Chapter 1:

Network Architecture

1.1 Introduction

Current telecommunications services--which include circuit- or packet-switched data, private lines, or voice–are quite often provided by independent and parallel networks. Since each network is optimized for the application it supports, one is not efficient in supporting other applications.

In an era of worldwide competition, end users as well as network providers want the flexibility and versatility to respond quickly to the changing environment. The ability to define, manage, and monitor services, along with the ability to introduce new services tailored to specific applications, are characteristics expected of the networks of the future.

The idea of an integrated communications system that can support a diversity of services with different requirements has been discussed for some time. The Integrated Services Digital Network (ISDN) is an attempt to provide such an integrated access. ISDN specifies a standard interface based on digital transmission. It consists of a number of circuit-switched B channels for voice and bulk data, and a packet-switched D channel for signaling data. As such, ISDN is not a truly integrated network. Using a limited set of well-defined interfaces, ISDN allows user traffic to be integrated at the transmission level at the user network interfaces. Within the network, separate switching methods carry circuit-switched and packet-switched connections.

Providing a single network that concurrently serves many users with differing needs has a number of benefits because of the economy of scale in telecommunications. The network should be responsive to interactive traffic; provide high bandwidth for file transfer, high-speed data, and video applications; and have a low delay for packet voice. Each of these functions requires a different transport protocol and different data representation. The challenge in designing broadband networks and switches is to accommodate this diversity without increasing the complexity. What is needed is a structure with simple components, simple interfaces, and simple protocols.

The goals of an integrated network based on a single switching technique are good performance, effective resource utilization, and unified network management, operation, and maintenance to support various existing services uniformly.

Another important goal is to provide the flexibility to respond to and support future services.

The distributed processing capability requires fast access to on-line information and expensive computing resources. The use of graphics-based user interfaces with diskless processing and networked file systems places the network at the center of distributed processing. Within a limited geographical distance, local area networks, with a capacity of a few megabits per second, can support a few tens of computers and servers. The networks provide low delay, and their intrinsically low error rate allows them to use simple protocols. The persistent demand for higher bandwidth interconnections has led to the next generation of LANs, which can support higher bandwidths (100 Mbps). Although these LANs provide higher bandwidths, they have not solved the distance problem.

Therefore, a high-performance network covering a wider area–say with the capability to support thousands of end users scattered over an entire metropolitan area–is needed. Such metropolitan area networks (MANs) will likely be owned and operated by common carriers. Issues such as reliability and security are of greater importance than in a LAN environment, which is usually owned and operated by the organization that uses it.

Perhaps the most important factor spurring the change is the steady increase of transmission bandwidth available at an affordable cost. In the local area network environment, cheap bandwidth has already radically changed the structure of local area communication. In the future, both near-term and long-term, there will be a definite need for connecting high-bandwidth equipment with similar equipment at other locations, and with central information or processing facilities.

The frame-based transfer capabilities, such as frame relaying and switched multimegabit data service, are candidates for near-term high-speed data applications. Alternative solutions for multimedia applications range from the Fiber Distributed Data Interface for integrated services LANs and the Distributed Queue Dual Bus (DQDB) for MANs, to the asynchronous transfer mode (ATM) for wide area public networks. A communication structure based on ATM requires new design rules, new control strategies, and new architectural approaches.

At any given level of circuit technology, it is possible to transmit and receive data much faster than software, executed by the same circuit technology, can process all the communication protocols. With the exception of digital video, it is almost impossible for any one application to cope with all the data that can flow through optical fibers. Hence, the most economical approach would be to share the medium. With a shared-medium network, the switch would also be shared. Any station on the net would communicate directly with any other. It would be possible to have graceful growth, with the initial investment being mainly in the cabling.

A broadband network will enable a single network to support voice, video, image, and data services. There is, of course, uncertainty about the type of and the demand for new services, and the traffic mix of services at particular times. The effects of this uncertainty can be minimized by sharing the network resources across different services. As traffic peaks often occur at different times for different services, the network can be dimensioned for a desired service quality during busy hours. Thus, the flexibility of broadband integrated networks enables them to provide service for traffic with different characteristics and also to cope with the uncertainty of the traffic mix of the future.

Integrated access allows statistical multiplexing and dynamic reallocation of resources without reconfiguring the access channels, and also provides bandwidth on demand. For the network provider, an integrated network provides transmission efficiency and versatility. Network management and control are simple.

The layers, interfaces, and protocols in a network constitute the network architecture. The communications services and applications of the future will require higher bandwidths and flexible architectures to support multiple classes of traffic. In such a network architecture, a fiber-optic medium can provide cost-effective broadband transmission capabilities. The other components that play an important role are the switching and the protocols. This chapter presents the various trends in network architectures intended to meet the high bandwidth demanded by users.

1.2 Overview of Papers

Tanenbaum's paper, "Network Protocols," is a tutorial on protocols as well as networking. Tanenbaum presents an overview of the circuit and packet switching technology along with telephone, satellite, and local area networks. The network architecture model in this paper closely

follows that of the International Organization for Standardization's reference model of Open Systems Interconnection (ISO OSI), and uses the ARPAnet terminology to make the abstract concepts of network architecture and protocols more concrete. Tanenbaum uses the X.21 standard to illustrate the physical layer protocol, and High-level Data Link Control (HDLC) to explain the data link layer protocol.

Tanenbaum provides a detailed explanation of the sliding-window protocol and its use in the X.25 packet-level protocol and discusses various routing schemes and congestion control techniques. The transport layer, which is responsible for providing an independent transport service to the session layer, is illustrated using the concept of a "transport station." Transport layer functions such as establishing and closing connections, flow control and buffering, and connection multiplexing are examined. The presentation layer, which performs a useful data transformation function, is explained using text compression and encryption.

