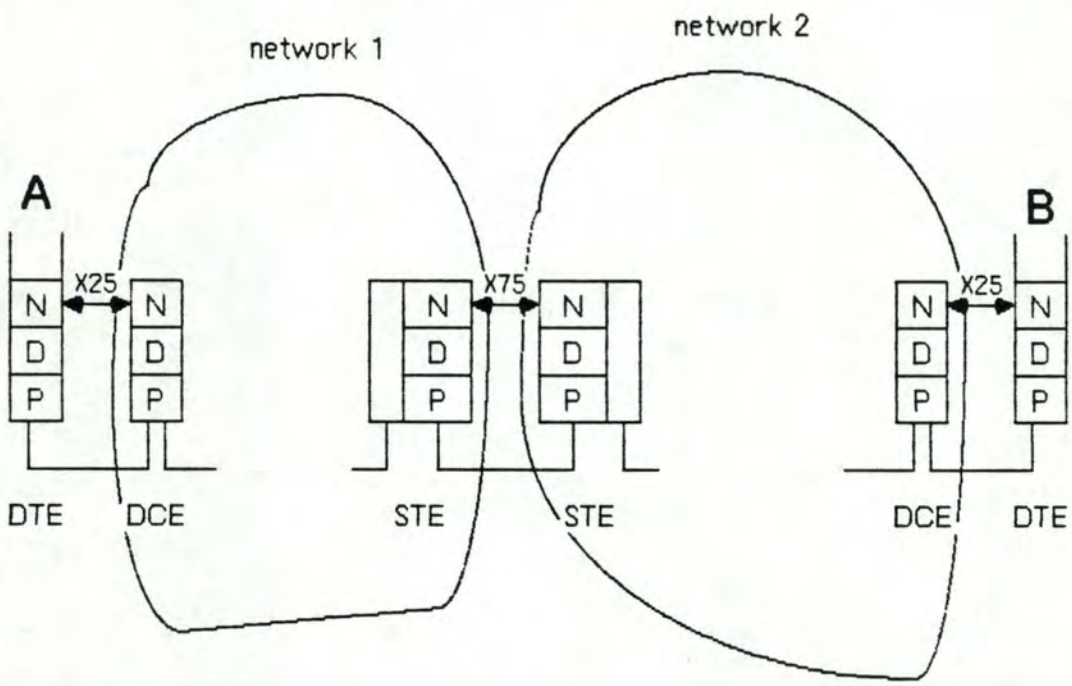


Figure 3.3: Functional model of an X75 gateway

The best way to understand how X75 operates is to explain how the Call Request Packet propagates through the networks. It is handled step by step but must propagates end-to-end. The Call Request Packet from A triggers the set up of a DTE-STE virtual circuit with its own virtual circuit number. The STE, using the Call Request Packet of the X75 Protocol (which is nearly identical to its X25 counterpart) sets up a STE-STE virtual circuit between network 1 and 2. The number of this last virtual

circuit may be different from the preceding one. The Call Request Packet then propagates to B's DTE, setting up a third virtual circuit with its own virtual circuit number. Finally, a Call Indication Packet is delivered to B by its DCE. The Call Accepted Packet follows the same path in the reverse way.

The main difference between X25 and X75 is the addition of a network-level utilities field in X75 packets. All the other fields are identical. This has two reasons [STAL85a]:

- There is no encapsulation of X25 packets by the STEs. The same headers are used to send information as well on X25 connection than on the X75 virtual circuit. Field values may differ (i.e. the virtual circuit number) but the format does no differ.

- Whereas a packet has the same format across multiple virtual circuits, all information (virtual circuit number, flow control, sequence numbers, ...) has local significance only. X75 is not an end-to-end protocol.

For example, a Data Packet sent by A to its DCE has the virtual circuit number that A associates with a connection to B. The network 1 transmits this packet to the STE. The STE uses the same format, but modifies the virtual circuit number (and other control information, if necessary) for the STE-STE virtual circuit. The receiving STE sends the packet to B's DCE with the virtual circuit number that B associates with a connection to A [STAL85a]. All these modifications of virtual circuit numbers are invisible for A and B.

If many paths exist between the source and destination hosts, one path is chosen by the communication subnetwork at the opening of the virtual circuit. Every packet of this connection will follow this path, whatever loading or congestion of this path. This fixed routing is a disadvantage but results from the service to provide end-to-end: a virtual circuit.

The X75 Recommendation is concerned with Physical, Datalink and Network Layers, as illustrated in figure 3.3. The Physical and Datalink Layers are necessary to manage the one or more links between two STEs. The Network Layer is an intermediate between the Network Layers of the two interconnected networks.

3.3 The Datalink Layer of X75

The Recommendation defines two procedures according to the number of links that tie the STEs. The Single Link Procedure (SLP) is identical to the Datalink Layer of X25 (LAP-B) and is to be used when one physical link exists between the STEs. The Multilink Procedure (MLP) allows the interface to operate over multiple lines [CCITTX75].

The principle of MLP is simple: "When multiple links exist, each link is governed by the SLP LAP-B" [STAL85a]. The set of links is used as a pooled resource for transmissing frames regardless of virtual circuit number of the packet included in the frame. Any available link may be chosen for transmission of any frame. Once a MLP frame is constructed, it is assigned to a particular link and further encapsulated in a SLP frame. To keep track of frames, a unique multilink sequence number across all links is used. It is necessary to allow the receiving MLP to reorder the frames transmitted on different links and to detect repeated transmissions of a frame.

3.4 The Network Layer of X75

The Network Layer of X75 is almost identical to that of X25. The only differences are those needed to accommodate internetwork administration and management functions of the STE. These include, for example, additional facilities information in the Call Request Packet. When transferring data, one may choose an acknowledgment with local or end-to-end significance. Resetting and restarting procedures are identical to those of X25.

Chapter 4 : THE INTERNETWORK PROTOCOL

The Internetwork Protocol is undoubtedly the best known network interconnection technique and the most illustrated in the literature. In this chapter, we first describe the principle of the Internetwork Protocol, more briefly named IN. To make a distinction with the gateway technique of the preceding chapter, we call the device interfacing between two networks an IN Gateway, whereas all authors call it a gateway. An example of the IN operation will allow us to define the functions of the Internetwork Protocol and the functions of an IN Gateway. Two examples of Internetwork Protocol are developed: The ISO International Standard 8473 and the well-known TCP/IP of the United States Defense Advanced Research Projects Agency (DARPA). Advantages and disadvantages of IN are conclusion of this chapter.

4.1 Definition

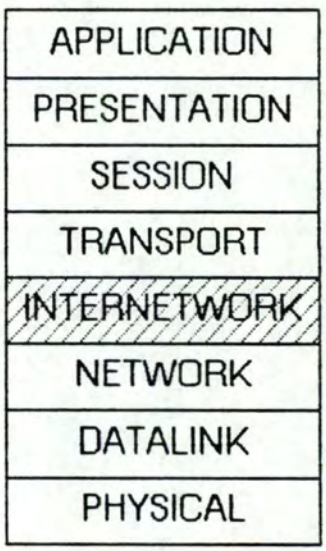


Figure 4.1: The place of the internetwork function

As explained in the introductory example, the Internetwork Protocol technique is based on a very simple principle: To allow intercommunication between networks, a specific protocol is devoted to the task of internetworking. In the OSI Model, this additional layer is logically located between the Network Layer, which is technology and topology dependent, and the Transport Layer, in charge of providing a full reliable transmission service to upper layers. Figure 4.1 illustrates the place of the Internetwork function.

This technique allows the interconnection of networks that have significantly different internal protocols and performance (and, of course, networks which are identical). The only condition for the use of an Internetwork Protocol is that each interconnected network provides at Network Layer a service which allows IN to operate. This service must be defined before specifying an Internetwork Protocol. Theoretically, this service could be full reliable connection oriented service (e.g., X25) as well as a very simple connectionless transmission service. In practice, the easiest way to allow the interconnection of very different networks is to require the minimal level service, i.e. the connectionless or datagram service. The IN Layer does not improve this minimal service level, because it is only concerned with the forwarding of packets between networks. Higher layers are supposed to provide a full reliable service to end-users.

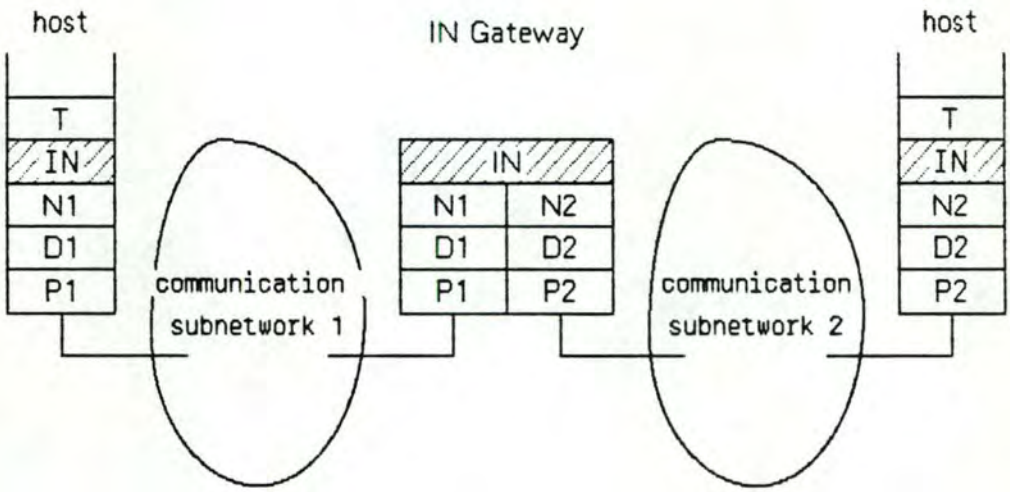


Figure 4.2: Operational Model of IN

Figure 4.2 is the operational model of IN. It shows that the IN Layer resides in the hosts as well as in the IN Gateway. This is an important difference from the previous interconnection techniques where no hosts protocol is directly concerned with internetworking. It means that each host of each interconnected network has to implement the Internetwork Protocol. This is a severe difficulty to the use of an IN on many already existing networks! This also implies an Internetwork Layer addressing. Whatever addressing scheme is used, IN Gateways must be able to route any internetwork packet to the correct destination network. This condition pleads in favour of a hierarchical addressing scheme: an internetwork address is made of a network address and a host address.

Another difference with the gateway of the previous chapter is that networks are interconnected at host level by the IN Gateway. This results from the fact that the IN Layer uses the services of the Network Layer to send packets to a gateway or to a destination host. An internetwork packet generated at source IN Layer is addressed to the destination host but sent by the source Network Layer to the IN Gateway. IN Gateways are thus addressed at Network Layer, but not at Internetwork Layer.

4.2 Description of IN operation

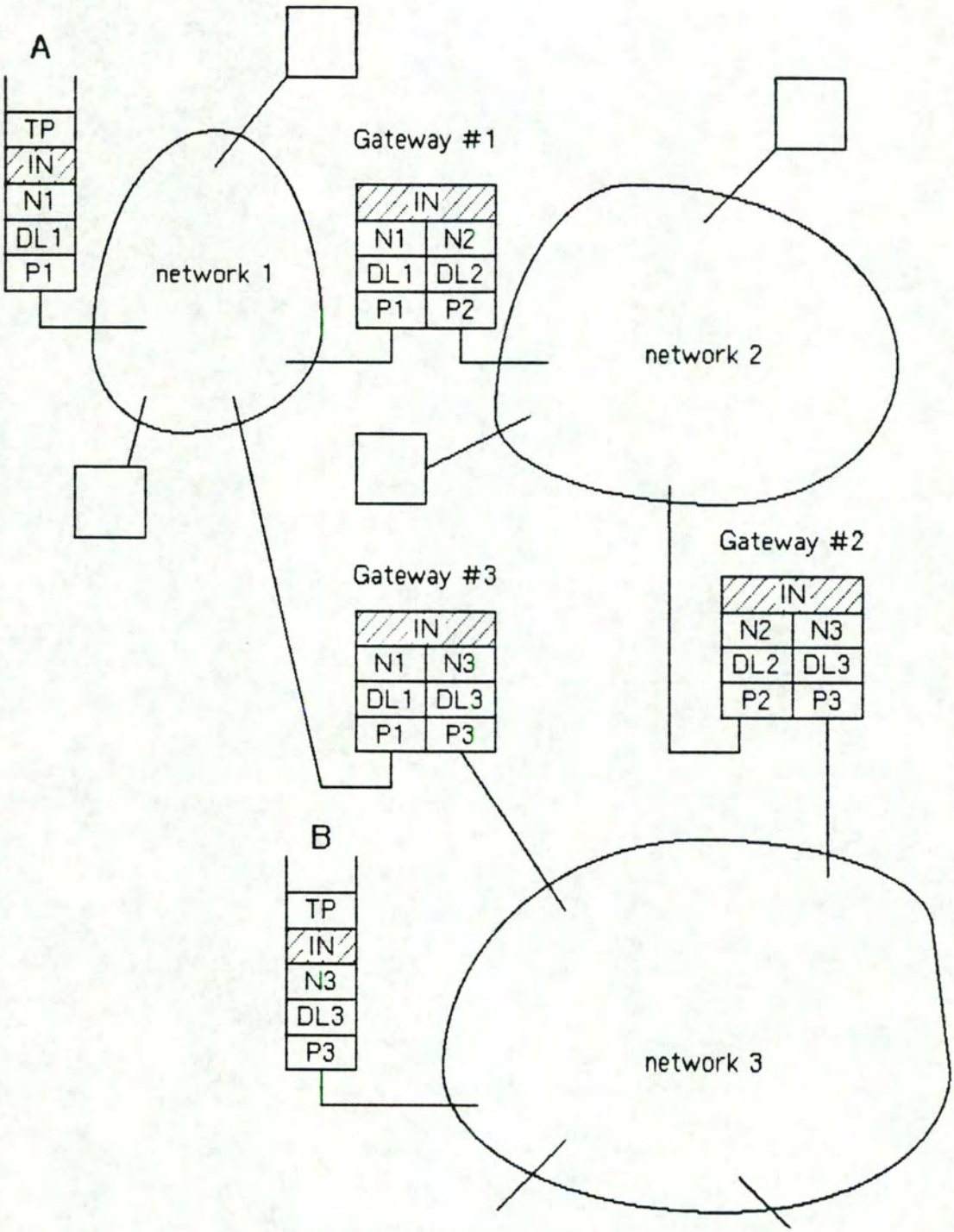


Figure 4.3: Example of IN operation

Let us suppose that the host A on network 1 wants to communicate with host B on network 3 (see figure 4.3). The A's Transport Protocol asks to A's IN Layer to send a stream of bits to B's Transport Protocol whom it gives the Internetwork Address. This TP stream of bits is called TP packet in the figure 4.4 illustrating the data manipulation in an IN operation. This data must pass through many networks environments to arrive to B. To do this, the data is encapsulated in an Internet Packet, whom header contains the IN destination address and any other control information. The IN Protocol asks to the Network Layer to send this IN Packet to an IN Gateway whom it gives the Network Address. For the Network Layer, the IN Gateway is a station exactly identical to any other hosts connected to the network. The IN Layer may choose any IN Gateway it wants. In our example, IN Gateways #1 and #3 are possible. The #3 is certainly on the shortest path but maybe not on the quickest path. This adaptive routing characteristic supposes that IN is based on a connectionless datagram service.

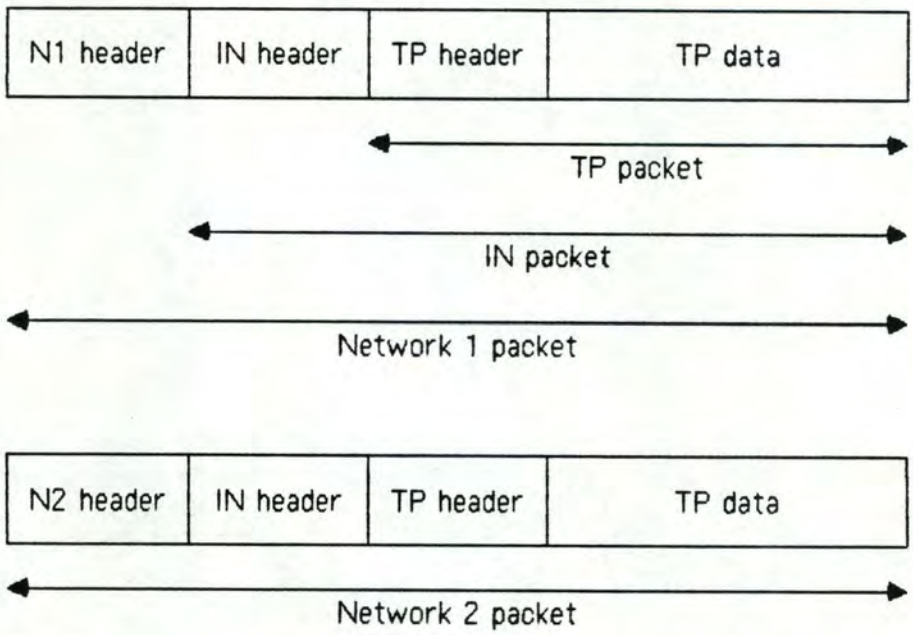


Figure 4.4: Data encapsulation in an IN operation

The Network Protocol encapsulates the IN Packet in a Network Packet specific to Network 1 Protocol (see figure 4.4). This packet crosses Network 1 and arrives to IN Gateway #1, for example. The IN Layer of the gateway receives the IN Packet, reads the destination address, takes a routing decision and sends the IN Packet to a following IN Gateway. This is thus encapsulated again in a Network 2 Packet.

This process goes until the IN Packet arrives at the IN Layer of the destination host, which delivers TP Packet to B's Transport Protocol. Thus, the model is "a series of encapsulation/extractions, not translations" [POST81].

In the case of source and destination hosts on the same network, the IN Protocol sends the IN Packets directly to the IN destination address. Thus, the IN Layer is redundant with the Network Layer in case of intra-network communication. But this is not a significant processing overhead when the Network and Internetwork services are connectionless.

The segmentation problem may occur when the IN interconnection technique is used between networks of different maximum packet sizes. This problem has already been studied in chapter 1.

4.3 Internetwork Layer functions

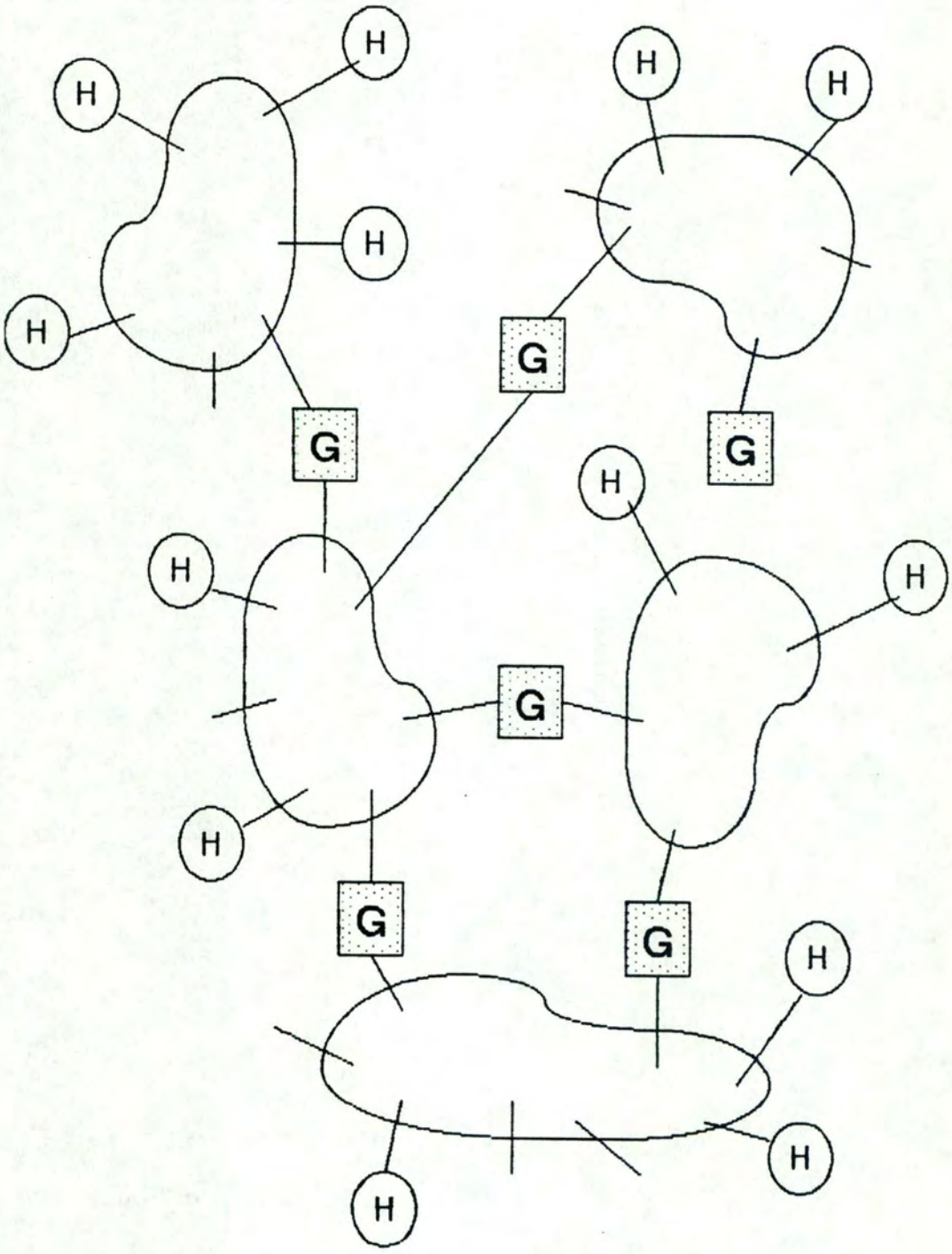


Figure 4.5: Connectivity of a set of interconnected networks

The main function of the Internetwork Layer is the forwarding and routing of IN Packets accross multiple networks. To accomplish this, the IN Protocol must reside in each host engaged in the internetwork communication and in each gateway. When considering a set of networks interconnected with IN Gateways, these appear as store-and-forward devices between networks, exactly in the same way as nodes are store-and-forward devices in a communication subnetwork. Figure 4.5 is not different from the one of a communication subnetwork where point-to-point links between nodes are replaced by networks between IN Gateways. The forwarding function of IN Layer is identically implemented in host IN modules than in IN Gateways. The question to solve can be stated as: To send this IN Packet towards its destination, to which gateway or host of my network do I have to forward it? The answer to this question is to be found using routing tables which may be different in hosts than in IN Gateways because IN Gateways have direct access to the two networks. The adaptability of routing depends on the service provided by the IN Layer: adaptive routing with datagram service and fixed routing with connection-oriented service.

The second function of the Internetwork Layer is the segmentation and reassembly of large packets when needed to cross networks with small packet size limits. The main characteristic of this function is the technique used to identify and to reorder IN Packets resulting from segmentation. This function has to be implemented in each IP module if inter-network segmentation is used. In this case, a packet may be segmented at the entrance IN Gateway of a network and segments will be reassembled at the destination host, which must be able to do it in its IN Layer. In the case of intra-network fragmentation, the fragmentation/reassembly function is also needed in any host because each host is capable of receiving a packet segmented at the entrance of its network.

Additionally to these two main functions, IN Protocol has to face with many other problems. One of these is to avoid that an IN Packet turns around in a set of networks an indefinitively long time, which may occur in case of routing adaptative to the load of networks. Others are more simply these of any Network Protocol, such as flow control and error control.

4.4 The Internetwork Protocol of ISO

The ISO work on Internetwork Protocol is based on the definition of the Network Layer Service. This service must deal with all considerations concerning the topology and the technology of networks effectively used. Christine Ware says in [WAR83] that:

> "The purpose of the OSI Network Layer is to provide an end-to-end communication capability to the Transport Layer Entities above it. The Reference Model specifies that this communication capability is independent of any operational characteristics of the specific "real-world" transmission facilities which underlies it. The Network Layer, and the layers below it, perform the functions required to deal with "real networks" and utilize them to provide a networking capability in the OSI context".

