KALKUNTE, RAGHAVAN RANGACHAR

GATEWAY DESIGN FOR LOCALNET 20-TO-ARPANET-TO-LOCALNET 20 INTERCONNECTION

THE UNIVERSITY OF ARIZONA M.S. 1984

PLEASE NOTE:

In all cases this material has been filmed in the best possible way from the available copy. Problems encountered with this document have been identified here with a check mark __√__.

1. Glossy photographs or pages _____
2. Colored illustrations, paper or print _____
3. Photographs with dark background _____
4. Illustrations are poor copy _____
5. Pages with black marks, not original copy _____
6. Print shows through as there is text on both sides of page _____
7. Indistinct, broken or small print on several pages _____
8. Print exceeds margin requirements _____
9. Tightly bound copy with print lost in spine _____
10. Computer printout pages with indistinct print _____
11. Page(s) _____ lacking when material received, and not available from school or author.
12. Page(s) __34__ seem to be missing in numbering only as text follows.
13. Two pages numbered _____. Text follows.
14. Curling and wrinkled pages _____
15. Other _____

GATEWAY DESIGN FOR LOCALNET 20-TO-ARPANET-TO-LOCALNET 20 INTERCONNECTION

by

Raghavan Rangachar Kalkunte

A Thesis Submitted to the Faculty of the

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

In Partial Fulfillment of the Requirements
For the Degree of

MASTER OF SCIENCE

WITH A MAJOR IN ELECTRICAL ENGINEERING

In the Graduate College

UNIVERSITY OF ARIZONA

1 9 8 4

STATEMENT BY AUTHOR

This thesis has been submitted in partial fulfillment of requirements for an advanced degree at The University of Arizona and is deposited in the University Library, to be made available to borrowers under the rules of the Library.

Brief quotations from this thesis are allowable without special permission, provided that accurate acknowledgement of source is made. Request for permission for extended quotation from or reproduction of this manuscript in whole or in part may be granted by the head of the major department or the Dean of the Graduate College when in his judgement the proposed use of the material is in the interests of scholarship. In all other instances, however, permission must be obtained from the author.

SIGNED: Raghavan Kalkunte

APPROVAL BY THESIS DIRECTOR

This thesis has been approved on the date shown below:

Ralph Martinez — 7 Sept 84

RALPH MARTINEZ
Associate Professor
Electrical and Computer Engineering

Date

Acknowledgements

I would like to gratefully acknowledge the sincere and helpful guidance of my advisor, Dr. Ralph Martinez, throughout the development of this thesis. It is through Professor Martinez's professional interest in computer networking that this project had its inception. I would like to extend my thanks to Dr. M. K. Sundareshan and Mr. K. H. Muralidhar, members of the gateway design team, for the stimulating discussions we had on the various aspects of this research. Many of the ideas presented in this work were born out of these discussions. My thanks also to Mr. David Kaufman of Sytek, Inc., for the beneficial suggestions on the gateway design at several stages of this project. Special thanks to Sytek, Inc., for providing the manuals and papers from which I have drawn heavily in this thesis.

To Dr. Jack Sheppard of the U.S. Army Information Systems Command. A portion of this work was partially supported by the above organization under contract to the University of Arizona.

I would like also to thank the other committee members, Dr. Robert Swanson and Dr. Olgierd Palusinski, for their suggestions.

Last but most important, I would like to thank my parents, Devanath and Lalitha, my sister Jyothi, my brother-

in-law Raghuram, and finally my wife Asha, who came into my life at a very crucial juncture, which sustained my enthusiasm and motivation during the completion of this work. I am grateful for their support, patience, sacrifice, and love. This work is dedicated to them.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

ABSTRACT

Gateways are special machines or computers that provide the interface between interconnected computer networks. Incompatibilities exist among different types of networks in existence today. Communication formats and protocols could differ between networks to be connected. These differences cause interconnection problems such as routing, addressing, flow control, packet formats, and protocols. The function of the gateway is to perform protocol translation that will allow system elements on different networks to understand and communicate with each other.

In this thesis a specific design of a gateway between Sytek LocalNet 20 broadband local area network and the ARPANET, a long-haul network, is described. The primary objective of the gateway is to provide an interface to the ARPANET so that two remotely located LocalNet 20 networks can communicate with each other via the ARPANET. The ARPANET provides the communication subnet for the transportation of packets between the remote locations.

CHAPTER 1

INTRODUCTION

Current trends in networking guarantee an increasing number of network systems and users. While the bulk of the activity may remain local to individual networks, users will desire even wider access to resources on other networks. The goals of local area networks (LANs) are to allow sharing of programs, data, and other resources like printers, plotters, storage devices, etc., among all users on the network. The span of most local area networks is currently limited to a maximum distance of about 50 kilometers. The motivations that led to local area networks in the first place are also the motivations for interconnecting them. Examples of LANs are Ethernet, Sytek's LocalNet 20, and Ungermann-Bass's Net-One, to name a few. Long-haul networks, like local area networks, allow computer resources to be shared, but over a greater geographic area, sometimes spanning different continents. The purpose of long haul networks is to interconnect computers at various centers, geographically dispersed, in such a way that users and programs at one center can have access to the data, programs, and computing power at any other center. The ARPANET and the Telenet are examples of long haul networks and have been in use for some

time now. Today there has been an increasing number of local area networks in use at universities, industries, commercial offices, military installations, and many other environments. A large number of LANs belonging to the same organization are found distributed, sometimes in adjacent buildings and sometimes elsewhere across the continent. It is becoming evident that there is a need to interconnect these LANs via microwave links, satellites, or long haul networks. Recently long haul networks have been successfully used in providing interconnection between remote LANs and other long haul networks. Such an interconnection using long haul networks can provide:

(1) LAN to LAN communication. Under this, two types of network interconnections can be identified:

(a) Interconnection among homogeneous LANs, and

(b) Interconnection among heterogeneous LANs.

Type (a) interconnection arises because the homogeneous networks are geographically dispersed. These networks may belong to the same organization. One may look upon such an interconnected network system as one single large network. Each of the individual LANs can be thought of as a dispersed segment of the same network. Type (b) interconnection arises because of the existence of several LANs from different manufacturers. Different companies, divisions, or departments may have different types of LANs, but may need to be interconnected for resource sharing purposes.

(2) Long haul network to LAN interconnection. For example, users on a LAN may want to communicate with other users on a long haul network like ARPANET or Telenet.

(3) Long haul to long haul interconnection. Users on two different long haul networks may desire communicating with one another.

It is evident that there is an ever-growing need for users to communicate with one another to share distributed resources and information, which calls for an urgent need for developing systems to provide interconnections between different types of networks. Interconnection of networks involves providing the hardware and software so that user nodes on one network can easily, conveniently, and effectively communicate with nodes on another network. The interconnection may be between homogeneous networks (networks from the same manufacturer) or between heterogeneous networks (networks from different manufacturers). An example of a heterogeneous interconnection may be connecting an Ethernet-type network with a broadband Sytek LocalNet 20 network. The biggest problem in such heterogeneous network interconnection is the protocol conversion between the interconnected networks. The protocols used on the interconnected networks may differ considerably from one another. Examples of such protocols could be Sytek's LocalNet 20 protocol, ARPANET's transmission control protocol, and X.25

packed-switching network protocol, to name a few. User devices on one network do not communicate according to the protocol consistent with the protocol of the other network, but rather according to its own network protocol. Hence, if two nodes on two heterogeneous networks have to communicate with one another, protocol conversion will be necessary if the two nodes are to understand each other. The problem of internetting is one of examining alternative strategies for mapping of protocols between different types of networks. The mechanism of protocol conversion is effected at the point of contact between the two networks by the gateway, a special node which can be thought of as belonging simultaneously to the two networks. The gateway has to capture the incompatibility between protocols on the two networks. The gateway communicates with nodes on each of the different networks with their own protocols and performs protocol translation at any of the required layers of the International Standards Organization (ISO) reference model for the Open Systems Interconnection (OSI) (Zimmerman 1980). Other issues confronting the researchers in network interconnection are ways to solve incompatibilities between networks, besides protocol differences like inherent differences in characteristics like bandwidth, error rate, network topology and time delays that exist between them. Gateway designers have chosen specific solutions to implement an interconnection to suit their needs, but there are no

standards as yet. Thus, as of today, a gateway is generally implemented as a special machine dedicated to a specific interconnection between networks.

Gateway Model

A gateway appears to each network as a node on that network. In this model, a gateway is indistinguishable from any other network node and will implement whatever node/network interface is required by the network to which it is attached. Figure 1.1 shows a gateway model as described above. It is convenient to view the gateway as consisting of two halves. One half implements the protocols associated with network A and the other half implements the protocol of network B. In addition, the gateway will, if necessary, do the protocol mapping between the two networks. It is easy to see that with the above model of a gateway, an interconnected system of networks from a user's viewpoint is not different from a single large network. Thus gateways provide an enlarged number of users (Figure 1.2) with additional services. This simple notion of the gateway is very important. The user, desiring access to resources on another network, should not be bothered by the interconnection problems and must be able to access the resource as if it were located on his own network.

In general, not all the levels in the hierarchy of the protocols are affected by internetworking.

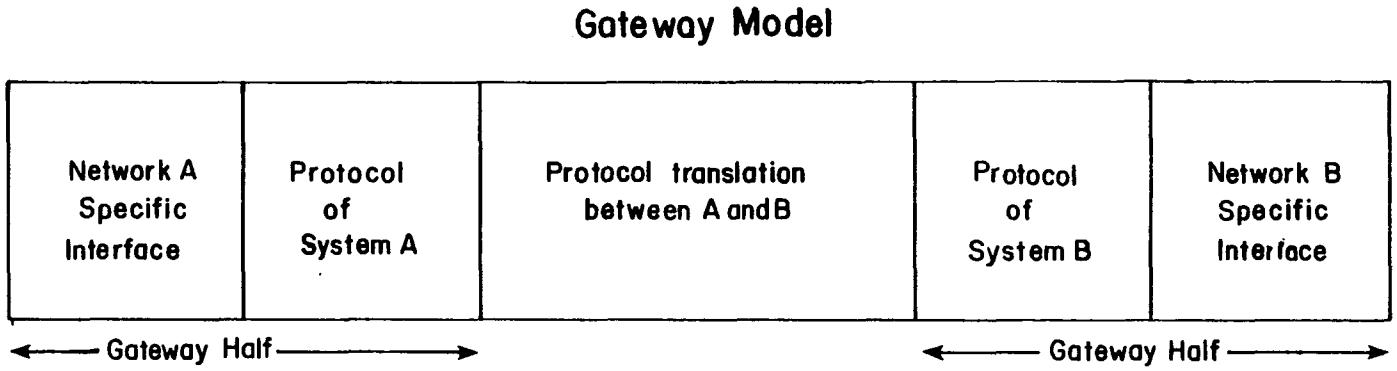


Figure 1.1. Gateway Model.

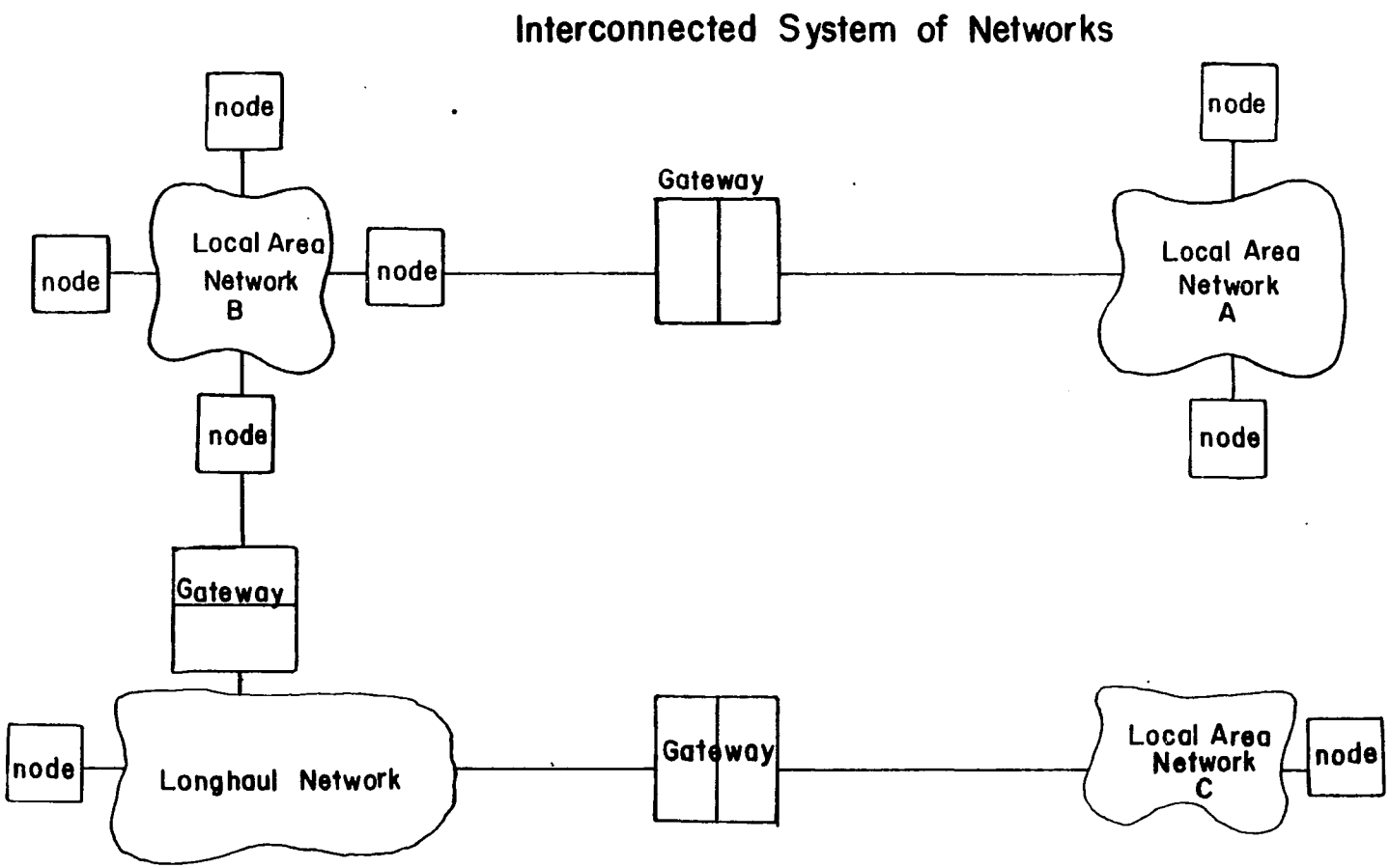


Figure 1.2. Interconnected System of Networks.

Specifically, the presentation and the application layers are so complex that interconnection of different protocols at these levels is currently not performed. The levels most affected by internetworking are the levels responsible for datagram and virtual circuit connection, namely the network, transport, and session layers. These levels must provide for internet mapping of addresses, flow control, error control, establishment of connections, etc.