Hoberecht's paper, "A Layered Network Protocol for Packet Voice and Data Integration," presents a layered packet protocol architecture that can be used for integrating voice and data in packet-switched networks. Hoberecht identifies network components and the factors that affect the protocol architecture, and defines a layered protocol for voice/data packet networks and the layer functions. After introducing the concept of an edge-to-edge error-recovery protocol, Hoberecht presents the delay analysis for data traffic using edge-to-edge and the traditional link-level error-correction protocols.

The Fiber Distributed Data Interface (FDDI) is the subject of Ross's paper. FDDI is a fiber-optic token-passing network that uses a dual-ring configuration for reliability. The rings are counterrotating, one clockwise and the other counterclockwise, and each station is connected to both the rings. Optical bypass switches can be used to remove any failed station from the network without disturbing the traffic flow to the other stations. FDDI is an upgraded version of the IEEE 802.5 token ring, with a transmission rate of 100 Mbps and a span of 100 kms. In addition to serving as a backbone LAN, it can be used to interconnect host computers in a computer-room environment. While FDDI operates only in the packet data mode, an upward-compatible enhancement, FDDI-II, will also provide circuit-switched capacity on demand.

Newman et al. describe the QPSX (queued packet and synchronous switch) MAN (metropolitan area network). It is a distributed switch/network based on a dual self-healing bus architecture. The head-ends generate time slots that propagate along the buses, and a user having data to send waits until an idle time slot appears at the sending location. Access to an idle time slot is determined by a distributed queueing algorithm, and the user writes data onto the appropriate bus to be transmitted toward the intended destination. This dual-bus architecture provides higher reliability by moving the logical gap in the system to the point of physical failure, so the system can operate using both the buses. QPSX provides high-speed packet and circuit switching capability at a transmission rate of around 150 Mbps. Although QPSX effectively provides both circuit and packet switching, the flexibility and versatility of the network are due to its packet switching protocol.

The switched multimegabit data service (SMDS) is discussed in the paper by Hemrick et al. The idea behind SMDS is to allow users to move high-speed data using a shared switched network. The motivation for this service is to interconnect LANs and handle the increase in applications based on distributed processing–for example, real-time access to files, desktop publishing, and shared high-resolution graphics in computer-aided design.

SMDS is a high-performance, connectionless, packet-switched data service designed to be easily integrated into the users' existing local environment. SMDS provides efficient and easy-to-implement protocols and procedures across the access interface. The connectionless packet transport is provided by the SMDS interface protocol, which provides error-detection, framing, and addressing functions within the network. The interfaces are designed to operate at 45 Mbps.

Lai's paper, "Packet Mode Services: From X.25 to Frame Relaying," traces the evolution of packet mode services from X.25 to frame relaying. While X.25 uses in-band signaling and control, frame relaying uses out-of-band signaling and control. The interworking between X.25 and frame relaying is also discussed.

The paper by Minzer, "Broadband ISDN and Asynchronous Transfer Mode," compares the limitations of the synchronous transfer mode (STM) with the flexibility of the asynchronous transfer mode (ATM), thus explaining the choice of ATM as the transfer mode for broadband ISDN.

The major technical considerations influencing the undefined ATM specifications are also discussed.

"The Evolution of Broadband Network Architectures," by Batorsky et al., explains the two major forces behind the emergence of broadband networks: the market for new services and technological advances. The authors describe the architecture of broadband networks for the long term and suggest approaches by which the initially available architectures might evolve toward the long-term architecture.

The last paper, "SONET: Now It's the Standard Optical Network," by Ballart and Ching, discusses the important technical attributes of the SONET (Synchronous Optical Network) standard. The advent of fiber-optic transmission systems with very large bandwidths has led to a new, simpler synchronous digital multiplexing hierarchy, which offers flexible multiplexing of high-speed channels. A major advantage of the system is the ease with which channels can be dropped and inserted at the network nodes. SONET defines standard optical signals, a synchronous frame structure for multiplexed digital traffic, and operations procedure. SONET forms part of the new synchronous signal hierarchy.

Network Protocols

ANDREW S. TANENBAUM

Wiskundig Seminarium, Vrije Universiteit, Amsterdam, The Netherlands

During the last ten years, many computer networks have been designed, implemented, and put into service in the United States, Canada, Europe, Japan, and elsewhere. From the experience obtained with these networks, certain key design principles have begun to emerge, principles that can be used to design new computer networks in a more structured way than has traditionally been the case. Chief among these principles is the notion of structuring a network as a hierarchy of layers, each one built upon the previous one. This paper is a tutorial about such network hierarchies, using the Reference Model of Open Systems Interconnection developed by the International Organization for Standardization as a guide. Numerous examples are given to illustrate the principles.

Key Words and Phrases: computer network, data communication, ISO OSI Reference Model, layered architecture, network, protocol

CR Categories: 1.3, 4.9, 6.9

INTRODUCTION

Ten years ago, only a handful of computer networks existed, mostly experimental networks built by research organizations. Today dozens of national and international networks and innumerable local networks operate on a commercial basis around the clock. From the beginning, many networks were designed hierarchically, as a series of layers, each one building on the one below. At first, each network design team started out by choosing its own set of layers. However, in the past few years, a consensus has begun to develop among network designers, a consensus embodied in the International Organization for Standardization's Reference Model of Open Systems Interconnection (ISO OSI). In this paper we present an informal introduction to computer networking using this model as a guide. A more thorough treatment of the ISO OSI model itself can be found in ZIMM80.