This principle means that, if an Internetwork Protocol technique is used to interconnect many networks, this IN Protocol is not defined as an additional layer in ISO terminology. This IN Protocol is thus included in the Network Layer. It does not change anything to IN principles and functions explained above. This inclusion of IN notion in the Network Layer is necessary to respect the service principle on which ISO Layering Model is based.

Throughout this section, there is a distinction between the use of the term "network" and the term "subnetwork". The term "subnetwork" is used to emphasize that a particular network is part of an interconnected set of networks, whereas the term "network" is used to emphasize the individual network as a separate entity. Thus a "network" may also be a "subnetwork" if it is interconnected to other networks.

4.4.1 The internal architecture of the OSI Network Layer

A description of the internal architecture of the Network Layer helps to understand how different types of communications facilities are used and interconnected in the ISO Network Layer.

Figure 4.6 shows the three sublayers of the Network Layer and the functions associated to each sublayer [CALL83]. The highest functional grouping, SNIC, contains the internetwork relay and routing functions, and may contains other functions necessary to effect network interconnection. The middle of the three sublayers, the SNDC Sublayer, contains those functions, if

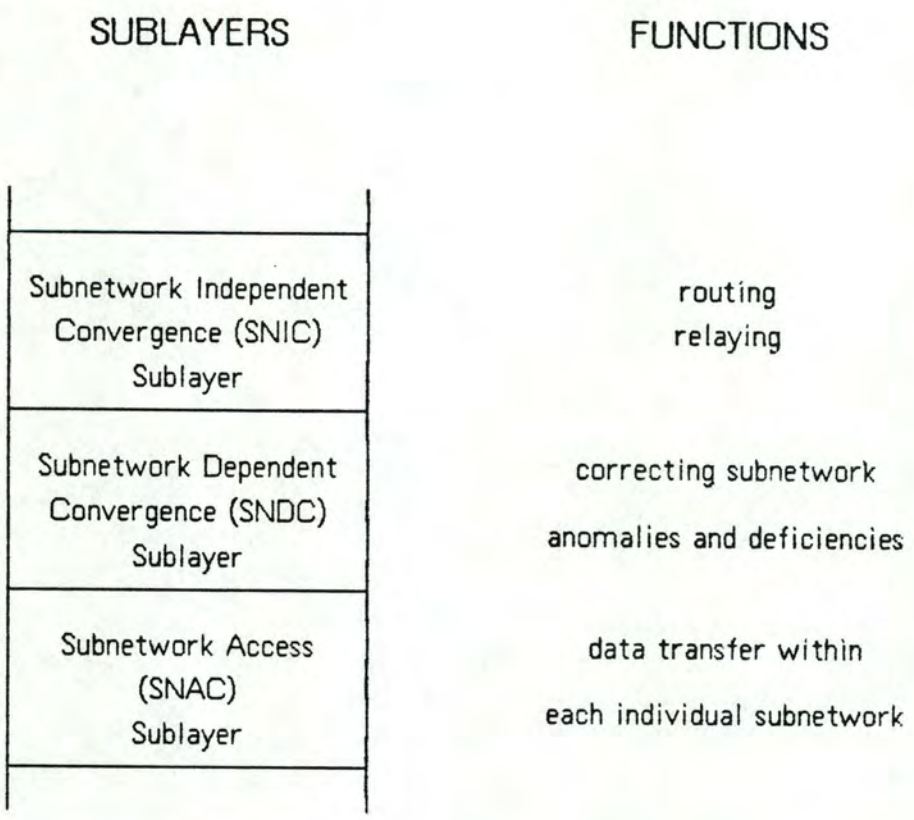


Figure 4.6: ISO Network Layer internal organization

any, necessary to bring the services offered by each individual subnetwork up to the common level necessary for interconnection. The lowest functions group, SNAC, contains those Network Layer functions specific to each individual subnetwork.

The SNIC Sublayer is the Internetwork Layer in the sense defined at the beginning of this chapter. The SNDC Sublayer transforms the subnetwork service into the uniform service which is expected by the SNIC Sublayer. To do this, the SNDC Sublayer may supply additional capabilities or may actually abrogate the functionality that the subnetwork provides. The SNAC Sublayer is what we previously called the Network Layer, i.e. a connection oriented or connectionless protocol. Figure 4.7 illustrates the OSI layering model of two subnetworks interconnected by an IN Gateway. The Internetwork Protocol (SNIC) resides in each host and in the IN Gateway, that ISO calls an Intermediate System. The SNIC Protocol is unique for the whole set of interconnected subnetworks, whereas SNAC Protocols may be different, depending on the service provided by each subnetwork. On the top of this architecture, a uniform service is provided to Transport Layers.

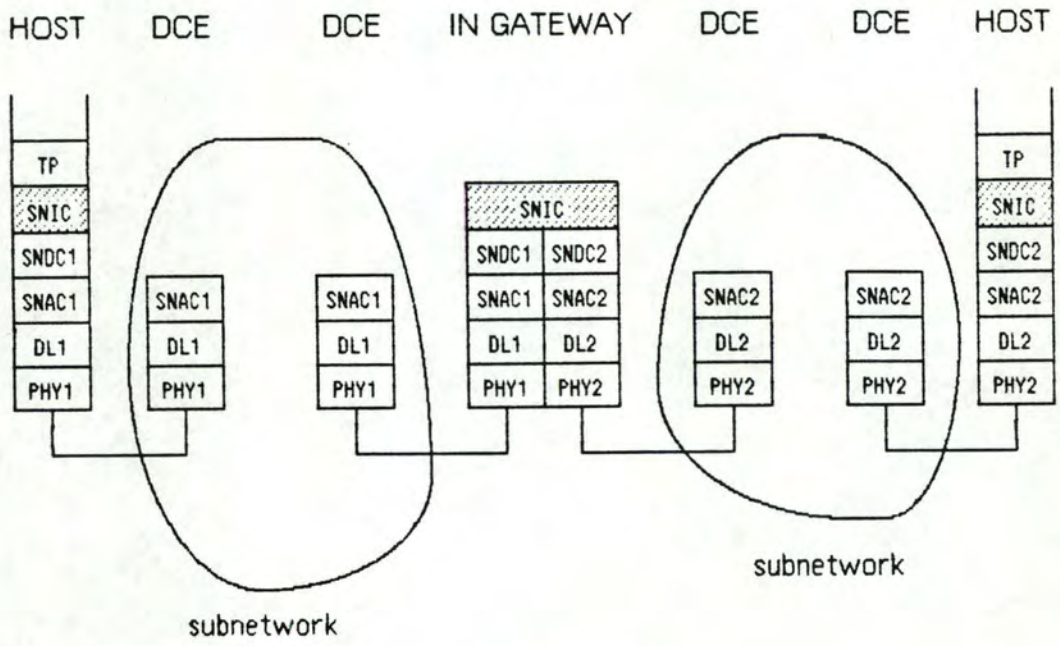


Figure 4.7: OSI layering model of internetworking

4.4.2 The ISO/IS 8473 Standard

The ISO International Standard 8473 is a Subnetwork Independent Convergence (SNIC) Protocol. This protocol provides the Connectionless-Mode Network Service. It relies upon the provision of a connectionless-mode subnetwork service [ISO8473]. In other words, this standard defines a Datagram Internetwork Protocol based on the assumption that each subnetwork provides at least a datagram service.

Figure 4.8 shows the format of a Protocol Data Unit (PDU) defined by the Standard 8473. It is made of four parts. In the Fixed Part, the Lifetime Field represents the remaining lifetime of the PDU, in units of 500 milliseconds. It is decremented by every entity which processes the PDU. The PDU will be discarded if the value of the field reaches zero. The flags in the fifth octet are:

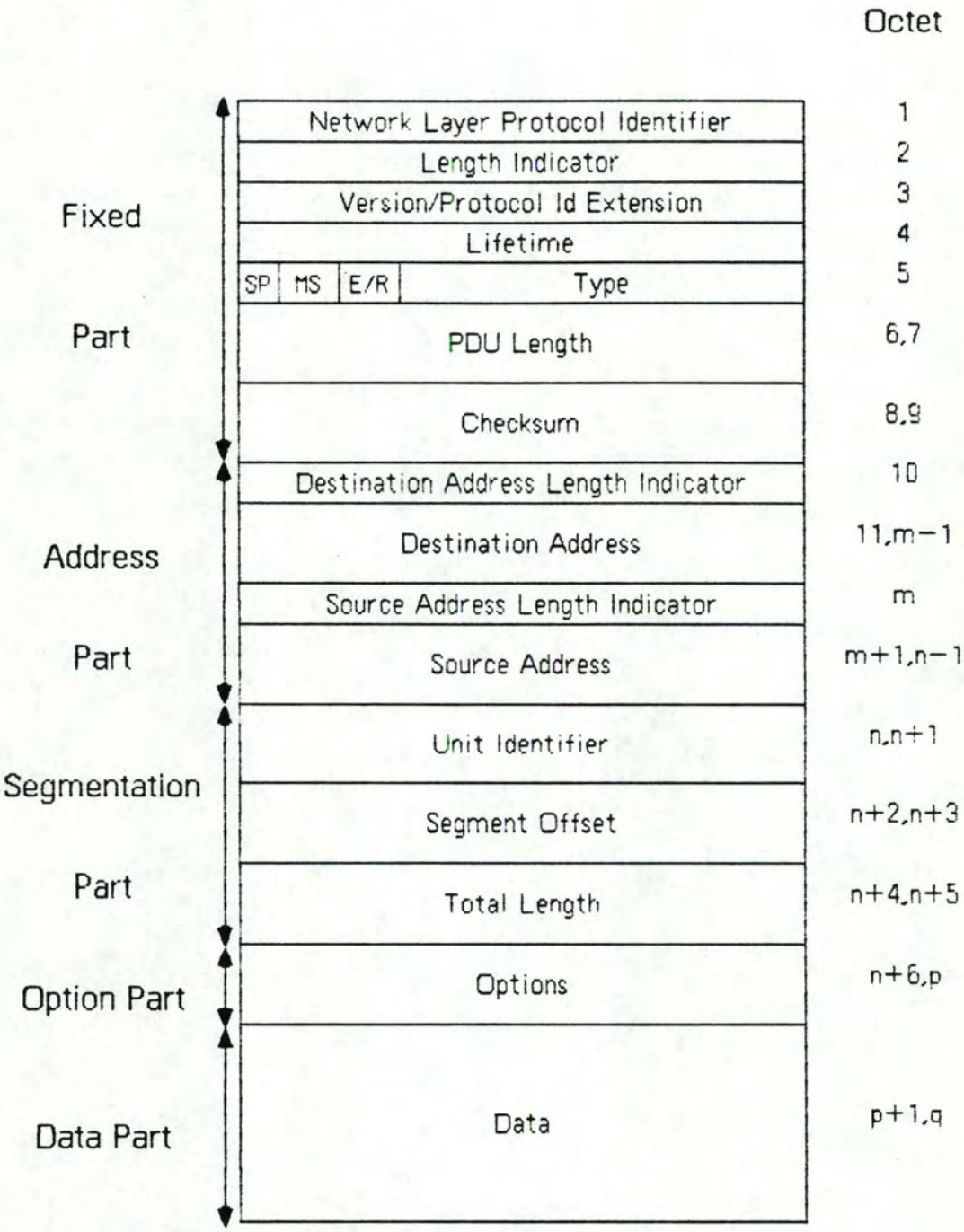


Figure 4.8: The Protocol Data Unit format in ISO/IS 8473

- **Segmentation Permitted Flag;**
- **More PDUs Flag;**
- **Error Report Flag.**

The Error Report Flag determines whether to generate an Error Report PDU upon discard of the PDU. The Type Code Field identifies the type of the Protocol Data Unit: Data PDU or Error PDU.

The Segmentation Part contains information necessary for the segmentation/reassembly function. The Standard specifies that reassembly takes place at the destination, but other reassembly schemes are not precluded.

The Option Part is used to convey optional parameters. Each parameter is specified by a Parameter Code (1 octet), a Parameter Length (1 octet) and a Parameter Value. The following parameters are permitted:

- Padding (to lengthen the PDU to a convenient size),
- Security,
- Source Routing,
- Recording of Route,
- Quality of Service (requested by the originator).

4.5 The Internetwork Protocol of DARPA

Since the development of the ARPANET in the early 1970's, a variety of new networks have been developed under DARPA (Defense Advanced Research Projects Agency) sponsorship, including satellite, packet radio and local networks. In order to allow users on different networks to communicate with each others, a means for interconnecting networks has been developed. The method chosen by DARPA is the Internetwork Protocol technique. An unreliable Internetwork Protocol called IP has been developed and on top of this the Tranport Control Protocol (TCP) provides a reliable communication service.

In the DARPA terminology, a set of interconnected networks is called a Catenet.

4.5.1 The DARPA Architecture Model

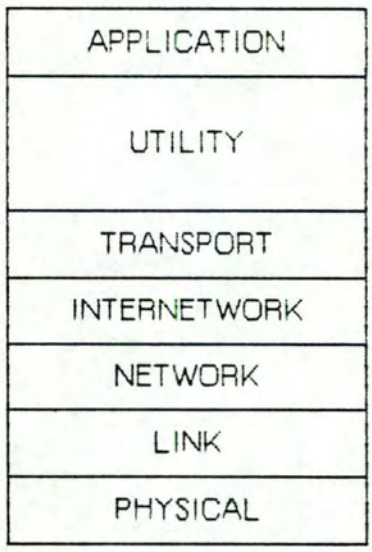


Figure 4.9: The DARPA Layering Model

As shown in figure 4.9, the layering model developed by the DARPA is different from the OSI Reference Model. An Internetwork Layer is explicitely defined between Transport and Network Layers. At the higher layers, protocols accomplishing the

functions of OSI Session and Presentation Layers are combined into a single "Utility Layer".

Beyond the layering principle, which is used by ISO and DARPA, Vinton Cerf and Edward Cain underline a most fundamental difference between ISO and DARPA points of view [CERF83]:

> "There is often an implicit assumption that one can easily substitute one protocol for another in a particular layer without affecting the functionality of the protocols which depend on it. This assumption (or goal) is unwarranted, although it seemingly makes life easier for the protocol architecture designer. The problem lies in the nature of the functionality of the protocols in a particular layer and the nature of the services they can easily offer. (...)
> It is the view of the authors that the goal of total interchangeability of layer N protocols is unnecessary."

This consideration leads to the notion of protocol hierarchy, which rapidly appears when one tries to classify the protocols developed by DARPA (and by the Departement of Defense from which DARPA is dependent). Figure 4.10 shows the hierarchy of the DoD protocols [CERF83].

All these differences between ISO and DoD networking arise from the fact that for the DoD multiple networks of widely differing internal characteristics will be a natural and necessary part of military networking.

4.5.2 The Internet Protocol

The Internet Protocol is specifically limited in scope to provide the functions necessary to deliver a package of bits (an Internet Datagram) from a source to a destination over an interconnected system of networks. There are no mechanisms to promote data reliability, flow control, sequencing, or other services commonly found in host-to-host protocols [TCP/IP]. Each underlying network is required to provide only a minimal datagram level of service. The Internet Protocol does not provide any reliable communication facility. It simply transmits Internet Datagrams.

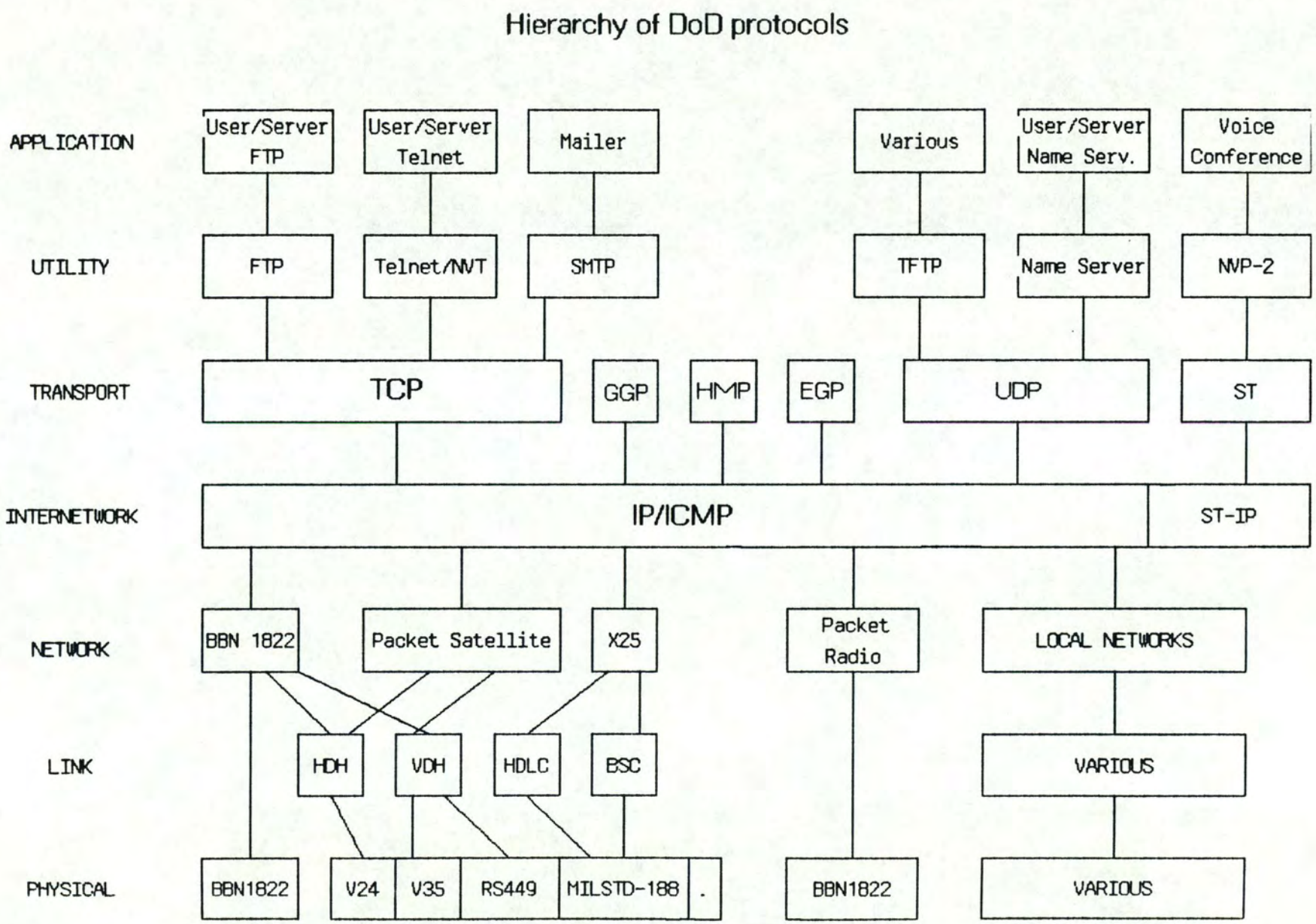


Figure 4.10: Hierarchy of DoD protocols

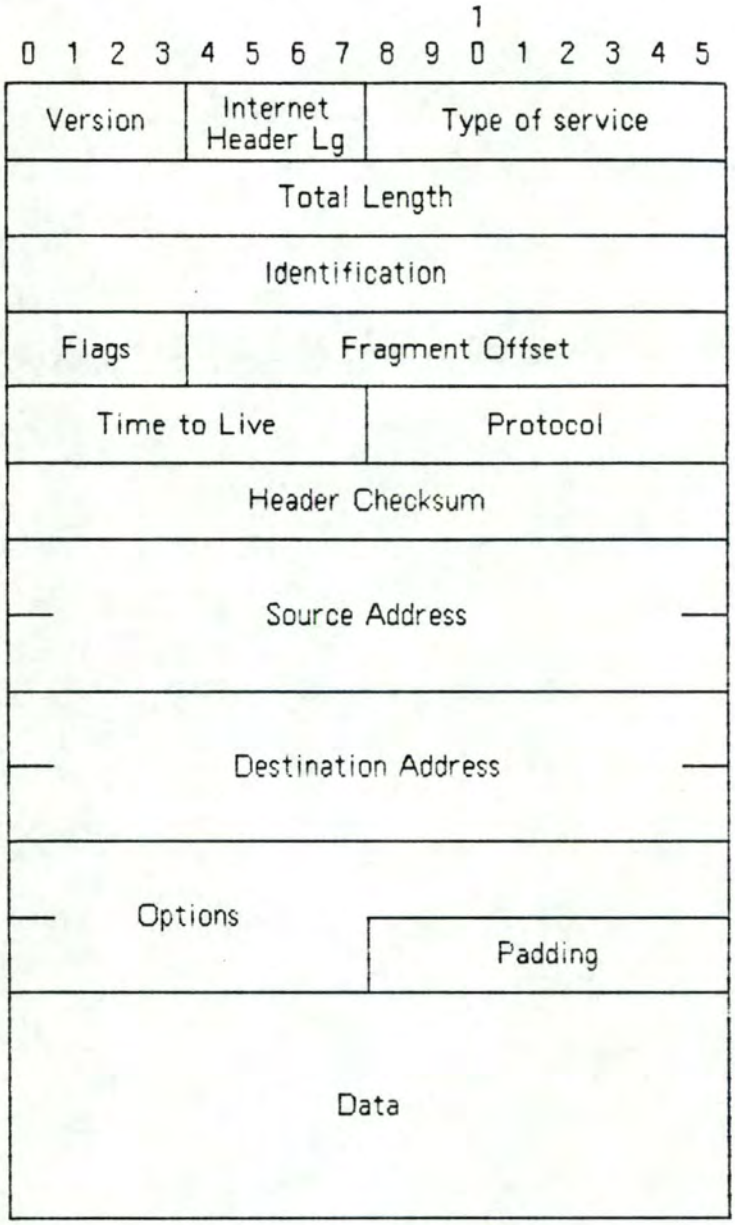


Figure 4.11: Internet Datagram format

As previously defined, the Internetwork Protocol resides in each interconnected host and in each IN Gateway between networks.

The Internet Protocol implements two basic functions. The first one, addressing, is provided via a two-level addressing hierarchy. The upper level of the hierarchy is the network number, and the lower layer is an address within that network. The second basic function is fragmentation (i.e. segmentation) of datagrams too large to be forwarded through a network. As shown in figure 4.11, the fragmentation is performed thanks to a More

Fragments Flag and a Fragment Offset Field. Additionally, a Don't Fragment Flag allows the originating host to specify that an Internet Datagram may not be fragmented, although it may force it to be discarded. Reassembly of fragments is performed at the destination IP (inter-network fragmentation).

The Type of Service Field is used to indicate the quality of the service desired: Interactive, Bulk, Real Time, etc . The Time to Live Field is an indication of the lifetime of an Internet Datagram. It is set by the sender of the datagram and reduced each time the datagram is processed by an IP Module. The time is measured in units of seconds. The datagram is destroyed when the time to live reaches zero before the datagram reaches its destination.
The Protocol Field indicates the next level protocol used in the data portion of the datagram.This allows the Internet Module to demultiplex the incoming datagram to higher level protocol modules for further processing.
The Header Checksum provides a verification that the information used in processing Internet Datagram has been transmitted correctly. The data may thus contains errors.
The Options Field provide for control functions needed or useful in some situations. The following internet options are defined [TCP/IP]:

- Security,
- Source Routing,
- Return (or Record) Route,
- Stream Identifier,
- General Error Report,
- No Operation,
- End of Option List.

4.6 Advantages and disadvantages of IN

A first advantage of the Internetwork Protocol is the adaptive routing provided by a connectionless Internetwork Protocol. This allows the selection of an optimal path for packets crossing multiple networks.

Another advantage is that it can deal easily with a great variety of networks because it does not perform any translation between Network Protocols. This is particulary true when the IN Protocol is connectionless. In this case it can be used with any kind of network without difficulty (LANs, WANs, Satellite, ...).

The corresponding disadvantage is that the IN Protocol must reside in each host, and not only in the IN Gateways. This is a severe limitation to the installation of an IN Protocol on many already existing networks. In the previous interconnection techniques, hosts of interconnected networks were not modified for installation of an interconnection technique.

Chapter 5 : THE PROTOCOL TRANSLATOR

The Protocol Translator is the last interconnection technique studied in the first part of this thesis. After defining the technique and the dual meaning of the term "Protocol Translator", we will illustrate this technique with a concrete problem, the interconnection of two electronic mail systems.

5.1 Definition

Let us first explain an important remark about the term "Protocol Translator". This term is used to designate two techniques radically different.