Types of Gateways

Two types of gateways may be identified, depending on their functions: (1) media translation gateways, and (2) media and protocol translation gateways (Estrin 1982). Media translation gateways connect two different networks and are implemented using different media access methods such as a broadband coaxial cable and leased digital communication lines. The two networks may have the same protocols above layer 2 (Figure 1.3). The gateway will interpret layer 1 and layer 2 protocols of the ISO reference model and will translate one to the other. As the higher level protocols are the same and can be understood by the communicating nodes on the two networks, no higher level protocol translation will be required. As an example, a baseband carrier sense multiple access with collision detection (CSMA/CD) type network and a broadband token passing network may both have the same protocol above layer 2. Here the gateway needs to perform only the physical and data link layer

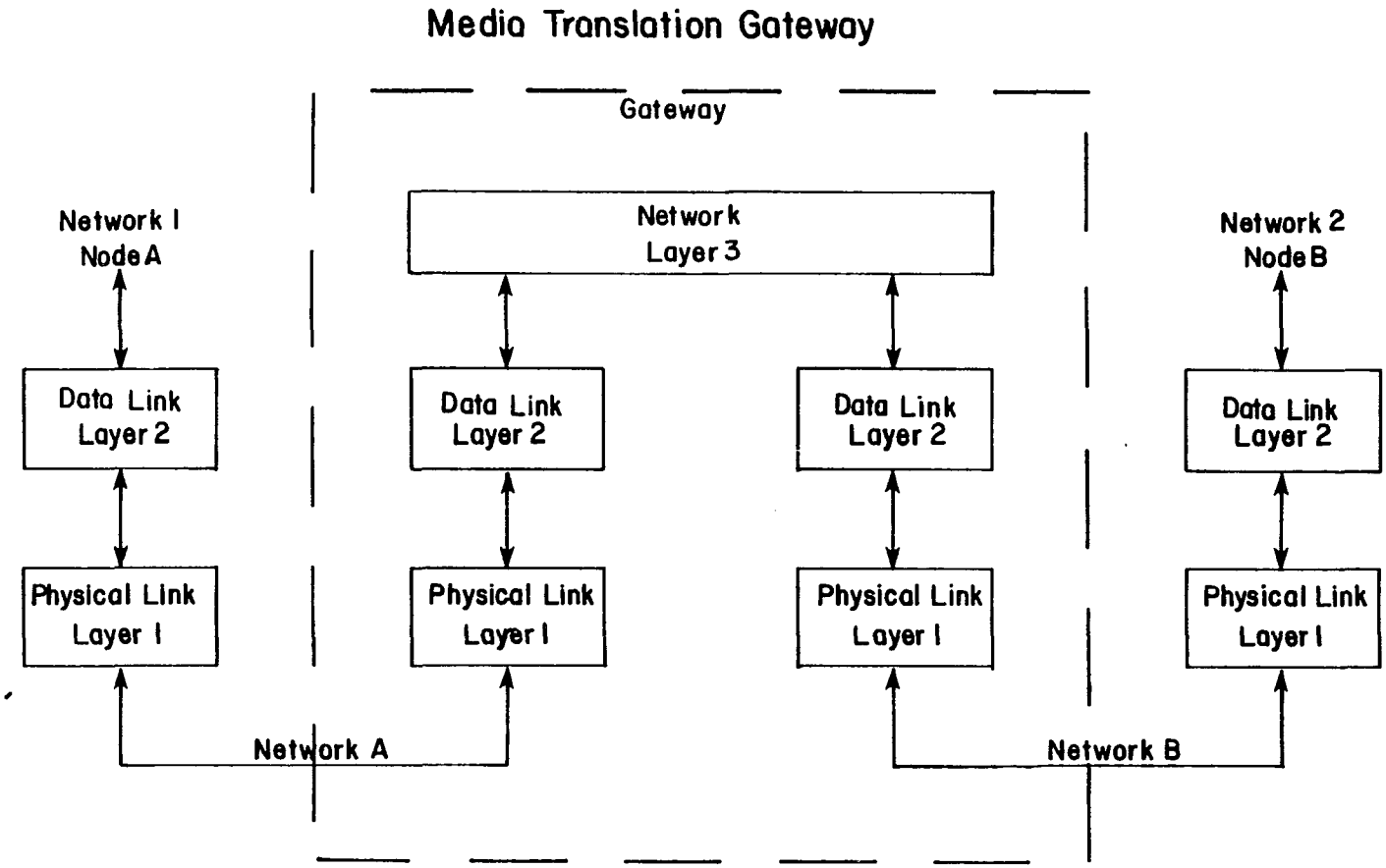


Figure 1.3. Media Translation Gateway

translations. Media translation gateways are quite simple to implement. The media and protocol translation gateways (Figure 1.4) have to translate the protocols above layer 2 used in one network to those used in another network, in addition to the media translation at layers 1 and 2. Such gateways communicate with each network with their own protocol and translate the protocols between the two networks. Generally these gateways provide protocol translation up to and including the transport layer. The first four layers of the ISO reference model are quite well understood, and are standardized. The protocol layers above the transport layers have not yet been clearly defined and established. The top three layers are quite complex and are less formalized than the lower level protocols. There is less likelihood of one-to-one translation of the higher layer protocols at the present time. Several incompatibilities exist at these layers and not many higher layer protocol translation gateways have been implemented thus far.

It is apparent from the above discussion that there is a growing need to interconnect different types of networks via gateways. The lower levels of the ISO reference model, namely the physical and the data link level interconnections, do not pose any real problems. The highest three levels, the session, presentation, and application levels, are highly incompatible, involve complex protocol conversions, and are difficult to implement at the gateway.

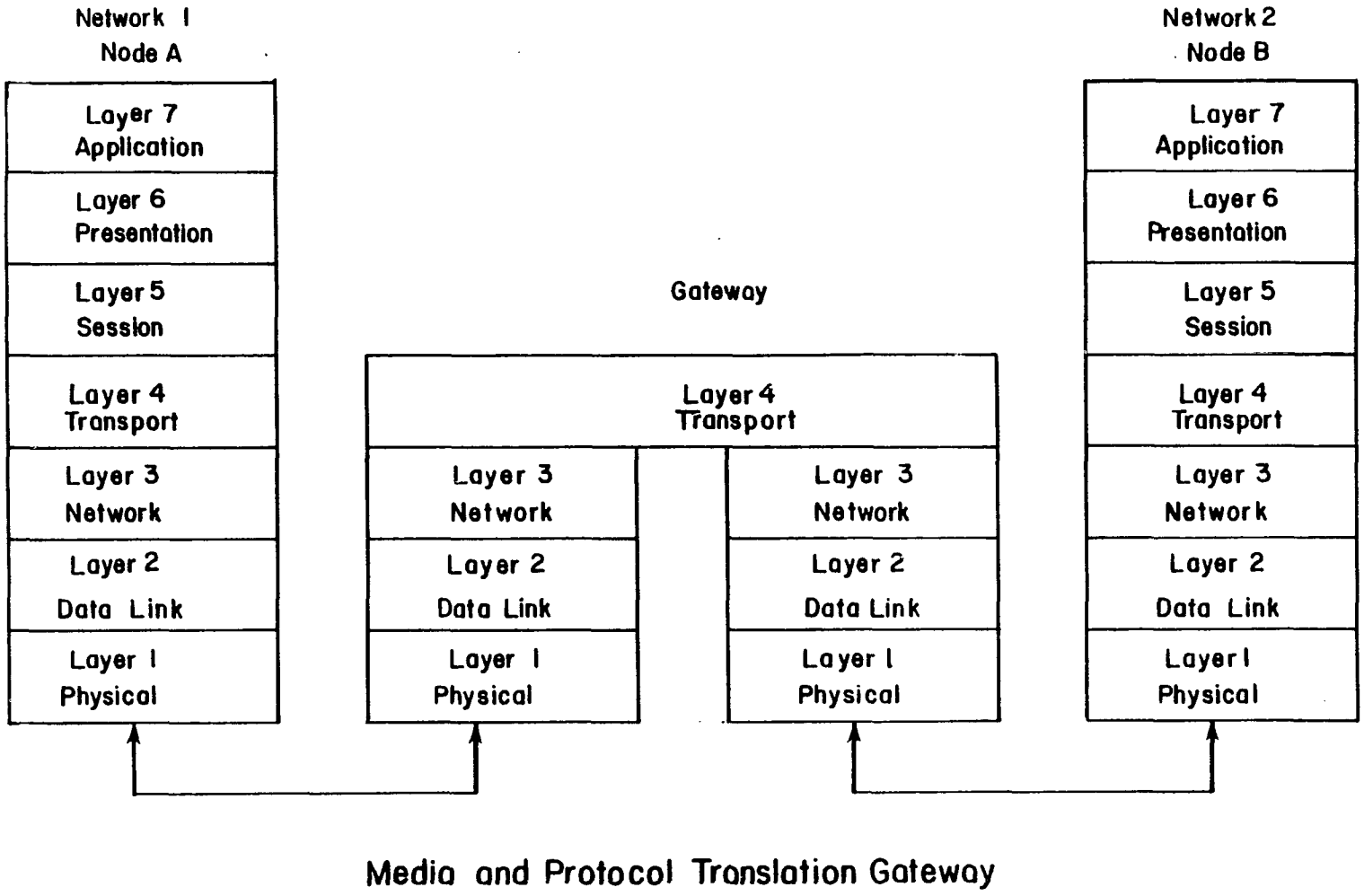


Figure 1.4. Media and Protocol Translation Gateway.

This leaves us with the network and the transport layers that are of immediate concern to gateway designers.

CHAPTER 2

ISSUES IN NETWORK INTERCONNECTION

There are several problems that need to be considered before determining the type of interconnection between networks. The heterogeneous nature of networks makes it impossible for users on one network to communicate with users on another network. Interconnecting these heterogeneous networks and solving the problems of interconnection is one of the biggest challenges to the people in the computer-communications research area. In this chapter we will discuss, in particular, some of the issues involved in interconnecting the high speed local area networks and the relatively slower long haul networks like the ARPANET. The gateway under development at the University of Arizona belongs to this type of interconnection. The focus is on some of the major issues in network interconnection.

Network Bandwidth

Local area networks in general offer very high bandwidths, typically in the range of 1 megabit per second to 10 megabits per second and higher. On the other hand, long haul networks offer bandwidths up to 56 kilobits per second and are much slower. Obvious mismatches in operating speeds exist between these types of networks. The problem here is to determine a way of connecting the different speed

networks. The gateway must cope with the speed differences and should essentially act as a store-and-forward system, buffering data between the two networks. The difference in bandwidth affects two very important design issues, e.g., in determining the ways in which flow control and packet acknowledgements are handled.

Flow Control

Flow control is concerned with a pair of nodes trying to ensure that the rate of transmission of packets from the source shall not exceed the capacity of the destination to receive the packets. The sender can create packets faster than the destination is able to receive or process them. In an internetwork environment, flow control is essentially between a source node and the gateway. The gateway may not be able to process packets as fast as it receives them because of lack of resources like buffers or CPU time. The gateway must protect itself against over-filling its packet buffers and overtaxing its processing capacity. The gateway must in some way control the rate of the sender. A virtual connection service between the source and the gateway will help implement the flow control. In a virtual connection, the sender will ascertain if the gateway is ready with the required buffers to accept the packets before actually sending them out. A simpler solution would be to discard the packets if there is no buffer space, but

send some information to the source indicating that the packet had to be discarded. Another possible approach would be to discard the packet and let the source time out and retransmit the packet.

Congestion Control

Congestion control is used to control the number of packets arriving from all sources at the gateway and preventing it from being overloaded. As internet traffic increases, a gateway could encounter a period of severe congestion. A slow gateway will not be able to cope with the sudden burst of traffic. A gateway must have greater processing power and buffer capacity to encounter the sudden traffic situation. Congestion may also occur because an adjoining network may be slow to accept packets from the gateway. The gateway is bound to be a bottleneck if there exists much internet traffic. ARPANET gateways implement a simple strategy. If a packet entering a gateway cannot be found a buffer space, then the packet is discarded and the source is notified of the cause of the discard. Retransmission is expected from the source. This approach is simple but leads to inefficient utilization of the network bandwidth and also a large portion of the source resources are used up for retransmission. Perhaps a more suitable form of congestion avoidance is the one suggested for the Cyclades network. The basic idea is that a gateway lacking in buffer space signals the source nodes contributing to

internet traffic to cut down their contribution. However, this approach requires a complex scheme which has to keep track of when the gateway buffers are getting full.

There are several approaches. The choice and implementation depends on the type of interconnection and the amount of traffic involved. Thus the problems of flow control and congestion control are apparent when gateways join high speed LANs to long haul networks. The transmission speeds of LANs may exceed the transmission speeds of long haul networks by several times. These problems are further amplified in a large internet environment involving several networks of different operating speeds.

Routing

LANs require very simple routing mechanisms. This is essentially due to the simple network topology used in LANs. The most common network topology in LANs is either the bus or the ring topology. In a baseband network with bus topology there is only one possible path between the nodes on a network. The routing algorithm is quite simple, requiring no routing tables as a necessary feature. Long haul networks, on the other hand, have complex network topologies. The commonly used topology is the point-to-point like in ARPANET. The routing scheme must determine the outgoing lines of the packet. The routing scheme is greatly influenced by the network topology. If a gateway

happens to connect a bus type LAN and a point-to-point long haul network, then the major gateway decision in routing is to determine through which directly connected node/network the packet must be transmitted. The gateway must determine an immediate destination which is an address on that network of either the ultimate destination or some intermediate gateway, believed to be closer to the destination, if the destination happens to be on another network. Thus a gateway connecting to a point-to-point network must execute the routing protocol on every outgoing packet. The most commonly used method of selecting routes is the hop count, the number of intermediate networks between the source and the destination. Each gateway must maintain a table having the delay to some number of destination networks that are one or more hops away, and the identity of an internet gateway by which they can be reached. The internet delay indicates the delay experienced by a packet to reach a destination via the gateways. In a point-to-point interconnection, gateways must maintain the image of the entire internet. For a large interconnected system of networks with complex network topologies, this approach becomes very inefficient. First, it is very time consuming and inefficient for the gateways to hold and update an image of the entire internet. Second, large internets may have different bandwidth characteristics which may result in accumulation of packets at the gateway, resulting in congestion. Information about delay-bandwidth

characteristics must be exchanged between gateways if the gateways are to pick an optimal route to the destination. We see that the complexity of the gateway increases significantly if it needs to connect to a long haul point-to-point network on one side. Its routing function on the LAN side, which could be a simple bus topology network, is minimal. It merely has to broadcast the packet on to the destination network and no routing information has to be maintained.

Addressing

The addressing schemes used on local area networks and long haul networks differ a lot. Many LANs seem to use the flat addressing scheme. In this approach, each node on the internet is assigned an absolute or unique address which is universal over the entire address space. Many long haul networks use the hierarchical addressing method. In this approach, each node is assigned a network-specific address. The network-specific address of the node combined with a unique network number produces an unambiguous internet address. In the hierarchical approach, each node has a unique address on its network, but this does not preclude the possibility of another node on another network having the same address. Obvious problems in interconnecting a long haul network with a hierarchical addressing scheme and a LAN with a flat addressing scheme is the possibility of two nodes having the same address. It may be necessary to include another address parameter in the form of network

number into the addressing scheme of the local network protocol in order to provide an unambiguous node number within the internet. It is also not certain how a gateway would be able to distinguish between an internet address and a local address.

Fragmentation

Local area networks have a smaller packet size than long haul networks. Long haul network packets have a larger packet size and carry a bigger header which carries information about complex routing and addressing schemes. Because of the complex network topology of long haul networks, the large number of nodes on them, long delays in end-to-end transmission, complex flow control and congestion control mechanisms used, and routing functions involved, the message size is larger, due to the greater amount of information carried in the header. LANs, because of their inherent simplicity, do not require such complex header information. The difference in maximum message size of the packets will introduce the need for the gateway to perform packet fragmentation and reassembly. For example, if a long haul network with a larger packet size wants to communicate with a LAN with a smaller packet size, then the gateway must fragment the larger packet. Another issue is to determine where the fragmented packets will be reassembled. The gateway now has to provide a scheme to number all the packets

generated by fragmentation. The gateway must also determine the size of the fragmented packets to ensure that they are large enough to hold both the copy of the header and a portion of the data. Another issue to be considered is whether the gateway should combine several small packets it receives from the LAN side before sending them out on the long haul network. If the gateway transmits individually each of the smaller packets it gets from the LAN, it would result in underutilizing the bandwidth of the long haul network. We see that the difference between the maximum packet size of LANs and long haul networks further complicates the role the gateway plays in the interconnection.

Protocols

The protocols used on LANs and long haul networks differ considerably. The protocols used in LANs are uncomplicated and are tailored for speed. Long haul network protocols are much more complicated. The major role the gateway plays in LAN to long haul network interconnection is protocol conversion. Protocol differences could exist in the form of different message formats like packet size, addressing scheme, acknowledgement methods, error control methods, flow control schemes, etc. The primary objective of protocol conversion is to facilitate communication between users on two diverse types of networks with inherent incompatibilities. The gateway has to map from the long haul protocol to the LAN protocol and vice versa. A very

important issue is to determine the degree of conversion required, which depends on the differences between the protocol layers of the interconnected networks. For example, a LAN and a long haul network may both be implementing the same type of protocol above layer 2 of the ISO reference model. In this case, the gateway functions as a media translation gateway. It interprets each network's layer 1 and layer 2 protocols, where the incompatibility exists, and maps one to the other. If incompatibility in protocol exists at higher layers, then a media-and-protocol translation gateway would be required. The gateway must interpret the different higher layers and map one protocol to the other.

Error Rate and Time Delays

LANs have relatively lower transmission errors and lower delays than long haul networks. LANs operate under less hostile conditions. Long haul networks use telephone lines, microwave links, etc., which are more susceptible to noise. These factors have significantly affected the design and construction of protocols used on these networks. LAN protocols have exploited the fact that LANs provide low error rate and low delay rate and are designed to minimize processing overhead normally involved with complicated header formats. The design of LAN protocols and their implementation are governed by the above factors, and are

tuned to LAN characteristics such as speed. Long haul network protocols are more complicated and include parameters to provide reliability in communication for the fact that they have inherent problems such as greater delays and higher error rates. The above differences usually result in implementation of different protocols in LANs and long haul networks. Incompatibilities in different network protocols have the utmost effect on gateways, and protocol translation is the most complicated task performed by the gateway system.