Before getting into the subject of network protocols, it is worth saying a few words about what we mean by a computer network. A computer network is a collection of computers, called *hosts*, that communicate with one another. The hosts may be large multiprogrammed mainframes or small personal computers. Networks can be classified as *local networks* or *long-haul networks*. The hosts on a local network are typically contained in a single building or campus and are connected by a high-bandwidth cable or other communication medium specifically designed for this purpose. Long-haul networks, in contrast, typically connect hosts in different cities using the public telephone network, an earth satellite, or both.

Local networks are nearly always completely owned by a single organization, whereas long-haul networks normally involve at least two organizations: the *carrier*, which operates the communication facility (telephone lines, microwave dishes, satellite, etc.), and the users, who own the hosts. This division of labor into (1) the provider of the communication facility and (2) the

Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the ACM copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Association for Computing Machinery. To copy otherwise, or to republish, requires a fee and/or specific permission.
© 1981 ACM 0010-4892/81/1200-0453 $00.75

"Network Protocols" by A.S. Tanenbaum from *Computing Surveys*, Vol. 13, No. 4, December 1981, pages 453-489.
Copyright © 1981 by The Association for Computing Machinery, Inc., reprinted by permission.

CONTENTS

INTRODUCTION
Protocols
Overview of the ISO OSI Layers
1. THE PHYSICAL LAYER
1.1 The Telephone System
1.2 Communication Satellites
1.3 Local Networks
1.4 An Example Physical Layer Protocol: X.21
2. THE DATA LINK LAYER
2.1 Stop-and-Wait Protocols
2.2 Sliding-Window Protocols
2.3 An Example Data Link Layer Protocol: HDLC
2.4 Channel Allocation in Satellite Networks
2.5 Channel Allocation in Local Networks
3. THE NETWORK LAYER
3.1 Routing in Point-to-Point Networks
3.2 Congestion Control in Point-to-Point Networks
3.3 An Example Network Layer Protocol: X.25
4. THE TRANSPORT LAYER
4.1 The Transport Station
4.2 Establishing and Closing Connections
4.3 Flow Control and Buffering
4.4 Connection Multiplexing
5. THE SESSION LAYER
6. THE PRESENTATION LAYER
6.1 Text Compression
6.2 Encryption Protocols
6.3 Virtual-Terminal Protocols
6.4 File Transfer Protocols
7. SUMMARY
ACKNOWLEDGMENTS
REFERENCES

users of the communication facility has important ramifications for network architectures, as we shall see later.

The communication facility in a long-haul network is called the (*communication*) *subnet*, and often consists of a collection of minicomputers variously called *IMPs* (interface message processors), *nodes*, or *switches* connected by high-bandwidth leased telephone lines or a satellite. Figure 1 shows a network using telephone lines. Such a network is called a *point-to-point* or *store-and forward* network, as opposed to a *broadcast* network, such as a satellite network. The terms "host," "IMP," and "communication subnet" come from the U.S. Department of Defense's ARPANET, one of the first large-scale networks [McQu77]. We use this terminology generically because no consensus on nomenclature exists.

When the IMPs are connected by telephone lines, they are normally located on the carrier's premises, with each IMP servicing multiple hosts. To save on long-distance leased-line line charges, hosts and terminals are often funneled through remote concentrators. When the IMPs are connected by a satellite, the IMPs may be located on the customer's premises (e.g., on the roof). Local networks do not have IMPs; instead, each host has an interface card inserted into its backplane to control access to the network. This card is attached to the communication subnet, which is typically just a cable.

Although the ISO Reference Model can be used for both long-haul and local networks, it was designed primarily with the former in mind. Accordingly, in this paper we also treat both kinds of networks, but we emphasize slightly the long-haul variety, since issues such as routing and congestion control play a more prominent role in long-haul networks than in local networks.

In passing, we note that the subject of connecting distinct networks together is an increasingly important one, although it lies beyond the scope of this article. For an introduction to this subject see Bogg80 and Post80.

Protocols

As mentioned above, networks are almost always organized as a hierarchy of layers. Each layer performs a small set of closely related functions. The ISO Reference Model has seven layers:

(1) the physical layer,
(2) the data link layer,
(3) the network layer,
(4) the transport layer,
(5) the session layer,
(6) the presentation layer,
(7) the application layer,

as shown in Figure 2. All layers are present on the hosts, but only layers 1, 2, and 3 are present on the IMPs.

Each layer should be thought of as a program or process (possibly embedded in a hardware device) that communicates with

