CANADIAN THESES

THÈSES CANADIENNES

NOTICE

The quality of this microfiche is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Previously copyrighted materials (journal articles, published tests, etc.) are not filmed.

Reproduction in full or in part of this film is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30.

AVIS

La qualité de cette microfiche dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

Les documents qui font déjà l'objet d'un droit d'auteur (articles de revue, examens publiés, etc.) ne sont pas microfilmés.

La reproduction, même partielle, de ce microfilm est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30.

THIS DISSERTATION HAS BEEN MICROFILMED EXACTLY AS RECEIVED

LA THÈSE A ÉTÉ MICROFILMÉE TELLE QUE NOUS L'AVONS REÇUE

Canada

Enhancing CUENET for Ease of Growth and Possibility of Internetworking

Alfred Kwan-Cheuk Yu

A Major Report

in

The Department

of

Computer Science

Presented in Partial Fulfillment of the Requirements
for the Degree of Master of Computer Science at
Concordia University
Montréal, Québec, Canada

April 1986

ISBN 0-315-30631-9

ABSTRACT

Enhancing CUENET for Ease of Growth
and Possibility of Internetworking

Alfred Kwan-Cheuk Yu

A packet-switched protocol adapted from CCITT Recommendation X.25 is proposed for use in the Concordia University Educational NETwork (CUENET). It facilitates CUENET to reach out to other computer communcation networks and public data communication networks through the use of a gateway which can be implemented with any of the processors within CUENET. It also enhances CUENET to comply with the principles of the Open Systems Interconnection (OSI) reference model of ISO. It is projected that the throughput of the system bus, C-bus, will be reduced by implementing this new protocol. However, ways of increasing the throughput of C-bus have been examined and proposed for implementation.

Existing hardware and protocol have been retained as far as possible. The proposed modifications to the system architecture will enable CUENET to implement as much as possible the communcation protocol handling functions in off-the-shelf link and/or packet level electronic circuit boards or VLSI chips as and when they are available and economical.

To my wife,

Mary,

the greatest wife.

To my daughters,

Miranda and Celeste,

the sweetest little girls.

ACKNOWLEDGEMENTS

Particular thanks are due to Dr. J.W. Atwood who gave advice and guidance in the preparation of this report.

Thanks also go to Mr. C. Grossner who so kindly explained to the author the details of the design of CUENET.

The review efforts and valuable suggestions by Dr. T. Radhakrishnan, Mr. J. Olizar and my wife, Mary, are very much appreciated.

TABLE OF CONTENTS

LIST OF FIGURES AND TABLES

FIGURES

TABLES

CHAPTER I

INTRODUCTION

1.1 Computer Communications: The Past Decade

In the past decade, there have been great advances in the field of telecommunications. New data communication networks such as Datapac and Infonet in Canada, Telenet and Tymnet in the United States, Transpac in France, International Packet Switched Service (IPSS) in the United Kingdom and Venus in Japan, have been implemented over the past few years. Through the use of packet switching technology, these new data communication networks are offering better network performance in terms of low error rate, fast call set-up time, shorter network transit time, improved flow control and lower cost. All these are the ingredients which are required to foster the implementation of distributed systems over large geographical areas.

In parallel to the advances in telecommunications, new technologies have brought about lower computer system costs and resulted in lower computation costs. To achieve further economies, huge data bases can be segmented, or replicated among a number of interconnected data base systems. Sharing of processing and information resources among different systems can be accomplished by message exchanges which take the form of commands, inquiries, responses, file transfers, etc.

Data processing costs have in recent years declined faster than communication costs. In general, communication cost has been found to be more expensive than computing cost. Thus, the topology of a present day distributed computing system is very much dependent on communication costs [Adiba 80]. The advent of fiber optics and private satellites would accelerate the decrease of communication costs. This trend has been a major factor in stimulating the implementation of more distributed systems.

1.2 Distributed Systems Research at Concordia University

A group in the Computer Science Department of Concordia University has been actively conducting research on the decomposition of algorithms for parallel processing in the area of distributed processing. CUENET (acronym for Concordia University Educational NETwork) was constructed as a configurable network of microcomputers to aid the research. Depending on the user's requirement, the processors within CUENET can assume various architectural configurations under software control. Given the basic services provided by C-bus, it is intended to enhance CUENET to support distributed operating systems, distributed databases and to connect to a host computer which has the intelligence to segment an application program for concurrent processing.

1.3 Scope and Outline

As expected in any distributed system, there is a definite need for some processes within CUENET to communicate with other processes residing outside of CUENET. The communication protocol which CUENET uses to communicate outside its own environment has a great impact on the performance of the communication channel in terms of channel throughput, response time, etc. This in turn impacts on the viability and effectiveness of a distributed system.

The scope of this report is to examine the communication requirements of distributed computing systems, their impact on the communication protocol, various communication protocol design issues and then to propose a suitable protocol for use within CUENET. The proposed protocol is designed to enhance CUENET for ease of growth and to enable it to reach out to other local area networks and long haul public networks.

In designing the proposed protocol, all efforts have been made to keep the existing CUENET hardware and protocol as far as possible while attempts are made to facilitate the internetworking of CUENET.

In this report, much of the discussion is centred around the ISO Open Systems Interconnection (OSI) reference model, local area network standards and transport class of service. Appendices A, B and C are prepared to provide

background information on those subject areas. They should be read in conjunction with the main text of this report. A detailed description of the OSI reference model can be found in [ISO 83]. Various articles such as [White 80], [Canes 83], [McGov 80], [Moult 80], [Piatk 80], [Stall 84], [Stall 85] and [TG100 84] presented overviews of the model and discussions on some particular layers of the model.

1.4 Terminology

In this report, the term "protocol" means a set of rules and/or codes for the exchange of information between two or more components in the context of computer and data communications.

Throughout this report, CUENET is used to denote CUENET in its current form. CUENET-2 is used to denote the new CUENET which uses the new communication protocol and architecture proposed herein.

For some time, the word "network" had been generally used as a collection of physical equipment and/or transmission facilities. Since the introduction of the OSI reference model, a "network" is defined (in OSI terminology) as a collection of distributed entities working together, through the use of a common set of protocols, to provide a number of services, including the determination of a path or route for the communication users. To avoid confusion, the term "subnetwork" is used for a collection of physical equipment and/or transmission facilities.

CHAPTER II

CUENET - A LOCAL COMPUTER NETWORK AT CONCORDIA UNIVERSITY

2.1 CUENET Overview

CUENET is the acronym for Concordia University Educational NETwork. It is a reconfigurable network of loosely coupled microcomputers. It has been used by the Computer Science Department in research on decomposition of algorithms for parallel processing. Figure 2.1 is a block diagram of CUENET in its present form.

CUENET is based on a time shared bus called C-bus. Under software control, it can assume various architectural configurations at the processor, memory, switch (PMS) level in order to match it with the nature of the problem being solved. Furthermore, CUENET offers two unique features. C-bus can disallow a user program from sending a message to a computer for which it has no access rights. This is achieved through the use of an access vector control mechanism. The other feature is to use lock and key registers to prevent C-bus from passing illegal messages generated by unauthorized software.

Simulation results [Gross 82] indicated that a C-bus under the control of a microprocessor based controller is capable of supporting between six and ten computers. Its estimated maximum data transfer rate is 1.6 Mbit/s.

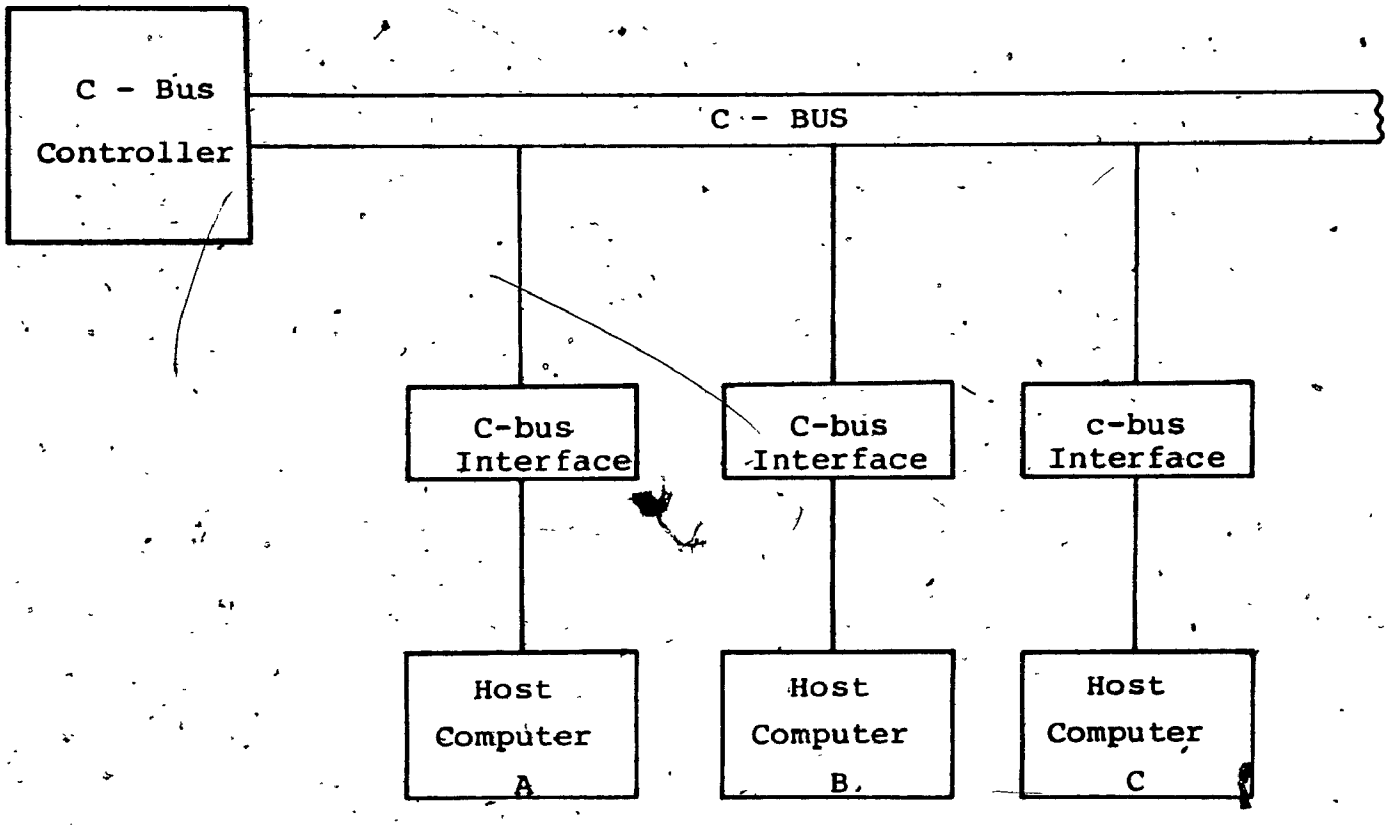


Figure 2.1

Block Diagram of CUENET

There are three types of functional units attached to C-bus: a master processor, several slave processors and network memory units (NMU). The master processor runs the operating system of CUENET. It responds to a user's request for computational resources and sets up the configuration to fulfill the demand. The tasks submitted by the end user are carried out by the slave processors. The master processor is responsible for coordinating the other processors of CUENET. The decomposing of an application program into tasks is the responsibility of the user. However, such decomposition will be aided by a processor within CUENET in the future.

2.1.1 The C-bus

The C-bus is composed of twisted pairs of wires. It is 50 feet long in its present form. There are a total of 49 signal lines for data transmission, addressing and control purposes. Attached to one end of the C-bus is the C-bus controller which oversees the operation of the bus.

C-bus provides a high speed communication medium through which microcomputers can transfer information among themselves in the form of messages on a time sharing basis. All computers connected to the C-bus are daisy chained together.

In its present operation, CUENET has no message out of sequence problem as there is only a single communications

path and only one message is allowed to flow at any one time.

2.1.2 The C-bus Controller

The C-bus controller oversees the operation of the C-bus. It functions as a mailman guaranteeing the safe delivery of messages from one computer to another. For the experimental prototype, it was implemented with a general purpose microprocessor which may be replaced in future by a bit slice processor to permit improved performance.

Information is sent via C-bus in the form of messages. Once a host computer has placed its message in the output buffer of its associated C-bus interface, the C-bus controller will control the transfer of the message to the input buffer of its destination computer. The C-bus controller can also monitor the message flow and collect statistics. The sending computer is free to perform other tasks once its message has been deposited into its output buffer.

An asynchronous handshaking scheme is used for the selection of the next sender to be serviced. Once the sender and receiver are established, the entire message is transmited using a synchronous protocol.

The parity control hardware is monitored by the C-bus controller arbitration and control module to detect the occurrence of a parity error. When such an error is

detected, the last message is retransmitted.

Contention between the C-bus controller and the host computer for the use of the C-bus interface is resolved through the use of an interface status and control block.

2.1.3 The C-bus Interface

Each host computer is connected to the C-bus via its associated C-bus interface. A host computer can deposit a message in its output buffer of the C-bus interface while the bus controller delivers a message to the input buffer. The processors are not directly involved with the transfer of messages. As a result, the speed of C-bus is not limited by the speed of the processors. Also, the delay due to contention between the processor and the C-bus controller for memory access is eliminated. Therefore, C-bus can suport a wide variety of heterogeneous microcomputers without being affected by the speed of each individual computer.

Each C-bus interface contains an access vector which is a hardware table which can only be read by a host computer. One computer connected to C-bus with an authorization from the C-bus controller can become the master computer which has the authority to configure the intercomputer communication topology by setting the access vectors of the other computers. The setting of a host computer's access vector will determine the computers with whom that computer

can communicate. This is one mechanism used in CUENET to guarantee the integrity of the interconnection topology.

The interface lock control unit contains two registers called lock and key registers. The contents of the lock register is set to a fixed predetermined value, called a combination. A software process will be allowed to access the interface if it unlocks the interface by writing the combination into the key register. When the contents of these two registers match, the interface will be unlocked.

The address decoders and control units are used by the host computers and the C-bus controller to access the various interface hardware modules.

The daisy chain logic module enables the interface to identify itself to the C-bus controller when the output buffer contains a message to be sent.

As the control of C-bus is centralized at the C-bus controller, the C-bus interface has been significantly simplified and has become more cost effective.

2.1.4 Master Processor

Any computer in the network can function as the master computer. If the CUENET master computer fails, the down time of CUENET is reduced to the minimum as the master software can be quickly loaded into any other computer and CUENET can be restarted promptly.

2.1.5 Slave Processors

The slave processors of CUENET need not be homogeneous. Each slave processor is assigned a Physical Processor Number (PPN). The user must indicate the primary physical properties (such as storage requirements, special purpose computing facility like an arithmetic or fast fourier transform) which will be required to execute his algorithm. The user assigns a Logical Processor Number (LPN) for each processor he specifies. Mappings between the LPNs and PPNs are carried out by the operating system to match the processor capabilities requested by the user with the computers that are available within CUENET.