One possible solution to the problem of having incompatible protocols is to use standard internet protocols throughout the interconnected network system. Each LAN appears as just another network in the internet. ARPA's Internet Protocol, for example, could, be used as a standard protocol throughout the LANs and long haul networks, independent of the network technology, providing a common communicating language for all internet packets. This would reduce the protocol translation problems at the gateway. However, using Internet Protocol throughout may add a substantial overhead in terms of header processing for each packet, and would introduce additional delays due to packet processing which may not be comparable to the underlying local network speed. We see that the inherent difference in network characteristics introduces several interconnection problems in interconnecting long haul networks and LANs.

Each gateway has to be designed to meet specific applications. Therefore, the gateway cannot be conceived as a machine that can provide a general solution for overcoming the problems of incompatibility, introduced as a result of inherent differences in network characteristics and reinforced by different network architectures from different vendors.

CHAPTER 3

OVERVIEW OF SYTEK'S LOCALNET 20 AND ARPANET PROTOCOLS

The gateway under development interconnects Sytek's LocalNet 20 broadband local area network and the ARPANET network (Heart 1982). Each of the networks has its own protocol. ARPANET uses the Internet Protocol and the Transmission Control Protocol and Sytek uses its LocalNet Protocol (Ennis 1983). In this section, an overview of the LocalNet 20 Protocol and the Internet Protocol will be discussed and should provide the necessary background for understanding the design and operation of the gateway in the subsequent chapter.

LocalNet 20

LocalNet 20 is a packet switching local area data communication network providing computer communication functions via broadband cable television (CATV) data distribution system. The properties of the broadband CATV permit LocalNet 20 to construct many independent subnetworks termed channels on a single CATV coaxial cable system. Each of these subnetworks provides data communication to hundreds of user devices. The broadband system appears to have a bus type structure from a user's point of view. Standard frequency modulation methods are used on RF carriers throughout

the LocalNet 20 portion of the RF spectrum, creating many independent digital information channels through frequency division multiplexing. Every user on LocalNet 20 is assigned a specific channel. Each channel is time division multiplexed among several hundred other users through the use of packet switching. These frequency channels form the first level of connectivity. At the second level, LocalNet 20 allows communication between users on different frequency channels via a channel intercommunication product called the "bridge." LocalNet 20 uses the carrier sense multiple access with collision detection (CSMA/CD) media access scheme to provide a fair and efficient allocation of a channel's bandwidth among its users. The LocalNet 20 network used in the gateway development supports 120 channels, each with a data rate of up to 128 kilobits per second, and provides packet switched service to thousands of low duty cycle devices like terminals over a geographic area of up to 20 kilometers.

Packet Communication Unit

Terminals and computers are connected to the LocalNet 20 broadband network via packet communication units (PCUs). The Local Net 20 PCUs provide low cost communication to low-throughput users (less than 20 kilobits/second). The PCU (Figure 3.1) is connected between the user terminal or computer and the broadband coaxial cable, and provides

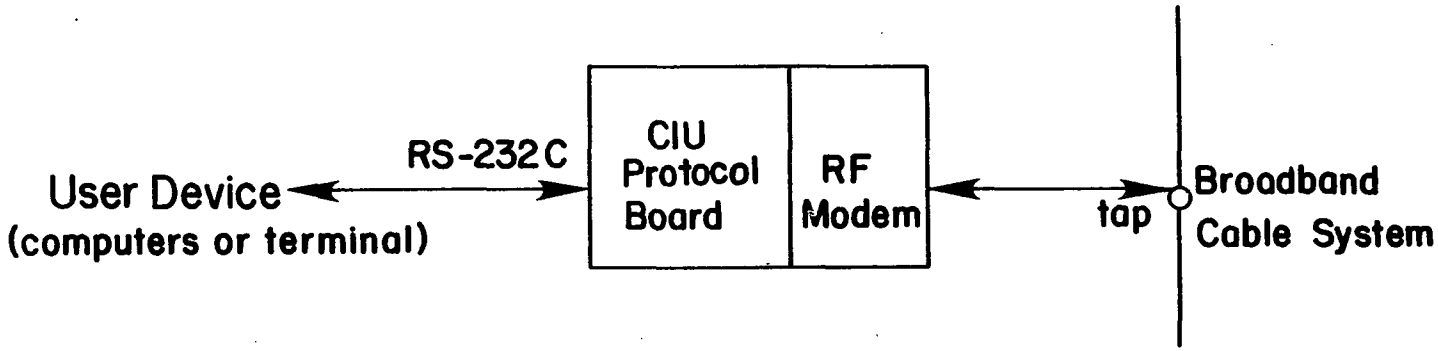


Figure 3.1. Packet Communication Unit.

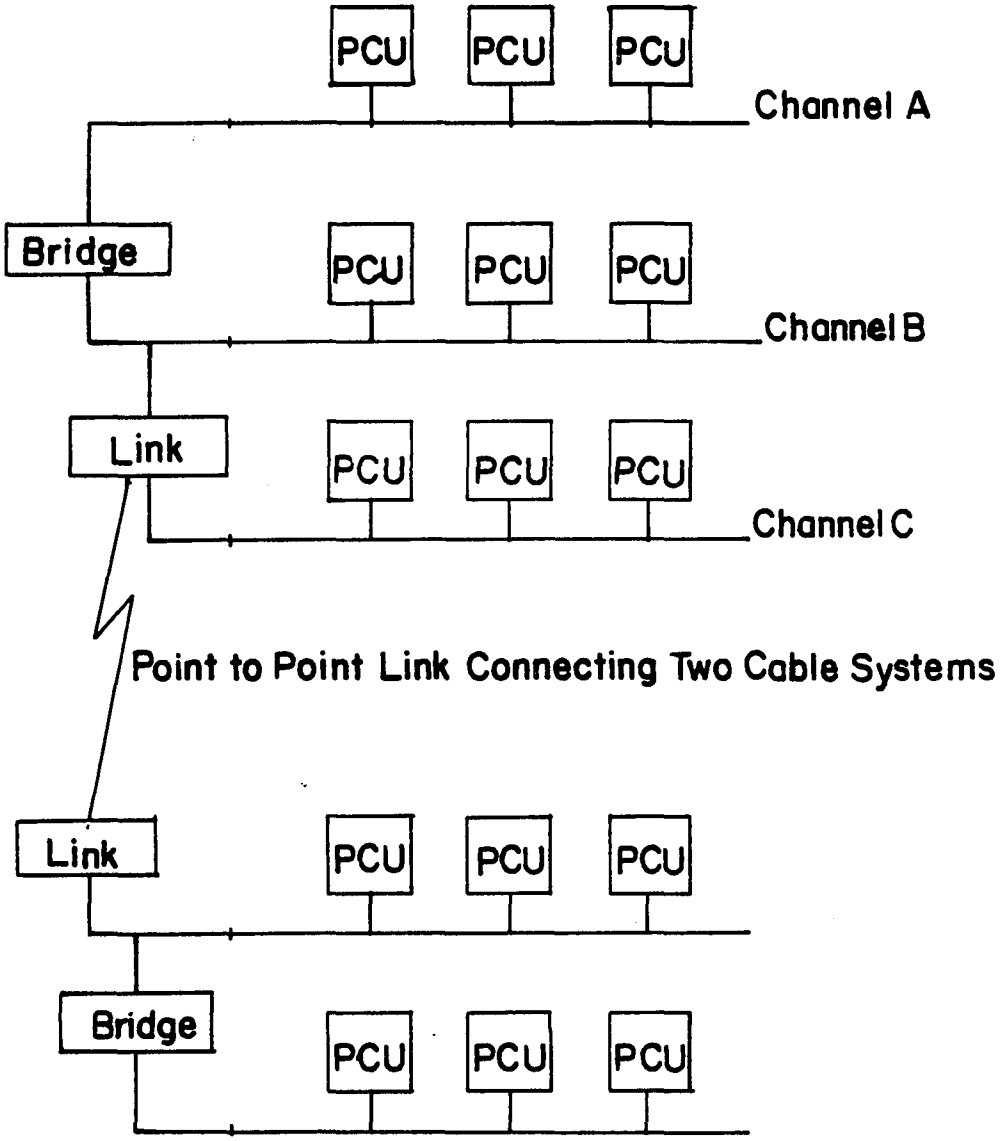
communication functions over the LocalNet 20 with several other remote terminal resources. Specifically, PCUs provide full duplex data transfer between user devices over packet switched sessions. The PCU provides all the necessary services such as packet assembly and disassembly, buffering, session management, error and flow control, speed matching between PCU and user device, data security, and media access function. In brief, all the required communication intelligence at each user interface point is built into the PCU. The above mentioned user services are provided by implementing a common packet switched protocol architecture throughout all the component elements of LocalNet. Such an approach offloads the time consuming and complex protocols from the user devices and provides a single standard implementation of protocols among heterogeneous computers and operating systems. A user device is responsible only for supplying the data required to be transmitted to the PCU. The PCU takes the data provided by the user device, formats the data into packets using the network-specific protocol, and routes the packet to the destination PCU. The destination PCU removes the data from the received packet and transmits it to the user devices connected to it. The PCU is attached to user devices via an RS-232C interface on one side and, on the other, the PCU communicates over the broadband cable via radio frequency (RF) modems that convert the digital data to a frequency of the PCU. Another product of

particular interest to the gateway design is the "bridge," a Sytek product which provides communication facilities for users on different frequency channels of the broadband network.

Bridge. One can identify two kinds of connectivity between users on a broadband multichannel system. LocalNet provides a one-hop connectivity between users attached to different channels on the same cable system via a store-and-forward bridge (Sytek, Inc., Bridge User Manual) which plays the role of a packet switch between different LocalNet 20 channels. The bridge allows communication between PCUs connected to different channels. A variant of the bridge, called a link, provides interconnection between distinct cable systems. Figure 3.2 shows an example of a LocalNet 20 system which includes two separate cable systems. Individual broadband channels of each cable system are joined by bridges and the two cable systems are joined by a link.

LocalNet 20 Protocol

The LocalNet 20 protocol is fully implemented within the PCUs. The LocalNet 20 protocol allows user devices attached to the PCU to communicate with one another via sessions. Through a port on the PCU, a user device can use the services provided by the PCU. User-initiated sessions provide the communication services to each port on the PCU. The protocol architecture enables each user device to

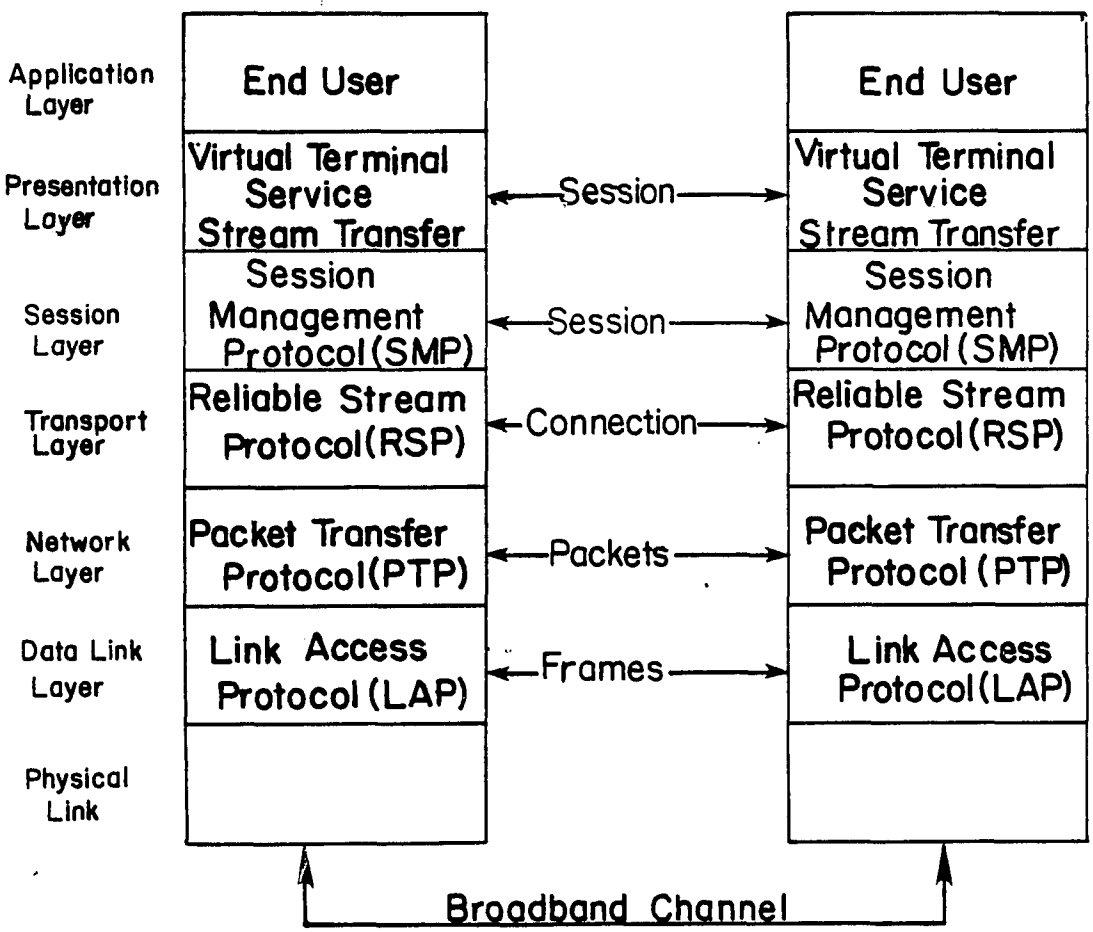


Example of a LocalNet-20 System

Figure 3.2. Example of a LocalNet 20 System.

control many ports and each port can have several sessions attached to it. The LocalNet 20 protocol architecture is layered in a manner which conforms to the ISO Reference Model. This model specifies a seven-layer architecture (Physical Data link, Network, Transport, Session, Presentation, and Application), and any system which conforms to this standard will be able to communicate with all other systems obeying the same standard.

The LocalNet 20 protocol architecture and its correspondence to the ISO Reference Model is shown in Figure 3.3. At the lowest level of the protocol architecture is the physical layer. This layer specifies the mechanical, electrical, and functional characteristics required to connect a LocalNet 20 PCU to a 128-kilobits-per-second broadband channel. At level 2 resides the link protocol. This layer is the one that decides the destination of the data and provides channel addressing and packet framing. In addition, the link layer is responsible for a user's device access to the channel bandwidth of the broadband coaxial cable. LocalNet 20 uses CSMA/CD media access mechanism. The control procedure used for information exchange at level 2 is the High Level Data Link Control (HDLC) protocol. The link layer also provides the functions of generating cyclic redundancy check (CRC) and error detection and enables correction through retransmission by the Transport layer protocol (to be discussed subsequently). The link layer is



LocalNet Protocol Architecture

Figure 3.3. LocalNet Protocol Architecture.

referred to as the Link Access Protocol (LAP) within the LocalNet architecture. Level 3 is the Network layer and is called the Packet Transport Protocol (PTP). The PTP provides packet routing services. The PTP also defines the rules for the delivery of a packet as a datagram. A point to note is that the Sytek network is a broadband multi-channel system with a bus topology. Evidently the issue involved in routing is one that is concerned in routing of packets between channels within the same cable system. The routing function is the primary responsibility of the bridge, a Sytek product designed to interconnect multiple channels on the same cable system. The Reliable Stream Protocol (RSP) is the next higher layer in the protocol architecture and corresponds to the Transport layer. The RSP provides reliable delivery of successive packets on behalf of session entities over a network connection. Each user packet is assigned a sequence number before transmission by the RSP within the PCU. The destination PCU will use the sequence number to order the packets and to acknowledge receipt of correctly received packets. Flow control of data from the sender to the receiver is by using the sliding window mechanism (Tanenbaum 1981) which regulates data flow over the session. The receiver can signal to the sending RSP entity the current amount of data it can accept. The Session layer is called the Stream Management Protocol (SMP), and provides Session management functions to the

LocalNet PCU. Session layer protocols are responsible for authorization of remote privileged users, data security, resource protection via flow control, and symbolic name translation. The Presentation layer protocol provides virtual terminal services, which allow many dissimilar devices like terminals to communicate with one another. Some of the Presentation interfaces supported by the LocalNet 20 protocol are the asynchronous ASCII RS-232C for standard ASCII terminals, and the Bisynch interface which allows IBM or other bisynch terminals to be connected to the PCU. The highest layer, the Application layer, is currently not implemented by the LocalNet PCU. The Application layer protocol is very user-specific and depends on the user application. This layer may include application-specific translations, document-format conversion, and other types of specific conversions. The gateway will use the Link Access Protocol (LAP) and the Packet Transport Protocol (PTP) for its operation. Therefore, these two layers will be discussed in more detail. As these two layers are concerned with addressing and routing functions, an introduction to addressing scheme within the LocalNet system would be useful at this stage.