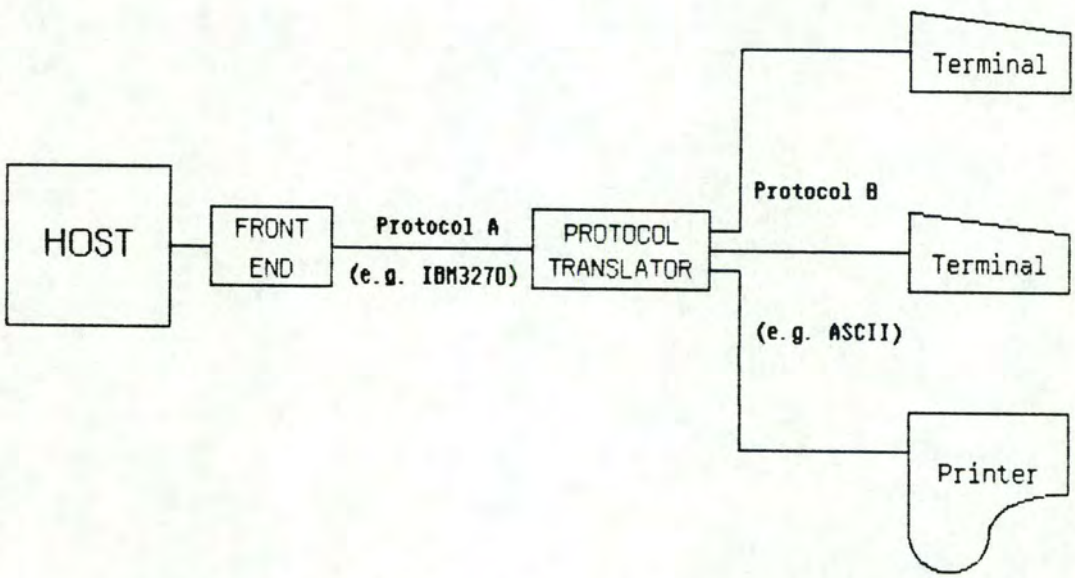


Figure 5.1: A terminal-to-host Protocol Translator

Peter Robinson defines a protocol translator as "a device that converts transmission signals from many different peripherals into the protocol used by a host computer" [ROB82]. For example, such a Protocol Translator is necessary to connect asynchronous terminals on a computer which only accepts synchronous terminals, or to convert an EBCDIC signal for use by an ASCII terminal. Figure 5.1 illustrates this kind of Protocol Translator. It may be designed to connect one or more peripherals and to provide conversion between two or more

protocols. Protocol Translators allow the owner of a mainframe computer to connect terminals from different suppliers. This attitude has three main reasons [ROB82]:

cost: The peripherals of the main supplier may be more expensive than these of other vendors. The Protocol Translator thus widens the range of products that can be used in a computer configuration.

upgradebility: Upgrades in software may need compulsory change of hardware. A Protocol Translator can avoid changing of peripherals and then lengthen the life of a product.

technology: Some specialised devices (e.g., graphic terminals) may be technically more advanced than those offered by the main supplier. The use of a Protocol Translator allows the connection of such devices to the standard interface of the mainframe.

The mandatory requirement is that the implementation of such a Protocol Translator within an existing network must be performed by a simple connection, without any change to the hardware or software principles of the host computer.

We agree with this use of the term "Protocol Translator" but this meaning does not fit into the subject of this thesis: interconnection of networks - a network being a set of interconnected computers. This kind of Protocol Translator does not interconnect computers. It is only concerned with the protocols defined between computers and non-intelligent devices, such as terminals or printers. In the OSI Reference Model, this kind of protocol translation is located in the Physical Layer and/or in the Datalink Layer. It is better termed a "hardware protocol converter".

Our definition of a Protocol Translator is a technique interconnecting two networks at a layer above the Network Layer. Thus, we call Protocol Translator every device that performs the mapping between two different Application, Presentation, Session or Transport Layer Protocols. Figure 5.2 illustrates the layers of the ISO Reference Model concerned with this Protocol Translator technique.

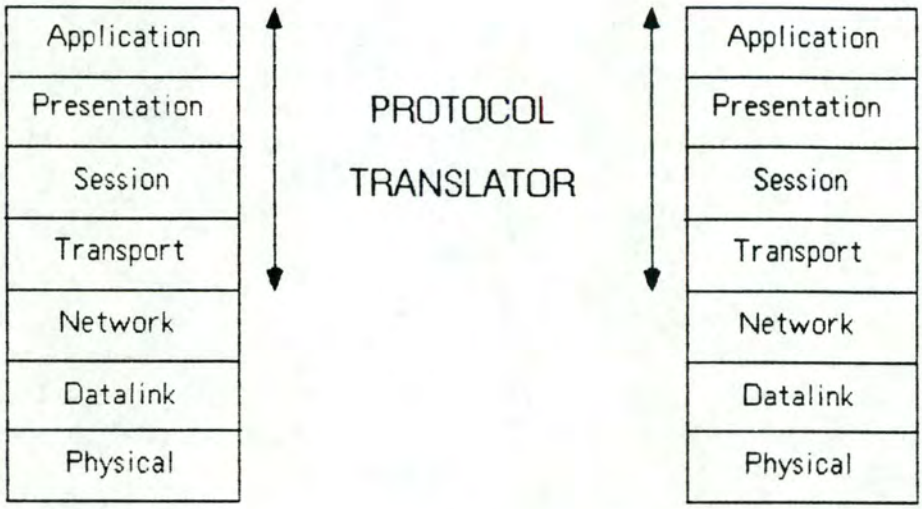


Figure 5.2: The OSI Layers concerned with the Protocol Translator

Eric Benhamou calls a Protocol Translator an application level-gateway [BENH83]:

> "Application-level gateways are used to interconnect networks that have different protocol architectures, and there are as many different types of application gateways as there are applications. The gateway communicates with each network in its own "language" and translates between the two. Some of the more common translations involve terminal protocols, document formats, mail formats, and file formats. (...) In application-level gateways, protocol translation occurs above the Transport Layer".

Figure 5.3 shows the operational model of an Application Layer Protocol Translator. An identical scheme could be drawn at Presentation, Session or Transport Layers. The Protocol Translator is a host for each interconnected networks because the protocols that it translates resides only in the hosts, not in the nodes.

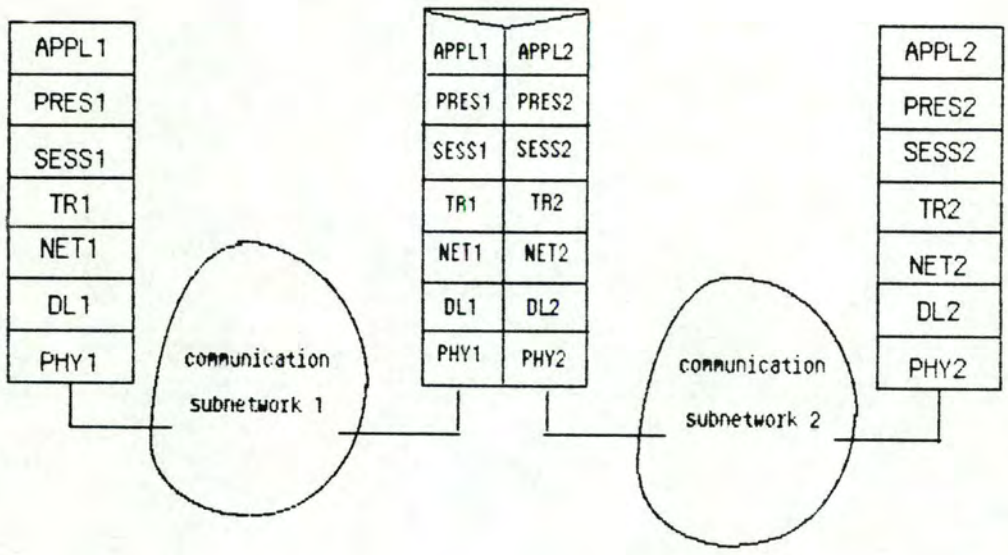


Figure 5.3: Operational model of an Application Layer Protocol Translator

The functions of a Protocol Translator may not be stated in a general way because they are application-dependent. This special-purpose nature is furthermore a major objection to this interconnection technique. A consequence of this special-purpose nature is that a Protocol Translator translates between two and only two protocols. It means that, if they are N different protocols to interconnect, and if full connectivity is required, then the number of Protocol Translators grows with the square of N (more precisely, N(N-1)/2). The design and implementation of a Protocol Translator is thus cost-effective if the number of users to interconnect is great enough. This is also called the "critical mass effect", i.e. how many people are reached by the Protocol Translator [RED83].

This explains why Protocol Translators are designed for use between applications such as:

- electronic mail,
- virtual terminal,
- virtual file,
- job transfer and manipulation,
- etc.

Further on in this chapter , we will study in more details the function of a Protocol Translator between two Electronic Mail Systems.

Care must be taken not to confuse the two meanings of Protocol Translator. For example, consider the figure 5.4.

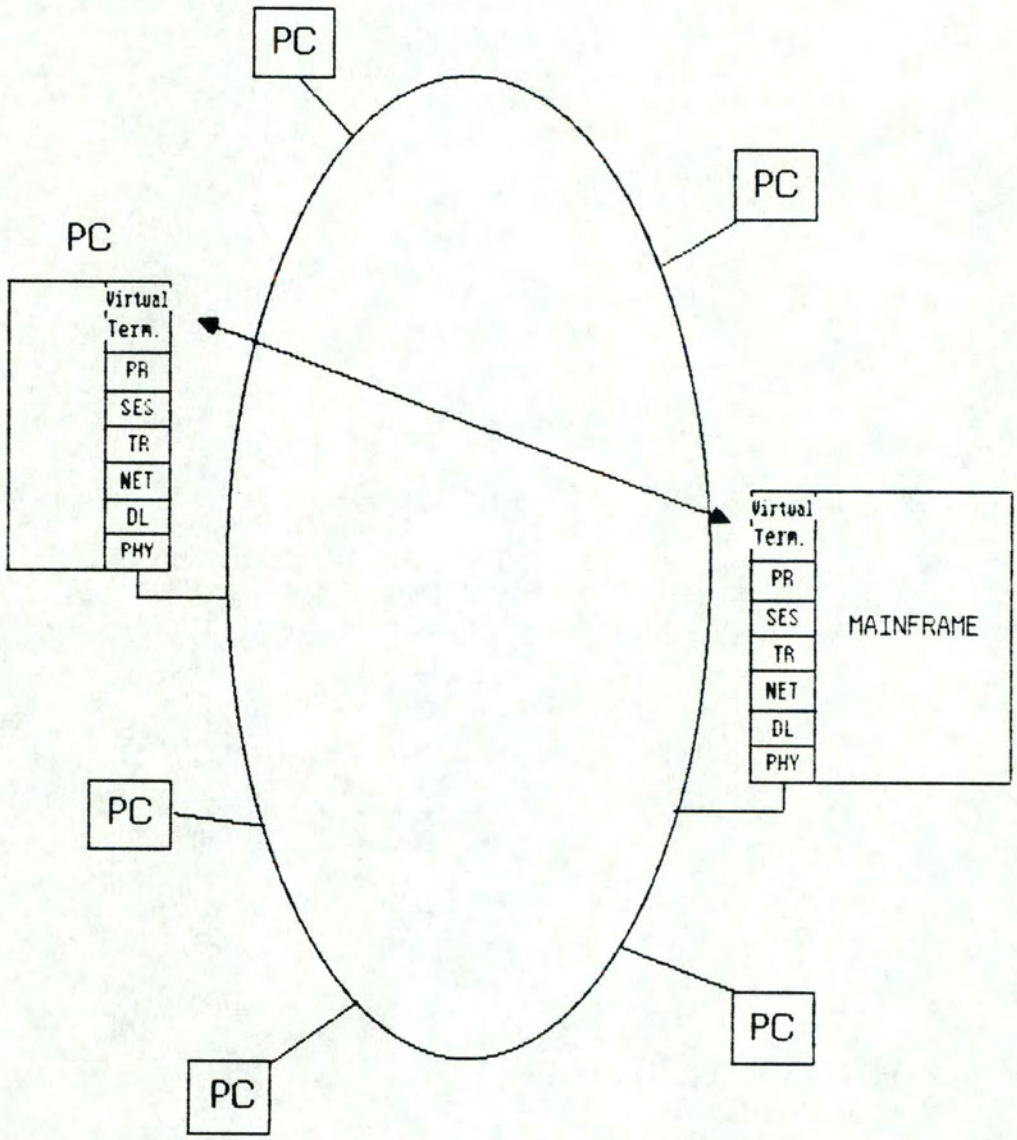


Figure 5.4: PCs used as terminals

It illustrates a network of Personal Computers (PC) and a mainframe computer. The PCs may be used as individual workstations or as terminals of the mainframe. In this latter case, the PCs and the mainframe run a Virtual Terminal Protocol at the Application Layer. Such a Protocol is often called an emulator (for example, an IBM3270 emulator). A PC running this emulator may converse with the mainframe exactly in the same way as a non-intelligent terminal directly connected to the mainframe. This emulator is not a Protocol Translator in the sense that we have defined because there is no interconnection of two networks. But this emulator is a "hardware protocol converter" as defined at the beginning of the chapter, because it allows a terminal (the PC) to be connected as a simple terminal of the mainframe. The fact that this is realized via a network does not change the service provided. This service is the one of a "hardware protocol converter".

We deplore this confusing terminology but it reflects the literature's and vendor's confusion.

5.2 A Protocol Translator between two Electronic Mail Systems

An Electronic Mail System is a typical Application Layer Protocol. The fundamental goal of interconnecting two or more Electronic Mail Systems is to enable users in one system to send messages to, and receive messages from, users in another. We call each component mail system a domain. Each domain has its own internal protocols and its own services provided to its users. Domains are interconnected by Protocol Translators.

Whenever a message (or information about a message such as delivery acknowledgment) crosses a Protocol Translator between two domains, some problems can arise. These are enumerated by D. Redell and J. White [RED83]:

1. Reconciling system features:
   The functions of two mail systems may be different. One may provide confirmation of message delivery, but others does not. It may result in inconsistency of the service provided in one domain. A solution to this problem is to define the features of a mail system as being mandatory or optional. The Protocol Translator have to be able to provide at least the set of mandatory functions.

2. Converting between content formats:
   This format conversion encompasses two quite different kinds of transformation. The first is the conversion from one representation to another, for example, the conversion from EBCDIC to ASCII. The second is the conversion from one medium to another, for example, the conversion of textual information from a sequence of characters to a facsimile image or to voice. The more viable approach to conversion problem is to define a standard format and to provide in the Protocol Translator facilities for converting any non-standard format to or from it. This approach is pursued by some vendors and standards organizations.

3. Naming recipients:
   Each mail system has its own naming principles. The ideal approach for Electronic Mail Systems interconnection would be to design a new worldwide standard name space, allowing all users in all systems to be uniformely named. The practical alternative is to form a single global name space by simply combining the existing name spaces. It results in two-parts names, in which the high order part identifies a domain and the lower order part a name in that domain. But originators must be able to specify the two-part name of their recipients.

4. Maintaining distribution lists:
Distribution lists are an important feature of Electronic Mail Systems. They identify logical groupings of those who should receive certain sets of messages. The names used to identify foreign recipients must be acceptable in local distribution lists.

5. Crossing political boundaries:
Each domain of a set of interconnected mail systems is administratively unique. This may result in some problems such as accounting between mail systems.

6. Ensuring message system security:
Users of the mail system must have confidence that the system will make messages available to their intended recipients and no others. Originators must be confident that others are unable to send forged messages in their names.

PART 2 : A PRACTICAL IMPLEMENTATION : THE FRIGATE PROJECT

This second part is a description of a practical network interconnection realized at the European Center for Nuclear Research (CERN), Geneva. This realization is called FRIGATE (Flexibly Reconfigurable Internet Gateway). This is a MAC-Sublayer bridge as defined in chapter two.

The first chapter describes the networking environment in which the Frigate Project is born. It consists of a description of CERNET, the high-speed packet-switching network of the CERN, and of Ethernet and IEEE 802.3 Protocols, which are widely used at CERN. Chapter 7 explains the needs to be satisfied by Frigate and specifies the functions to be realized. The Frigate Project has two constituent parts: a hardware part and a software part. Chapter 8 describes the hardware and the software of Frigate and the tools available to implement it. The ninth chapter is an evaluation of the Frigate Project as a whole and an evaluation of our work in this project, which concerned a software component of the project. The last chapter draws the conclusion of this thesis.

The descriptions of the hardware components and the software tools contained in this part are not exhaustive. They aim at giving to the reader a general overview of these subjects. An interested reader may refer to the reference manuals (see the bibliography) for a complete description. In the same way, our programming work is fully described in the annex which contains a programming documentation file produced by the documentation utility CWEB (see paragraph 8.3).

It should be noted that some CERN internal documents have been used to describe CERNET and the Frigate Project. They are not all explicitely cited in the text but a complete list of the CERN documents used may be found in the bibliography.

Chapter 6 : THE FRIGATE PROJECT ENVIRONMENT

This chapter is a general description of CERNET, the main network developed at CERN. It is based on a layered model a bit different from the OSI Reference Model. The layers are described and their differences from the ISO layers are emphasized. The protocols of each layer are summarized and compared with the X25 protocols. Because Frigate interconnects many Ethernet or IEEE 802.3 networks, these protocols and their differences are reminded.

6.1 CERNET: a High-Speed Packet-Switching Network

6.1.1 Introduction

In 1975, it was decided at CERN to construct a general-purpose data communication network to be used for computer-computer communications. The first aim was to provide communication between minicomputers located on experiment sites and mainframes at the computer center. The objective was not to provide centralised recording of raw data but rather to provide the possibility to send samples of the raw data to the central computer for analysis on a much shorter timescale than that obtainable by physical transfer of a magnetic tape. The definition of this network, called CERNET, includes general purpose facilities, such as file access and transfer, remote job entry and resource sharing between mainframes. By the end of 1980, CERNET was completely installed with 50 user computers.

CERNET was designed as a packet-switching network. It is thus composed of a communication subnetwork and a set of user computers communicating via the communication subnetwork (see figure 6.1). It is made of node computers and data-links. User computers will be called subscribers in the following of this thesis. There are two categories of subscribers:

- Minicomputers located on physic experiment sites. They aim at communicating one with another and with computers of the second category of subscribers.
- Mainframes computers of the computer center. They are called host because they provide computing services to other subscribers.

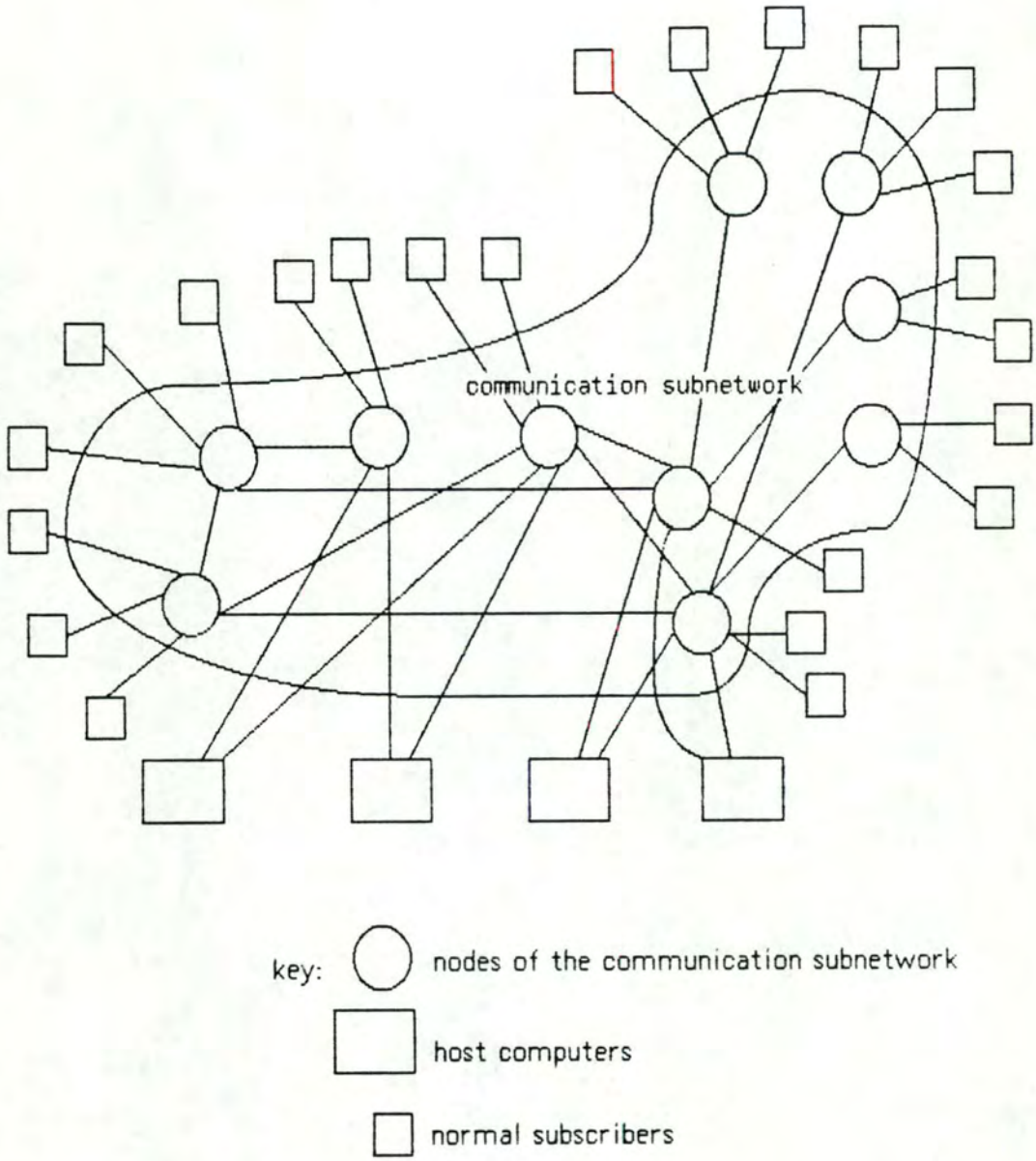


Figure 6.1: CERNET communication subnetwork and subscribers

This explains why first category subscribers are connected to one node, whereas host computers are connected to at least two nodes. This redundancy has been provided to ensure a very high-level of availability of the communication subnetwork for such machines.

6.1.2 CERNET Layering Model

The development of CERNET was started before the ideas and standards of ISO and CCITT had been developed. The CERNET architecture was therefore dictated by considerations of hardware design, software modularity and programming practice. This has led to a logically layered structure which can, from certain points of view, be contrasted with the OSI Reference Model.

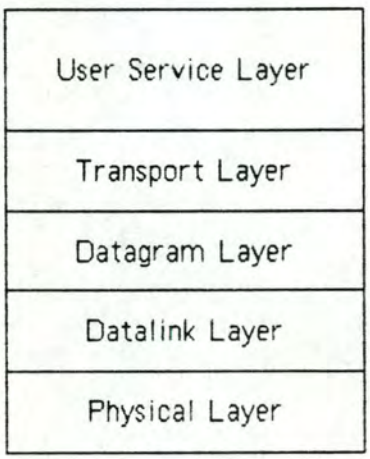


Figure 6.2: CERNET Layering Model

As shown in figure 6.2, the CERNET protocol architecture is divided in five layers. As in the OSI Reference Model, each layer is built on top of the underlying layer. Each layer provides a set of services to the higher layer. A layer provides its services by the use of the services of the lower layer. But one major difference with the OSI Reference Model is the following: In OSI, each layer is implemented in one protocol; There is a one-to-one relation between layers and protocols. In CERNET, this is not the case because the Datagram and Transport Layers are implemented in one common protocol. This is due to the fact that CERNET was developed with protocol modularity constraints in mind rather than protocol layering. This melting of two layers in one protocol has two consequences:

- The Protocol Data Unit format of this protocol has two headers, one for each layer;
- This protocol is not implemented in the same way in subscriber computers than in node computers because nodes are not concerned with Transport Layer functions.

These consequences will be explained in details later.