2.1.6 Centralized Storage

In CUENET, there can be one or more Network Memory Units (NMUs) to form the central storage which can be accessed and shared by all host computers. Each NMU is associated with a microprocessor based controller to encode and decode messages. Other functions such as synchronization control, searching local data for specific elements and sending the results to other processes, and arbitration between simultaneous read and write requests for the shared data can also be implemented on this intelligent NMU controller.

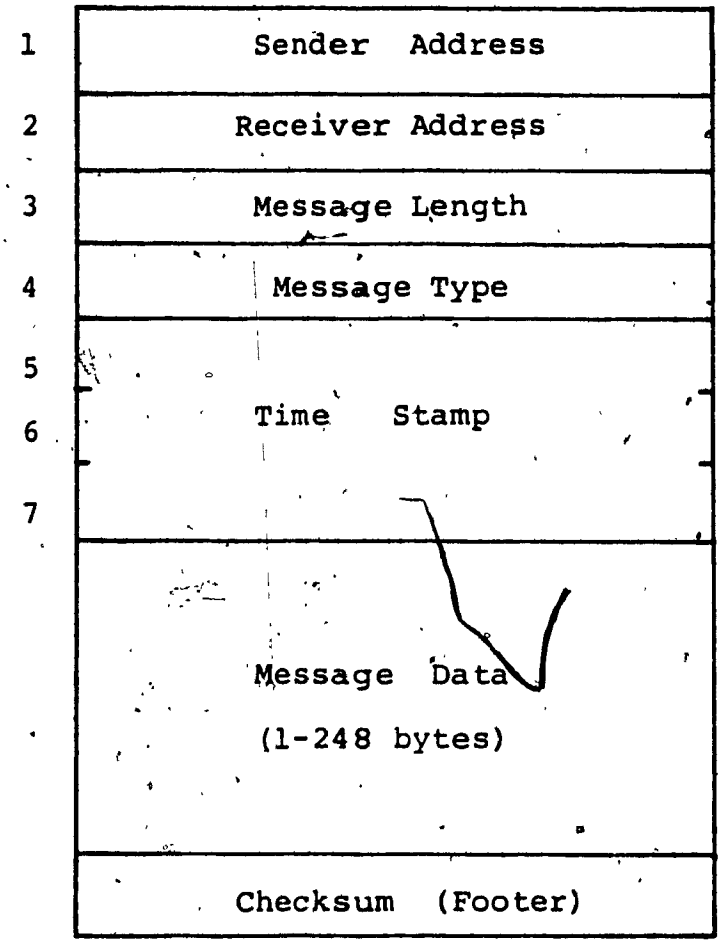


Figure 2.2

CUENET Message Format

2.2 Message Communications

The CUENET message format is illustrated in Figure 2.2. It consists of three sections: message header, body, and footer. The ordered pair <sender address, time stamp > will uniquely identify each message passed in CUENET. The time stamp in the message header indicates the time at which the message was mailed by the C-bus controller. The addresses specified in a message will correspond to physical processor numbers (PPNs). The message body may vary in length, but the overall message length should fit into the output and input buffers of the sender's and receiver's interface respectively. The message footer will be used as a checksum.

The verification of transmitted messages is performed in a two step process involving the C-bus controller and the message receive software. During transmission, any error detected by the hardware parity checker will cause a retransmission of the message. The second verification step is performed during message decoding. If a checksum error occurs, a special message is sent to the originator of the message requesting a retransmission. Software in each computer maintains a backlog of the last "k" messages it has transmitted. If the message requested for retransmission is found in the backlog it is sent a second time, otherwise the master computer is notified that an error has occurred. The action taken by the master computer upon receipt of the

error messages is basically to inform the user of the errors. The end user will have to decide what further actions to take.

There are three message types. They are: intercomputer messages, access vector loading messages and error messages. Provision for the introduction of new message types has been made.

When a user task requests a message transfer, it must supply the LPN of the receiver, the starting address of the location of the message and the message length. The message transmit routines will then be invoked to map the LPN of the receiver address to its PPN. If the message is too long, it is subdivided into multiple smaller messages which will be numbered, for example, as 1 of 3, 2 of 3, etc.

The message receive procedure is invoked by the scheduler when the receive queue is not empty. If the message is a retransmission request, the appropriate message is retrieved and retransmitted. If the message is not a retransmission request then the checksum is verified. If checksum error is detected, a retransmission request message is placed into the send queue, else the message header and footer are removed and the message body is placed into the decoded message buffer. An entry in the message log is then made. The PPN contained in the message header will be mapped to an LPN.

No message can be delivered to a computer if its input buffer is occupied. The interrupt handler (of the host computer) is activated when either the input or output buffers of the C-bus interface need servicing. When invoked, the interrupt handler will determine if the interrupt is due to an empty output buffer or a full input buffer. If the interrupt is due to a full input buffer, the message is copied into the receive queue. The C-bus controller is then notified that another message transfer to this computer can be performed. If the interrupt is due to an empty output buffer, the next message in the send queue is placed into the output buffer. The status register is then set to indicate to the C-bus controller that a message is waiting for transmission.

When the C-bus controller is free, it signals the interface units through the "bus-grant" line. All the interfaces on a bus are daisy chained with respect to the bus grant line. A mask bit is provided at each interface unit for selective setting or resetting by the bus controller. If the mask bit of an interface unit is set, that unit would not respond to the bus-grant signal. When the interface unit of the requesting processor receives a bus-grant signal, it stops its propagation to the following processors. It puts its address on the interprocessor data bus to identify itself to the bus controller. Then, the bus controller reads the receiver address from the header of the message which is in the output buffer of the sender. If the

input buffer of the receiver processor is free, the message is transferred. Otherwise, the bus controller will set the mask bit on the sender's interface unit and enable the bus grant line to propagate further, so that other processors may send their messages. After transmitting a message, the controller checks the validity of the transmission via the parity bits supplied by the interface units. If an error is found, retransmission is requested before relinquishing the bus from that particular interface unit. After a certain number of trials, if no successful transmission is possible, the exception condition is reported to the master processor.

If a transmission is successful, the controller marks the output buffer of the sender "empty" and the input buffer of the receiver "full".

2.3 Analysis of the CUENET Messaging System

CUENET was not designed in accordance with the OSI reference model. In the following analysis of the CUENET messaging protocol, attempts are made to classify its protocol functionalities according to the OSI reference model with the "best fit" approach.

2.3.1 Physical Interface - Level 1

In CUENET, the physical interface consists of :

16 wires for Address A

16 wires for Address B

9 wires for Data and Parity

8 wires for control and bus arbitration

49 wires in total

The C-bus controller would initiate a retransmissiopn of a message if parity error is detected by the hardware parity checker.

2.3.2 Link Level Interface - Level 2

The communications link is assumed to be available and ready at any time. There are no link level control functions at all except that there is a checksum mechanism.

CUENET was planned to support more than one C-bus. This implies the support of multiple link operation. However, no details have been specified as to how the multiple links are to be managed.

2.3.3 Network Level Interfacte - Level 3

In the present CUENET messaging system, all the functions are placed in level 3. The elements of the network level interface include:

Calling Address

Called Address

Message Length

Message Type and Code

Time Stamp (supplied by the C-bus controller)

There is no call set up state as the C-bus is assumed to be available at all times.

There is no message numbering for reference. Instead, the <sender address, time stamp> pair is used to identify a message. The physical processor number is used as the sender address rather than the logical processor number.

There is no mechanism in the CUENET protocol to distinguish one logical connection from another when there is more than one logical connection between the same two processors. There is no distinction between control and data messages. Also, the present protocol does not support multi-addressing and message broadcasting.

2.4 Observations

CUENET uses a message switching scheme which performs message transfer through direct user-to-user memory content transfer after the C-bus controller has checked that the receiver is ready to receive messages. One would note that the current CUENET messaging system has the following characteristics.

a) CUENET has minimum overhead for the control of the communications media;

b) It does not conform to the OSI reference model;

c) It does not conform to any IEEE 802 local area network standards or CCITT physical interface standards such as X.21;

d) It is heavily hardware based. Thus, it is going to be difficult to implement any enhancements or improvements to it.

e) The planned implementation of time stamping function in the C-bus controller is a simple scheme which can satisfy distributed database needs. However, it violates OSI principles and would not operate beyond CUENET (see Section 5.8 for details).

f) Any added overhead from protocol improvements could significantly reduce the C-bus throughput.

As a result, CUENET in its current form would have much difficulty when it is required to interconnect with either other local area networks or the public long haul data communication networks.

2.5 Other Characteristics of CUENET

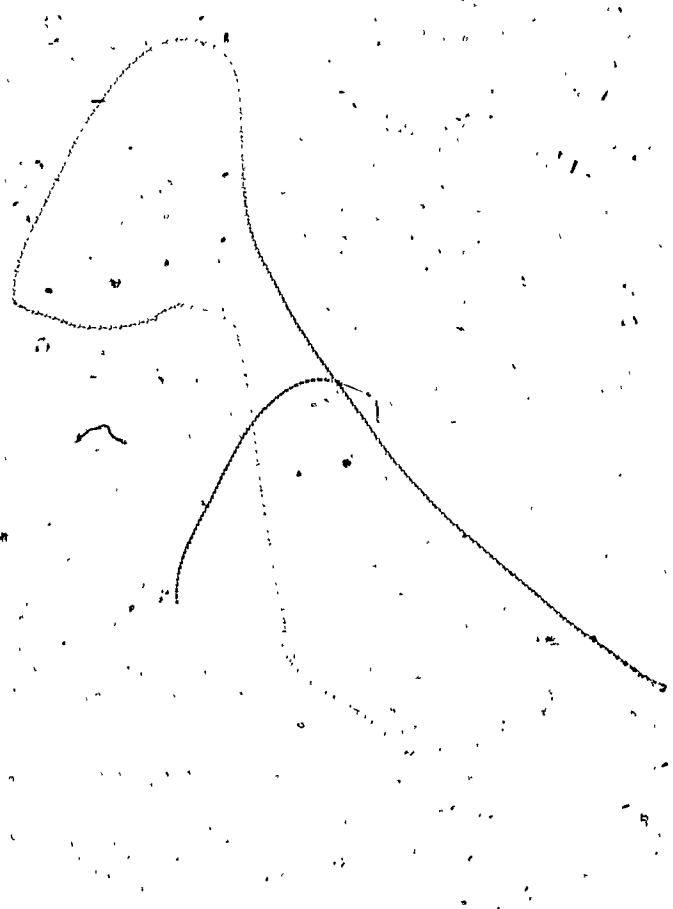
2.5.1 Expansion and Growth

The current C-bus length is only fifty (50) feet. It is a custom designed bus. Therefore, there are no off-the-shelf bus repeaters or bridges which can be readily obtained to extend the bus or to link up to another C-bus. Such devices will also require custom designs. The designing, prototyping and debugging of new devices will be costly and time consuming.

2.5.2 To Reach Out

Common computer communication networks are based on Ethernet [Metca 83], ARPA Internet [Hinde 83] or Xerox XNS. Most public data networks are based on CCITT X.25 packet-

switching protocol, whereas CUENET uses a special protocol which is designed to optimize the throughput of C-bus. This means that CUENET's present protocol would require complex protocol conversion functions when it is going to interwork with other computer communication networks or public data networks. Due to the fact that both C-bus and its protocol are custom designed, much manpower and a long development period will be required.



CHAPTER III

PROTOCOL REQUIREMENTS OF DISTRIBUTED COMPUTING SYSTEMS AND WHAT CUENET HAS PROVIDED

3.1 Distributed Computing Systems and Interprocess Communications

J. Mukerji [Muker 80] defines distributed computing as the use of multiple, quasi-independent processing modules, which operate asynchronously and yet in a coordinated way, to accomplish a large task. The potential benefits include modular expandability, increased reliability and availability, high performance, low cost performance ratio and distribution of complexity.

Processes of a distributed system are typically small [Jones 79] in comparison to counterparts in a uniprocessor multiprogramming system, and there are more of them. The use of many small processes could provide the potential benefits of enhancing reliability (as no one process is indispensable); eliminating duplicate implementation of large or specialized databases or processing functions; and maximizing the usage of available parallelism for improved performance.

The end result is that interprocess communication is used substantially more frequently for passing parameters, commands and data [Wells 83]. The typical types of functions performed via inter-process communications are:-

a) request/accept/terminate an interprocess logical link

b) pass data block between processes

c) activate/kill a remote process

d) transfer a file from one computer to another

e) retrieve/update records on a remote file.

This decentralization of data processing could lead to anarchy unless it is carefully controlled by adequate communication and co-operation protocols. The following discusses the various characteristics of distributed systems and their respective demand on the communication protocol. The discussion is structured into lower layer and higher layer functions in terms of the seven layers of the ISO Open System Interconnection (OSI) reference model.

3.2 Demands on Lower Layer Protocol Functions

3.2.1 Concurrent Processing

The use of concurrent, parallel processing puts a very high demand on the communication network and its protocol. Execution of a process may need to be suspended until the result from a remote process is correctly received [Brinc 78]. This requires a very fast communication link set up time and a very short network transit time.

CUENET has satisfied this requirement within its own local environment. The communication link can be assumed to be available and ready at any time.

3.2.2 Multiple Users

A processor in a distributed computing system environment could be accessed by more than one user at any instant. They can be from within the local environment or from a geographically remote site. This implies that the communication protocol should be able to support multiple ports at each processor or each communication port should be able to support multiple logical connections.

Within CUENET, messages are delivered to a physical processor address. There is no mechanism to identify different logical connections within a single physical processor address.

3.2.3 Transparency of Location

Transparency of location is the ability to access data without knowing where it is stored [Mager 80]. It allows end users and application programs to address and manipulate data stored on remote computers and data stored on the local node in a uniform manner [Mager 80]. In principle, data access should be independent of which node the required data is stored on. Users should be able to perform the same operations on local and remote data with a uniform procedure. Data transfers between interconnected systems must be carried out at a high rate in order to present this transparency to the system user.

Within CUENET, transparency of location is presented to

the user.

3.2.4 Access to External Computing Systems

One main benefit of distributed computing systems is that it eliminates the duplication of large and/or specialized databases or programs. Those databases or programs are very often outside one's local environment. The percentage of accesses to external computing systems varies from one application to another and it is different from one system to another. On average, it has been estimated [Burg 84] to be between 10 to 30 percent of all computer-to-computer communications. This figure is not insignificant to the designer of a distributed computing system. The communication protocol used in a distributed system environment should facilitate interworking with external cooperating computing systems.

This is the facility which CUENET is particularly lacking.

3.2.5 Message Security and Data Integrity

In an open, distributed computing environment, the most basic message security requirement is to protect the computer to computer messages against wrongful delivery of messages to other systems. The integrity of user data must also be preserved. The communciation protocol must have safeguards against delivery to wrong addresses and must

allow communicating users the freedom of implementing other security measures to ensure integrity and confidentiality of data.

This is a strong area of CUENET. Its use of the access vector and lock/key registers provide high security.

3.3 Demands on Higher Layer Protocol Functions

3.3.1 Unauthorized Use or Corruption

In order to prevent access by unauthorized users, some distributed applications would implement access checking and authorization schemes [Hsiao 79]. Some may require the use of "signed" messages. A signed message contains some trustworthy information as to the identity of the process originating the message [Panzi 85]. Another feasible security scheme [Nowit 80] is that each computer can maintain a sequence count for interactions with other computers and require a verification of the count at the start of each interaction. A would-be impersonator would have to steal not only the correct network address, user name and password, but also the sequence count.