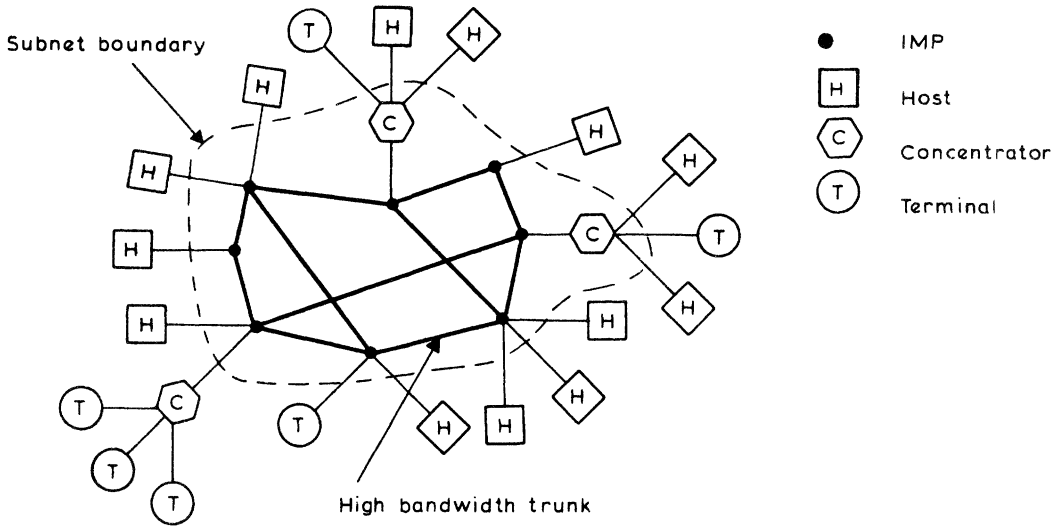


Figure 1. A typical point-to-point long-haul network.

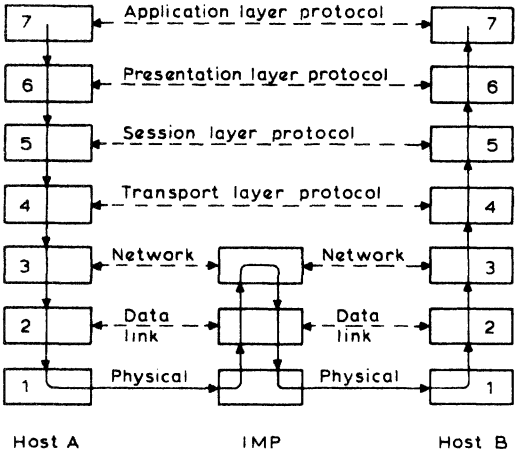


Figure 2. The seven-layer ISO Reference Model.

the corresponding process on another machine. In Figure 2, host layers 1, 2, and 3 think that they are communicating with their corresponding layers on the IMP, called *peers*. (In this example, hosts *A* and *B* are serviced by a common IMP; in general, multiple IMPs may intervene.) Layers 4–7, in contrast, communicate directly with their peer layers on the other host. The rules governing the layer *k* conversation are called the *layer k protocol*. The ISO model thus has seven protocols.

In reality, data are not transmitted horizontally, from machine to machine within a given layer, but are passed vertically down the layers of the sending machine and up the layers of the receiving machine. Only in layer 1 does actual intermachine communication occur. When an application program, running in layer 7 on host *A*, wants to send a message to the application in layer 7 on host *B*, it passes the message down to the presentation layer on its own machine. The presentation layer transforms the data, adds a layer 6 *header* containing control information used by the layer 6 protocol, and passes the resulting message down to the session layer. The session layer then adds its own header and passes the new message down to the transport layer. The complete path from layer 7 on host *A* to layer 7 on host *B* is shown in Figure 2 by the solid line. The boundary between adjacent layers is called an *interface*. The layers, interfaces, and protocols in a network form the *network architecture*.

No layer is aware of the header formats or protocols used by other layers. Layer *k* on the sending machine regards its job as getting the bits that come in from layer $k + 1$ over to the receiving machine somehow (using the services of the lower layers). It neither knows nor cares what the bits mean.

A three-layer analogy may be helpful in understanding how multilayer communication works. Consider the problem of the

two talking philosophers. Philosopher 1 lives in an ivory tower in Kenya and speaks only Swahili. Philosopher 2 lives in a cave in India and speaks only Telugu. Nevertheless, Philosopher 1 wishes to convey his affection for *Oryctolagus cuniculus* to his Indian colleague (the philosophers are layer 3 peers). Since the philosophers speak different languages, each engages the services of a translator (layer 2 process) and an engineer (layer 1 process).

To convey his thoughts, Philosopher 1 passes his message, in Swahili, to his translator, across the 3/2 interface. The translator may convert it to English, French, Dutch, or some other language, depending only on the layer 2 protocol. The translator then hands his output to his engineer across the 2/1 interface for transmission. The physical mode of transmission may be telegram, telephone, computer network, or something else, depending only on the layer 1 protocol. When the Indian engineer receives the message, he passes it to his translator for rendition into Telugu. Finally, the Indian translator gives the message, in Telugu, to his philosopher.

This analogy illustrates three points. First, each person thinks of his communication as being primarily horizontal, with his peer (although in reality it is vertical, except in layer 1). For example, Philosopher 1 regards himself as conversing with Philosopher 2, even though his only physical communication is with translator 1. Second, actual communication is vertical, not horizontal, except in layer 1. Third, the three protocols are completely independent. The philosophers can switch the subject from rabbits to guinea pigs at will; the translators can switch from English to Dutch at will; the engineers can switch from telegram to telephone at will. The peers in any layer can change their protocol without affecting the other layers. It is for precisely this reason that networks are designed as a series of layers—to prevent changes in one part of the design (e.g., caused by technological advances) from requiring changes in other parts.

Overview of the ISO OSI Layers

The remainder of this article concerns the various layers in the ISO Reference Model, one section per layer. Before looking at the layers in detail, we first present a brief overview of each layer, to put the hierarchy in perspective.

The physical layer protocol is concerned with the transmission of a raw bit stream. Its protocol designers must decide how to represent 0's and 1's, how many microseconds a bit will last, whether transmission is full- or half-duplex, how the connection is set up and torn down, how many pins the network connector has, what each pin is used for, and other electrical, mechanical, and procedural details.