The Physical Layer is defined by CERN interfaces built for nodes and subscribers. Asynchronous serial transmission is used on full-duplex links capable of transmission speeds of several megabits per second over several kilometers. The links are connected to computers via controllers based on the CAMAC Standard. This allows the same data-link interfaces to be used for all types of subscriber computers, whereas these come from different manufacturers.

The Datalink Layer is concerned with the communication on a link between two nodes or between a subscriber and a node. It relies on the exchange of data units called packets. This exchange is managed by the use of control-words. The characteristics of this layer are the following:

- no connection or multiplexing service,
- normal transfer of data with flow control,
- sequential delivery guaranteed,
- no segmenting or blocking of packets,
- error detection and notification,
- error recovery by retransmission.

As in the OSI Reference Model, the CERNET architecture identifies two sublayers in the Datalink Layer. The Basic Datalink Sublayer recognizes the basic units from which the protocol is constructed (control-words and data packets). It ensures delimitation of packets and checksum management. The second sublayer, the Upper Datalink Sublayer then uses these basic units to control the data exchange. It ensures the acknowledgment, flow control, loss detection and error recovery functions.

The Datagram Layer corresponds roughly to the OSI Network Layer, in that it provides an end-to-end service between two subscribers. Protocol Data Units exchanged at this layer are called datagrams. The characteristics of this layer are the following:

- no connection and multiplexing service,
- normal transfer of datagrams without flow control,
- no segmenting and blocking,
- detection and notification of datagram protocol errors, but with no recovery.

The Transport Layer controls the interchange of messages. It provides all the services required by the OSI Model for a Connection-Oriented Transport Layer:

- establishment and release of a connection,
- normal transfer of data with flow control in the form of read-master requests,
- acknowledgment of messages,
- segmenting of messages into datagrams (and their blocking on receipt),
- sequencing of datagrams within messages and of messages,
- error detection and notification of loss, duplication and disordering of datagrams,
- no automatic recovery from these errors,
- no reset function on logical links.

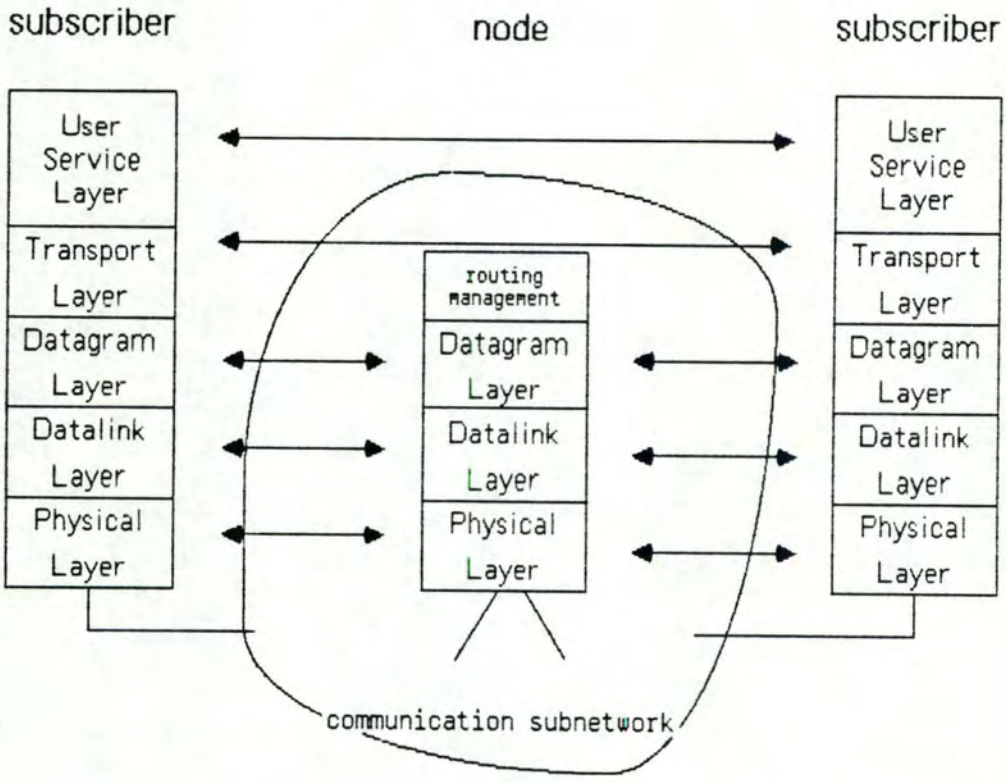


Figure 6.3: CERNET Operational Model

The User Service Layer Protocols are all provided directly on top of the Transport Layer. Process-to-process communication is thus performed directly through the Transport Layer, and any session control recovery, data formatting, etc... are the responsibility of the processes involved. Two protocols are provided at this layer: a Virtual Terminal Protocol and a File Access Protocol. This last one performs certain functions which inherently belong to the various higher layers of the OSI Reference Model:

- Session Layer: user identification;
- Presentation Layer: file opening parameters, data type, record length, ... ;
- Application Layer: the file may be written to disc (virtual file protocol), to the printer or to the job input queue (job submission protocol).

As shown in figure 6.3, the three lower layers of CERNET are present in subscriber machines and in nodes. But because the Datagram and Transport Layers are implemented in a single protocol, there are some differencies in the Datagram Layer implementation in nodes and in subscribers. In a subscriber machine, the upper interface of the Datagram Layer is not accessible, being part of a single module. Within a node, the services of the Datagram Layer are directly available through a strictly defined interface. These services are used to provide functions of remote access, line management and routing control.

6.1.3 CERNET Protocols

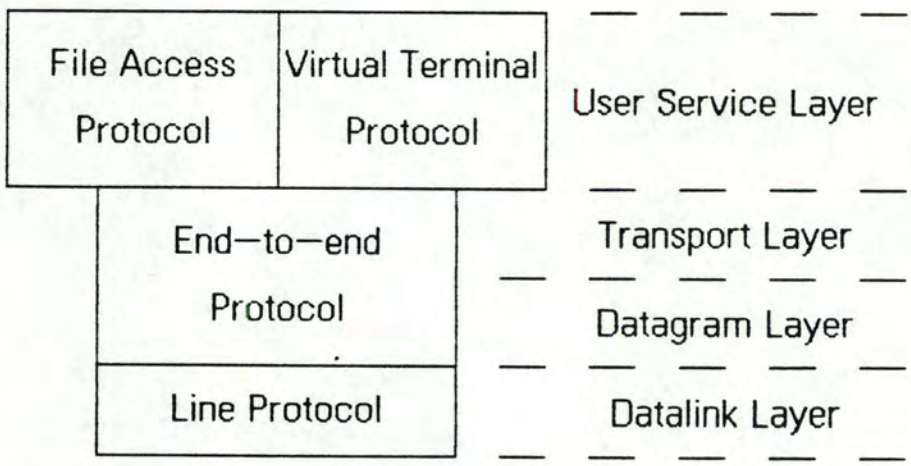


Figure 6.4: CERNET Protocols

Figure 6.4 illustrates the protocols of CERNET and their corresponding layer. As previously explained, a single protocol, the End-to-end Protocol, provides the functions of the Datagram and Transport Layers. At the User Service Layer, two protocols have been defined: the File Access Protocol and the Virtual Access Protocol.

In order to prevent excessive data storage in the communication subnetwork, a common scheme has been adopted in the design of CERNET Protocols: data should never be sent before it is requested [CERNET81]. This is known as the receive-master scheme, and simplifies decisions as to who should store data, set timeouts and who should take the initiative in the case of errors. In a receive-master scheme, the slave - i.e. the sender - has to follow the guidance of the master - i.e. the receiver.

The CERNET Protocols are the following:

- The Line Protocol,
- The End-to-end Protocol,
- The File Access Protocol,
- The Virtual Terminal Protocol.

6.1.3.1 The Line Protocol

The Line Protocol is the Datalink Layer Protocol. It is designed to ensure the safe delivery of data from one node to its neighbour or from a subscriber to a node. This implies that correct delivery is reported to higher layer protocol only if the data arrives error-free. The Line Protocol is full-duplex and provides flow-control, error checking and recovery by the exchange of control-words and data.

There are two main differences between the Line Protocol and the Datalink Layer of X25 (LAP-B):

- The absence of encapsulation of the data from the higher layer with a header and a trailer. Protocol Data Units manipulated by the Line Protocol are thus the datagrams of the upper layer protocol. The only "trailer" added by the Line Protocol is the Cyclic Redundancy Checksum (CRC). For this reason, what we call a "packet" in the Line Protocol is a "datagram" of the End-to-end Protocol plus a CRC.

- The use of control-words to exchange all the control information between the transmitter and the receiver. This is a logical consequence of the absence of specific header in the Line Protocol. In LAP-B, this header contains all the control information which is here sent separately before or after data transmission. Because of the receive-master

scheme, control words sent by the receiver are not the same than those sent by the transmitter. The receiver may acknowledge a transfer, indicate an error or authorize a packet transfer. The transmitter may acknowledge a transfer authorization.

Another consequence of the absence of Line Protocol header is that the length of a packet must be given to the receiver before the packet would be sent. This allows the receiver interface to calculate the CRC during the packet arrival and to check it immediately after. The control-word containing the length of a packet is called a wordcount. Figure 6.5 is an example of the working of the Line Protocol.

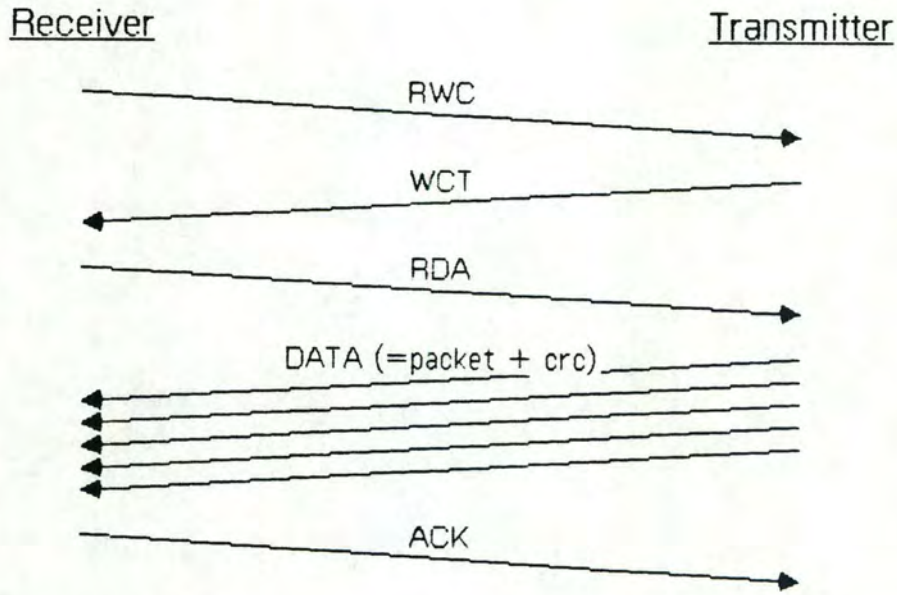


Figure 6.5: Example of a simple data transfer

Because of the receive-master scheme, the receiver initiates the transmission by sending a "Ready for wordcount" (RWC). If the transmitter has a packet to transmit, it sends a wordcount containing the length of this packet. The receiver acknowledges the wordcount and asks for the packet by sending a "Ready for data" (RDA). The packet and its CRC are then sent by the transmitter. When the transfer is complete and if no error bits are present, the receiver sends an "Acknowledge" (ACK). If there have been error bits, it would have sent an "Error Indication" (ERR) control-word.

Additionally , timeouts are set up to prevent loss of data or control-words or to prevent protocol errors. Flow control is possible for the receiver by speeding up or slowing down the sending of wordcount and data acknowledgments. An upgrading of this simple and robust scheme is the possibility of combining many control-words in one specifying multiple informations. The meaning of a bit in a control-word depends only on its position in the word. This allows the combination of multiple control-words with a logical OR and their sending in one control-word to accelerate the transmission.

6.1.3.2 The End-to-end Protocol

The End-to-end Protocol covers the Datagram and Transport Layers. It manages the communication between two subscribers and provides for the propagation of a packet across the communication subnetwork. The software implementing this protocol is called the Transport Manager (TM). The upper interface of this protocol is accessible to the user. The Protocol Data Unit used at this upper interface is the message. A user may ask to the End-to-end Protocol to send a message of whatever size. Inside the protocol, a message is sent via one or more datagrams. A Transport Manager is identified by a Transport Manager Address (or CERNET Address). CERNET addressing and routing are studied in the paragraph 6.1.4.

In a similar way to that in which the Line Protocol uses control-words, the End-to-end Protocol relies on the exchange of special control packets. But the difference is that control information and data can both travel in the same datagram, which is not the case is the Line Protocol. Figure 6.6 shows the format of a datagram. Because the End-to-end Protocol covers two layers, there are two different headers is a datagram. The routing header corresponds to the Datagram Layer and the protocol header to the Transport Layer.

The routing header contains the full size of the datagram, the source and destination addresses and a routing facilities field.
The protocol header contains the link number at source and destination side. As in the X25 Network Layer Protocol, link numbers may be different for the source and for the destination. But in the End-to-end Protocol, the two link numbers are present in each datagram crossing the logical link.
The Datatype field indicates that the data of this datagram is the last data of a message.
The control type field indicates that the rest of the protocol header is to be interpreted as control or status information.

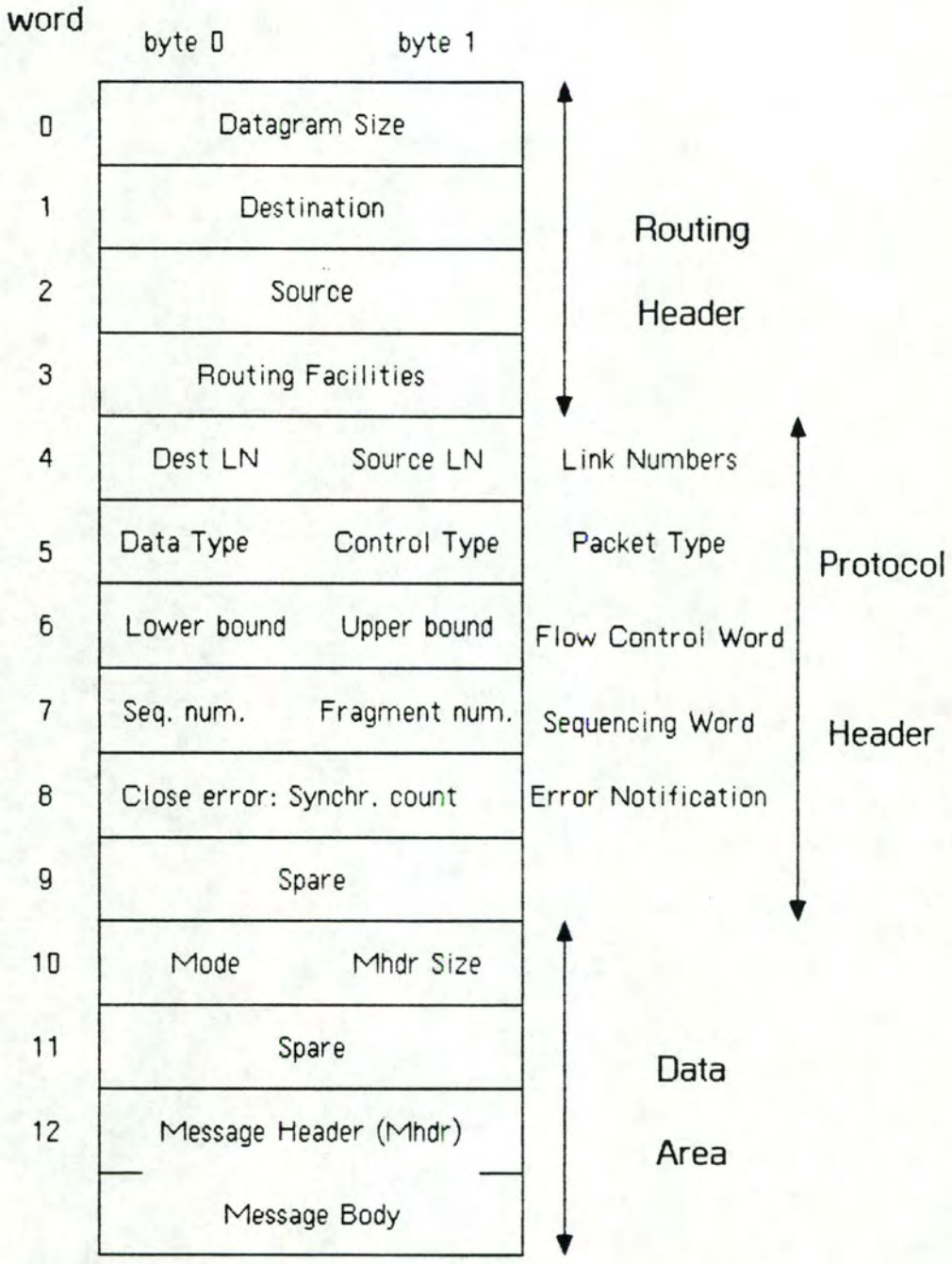


Figure 6.6: A datagram of the End-to-end Protocol

A window mecanism is used to provide flow control following the receive-master scheme. An important difference with X25 is that it is not the datagrams which are cyclically numbered, but rather the messages. The receiver indicates to the sender the

range of message numbers that it will accept. The sender must acknowledge this window information, and can send, in correct sequence, the messages whose numbers have been authorized in this way. The receiving Transport Manager checks the sequencing of messages as they arrive. The sixth word of a datagram contains the lower and upper window bounds.

The following word contains sequencing information. A message can be segmented for transmission, so that each fragment (or segment) can fit into a single datagram. Thus, the message sequence number is the same for all the datagrams of a single message. The fragment number identifies each datagram (i.e. each fragment) of a message. The receiving Transport Manager will check the fragment sequencing and reassemble the complete message before passing it to the higher layer. But the absence of upper limit to the message size implies that buffer shortage can occur. This problem is avoided by giving to the upper layer the buffer allocation responsibility, as well for sending than for receiving of messages.
The eighth word contains some error notification information and the last word of the protocol header is always zero.

Additionally to these features, the interface with the next higher layer provides the capability of splitting the data of a datagram into two parts: header and body. This simplifies the task of the entities at the higher layer, for whom the header will embody its protocol header and the data will carry the "transparent data". But such an interface definition does not respect the ISO layering principle which recommends full data transparency between two consecutive layers. The Data Area of a datagram thus contains a message header length field, a message header field and a message body field.

Other differences between the End-to-end Protocol and X25 are the absence of resetting or restarting of a link and the absence of expedited data. Multiplexing between the Transport Layer and the Datagram Layer is not relevant because of the connectionless characteristic of the Datagram Layer.

These differences are essentially due to the fact that CERNET was designed before the X25 recommendation and to the different requirements between private and public networks. The management of a link status, although useful, is less essential on a private network where a user is much more a part of the entire network. In the same way, the connection service is less essential because there is no accounting and billing, and because confidentiality between users is less crucial [NGN34].

6.1.3.3 The File Access Protocol

The File Access Protocol is the first protocol of the User Service Layer. It allows remote access to file services between computers of different types. In particular, it permits that the host computers offer a service by which any subscriber can access to the host's file base in a consistent manner. The main characteristic of the File Access Protocol is that it is an asymmetrical protocol. The hosts of the computer center run the full protocol and provide access to their files, whereas other subscribers - i.e. minicomputers on experiment sites - run only the part of the protocol giving them access to the files of the host computers. These subscribers do not provide access to their own files. Figure 6.7 illustrates the asymmetry of the File Access Protocol. To provide the full protocol, host computers run two modules:

- The File Manager (FM) that manages the access to the host's file base.
- The File User Interface Package (FUIP) which is a set of routines interfacing with the File Manager and allowing all facilities of a system file (open, close, read, write, ...).

To have access to the files of a host, a subscriber has only to implement a File User Interface Package.

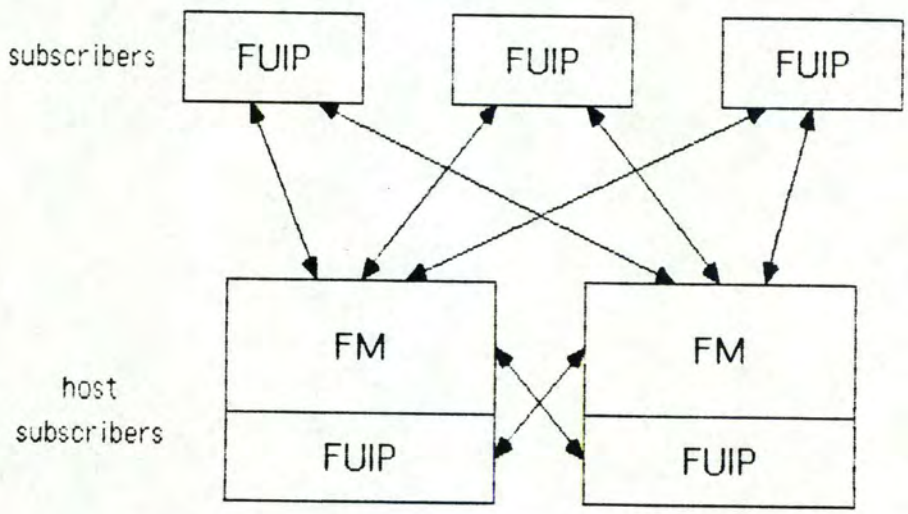


Figure 6.7: The asymmetry of the File Access Protocol

The facilities provided by the File Access Protocol are the following [NPN04]:

1. The access to a file is performed on a connection established between an FUIP and a host FM. On any connection to a FM there can only be a single open file at any time. This restriction greatly simplifies the protocol.

2. The protocol allows a subscriber to perform a sequence of basic operations on a file: open, read, write, close.

3. Sequential, partitionned and random access files can be accessed, if supported by the host File Manager.

4. The protocol recognizes many data types: text, binary blocked, binary unblocked, etc.

5. At any one time, a file can be used only for input or only for output (exception in the case of random access files).

6. The protocol provides for passing account and authorization information to the host file manager when accessing a file.

The user can choose between two speed/reliability compromises depending on the opening mode. In normal mode, the File Manager performs full handshaking with the host file system. The status of each operation is available, which allows the subscriber to work exactly as if the file was on its machine. In accelerated mode, there is no handshaking and no flow control. In case of problem, a simple closing message with an error code is sent to the user.

In practice, the File Access Protocol has been almost exclusively used to write programs for the transfer of complete files between hosts and subscribers. Its use has shown that high data rates can be achieved [CERNET81].

6.1.3.4 The Virtual Terminal Protocol

The Virtual Terminal Protocol is the second protocol of the User Service Layer. It has been developed when it became apparent that a requirement existed to communicate, via CERNET, between terminals attached to minicomputers subscribers and the terminal system of the large host computers.

The Virtual Terminal Protocol (VTP) is the set of rules for the process-to-process communication which drives a Virtual Terminal (VT), where one process is the host computer terminal system and the other process is a program in the subscriber machine which is driving the user's real terminal. The Virtual Terminal is an imaginary device which provides a standard representation of a canonical terminal. The task of the remote program driving the real terminal is to map the VT onto the real terminal following its characteristics. [CERNET81].

The protocol is simple, easy to implement and efficient in CERNET link usage. It is symmetrical because there is no concept of sender and receiver. Processes exchange messages built up with items chosen in a list of valid items. The symmetry also comes from the fact that these lists are identical for the two processes. Figure 6.8 shows a message composed of four items [NPN78]. Each item is made up of an item code field followed by a length field and a variable length parameter field. This message will screen the text "this is text" followed by a carriage return and a bell. The fourth item indicates the end of the message. This message structure permits the program driving the real terminal to process messages in a data driven way, without the need to inspect each character of a text string.