However, the "signing" of a message or interaction sequence number checking should not be functions of the communication network. They should be dealt with by either the operating systems or application software.

3.3.2 Data Incompatibilities

In the case of heterogeneous processors, there are requirements over and above the fundamental task of transferring a set of bytes between ports. In particular there are problems due to incompatible character sets, byte ordering, floating point number representations, differences in sending and receiving formats, data types, data codes and data representations. In accordance with the OSI reference model, these issues should be handled independently by the higher (presentation or application) layers [Ranzi 85].

3.3.3 Exception Reporting

In some circumstances, it is necessary for the application software to be aware of the particular network-specific exception that has occurred [Panzi 85]. An exceptional response indicates that a fault has occurred and the operation should not be presumed to have been completed successfully. The information would allow the application programs to incorporate error recovery procedures. However, the system users normally expect and assume that the communication network services are reliable. When errors do occur in the transmission medium, they should be rectified without intervention from the users. Users should be notified only in cases of abnormal higher level error conditions.

3.3.4 Debugging Aids

System designers and implementors require some debugging tools [Stank 80] to allow them to determine when and where processes are created and destroyed, determine which hosts the individual elements of a process reside in, identify when, where and which copy of a program and/or data item is being used, identify communicating partners at all levels of the system including the frequency of communication and the amount of information transferred, and identify the actual data used by a process versus the true value of the data. Communication protocols should supply network related diagnostic or testing information to the system users. Furthermore, it should support the transmission of higher level debugging and/or testing information to the application programs.

CHAPTER IV

CUENET-2 ARCHITECTURAL DESIGN ISSUES

4.1 Design Objectives

Having analyzed the properties of CUENET in its current form and studied the requirements of distributed computing systems on the communication network services, the following design objectives have been set for CUENET-2.

(a) Enhance internetworking capability of CUENET.

(b) Compliance with ISO's OSI reference model.

(c) Adhere to existing communication protocol standards.

(d) Require minimum change to existing CUENET hardware and software.

(e) Provide adequate throughput, reliable and efficient.

(f) Facilitate growth and expansion.

(g) Be easy to implement.

The remainder of this chapter discusses how these objectives can be met by CUENET-2.

4.2 Switching Strategy

The current messaging system of CUENET uses a non-standardized form of message switching method. While it can serve the purpose of conveying messages from one processor to another within CUENET, it cannot be used outside its own local environment. Therefore, another messaging scheme is required for CUENET-2.

Packet switching has proved to be well suited for bursty transmission of messages which is typical of distributed computing systems. Furthermore, most of the currently available public data networks are using packet switching technology. There is no indication that the trend is changing. As a matter of fact, more and more circuit switched networks are using data packets for data transfer after network connections are set up.

The use of packet switching within CUENET will facilitate its internetworking with other networks. Therefore, packet switching is recommended for use in CUENET-2.

4.3 Protocol Structure

The most basic key to internetworking is complying with the principles of the standardized OSI reference model. The OSI model provides a reference for communication protocols, functions, services and interfaces required to interconnect application processes of end-systems. Appendix B provides an overview of the model.

The OSI model is structured in seven layers. The communication protocols which have been standardized recently or which are being revised by ISO and CCITT reflect such a layered structure. Such structuring of a protocol not only allows easy verification against the OSI reference model but also permits a modular implementation of the

protocol.

The protocol structure of CUENET-2 should follow the same layered, modular structure.

4.4 Internetworking Strategy

Studies suggest that most traffic flowing within a local network tends to stay within that local network. Still, figures ranging from 10 to 30 percent have been reported [Burg 84] for the off-net portion of the local-network traffic. Interconnection between a local network such as CUENET and long haul public data networks is one of the major considerations of network designers. This problem is examined in the following subsections and an interconnect strategy is suggested for CUENET-2.

4.4.1 Differences Between CUENET-2 and Public Data Networks

Public data networks are generally referred to as long haul networks as they are characterized by their topology's large geographic scope. Usually leased voice grade (telephone) lines, satellite and microwave links are utilized. Very often, the time required by a message to traverse a long haul network is a major factor to message throughput and message delay. When wide bandwidth is required, the network cost will be very expensive. To reduce the network cost, complex communications protocols are often used to efficiently utilize the communication

channel bandwidth.

A typical long haul network protocol would facilitate the performance of the following functions:

a) breaking down of messages into packets,

b) sequencing and reassembly of received packets,

c) retransmission or request for retransmission of lost or errored packets,

d) flow control, and

e) detection and destruction of duplicated packets.

A local communication network has a relatively small geographical scope. Bandwidths of 10 to 100 megabits per second are feasible and economical through the use of coaxial cable and optical fibres. A local network offers very high accuracy (typically less than one undetected error per 10^{12} bits); low and declining hardware cost; and high network reliability/availability. It usually serves computer systems and/or stations in the same building or campus within a geographic distance of less than one kilometer [Cotto'80]. Simpler protocols which require less processing time are normally used.

Table 1 presents a comparison of the two network types.

4.4.2 Four Internetworking Strategies

From the above examination of the characteristics of the two network types, one can observe that there are

TABLE 1

Comparison of Local and Long Haul Network Characteristics

Characteristic	Local Network	Long Haul Network
Typical Bandwidth	10 Mbit/s	9.6 Kbit/s typical 56 Kbit/s maximum
Acknowledgement	One at a time.	N messages at a time.
Message Format	Simple.	Large header.
Network Control	Minimum due to simple topology.	Extensive due to complex topology.
Flow/Congestion Control	Minimum due to high bandwidth and simple topology.	Extensive due to low bandwidth and complex topology.
Error Rate	Relatively low.	Relatively high.
Message Sequence and Delivery	Simple.	Complex due to complex topology.
Routing	Simple.	Complex.
Transit Delay	Small due to short distance and bandwidth.	Long due to distance and bandwidth.
Addressing	Simple for intra-net. Complex for inter-net.	Complex because of many nodes and links.

conflicts when a long haul network is interconnected with a local network. The following internetworking strategies are available to network designers [Warne 80]:-

First strategy - use the long haul network protocol for both long haul and local communications. This eliminates the need to design a new protocol for use within the local network. However, it will be at the expense of suboptimal local network performance.

Second strategy - implement both the long haul and local network protocols at each local network node. The disadvantage of this is that both the development and operating costs of the local network will be higher.

Third strategy - implement at each local network node only the local network protocol and place the long haul protocol functions at the gateway between the local network and the long haul network. The functionality needed in devices on a local network for internetworking would be minimized to reduce cost and complexity. The disadvantage of this approach is that the large amount of processing time for each message could result in a bottleneck when a large amount of traffic flows through the gateway.

A gateway is seen as constructed in two halves. The half towards a particular network consists of the network access functions and enhancement functions particular to that network. Putting the two halves together requires the

addition of functions mapping between the different encodings used in the two halves. Routing and parameter management functions are also needed to cater for the multiplicity of paths. In order to reduce the complexity of gateways, common encoding methods are used where possible in different enhancement protocols.

Fourth strategy - implement a local network protocol which retains all of the long haul protocol features not significantly affecting message delay or throughput in the local network. For example, the long haul protocol recovery (from host crashes) procedure could be implemented in the local network as they are used only rarely. The complex functions of flow control, resequencing and duplicate detection could be simplified or substituted to take advantage of the high reliability of the local network. The resultant protocol would well suit a local network yet it retains a high level of commonality with the long haul protocol. As both protocols could share substantial portions of software, each local network node could implement both the local and the internetworking protocols. Another option within this strategy is that each local network node will implement the resultant protocol and leave the resultant to long haul protocol translation function to the gateway to the long haul network. This translation will be relatively simple and thus a shorter processing time is required by the translator.

4.4.3 Selected Strategy

The fourth strategy (optimizing a long haul network protocol for use by CUENET-2) is chosen because of its flexibility, cost effectiveness and in keeping with the principle that the functionality needed in devices on a local network for internetworking should be minimized. (This principle is known as the minimum internetworking functionality (MIF) principle.) Within this strategy, the option of implementing the protocol translation at the gateway is selected.

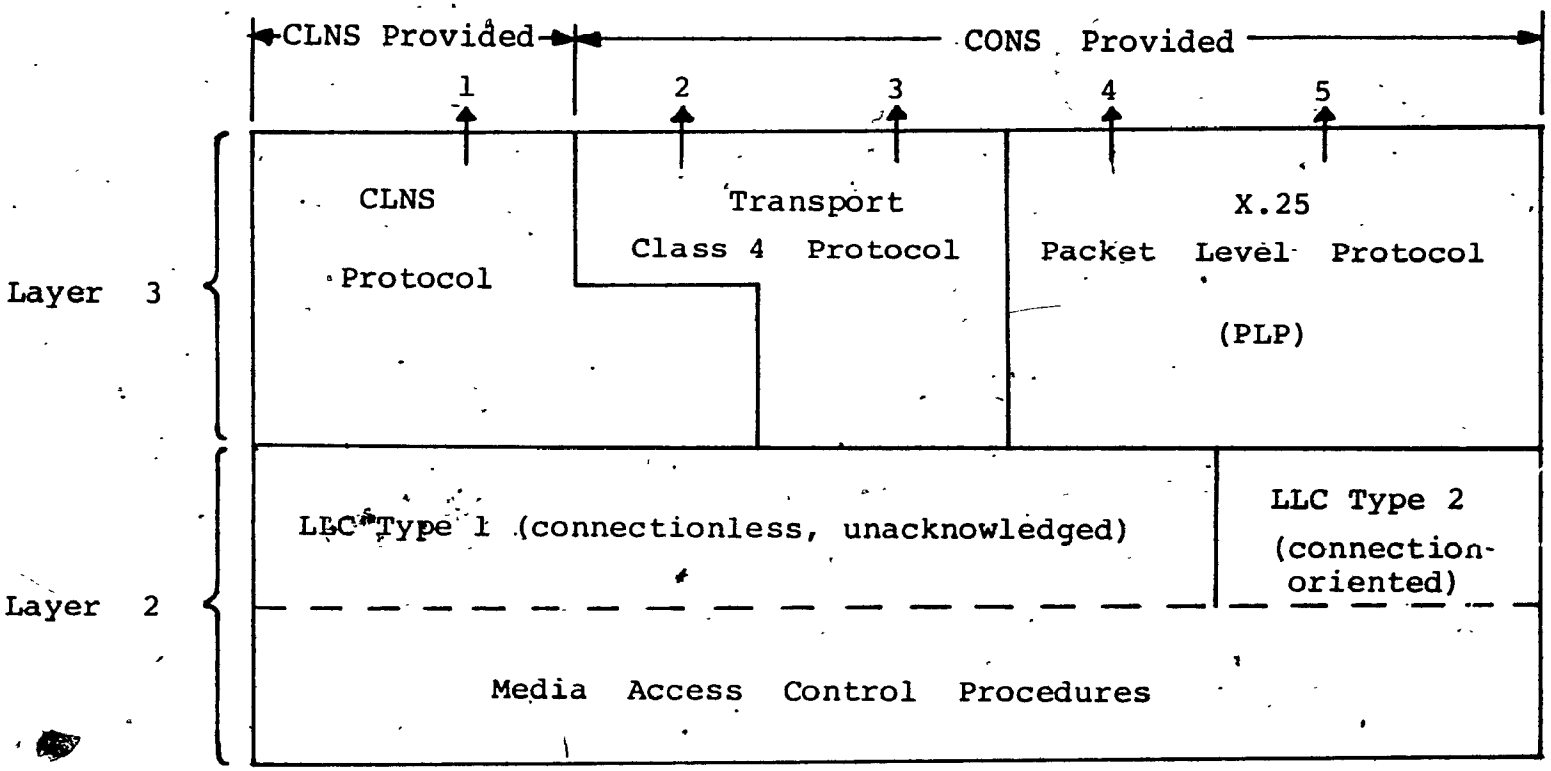
4.5 Network Layer

4.5.1 Available Alternatives

With the internetworking strategy selected above, this section examines the protocol alternatives for layer 3 with due considerations given to layer 4 and layer 2 functions and requirements. Figure 4.1 presents five combinations of protocols that have been suggested by members of ISO for implementation.

As shown in Figure 4.1, there are two types of OSI services on top of layer 3: the connectionless network service (CLNS) and the connection-oriented network service (CONS). A brief description of connection-oriented and connectionless protocols is provided in Appendix D.

Combination 1 provides the connectionless service at



CLNS : Connectionless Network Service

CONS : Connection-Oriented Network Service

Figure 4.1
Protocol Architecture Alternatives
For CUENET-2

the boundary of layers 3 and 4. It offers only addressing and data-transfer capabilities in layer 3. In contrast, the connection-oriented service provides more functionality at layer 3 while requiring less functionality at layer 4.

Combinations 2 and 3 in Figure 4.1 presume an unreliable service from the underlying subnetwork. They use a network layer protocol with the functionality of the Transport Class 4 protocol to turn this service into a connection-oriented one at the boundary of layers 3 and 4. While this approach appears to be viable for use with unreliable networks it should not be applied to local networks as they are generally very reliable and do not tend to missequence, misdeliver or delay data. Therefore, these two combinations are not discussed further in the following.

Combinations 4 and 5 make use of the X.25 packet level protocol (PLP) for layer 3. As for layer 2 LLC functions, both LLC Type 2 (which is very similar to LAPB OF X.25) and LLC Type 1 (connectionless service) can be suitable. This is made possible by the fact that X.25 PLP does not necessarily require the connection-oriented LAPB protocol. It only requires a "reliable" service in terms of undetected bit errors, sequence preservation and lack of duplication. By applying the principle of "layer independence" of the OSI reference model, as long as the services required of the lower layers are provided, it does not matter what protocol is used in the lower layers to provide the service.

Therefore, combinations 4 and 5 provide the same services to the transport layer. These two combinations are considered as one alternative (Alternative A) in the following discussion.

4.5.2 Evaluating the Alternatives

Figure 4.2 shows the two protocol architectures that can be obtained from combinations 1, 4 and 5 of Figure 4.1. An appropriate choice of a transport layer class protocol is also shown for each of the two protocol alternatives.

The main difference between the two alternatives is that Alternative A uses a connection-oriented (X.25 PLP) network layer protocol while Alternative B uses a connectionless network layer protocol. The following addresses how the two alternatives meet the functional requirements.