The data link layer converts an unreliable transmission channel into a reliable one for use by the network layer. The technique for doing so is to break up the raw bit stream into frames, each containing a checksum for detecting errors. (A *checksum* is a short integer that depends on all the bits in the frame so that a transmission error will probably change it and thus be detectable.) The data link protocol usually ensures that the sender of a data frame will repeatedly transmit the frame until it receives an acknowledgment frame from the receiver.

The network layer in a point-to-point network is primarily concerned with routing and the effects of poor routing, namely, congestion. In a broadcast network, routing is not an issue, since only one channel exists.

The task of the transport layer is to provide reliable host-to-host communication for use by the session layer. It must hide all the details of the communication subnet from the session layer, so that, for example, a point-to-point subnet can be replaced by a satellite link without affecting the session, presentation, or application layers. In effect, the transport layer shields the customer's portion of the network (layers 5–7) from the carrier's portion (layers 1–3).

The session layer is responsible for setting up, managing, and tearing down process-to-process connections, using the host-to-host service provided by the transport layer. It also handles certain aspects of synchronization and recovery.

The presentation layer performs generally useful transformations on the data to be sent, such as text compression. It also

performs the conversions required to allow an interactive program to converse with any one of a set of incompatible intelligent terminals.

The content of the application layer is up to the users. Nevertheless, standard protocols for specific industries, such as airlines and banking, are likely to develop, although few exist now. For this reason we say no more about the application layer in this paper.

Although the ISO OSI Reference Model says nothing about how the layers are to be implemented, one possible configuration might have the physical layer in hardware, the data link layer in a special protocol chip, the network layer in a device driver, the transport and session layers in the operating system proper, the presentation layer in a set of library routines in the user's address space, and the application layer be the user's program.

At this point we have covered enough background material to say a little bit about the ISO OSI Reference Model itself. Basically, it is a framework for describing layered networks. It discusses the concept of layering in considerable detail, and introduces a uniform terminology for naming the various entities involved. Finally, it specifies the seven layers mentioned thus far, and for each layer gives its purpose, the services provided to the next higher layer, and a description of the functions that the layer must perform. The value of the model is that it provides a uniform nomenclature and a generally agreed upon way to split the various network activities into layers.

However, the ISO OSI Reference Model is *not* a protocol standard. By breaking a network's functions up into layers, it suggests places where protocol standards could be developed (physical layer protocols, data link layer protocols, and so on), but these standards themselves fall outside the domain of the model. With the model in hand, other organizations such as the Consultative Committee for International Telephony and Telegraphy (CCITT), the International Federation for Information Processing (IFIP), and the American National Standards Institute (ANSI) may develop specific protocol standards for the various layers. Although these standards may eventually be officially approved by ISO, such work is still in progress and, in any event, falls far outside the scope of the model.

As a final note, before plunging into the details of the various layers, we would like to point out that this article is about network protocols, with the ISO OSI Reference Model used as a guide; it is *not* an article about the model itself. We emphasize the communication algorithms and protocols themselves, a subject about which the Reference Model says nothing.

1. THE PHYSICAL LAYER

In this section we look at a variety of aspects related to the physical layer. Our emphasis is on the conceptual organization of the physical transmission facilities, not on the hardware details themselves. Point-to-point, satellite, and local networks are discussed. We conclude with a brief discussion of the X.21 physical layer protocol.

The function of the physical layer is to allow a host to send a raw bit stream into the network. The physical layer is in no way concerned with the way the bits are grouped into larger units, or what they mean. Nor does it rectify the problem of some bits being garbled by transmission errors. Recovery from such errors is up to the data link layer.

The communication subnet can be organized in one of two ways. In *circuit switching*, a fixed amount of transmission capacity (bandwidth) is reserved when the source initiates a conversation and released only when the conversation is over. The telephone system uses circuit switching. When someone calls a time-sharing service in a distant city, the connection is established after dialing and remains in force until one end hangs up. If the user goes out to lunch while still logged in, the connection remains intact and the charges continue to accumulate, even though the connection is actually idle.

With *packet switching*, in contrast, the user initially sets up a connection between his terminal or host and the nearest IMP, not the destination host. (We assume that the destination host also is connected to some IMP.) Whenever the user has data to send, he sends them to the IMP as a series of *packets*, typically 10–1000 bytes long.

Packets are routed from IMP to IMP within the subnet, until they get to the IMP which services the destination host. No circuits are reserved in advance within the subnet for the terminal-to-host connection (except the terminal-to-IMP and IMP-to-host circuits). Instead, the high-bandwidth IMP-IMP lines are dynamically shared among all the users on a demand basis; IMP-IMP bandwidth is only tied up when data are actually being transmitted.

Although the above discussion is cast in terms of a point-to-point network, the same considerations apply to broadcast channels. If a portion of the channel (e.g., one frequency band) is dedicated to a given conversation throughout its duration, without regard to actual usage, the network is circuit switched. If, however, the channel is dynamically requested, used, and released for every packet, the network is packet switched.

Circuit-switched networks are best suited to communication whose bandwidth requirements do not change much over time. Transmission of human speech is such an application, so it makes sense for the telephone network to be circuit switched. Terminal-to-computer and computer-to-computer traffic, however, is usually bursty. Most of the time there are no data to send, but once in a while a burst of data must be transmitted. For this reason, most computer networks use packet switching to avoid tying up expensive transmission facilities when they are not needed. However, in the future, all digital transmission systems will allow computers to dial a call, send the data, and hang up, all within a few milliseconds. If such systems become widespread, circuit switching may come back into favor.