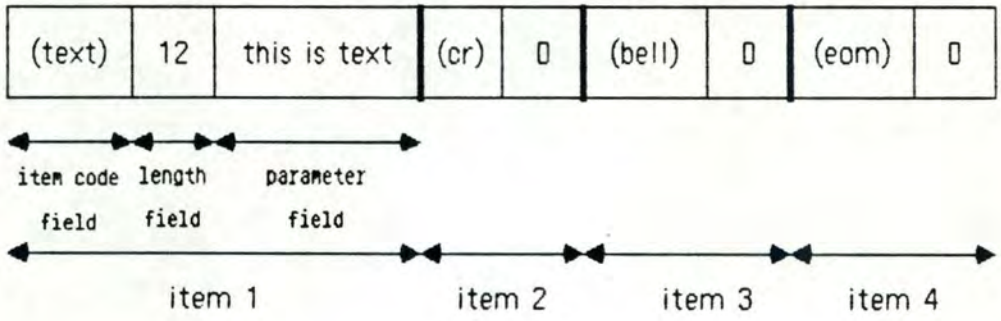


Figure 6.8: A four-item message

Two communication modes are defined. The Alternate Mode (half duplex) is basically that only one partner is allowed to transmit at any given time. The Free Running Mode (full duplex) allows either party to send messages when they wish, subject to flow control rules performed with flow control items.

6.1.4 CERNET addressing scheme and routing

The CERNET addressing scheme is involved in the End-to-end Protocol. A CERNET Address (also called a Transport Manager Address) is not the address of a physical subscriber machine, but rather the address of a Transport Manager running on a subscriber machine. This allows possible multiple TM on a single machine.

A CERNET Address is a 16-bit quantity, as illustrated by the figure 6.9. The least significant byte represents the relevant Transport Manager in the subscriber machine. The most significant byte is structured in two 4-bit values: region and machine. This scheme simplifies the naming of the various subscriber machines.

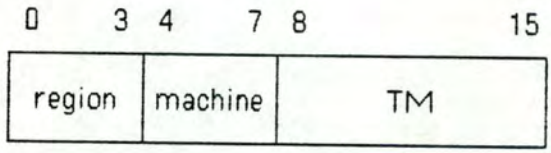


Figure 6.9: CERNET Address Format

The routing of a packet towards its destination is done by the communication subnetwork, examining only the most significant byte of the CERNET Address. In CERNET, the routing is [NPN69]:

- semi-static, because packets paths are fixed in the absence of network topology changes. Thus, paths are not modified due to loading modifications inside the communication subnetwork;
- decentralised, because the routing tables are updated in a diffusion process with no central control.

Every node contains a distance table and a best link table. The distance between two subscribers is measured in the number of links traversed. The distance table is a collection of triplets (link, destination, distance) giving for each link of this node all the reachable destination and the distance associated. The best link table gives for each destination the link of the shortest distance. Each node continually updates these two tables following every change in the state of the link (open, close). In addition, it diffuses each updating to its neighbour nodes, allowing them to update their tables.

6.2 The Ethernet and IEEE 802.3 Protocols

As introduced in the first chapter, Ethernet and IEEE 802.3 are two nearly identical protocols. Ethernet has been published in 1980 by three commercial manufacturers: Digital Equipment Corp., Intel Corp. and Xerox Corp. It is designed for a single-channel baseband transmission on a coaxial cable in a bus topology. In 1982, it was adopted by the ECMA (European Computer Manufacturer Association) and by the IEEE in a global Local Area Network project known as the IEEE 802 Project. Figure 1.4 of chapter one illustrates the components of this project. Ethernet was adopted under the reference 802.3. This section describes briefly the principles of the Ethernet Protocol and explains the differences between the original Ethernet and the actual 802.3 Standard.

6.2.1 The CSMA/CD technique

Ethernet is a contention-based random access procedure. This allows any station connected to the bus to transmit its message at any given time. This freedom implies the possibility of collisions. The random access sheme adopted to reduce the probability of collisions is CSMA/CD: Carrier Sense Multiple Access with Collision Detection. It is based on two rules:

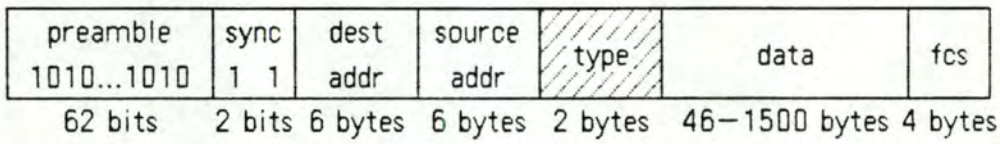
1. Listen before talking: If the channel is idle, the sender transmits its message. If the channel is busy, the sender waits (or backs off) before attempting to transmit.

2. Listen while talking: If a collision is detected, the station aborts its transmission and waits a certain period of time before attempting to retransmit.

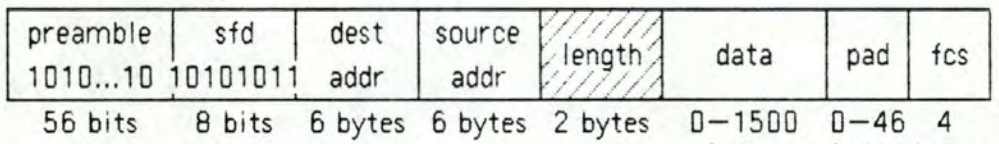
This backoff delay is a random multiple of the round-trip propagation delay, often called slot time. To avoid accumulation of retransmissions, this interval is adaptively adjusted to the actual traffic load (binary exponential backoff algorithm).

6.2.2 Frame format

As previously explained , Ethernet and IEEE 802.3 are Medium Access Control (MAC) Sublayer Protocols. The protocol data units exchanged at this sublayer are called frames. Figure 6.10 illustrates the formats of an Ethernet frame and of an IEEE 802.3 frame.
The preamble allows the physical interface to reach its ready state.
Ethernet defines a two-bit synchronisation field whereas IEEE 802.3 defines a eight-bit start frame delimiter. But this definition difference does not result in any difference in the first 64-bits values, as we can see on the figure 6.10



(a) Ethernet Frame Format



(b) IIIE 802.3 Frame Format

Figure 6.10: Ethernet and IEEE 802.3 frame format

Destination and source addresses are 48-bit values. The destination address may be [ETH82] [IEEE802.3]:

- individual address: the address associated with a particular station on the network;
- multicast-group address: an address associated by higher-layer conventions with a group of logically related stations. In this case the first bit of the address has the value 1;
- broadcast address: a distinguishable, predefined multicast address which always denotes the set of all stations on a given Local Area Network (on a given coaxial cable). In this case, the destination address field contains all 1's.

Ethernet defines a type field as a two-bytes value reserved for use by higher layers. The type field is uninterpreted at the MAC Sublayer. Because this upper layer dependence violates the OSI layering principle, the IEEE 802.3 Standard defines, at the same place in the frame, a length field whose value indicates the number of data bytes in the frame data field.
The data field must be at least 46-byte length in an Ethernet frame, whereas 802.3 permits a smaller data field size and adds, in this case, a pad field to reach the minimum 46-byte size.
The Frame Check Sequence field contains a 4-byte cyclic redundancy checksum (CRC) value computed as a function of the contents of the source, destination, type (or length) and data fields. The encoding is defined by the same polynomial in the two standards.

The difference in the type/length field will be of particular importance in the Frigate software design, due to the existence of both Ethernet and IEEE 802.3 Local Area Networks on the CERN site.

6.2.3 The physical interface

Ethernet and IEEE 802.3 Standards define the electrical and physical interface between a station and the coaxial cable. The only important thing to remind is that the physical device which is connected directly on the coaxial cable is a transceiver (see figure 6.11). It serialises/deserialises groups of eigth bits going to/coming from the Ethernet cable. Between the transceiver and the user machine, a parallel multi-wire cable is used. Additionally to the data, the transceiver transmits to the user station informations such as: channel busy, channel free, collision detected, ...

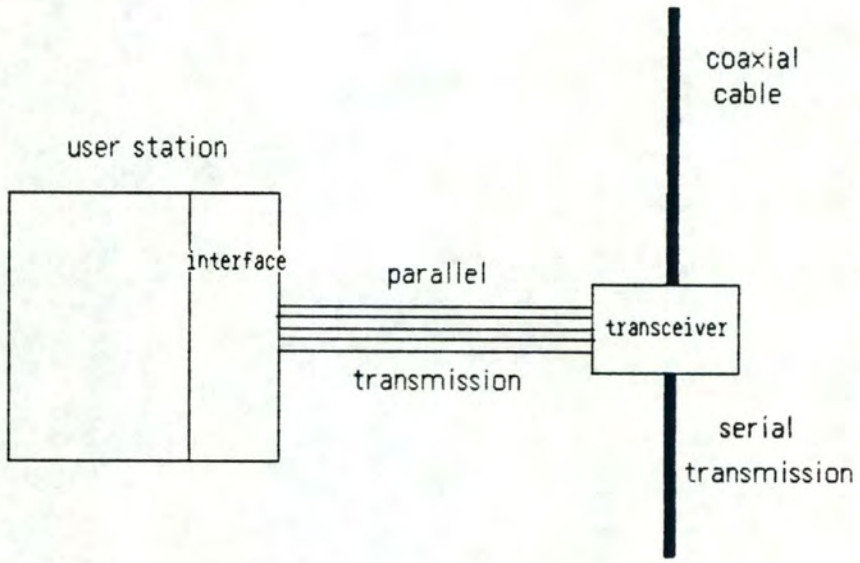


Figure 6.11: Ethernet and IEEE 802.3 physical interface

Chapter 7 : FRIGATE PROJECT DESCRIPTION

This chapter describes the Frigate Project from a functional point of view. The needs leading to the adoption of the Frigate Project are explained. The services initially defined in the Frigate Project are enumerated. Some of them have been reviewed since the adoption of the Project. The basic service providing network interconnection, the AID (Automatic Internet Datagram), is then fully described and specified.

In the following of this thesis, the term "segment" will be used to designate an Ethernet or IEEE 802.3 Local Area Network. This is due to the presence at CERN of multiple such LANs. The term "segment" reminds the piece of coaxial cable which physically represents the LAN. A "segment" is one of the Ethernet or IEEE 802.3 LANs installed on the CERN site. In the same way, the term "station" is used to represent a (generally intelligent) device connected to a segment.

7.1 Introduction

In 1982, a need was expressed by many users at CERN for an efficient and cheap way of connecting minicomputers to CERNET. Futhermore, the trend towards personal computers, workstations and other microprocessor-based systems has led to the installation of a large number of machines that cannot easily connect to CERNET because of the cost this would involve in hardware and software development. The emergence of Local Area Networks such as Ethernet promised a commercially available set of products for such machines.

In 1983, a general reassessment of computing facilities for CERN was carried out. One recommendation emerging of this work was that connectivity should be provided for all users of network services within the CERN. Following this recommendation, it was decided to promote the installation of Ethernet LANs and to design a gateway called FRIGATE: Flexibly Reconfigurable Internet Gateway.

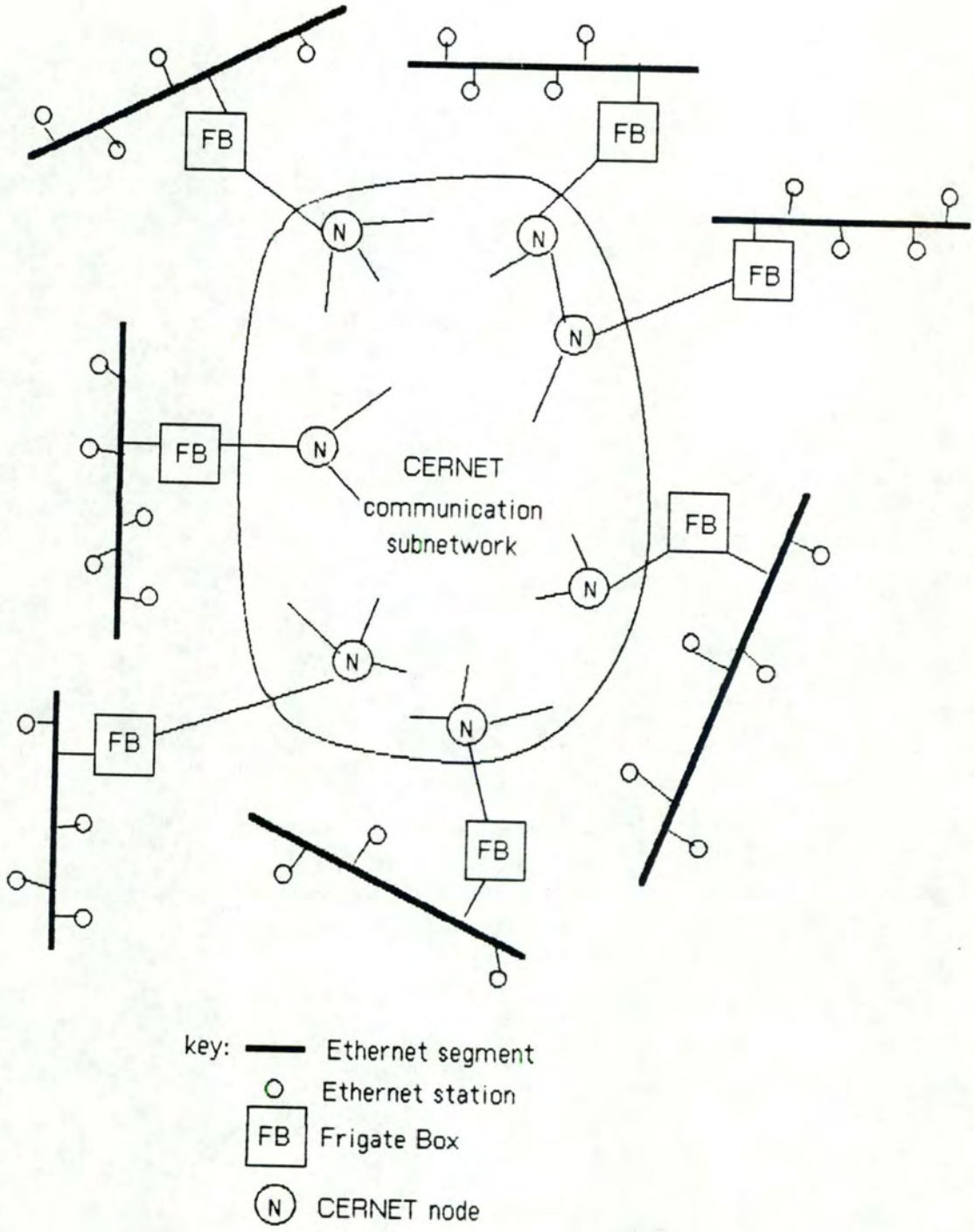


Figure 7.1: The Frigate Interconnection Project

The overall requirement of the Frigate Project was to provide an integrated set of services to all networking users within the CERN community. This would initially covers users

connected to CERNET or to LANs supported at CERN. Users of these LANs will be offered the same service functions as are available to CERNET subscribers. In addition, services to interconnect LANs is also provided. The first version of Frigate connects Ethernet segments to CERNET, as shown in figure 7.1. Further versions will be needed when other LANs are officially adopted at CERN or when a new "backbone" network starts to replace CERNET. All the services provided by Frigate run on special-purpose hardware, called the Frigate Box.

7.2 Frigate services

The Frigate Project provides a set of internetworking services in a flexible and open-ended manner [FRIG-P2]. Each service has been given a simple mnemonic name, to ensure easy identification. As the first LAN to be supported by Frigate is Ethernet, the descriptions are given relative to this technology, although the design will be general.

The first service offered is the Automatic Internet Datagram (AID) service. It allows all the Ethernet segments connected to this service to appear and operate as a single, homogeneous Ethernet. This transparent service is in fact the MAC-Sublayer bridging service defined in the chapter two. MAC frames emitted in one segment and addressed to a station located on another segment are forwarded from source to destination segment, in a transparent way for the users (see figure 7.1). This service will be studied in the rest of this thesis.

In addition, some CERNET subscribers need to use their CERNET software even when they are connected only to an Ethernet segment. The Manager-AID (MAID) service is provided to map CERNET and Ethernet addresses into each other. In this way, CERNET Transport Managers running on machines connected to CERNET or to an Ethernet can interwork tranparently. This service could be built on top of the AID service, but it is presently implemented in a CERNET node.

This service consists of catching CERNET datagrams whom destination address designates a Transport Manager running on a machine connected to an Ethernet segment. This datagram is then forwarded to the appropriate segment, via the Frigate Box.

The File-AID (FAID) service provides the same service as the File Access Protocol on CERNET. It allows Ethernet users to access, via a simple protocol, to the file bases of any mainframe computer connected to CERNET and running a File Manager. The Frigate Box operates as a server offering CERNET File Manager facilities to Ethernet users. This service is presently on installation on the Frigate Boxes.

Longer-term plans also included using a Frigate Box on which the Ethernet side is replaced by an X25 one, in order to provide an X25 "envelope" to CERNET users. This would allow CERNET to provide X25 network support. This service would be named X-AID. It has been abandonned due to the large choice of X25 commercial products.

7.3 The Automatic Internet Datagram Service

The AID service provides forwarding of MAC frames between two Ethernet segments in a transparent way. It is the set of all Frigate Boxes which provides this service, not only one Frigate Box. The use of CERNET is internal to the MAC-Sublayer bridging service, because AID does not allow an Ethernet station to communicate with a CERNET subscriber. The use of CERNET is fully transparent for two communicating remote Ethernet stations.

7.3.1 AID operation

Let us suppose that station A on segment 1 sends a frame to station B on segment 2 (see figure 7.2). The destination address of the frame contains the MAC address of station B. The Ethernet interface of the Frigate Box #1 receives this frame, because it works in promiscuous mode. It sees that the destination station of this frame is not located on segment 1 and builds a CERNET datagram which contains the MAC frame. Because Frigate Boxes are connected to CERNET as normal subscribers, the Frigate #1 have to find the CERNET address of the Frigate Box which is connected to segment 2, i.e. the Frigate Box #2. This mapping is done via addresses tables. The CERNET datagram is sent to this address. On reception of the datagram, the Frigate #2 extracts from it all the information necessary to rebuild the MAC frame, which is then sent on the segment via the Ethernet interface of the Frigate Box #2. The station B thus receives the frame exactly as if the source station was on segment #2.

7.3.2 AID functions

To provide the service described above, the following functions have to be performed:

- interfacing with Ethernet and CERNET networks. The Ethernet interface must work in a promiscuous mode to receive all frames generated on the segment. Frigate interfaces with CERNET as a normal subscriber. It thus runs the Line Protocol and the End-to-end protocol (Transport Manager).

- buffering on input and output in each Frigate Box. Ethernet input buffering must deal with peak traffic; output Ethernet buffering waits for the freeing of the channel. CERNET input and output buffering is necessary because processing time of bridging function is not negligible.

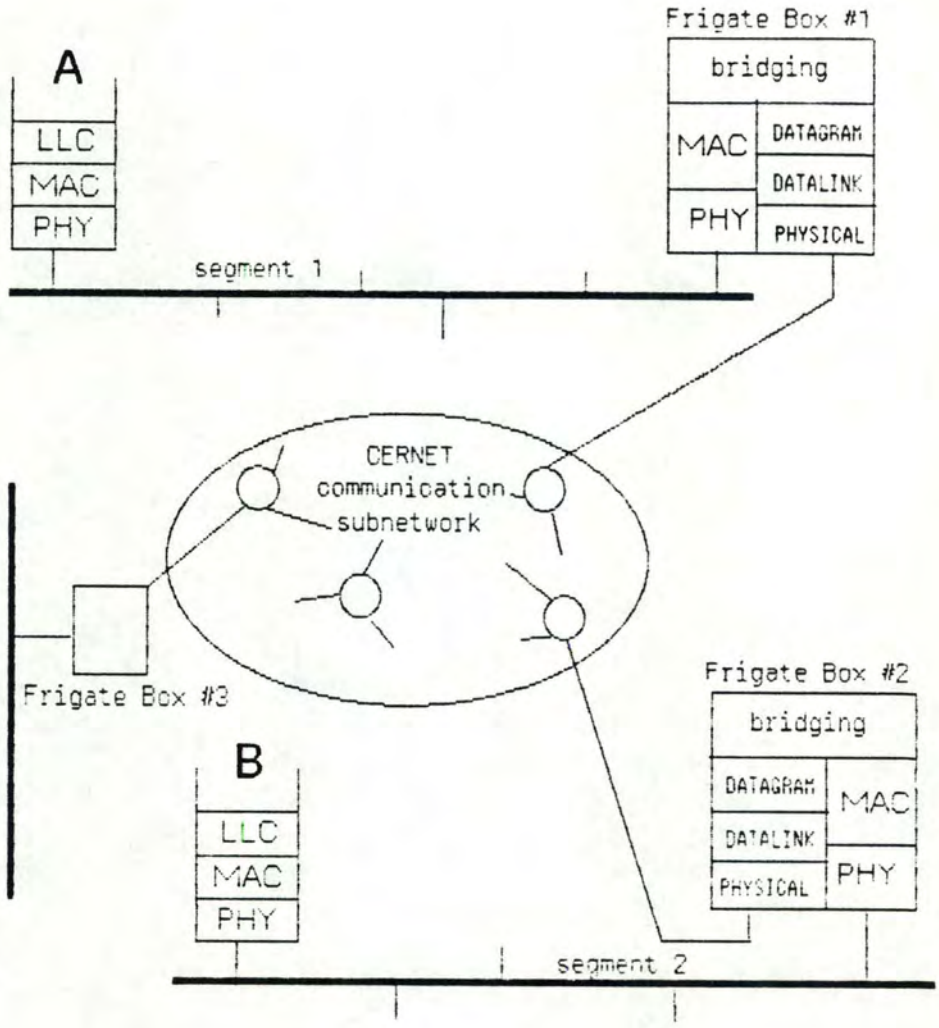


Figure 7.2: The AID Service Operational Model

- filtering of incoming Ethernet frames. Frames whom destination address is known to be local to the segment are discarded by the Frigate.

- segment identification and routing between segments. The first step is the ability to map any Ethernet address onto the CERNET address of the Frigate connected to the segment on which this address is located. The second step is automatically performed by the CERNET communication subnetwork.

The CERNET address of the Frigate Box connected to a segment will be used as segment identifier.

An additional problem arise from broadcast and multicast frames, i.e. frames with a destination address designating a set of all the stations. The Frigate must be able to forward such a

frame to all the other segments. This problem will considerably increase the CERNET usage.

7.3.3 The addresses mapping function of Frigate

From the functions listed above, the most important is the addresses mapping because this is necessary to perform Ethernet input filtering and switching to the appropriate Frigate Box.

In order to be able to filter Ethernet frames, it is necessary to obtain the correspondence between the address of a station and its segment identifier (i.e. the CERNET address of the Frigate connected to this segment). Segment identifiers are thus 16-bit values (see 7.1.4).

Because of the international agreement on the IEEE 802 Project, the MAC address of each individual piece of equipment is represented by one, unique 48-bit value, whatever the 802 MAC Protocol is used. It means that every Ethernet station around the world has a different MAC-address. It is obviously true for all the Ethernet stations on the CERN site.

7.3.3.1 The BISE-table

Each Frigate maintains a MAC address table of up to N entries (where N is the maximum number of possible stations on all the interconnected segments). As shown in the figure 7.3 each entry in the MAC table is 64-bit wide, being composed of a 48-bit field, to accomodate a full MAC address, and a 16-bit field, to accomodate a segment identifier. An entry in this table gives the MAC address of a station and the corresponding segment identifier to which this station is attached. Because there is usually many stations on a segment, the same segment identifier appears in more than one MAC table entry. The table is ordered increasingly on the MAC addresses. A binary search routine called BISE (Binary Search Engine) is used to access the MAC table, which is also called the BISE-table.

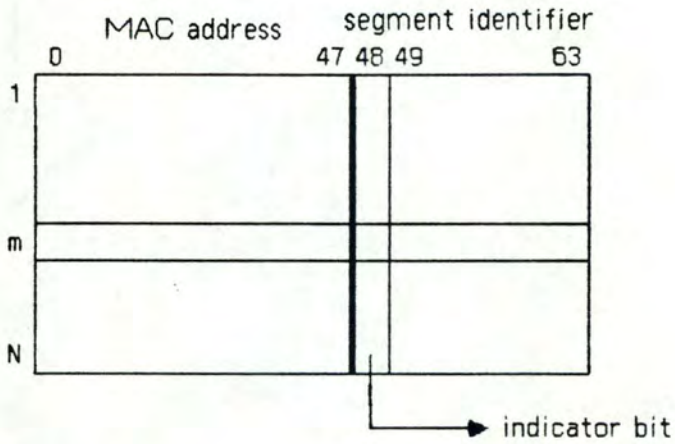


Figure 7.3: The MAC (or BISE) table

Additionally, segment identifier fields of some entries in the BISE table may have special meaning. In this case, the leftmost bit of the segment identifier field contains the value "1" (otherwise it is always zero). This "indicator bit" set to one means that the field does not hold the actual segment identifier, but rather a special segment information (see below, paragraph 7.3.5).