Network Service Access Point (NSAP) Identification

In OSI terminology, an NSAP address identifies a transport-layer entity but not necessarily the end-system containing that entity. For communications within a local network where there is only one NSAP that corresponds identically to the subnetwork address of the end-system, the NSAP address can be carried in the address fields of the MAC sublayer. In cases where there exist multiple NSAP addresses under the same subnetwork address of the end-system, the address extension facilities (AEFs) of the X.25

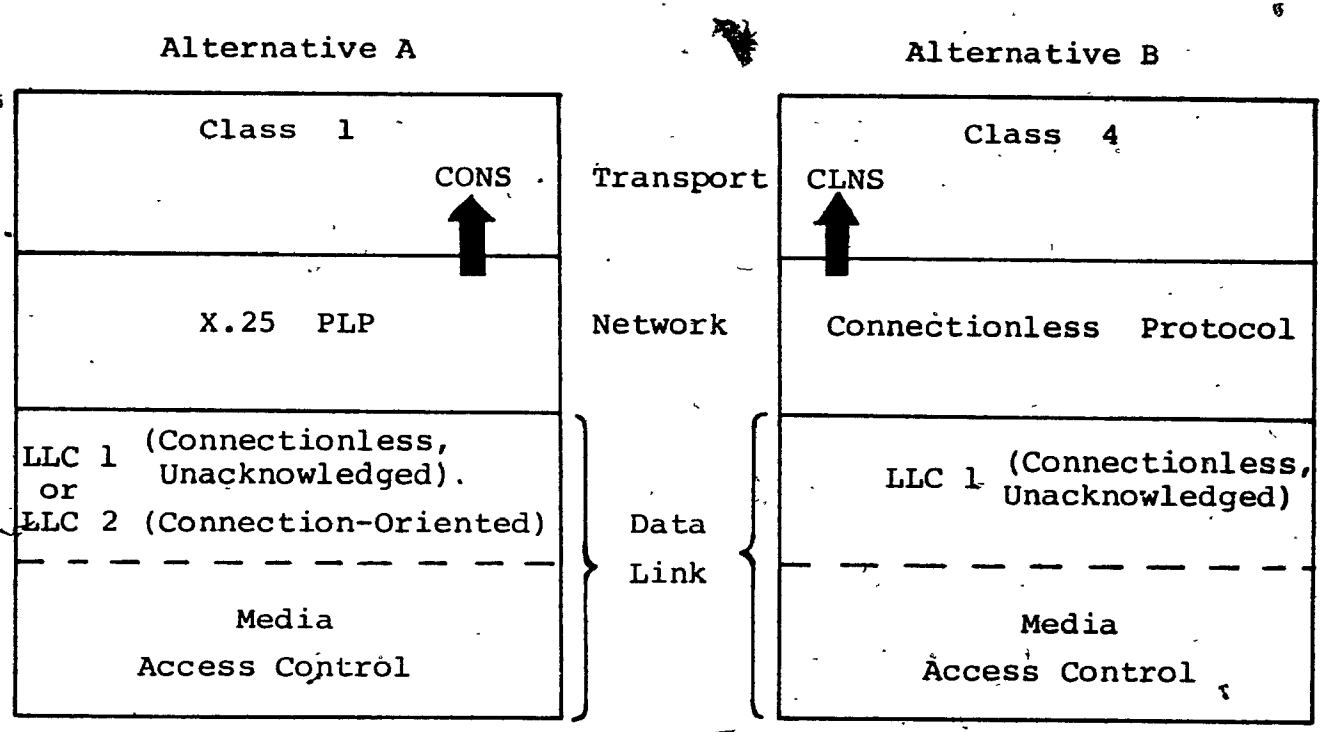


Figure 4.2

Two Alternatives from Combinations 1, 4 and 5

PLP are used to carry the NSAP address under Alternative A.

The inactive subset of the connectionless protocol (CLP) in Alternative B does not provide any means of conveying NSAP addresses. In the internetworking case, both the CLP and the X.25 PLP using AEFs, provide a mechanism for carrying the NSAP addresses to the gateway. Under Alternative B, the local network must know whether the destination is off-net, in which case it needs to use the full CLP. This requirement is not in keeping with the spirit of the MIF principle.

Multiplexing

Multiplexing can be provided by both alternatives. In Alernative A, the X.25 PLP supports the function. In Alternative B, it is provided by the Transport Class 4 protocol.

End-to-End Flow Control

End-to-end flow control is supported by both alternatives in a similar manner. Using the ISO X.25 PLP in DTE/DTE connections, there is no intervening network to decouple the ISO X.25 PLP flow control mechanisms at the two interfaces. When networking is via a packet switched network, the network provides an extra buffering capability. In this case, end-to-end flow control is still achieved through back-pressure by using the layer-3 flow control to toggle data flowing across the boundary between layers 3 and

4.

Another advantage of using the X.25 PLP when internetworking with a packet switched network is that the flow control packets used by the local network are then mapped directly to the corresponding flow control packets on the interface between the gateway and the packet switched network. This avoids the extra packet cost. On the other hand, the layer 4 flow control messages under Alternative B must be carried as billable extra data packets across the network, or else must somehow be recognizable by the gateway.

End-to-End Acknowledgement

Both alternatives use the same transport layer acknowledgement procedures. However, the D-bit (for delivery confirmation) of X.25 PLP may be negotiated under Alternative A.

Error Detection and Recovery

Both alternatives first rely on the frame check sequence (FCS) of the MAC sublayer to detect errors.

Lost data under Alternative B is detected only after a transmitter time-out in the Class 4 transport protocol. When this happens, the transport layer would retransmit any unacknowledged data. This approach has two disadvantages. First, the transmitter must wait for a timer to expire

before resending data, thereby introducing extra delay. Second, if it is the acknowledgement that had been lost, then this action would result in wasted retransmission of data already received.

The X.25 PLP is known to be robust enough to recover from many types of lost packet conditions.

Segmentation and Reassembly

Both alternatives provide a mechanism for segmenting and then reassembling the packets back to a large data block. Alternative A provides it through the use of the M-bit of the X.25 PLP while Alternative B provides an EOT-bit in the transport layer protocol.

Internetworking and Gateways

The use of the X.25 PLP in a local network simplifies the gateway to an X.25 network. Figure 4.3 shows the architecture of such a gateway.

The local network side of the gateway can logically be viewed as consisting of multiple DTE/DTE interfaces, each one being dynamically created as the local network needs to communicate to the X.25 network. The logical channels of the various DTE/DTE interfaces are then mapped to a single DTE/DCE interface. To do this, the gateway must be involved in the X.25 call setup phase to patch two calls together. During call setup, the gateway must also map the NSAP

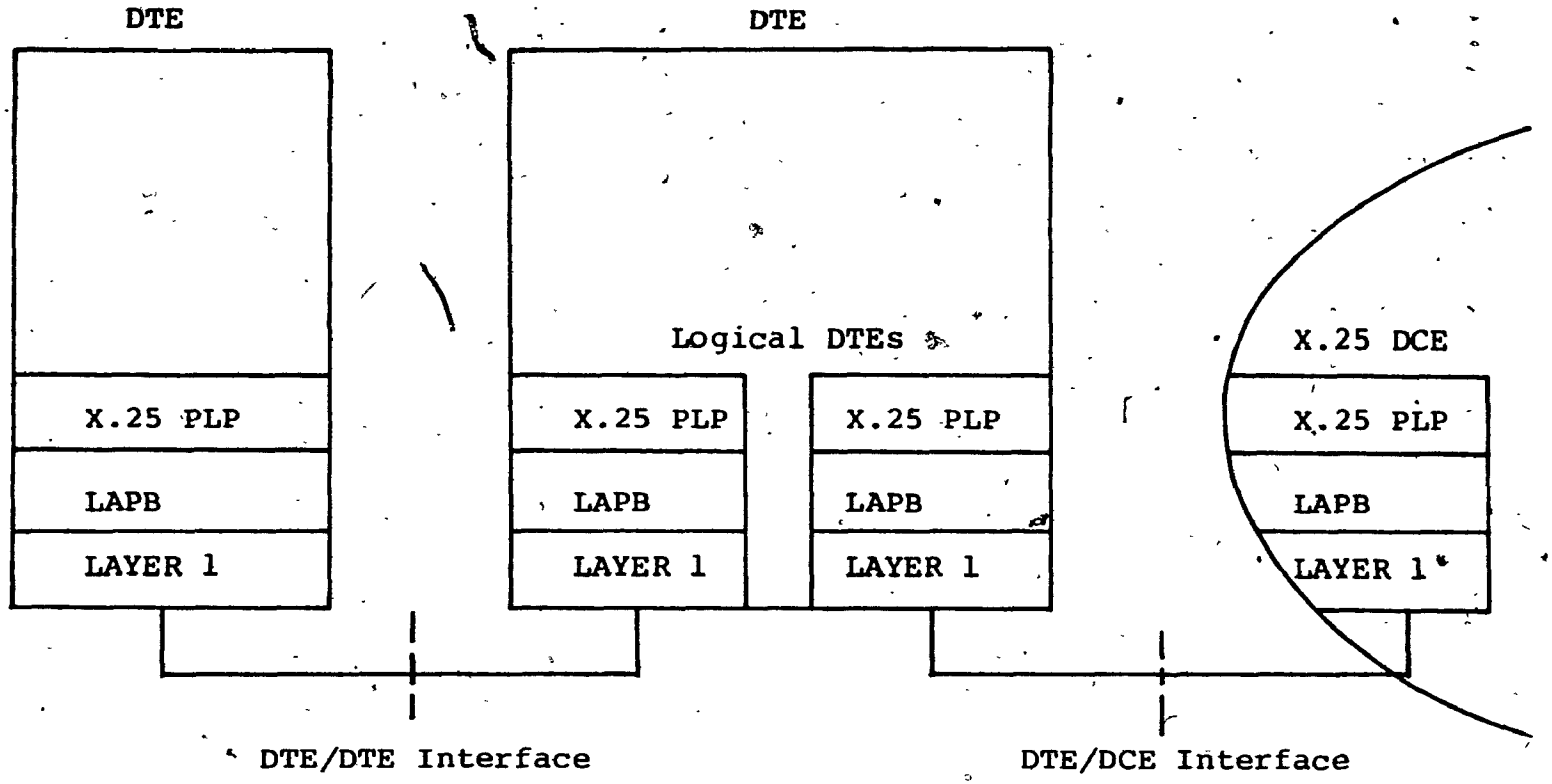


Figure 4.3

Architecture of An X.25 Gateway

address to an X.25 subnetwork (CCITT X.121 based) address. The X.121 [CCITT 84b] address could identify an end-system on the X.25 network, another gateway, or even another system on a distant local network.

The gateway generally does not need to be involved with the data transfer phase of X.25 except for mapping logical channel numbers, handling optional X.25 user facilities or handling variable X.25 packet and window sizes.

For Alternative B - using the connectionless protocol and the Transport Class 4 protocol - internetworking is more complex and less efficient. First of all, the local network device must know that internetworking is necessary so that it can use the fully expanded version of the connectionless protocol, which violates the MIF principle. Mapping of NSAP addresses to X.121 addresses in the gateway is handled in much the same way for either alternative. However, the connectionless protocol of Alternative B provides no information to the gateway regarding when a call to the X.25 network should be set up or taken down. To get this information, the gateway must decode the transport layer (Class 4) protocol to determine when a connection is being made.

Decoding the transport layer protocols by the gateway violates the spirit of the OSI reference model as it is inconsistent with the principle of layer independence. Alternatively, if the gateway only looks at the network

layer connectionless protocol, then it must decide when to set up and take down X.25 calls. This leads to inefficient use of gateway and network resources.

The Transport Class 4 protocol is designed to guard against errors that do not normally occur in an X.25 network or in local-net-to-local-net interworking or in intra-local-net environment. Therefore, Class 4 transport provides excess functionality which is a waste of resources. Measurement results show that the addition of the Class 4 protocol results in a tripling of the number of central processing unit cycles needed to run a transport layer protocol, even if only the features of a Class 2 transport are actually used [Burg 84].

4.5.3 The Selected Layer 3 Protocol

The selected alternative is to use the X.25 PLP at layer 3. In this case, only the basic functions of the X.25 PLP are used in providing the OSI network layer services in a local network environment. For communications within the local network, this alternative has several advantages, particularly in the areas of addressing and error recovery. Also, this approach does not put the burden of unneeded functionality on CUENET-2 for atypical events that occur within the local network. At the same time, this approach exhibits several advantages when typical internetworking scenarios involving local networks are considered. The

operation of the X.25 PLP for internetworking is no different from that for communications within the local network. There is no specific functionality set aside just for internetworking. Furthermore, it should be noted that X.25 PLP products are widely available for implementation.

On the contrary, the use of the connectionless protocol (CLP) with the Transport Class 4 protocol provides excess functionality for communications both within the local network and in typical internetworking cases. The use of CLP and Class 4 transport for internetworking over an X.25 network is highly redundant. Also, this alternative requires that a station distinguish between internetwork and intra-local-network communications so it can use the proper set of procedures for the connectionless protocol. Finally, this approach, by providing network "reliability" in the transport layer, does not take advantage of inherently high local network reliability and is not a requirement of the OSI reference model.

4.6 Link Layer

4.6.1 Connection-Oriented or Connectionless Link Control

There are two candidates for the link layer protocol in any particular environment: connection-oriented and connectionless. The common carriers [TG100 84] had decided to support connection-oriented network services over wide-area (long haul) networks. It has also been the position of

many common carriers that where connectionless operation has been selected for a local environment, it is regarded from outside as a single OSI end-system. Such end-system shall communicate with other end-systems in the wide-area environment using connection-oriented network service.

The simplicity and the corresponding lower overhead are strong arguments in favour of connectionless protocols. In the absence of errors, connectionless protocols have lower overhead than connection-oriented protocols. Simulation results [Meist 85] reveal that a connection-oriented protocol at the link level on a LAN may reduce throughput for file transfer by as much as fifty (50) percent with the use of a relatively slow (250 Kips) processor. However, the results also show that with the use of a high speed (10 Mips) processor, throughput of a connection-oriented data link becomes the same as that of a connectionless data link.

Most distributed systems have adopted connectionless protocols. However, there are good reasons for using connection-oriented link protocols in LANs in spite of the resulting performance penalty. LANs very often are connected to long haul networks which implement connection-oriented protocols. Furthermore, when the receiver has no buffer for an incoming packet, that packet will be lost. This could cause frequent losses of packets. A connectionless protocol may significantly degrade

performance. The use of a connection-oriented protocol would also allow for the neogtiation between all communicating entities of parameters and options that will govern the transmission of data. The entities can also discuss their requirements to enable them to reserve whatever resources (such as memory space) they may need. Then after transferring a series of related data units, the entities explicitly end their interaction and release the previously reserved resources.

4.6.2 The Selected Link Layer Protocol

A connection-oriented link control protocol is recommended for use by CUENET-2 in view of its various merits. It is chosen because of another factor that more and more VLSI/LSI chips are becoming available to implement data link control functions. Those specialized chips could provide high data link throughput while off-loading the link control functions from the host computer.

CHAPTER V

CUENET-2 PROTOCOL DESIGN ISSUES

5.1 Optimizing X.25 For Use in CUENET-2

It would have been ideal if a standard public packet switching protocol such as X.25 could be used within CUENET in its entirety without compromising its throughput and quality of service. Unfortunately, X.25 is designed for use in long haul networks which usually call for complex protocols that sacrifice processing time in order to efficiently utilize communication channel capacity. On the other hand, local area networks very often call for simple communication protocols which waste channel capacity in order to reduce processing time. This creates a conflict of interest if a local area network is interconnected with a long haul network. It is particularly so if each network intends to use the communication protocol it employs for both internetworking and intranetworking.