1.1 The Telephone System

Since most existing long-haul networks use the telephone system for their transmission facilities, we shall briefly describe how the latter is organized. Most telephones are connected to a nearby telephone company switching office by a pair of copper wires known as a *local loop*. The switching offices themselves are connected by high-bandwidth *trunks* onto which thousands of unrelated calls are multiplexed. Although some trunks utilize copper wire, many utilize microwave relays, fiber optics, or wave guides as the transmission medium.

Because the bandwidth of the local loop is artificially limited to about 3000 Hz (hertz), it is difficult to transmit information over it by using, for example, +5 volts for a binary one and 0 volts for a binary zero. Such square wave signaling depends on high-frequency harmonics that are well above the 3000-Hz cutoff frequency. Only with very low date rates might enough information be below 3000 Hz to be intelligible. Instead, a device called a *modem* is inserted between the host and the telephone line. The input to the modem is pure digital data, but the output is a modulated sine wave with a base frequency of generally between 1000 and 2000 Hz. Since the modulated sine wave has fewer high-frequency components than the original square wave, it is affected less by the limited bandwidth.

A sine wave has three properties that can be modulated to transmit information: an amplitude, a frequency, and a phase. In amplitude modulation, two different amplitude values are used to represent 0 and 1. In frequency modulation, different frequencies are used for 0 and 1, but the amplitude is never varied. In phase modulation, neither the amplitude nor the frequency is varied, but the phase of the sine wave is abruptly switched to send data. In the most common encoding scheme, phase shifts of 45, 135, 225, and 315 degrees are used to send 00, 01, 10, and 11, respectively. In other words, each phase shift sends two bits. The three methods can be combined to increase the transmission capacity.

Many such transmission systems have been standardized and form an important class of physical layer protocols. Unfortunately, in many cases, the standards in the United States and Canada differ from those used by the rest of the world. For example, those ubiquitous 300-bit-per-second frequency modulation modems found near terminals around the world use different signaling frequencies in North America and Europe.

Probably the best known physical layer standard at present is RS-232-C, which specifies the meaning of each of the 25 pins

on a terminal connector and the protocol governing their use. However, a new standard, RS-449, has been developed to replace this aging workhorse. RS-449 is upward compatible with RS-232-C but uses a 37-pin connector to accommodate the new signals. Unfortunately, 37 pins are insufficient, so users wishing to take advantage of all the features of RS-449 (notably the secondary channel) need a second 9-pin connector as well.

The transmission technology and protocols used on the interoffice trunks are different from those used on the local loop. In particular, digital rather than analog techniques are becoming increasingly widespread. The most common digital system is *pulse code modulation* (PCM), in which the analog signal coming in from the local loop is digitized by sampling it 8000 times per second. Eight bits (seven data and one control) are transmitted during each 125-μs (microsecond) sampling period. In North America, 24 such PCM channels are grouped together into 193-bit frames, with the last bit being used for synchronization. With 8000 193-bit frames per second, the gross data rate of this system, known as T1, is 1.544 Mbits/s (megabits per second). In Europe, the 1.544-Mbit/s PCM standard uses all 8 bits for data, with the 193rd bit (which is attached to the front rather than rear of the frame) used for signaling. Two different (and incompatible) 32-channel PCM standards running at 2.048 Mbits/s are also widely used outside North America. For more information about the telephone system see DAVI73 and DOLL78.

1.2 Communication Satellites

Although most existing long-haul networks use leased telephone circuits to connect the IMPs, satellite-based networks are becoming increasingly common. A communication satellite is a big repeater in the sky. Incoming signals are amplified and rebroadcast by a transponder on the satellite. The upward and downward signals use different frequencies to avoid interference. A typical communication satellite has 5–10 independent transponders, each with a capacity of about 50 Mbits/s.

Communication satellites are put into geosynchronous equatorial orbit at an altitude of 36,000 kilometers to make them appear stationary in the sky when viewed from the earth. Consequently, the ground station antenna can be pointed at the satellite when the antenna is installed and never moved. A moving satellite would require a much more expensive steerable antenna and would also have the disadvantage of being on the other side of the earth half the time. On the other hand, the great altitude required to achieve a 24-hour period implies an up-and-down propagation delay of 270 ms (milliseconds), which seriously affects the data link layer protocols and response time.

To avoid mutual interference, communication satellites using the 4/6-GHz (gigahertz) frequency band must be separated by an angle of 4 degrees as viewed from the earth. Since some orbit slots have been allocated by international agreement to television, military, and other use, the number of equatorial orbit slots available to data communication is limited. (As an aside, the allocation of orbit slots has been a political battleground, with every country, especially those in the Third World, asking for its fair share of slots for the purpose of renting them back to those countries able to launch satellites.) The 12/14-GHz band has also been allocated to data communication. At these frequencies, an orbit spacing of 1 degree is sufficient, providing four times as many slots. Unfortunately, because water is an excellent absorber of these short microwaves, multiple ground stations and elaborate switching are needed in order to avoid rain.

Three modes of operation have been proposed for satellite users. The most direct but most expensive mode is to put a complete ground station with antenna on the user's roof. This approach is already feasible for large multinational corporations and will become feasible for medium-sized ones as costs decline. The second approach is to put a small, cheap antenna on the user's roof to communicate with a shared satellite ground station on a nearby hill. The third approach is to access the ground station via a cable (e.g., a leased telephone circuit or even the same cable used for cable television).