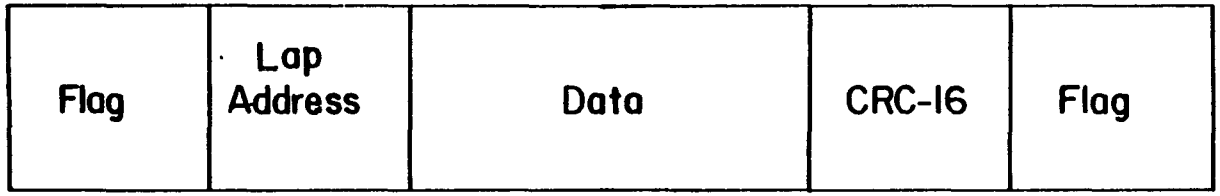
Addressing

Within a multi-channel LocalNet system, each node (i.e., PCU) is assigned a 16-bit address called the Unit Identification (Unit ID) which uniquely identifies it. In

packets, but does not uniquely identify nodes over the entire LocalNet cable system.

Link Access Protocol (LAP)

LAP provides packet transfer service between two nodes which are on the same frequency channel. Each node at the link layer is assigned an 8-bit LAP address which is used only to identify a node within a particular channel. The unit ID provided by the PTP is the one that uniquely identifies a node over the entire network. The LAP protocol serves the function of transferring data between nodes via the coaxial cable medium. The unit of information exchanged between nodes at the link level is called a frame. The LAP protocol uses the HDLC protocol. Error detection is accomplished through the use of cyclic redundancy check. Error correction through retransmission is the responsibility of the Transport layer protocol. The LAP protocol uses the CSMA/CD media access technique to obtain access to the cable transmission medium. The use of HDLC framing techniques and the CSMA/CD media access mechanism by the LAP provides for fair and equitable allocation of the channel bandwidth for all LocalNet users. A LAP frame is shown in Figure 3.4. Each frame is preceded by a flag character which has a bit pattern 01111110 (hex 7E). The flag byte provides byte synchronization at the receiver node end. Each frame is also delimited by another flag byte. The use of these delimiting flag characters allows a variable data field



Lap Frame Format

Figure 3.4. LAP Frame Format.

within a frame. Data transparency is provided by inserting a zero after a sequence of five ones and deleting such zeros upon reception. The LAP frame contains the link level address of a destination on its own channel which may not be the ultimate destination. If the destination is on the same channel, then the link level address used is the address of that destination. If the destination is on another channel, the link level address used is that of an intermediate bridge. The data field in the LAP frame contains the packet passed down from the previous higher layer protocol. Each layer adds its header to the basic data field supplied by a user process. As the packet moves down from a higher level, the next level treats the packet as data and encapsulates the packet within its headers. A series of encapsulations take place before the packet is eventually handed over by the PTP to the LAP. The LAP treats this encapsulated packet as the data and, in turn, embeds it in a LAP frame and transmits it over the network.

Packet Transfer Protocol (PTP)

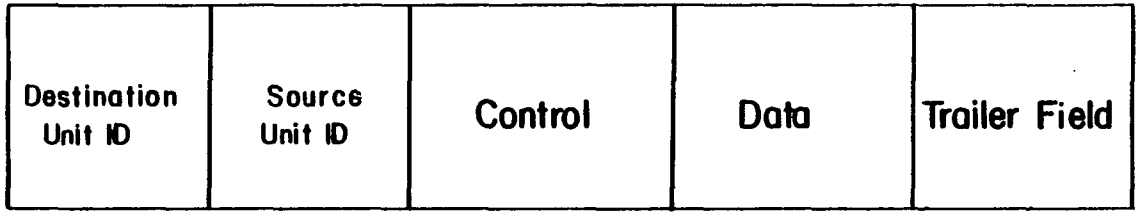
PTP is responsible for interchannel and intercable routing functions. Its scope is limited to providing the basic datagram service. There is no mechanism to promote reliability of packet delivery, flow control, sequencing, or other functions generally found in host-to-host protocols. These functions are assumed by higher level protocols like

the RSP. The PTP is implemented in all Sytek network elements involved in communications, namely the PCUs, bridges, and links. A packet will be forwarded from one PTP interpreter to another until the final destination is reached. When the source and destination are on the same channel, no intermediate PTP interpreter will be involved in the packet transfer. If the source and destination are on different channels, intermediate PTP interpreters within bridges will be required for forwarding the packets. If the source and destination are on the same channel, the packet is embedded within a LAP frame which includes the destination LAP address. The destination LAP interpreter, upon receipt of the frame, strips away the LAP header and forwards the packet to the destination PTP interpreter. The PTP will then process the packet and will strip the PTP header and pass it on to the next higher layer. Each of the higher layers in turn will process the packet and strip off its own header, and eventually the user data becomes available to the user process. If the source and destination are on different channels, then the source LAP interpreter embeds the packet within a LAP frame which includes the LAP address of a bridge on the current channel. The LAP interpreter at the bridge will pass the packet to the bridge's PTP interpreter. The bridge maintains a routing table, which contains a set of entries defining the adjacent hops in both directions. The PTP interpreter uses the routing

table information in determining the next channel and LAP address. The packet is then handed over to the appropriate channel's LAP interpreter. The LAP interpreter embeds the packet within a LAP frame which includes the LAP address of the next destination. If the next destination is the ultimate destination, then the LAP address is the address of the destination. If the ultimate destination is on another channel, the LAP address used is the address of another intermediate bridge along the route. The above description of packet routing assumes that a path between the source and the destination has already been established. However, if the source PTP does not know the location of the destination unit, then it initiates a "discovery" process (Sunshine 1983) which essentially sets up a path to the destination through the intermediate bridges. This route will be used by all subsequent packets between the source/destination pair. A PTP packet format is shown in Figure 3.5. The source and destination Unit IDs are 16-bit fields and indicate the Unit ID of the ultimate destination and the packet originator. The control field indicates which protocol level is to get the packet's data field. The optional trailer field is used for building route tables via broadcast discovery.

Reliable Stream Protocol

The Reliable Stream Protocol (RSP) corresponds to the transport layer of the ISO Reference Model and provides



PTP Packet

Figure 3.5. PTP Packet.

virtual circuit services to its users. The RSP resides on top of the PTP and assumes that the PTP provides the not-so-reliable datagram service. The RSP achieves reliability by guaranteeing that packets are delivered in proper order, without loss or duplication. RSP makes use of packet acknowledgements and retransmissions to provide the much desired reliability. RSP "flow controls" the packets that arrive from a sender to a receiver. The RSP uses a mechanism known as the sliding window, which is essentially a protocol indicating to the sender the number of packets it is permitted to send without the receiver's buffers overflowing.

RSP uses a simple method of retransmitting either lost or duplicated packets. The receiver checks each received packet to see if it is the next packet in the sequence expected. If it is, then the receiver passes the packet to the next higher layer. If it is not, then the receiver will send a special control packet called a negative acknowledgement packet to indicate to the sender to retransmit all unacknowledged packets in order, beginning with the last packet. In a network, there is also the possibility of an acknowledgement packet getting lost. To take care of such situations, RSP provides a scheme called a "retransmit timer." A timer expiration is typically caused by loss of an acknowledgement packet. At this time, the RSP will retransmit the last unacknowledged packet. The RSP

packet format, and the description of various fields within it, is available in Ennis (1983).

Discovery Mechanism

The discovery process can be used to establish routes in a multi-network system without prior knowledge of either the destination node's location or the topology of the network. It assumes that no node on the multi-network system knows the location of the other node on the network, i.e., it does not know the destination node's channel number or its LAP address. A user communicates with another user over the network via either a terminal or a computer connected to the network via the PCUs. For generality, all devices used by users for network communication, namely terminals and computers, will henceforth be referred to as user devices, and the device/PCU pair will be called a user node.

When a user node wants to communicate with another user node but does not know the destination's location, the PTP interpreter requests its LAP interpreter to broadcast the packet on its channel. Such a packet is called a broadcast discovery packet. All nodes on that channel check to see if the destination unit ID in the Broadcast packet matches their own unit ID. If one of them is the ultimate destination, then that PCU will respond with a packet containing its LAP address. All future packets directed from

this source to the destination will carry the specific LAP address which was returned from the ultimate destination in its response packet. No further broadcasts will be necessary. If the destination is not on the same channel, the bridge's LAP interpreter will receive the discovery packet and record the discovery attempt. The bridge proceeds to broadcast the discovery packet on all the channels to which it is connected. The Broadcast packet will be picked up either directly by the ultimate destination on one of the broadcasted channels or by other intermediate bridges. The broadcast process is repeated at all intermediate bridges. Eventually the discovery packet reaches the destination.

The destination PCU sends back a Response packet containing its LAP address. The Response packet returns to the source PCU along the path followed by the first discovery packet. The Response packet establishes a particular route on its return journey. Bridges along the route update their routing table entries, indicating new knowledge of the destination's location. All future packets will follow this route for the entire duration of the session between the source and the destination. The details of the actual routing table buildup at the bridge is described in (Sunshine 1983). Thus the bridge functions as a packet switch between channels. It maintains a route

table from which it determines the appropriate channel and LAP address to use in forwarding the packet to its next hop.

ARPANET

The objective of this section is to provide some background information about the ARPANET. The ARPANET is one of the earliest and best known packet-switched computer networks. It has been studied and documented extensively. The Advanced Research Projects Agency (ARPA) of the Department of Defense (DoD) sponsored in late 1968 the development of a new type of computer network, the purpose of which was to interconnect via common carrier lines, dissimilar computers located at geographically distributed ARPA-sponsored research sites. The primary purpose was to allow resource sharing among research sites and to allow research in packet-switching and protocol technology. The ARPANET has grown from an experimental four-node regional network in late 1969 to a fully operation eighty-node network interconnecting over 200 computer systems in the United States, Hawaii, England, and Norway. The computers connected to the ARPANET are referred to a host in the ARPANET environment.

The ARPANET can be divided conceptually and physically into two parts (Figure 3.6), a subnetwork which consists of small dedicated communications processors called Interface Message Processors (IMPs), interconnected by leased digital communication lines, and the hosts which are user-owned, application-oriented computers attached to the

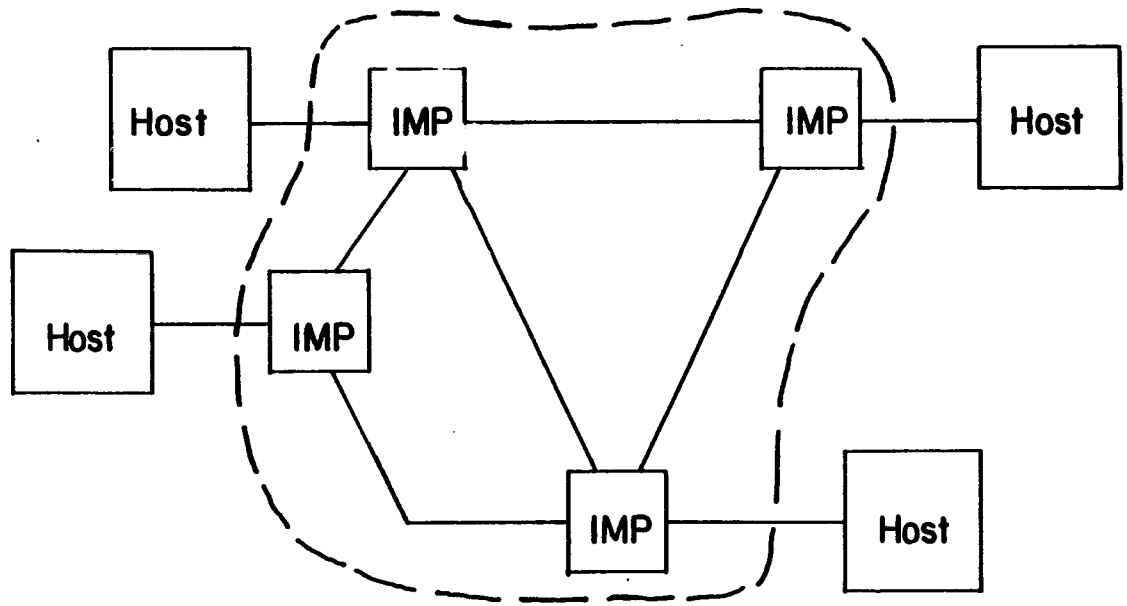


Figure 3.6. ARPANET Hosts and IMPs.

subnetwork communication processors. Two types of communication processors exist in the ARPANET subnet. The IMPs provide subnetwork access to the host computers, and the Terminal Interface Processors (TIPs) provide access to user terminals to the subnet. The leased lines interconnecting the ARPANET IMPs and TIPs can be operated from 9.6 kilobits per second to 230.4 kilobits per second. Typical bandwidth is 50 kilobits per second. Each host and terminal is connected to the subnet via single IMP or the TIP. Each IMP can accommodate several hosts. Each host is connected to its associated IMP via one of the three types of interfaces, depending on the physical proximity between the IMP and the hosts. A local host connection limits the distance between the IMP and the host to 30 feet, and a distant host connection allows a host to be located up to 2000 feet from the IMP. A very distant host connection will allow a host to be located at distances greater than 2000 feet from the IMP.

The host computers connected to the ARPANET may vary widely in hardware and software configuration. They could differ in terms of speed, word length, operating system, etc. To overcome this inevitable heterogeneity which makes communication impossible between hosts, all hosts must implement a common protocol. The two DoD-supported ARPANET protocols are the Transmission Control Protocol (TCP) and the Internet Protocol (IP). TCP establishes and breaks

connections and enables transfer of messages on behalf of end-user programs or processes executing in these computers. The Department of Defense mandates that all ARPANET hosts implement the TCP and IP as standard network protocols. The IMPs essentially act as front-end communication processors, offloading the host from the complex network functions. The IMP performs the functions of segmentation of host messages into packets, transmitting and routing of store and forward packets, generating headers, retransmitting unacknowledged packets, reassembling received packets into messages for transmission to the hosts, error checking, sequencing, flow control, host status monitoring, and other functions. It is evident that the above functions are complex and time consuming, and therefore require a special computer like the IMP to perform these functions rather than building the complex functions into the host computer itself. The IMP also serves as a multiplexer, connecting several hosts to the network. Finally, the IMP isolates the ARPANET from the idiosyncracies of the different types of hosts connected to it.

Internet Protocol

The Internet Protocol (IP) was developed for the United States Defense Advanced Research Project Agency (DARPA) and has evolved over the years into a DoD standard internet protocol. The Department of Defense requires that

all computers (called hosts on the ARPANET) and the gateways connected to the ARPANET must implement the IP protocol. The IP provides datagram services over an interconnected system of networks. IP does not assure reliability. The functions of flow control and retransmission due to errors are the tasks of higher level protocols like the Transmission Control Protocol (TCP).

The motivations of IP (Postal 1981) are to allow computers on different networks to communicate with each other without the need to change the internal operations of any member of the interconnected system of networks. Thus all computers use a common protocol throughout the internet. As mentioned previously, IP was intended to be implemented in every computer on different networks which wished to exchange information with each other. IP does not substitute for any of the existing protocols within any network, but is used along with them. By having a common internet protocol throughout, IP extends the communication range of the computers over an interconnected system of networks. Figure 3.7 shows how the IP fits into the protocol hierarchy of each individual interconnected network. The IP interfaces on one side to a higher level protocol like TCP or any other host-to-host protocol, and on the other side it interfaces to a local network protocol (the term local network denotes any individual network protocol used in the different networks and does not necessarily mean any local

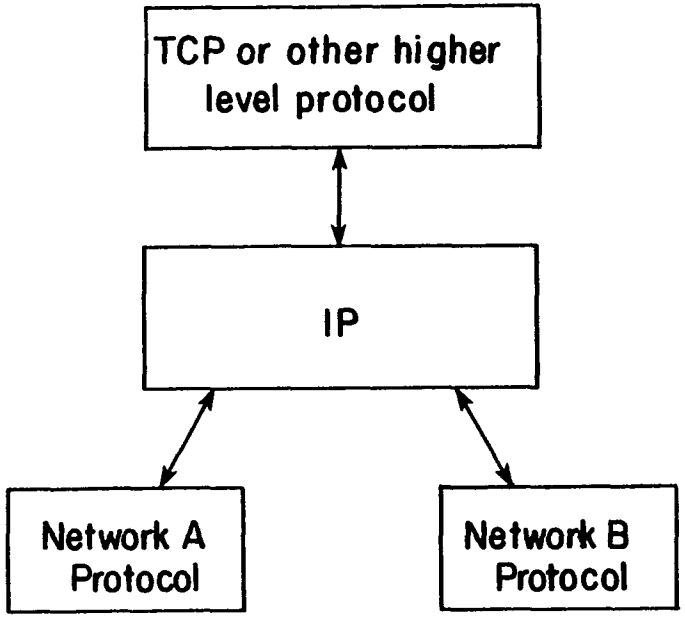


Figure 3.7. Protocol Layering.

area network protocol). IP is designed to be general enough to be integrated into a variety of different network protocols.