7.3.3.2 The BISE table maintenance

The information that can be used to update the BISE table is available from two places:

1. Frames generated by stations on the Ethernet segment: The source MAC address of these frames indicates which addresses are located on the local segment.

2. Datagrams from CERNET: They contains a CERNET source address - which is the source segment identifier - and a source MAC address. The combination of these two informations informs the destination Frigate about remote addresses.

With this information, it is clear that not all Frigate Boxes will contain identical BISE tables. Each Frigate will know about its local addresses which have been active, all their correspondents, and all sources that had to broadcast (because broadcast frames are forwarded to all other segments). The BISE table is thus dynamically created and updated.

It seems important to emphasize that an Ethernet station becomes known for a Frigate Box when a frame addressed to or coming from this station crosses the Frigate. It is the only way by which a new entry is inserted in a BISE table; There is no procedure for expanding a BISE table knowledge from one Frigate to another. We will see in the following that this dynamic address knowledge scheme has advantages and disadvantages. Every time a frame crosses a Frigate - in whatever direction - it checks the source and destination addresses for the case where a "known" station may have moved from one segment to another.

7.3.4 Data and header manipulation

In the AID operation paragraph (7.3.1), we have explained that a Frigate Box builds a CERNET datagram with an Ethernet frame addressed to a remote station. This paragraph explains how this building is made and why some choices have been made.

The most obvious approach would have been to simply encapsulate the entire Ethernet frame in a CERNET datagram and to forward it into CERNET. It was rather decided to carry the LAN data by means of the ISO Connectionless Network Service ISO/IS 8473. This is the protocol explained in chapter four as the Internetwork Protocol of ISO. This choice has two reasons:

- The general reassessment of computing carried out at CERN in 1983 recommends that CERN protocols have to evolve towards ISO protocols.

- The Frigate Project must be flexible. The replacement of any interface, Ethernet or CERNET, must be as straightforward as possible. And the replacement of CERNET by a new site-wide "backbone" network is planned for the next coming years.

Thus, the Ethernet frame is preceded by an IEEE 802.2 LLC1 header and then used to build an ISO 8473 Protocol Data Unit which is encapsulated in a CERNET datagram. The format of an ISO 8473 Protocol Data Unit is explained in details in the chapter four and illustrated in the figure 4.8. Figure 7.4 illustrates the "step up" performed in a source Frigate in order to send across CERNET and the corresponding "step down" on reception of the CERNET datagram, whereas the Frigates communicate with the segments at the MAC-Sublayer.

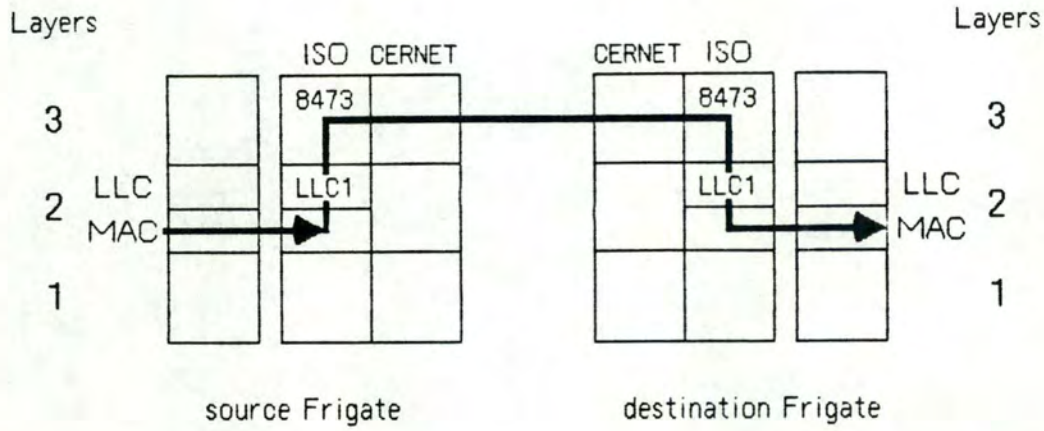


Figure 7.4: MAC to Network Layer conversion

This choice has an important consequence. Because the data part of the ISO 8473 Protocol Data Unit carries only the data part of the Ethernet frame, the information from the Ethernet Header is copied in the 8473 Internet header. This process is illustrated in the figure 7.5 for a conversion from Ethernet to CERNET.

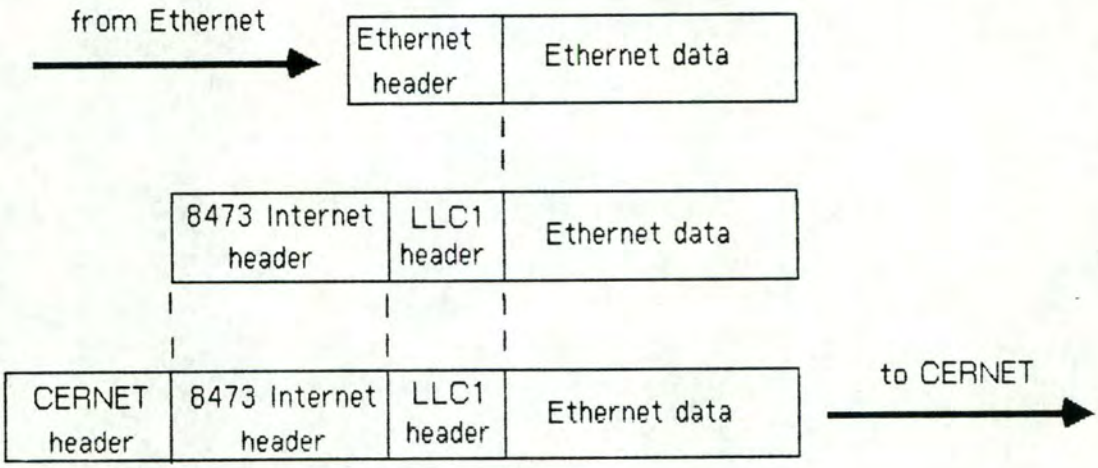


Figure 7.5: Header manipulations in Frigate

The presence of the LLC1 header is used to recognizes the datagrams addressed to the AID service than those addressed to other services running on the Frigate Box and than management datagrams between Frigates [FRIG-P12]. Each AID of each Frigate is given a 8-bit LLC address. This address is used to fill the Destination and Source Address Fields of the LLC1 header.

The ISO 8473 Internet header is built in the following way. The Fixed Part of the Internet header is left to zero except the PDU length. In the Address Part, the address fields are used to carry the 6-bytes MAC address plus an extra two bytes that are used differently for source and destination addresses. To the destination address are affixed the two "type" (or length in 802.3) bytes that directly follow the 12 address bytes in the Ethernet header. The source address is suffixed with two bytes which are the source segment identifier. These are set by the source Frigate for use by the destination Frigate to update its BISE table. This process is illustrated in the figure 7.6, which represents a full CERNET datagram as generated by a Frigate. The Ethernet Data Part is copied into the ISO 8473 Data Part, after the LLC1 header.

Let us remark that the source Frame Check Sequence is not transported across CERNET. This is a non-negligible restriction to the transparency of the MAC-Sublayer bridging service. For full end-to-end transparency (and for maximum end-to-end error detection) it would be desirable to transport and reissue the checksum exactly as it was in the source frame received by the Frigate. This is not currently done, although the hardware would allow it.

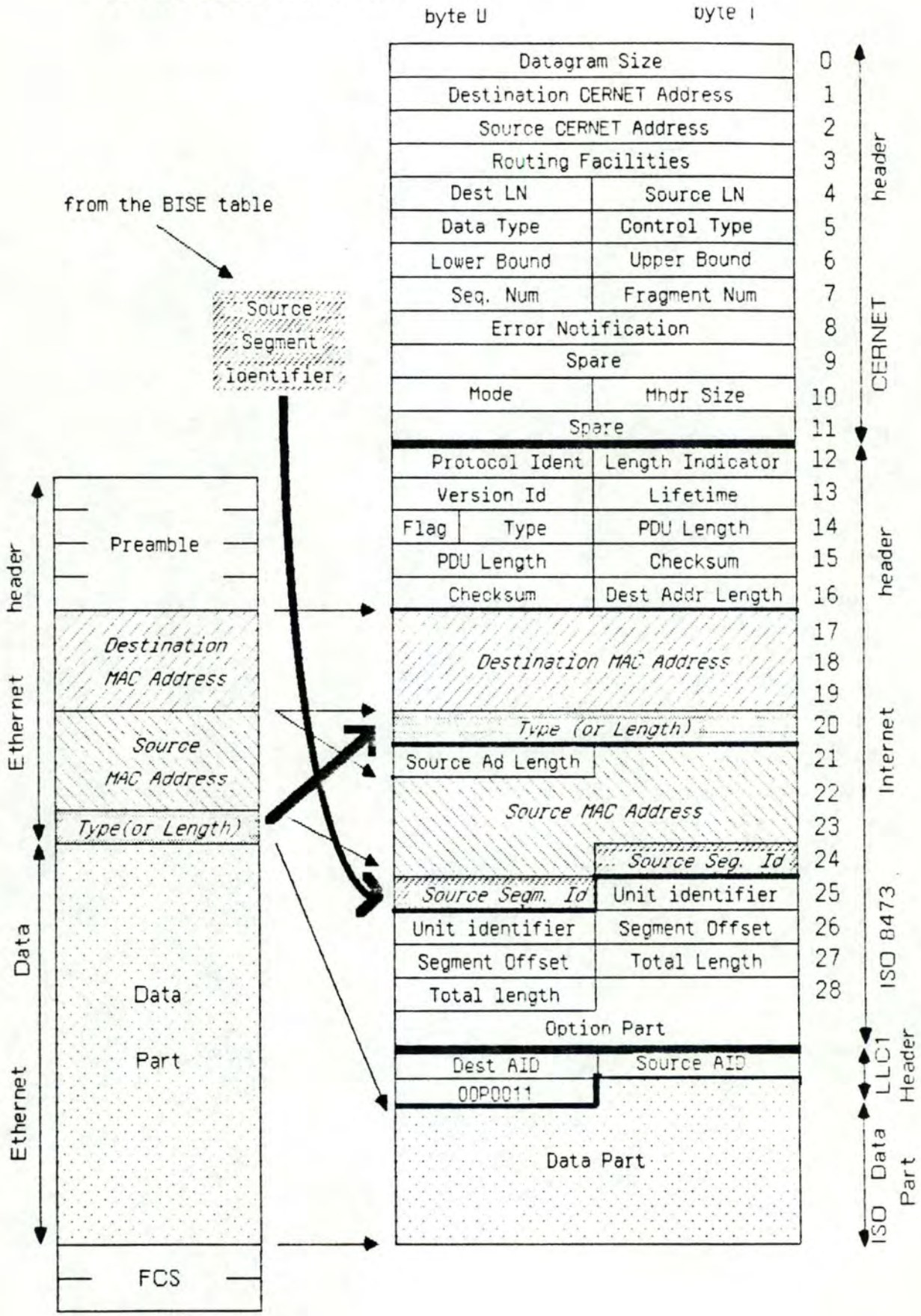


Figure 7.6: A full Frigate CERNET datagram

7.3.5 The broadcasting problem

The BISE table as described above is very easy to use when the destination address of a frame is found in the table. The segment identifier corresponding to a MAC address is automatically given in the table. When the search into the BISE table reveals an unknown address the only solution is to broadcast this frame, i.e. to forward it to all the other segments. When this happens, a new entry is created in the BISE table for this unknown MAC address, the segment identifier field of this new entry having its indicator bit set to 1. This broadcasting principle relies on the supposition that the unknown station to which the frame is addressed will necessarily reply. This reply frame, when arriving to the source Frigate will indicate the segment identifier on which the unknown station is located.

But this scheme has one major problem if the addressed station does not exist or does not reply. In practice, the source station will retry a certain number of times before abandoning. Each of these retries will be broadcast and will generate a number of Internet datagrams equal to the number of segments connected to the AID service (i.e. the number of Frigate Boxes). If the source station retries every second during 30 seconds, it will result in an inacceptable overloading in CERNET usage and in Frigate software processing.

This problem is solved in the following way: when the search in the BISE table reveals totally unknown address, the frame is broadcast and an entry is made in the table, corresponding to the MAC address. The indicator bit of the new segment identifier is set, indicating that sending for this destination should now be blocked. The remainder of this field contains the time at which this "block" may be removed. Any further retry attempted before the block-delay is elapsed will be ignored. If the delay is elapsed, the request is broadcast, and a new block-delay inserted in this field. The "block" is, of course, automatically overwritten when a reply from the previoulsy unknown destination arrives.

The procedure of multicast frame forwarding is designed for addressed stations which respond. It is ineffective in the case of communication protocols which are unacknowledged at all layers. Luckily, this is not the case for any of the protocols running on the CERN site.

7.3.6 Frigate basic logic

Having explained the functions of the AID service and its main features, we may now define the basic logic of Frigate [FRIG-P15] [FRIG-S6].

7.3.6.1 Frame received from the Ethernet segment

1. Check the BISE table for source address:

   (a) If the source address is unknown, create a new entry in the table with the segment identifier field containing the CERNET address of the Frigate. It indicates that this address is local, i.e. that this address can be accessed through this Frigate.

   (b) If the source address is found in the BISE table, ensure that it is known as local. If this is not the case, that means that this address was on another segment and has changed of location to become local to this segment. Update its entry in the table to indicate its new segment identifier.

2. Check the destination address:

   (a) If it is a broadcast or multicast address (i.e. that designates a set of destinations), checking the BISE table is not necessary. The frame has to be forwarded to all other Frigates.

   (b) Otherwise, check the BISE table for destination address:

      - If this address is not found in the table, it is unknown. A new entry is to be created for this address and it is to be "blocked" with the indicator bit set and a time-stamp in the rest of the segment identifier field. The frame is then broadcasted to all other Frigates (It is the only way to be sure that the destination will receive the frame).

      - If the destination address is found in the BISE table, check the segment identifier field of this entry:

         * If the indicator bit is not set, the destination address may be:
            - local (i.e. on the same segment than the source). In this case, the frame is discarded.

- remote : The frame is forwarded to the appropriate Frigate Box.

* Otherwise, the indicator bit is set, indicating that this address is unknown and has already been requested:
  - if the time-delay has elapsed, reset the time-stamp and broadcast the frame,
  - otherwise, discard the frame.

7.3.6.2 Datagram received from CERNET

The source Ethernet address contained in the source address field of the Internet header is known to reside on the segment whom segment identifier is given in the same field. For each datagram, check its Ethernet source address:

(a) If it is known in the BISE table, ensure that it matches with the source segment identifier given in the Internet header.

(b) If it is known and if the indicator bit is set, overwrite the segment identifier given in the Internet header. This action "unblocks" that Ethernet address.

(c) If it is unknown, create a new entry and fill it with the source Ethernet address and its segment identifier.

Then, create an Ethernet frame and send it on the segment. There is thus no checking of the Ethernet destination address on incoming CERNET datagrams. It would be possible to make use of this address to reduce, in rare cases, the load on the receiving segment, i.e. check the BISE table and not issue a frame whose destination address is known to be non-local.

Chapter 8 : THE IMPLEMENTATION

This chapter describes the Frigate Project as it is implemented. This implementation has two constituent parts: a hardware part and a software part. The hardware part is made of a CERNET interface, a Motorola M68000 processor and its memory, and an Ethernet interface. In the software part, we first describe the real-time multitasking operating system RMS68K running on the processor and then the program architecture. Each software module is specified and the interface between the operating system and the modules, and between modules, are defined and explained.

8.1 The Frigate Hardware

Because the Frigate Project is very specialized, it needs specialized hardware. It would be impossible to develop such a project on any commercially available microcomputer. Satisfying performances are a prerequisite to Frigate development because a too long delay in MAC frames forwarding will prevent higher layers to run. The time between frame sending and receiving may not be higher than the maximum propagation/transit delay time allowing timers in higher layers not to be modified. ISO recommends an upper bound of five seconds. The analysis of all IEEE 802 MAC Technologies leading to the choose of this upper bound may be found in [ISO3751].

As shown in the figure 8.1, the Frigate Box is based on, as far as possible, commercially available modules. It is built on a VME Bus, which is the CERN recommended standard for microprocessor-based applications. The Frigate is essentially composed of three modules: interfaces to each of the networks, plus a processor and memory module.

The VME bus has been chosen for its technical excellence allowing few boards with very powerful computing power and logic onboard and for the multi-master scheme governing the bus.

The CERNET interface module was, of necessity, fully developed at CERN. It provides all the functions necessary for the CERNET Basic Datalink Sublayer. It is fully compatible with the interfaces in the CERNET subscriber machines and in the nodes [FRIG-H3].

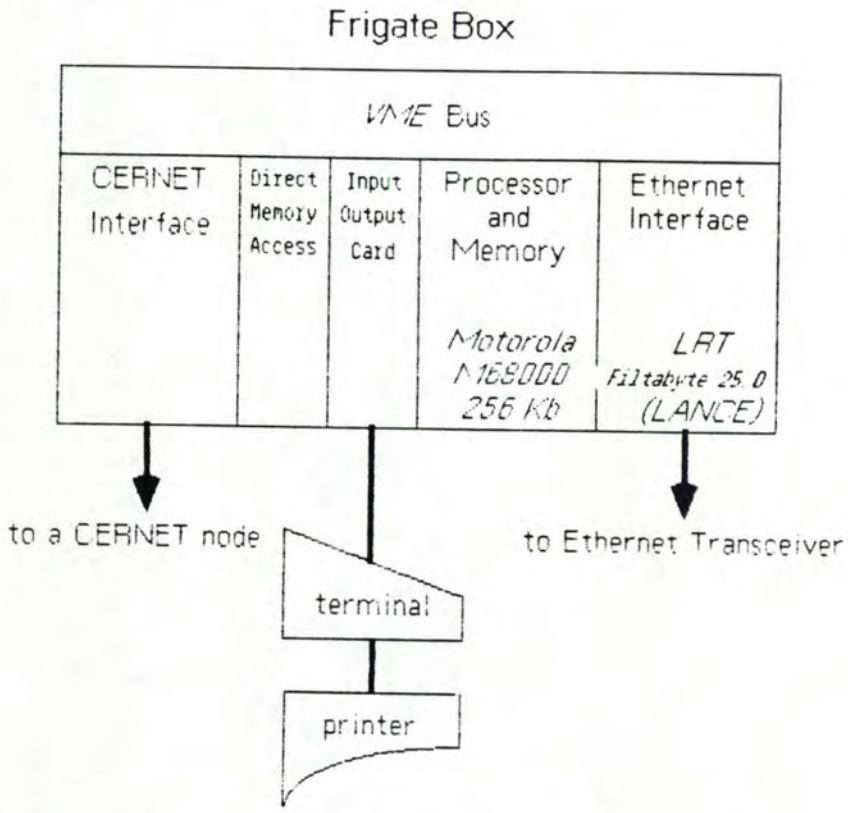


Figure 8.1: The Frigate Box Hardware

The processor is the well-known 16/32-bit Motorola M68000 [M68000]. This choice has been made in line with CERN recommendations. It is installed on a CPU2B card marketed by Force Computers Ltd. This card also supports 256 Kbytes of RAM memory and 32 Kbytes of EPROM memory.

When the project started, there was no VME-based Ethernet interface commercially available, so a development was undertaken to adapt a general Ethernet card, the Interlan NM10, to the VME environment. Since Ethernet cards are now available for VME Standard, one of these has been chosen and installed: the LRT Filtabyte 25.0 card. It is essentially based on the "Amd 7990 Local Area Network Controller for IEEE 802.3/Ethernet" (LANCE) developed by Advanced Micro Device Ltd. (AMD). Our work in the Frigate Project was concerned with the software modifications and rewriting consequent of the replacement of the Interlan NM10 board by the LRT board.

Additionally to these cards, a Direct Memory Access (DMA) card is used to accelerate the processing of the CERNET interface. An additional input/output allows the connection of a printer in parallel with the system terminal.

The modular hardware structure of Frigate allows an easy change of any of the two interfaces and its replacement by another interface. This modularity is also respected in the software design.

8.1.1 The Local Area Network Controller for Ethernet (LANCE)

This section is an overview of the features of the LANCE chip which is the main component of the LRT Filtabyte board [LANCE]. The study of the LRT board is of smaller interest because it only provides interfacing between LANCE signalling and VME Standard, and encoding and decoding of the serial bit stream to/from the transceiver [LRT]. The figure 8.2 is a detailed architecture scheme of the hardware Ethernet interface.

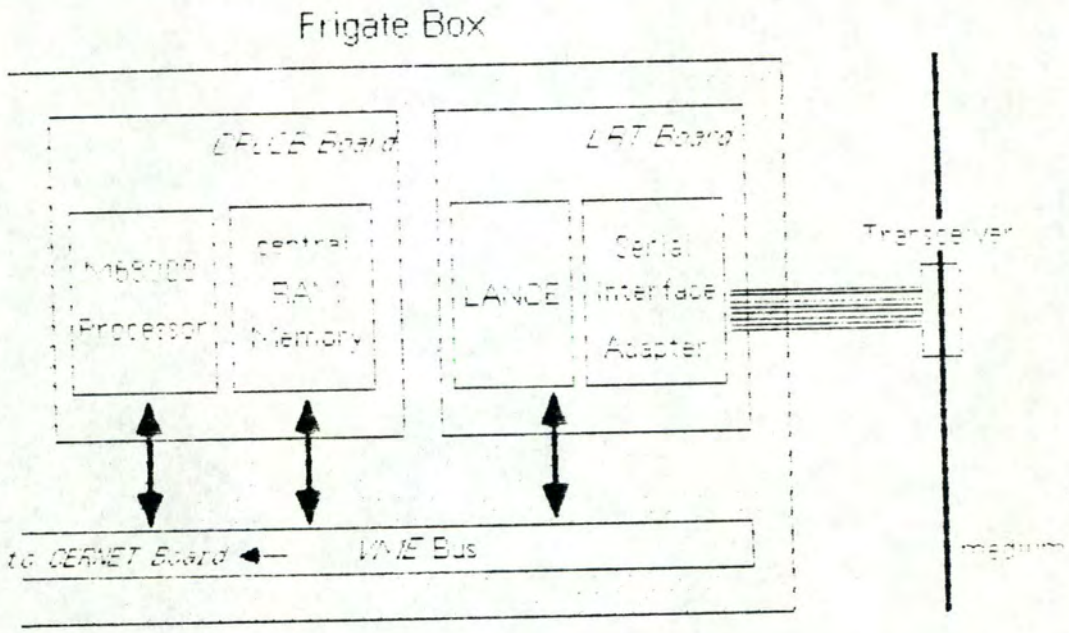


Figure 8.2: Architecture of the hardware Ethernet interface

The LANCE, along with the transceiver, provides all the 802.3 MAC Subalyer functions. The Ethernet CSMA/CD network access algorithm is implemented completely within the LANCE. Its main distinctive characteristics are the following [LANCE]:

- Ethernet and IEEE 802.3 compatible;
- Easily interfaced to 8086, 68000, Z8000 and LSI-11 microprocessors;
- Onboard Direct Memory Access and buffer management;
- 24-bits wide addressing;
- network and packet error reporting;
- diagnostic routines.

8.1.1.1 LANCE to M68000 Processor communication

The LANCE and the processor communicate one with each other in the following cases:

- for the LANCE chip initialization;
- when the processor asks the LANCE to send a frame;
- to know the result of the transmission;
- when the LANCE informs the processor of a frame arrival.

This communication is done in two ways: the LANCE may write onto or read specific memory fields, and it has four internal registers that the processor may read or write. The figure 8.3 shows the four 16-bits Control and Status Registers (CSR) resident within the chip. The CSR0 contains status bits indicating errors such as: the frame to transmit is too long; a frame has been missed due to buffer shortage; a memory error has occur on accessing the memory, etc. It also contains non-error status indications: a frame has been received, the frame transmission is finished, etc.
The CSR1 and CSR2 contain the address of the Initialisation Block. This block is a memory field fulfilled by the processor before starting up the LANCE and defining the chip's operating parameters (for example, promiscuous mode, disable the transmitter, perform internal loopback diagnostic test, ...)
The CSR3 contains status bits allowing the redefinition of bus interface mode (for example, "byte swapping" to swap the most and least significant byte of a word).