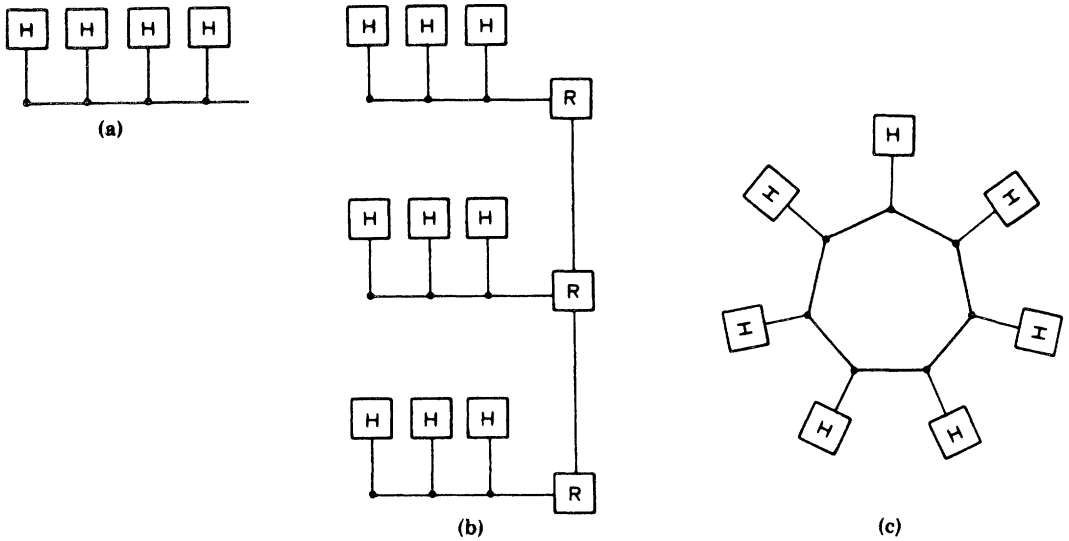


Figure 3. Local network topologies. (a) Linear cable with four hosts. (b) Segmented cable with repeaters and hosts. (c) Ring.

Physical layer satellite protocols typically have many PCM channels multiplexed on each transponder beam. Sometimes they are dedicated (circuit switched); at other times they are dynamically assigned as needed (packet switched). For more information about communication satellites see MART78.

1.3 Local Networks

In most local networks, the hosts are connected by a linear, tree-shaped, or ring-shaped cable, as shown in Figure 3. In Figure 3a, all hosts tap onto a common cable. In Figure 3b, multiple cables are used (e.g., one per floor of an office building), with repeaters connecting the segments. In Figure 3c, all hosts tap onto a unidirectional ring.

A widely imitated linear or tree-shaped local network is the Ethernet™ network [METC76]. The proper term for this kind of network is CSMA/CD (Carrier Sense Multiple Access/Collision Detect), although many people incorrectly use the term "Ethernet" (which is a trademark of the Xerox Corporation) in a generic sense. In these networks, only one packet may be on the cable at any instant. The cable is known as the *ether*, after the luminiferous ether through which electromagnetic radiation was once alleged to propagate. The principle behind CSMA/CD is simple: when a host wishes to send a packet, it first listens to the ether to see if the ether is being used. If it is, the host waits until the current transmission finishes; if not, the host begins transmitting immediately.

The interface hardware must detect collisions caused by two hosts simultaneously starting a transmission. Collision detection is done using analog circuitry, in essence monitoring the ether to see if it agrees with the signal being transmitted. When a host interface (the analog of an IMP in this system, since the ether itself is totally passive) detects a collision, it informs the data link layer. The collision recovery action consists of aborting the current transmission, broadcasting a noise burst to make sure that everyone else detects the collision as well, waiting a random length of time, and then trying again. Collision detection is only feasible if the round-trip propagation delay is short compared to the packet transmission time, a condition that can be met with cable networks, but not, for example, with satellite networks.

Cable networks similar to the Xerox Ethernet network, but without the collision detect feature, also exist. Network designers can trade off the cost of collision detection circuitry against the time lost by not aborting colliding packets quickly.

Ring nets use a different principle: in effect, the whole ring is a giant circular shift

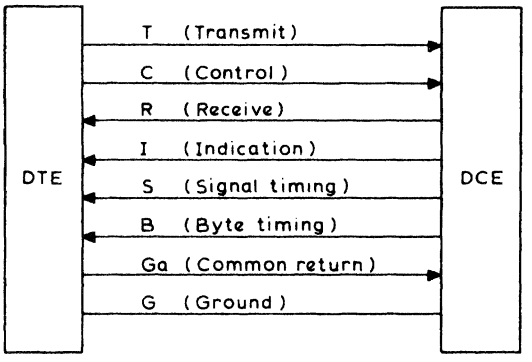


Figure 4. The DTE/DCE interface in X.21.

register. After each shift, the host interface can read or write the bit just shifted into it. Several different kinds of rings have been proposed [CLAR78, FARB72, FRAS75, LIU78, WILK79], differing primarily in their layer 2 organizations, which we describe later. Both CSMA/CD networks and rings typically operate at data rates of 1–10 Mbits/s. A substantial bibliography about local networks can be found in FREE80 and SHOC81.

1.4 An Example Physical Layer Protocol: X.21

At present, most physical layer standards, like RS-232-C and RS-449, utilize analog signaling. In the future, true digital interfaces will be needed. Recognizing this need, CCITT, the international standardization body for telephony, has developed a fully digital interface called *X.21*. X.21 is intended to be used to connect a host computer to a network. This connection remains established as long as the host wants to communicate with the network. Consequently, X.21 is a circuit-switched protocol, but host–host connections set up over the X.21 line may be either circuit switched or packet switched.