Internetworking is even a bigger problem with CUENET because it does not have the typical high bandwidth of LANs (its maximum throughput is estimated at only 1.6 Mbit/s). Therefore, it does not waste any channel capacity. At the same time, its messaging protocol and communication software are already very much optimized in order to achieve its present throughput. Thus, we cannot reduce any communication processing time either.

The following sections discuss the various aspects of

X.25 which can be optimized for application in CUENET in order to achieve the highest possible throughput within CUENET while retaining all of the essential properties of X.25 for the ease of internetworking.

5.2 Addressing

An "address" in a data communications network indicates the location of a resource on that network [Panzi 85]. Within CUENET, only one byte is required for the sender or receiver address. Thus, for intranetwork messages, the address length fields are always set to one. For internetwork messages, the full addressing capability of X.25 is required. Consequently, it is possible for the C-bus controller to readily differentiate intranetwork and internetwork addresses.

In order to realise the full potential of world-wide internetworking, it is necessary to have a unified, global scheme of network layer addressing similar to the telephone network. CCITT and ISO are still studying this problem and there is no international agreement on the detailed structure or maximum length of OSI network addresses. In the mean time, it can be assumed that the scheme(s) ultimately standardized will include addresses of up to 32 decimal digits [TG100 84]. Also, it should be noted that the address extension facility (AEFs) can be used to convey the Network Service Access Point (NSAP) on the network/transport boundary with an end-system.

For a pair of communicating end-systems, multiple logical connections are supported through the use of multiple logical channels. This would meet the requirement of multiple user access of distributed computing systems.

5.3 Link Layer Flow Controlling, Windowing and Acknowledging Received Packets

Flow control is a mechanism used to regulate the flow of data into, out of, and within the network to prevent the situation wherein there are more packets than the network can accommodate. Flow control is particularly important in a gateway between a fast LAN and a slow long haul network. Multiple packets can be arriving in quick succession over the DTE/DTE interface while packets are going out over the DTE/DCE interface rather slowly. There could be no free receive buffer for an incoming packet over the DTE/DTE interface. Consequently, that packet will be lost.

One method to detect the loss of a packet is to implement an acknowledgment scheme. With X.25 LLC, the timing of acknowledgments is determined by the sink of the data transfer. This scheme is called sink triggering. When sink triggering is used, the sink is free to generate acknowledgments at its own discretion. This causes the window to move by sliding, hopefully without ever closing. Acknowledgments can be piggybacked onto data packets, thus eliminating the overhead of processing and transmitting

separate acknowledgment packets. Applications such as unidirectional file transfers do not generate sufficient reverse traffic and thus require separate acknowledgment packets. To minimize the ensuing overhead, one implementation technique consists of accumulating acknowledgments up to some fixed number N so that an acknowledgment packet is generated after every N data packets. N must not be larger than the window size or else the two communicating entities would end up in a deadlock. N has been found to be optimal at 3 [Meist 85] for a typical processor speed of 1 Mips.

Some network designers [Warne 80] have found that flow control in a LAN is an unnecessary overhead and have implemented other schemes such as the one for one acknowledgment scheme of the Command Center Network of the U.S. Navy [Warne 79]. This is the equivalent of setting N = 1 in the above scheme. Simulation results [Meist 85] indicate that system throughput is reduced to about fifty percent of that obtainable at N = 3.

The choice between an acknowledgment accumulation and one-for-one acknowledgment strategy is a local implementation matter. Also, it is not required that the same strategy is used by both communicating partners. However, if only one side uses acknowledgment accumulation, performance benefits will be apparent in only one direction.

As for CUENET-2, throughput is a major concern.

Therefore, the use of acknowledgment accumulation is recommended.

5.4 Maximum User Data Field Length

The standard maximum user data field length of X.25 is 128 octets. In addition, other nonstandard maximum user data field lengths which may be available are: 16, 32, 64, 256, 512, 1024, 2048 and 4096 octets.

In CUENET, it has been found that a message size of 64 octets is adequate [Gross 82]. However, it was decided to select a maximum message size of 256 octets in order to make it the same as the sector size of the floppy disks in the host computers.

It is recommended that CUENET-2 should use the standard size of 128 octets since it is already more than adequate for the host computers. This would avoid the use of a nonstandard maximum user data field length.

5.5 Error Recovery

Error recovery software accounts for many instructions. However, it is invoked only in case of errors. In their absence, the overhead of the error recovery mechanisms is limited to a few tests and calls to start and stop timers, and is usually negligible in implementations.

5.6 X.25 Functions and Features Not Supported

5.6.1 D-bit

The D-bit is used for delivery confirmation. In view of the fact that the local communication medium is reliable and that there already exist a link layer data acknowledgment scheme, there is no requirement of this feature for intranetwork communcations.

5.6.2 Q-bit

The qualifier bit (Q-bit) is used to designate a number of packets for the network to deliver in sequence. This is not required within CUENET as only one packet can be sent on C-bus at any time. There is no packet out of sequence problem.

5.6.3 Permanent virtual circuits

One important property of CUENET is its reconfigurability. The use of permanent virtual circuits will remove such property. Therefore, the use of permanent virtual circuits is not required.

5.6.4 Other optional user facilities

Optional user facilities are hard to manage and they introduce a large overhead. Therefore, no optional user facilities should be supported by CUENET-2 except the address extension facility for signalling NSAP addresses.

5.6.5 Cause and diagnostic codes and Diagnostic Packets

Diagnostic codes and packets are used for pinpointing network problems. As a local network is usually very reliable, there is no need for these facilities within CUENET-2. They should not be supported by CUENET-2.

5.7 CUENET Message Types

In CUENET, the Access Vector is used for configuring the processors at the set up of the processors to perform a certain job upon request. This is considered a high level function in the OSI reference model. Unfortunately, CUENET implemented this function at the low (C-bus controller) level. In an effort to use the existing CUENET hardware and software as far as possible, it is recommended to modify the usage of the format identifier bit of the link level control field. This modification would accommodate the three message types of CUENET. The interpretation of these bit is as follow:-

Intranetwork Use	Bit 1 2	Internetwork Use
Intercomputer Message	0 X	I format
Access Vector Load	1 0	S format
Error Message	1 1	U format

An X denotes a don't care condition.

These two bits should be interpreted in conjunction

with the network layer general format identifier (bits 5 and 6). When the latter two bits indicate that the packet is an intranetwork packet, the left side of the above interpretation table is used. For internetwork packets, the right side of the table should be used.

This modification could be used until full link layer control is implemented. At that point, the function of setting the access vectors should be moved to the application layer.

5.8 Time Stamping of Data Packets

In updating distributed databases, sometimes it is important for the database manager to know the time of an update. In some other occasions, one may like to trace a message to check its network transit time. These are some of the reasons for incorporating a time stamping feature in CUENET to put down the time at which the message passes through the C-bus controller.

These time stamps need to propagate towards the upper layers to reach the application program for analysis. This violates the principles of the OSI reference model as the time stamp is injected by the network and not by the other communicating application entity. Normally, when end-user information propagates upward from a lower layer, its header information is stripped. When the information finally reaches the user or application program, the time stamp

which was put in by the C-bus controller would have been removed by the layers in between. Figure 5.1 depicts how protocol elements are added to the information to be transferred by the transmitting equipment's entities. It also shows how each of the entities at the receiving terminal equipment evaluates those protocol elements which concern it and removes them again. Therefore, it is not possible to convey time stamp information through the use of protocol elements of a packet header.

By carefully analyzing Figure 5.1, one would notice that the only possibility of conveying time stamp information upward to the application entity from the network is by placing the time stamp at the end of the user information field. With a time stamp length of 3 bytes [Gross 82], the really usable maximum user data field length is 3 bytes less than the maximum length selected in section 5.4 above. Furthermore, there should be an indication to the receiver that the actual data has ended and what follows is a time stamp which the receiver has the option of using or not. This indicator takes up one byte. Therefore, the maximum user data field length is reduced to 124 bytes.

While the above suggested scheme can accommodate this special requirement of CUENET, it must be noted that it is a major violation of the OSI principles. It should not be done if any other mechanism can be found to satisfy the requirement. It should also be noted that once a full

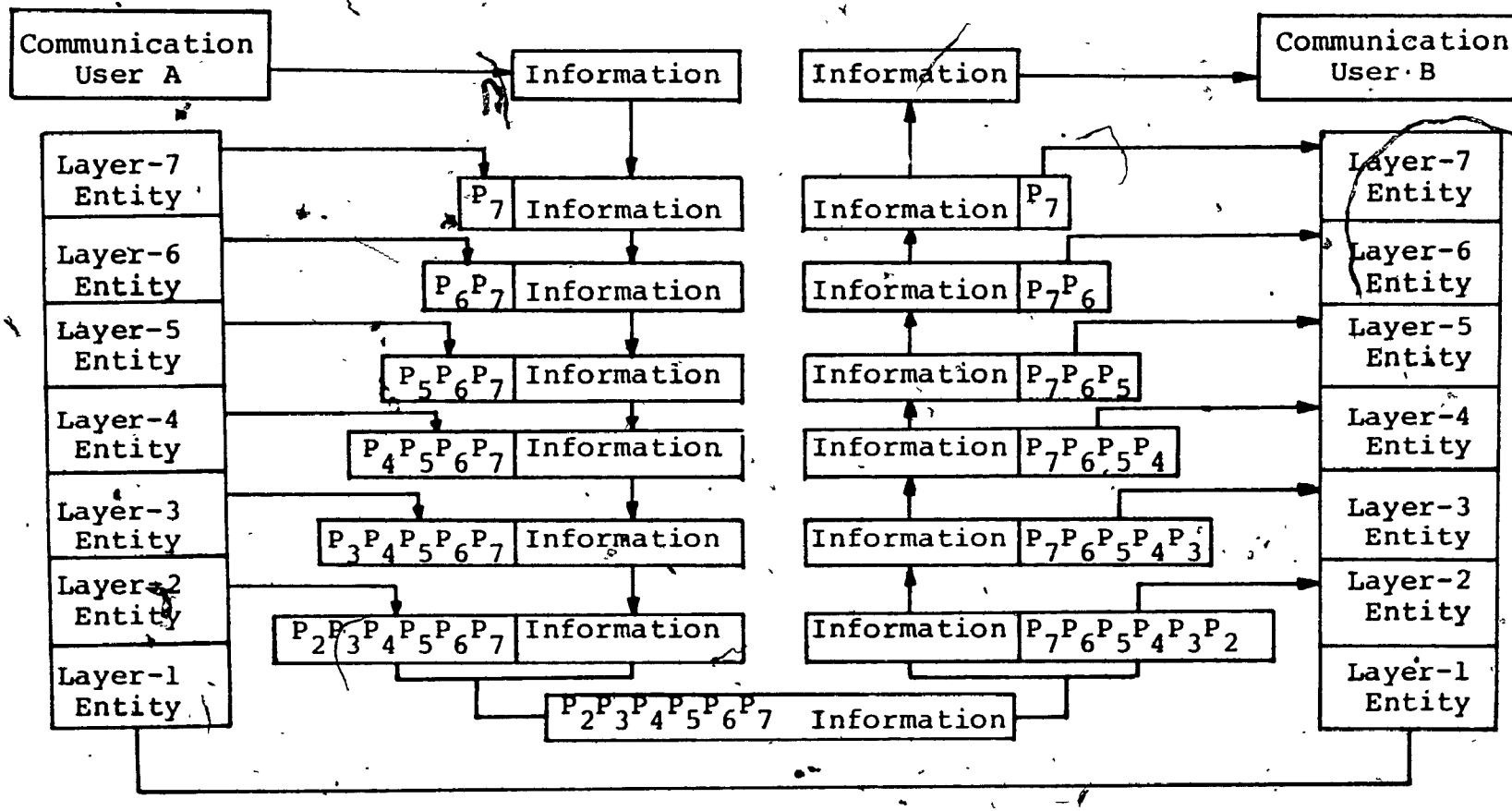


Figure 5.1

Propagation of Protocol Elements

standard link control procedure is implemented, this scheme may not be usable anymore.

5.9 Multi-bus Operation

If in the future it is found that a single C-bus does not provide an adequate throughput or reliability, a second C-bus can be constructed and connected in parallel to the same host computers. If this does occur, the protocol recommended herein will still be applicable except that the standard multilink procedure (MLP) of X.25 will be used for the link control layer.

CHAPTER VI

CUENET-2 PROTOCOL SPECIFICATION

From the discussions made in the last chapter, LAPB of CCITT Recommendation X.25 [CCITT 84a] has been selected as the link layer protocol in CUENET-2. The frame format is shown in Figure 6.1. As for the network layer, the ISO X.25 DTE/DTE packet level protocol [ISO 84] has been found the most appropriate. Figure 6.2 presents the format of a call request packet in order to give an example of what a packet format is like. In this section, the ways of implementing these protocols in the current CUENET environment are discussed. In the following discussions, references are very often made to the two above mentioned CCITT and ISO standard documents. To ease the task of cross referencing to those two documents, the appropriate subsection or item number(s) of the respective CCITT or ISO document are indicated in brackets.

6.1 Link Layer Protocol

CUENET, in its present form is not based on a X.21 or X.21-bis physical layer which X.25 specifies. The following examines how LAPB of CCITT X.25 can be accommodated in CUENET without too much penalty on the throughput of C-bus.

6.1.1 Frame Structure (CCITT 2.2)

In principle, the entire frame structure of LAPB of X.25 can be accommodated. The following fields of the LAPB

Flag	Address	Control	Information	FCS	Flag
01111110	8-bits	8-bits		16-bits	01111110

Figure 6.1

Link Layer Frame Format

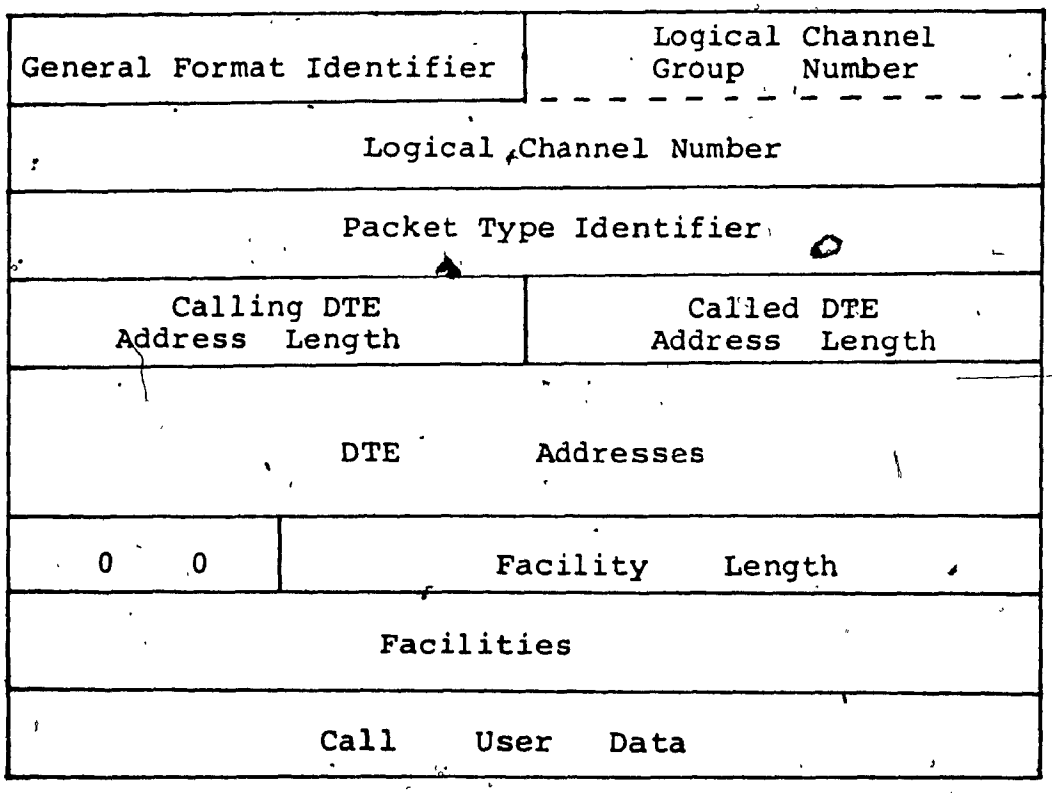


Figure 6.2

Call Request Packet Format

header will not be used until full implementation of LAPB is made after replacing the C-bus at a later date. This is because implementation of them on the existing C-bus will introduce significant overhead thereby reducing the throughput of C-bus very drastically.