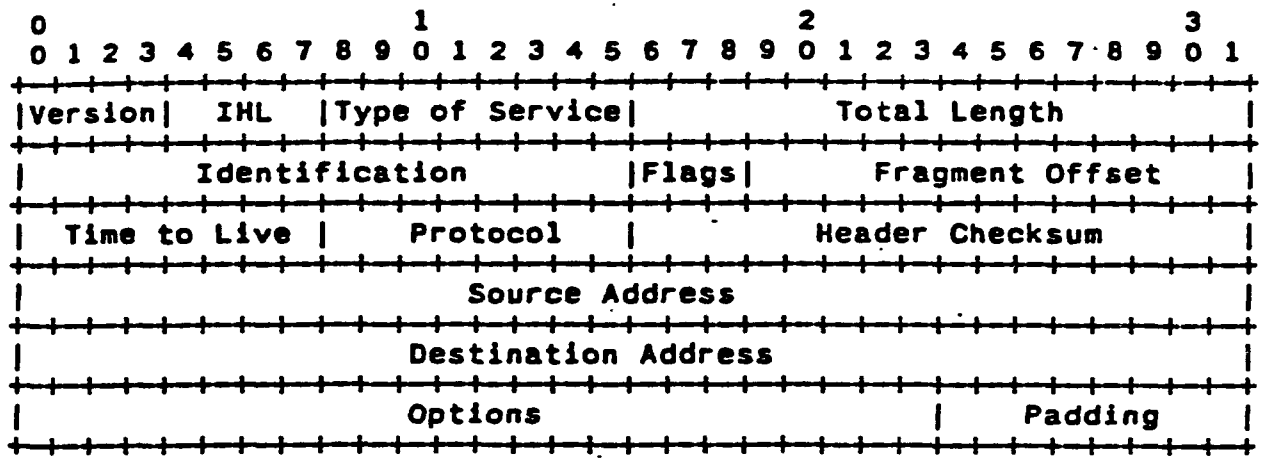
IP provides two basic types of functions: routing a datagram across an interconnected system of networks, and fragmentation and reassembly of packets of long datagrams. IP allows datagrams to be fragmented for transmission through networks imposing small size packet limits. The first step in sending a datagram across the internet is for the application program to prepare the data and call on its IP to send the data as a datagram. The user program supplies the necessary destination and other required parameters to the IP. The IP prepares a datagram header and attaches the data to it. The IP determines a local network address for sending the datagram. This address may be the address of a destination if it is on the same network as itself, or it could be the address of an intermediate gateway. Once IP has determined the destination address, it forwards the packet to the local network interface, which could be either the 1822 or the HDH (BBN Report No. 1822), in case of the ARPANET. The local network interface will then create a local network header and will encapsulate the datagram within the local network header and send the resulting packet on to its local network. The datagram arrives at the gateway encapsulated in a local network packet. The gateway will strip off the local network header

and will pass the IP datagram to the IP module. The IP module determines from the internet address that the datagram should be forwarded to another host on a second network. The IP determines the local network address of the destination host. It then requests the local network interface of the other network to send the datagram to the destination. Recall that the gateway implements whatever network interface is required by the two networks to which it is attached. At the destination, the datagram is stripped off the local network header by the local network interface and handed to the IP module. The IP module determines which program or process should receive the data and passes it accordingly. It is easy to see from the above description that the IP was developed with the idea that it could be implemented in all hosts and gateways involved in internetwork communication.

This approach of having IP everywhere is highly impractical, particularly in an application like the gateway development project at the University of Arizona, which requires two Sytek networks to be connected via the ARPANET. The LocalNet 20 has all its network protocols built into its PCUs and the hosts and terminals merely connect to the PCUs. The hosts/terminals on the Sytek network are not required to implement any protocol at all, and only supply the data to the PCUs. Evidently if the IP has to reside in each host, then the IP must be incorporated into the hierarchy of

LocalNet 20 protocol architecture. All Sytek PCUs engaged in internetwork communication will have to be modified to accommodate IP, and therefore this is not a practical approach. The best solution at present in providing an interconnection to the ARPANET is to allow each network to have its own set of protocols which have been optimized to handle communication characteristics of traffic for which it was intended and to let the gateway take care of the protocol and other incompatibilities. This way, a host or the network interface unit like the PCU will not have to worry about implementing many different protocols for each new communication requirement which comes along. For this reason, the gateway implements both the LocalNet 20 protocol and the IP. Sytek hosts communicate with the gateway using its LocalNet 20 protocol. IP is implemented only at the gateway, which conforms to the DoD requirement that all ARPANET hosts/gateways use the IP as a standard protocol for internetwork communication.

In addition to providing the datagram service described above, IP provides functions like fragmentation and reassembly of larger packets for transmission through networks with smaller packet size limits. An IP header is shown in Figure 3.8. Details describing the different fields are available in (DARPA - DoD Standard Internet Protocol). Only the key fields of the header are described below. The source and destination fields uniquely identify



IP HEADER FORMAT.

Figure 3.8. IP Header.

a source/destination pair. Each field is 32 bits. The first byte of the source address is the network address to which the source is connected. Similarly, the first byte of the destination address is the network address of the destination. The remaining 24 bits of the source/destination address field identify the actual node address. IP provides error checking via checksum but the checksum is only on the header. The checksum is computed and verified at each point the internet IP header is processed, and is referred to as a hop-by-hop checksum. The total length field gives the length of the IP header and data. The time to live field indicates the maximum time a datagram is allowed to exist in an internet. The time to live field is essentially a counter and is decremented as the IP datagram moves along the internet. The datagram is destroyed when the count reaches zero. If for some reason the datagram gets lost due to a routing malfunction and cannot be delivered within a reasonable time period, it must be discarded. Otherwise the datagram would be in an infinite loop in the network, occupying an unnecessary transmission bandwidth. The maximum time to live on the ARPANET is 255 seconds.

Transmission Control Protocol

The Transmission Control Protocol (TCP) is a host-to-host protocol used in packet-switching computer networks. TCP is a connection-oriented, end-to-end reliable protocol which enables interprocess communication between hosts in an

interconnected multi-network system. TCP is used as a standard high level protocol for DoD-wide interprocess computer communication.

The TCP interfaces with application processes on one side and on the other, it interfaces with lower level protocols like the IP. TCP expects only an unreliable datagram service from the lower level protocol.

TCP was designed to fit into a variety of environments. Usually TCP is assumed to be a module in an operating system. Users access TCP just like they would access any other files within the system. The interface is similar to the calls an operating system provides to application processes for file manipulation. Examples of such calls could be open and close connections to send and receive data on established connections. Application processes intending to transmit data to the network call on the TCP and pass buffers of data as arguments. TCP packages the data into segments and calls on the IP module to transmit each segment to the destination TCP. The destination TCP receives the segment, places it in a receive buffer, and notifies the receiving user.

The IP provides the interface between the TCP and the network. IP wraps each TCP segment inside an internet datagram and routes it to the destination IP module or gateway via the local network. If a gateway is involved between the source and the destination, then the gateway will unwrap the

internet datagram from its local packet to determine the next network on which the datagram should be transmitted. The datagram is then encapsulated in a local packet of the next network, and is once again routed to the next gateway or its final destination. At the destination, the IP unwraps the segment from the datagram and passes it to the destination TCP.

As mentioned before, TCP was designed to provide reliable connection-oriented services between pairs of processes. To provide this service, TCP performs several basic operations.

Reliability. TCP provides error-free point-to-point communication, delivering messages in the order they were sent. In order to recover data that might be damaged, lost, duplicated, or delivered out of sequence, TCP takes two approaches:

(a) A sequence numbering scheme. Here each segment that is transmitted is assigned a sequence number. This enables the receiver to determine the correct order of the segments received that could have arrived out of order, and to eliminate duplicates. Checksum is used to detect damaged packets.

(b) An acknowledgement scheme. Here, for each transmitted packet the source expects a positive

acknowledgement from the destination. If no acknowledgement is received within a certain time, the source TCP will time out and retransmit the data.

Thus the TCP provides an end-to-end service via the mechanisms described above, assuring the user of very reliable communication.

Flow control. TCP provides flow control services to its users. It makes use of a "window" mechanism whereby the receiver indicates in each acknowledgement it sends back the range of acceptable sequence numbers it can receive after the last segment it has successfully received. The sender has to wait for permission before it can send any new segments. This mechanism ensures that the source does not flood a receiver with more data than the available buffer size can accommodate.

Multiplexing. TCP allows many processes within a host to make use of TCP communication facilities simultaneously. TCP provides a set of ports within each host. The port address along with the network and host addresses form a socket. A pair of sockets uniquely defines a connection. TCP allows each socket within a host to be simultaneously used in multiple connections.

Usually some of the very frequently used processes are bound to certain ports and the sockets formed from such

ports are made public. Then the services provided by these processes can be used by every user in the internet.

Connection. In order to support reliability and flow control mechanisms, TCP must initialize and maintain certain status information for each data stream. The status information along with the sockets, sequence numbers, and window sizes is called a connection. A connection is a means of identifying the two ends and is uniquely defined by a pair of sockets. When two processes want to communicate with each other, the TCPs at either end must establish a connection with each other, exchange data over the connection, and terminate the connection when the communication is over.

We see that TCP provides several functions which are vital for reliable interprocess communication in a multinetwork system. Implementing TCP at a gateway connecting a LAN to a long haul network will play an important part in isolating the time-out problems caused on LANs because of additional delays of long-haul communication. Since acknowledgement is provided by the gateway, the LAN time-out parameters could remain unaltered. In addition, a TCP gateway will help control to flow between the gateway and LAN sources. At present there is no flow control mechanism at the gateway, and when there is a large internet traffic, the gateway is bound to become a bottleneck.

CHAPTER 4

GATEWAY DESIGN

In this chapter a specific design and interconnection scheme to connect a Sytek local area network to the ARPANET, and the operation of the gateway, are discussed. The approach has been to integrate the gateway using multi-vendor network products and optimizing the design to best suit the application. Such an approach is probably the fastest way to meet the urgent needs for interconnecting several heterogeneous networks that are in existence today. The description below is intended to give an overview of the gateway design and does not give the detailed specification.

Interconnection Scenario

Several factors affect the design of the gateway. One is the network topology of the interconnected system of networks. Sometimes the networks to be interconnected already exist and their specifications are known. For example, when any two local networks, remotely placed, need to communicate with each other via a long haul network. Another design issue is to determine whether any of the intermediate networks will be involved in the actual communication, or whether they will be used merely as a transport medium. Yet another design consideration is

whether the communication is between two homogeneous networks (same manufacturer) or between two heterogeneous networks (different manufacturers).

The purpose of the gateway project at the University of Arizona is to enable the interconnection of two remotely placed Sytek local area networks (homogeneous networks) via the ARPANET. The gateway will provide the interface between the local area network and the ARPANET. Figure 4.1 shows the interconnection scenario. There will be two identical gateways involved in this interconnection scheme. Evidently, such an interconnection will extend resource sharing beyond the distance limits of local area networks. It is possible to interconnect two remotely placed local area networks from different manufacturers (heterogeneous). This would require two different types of gateways. One gateway will provide an interface between the ARPANET and the first type of network and the other will provide an interface between the ARPANET and the second type of network. Discussion in this thesis will be restricted to a specific interconnection between two similar but remotely placed Sytek networks. It may also be noted that the communication is only between the remote Sytek networks. There will be no communication between an ARPANET host and a Sytek host. The ARPANET subnet will merely serve as a vehicle in transporting the Sytek packets between the remote locations. The gateway design uses the Sytek's bridge product as the basic

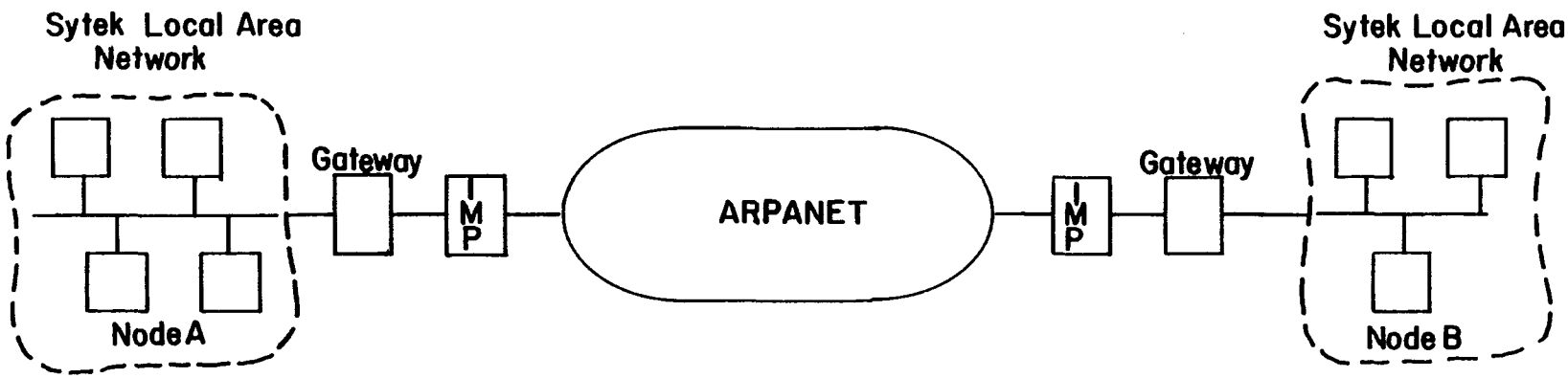


Figure 4.1. Interconnection Scenario.

hardware component into which the additional custom hardware and software is being integrated. Thus a modified Sytek bridge will serve as the internet gateway. The reasons for using the bridge product for the gateway design will become evident when the function of it and the design of the gateway are discussed in the subsequent sections.

The bridge product which has been described in an earlier chapter essentially serves as a packet switch between different broadband channels. It performs two important functions: (1) enables a PCU to "discover" a communication path to another PCU located on a different channel, and (2) provides the actual packet transportation between PCUs on different channels, i.e., all packets pass through the bridge. The discovery mechanism discussed before was intended to establish routes between nodes in a multi-network system without requiring prior knowledge of either the destination node location or the network topology. Initially no node knows the location of the other node, i.e., it does not know either the destination node's channel number or the LAP address.

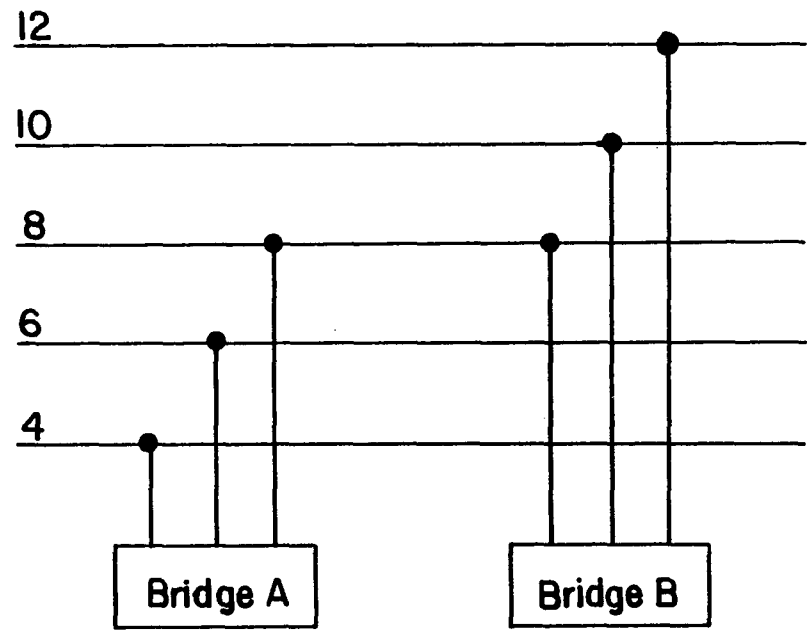
The key design feature in the gateway development has been in extending the idea of the discovery mechanism in order to provide an interconnection to the ARPANET. The most important concept here is to treat the ARPANET interface as just another broadband channel of the Sytek network. Thus, as far as the bridge is concerned, the ARPANET appears

as one of the broadband channels to which it is connected. With this notion, it is easy to see that a modified bridge with all the hardware and software required for the ARPANET interface can now serve as a packet-switch between a Sytek channel and the ARPANET. Thus by taking advantage of an already existing bridge product and the discovery mechanism, a simple and efficient interconnection can be provided between multi-network systems. The concept becomes clear when the gateway operation is explained in a later section. In this section we will describe how a bridge is being customized to function as a gateway between the Sytek network and the ARPANET.