The Initialization Block contains, additionally to the operating parameters, the addresses and lengths of two descriptors rings, one for transmission and the second for reception. Each descriptor contains a pointer to a data memory buffer which is used to contain a frame. The buffer management is fully explained in the following section.

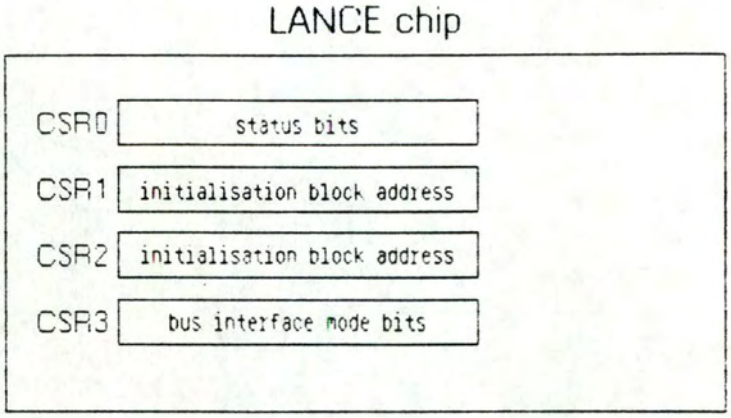


Figure 8.3: The Control and Status Registers of the LANCE

When the LANCE "has something to say to the processor", it modifies its internal CSR registers and/or some descriptors and, then, interrupts the processor. This latter may check the CSRs and the descriptors to "be informed of what did happen".

8.1.1.2 The LANCE memory management

In the LANCE, the buffer management is accomplished through message descriptors organized in ring structures in memory (see figure 8.4). There are two rings allocated for the LANCE: the Receive Ring and the Transmit Ring. The LANCE is capable of polling each ring for buffers to either empty or fill with frames to or from the channel. The LANCE is also capable of entering status information in the descriptor entry.

A descriptor is composed of :

- the data buffer address and length;
- the length of the data contained in the buffer (in reception descriptors only);
- status bits;
- one ownership bit which indicates who, from the processor or the LANCE, presently owns the descriptor and its buffer. Each device can only relinquish ownership of the descriptor entry to the other device. It can never take ownership, and each device cannot change the state of any field in an entry after it has relinquished ownership.

8.1.1.3 The LANCE operation

The M68000 processor initializes and starts the LANCE by setting some bits in the CSRs. The LANCE access to the first transmit descriptor. If it does not own this descriptor, nothing happens. When the processor gives this descriptor ownership, it means that there is in the first data buffer a frame to send. The LANCE transmits this frame onto the medium and set some bits in the descriptor to indicate the success or the failure of the transmission. It then gives the ownership of the descriptor to the processor and interrupts it. The processor may check the transmission result, while the LANCE automatically access to the following descriptor.

Because each ring may have up to 128 descriptors, it is possible to queue up to 128 frames awaiting their sending by the LANCE. This scheme has proved to be very powerful.

On reception, the operation is identical, except that the LANCE needs the ownership of at least one descriptor for being able to store in the data buffer the next incoming frame. It then stores in the descriptor the length of the frame, status and ownership bits, and then interrupts the processor. If the LANCE does not own the next descriptor entry when a frame arrives, it interrupts the processor to tell it that a frame have been missed

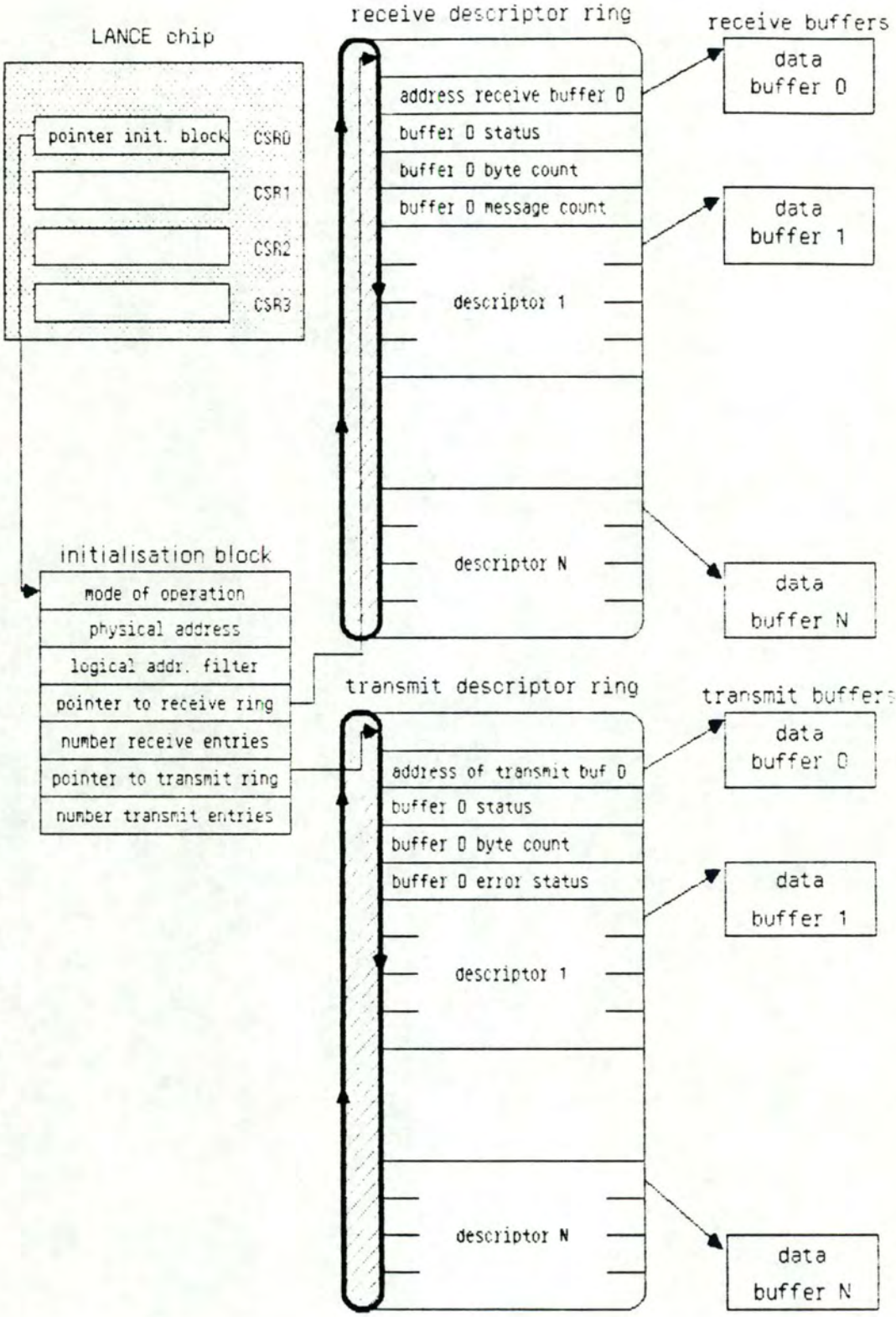


Figure 8.4: The LANCE memory management [LANCE]

because it does not own any receive buffer.

8.2 The real-time multitasking operating system RMS68K

RMS68K is the operating system developed by Motorola for its M68000 family of processors. This is a real-time and multitasking kernel around which real-time applications can be built. A task is a set of instructions being executed concurrently with another set of instructions. This mechanism is called the concurrent processing: the operating system can delay the completion of one task in order that another operation can be started, continued or completed.

8.2.1 RMS68K functions

RMS68K is composed of a task controller, an inter-task communication facility, an optional memory management facility and an initialization facility. These facilities allow RMS68K to provide the following functions [RMS68K]:

- receive all hardware and software interrupts and redirect them to the appropriate task;
- act as a dispatcher of tasks competing for use of the microprocessor unit;
- provide inter-task communication and synchronisation;
- manage and allocate memory;
- provide a system initialization capability;
- provide protection of the user environment;
- provide diagnostic feedback during error condition.

8.2.2 RMS68K structure

RMS68K is structured in six levels, with each level performing a particular range of functions [RMS68K]:

level 0: process management functions, including task dispatching, primitive synchronisation, exception handling and interrupt dispatching.

level 1: physical memory management functions.

level 2: utility functions.

level 3: task address space and memory management functions.

level 4: task creation, deletion and execution controlling functions.

level 5 (optional): physical input/ouput functions.

The functions provided by levels 3, 4 and 5 are directly available to user tasks. The other levels provide support to level 3, 4 and 5. User tasks can request RMS68K to perform a function by using executive directives. An executive directive contains all of the information needed by RMS68K to perform the desired function.

RMS68K is written in M68000 assembly language. Unfortunately, no interface or preprocessor has been designed to allow tasks written in the C-language to use RMS68K directives directly form the source code. They are called only by assembler instructions. This is a non-negligible programming inconvenience.

8.2.3 Task structure

A task is the basic processing unit in RMS68K. It can be described as a program performing a functional unit of work and its associated data areas. As illustrated in the figure 8.5, a task is made of many components.

The Task Control Block (TCB) contains information about the task which allows RMS68K to maintain control of the task's execution, account for resources allocated to the task, and ensure task protection. Task control is accomplished by RMS68K moving the task through various task states. A ready state task is dispatched into execution depending on its task priority and following an optional time-slicing principle.

The Asynchronous Service Queue (ASQ) is a circular FIFO queue which contains events to be processed by the task. An event is a way of communication between tasks or between RMS68K and a task. The general structure of an event is illustrated in figure 8.6. The two first bytes are always the length and the code of the event, whereas the format of the event content depends upon the event code. The code indicates the nature of the event. Some codes are defined by RMS68K (system events) and allow communication between RMS68K and the tasks. Other codes are reserved for user created events (user events) and allow a task-to-task communication.

The Program Code Segment of a task contains instructions used during execution. The main code is the basic element of a program code and is executed when the task is started. One or more Asynchronous Service Routines (ASR) are the event servicing code. An ASR is a routine which is executed on reception of an

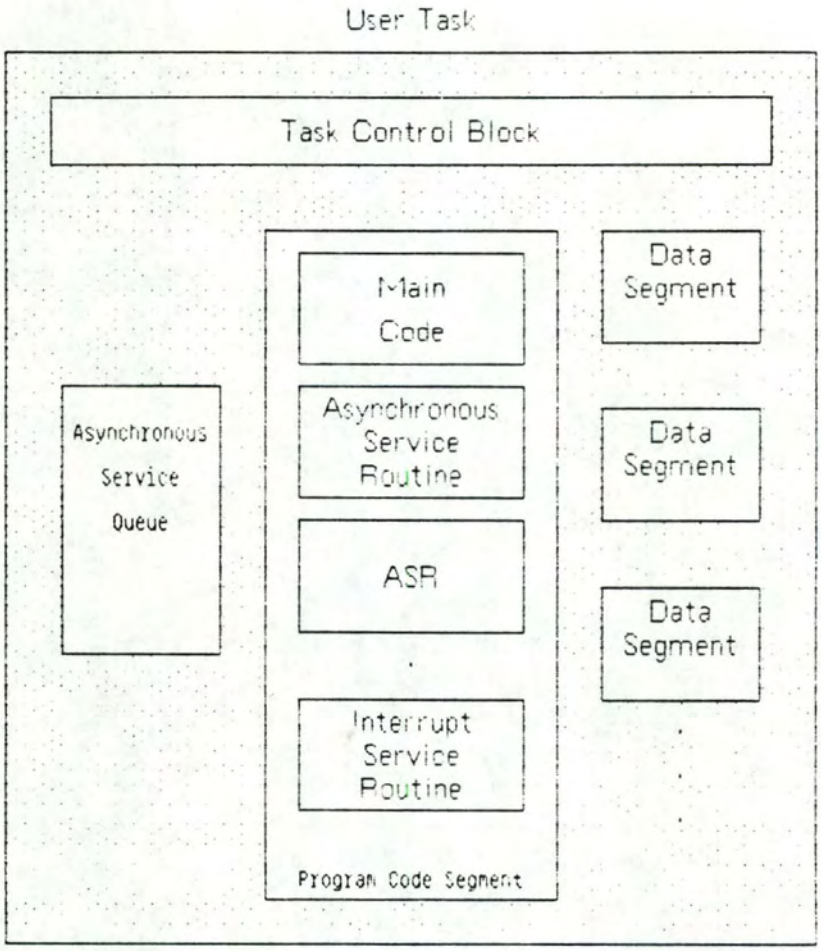


Figure 8.5: A user task under RMS68K

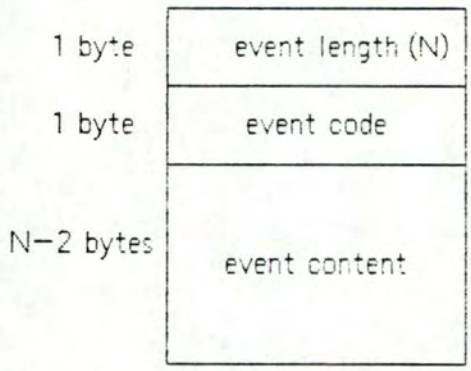


Figure 8.6: The general structure of an event

event. The task queuing the event can select an non-default

starting address for the ASR, thereby effectively creating several ASRs. An ASR can run simultaneously with the task, at the same software priority as that task, and only one ASR per task can ever be active. The reason for activation of the ASR is held in the event-code field (see figure 8.6). When an event is stored on the ASQ, the appropriate ASR is started and it executes concurrently with the task which generated the event.

A task may be "associated" with an external peripheral. In this case, the hardware interruptions generated by this peripheral are redirected to the task by the operating sytem. The Interrrupt Service Routine (ISR) is the code activated as the result of an external interrupt. The ISR is the highest priority code of the task and its activation suspends the main code and any ASR execution. The ISR mechanism is very useful in creating I/O device drivers, such as the Ethernet driver of the Frigate Software.

8.2.4 The Server task

The server notion allows the creation of user-defined directives and their use exactly in the same way as the RMS68K directives. A server task is able to receive and process requests from any other task in the system. These requests take the form of user-defined events and the server is in fact the ASR of the task declared as server. The distinctive characteristic of a server ASR is that its execution suspends the code of the calling task, i.e. the task which generated the server event. A normal ASR executes concurrently with other tasks.

8.3 The CWEB documentation utility

CWEB is a system of structured documentation for use with the C-language [CWEB]. Its phylosophy is that an experienced programmer needs to things: a language for formatting and a language for programming. CWEB allows to interleave documentation and program in a single source file. The CWEB utility program is then applied to this file and produces two things:

- A C-file to be submitted to the C-compiler,
- A formatted documentation file.

CWEB is based on the principle that a software program is made up of many interconnected pieces, called modules. These modules are organized in a "top-down" architecture, each module being defined by a set of lower-level modules, except these of the lowest level. Each CWEB module has two parts:

- A documentation part, containing explanatory material about what is going on into the module;
- A C-part, containing a piece of program.

In summary, the main features of CWEB are:

- Production of separated files for program and documentation;
- Ensuring that any diagnostics given at compile-time refers to the line numbers in the original CWEB file;
- Naming of modules of code;
- Incremental updating of modules;
- Numbering and cross-referencing of modules;
- Indexing of variables with respect of modules;
- Use of simple or sophisticated document formatting tools.

Whereas its module structure, CWEB still allows the use of all the C-language features, such as header files (for declarations), and routines. CWEB also checks for the definition of all the modules and for any recursive module definition. The advantages of CWEB are the following:

- The C-code structure is identical to the logical structure of the program;
- Lowest error risk in programming and implementation stages;
- Increased maintainability due to modularity;
- Negligible overhead in programming time for a sound investment in documentation.

8.4 The Frigate Software

This section describes and specifies the Frigate Software developed on the hardware described above to provide the AID service. The tools available to develop this software were, additionally to the RMS68K operating system and the CWEB documentation utility, a C-language cross compiler, a M68000 assembler ,and a utility allowing automatic compiling, assembling and linking of user tasks with RMS68K to build a loadable code file. All these tools run on a Digital Equipment Corp. VAX 7800 under Unix 4.2 BSD. At the beginning of the Frigate Project implementation, some parts of the software have been written in Pascal or Assembler language because the C-language tools were not yet available. They are now being rewritten in C-language.

Our work in this project was the designing and the writing of a new Ethernet driver for the LANCE chip.

8.4.1 The Frigate Software architecture

Figure 8.7 illustrates the tasks of the AID service. Theses are tasks for the RMS68K operating system.

The Dispatcher Task (DT) controls the operation of the two drivers and manages the overall bridging functions. Its main functions are:

- buffer allocation for CERNET input;
- buffer deallocation;
- BISE table maintenance;
- header manipulations as defined in 7.3.4;
- CERNET Datagram Layer functions;
- transit of packets between the two connected networks.

The Ethernet Task (ET) is the basic driver for the Ethernet. Its main functions are:

- buffer allocation for frames received from Ethernet;
- LRT interrupt handling.
- BISE table updating;
- destination identification;
- input frame filtering.

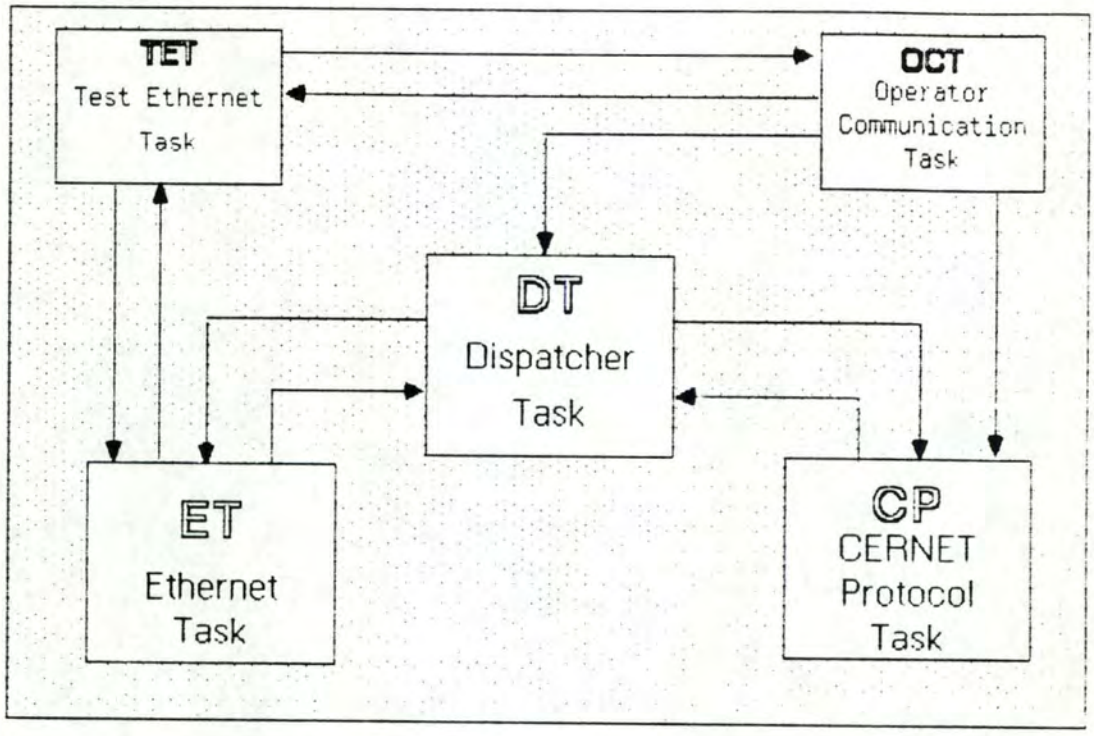


Figure 8.7: The Frigate Software architecture

It should be noted that the functions of BISE table updating, destination identification and packet filtering do not logically belong to this task. They are, however, carried out here to optimize the overall performance of the service. This is the case since a large proportion of received frames (90%) does not need to be forwarded and can therefore be rejected directly by the Ethernet Task, thereby avoiding unecessary task activations.

The CERNET Protocol Task (CP) is the CERNET driver. It deals with:

- datagrams input;
- datagrams output;
- CERNET interrupt handling.

The Operator Communication Task (OCT) manages the terminal connected to the Frigate Box and waits from inputs on this terminal. These inputs are processed to request actions by other tasks in the system. This task also performs some statistics on the overall software performance.

The Test Ethernet Task (TET) performs testing function of the Ethernet driver. It allows to run the ET task independently of the Dispatcher Task. Running ET in this testing mode is very useful for hardware interface and medium checking.

These two last tasks are mainly used for development purposes but are also included in running release of the Frigate Software.

8.4.2 Inter-task communication

As explained above, the mechanism provided by RMS68K for inter-task communication is the event. The programmer may define user events. A task communicates with another task in putting on its Asynchronous Service Queue (ASQ) a user event. To be able to process user events, two communicating tasks have to know the format of the user-defined event-block that they exchange.

Between the Frigate's tasks, a user event is defined. It is called the Data-Received Event-Block because its main use is for packet transfer between tasks. This event is used for the communication between all the tasks except for the communication DT->ET and TET->ET. This is due to the fact that ET is a server task. A request to a server task takes the form of a system event.

8.4.3 The Ethernet Task (ET)

The Ethernet Task is the task that we have fully rewritten to adapt it to the LANCE mode of operation. ET dialogs with DT or TET. This dialog is exclusive because ET communicates or with DT (in production mode), or with TET (in testing mode).

8.4.3.1 The ET interface

The ET interface has two sides. The first side is the interface in the sense ET->other task and the second is the opposite. In the sense ET->other task, the interface is defined by the Data-Received Event-Block. When ET receives a frame from the Ethernet segment, it builds a Data-Received Event-Block and queues this block on the ASQ of DT or TET. In the opposite sense, ET is activated via some commands, which are identified by command numbers. The commands performed by ET are the following:

1. Report and Reset statistics:
   These statistics can be displayed at the terminal and provide some information about the Ethernet physical interface and the Ethernet Task:

   - number of frames received,
   - number of frames sent,
   - number of frames missed due to buffer shortage,
   - number of multicast frames,
   - number of Frame Check Sequence errors,
   - number of late collisions,
   - number of "local" frames,
   - number of "remote" frames,
   - etc.

2. Report collision delay time: This delay time is valuable only when a frame transmission has failed due to collision after 16 attempts. It is the mean time between the beginning of the transmission and the collision detection. This is useful to detect a physical failure on the cable (Time Domain Reflectometry).

3. Send Data:
   This is the main command of ET. It is used to ask ET to send a frame on the Ethernet segment. When the transmission is finished, ET gives back to the calling task the status of the operation.

4. Reset:
   This command allows the reinitialization of the LANCE chip without having to restart the full Frigate Software.

5. Release Buffer:
When ET receives a frame, it allocates a buffer to store it. When this frame is to be forwarded, this buffer is given to DT (or TET) for further processing. When this processing is complete, ET receives the command "Release Buffer" which allows it to free this buffer and to re-use it for another frame.

6. Set Header Length::
The ET task performs buffering of incoming frames in a way that will allow DT and, later, CP, not to have to copy or shift the data part of the frame from one place to another. The same memory buffer is used for a frame passing from ET to CP and some place is let free at the beginning of this buffer as illustrated in the figure 8.8. This free place will be used by DT and CP for putting there the LLC1 header, the ISO 8473 Internet Header and the CERNET header.