(a) Flag Sequence (CCITT 2.2.2)

(b) Address Field (CCITT 2.2.3; 2.4.2)

(c) Sequence Numbering of Control Field (CCITT 2.3.2.1)

(d) Frame Check Sequence (CCITT 2.2.7).

However, buffer space (a total of 4 bytes) for the above fields in the output buffer of the C-bus interface should be reserved for future use. They should be implemented on special link layer VLSI/LSI chips.

6.1.2 Control Field (CCITT 2.3.2.1)

For intranetwork communications, the use of the format identifier should be interpreted as follows:-

Bit	1	2
Intercomputer Message	0	X
Access Vector Load	1	0
Error Message	1	1

Note: X denotes a don't care condition.

6.1.3 Commands and responses (CCITT 2.3.4)

The commands and responses through the use of the

control field should not be implemented on C-bus.

6.1.4 Frame Check Sequence (CCITT 2.2.7)

Frame check sequence will not be implemented on C-bus. The use of hardware parity check will be continued until C-bus is replaced.

6.2 Network Layer Protocol

Incorporating ISO X.25 PLP into CUENET is relatively more straight forward than incorporating the CCITT X.25 link layer protocol. This is due to the fact that the PLP functions can be implemented by software of each host computer. Presently, what a host needs to do is to perform a checksum on the received message. All PLP functions will be added to it. In view of the fact that while a host computer is performing these PLP functions, it is not using the C-bus, these added functionalities will not impact on the throughput of the C-bus.

6.2.1 Basic Structure of Packets (ISO 12.1)

No change is required for incorporation into CUENET-2. The standard maximum user data field length of 128 octets is adopted.

6.2.2 General Format Identifier (ISO 12.1.1)

No change is required except that zeros in bits 5 and 6 would identify that the packet is an intra-network packet.

6.2.3 Logical Channel Identifier (ISO 12.1.2)

The use of logical channel identifier could be delayed until CUENET processors can handle multiple users whereby multiple logical channels are required.

6.2.4 Address Length Fields (ISO 12.2.1.1)
Address Fields (ISO 12.2.1.1)

No change required.

6.2.5 Facility Length Field (ISO 12.2.1.1)
Facility Field (ISO 12.2.1.1)
Call User Data Field (ISO 12.2.1.1)

No change. The Address Extension Facility (ISO 14.1 and 14.2) is required for carrying Network Service Access Point (NSAP) address to the transport layer when interworking with long haul networks. The other optional facilities are not supported within CUENET-2.

6.2.6 Qualifier Bit (ISO 12.3.1)
Delivery Confirmation Bit (ISO 12.3.1)

Not required for intra-network use.

6.2.7 More Data Bit (ISO 12.3.1)

No change.

6.2.8 Packet Receive Sequence Number (ISO 12.3.1)
Packet Send Sequence Number (ISO 12.3.1)

No change.

6.2.9 Flow Control Packets (ISO 12.4)

No change.

6.2.10 Reset Packets (ISO 12.5)
Restart Packets (ISO 12.6)

No change.

6.2.11 Diagnostic Packet (ISO 12.7)

Not supported within CUENET-2.

6.2.12 Reject Packet (ISO 12.8)

No change.

6.2.13 Registration Packets (ISO 12.9)
On-line Facility Registration (ISO 13.1)
D-bit Modification (ISO 13.3)

No need for such packets for intra-network use.

6.2.14 Extending Packet Sequence Numbering (ISO 13.2)
Packet Retransmission (ISO 13.4)

No change.

6.2.15 Incoming Calls Barred (ISO 13.5)
Outgoing Calls Barred (ISO 13.6)

Not required for intra-network use.

CHAPTER VII

CUENET-2 IMPLEMENTATION AND FUTURE DEVELOPMENT

7.1 Protocol Software Design

7.1.1 Software Structure

One purpose of structuring the recent CCITT and ISO standardized communication protocols into layers in accordance with their respective functions is to allow modular implementation of protocol functions either in software or firmware. This is due to the fact that a layered protocol structure can easily be mapped onto a layered process structure. CUENET-2 should take advantage of this and implement its protocol software in such a modular structure. There are a few advantages in doing so:

(a) testing and verification against the layered protocol is made easier,

(b) the current trend in the design of new VLSI/LSI protocol chips is in the basis of one chip per layer structure; by take the same modular approach, new protocol chips can readily replace some software modules as and when those chips are employed.

It is also important to separate the communication processing software from the system software. This will allow the use of specialized high speed communication processors to implement all those communication handling tasks.

7.1.2 Shared Software Codes

The selected protocol for CUENET-2 is basically a version of X.25 optimized for local area network applications. Most of the X.25 functions and features are retained for use within CUENET-2. With this protocol, we can design the software in such a way that coding for a full X.25 implementation is done. Those sections of software unique to either internetworking or intranetworking could be bypassed or executed, depending upon the situation. A full X.25 implementation is required for the gateway's DTE/DCE interface in any case. This strategy also allows the use of identical protocol software in every host computer. Any one of them can be appointed as the gateway to the long haul networks.

7.1.3 Buffer Passing

Copying data to and from buffers between protocol layers is one of the major sources of overhead, especially when handled by a programmed loop. Even with a hardware-supported memory block move, copying is performed with full memory bus capacity which is comparable to the network capacity, meaning that every copy operation lengthens packet processing by an amount of time comparable to packet transmission time. Thus, it is preferable to pass buffers between layers by reference. For out-bound packets, space for the additional headers can either be preallocated in front of each packet, or provided in physically scattered

memory segments, logically prefixed as packets pass through the layers, and gathered in one packet before transmission.

A buffer management scheme that allows fragment decomposition, tagging, and reassembling to take place without actual data movement would also improve throughput [Powel 80].

7.1.4 Overlapped Operations

In a file-transfer application, one prerequisite for achieving high transfer rates is to overlap the LAN I/O, disk I/O, and packet preparation. Thus, it is important that the modules handling these different tasks can trigger each other's operations in an asynchronous fashion [Meist 85].

7.1.5 Task Priority

Link control has priority over transport control on the grounds that it may serve several parallel transport connections at a time.

7.2 VLSI/LSI Protocol Chips

Since CCITT Recommendation X.25 was first approved in 1976, the standard has gained wide spread acceptance. To date, more than 30 [Colli 86] public data networks provide X.25 communications services worldwide, including international connection capability. In addition, many

private X.25 networks exist.

The early implementations of X.25 were mainly implemented with software which can provide moderate throughput. Manufacturers can thus avoid the lengthy and capital intensive development of special X.25 VLSI/LSI chips. This would also allow them to determine the general acceptance of the protocol and assess its market potential. By keeping the protocol implementation in software, manufacturers can also maintain maximum flexibility to meet varying country and customer requirements.

With the great acceptance of X.25 and the general trend toward higher-speed lines, many manufacturers [Thurb 80] [Weiss 83] have gone into developing special VLSI/LSI chips to implement X.25 protocol function in order to meet their market demand. Semiconductor manufacturers split the communication problem at the physical level (layer 1) and the data link level (layer 2) interface. This division is based upon the different technologies required by the two levels. The drive capability and/or speed requirements of the physical level often require bipolar processes, while link-level functions are best designed in low-power, highly integrated high-speed CMOS (HCMOS) processes. Moreover, this division also allows the link level chip to have a larger potential volume, as it is independent of the physical media.

7.2.1 Level 2 Chips

As an example, Motorola's MC68605 [Colli 86] X.25 Protocol Controller (XPC) is a full level 2 device which implements the 1984 CCITT X.25 LAPB data link procedure. It is implemented through the use of Motorola's "communication engine" concept. A communication engine is composed of four major blocks: the microcontroller, DMA/bus interface, register file/ALU and serial/FIFO. Its structure is illustrated by Figure 7.1.

The microcontroller block includes two independent engines which operate in parallel to each other for high performance. The microcode can be extended to 4 Kbyte without redesign of the microcontroller. Four DMA channels support sophisticated memory structures for host communication. The DMA/BUS interface block is coupled closely with the FIFO structure to optimize data transfer handling. The serial/FIFO block is designed to operate at 10 Mbit/s full duplex.

The XPC performs the following operations in the connect mode:

(a) Transmit a chain of I (information) frames when instructed by the host.

(b) Independent generation and transmission of S (supervisory) frames.

(c) Generate and transmit U (unnumbered) commands when required and when instructed by the host.

(d) Independently generate and transmit U responses.

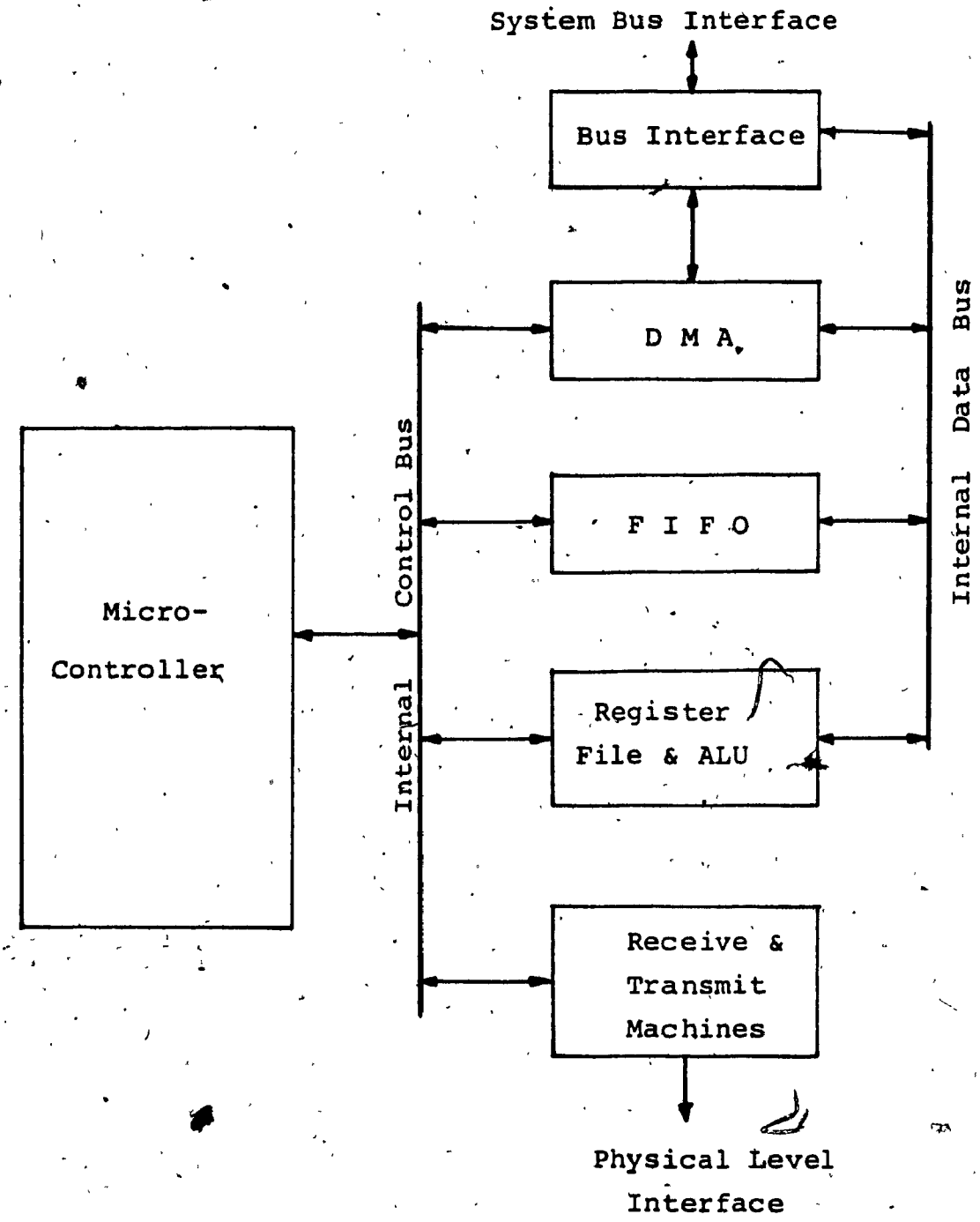


Figure 7.1

Block Diagram of Motorola's Communications Engine

To provide protocol flexibility, the XPC allows the user to program many parameters such as timers, the number of retransmission attempts and the number of outstanding frames. The host issues commands directly to the XPC via the command register. The commands are divided into four categories: initialization, table handling, link handling and test/diagnostic. The host controls the link using the start link and stop link commands. The test/diagnostic commands include register dumping, DMA transfer circuitry testing, serial circuitry loop back and external loop back.

7.2.2 Level 3 Chips

Placing level 3 in silicon chips is a much greater challenge than level 2 in terms of the operation complexity. Due to the many variations, the trend seems to be to implement a subset of the full packet level with some programmability to satisfy a large number of applications.

Another alternative which may be more cost effective, is to produce ROM-based level 3 products.

7.2.3 Network Design Trend

Without regard to whatever designs are pursued by various semiconductor manufacturers, it is very clear that VLSI/LSI implementation of standard protocols will be the standard of the industry. With this in mind, networks should be designed in such a way that new VLSI/LSI products

could be incorporated as and when it is feasible and economical to do so. This implies that standard communication protocols should be used as far as possible.

7.3 System Bus Up-grade

7.3.1 Transmission Medium

There exist four main types of transmission media generally used in today's LANs. They are twisted pairs, baseband coaxial cable, broadband coaxial cable and optical fiber cable. A comparison of the relative strengths of them is presented in Table 2.

CUENET's current twisted pair bus (C-bus) has an estimated throughput of about 1.2 Mbit/s. The next logical step would be to up-grade it to a coaxial cable based bus. The coaxial cable was initially rejected due to the high cost of the cable itself and its interface hardware. However, the recent introduction of Thin Ethernet (also known as Cheapernet) has reduced the per node cost to about U.S. $100 [Nelso 84]. Cheapernet typically delivers 10 Mbit/s for a segment length of up to 200 meters without repeaters. Up to 30 nodes per segment can be accommodated. Up-grading to a Cheapernet will be a big step forward for CUENET.