Gateway Hardware

It is meaningful to describe the basic bridge hardware before describing the custom hardware that will be incorporated into it.

The LocalNet 50/201 bridge product supports from two to eight LocalNet channels, depending upon the network configuration and the size of the bridge used. A typical bridge configuration is shown in Figure 4.2. Bridge A is connected to channels 4, 6, and 8, and bridge B is connected to channels 8, 10, and 12. Channel 8 is common to both the bridges. Users on channels 4 and 6 can now communicate with users on channels 10 and 12. Packets sent by users on channels 4 and 6 to bridge A will be retransmitted to bridge B on channel 8. Bridge B, in turn, will send out



Bridge Configuration

Figure 4.2. Bridge Configuration.

packets on channels 10 and 12. Thus a multi-hop connectivity is being provided by the bridges.

The basic bridge product consists of three main components: (1) a controller board which performs packet routing between broadband channels; (2) a channel interface unit (CIU) which interfaces with and supports two LocalNet 20 channels; and (3) a LocalNet 20 modem board that changes RF signals into digital signals and vice versa, thus providing an interface to the broadband coaxial cable. The number of channels that the bridge can interconnect may be expanded by adding additional CIU boards and modem boards to the bridge. As an example, a four-channel bridge will consist of a controller board, two CIU boards (one for each two channels), and four modem boards (one per channel). An RS-232C console port is available to monitor the controller status via a terminal or a computer system. In addition to the boards mentioned above, the bridge consists of the following hardware: (a) a multibus bridge chassis; (b) a card cage; (c) power supplies; (d) RS-232C assembly; (e) fans; and (f) multibus connectors. Some of the important hardware components relevant to the gateway design discussion have been described below.

Bridge Chassis

The bridge chassis contains the card cage, the power supplies, and a set of boards as its primary components. In

addition, the chassis provides a panel board to mount the RS-232C connector, the AC power inlet to the bridge, and an RF assembly consisting of a splitter, a coaxial cable, and a connector.

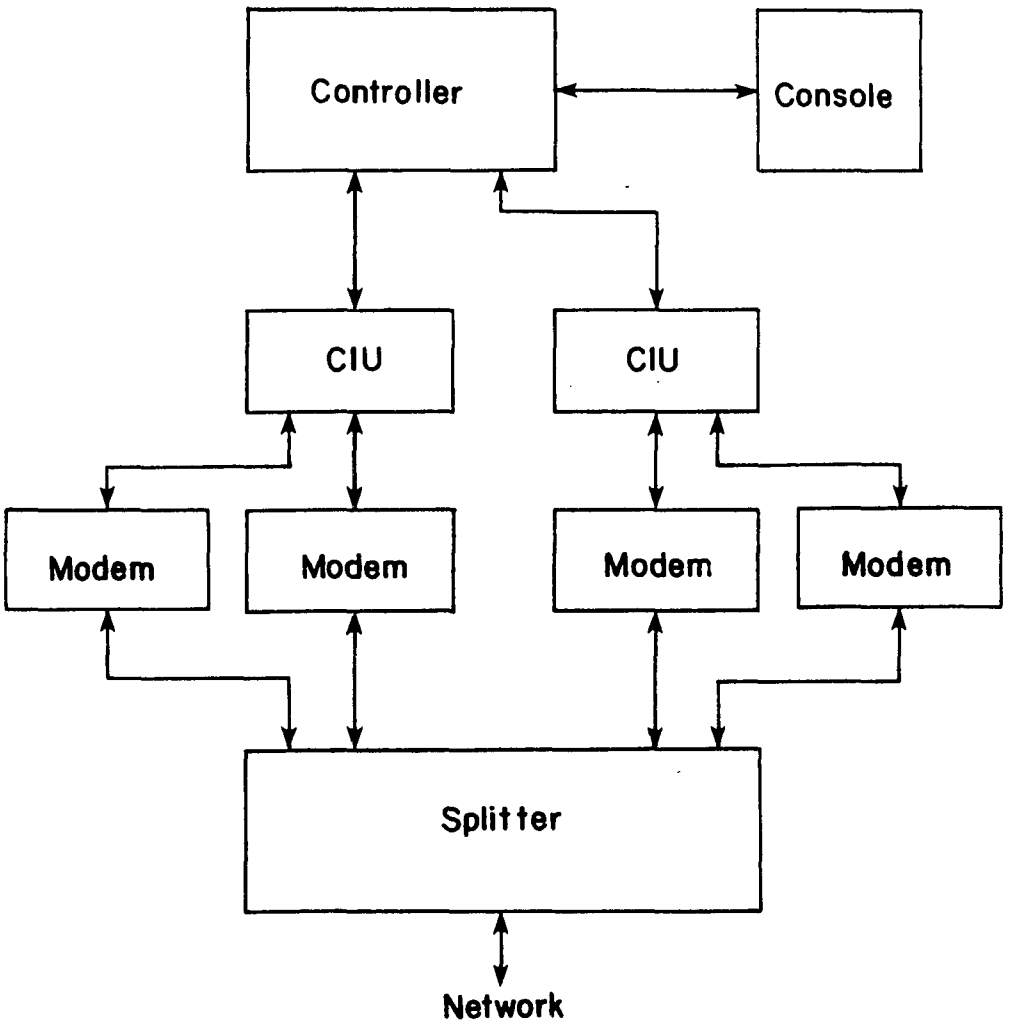
RF Assembly

The RF assembly consists of an eight-way splitter that can be connected to a maximum of eight modem boards. A coaxial cable connects the splitter common port to a connector on the back panel.

Card Cage

The card cage is a 21-slot metallic structure mounted to the chassis. The card cage has a multibus back panel and houses the controller board and the CIU boards. Figure 4.3 shows the block diagram of the bridge, illustrating the interconnection of different components mentioned above.

The RF signal from the network enters the bridge through the RF cable on the back panel, from which it goes to the splitter to be distributed to the modem boards. The modem boards convert the RF signal to digital data and transmit it to the CIU boards. The CIU board acts as an interface between the controller board and the modem boards. The CIU board receives packets from the modem boards, checks them for transmission errors and signals the controller board of the arrival of a packet. The controller board



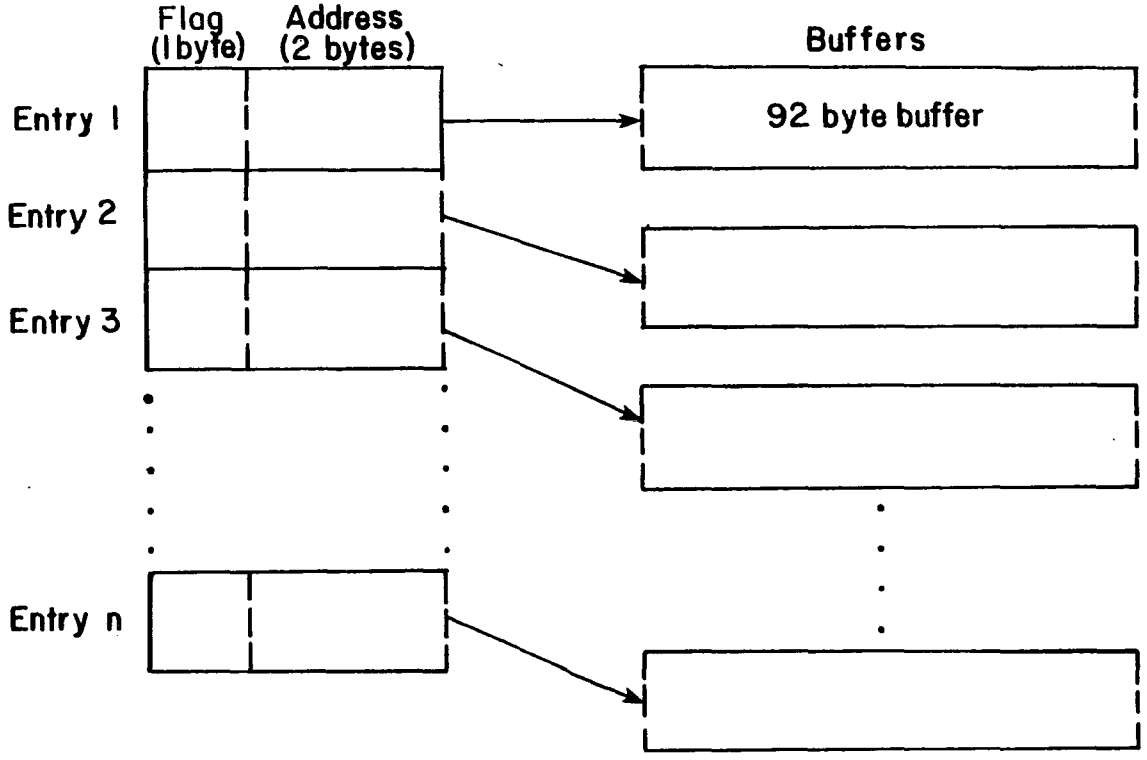
Block Diagram of a 4 Channel Bridge

Figure 4.3. Block Diagram of a 4-Channel Bridge.

processes the packet and sends it back to the CIU board from which it is retransmitted to the appropriate channel.

CIU Board

The CIU board is a microprocessor-based board consisting of a Z-80 CPU, a serial interface (SIO) for controlling the modems, and a parallel interface (PIO). The CIU board has 16 kilobytes of memory, a 16-bit address bus, and an 8-bit data bus. The CIU board has the necessary hardware to check for CRC errors. The CIU board communicates with the controller board via the multibus on one side and with a pair of modems on the other. The interface between the CIU and the controller is via a set of queues maintained in the multibus memory address space of the CIU boards. There are in fact two transmit and one receive queues and the associated queue pointers for every CIU/controller interface. The receive queue is for packets received from the Sytek channels and is common to both the channels to which the CIU board is connected.. The transmit queue is for packets that are to be transmitted by the CIU out onto the broadband channel. There is one transmit queue for each channel. The transmit and receive queues are identical and are shown in Figure 4.4. Each entry in the queue is three bytes, one a flag byte and two address bytes. The flag byte indicates if there is any valid data in the associated buffer. Each buffer can hold one packet. There are two pointers for each queue, one for the controller and



Transmit and Receive Queues

Figure 4.4. Transmit and Receive Queues.

the other for the CIU. The CIU and the controller can each modify only its own respective pointer. After a packet is received, the CIU puts the packet into the receive queue and sets the flag byte associated with that buffer. This signals the controller that a packet has arrived from the network and is waiting in the receive queue.

Controller Board

The controller board is based on a 16-bit microprocessor and has 128 kilobytes of memory. Upon power on or system reset, the controller checks specific memory locations (Schultz 1983) of the CIU's multibus memory space to retrieve queue pointers and other information. It should be noted at this stage that each CIU board (up to a maximum of four) used on the bridge must be assigned a specific multibus address. The CIU will initialize particular memory locations within its multibus memory space upon power on or reset, and the controller checks the multibus memory of each CIU board to determine the presence of a CIU board. If the controller does not find the required information at the memory address assigned to the CIU boards, then it assumes that there is no CIU board assigned to that particular memory slot. Once the controller has determined the state of the CIU multibus addresses, it goes into a polling mode where it continually polls each CIU receive queue for reception of a packet. If the controller detects that a CIU has

set a flag byte on its receive queue, indicating that a packet has arrived, the controller retrieves the queue pointer, pointing to the buffer that was filled. It then proceeds to process the packet and determines from its routing table which channel must be used for retransmission. After determining the packet destination, the controller changes the LAP address from the bridge LAP address to that of a destination PCU or another intermediate bridge. (Recall that if the source and destination are not on the same channel, then the source PCU would have sent the packet embedded in a LAP frame containing the LAP address of an appropriate bridge.) After changing the LAP address, the controller will move the packet from the receive queue on the receiving CIU board onto the transmit queue on the transmitting CIU board. After moving, the controller will set the flag byte of the queue entry, signalling the CIU that it has a packet ready for transmission onto the cable. The CIU will retrieve the packet from the transmit queue and will serialize the data and transmit it to the modem board, where it will be modulated onto the destination channel. Two points to be observed from the above bridge operation are: (a) the controller board communicates with the CIU boards via a set of queues maintained in the multibus address space of each CIU. Each CIU is assigned a specific multibus memory address slot; and (b) on power on/reset, the controller checks specific memory locations within the

multibus address space assigned to each CIU board. If it finds the desired information, it knows that the particular CIU board has been installed in the bridge. Remember that the basic bridge configuration has one CIU, one controller, and two modem boards, and can support only two channels (each CIU connects to two channels). The bridge's capacity to support additional channels can be expanded by adding CIU boards and modem boards.

Evidently it is now easy to see that if, into one of the slots meant for the CIU board, an ARPANET interface board with a multibus memory address configured to be identical to that of one of the CIU boards is inserted, the controller will not notice the change. As long as the new ARPANET interface board mimics the CIU board in maintaining queues and passing other information such as pointers and control information, the controller will be unable to distinguish between the ARPANET interface and the CIU board. The entire ARPANET interface wil now appear as just another Sytek broadband channel. This is precisely what is being done. The details of this modification to the bridge are discussed in detail below. The modified bridge will henceforth be referred to as the gateway.

Gateway Architecture

The gateway hardware architecture, including the ARPANET interface, is shown in Figure 4.5. The basic components of the gateway are the controller board, the CIU

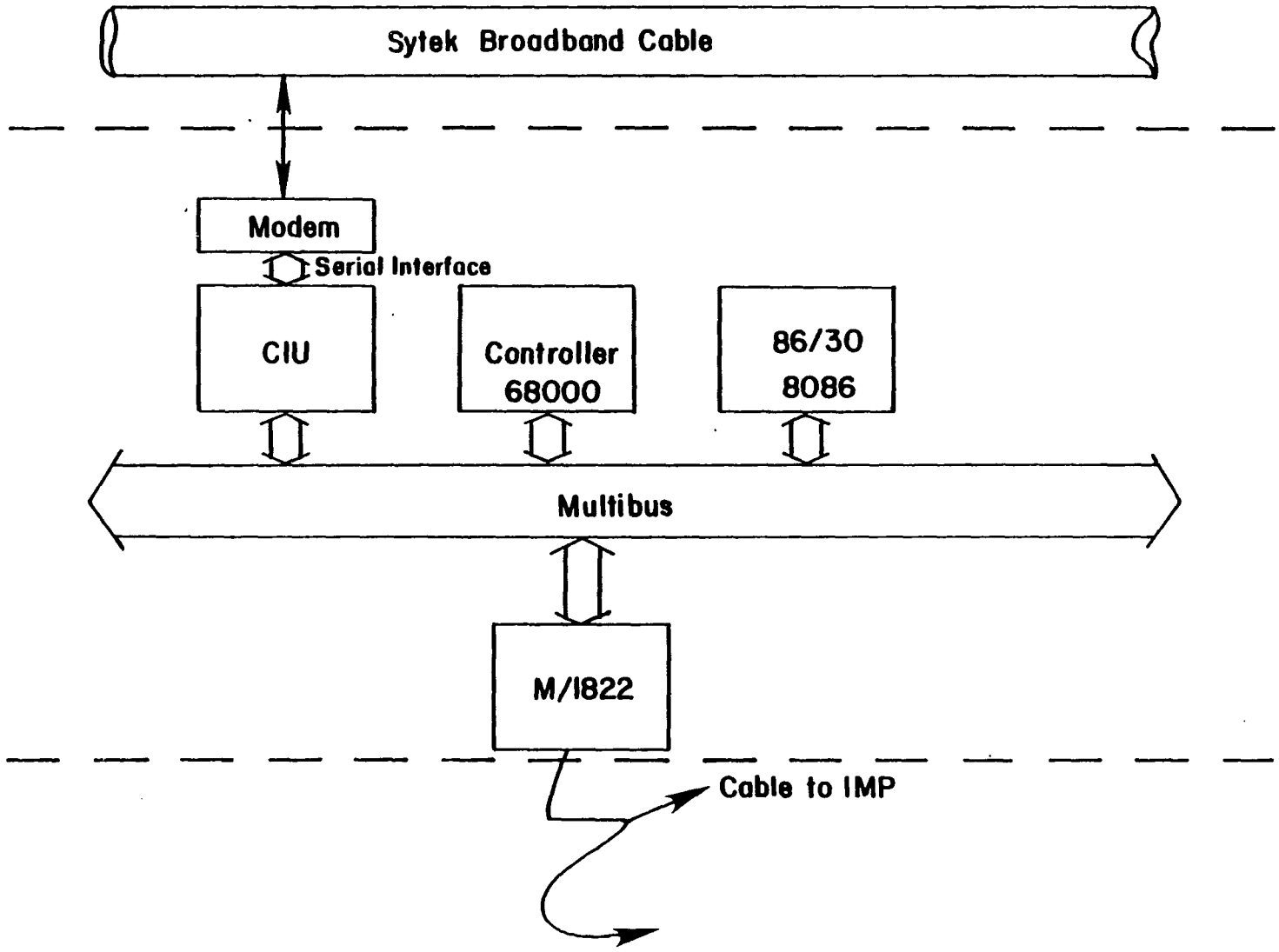


Figure 4.5. Gateway Architecture.

board, the modem boards which form the Sytek half of the gateway, and an 1822 board which forms the gateway's interface to the ARPANET IMP. In addition, an INTEL 86/30 single board computer (Intel, iSBC 86/30 Hardware Reference Manual), which mimics the CIU board and does the interface function between the Sytek protocol and the ARPANET protocol, forms the heart of the gateway. The 86/30 board and the M/1822 board are both multibus-based and can easily be integrated into the bridge. Architecturally, the gateway may be visualized as consisting of an 86/30 processor board with dual-port memory, with interfaces to the two networks on either side of it. Packets traveling between the Sytek network and the ARPANET are buffered in the 86/30 memory where the protocol transformation is applied by the 86/30 processor. With this background of the gateway architecture, we will see how the gateway is connected to the ARPANET. The ARPANET half of the gateway is the one that has all the hardware and software modifications done to the bridge.