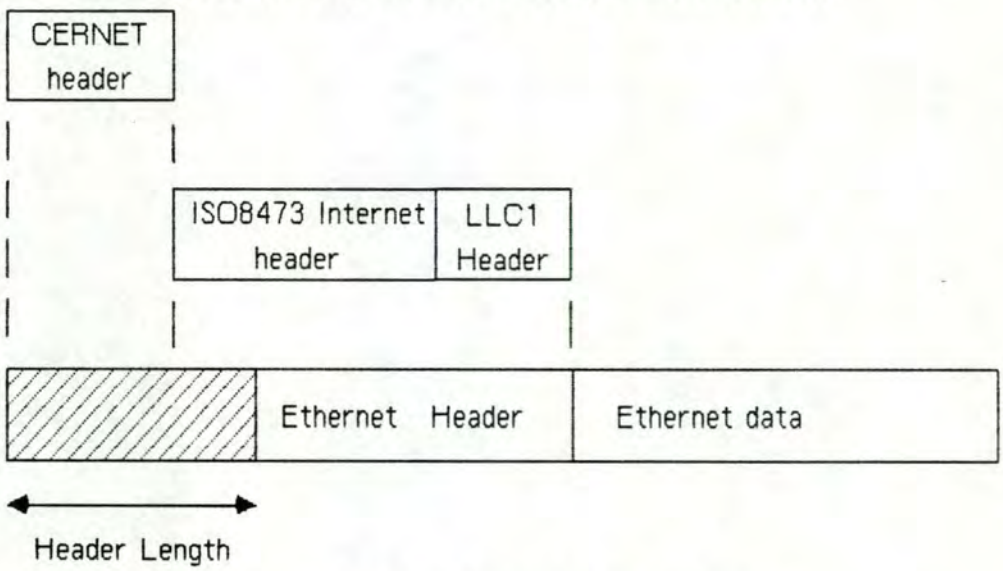


Figure 8.8: The Set Header Length command

The Set Header Length command is used only at the initialization of the Frigate software for that ET knows the place to leave free at the beginning of the buffers. This is needed for efficient header manipulation.

7. Set task to wake:
This command allows the switching between production and testing modes. It may be used from the terminal via OCT and TET. In testing mode, ET exclusively dialogs with TET, whereas it communicates with DT in production mode.

8.4.3.2 The ET architecture

The Ethernet Task software has two parts, the main part being written in C-language and the second part in the M68000 assembler language. The assembler part contains some assembler routines which are necessary for two main reasons:

- The RMS68K directives may not be called directly from the C-language. These routines are thus interfacing between the C-code and the RMS68K operating system.

- The C-routines may not be used as entry-points for Interrupt Service Routine (ISR) or Asynchronous Service Routine (ASR) because of incompatible stack manipulations automatically generated by the C-compiler.

Figure 8.9 shows the components of the C-part of the ET task. The initialization part performs the software initialization of the task and the hardware initialization of the LANCE chip. The command handling part is activated at each occurrence of a command to perform the desired action. The interrupt handling part is activated each time the LANCE generates a hardware interruption. The LANCE generates an interruption on frame input (receiver interrupt) or on frame output (transmission interrupt). the interrupt handler performs the interrupt recognition and takes the appropriate action.

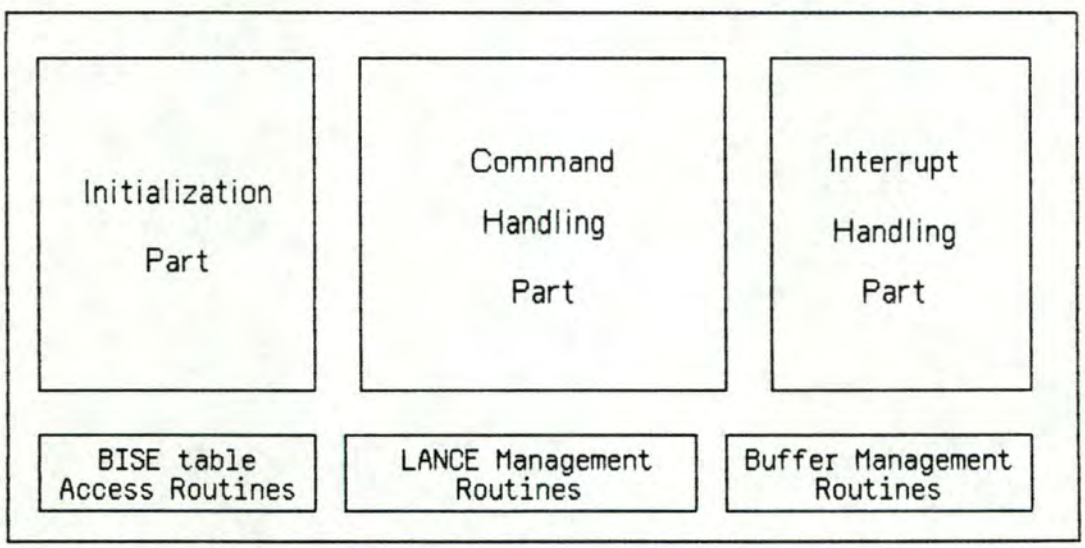


Figure 8.9: The ET architecture

Chapter 9 : EVALUATION

This chapter tries to evaluate the Frigate Project and our work in this project. The reasons of the hardware change leading to the rewriting of the Ethernet Task are explained. Some shortcomings of the LANCE and of the Ethernet Protocol are emphasized and a discussion of the software tools ends this chapter.

9.1 The hardware Ethernet interface replacement

At the beginning of the Frigate Project, the Interlan NM10 board was used as Ethernet interface. Its replacement by the LRT board was decided in June 1985 for the following reasons:

- Interlan no longer provides production and maintenance of the NM10 board.
- The NM10 board has no onboard Direct Memory Access (DMA), which is the case of the LRT board.
- The NM10 has no VME interface. Its installation in the Frigate Box has needed the development at CERN of an NM10-VME interface.
- When the NM10 works in promiscuous mode, it automatically receives the frame it is transmitting. This was a non-negligible processing overhead.
- When the NM10 receives a frame, this frame has to be fully transferred to the memory, even if no further processing on it is required.

9.2 LANCE shortcomings

The LANCE chip and its new Ethernet task has greatly increased the performance of the AID service. The number of frames local (not to be forwarded) or remote (to be forwarded) that the Frigate is able to process has at least doubled.

Whereas the memory buffer management of the LANCE has proved to be very powerful, a non-negligible shortcoming has appeared: There is no feature which allows the central processor to know the current descriptor used by the LANCE, both in the receive ring and in the transmit ring. This has entailed the following problem: If two frames are very "close" one to the other, their arrival will generates only one interrupt in the M68000. If this problems repeats, arriving frames are processed late by the Ethernet Task, which may cause troubles in higher-layer protocols. Once identified, this problem is avoidable by software, but a hardware pointer to the current descriptor would make this extra-processing unnecessary.

To solve this problem, we propose, rather than the pointer to the current descriptor, a register which would contains the number of received frames not yet processed. The LANCE increments this count register every time a frame is received and the software causes the hardware to decrement it at each frame processed. A received interrupt would correspond to a non-zero counter.

9.3 Ethernet shortcomings

The Ethernet Protocol allows easy MAC-Sublayer bridging. Manufacturers have developed Ethernet hardware interfaces able:

- To work in promiscuous mode;
- To generate frames with an arbitrary MAC source address;
- To generate frames with a Frame Check Sequence (FCS) computed by the software and appended to the frame.

These three conditions are mandatory to provide a full transparent MAC-Sublayer bridging service. Presently, the FCS transportation across CERNET is not implemented in the Friagte, whereas the hardware would allow it.

For the Ethernet Protocol, all stations have the same priority for accessing the medium. But because the Frigate station represents a great number of stations, it would be useful if it has a highest priority for accessing the medium. This has been realized in the Frigate Software in the following way: When a collision is detected on the first transmission attempt, an option of the operating mode of the LANCE allows us to be informed of this collision. In this case, the LANCE does not perform the backoff algorithm which defines the time to wait before attempting again. This allows the Ethernet Task to force the LANCE to immediately try again to transmit. In this way, the Frigate Box takes a higher priority in the implicit FIFO list represented by the backoff algorithm. This procedure violates the Ethernet Protocol definition but has proved to cause a non-negligible increase of the performance.

9.4 Software tools discussion

From the software tools available to develop the Frigate Software, some of them have shown to be useful and powerful, whereas others did not.

The CWEB documentation utility is very useful because the programmer has to think about his program in a modular manner. It emphasizes the different functionalities of a source code [PINEY86b].

The C cross-compiler was not optimized at all. The M68000 assembler code generation was redundant in some cases. Some crucial parts of the code have just been "manually" optimized between compilation and assembling.

The RMS68K version 4.3 was not satisfactory at all. Tests performed independently of the Frigate Software have shown that task-switching was too slow for a correct interrupt servicing. In the Frigate Software, some parts of the RMS68K code have been "short-circuited" because they were superfluous in the Frigate task-switching context. The version 4.4 recently installed has shown an important performance upgrade.

10 CONCLUSION

The network interconnection problem may be solved in many different ways. In the absolute, two networks may be interconnected at any layer. But, as in the ISO Reference Model for Open System Interconnection, low-layer interconnection is close to the hardware and dictated by medium constraints, whereas high-layer interconnection is close to the application and dictated by considerations of the application for which the intercommunication capability is used. Between these two extremes, the Network Layer is the most "natural" layer for network interconnection because it is not too much concerned with physical medium considerations and it is not yet application-dependent.

We have classified and studied in this work the four categories of interconnection architectures. Each of these, whereas being defined in regards to the layer concerned and the level (node or host) at which the interconnection is made, is designed for a specific practical use. The Bridge is used for the interconnection of many Local Area Networks of the IEEE 802 Project. The Gateway is designed for the interconnection of Wide Area Networks based on a packet-switching communication subnetwork. The Internetwork Protocol is undoubtedly the technique which may accomodate to the greatest diversity of networks, but with the drawback of its installation in each host's protocols. The Protocol Translator is fully specific to the application and the costs involved in its development are justified only if it greatly increases the number of users able to intercommunicate. These interconnection techniques are not exclusive and can be combined in order to create a huge set of interconnected networks.

Each of these techniques could be the subject of a detailed separate thesis. This thesis would integrate a full theoretical study of the technique and a certain number of implementations, either in commercial products, either for private use.

CONCLUSION

The Frigate Project is a typical example of a MAC-Sublayer bridging service, the routing function of this service being performed by CERNET. The Frigate implementation has shown the crucial importance of the hardware Ethernet interface and of the operating system on the overall performance. My work in this project was close to the Ethernet hardware interface and was constrained by a hardware-dependent inter-task interface. It is obvious that a well-defined abstract inter-task interface would have facilitate my integration in the Project and the designing and writing of the Ethernet Task. But this abstract interface may not decrease te overall Frigate's performance, which is a mandatory condition to its existence.

An improvement of the Frigate AID service can be to provide a backup capability for each Frigate Box, i.e. to install two Frigate Boxes on each Ethernet segment. The problems related to this new capability are relevant to the dialog between the two Frigates on the same segment, to avoid double forwarding of frames. A master-slave scheme or a processing partitionning based on two subsets of the BISE table can be adopted. This problem is presently studied at CERN [FRIG-S14] [FRIG-P19]

We think that the internetworking will take a more and more important place in any telecommunication problem because computer networks are designed to meet specific needs and because communication between users of any community of interest will become more and more necessary.

11 BIBLIOGRAPHY

[BENH83] Benhamou, E. and Estrin , J.:

Multilevel Internetworking Gateways: Architecture and Applications.
Computer, vol. 16 num. 9, September 1983, pp. 27-34.

[BLAKE84] Blake, J.:

Use of Microprocessor Cross Software under Vax Unix.
CERN-DD-JDB-jb, December 1984.

[BURG84] Burg, F., Chen, T.C. and Folts, H.C.:

Of Local Networks, Protocols and the OSI Reference Model.
Data Communications, November 1984, pp. 257-270.

[CALL83] Callon, R.:

Internetwork Protocol.
Proceedings of the IEEE, vol. 71 num. 12, December 1983, pp. 1388-1393.

[CCITTX75] CCITT:

Recommendation X75 : Terminal and Transit Call Control Procedures and Data Transfer System on International Circuits between Packet-Switched Data Networks.
CCITT, 1984.

[CERF74] Cerf, V. G. and Kahn, R. E.:

A Protocol for Packet Network Intercommunication.
IEEE Transactions on communication, May 1974, pp. 205-216.

[CERF78] Cerf, V. G. and Kristein, P. T.:

Issues in Packet-Network interconnection.
Proceedings of IEEE, vol. 66 num. 11, November 1978, pp. 1386-1408.

[CERF83] Cerf V. G. and Cain, E.:

The DoD Internet Architecture Model.
Computer Networks, vol. 7, October 1983, pp. 307-318.

[CERNET81] CERN DD Divsion/Communication Section :

CERNET - A High-Speed Packet-Switching Network.
Edited by J.M. Gerard.
CERN, Geneva, December 1981.

BIBLIOGRAPHY

[CWEB] Thimbledy, H. W.:

Literate Programming in C: Cweb Manual and Small Example.
University of York, UK, August 1984.

[DRI79] Driver, H., Hopewell, H. and Iaquints, T.:

How the Gateway Regulates Information Flow.
Data Communications, September 1979, pp. 61-70.

[ECMA82] European Computer Manufacturers Association:

Local Area Networks: Layer 1 to 4: Architecture and Protocols (TR/14).
ECMA, September 1982.

[ECMA84] European Computer Manufacturers Association:

Local Area Networks CSMA/CD Baseband: Link Layer.
ECMA, Second Edition, March 1984.

[ETH82] Digital Equipment Corp., Intel Corp and Xerox Corp.:

The Ethernet: A Local Area Network
Datalink Layer and Physical Specifications version 2.0.
November 1982.

[FLINT83] Flint, D. C.:

The Data Ring Main: An Introduction to Local Area Networks.
Wiley, 1983.

[FRIG-S6] Piney, C.:

Options for improving the performance of the Frigate AID Service.
CERN-DD/Frigate-S6, version 1, August 1984.

[FRIG-S7] Davids, D.:

A tool for compiling, assembling, linking and for keeping history-information on new versions and releases.
CERN-DD/Frigate-S7, version 2, October 1984.

[FRIG-S14] Piney, C.:

The algoritms for providing backup within the AID Service.
CERN-DD/Frigate-S14, version 1, December 1985.

[FRIG-H3] Anthonioz-Blanc, J., Brobecker, S. and Joosten, J.:

Specification of a VME-CERNET Interface.
CERN-DD/Frigate-H3, version 2, July 1983.

BIBLIOGRAPHY

[FRIG-P2] Piney, C. and Gerard, J.M.:

The Frigate Development Strategy.
CERN-DD/Frigate-P2, version 1, November 1983.

[FRIG-P12] Piney, C.:

Frigate usage of the Link and Network Layers.
CERN-DD/Frigate-P12, version 4, February 1985.

[FRIG-P15] Joosten, J. and Piney, C.:

Options and algorithms to resolve the correspondence between addresses and routes in interconnected Local Area Networks.
CERN-DD/Frigate-P15, version 5, June 1985.

[FRIG-R1] Piney, C.:

The Frigate AID Service system organisation and current status.
CERN-DD/Frigate-R1, version 2, June 1984.

[GIEN79] Gien, M. and Zimmerman, H.:

Design Principles For Network Interconnection.
Proceedings, Sixth Data Communication Symposium, INRIA, IEEE, May 1979, pp. 211-221.

[HAG82] Hagen, R.:

Interworking of Text Communications Services and Networks.
Computer Communications, vol. 5 num. 3, June 1982, pp. 115-118.

[HAW85] Hawe, B.:

Technology Implications in LAN Workload Characterization.
Workshop on Workload Characterization of Computers
(Pavia, Italy).
October 1985.

[HEARD83] Heard, K.S.:

Local Area Networks and the Practical Aspects of Internetworking.
Computer Networks, vol. 7, Jully 1983, pp. 343-348.

[HIND83] Hinden, R., Haverty, J. and Shelter, A.:

The DARPA Internet: Interconnecting Heterogeneous Computer Networks with Gateways.
Computer, vol. 16 num. 9, September 1983, pp. 38-48.

BIBLIOGRAPHY

[IEEE802.3] IEEE Computer Society:

Draft IEEE Standard 802.3: CSMA/CD Access Method and Physical Layer Specifications, Revision D.
IEEE Computer Society, December 1983.

[ISO3751] ISO / TC97 / SC6:

Forwarding of IEEE paper on LAN interconnection using MAC Sublayer bridges.
ISO, September 1985.

[ISO3781] ISO / TC97 / SC6:

Specification of Protocols to Provide the OSI Network Service, Part I: General Principle and Conformance.
ISO, September 1985.

[ISO3782] ISO / TC97 / SC6:

Specification of Protocols to Provide the OSI Network Service, Part II: Use of X25 Packet Level Protocol to Provide the Connection Mode Service.
ISO, September 1985.

[ISO3783] ISO / TC97 / SC6:

Specification of Protocols to Provide the OSI Network Service, Part III: Use of ISO 8473 to Provide the Connectionless Network Service.
ISO, September 1985.

[ISO3604] ISO / TC97 / SC6:

Final Text of DIS 8348: Information Processing Systems - Data Communications - Network Service Definition.
ISO, July 1985.

[ISO8473] ISO / TC97 / SC6:

Information Processing Systems - Data Communications - Protocol for Providing the Connectionless-mode Network Service.
ISO, January 1985.

[KEAR85] Kearsey, B.N. and Jones, W. T.:

International Standardisation in Telecommunications and Information Processing.
Electronics and Power, p. 643, September 1985.

[LANCE] Advanced Micro Devices, Inc.:

Local Area Network Controller Am7990 (LANCE): Technical Manual.
AMD, California, USA, June 1985.

BIBLIOGRAPHY

[LEVY84] Levy, D., Dam,.N. and Koiste, J.:

Protocol for packet/circuit switch interworking.
Computer Communications, vol. 7, num. 5, October 1984,
pp. 243-246.

[LLOYD] Lloyd, D. and Kirstein, T.:

Alternatives approaches to the interconnection of
computer networks.
in Communication Networks, pp. 499-518.

[LRT] L.R.T. Ltd.:

The Filtabyte 25.0 Installation Guide.
Reading Berks UK, January 1985.

[NGN34] Fluckiger, F. and Piney, C.:

Network General Note 34: Comparison between X25 and CERNET
architectures.
CERN-DD Division, August 1980.

[NPN04] Davies, H.E., Standley, M., Wiegandt, D. and Garnett, D.:

Network Project Note 04: The File Access Protocol.
CERN-DD Division, version 7, September 1983.

[NPN13] Davies, H.E. and Piney, C.:

Network Project Note 13: The Line Protocol.
CERN-DD Division, version 2, August 1979.

[NPN16] The network Team:

Network Project Note 16: The End-to-end Protocol.
CERN-DD Division, version 2, April 1983.

[NPN47] Gerard, M.:

Network Project Note 47: Network Computers, Topology and Routing.
CERN-DD Division, version 10, September 1980.

[NPN57] Fjellheim, R.:

Network Project Note 57: Network Control - Overview.
CERN-DD Division, version 2, April 1978.

[NPN69] Fjellheim, R. and Piney, C.:

Network Project Note 69: Packet Routing.
CERN-DD Division, version 4, April 1980.

BIBLIOGRAPHY

[LEVY84] Levy, D., Dam,.N. and Koiste, J.:

Protocol for packet/circuit switch interworking.
Computer Communications, vol. 7, num. 5, October 1984, pp. 243-246.

[LLOYD] Lloyd, D. and Kirstein, T.:

Alternatives approaches to the interconnection of computer networks.
in Communication Networks, pp. 499-518.

[LRT] L.R.T. Ltd.:

The Filtabyte 25.0 Installation Guide.
Reading Berks UK, January 1985.

[NGN34] Fluckiger, F. and Piney, C.:

Network General Note 34: Comparison between X25 and CERNET architectures.
CERN-DD Division, August 1980.

[NPN04] Davies, H.E., Standley, M., Wiegandt, D. and Garnett, D.:

Network Project Note 04: The File Access Protocol.
CERN-DD Division, version 7, September 1983.

[NPN13] Davies, H.E. and Piney, C.:

Network Project Note 13: The Line Protocol.
CERN-DD Division, version 2, August 1979.

[NPN16] The network Team:

Network Project Note 16: The End-to-end Protocol.
CERN-DD Division, version 2, April 1983.

[NPN47] Gerard, M.:

Network Project Note 47: Network Computers, Topology and Routing.
CERN-DD Division, version 10, September 1980.

[NPN57] Fjellheim, R.:

Network Project Note 57: Network Control - Overview.
CERN-DD Division, version 2, April 1978.

[NPN69] Fjellheim, R. and Piney, C.:

Network Project Note 69: Packet Routing.
CERN-DD Division, version 4, April 1980.

BIBLIOGRAPHY

[NPN78] Watts, R.P.:

Network Project Note 78: A Virtual Terminal Protocol for CERNET.
CERN-DD Division, version 1, December 1978.

[NPN98] Gerard, J.M.:

Network Project Note 98: The CERNET Transport Manager.
CERN-DD Division, version 4, August 1982.

[M68000] Motorola Semiconductors:

M68000 16/32-bit microprocessor: Programmer's Reference Manual.
Motorola Inc., fourth edition, 1984.

[PINEY79] Piney, C.,Parkman, C. and Fluckiger, F.:

Lessons from an Endpoint Interconnection of High-Speed
Local Packet Networks.
CERN, December 1979.

[PINEY86] Piney, C.:

Bridges provide third generation networking.
CERN, January 1986.

[PINEY86b]: Piney, C.:

Lessons learned in using Cweb for programming in C.
CERN, CJP-GEN/N22, version 2, June 1986.

[POST80] Postel, J.B.:

Internet Protocol Approaches.
IEEE Transactions on Communications, vol. 28, num. 4,
April 1980, pp. 223-229.

[POST81] Postel, J., Sunshine, C. and Cihen, D.:

The ARPA Internet Protocol.
Computer Networks, vol. 5, num. 4, July 1981, pp. 261-271.

[RED83] Redell, D. and White, J.:

Interconnecting Electronic Mail Systems.
Computer, vol. 16, num. 9, September 1983, pp. 55-63.

[RMS68K] Motorola Inc.:

M68000-Family real-Time Multitasking Software User's Manual.
Eight edition, March 1984.

BIBLIOGRAPHY

[ROB82] Robinson, P.:

Protocol Converter: the answer to compatibility problems ?
Computer Communications, vol. 5, num. 3, June 1982, pp. 148-150.

[SCHIN82] Schindler, S.:

Keywords in communication technology.
Computer Communications, vol. 5, num. 3, June 1982, pp. 140-147.

[SCHNEI83] Schneidewind, N. F.:

Interconnecting Local Networks to long-distance Networks.
Computer, vol. 16, num. 9, September 1983, pp. 15-24.

[SHEL82] Sheltzer, A., Hinden, R. and Brescia, M.:

Connecting different types of Networks with gateways.
Data Communications, August 1982, pp. 111-122.

[SHO79] Shoch, J.:

Packet Fragmentation in Inter-Network Protocols.
Computer Networks, vol. 3, February 1979, pp. 3-8.

[STALL83] Stallings, W.:

Beyond Local Networks.
Datamation, August 1983, pp. 167-176.

[STALL84] Stallings, W.:

Local Area Networks: An Introduction.
Macmillan, New-York, 1984.

[STALL85] Stallings, W.:

Data and Computer Communications.
Macmillan, New-york, 1985.

[STRUI84] Struif, B.:

Transparent LANs and LANs as OSI subnetwork.
Computer Communications, vol. 7, num. 6, December 1984,
pp. 296-300.

[SUN77] Sunshine, C.:

Interconnection of computer networks.
Computer networks, vol. 1, January 1977, pp. 175-195.

BIBLIOGRAPHY

[TAN82] Tanenbaum, A. S.:

Computer Networks.
Prentice Hall Int., 1982.

[TCP/IP] Defense Advanced Research Projects Agency:

DoD Standard: Internet Protocol
Transmission Control Protocol.
Information Sciences Institute, California, January 1980.

[THORN83] Thornton, J. and Christensen, G.:

Hyperchannel Network Links.
Computer, vol. 16, num. 9, September 1983, pp. 50-54.

[UNS82] Unsoy, H.:

X75 Internetworking of Datapac with other packet switched networks.
Journal of telecommunication Network, vol. 1, num. 3,
Fall 1982, pp. 243-255.

[VANB86] Van Bastelaer, Ph.:

Notes pour un cours de Téléinformatique:
interconnection de réseaux.
FNDP, Namur, 1986.

[WAR83] Ware, Christine:

The OSI Network Layer: Standards to cope with the real world.
Proceedings of IEEE, vol. 71, num. 12, December 1983.

[ZIM83] Zimmerman, H. and Day, J.:

The OSI Reference Model.
Proceedings of the IEEE, vol. 71, num. 12, December 1983.

12 LIST OF FIGURES

LIST OF FIGURES

TABLE OF CONTENTS

13 TABLE OF CONTENTS

TABLE OF CONTENTS

TABLE OF CONTENTS