TABLE 2

Comparison of Various Local Network Transmission Media

Media	Bandwidth	Ability to handle many nodes	Distance	Noise immunity	Cost
Twisted Pair	Low	Low	Short	Low to Moderate	Low
Baseband Coax	Low to Moderate	Moderate	Medium	Moderate	Moderate
Broadband Coax	High	High	High	High	Moderate to High
Optical Fiber	Very High	Low	Very High	Very High	Very High

CHAPTER VIII

SUMMARY AND CONCLUSION

8.1 Choice of Protocol

The author has studied the design and operation of CUENET, its associated C-bus and protocol. It has been found that its major drawback is its lack of internetworking capability with other computer networks. The author recommends to implement a reduced subset of the X.25 packet switching protocol as its intranetwork communication protocol. The suggested modifications to X.25 would optimize the protocol to achieve the highest possible throughput from the system bus (C-bus) of CUENET. The use of the modified X.25 protocol in CUENET would facilitate its internetworking with public packet switching data networks or other computer networks. With the suggested protocol, its internetworking gateway will be greatly simplified. The suggested protocol will also align CUENET with the OSI reference model in terms of layer independency and openness for interconnection.

8.2 Protocol Software Architecture and Design

The author recommends the use of a protocol software architecture within which intra-CUENET communication software and internetworking software will share common codes. Those sections of software unique to either intra-CUENET or internetworking could be either executed or

bypassed, depending upon the situation. This scheme not only simplifies software development and testing but also would allow any of the CUENET processors to function as the gateway to the outside world.

In order to obtain the highest throughput possible from C-bus, a buffer management scheme is also recommended. This scheme eliminates the need to copy the packet from the sender's output buffer into the input buffer of the receiver. This copying process is very time consuming and reduces the C-bus throughput. The suggested scheme uses a common pool of packet buffers which will be dynamically assigned upon demand. The ownership of a packet buffer is marked. Ownership is changed from the sender to the receiver when a packet is sent from a sender to a receiver.

8.3 VLSI/LSI Protocol Implementation

The use of a standardized and widely supported protocol such as X.25 will ensure the adaptability to the use of specially customized VLSI/LSI chips and other interface equipment at reasonable costs. This will allow CUENET to evolve with new technology as and when it is available.

8.4 Bus Up-grade

The recommended system architecture would enable CUENET to up-grade its system bus to a higher throughput (10 Mbit/s) and longer distance (200 meters) by replacing its twisted pair based C-bus with a coaxial cable based

Cheapernet.

8.5 Meeting Design Objectives

The proposed protocol, software architecture and evolution strategies have fulfilled the following design objectives for CUENET-2.

(a) Enhance internetworking of CUENET.

(b) Align CUENET to comply to the OSI reference model.

(c) Adhere to existing communication protocol standards.

(d) Require minimum change to existing hardware and software.

(e) Provide a reliable and efficient network with adequate throughput.

(f) Facilitate growth and expansion.

(g) Ease implementation.

It is important to note that internetworking of CUENET cannot be achieved without changes to its messaging protocol and architecture. This is true for whatever method of internetworking is chosen. The OSI approach also needs such change, but the change, once made, permits CUENET to interwork with many other networks, which also comply with the OSI reference model. More and more other networks are supporting this same approach. With this recommended approach, CUENET will be opened up both in terms of accessing others and being accessed by others.

GLOSARY OF TERMS

AEFs	Address Extension Facilities of X.25
Cheapernet	A low-cost, 10 Mbit/s, CSMA/CD LAN having a coaxial cable segment up to 200 metres and up to 30 access points.
CCITT	International Consultative Committee on Telephony and Telegraphy.
CLNS	Connectionless network service; describes a set of services made available to OSI layer 4.
CLP	Connectionless protocol.
Contention	Interference between colliding transmissions.
CONS	Connection-oriented network service; describes a set of services, different from CLNS, made available to OSI layer 4.
Datagram	A message sent in a packet switched network.
D-bit	Delivery confirmation facility of X.25.
DCE	Data circuit-terminating equipment.
DTE	Data terminal equipment.
EOT	End of transport service data unit; a marker indicating the last bit of data received from the session layer.
FCS	Frame check sequence. A encoded value appended to each frame by the data link layer to allow detection of transmission errors in the physical layer.
Field	Subdivision of the physical frame allocated to convey specific information (i.e. address field, data field, etc.).
FTP	File transfer protocol.
Header	Control information at the beginning of a message, segment, fragment, packet or block of data.

HILI	High level interface; a working group (802.1) of IEEE 802.
IEEE 802	Institute of Electrical and Electronics Engineers - Project 802. The committee responsible for standardization of LANs operating at data rates of up to 20 Mbit/s
ISO	International Standards Organization.
LAN	Local Area Network. A data communication system supporting layers 1 and 2 of the ISO OSI Reference Model; having a geographic coverage of at least 1 km end-end; and possessing sufficient performance to support the aggregate data throughput required by the stations being served.
LAPB	Link Access Procedure-B of X.25.
LLC	Logical Link control; upper layer 2 sublayer protocol of IEEE 802.
MAC	Media Access Control. The medium-dependent sublayer of the data link layer, which interfaces with the physical layer.
M-bit	An X.25 feature used to indicate the last bit of data received from the transport layer.
MIF	Minimum Internetworking Functionality; a network that calls for as little complexity as possible in local network stations or devices for the relatively infrequent requirements for interconnecting with resources outside the local network; relegates such chores to a gateway.
NSAP	Network service access point; an abstract point at the boundary between layers 3 and 4 of the reference model through which network layer services (CONS or CLNS) are made available to the transport layer.
Octet	An eight bit word.
OSI	Open Systems Interconnection (ISO Reference Model).

PDU	Protocol data unit.
PLP	Packet level protocol of X.25.
Process	A program in execution.
PSN	Packet Switching Network.
TLC	Transport layer class; Class 0 through 4 of the transport layer protocol.
Type A,B,C	OSI network layer service classifications.

REFERENCES

[Adiba 79] M. Adiba et al. Issues in Distributed Data Base Management Systems. Issues in Data Base Management, North-Holland Pub. Co., 1979.

[Brinc 78] P. Brinch Hansen. Distributed Processes : A Concurrent Programming Concept. Commun. of the ACM, Nov., 1978.

[Burg 84] F.M. Burg, C.T. Chen and H.C. Folts. Of local networks, protocols, and the OSI reference model. Data Communications, November, 1984.

[Canes 83] F. Caneschi. The Role of the Session Layer in OSI Architecture. Communications Engineering International, April, 1983.

[CCITT 84a] CCITT Recommendation X.25, 1984.

[CCITT 84b] CCITT Recommendation X.121, 1984.

[Colli 86] C.M. Collins. VLSI Performance for X.25 Communications. Telecommunications, Mar. 1986,

[Cotto 80] I.W. Cotton. Technologies for Local Area Computer Networks. Computer Networks, Vol. 4, No. 5, 1980.

[Gilho 85] D. Gilhooly. Local Area Networks: Market Still Maturing. Communications Engineering International, June 1985.

[Gross 82] C. Grossner. The Design and Implementation of CUENET: A Reconfigurable Network of Loosely Coupled Microcomputers", Master's Thesis, Department of Computer Science, Concordia University, 1982.

[Hinde 83] R. Hinden, J. Haverty and A. Sheltzer. The DARPA Internet: Interconnecting Heterogeneous Computer Networks with Gateways. Computer, September, 1983.

[Hsiao 79] D.K. Hsiao et al. Privacy and Security of Data Communications and Data Bases. Issues in Data Base Management, North-Holland Pub. Co., 1979.

[IFS 79] IFS. Transmission Control Protocol 4.0 Specification. IEN 112, Information Sciences Institute, University of Southern California, Aug. 1979.

[ISO 83] ISO/TC97/SC16 - N1562 : OSI - Basic Reference Model.

[ISO 84] ISO/DIS 8208 : Data Communication - X.25 Packet Level Protocol for Data Terminal Equipment. Dec. 1984.

[Jones 79] A.K. Jones, R.J. Chansler Jr., I. Durham, K. Schwans, and S.R. Vegdahl. StarOS, a Multiprocessor Operating System for the Support of Task Forces. Proc. 7th Symp. on Oper. Syst. Principles, Dec. 1979.

[Mager 80] P.S. Mager. Alternative Architectures for Distributed Data Sharing : Functional Issues. IEEE COMPCON '80.

[McGov 80] J.P. McGovern and D. Basu. Middle Layers of Open Systems Interconnection : Session and Transport. IEEE COMPCON '80.

[Meist 85] B.W. Meister, P.A. Janson and L. Svobodova. Connection-Oriented versus Connectionless Protocols: A Performance Study. IEEE Transaction on Computers, December, 1985.

[Metca 83] R.M. Metcalfe and D.R. Boggs. Ethernet : Distributed Packet Switching for Local Computer Networks. Commun. of the ACM, Vol. 26, Jan. 1983.

[Moult 80] J. Moulton. High Level Protocol Boundaries in the ISO Model. IEEE 1980 Trends and Applications - Computer Network Protocols.

[Muker 80] J. Mukerji and R.B. Kieburtz. A Kernel for Supporting A Distributed File System. IEEE COMPCON '80.

[Nelso 83] J. Nelson. 802: A Progress Report. Datamation, Mar. 1983.

[Nelso 84] L. Nelson-Rowe. Vendors Applaud Move by IEEE Unit to Back Thin Ethernet Lan Standard. Communications Week, Aug. 13, 1984.

[Nowit 80] D.A. Nowitz and M.E. Lesk. Implementation of a Dial-up Network of UNIX System. IEEE COMPCON '80.

[Panzi 85] F. Panzieri, B. Randell. Interfacing UNIX to Data Communication Networks. IEEE Transactions on Software Engineering, October, 1985.

[Piatk 80] T.F. Piatkowski. The ISO-ANSI Open Systems Reference Model - A Proposal for a Systems Approach. Computer Networks, Vol. 4, No. 3, 1980.

[Powel 80] J.I. Powell, R. Fico, W.H. Jennings, E.R. O'Bryan and A.R. Schultz Jr.. A Local Network for Distributed Laboratory Microcomputers. IEEE COMPCON '80.

[Schne 83] Schneidewind N.F., 1983. "Interconnecting Local Networks to Long-Distance Networks", Computer, September 1983.

[Stall 84] W. Stallings. A Primer - Understanding Transport Protocols. Data Communications, November, 1984.

[Stall 85] W. Stallings. Can We Talk? Datamation, October 15, 1985.

[Stank 80] J. Stankovic. Debugging Commands for a Distributed Processing System. IEEE COMPCON '80.

[TG100 84] Department of Trade and Industry, U.K. Technical Guide TG 100/1: Intercept Recommendations for the OSI Network Layer. Mar. 1984.

[Thurb 80] K.J. Thurber. An Assessment of the Status of Network Architectures. IEEE COMPCON '80.

[VonTa 84] E. Von Taube. Internetworking: Connecting LAN's. Telecommunications, December, 1984.

[Warne 79] C.J. Warner. Local Network Transmission Control Protocol (LNTCP). Technical Note 793, Naval Ocean Systems Centre, San Diego, California, Dec. 15, 1979.

[Warne 80] C.J. Warner. Connecting Local Networks to Long Haul Networks - Issues in Protocol Design. IEEE 5th Conference on Local Computer Networks, 1980.

[Weiss 83] A.J. Weissberger. Bit Oriented Data Link Controls. Computer Design, March and April, 1983.

[Wells 83] M.F. Wells. Distributed Intelligent Architecture Improves System Performance. Digital Design, Jan. 1983.

[White 80] G.W. White and R. DesJardins. ISO/ANSI Reference Model of Open Systems Interconnection. IEEE 1980 Trends and Applications - Computer Network Protocols.

APPENDIX A

ISO OPEN SYSTEM INTERCONNECTION (OSI) REFERENCE MODEL

A.1 Introduction

Through the recent extensive research in computer networking and communication protocols, it has been realized that a common architectural reference model will be most useful to the future studies and developments of those two fields. The international standardization of the Open System Interconnection (OSI) reference model has been under development over the last few years jointly by the International Standard Organization (ISO) and the International Consultative Committee on Telephony and Telegraphy (CCITT) of the International Telecommunications Union. The OSI model provides a reference for the protocols, functions, services and interfaces required to interconnect application-processes within open systems. An application-process is the "end-user" of the information. It may be a human user or computer program. A sketch of the seven layered model is given in Figure A.1. Further details can be found in [ISO 83].

A.2 Definition of "Open System"

By "system", it means complete installations, i.e., data processing systems including peripherals, storage, front end processors, terminals, system and user software. In such a system, communication partners and a

communications system are united.

Such a system becomes an "open system" in the sense of the OSI reference model if it employs standardized protocols toward the outside, so that the communication partners contained in the system can communicate with any other open systems and the communication partners contained in them as long as they observe the protocol agreements.

A.3 Objectives and Uses of the OSI Reference Model

The OSI model is a general model which covers all communication protocols. The model is in itself not a standard for protocols, but it represents the framework necessary for the development and integration of protocols and interfaces for communication. It does not prescribe to any manufacturer what the technology of his products should look like. It merely prescribes how this technology should behave toward the outside environment.

For a successful interchange of information, both communication partners and the components of the communications system connecting them must provide many functions. This model for the first time defines which functions are actually expected from the components involved in a communication, and how these functions build upon each other. The OSI reference model plays a part in:-

(a) recognizing and interrelating standards, and if necessary, improving them;

(b) integrating existing standards and, if necessary, improving them;

(c) creating the basis for new standards;

(d) designing a collection of complementary, non-contradicting standards.

F.M. Burg et al [Burg 84] noted that it is the use of OSI protocols, not just any set of protocols functionally compatible with the reference model, that will make a system truly compatible with OSI.

A.4 The Seven Layers of The OSI Reference Model

The OSI model consists of seven layers as depicted by Figure A.1. It identifies all of the functions required to enable separate communication devices to communicate with each other. It then arranged those functions into seven groups, called layers or levels. The seven layers do not dictate the actual standards that should be used to bring about a network operation. Instead, they describe, in an implementation independent manner, the functions that each layer must perform. The seven layers of the OSI model are briefly described in the following.

Layer Seven, the Application Layer

This layer is responsible for bringing to the end user network services such as password checks, document transfers, and various protocols specific to an industry. It is the only layer that is not completely transparent to the user.