Gateway to IMP Interface

The gateway is connected as a host computer to the ARPANET and is indistinguishable from any other ARPANET host. All hosts and gateways are connected to the ARPANET via the IMPs. Recall that the IMPs form a subnet and that this subnet operates completely autonomously. Since the

subnet functions as a store and forward system, an IMP should not depend on the host for its operation. The IMP must always be able to function, irrespective of whether the host is operating or not. In effect, the host must be isolated from the network and must not be able to change the logical characteristics of the subnet. (Henceforth the terms host and gateway will be used interchangeably.)

Each IMP may be connected to a variety of hosts. Some parts of the host-IMP interface are customized to each host. Therefore, the host (gateway)/IMP interface has to be partitioned into two separate units as shown in Figure 4.6. The standard host/ IMP portion is built into the IMP, which contains the logic standard to all the host/IMP interfaces. The special portion of the interface is unique to each type of host. At the gateway, this specific piece of hardware is the M/1822 board (ACC,M/1822 User's Manual), which provides communication with the IMP according to protocol defined by the BBN Report No. 1822. The IMP supports three types of host connections, local, distant, and very distant. The local host connection limits the distance between the IMP and the host to a maximum of 30 feet. A distant host connection allows the distance between the host and the IMP to be up to 2000 feet. Beyond 2000 feet, a very distant host connection will be required. The host and the IMP communicate over a host cable, which connects the standard host/IMP interface built into the IMP with the special

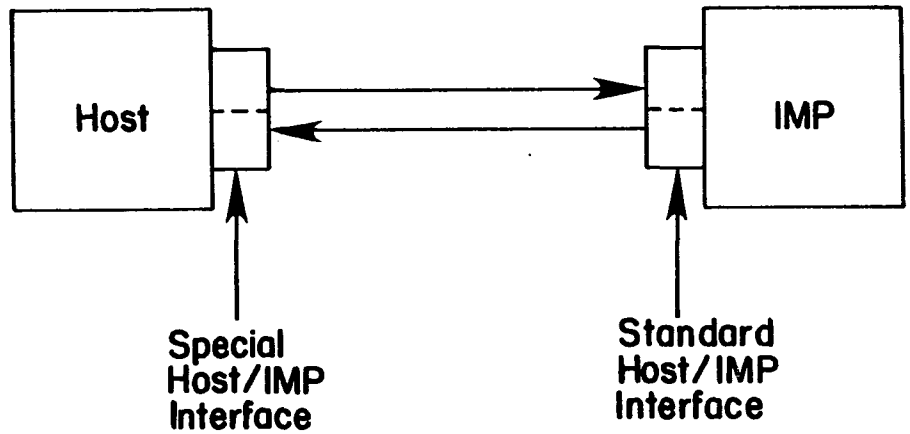


Figure 4.6. Host/IMP Interface.

host/IMP 1822 interface that has been added to the host. The 1822 protocol is an asynchronous, full duplex, bit-serial interface that can be divided logically into a host-to-IMP section and an IMP-to-host section. Each section contains a 16-bit shift register, one of which is for shifting bits to the host and the other for receiving bits from the host. Figure 4.7 shows a simplified host/IMP interface. Each bit is passed across the interface via a ready-for-next-bit/there-is-your-bit handshake procedure. This technique permits the bit rate to adjust to the rate of the slower member of the host/IMP pair and allows for necessary interruptions when words must be stored after being assembled from the serial data or retrieved, when a word has to be transmitted, from memory.

The 1822 host/IMP interface has taken into consideration some of the possible effects of a system failure in the subnet. Minor failures such as temporary line failures will only degrade the rate of service. Serious failures could, however, result in loss of message or loss of subnet communication. IMPs inform hosts of system failures via an IMP master-ready line, which indicates to the host whether the IMP is ready to communicate. Similarly a host computer, should it go down, sends out a signal on its host-master ready line indicating that it is no longer ready to communicate with the IMP. This message is further propagated throughout the subnet to all the IMPs.

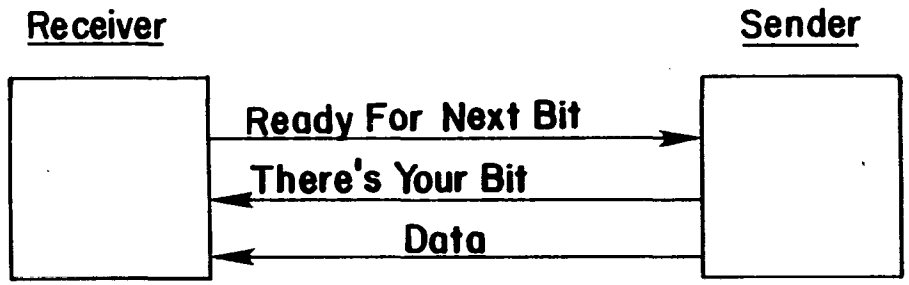


Figure 4.7. Host-IMP Handshake.

The IMPs, in turn, notify their respective hosts of the unavailability of the faulty host, should they try to send a message to it.

Gateway to Sytek Connection

This connection is simple. The bridge already comes with this interface. An RF coaxial drop cable is connected between the Sytek main truck network cable and the gateway back panel. The back panel in turn is connected to the splitter common port. The splitter distributes the RF signal to the modem boards.

Gateway Software

The interconnection scheme plays an important role in determining the software architecture at the gateway (Martinez 1984). Our gateway design and interconnection will allow communication between two remote Sytek networks, with the ARPANET providing only the packet transport services. However, there will be no Sytek host to ARPANET host communication. The Sytek packets only traverse through the ARPANET subnet. Since no ARPANET host will be communicating with a Sytek host, there is no need for the ARPANET to interpret the message contained in the packets.

The key word in this section is encapsulation. There is no real protocol conversion that will be done at the gateway, i.e., there is no mapping done to translate the protocol of the Sytek network into the ARPANET protocol in

either direction. Encapsulation is an information-preserving transformation. The entire Sytek packet will be sent over the ARPANET as the data portion of an IP datagram (IP is the standard protocol used on the ARPANET). This will allow the Sytek packet to be carried transparently across the ARPANET. The gateway will merely perform the task of adding and deleting the IP headers. Phase 1 does not provide for flow and error control between the gateway and the Sytek nodes; these functions will be provided on an end-to-end basis using the LocalNet 20 higher layer protocols. The gateway is presented with a Sytek packet on one side. It encapsulates this packet in an IP datagram and transmits it over the ARPANET. The receiving gateway will decapsulate the IP header, yielding the original Sytek packet, which it proceeds to transmit onto the second Sytek network, addressed to a destination node. The basic function provided by the ARPANET in this type of interconnection is the transportation of the Sytek packets as datagrams. The ARPANET does not guarantee reliable delivery of these internet datagrams. This responsibility lies with the higher level Reliable Stream Protocol (RSP) at the two remote PCUs on the two Sytek networks. The RSP provides error free virtual connection to its users. The RSP assumes only a minimal datagram service from its lower level protocol. RSP achieves reliability through end-to-end acknowledgements and retransmission. With our gateway design, the

ARPANET will have no notion of the existence of a session between the two remote PCUs. It will transport each Sytek packet it receives independently between the two gateways. The mechanism of opening and closing sessions between the remote PCUs is still the job of the RSP at the two remote ends. This idea will be better understood when the gateway operation is explained in the next section. Having understood the function of the gateway and the role it plays in our interconnection scheme, we will look into the software architecture at the gateway.

The gateway software will essentially consist of four parts, (1) the LAP and PTP protocols, which reside on the CIU and the controller boards; (2) the IP protocol which will be implemented on the 86/30 board; (3) a bridging software on the 86/30 board that provides the required interface between the PTP and IP codes; and (4) the ARPANET 1822 driver software which will enable the IP to communicate with the IMP via the 1822 interface. The LocalNet software that will perform level 2 functions at the gateway reside on the CIU board. The level 3 PTP software runs on the controller board. The CIU and the controller communicate via two transmit and one receive queue, which are maintained in the multibus address space of each CIU board. The pointers to these queues are maintained individually by the CIU and the controller board. The CIU notifies the controller, using the flag byte of the queue, when it has received a

packet from a channel. The PTP, which polls the receive queues constantly, retrieves the queue pointer to the packet, and processes the packet (i.e., determines the destination channel of the packet and copies the packet to the appropriate transmit queue on a CIU board). The determination of the destination channel and the LAP address is the function of the PTP software. However, since the gateway interfaces one Sytek channel and an ARPANET channel, the function of the PTP at the gateway is to build up the routing table from the discovery mechanism. The IP software is implemented on the 86/30 board. The IP code was supplied to us by the MITRE Corporation, is written in "C" language, and runs under the CMOS operating system. The IP code was written at MITRE to run on an MC-68000 single-board computer and has portions of the code written in the 68000 assembly language. These portions have to be rewritten in 8086 assembly language. Also, portions of IP have to be modified to facilitate interface of the IP with the 1822 board. In addition, an interface software between the Sytek protocol and the IP protocol has to be written. This interface software has to mimic the functions of a CIU board, in addition to its function as a interface between the two network protocols. Recall that the ARPANET appears as another channel on the Sytek network. Therefore, when the gateway receives a packet from the Sytek network, the PTP on the controller board will transfer the packet to a destination

CIU board, which in our design is being mimicked by an 86/30 board. In order to mimic a CIU board, the 86/30 board has to maintain the transmit and receive queues and their associated pointers. After processing a packet, the controller transfers the packet to the 86/30 board, which appears as a CIU board to it. Once this transfer takes place, the interface software on the 86/30 will retrieve the Sytek packet and pass it on to the IP software, which in turn will append its header and pass it out onto the ARPANET, with the Sytek packet embedded as the data portion of an IP datagram. Thus the entire ARPANET will appear totally transparent to the Sytek network, and therefore no modifications to Sytek protocols will be required.

Gateway Operation

The basic building blocks of a Sytek-to-ARPANET gateway have been discussed in the previous section. The necessary background information on the ARPANET and Sytek networks has been described adequately to provide an understanding of the gateway operation. In this section, we will trace the flow of a packet from a source node on the Sytek network to a destination node on a remote Sytek network via the ARPANET. The process of encapsulation of the Sytek packet in an IP datagram at the gateway is discussed. The decapsulation of the IP datagram at the remote gateway and the subsequent routing of the packet from the gateway to the destination is described.

The objective of the gateway project is to allow communication between two remote Sytek networks, and the ARPANET merely acts as a vehicle for carrying the Sytek packets between the two remote networks. Communication between PCUs on a Sytek network takes place via virtual connections, i.e., the communicating PCUs are involved in a request and agreement procedure before the actual data transfer takes place. The establishment of this connection involves an exchange of messages between PCUs and is referred to as a virtual call setup. Each time a user requests a new LocalNet session, the discovery mechanism for establishing a route is employed. The discovery mechanism is intended to establish routes between user nodes without requiring any previous knowledge of either the destination's location or the network topology. The creation of a session requires that a connection be established between the source and destination PCUs, which may be located within the same LocalNet installation, but neither a PCU's cable nor its current channel affiliation are known prior to the session request. The discovery process results in the establishment of a route, and all subsequent data will be transferred along this route for the duration of the session. The role of the bridge in the discovery process has enabled us to customize a gateway which is quite a simple and straight-forward approach of interconnecting the two remotely located Sytek networks via the ARPANET. The ARPANET and the remote

Sytek network together appear as just another broadband channel to the Sytek network and the two Sytek networks are not even aware that they are tied together. The gateway and the ARPANET are totally transparent to the two Sytek networks. All messages which need to be sent from one Sytek network to the other are simply transmitted as if the destination node lies on the same Sytek LocalNet 20 broadband cable system. Thus one may think of the gateway assuming a dual role as: (a) an interconnection machine between the Sytek 20 network and the ARPANET, and (b) the two gateways and the ARPANET together can be assumed to play the role of an intercable "link" extending the connectivity between two Sytek LocalNet 20 cable systems. With this background, we will describe a scenario for opening a session between two remote PCUs via the ARPANET.

Sytek to ARPANET Packet Transfer

Assume that a user at node A on Sytek network 1 wants to communicate with a node B which happens to be on Sytek network 2 located at a remote location (Figure 4.1). The user at node A will initiate a session via its PCU A by sending out a discovery packet onto the channel to which it is attached. At this time the user node A knows only the unit ID of node B, but is unaware of either the destination's location or the network topology. The discovery packet is received by the gateway. The CIU board gets the

data from the modem board, deserializes it, puts it into a receive buffer, and sets the corresponding queue pointer and the associated flag byte. The flag byte alerts the controller to the arrival of a packet. The controller board, which periodically polls the receive queue, finds the flag set and proceeds to process the packet. The controller retrieves the buffer pointer from the queue and checks whether the received packet is a discovery packet (i.e., whether it has the special format). If it is, then the controller records the discovery attempt and broadcasts the discovery packet on to the ARPANET channel. The PTP interpreter on the controller board does this by copying the discovery packet onto the transmit queue in the 86/30 board. After copying the packet into the 86/30 transmit buffer, the controller sets the corresponding flag byte. The 86/30 board, which mimics the CIU board, periodically polls its transmit queue. Once the 86/30 board finds out that it has a packet waiting for it from the controller board, it retrieves the packet from the buffer and calls on the IP module to take the discovery packet as the data portion of an IP datagram. The IP module creates an IP datagram by appending the IP header to the Sytek packet. A point to note at this stage is that the IP does not distinguish between a discovery packet and an ordinary packet. It treats all packets received from the controller equally. Once the IP datagram is created, IP will call on the 1822

interface to transmit the packet to the IMP. The IMP receives the packet from the 1822 and extracts the destination gateway's address from the IP header. It will then address the packet to the destination gateway.

ARPANET to Sytek
Packet Transfer

The IP datagram from gateway 1 passes through the ARPANET and reaches the IMP, to which gateway 2 is attached. The IMP passes the packet to the 1822 board on gateway 2. The 1822, in turn, will transmit the data to a receive buffer as it arrives from the IMP. Finally, when the entire packet has arrived, it alerts the 86/30 board to the arrival of an ARPANET packet. The IP module will validate the packet and then strip off the IP header. At this point, we have the original discovery packet at the remote gateway. This packet is posted in the receive queue maintained on the 86/30 board and the flag is set. The controller board at gateway 2 thinks it has received a packet from a CIU board. The controller's PTP interpreter finds that it has received a discovery packet. It records the discovery attempt and then moves the packet from the 86/30 board onto the transmit queue on the CIU board. The CIU, in turn, sends the packet to the broadband channel. Eventually the discovery packet that originated at a remote Sytek network reaches a destination PCU on another network. The destination PCU upon receipt of the discovery packet will make a note of the link

address of the adjacent node, which happens to be gateway 2 in our application, from which it received the discovery packet. It then sends back a response packet which returns along the path followed by the first discovery packet. The response packet contains the link address of the destination as well as the channel number. When the response packet reaches gateway 2, the PTP interpreter there will update its routing table by noting the link address and the channel number of the destination PCU. The response packet then moves from gateway 2 to gateway 1 via the ARPANET. Gateway 1 will update its routing table and will send the response packet on to the Sytek network. The source on network 1 updates its routing table by recording the link address of the node from which it received the response packet, which happens to be gateway 1. All future packets between the two remote nodes will follow the route thus set up. No further discovery packets are required to be sent for the duration of the current session between the two remote PCUs.