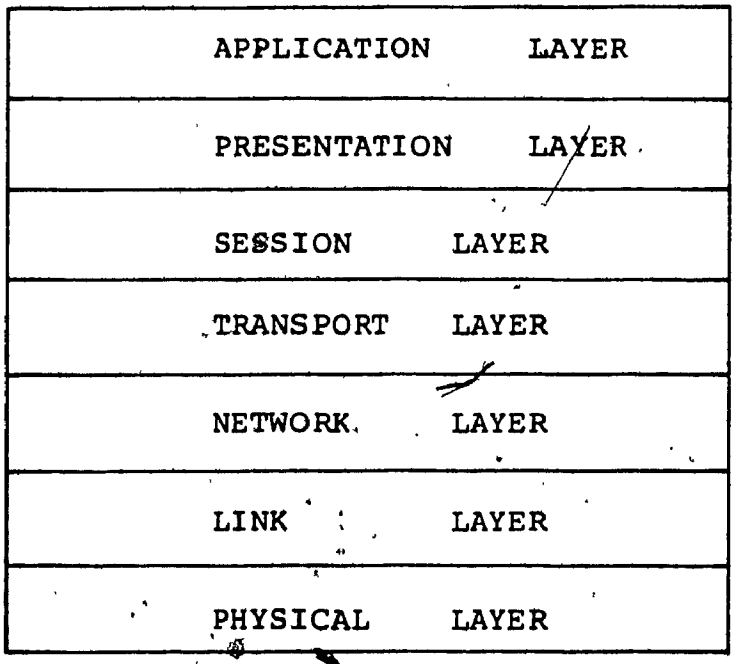


Figure A.1

The Seven Layered OSI Reference Model

Layer Six, the Presentation Layer

This layer converts the transmitted data into a form that can be used by the receiving device. After passing through this layer, data can be, for example, output by a printer as ASCII or EBCDIC characters.

Layer Five, the Session Layer

This layer is responsible for establishing and terminating connections between stations, and for mapping logical names onto physical addresses.

Layer Four, the Transport Layer

This is the layer that sees to it that a message (which could be made up of many frames) reaches its destination reliably. The transport layer establishes, controls and releases transport connections leading from one communication partner to the other and extends them into the interior of the end system.

The term "transport connection" is used for an imagined end-to-end connection between the two communicating partners. The real path "below" which the information to be transferred takes, is called a "network connection."

Layer Three, the Network Layer

This layer provides the functions of route selection, establishing the network connection path, sequence and flow

control, detection and correction of network errors, message segmentation and reassembly.

The main task of the network layer is to provide the OSI network service using the services available from underlying layers. The OSI network service is different from the underlying services in two very significant respects: it has to be independent of the communications media employed; and it has to be an end-to-end service provided between OSI end-systems across interconnected networks of various kinds.

Making the OSI network service independent of the communications media is part of the process of decoupling high-level operation from low-level so as to be able to perform any distributed processing activity over any communications media. In OSI this decoupling is achieved in two stages: the network layer gives independence of media in all things other than quality of service; it is then the task of the transport layer to make any enhancements to quality of service in order to satisfy the needs of users of the transport service [TG100 84].

Layer Two, the Data Link Layer

This layer deals with the composition of the frames of data sent out on the network, as well as the protocols for getting those transmissions onto the network and detecting transmission errors. Issues addressed here include the

format and length of each frame and the rules for how an individual station will obtain and relinquish access to the network.

Layer One, the Physical Layer

This layer deals with most of the hardware issues involved in networking. It describes the type of connectors which are required, data rates, signal levels and electrical and physical characteristics of the communications medium.

A.5 Benefits of Complying with the OSI Model

The OSI's multi-layered approach has brought two key features to network designs: modularity and upgradability. It allows for incremental improvements to a network without disrupting its entire operations. For example, an Ethernet-based LAN could be upgraded into a fibre optics based LAN without the need to change the rest of the design.

APPENDIX B

OSI AND ITS RELATIONSHIP WITH LOCAL NETWORKS AND IEEE 802 STANDARDS

B.1 OSI and Local Networks

ISO has been examining the application of the OSI reference model to local networks. Conceptually, the architecture of a local network does not differ from that of a wide-area network. The boundary between layers 4 and 5 (transport and session) is a key demarcation line. The protocols under this line are all concerned with reliable and cost-effective data transfer. Above this line, protocols perform common functions that are data-transfer independent, such as code conversion or the coordination associated with a file transfer. Therefore, the OSI protocol standards for layers 5 and above (session, presentation and application layers) apply just as much to an open system on a local network as they do to a system that uses any other networking technology. This would seem to indicate that the OSI reference model is equally applicable to both types of networks. In other words, nodes or processors attached to either a local or a long-haul network need to perform the same kind of functions, such as network access or file transfer.

There have been different opinions as to how the OSI reference model should treat local networks. One opinion regards each of the processors or nodes on the local network

as individual, distinct, OSI-supporting "systems." Each, in essence, would be individually "open". In this environment, the nodes, or systems, would be able to communicate with each other, as well as with OSI systems outside of the local network.

Another opinion is to regard an entire local network (including all nodes and devices collectively attached to it) as a single "open system." In this case, there would be little or no restriction placed on the internal protocols used by the individual local network nodes. However, it needs a gateway to communicate with other open systems outside of the local network. Also, there would be little or no restriction on the distribution of OSI functions and protocols within the local network, including the protocol between the gateway and the non-OSI local network nodes.

The third opinion is that the nodes on the local network would consist of a mix of both OSI and non-OSI equipment. In this scenario, non-OSI equipment would be able to share the local network medium with OSI systems with the assumption that both implement the same standard media access and control protocols. However, these two different groups would still be logically segregated from each other because of incompatibility at the higher layers.

The existence of these different opinions has created incompatible local network designs and hampered efficient

internetwork communications. These problems seem to have been recognized by network designers as there is a trend towards a compatible environment.

In favour of the creation of a universal compatible environment for computer data communication, the discussions in this report are based on the first scenario discussed above. That is, each local network node or processor is regarded as an individual open system conforming to the OSI principles and protocols.

B.2 OSI and IEEE 802 Standards

The work of the IEEE 802 committee has been concerned mainly with protocols for accessing and controlling the various types of local network media. As Figure C.1 shows, the protocol-standardization of the IEEE 802 corresponds to the lowest two layers of the OSI reference model.

The physical layer of the local network specifies the means for transmitting and receiving bits across various types of local network media. One or more physical layer specifications are associated with each type of media access control (MAC) sublayer. The MAC sublayer which is the lower sublayer in the IEEE's layer 2, performs those functions needed to control access to the physical medium. This use of sublayers in layer 2 of the IEEE 802 specifications is not in conflict with the guidelines of the OSI reference model. Sublayering is allowed in distinct

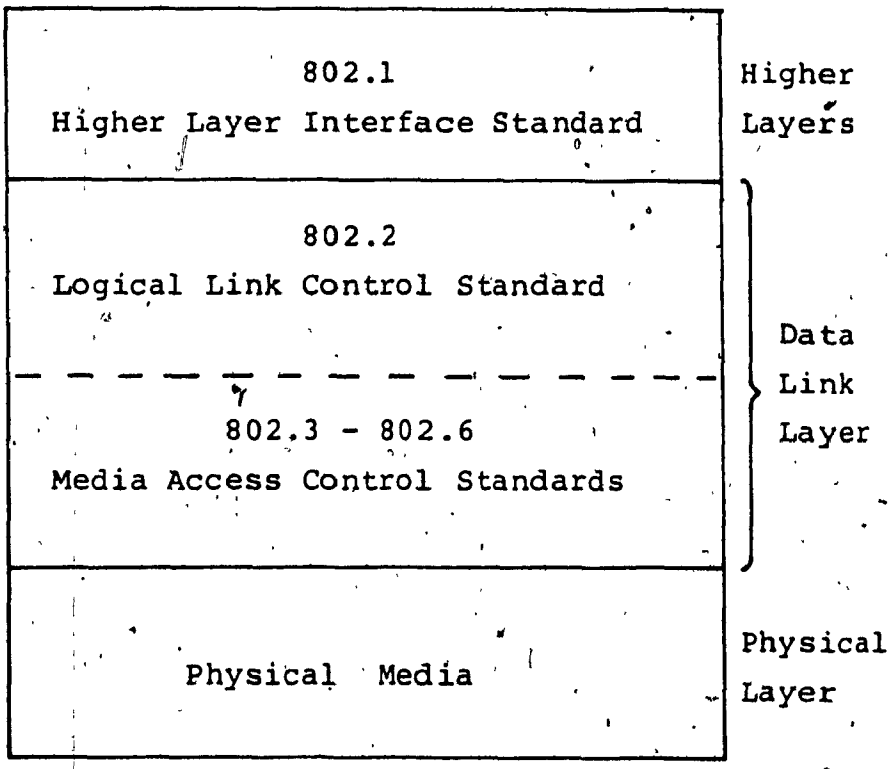


Figure B.1

IEEE 802 Standards Organization

cases where a group of functions is not always needed and can, therefore, be bypassed from time to time as necessary. However, in a local network environment, the functions of the MAC sublayer are always necessary to arbitrate access to the local network medium as it is shared by many "peer" systems.

The logical link control (LLC) sublayer is the upper sublayer in the IEEE's layer 2. It is defined in such a way that it is independent of the particular type of MAC procedure used. Three types of LCC procedures have been defined.

(a) LLC Type 1: Unacknowledged Connectionless Protocol

Data units are sent without any correlation to previous or subsequent data units and without any acknowledgement or guarantee of delivery.

(b) LLC Type 2: Connection-Oriented Protocol

This protocol provides a data link connection establishment procedure, transfer of multiple data units, acknowledgement, retransmission as appropriate, and the termination of the data link connection.

(c) LLC Type 3: Acknowledged-Connectionless Protocol

Single data unit is transmitted and then acknowledged, before a subsequent data unit is transmitted.

APPENDIX C

TRANSPORT LAYER CLASS OF PROTOCOL

The complexity of a transport protocol implementation depends on the nature of the underlying network service. ISO has defined three types of network service, which are used in specifying transport protocol standards:

(a) Type A: network connection with acceptable residual error rate and acceptable rate of signaled failures,

(b) Type B: network connections with acceptable residual error rate but unacceptable rate of signaled failures,

(c) Type C: network connections with residual error rate not acceptable to the transport service user.

In this context, an error is defined as a lost or duplicated network packet. If the error is caught and corrected by the network service in a fashion that is transparent to the transport entity, no damage is done. If the network service detects an uncorrectable error, it signals the transport entities. This is known as a signaled failure. There are also residual errors - those that are not corrected and about which the transport entity is not notified.

To account for differences among underlying network services, ISO has defined a family of five classes of transport protocols, all of which provide fundamentally the same transport service. The five classes and their relationship to the three types of network services is shown

in Figure C.1.

Class 0 provides the simplest kind of transport connection. It is assumed that a Type A, connection-oriented network service is available. Transport connections are mapped one-to-one onto network connections (e.g. an X.25 virtual circuit). No explicit ordering or error control is provided.

Class 1 is designed on an X.25 network and provides minimal error recovery (network-signaled errors). Its key difference from Class 0 is that it can resynchronize if some data are lost.

Class 2 is an enhancement to Class 0 and still assumes a highly reliable network service. The key enhancement is the ability to multiplex multiple transport connections onto a single network connection to reduce network cost, but the throughput may also be reduced.

Class 3 is basically the union of Class 1 and 2 capabilities.

Class 4 assumes that the underlying network service is unreliable. Thus the protocol must include elaborate mechanisms to deal with the loss or misordering of transport protocol data units (TPDU's). As an example, the U.S. Department of Defense (DOD) Transmission Control Protocol (TCP) is functionally equivalent to Class 4 Transport.

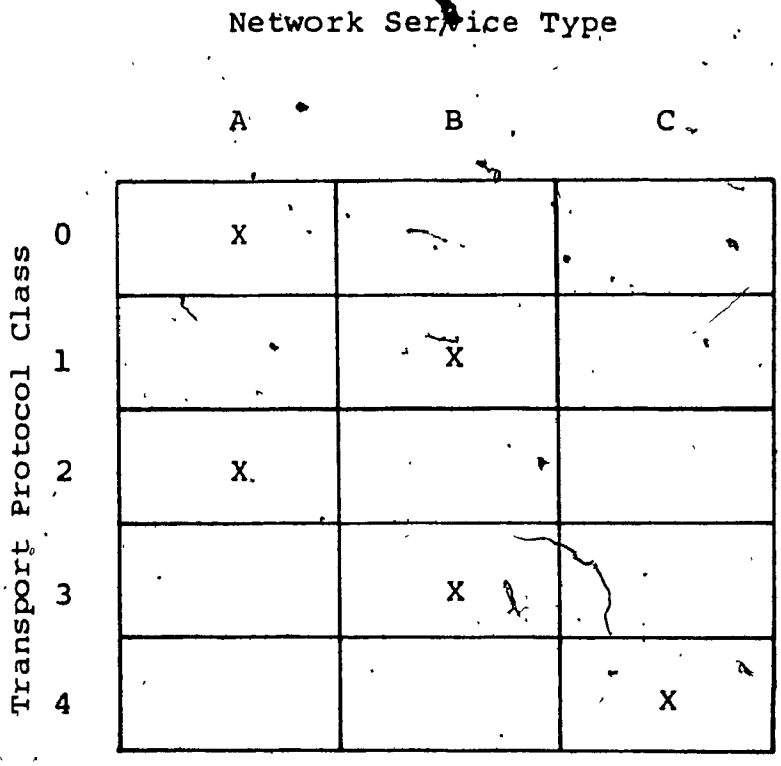


Figure C.1

Relationship Between Transport Protocol Class

And Network Service Type

Unfortunately, the TCP protocol [IFS 79] differs from that of ISO in such specifics as format and protocol procedures.

The transport service specification is the same for all classes since one main point of the transport service is to provide end-to-end data transfer, independent of the nature of the underlying network.

APPENDIX D

CONNECTION-ORIENTED AND CONNECTIONLESS PROTOCOLS

D.1 Connection-Oriented Protocol

A connection is an agreement between a pair of communicating entities to reserve a set of resources (such as buffers and programs) for their exclusive use, so as to maintain orderly communication and status information.

J. Nelson [Nelso 83] noted that making a connection is analogous to making a reservation at a high-class restaurant, which sets aside a table for the exclusive use of the parties who reserved it. The communication principles also have analogies in high-class restaurants: flow control is similar to making sure the courses arrive in correct sequence and on time, and data assurance is like making sure the food arrives properly prepared, and requesting a replacement dish if it is not.

Connection-oriented protocols have three distinct phases: establishment, transfer, and release. Connection establishment sets up a logical channel between two communicating entities, represented in each of these entities by a record containing data-transfer parameters and channel-state information. During the data-transfer phase, these parameters are used to control the addressing, formatting, sequencing, and flow of data units on the connection. Connection release destroys the state records

at both ends..

D.2 Connectionless Protocol

A connectionless service means that no agreement need to be made between a pair of stations before data can be transmitted between them. No buffering or other resources are reserved. If data arrives at a station that has no resources to buffer the incoming information, or if the station is busy for any other reason, the data is simply lost. The transmitting station is not informed of the loss and must depend on a higher-layer entity, perhaps with a time-out function, to determine that it has occured. The higher-layer entity must then attempt to retransmit the data.

A connectionless service is analogous to having a meal at a lunch counter [Nelso 83]. If no resources (stools) are available, the user must either go elsewhere to satisfy his needs, or wait until resources become available.

Connectionless services are defined to have none of the above characteristics of connection-oriented operation. There is only one phase of operation, no guarantee of preservation of sequence, no ability to exercise back-pressure flow control. Basic connectionless operation regards individual data units as unrelated to any data units that may have gone before or may come later. In the absence of any flow control mechanisms, it is a normal practice in

a connectionless operation to discard data in cases of congestion.