CHAPTER 5

SUMMARY AND CONCLUSIONS

Current Status

In this thesis we have presented the design of a gateway between LocalNet 20 and the ARPANET. However, at the time of completion of this thesis, the hardware and software modifications necessary to convert a Sytek bridge product to a gateway were implemented and the individual components tested at our laboratory. The interface software between the LocalNet 20 protocol and the 1822 driver software were developed on Intel's 86/380X development system and downloaded onto the 86/30 board for testing. Our initial gateway will connect a single 128-kilobit Sytek LocalNet 20 channel to the ARPANET. During the integration and testing phase, the gateway will be located at the University of Arizona. A 9.6-kilobit voice-grade telephone line will provide the connection between the gateway and the IMP which is installed at Fort Huachuca, Arizona, about 70 miles away. Eventually the tested gateway will be placed very close to the IMP at Fort Huachuca. The gateway will be put into operation and ready for demonstration by August 1984.

Summary

In this thesis the design and operation of a gateway for interconnecting a Sytek LocalNet 20 broadband network and the ARPANET is described. A scenario describing the communication between two remotely placed Sytek networks using the ARPANET as a packet transport medium is discussed. The key features of the gateway design are:

(a) Internet transparency. Communication between the two remote Sytek networks over the ARPANET is totally transparent, meaning that from a user point of view, communication between nodes on remote Sytek LANs is identical to communication between nodes on the same LAN.

(b) Using off-the-shelf products. The gateway implementation makes use of components all of which are available from different vendors. The most important design effort is in understanding the LocalNet Protocol and the Internet Protocol and in coming up with a hardware and software architecture for the gateway.

(c) Extension of the broadcast discovery concept. Extending the idea of Sytek's broadcast technique of discovering the best path to the destination node to provide an efficient mechanism for transporting packets between remote Sytek networks. The goals of the discovery technique were to provide

> interconnection between nodes in a multi-network system without requiring prior knowledge of either the node location or the network topology. We feel that this goal is achieved by a design that makes the ARPANET appear as just another Sytek broadband channel.

Subsequent phases of the project will investigate methods of using the Transport Control Protocol (TCP) at the gateway. However, a justification must be made for using TCP at the gateway at all, because the TCP gateway will be more complicated than an IP gateway. A brief discussion of the advantages of a TCP over an IP gateway may provide further insight into the need for a TCP gateway. In phase 1 (the current phase), the gateway will implement only the Internet Protocol (IP). Communication services such as error control and packet acknowledgement are provided on an end-to-end basis, meaning that a session will be established between the source and destination nodes on the remote networks before any data transfer takes place. The end-to-end approach assumes only an unreliable datagram service across the ARPANET. Reliability is achieved by performing end-to-end procedures for reordering, retransmission due to losses and detection of duplicates. Thus, end-to-end reliability is provided by entirely depending upon Sytek's LocalNet 20 protocols. This approach is particularly attractive because all the nodes that will be involved in internet

communication will be on the two Sytek networks, implementing a common protocol. The main attraction of an end-to-end approach of internetworking with IP gateways is the conceptual simplicity of the gateways. Since the gateways provide only datagram service, they do not need to store any connection-oriented information, and thus require no extra buffer space. The gateway merely has to encapsulate all the packets it receives under an IP header and send them out onto the ARPANET. In short, end-to-end approach requires minimum gateway service. However, in an end-to-end approach, packet acknowledgements may give rise to excessive transmission delays. Time out periods and maximum retransmission counts on the Sytek network would have been optimized for its network speed and delay. The ARPANET, which is much slower than the Sytek network, and the gateway will introduce delays and therefore the time out periods and retransmission counts may require further tuning to accommodate these additional delays. Assuming that there will be several simultaneous interconnections in progress at some point in time, there could be a situation when a gateway may receive retransmitted packets from different sources, which may overwrite the earlier received packets even before the first arrived packets were processed and transmitted to the ARPANET. This may cause reliability problems because the gateway could get flooded with transmitted packets faster than it can process and transmit the previously received

packets to the slower ARPANET. Sytek's network can operate at 128 kilobits per second and the gateway-to-IMP link is 50 kilobits per second maximum, a significant difference in speed. Implementing TCP at the gateway will overcome the above problems. TCP will provide flow control and acknowledgements between a Sytek's source node and the gateway, thus avoiding flooding due to retransmitted packets. TCP will provide transmission service between gateways across the ARPANET, and therefore internet packets can be acknowledged from the gateway to the source instead of waiting for acknowledgements to come from the remote destination node. Thus a TCP gateway will isolate the time out and retransmission problems caused by additional delays due to the ARPANET and the gateway. The above approach of acknowledging packets from the gateway using TCP is referred to as a hop-by-hop method, which implies that totally reliable service will be provided by concatenation of services: first, a virtual connection between the source node and the gateway, then a virtual connection between the two gateways, and finally, a virtual connection between the gateway and the destination node. The hop-by-hop approach will require TCP and RSP at the gateways. The gateway will now have to do complex translation between TCP and RSP. In other words, the gateway must do explicit mapping of session calls and termination of calls, and also must maintain connection-oriented information.

Subsequent phases of the gateway development activity will be targeted toward developing a generic gateway, which can provide an interface between different types of LANs to the ARPANET and other long-haul networks. Implementing a generic gateway may require significant protocol translation, because each LAN may have its own unique network protocol. If the interconnection is going to be provided primarily by the ARPANET, which has proven to be a low-loss network, then it may be advantageous to keep the gateway simple by implementing only the IP and relying on the end-to-end schemes to provide the required reliability. However, this approach requires tuning the LAN retransmission time out periods to take care of additional delays introduced due to the ARPANET and the gateway, but may be easier than implementing a TCP gateway. Providing more buffer space than is now present may be another solution to overcoming the problem of packets getting overwritten due to retransmitted packets. If there is a small buffer space, and the gateway is not able to process packets fast enough, then the newly arrived packets will overwrite the older unprocessed packets in the buffer. This could be a major problem with an end-to-end approach.

We can summarize from the above discussions that it seems more attractive to implement an IP gateway, especially if the remote LANs are of the same type and involve a common end-to-end protocol at all sites. If the remote LANs being

interconnected are heterogeneous (different manufacturers), then end-to-end is not feasible because the two remote communicating nodes implement two different protocols and hence cannot understand each other. In such heterogeneous interconnections, a hop-by-hop approach seems to be the only possible solution.

Another point to be noted is that the present Sytek-ARPANET gateway imposes a certain penalty for making use of the bridge product in its design. All broadcast packets that originate on either of the two networks will be transmitted to the other network, regardless of whether the discovery packet was destined for the remote network or not. This is because the ARPANET appears as another Sytek channel to the bridge, and therefore all the discovery packets that arrive at the bridge will be broadcast to all the channels to which the bridge is connected, including the ARPANET channel. This may lead to a considerable overhead if the Sytek channel that is connected to the gateway produces a number of broadcast packets and the unintended packets for the remote network will be needlessly sent over to the remote network via the ARPANET. Thus, if there is a large internet traffic and local traffic simultaneously, then the discovery packets meant for local communication within the same Sytek network will occupy valuable buffer space, and the gateway become a bottleneck. Also because of ARPANET delay, a source may time-out and retransmit several times

before it gets an acknowledgement. These retransmitted packets will be sent over the ARPANET, producing unnecessary traffic.

Another aspect that will need further study is the addressing scheme. Currently Sytek protocol uses a flat or constant addressing scheme in which each node is assigned a unique constant ID, with the idea that the node can move freely between channels. This scheme differs from the hierarchical addressing scheme in which the address part of the packet carries an explicit network number on which the destination resides. Flat addressing will cause problems in an internetworking scheme such as the one under discussion. Care must be taken that no two nodes involved in an internet communication should have the same number, since the internetworking makes the connected networks appear as one large network and the node numbers have to be unique over the entire address space. Sytek protocol uses a 16-bit node ID address field, and if several Sytek networks were to be interconnected, care must be taken to see that there is no duplication of unit IDs on any of the networks. A possible solution might be to modify the present addressing scheme to include a network number similar to the one used in Xerox's Internet Transport Protocol (Xerox, Internet Transport Protocol 1981), to combine a network address with the absolute node address. This will allow duplication of addresses, as the node ID is now unique to the network to

which it is connected. Also, this addressing scheme will greatly help internetwork communication because the gateway can now make use of the network address in determining whether the discovery packet should be sent to the remote network via the ARPANET. If the network address was not that of the remote network, then the gateway can discard the discovery packet, thus avoiding needless transmission over the ARPANET. Thus, as the gateway project evolves stage by stage, we will have to investigate newer problem areas and different design issues. The design methodology may have to be refined further as more and more of the requirements become clear.

Conclusions

The purpose of the gateway project at the University of Arizona is to facilitate interconnection of two remotely placed Sytek LANs via the ARPANET. The gateway will provide the interface between the Sytek LAN and the ARPANET. There will be two identical gateways involved in the interconnection. The ARPANET subnet will serve as a vehicle for transporting the Sytek packets between the remote locations.

The IP implementation at the gateway has resulted in a simple gateway design. The encapsulation of packets will allow the internet communication to be totally transparent. Remote nodes can communicate as if they were on the same Sytek LAN. This design approach does not require any modifications to the LocalNet 20 protocol.

The next stage of this project will incorporate TCP at the gateway. Using TCP will insulate the retransmission and time-out parameters used in the Sytek LAN from the ARPANET delays. Because of the delay, the Sytek LAN will generate retransmission packetsd, which will be transmitted over the ARPANET. Since TCP will provide reliable transport between the two gateways, packets can be acknowledged by the gateway, thus eliminating retransmissions which would have caused unnecessary traffic over the ARPANET.

We have shown that a modular design of the gateway is possible by making use of off-the-shelf hardware. The modular approach allows the network interface part to be customized easily to suit various types of network architectures. The design concepts discussed in this thesis could be used to develop generic gateways between LANs and long-haul networks.

APPENDIX

Glossary

address - A LocalNet address is a two-part entity specifying a particular user device attachment point. An address consists of a unit ID specifying the PCU and a port ID specifying the particular port on that PCU.

bridge - A bridge is a LocalNet product that is used to route packets among LocalNet channels on either the same or different broadband cables.

channel - A LocalNet channel is a frequency division multiplexing digital multipoint channel occupying 300 KHz of analog spectrum and modulated at 128K bps. One hundred twenty such channels can be configured on each broadband cable and each channel can be shared among a number of PCUs, using the CSMA/CD distributed channel arbitration mechanism. Twenty channels are in each "channel group," and each group is like a standard 6 MHz commercial video channel.

channel No. - ChannelNo is a numeric value in the range 0 to 119, identifying one of the LocalNet channels. A PCU's current channelNo (along with its channel group) specifies the LocalNet channel to which its modem is tuned.

CIU - Channel Interface Unit is a bridge component that has the primary functions of modem channel selection (tuning) and packet transfer between the modem and the controller.

Controller - A bridge component that handles all the routing functions between channels.

Datagram - A finite-length packet of data with a complete destination address and source address. In a datagram service, the network simply accepts packets from a source and attempts to deliver each packet independently from the other, and makes no attempt to do error recovery. Messages may arrive out of order or not at all.

Host - In any network there exists a collection of machines intended for running user programs. These machines are usually called hosts.

IMP - Interface Message Processors are small dedicated communications processors that provide subnetwork access to the host computers. All traffic to or from each host to the ARPANET goes via its IMP.

IP - Internet Protocol provides for delivery of individual messages (datagrams) with high but not perfect reliability in an interconnected system of networks.

LAP - Link Access Protocol is a data link level protocol within the LocalNet and provides the function of media access and framing of messages.

PCU - Packet Communications Unit is a LocalNet product thta connects terminals or other devices to the broadband network. Data communications intelligence is built into each PCU.

PTP - Packet Transfer Protocol is the network layer protocol and provides datagram-based addressing and routing of packets within the LocalNet.

protocol - A set of rules and conventions used in exchanging formatted information packets between correspondents. In most networks, protocols are organized as a series of layers or levels and each layer has a specific function. The purpose of each layer is to offer services to the higher layers, shielding those layers from the details of how the offered services were actually implemented.

RSP - Reliable Stream Protocol, provides reliable, flow-controlled transfer of data via a virtual connection.

TCP - Transmission Control Protocol is an end-to-end reliable protocol which enables interprocess communication between hosts in a multi-network system.

Virtual Connection - A logical connection across a packet-switched network that ensures reliability by providing error-free and sequenced delivery of packets.

REFERENCES

Advanced Computer Communications, M/1822 User's Manual. Santa Barbara, CA: December 1983.

Benhamou, Eric, and Judy Estrin, "Multilevel internetworking gateways: Architecture and applications." Computer, September 1983.

Bolt, Beranek, and Newman, Inc., Report No. 1822: "Interface Message Processor - Specifications for the Interconnection of a Host and an IMP." December 1981.

Braden, R., et al., "A distributed approach to the intercon nection of heterogeneous computer networks." SIGCOMM '83 Symposium on Communications Architectures and Protocols, Univ. of Texas at Austin, March 1983.

Cerf, Vinton G., and Robert E. Kahn, "A protocol for packet network interconnection." IEEE Transactions on Communications, Vol. Com-22, No. 5, May 1974.

Cerf, Vinton G., and P. Kirstein, "Issues in packet-network interconnection." Proc. of the IEEE, Vol. 66, No. 11, November 1978.

Comer, Douglas, and John T. Korb, "CSNET protocol software: the IP-to-X.25 interface." SIGCOMM '83 Symposium on Communications Architectures and Protocols, Univ. of Texas at Austin, March 1983.

Digital Equipment Corporation, Intel Corporation, and Xerox Corporation, "The ETHERNET: A local area network, data link layer, and physical link layer specifications," Version 2.0. November 1982.

Ennis, G., and P. Filice, "Overview of a broadband local area network protocol architecture." IEEE Journal on Selected Areas in Communications, Vol. Sac-I, No. 5, November 1983.

Estrin, Judy, and Bill Carrico, "Gateways promise to link local networks into hybrid systems." Electronics, September 1982.

Gien, M., J. Lans, and R. Scantelbury, "Interconnection of packet-switching networks." Computer Networks, 1975.

Green, P. E., "Special issue on computer architecture and protocols." IEEE Transactions on Communications, Vol. Com-28, April 1980.

Haverty, Jack, and Robert Gurwitz, "Protocols and their implementation: A matter of choice." Data Communications, March 1983.

Heart, F. E., et al., The Interface Message Processor for the ARPA Computer Network, Chapter 24, Computer structures: Principles and Examples. Prentice-Hall, 1982.

Hinden, Robert, Jack Haverty, and Alan Sheltzer. "The DARPA internet: Interconnecting heterogeneous computer networks with gateways," Computer, September 1983.

Information Sciences Institute, "DoD standard internet protocol." Los Angeles: University of Southern California for DARPA [Defense Advanced Research Projects Agency], January 1980.

Information Sciences Institute, "DoD standard transmission control protocol." Los Angeles: University of Southern California for DARPA, January 1980.

Intel Corporation, "iSBC 86/14 and iSBC 86/30 single board computer hardware reference manual."

Intel Corporation, "Intel multibus specification," 1982.

Martinez, Ralph R., "ECE 478: Data Communication Networks." Class Notes, Department of Electrical Engineering, University of Arizona, Tucson, Fall 1982.

Martinez, Ralph R., et al., "Gateway design methodology." Tucson, Arizona: University of Arizona Engineering Experiment Station Interim report No. 2, January 1984.

Martinez, Ralph R., et al., "Issues in gateway development." Tucson, Arizona: University of Arizona Engineering Experiment Station, Interim report No. 1, July 1983.

Postal, J. B., "Internetwork protocol approaches." IEEE Transactions on Communications, Vol. Com-28, No. 4, April 1980.

Postal, J. B., Carl A. Sunshine, and Danny Cohen, "The ARPA internet protocol." Computer Networks, July 1981.

Rauch-Hinden, Wendy B., "Upper level network protocols." Electronic Design, March 3, 1983.

Schneidewind, Norman F., "Interconnecting local networks to long-distance networks." Computer, September 1983.

Sheltzer, Alan, Robert Hinden, and Mike Brescia, "Connecting different types of networks with gateways." Data Communications, August 1982.

Schultz, G., "CIU multibus interface." Sytek, Inc., 1983.

Sunshine, C. JA., "Interconnection of computer networks," Computer Networks, Vol. 1, No. 3, 1977.

Sunshine, C., et al., The Interface Message Processor for the ARPA Computer Network, Chapter 24, Computer structure: Principles and Examples. Prentice-Hall, 1982.

Sytek, Inc., "LocalNet 20--Reference Manual and Installation Guide." Sytek: Users' Manual, 1983.

Sytek, Inc., "LocalNet 20--Bridge with Link Option." Sytek: Users' Manual, 1983.

Xerox Corporation, "Internet transport protocols." December 1981.

Zimmerman, Herbert, "OSI reference model--The ISO model of architecture for open systems interconnection." IEEE Transactions on Communications, Vol. Com-28, No. 4, April 1980.