In X.21 terminology, the host is a *DTE* (Data Terminal Equipment) and the IMP is a *DCE* (Data Circuit-Terminating Equipment). The DTE–DCE interface consists of eight lines, as shown in Figure 4. The *S* line provides a clock signal to define bit boundaries. The (optional) *B* line provides a pulse every eighth bit, to allow byte alignment. The *C* and *I* lines are used for control signaling, analogous to the on-hook/off-hook signal on a telephone. The *T* and *R* lines are used for data and also for signaling.

To see how X.21 works, let us examine how a DTE calls another DTE, talks to it, and then hangs up. When the interface is idle, *T, R, C,* and *I* are all 1. The series of events is as follows (with a telephone analogy in parentheses):

(1) DTE drops *T* and *C* (DTE picks up phone).
(2) DCE sends "+++++++···+++" on *R* (DCE sends dial tone).
(3) DTE sends callee's address on *T* (DTE dials number).
(4) DCE sends call progress signals on *R* (phone rings).
(5) DCE drops *I* to 0 (callee answers phone).
(6) Full duplex data exchange on *T* and *R* (talk).
(7) DTE raises *C* to 1 (DTE says goodbye).
(8) DCE raises *I* to 1 (DCE says goodbye).
(9) DCE raises *R* to 1 (DCE hangs up).
(10) DTE raises *T* to 1 (DTE hangs up).

The call progress signals in Step 4 tell whether the call has been put through, and if not, why not. The shutdown procedure in Steps 7–10 operates in two phases. After either party has said goodbye, that party may not send more data but it must continue listening for incoming data. When both sides have said goodbye, they then hang up, returning the interface to idle state, with 1's on all four lines. RS-449 and X.21 are described in more detail in BERT80 and FOLT80.

2. THE DATA LINK LAYER

As we have seen, neither X.21, RS-232-C, nor any other physical layer protocol makes any attempt to detect or correct transmission errors. Nor do these protocols recognize the possibility that the receiver cannot accept data as fast as the sender can transmit them. Both of these problems are handled in the data link layer. In the following sections we first discuss the relevant principles and then we give an example of a widely used data link protocol, HDLC (High-Level Data Link Control). Following the HDLC example, we look at some data link protocols for satellite and local networks.

As mentioned earlier, the approach used in the data layer is to partition the raw physical layer bit stream into frames so each transmitted frame can be acknowledged if need be. An obvious question is: "How are frames delimited?" In other words, how can the receiver tell where one frame ends and the next one begins?

Three methods are in common use on long-haul networks: *character count, character stuffing,* and *bit stuffing.* With the first method, each frame begins with a fixed-format frame header that tells how many characters are contained in the frame. Thus, by simply counting characters, the receiver can detect the end of the current frame and the start of the following one. The method has the disadvantage of being overly sensitive to undetected transmission errors which affect the count field; it also has the disadvantage of enforcing a specific character size. Furthermore, lost characters wreak havoc with frame synchronization. Digital Equipment Corporation's DDCMP (Digital Data Communication Message Protocol) uses the character count method, but few other protocols do. Use of character counts to delimit frames is likely to diminish in the future.

The second method for delimiting frames, character stuffing, is to terminate each frame with a special "end-of-frame" character. The problem here is what to do with "end-of-frame" characters that accidently appear in the data (e.g., in the middle of a floating point number). The solution is to insert an "escape" character before every accidental "end-of-frame" character. Now what about accidental "escape" characters? These are rendered as two consecutive escapes. Although these conventions eliminate all ambiguity, they do so at the price of building a specific character code into the protocol. IBM's BISYNC (BInary SYNchronous Communication) protocol uses character stuffing, but, like all other such protocols, it is gradually becoming obsolete.

Modern data link protocols for long-haul networks all use bit stuffing, a technique in which frames are delimited by the bit pattern 01111110. Whenever five consecutive one bits appear in the data stream, a zero bit is "stuffed" into the bit stream (normally by hardware). Doing so prevents user data from interfering with framing, but does not impose any character size on the data.

On local networks, one can use any of the above methods, or a fourth method: detecting frames by the presence or absence of a signal on the cable. This method is much more direct, but it is not applicable to long-haul networks.

Virtually all data link protocols include a checksum in the frame header or trailer to detect, but not correct, errors. This approach has traditionally been used because error detection and retransmission requires fewer bits on the average than forward error correction (e.g., with a Hamming code). However, with the growing use of satellites, the long propagation delay makes forward error correction increasingly attractive.

A simple checksum algorithm is: compute the Exclusive OR of all the bytes or words as they are transmitted. This algorithm will detect all frames containing an odd number of bits in error, or a single error burst of length less than the checksum, and many other combinations. In practice, a more complex algorithm based on modulo 2 polynomial arithmetic is used [PETE61, SLOA75].

2.1 Stop-and-Wait Protocols

As a first example of a data link layer protocol, consider a host *A* wishing to send data to another host *B* over a perfectly reliable channel. At first glance you might think that *A* could just send at will. However, this idea does not work, since *B* may not be able to process the data as fast as they come in. If *B* had an infinite amount of buffer space, it could store the input for subsequent processing. Unfortunately, no host has infinite storage. Consequently, a mechanism is needed to throttle *A* into sending no faster than *B* can process the data. Such mechanisms are called *flow control* algorithms. The simplest one calls for *A* to send a frame and then wait for *B* to send explicit permission to send the next frame. This algorithm, called *stop-and-wait,* is widely used.

More elaborate protocols are needed for actual channels that make errors. An obvious extension to our basic protocol is to

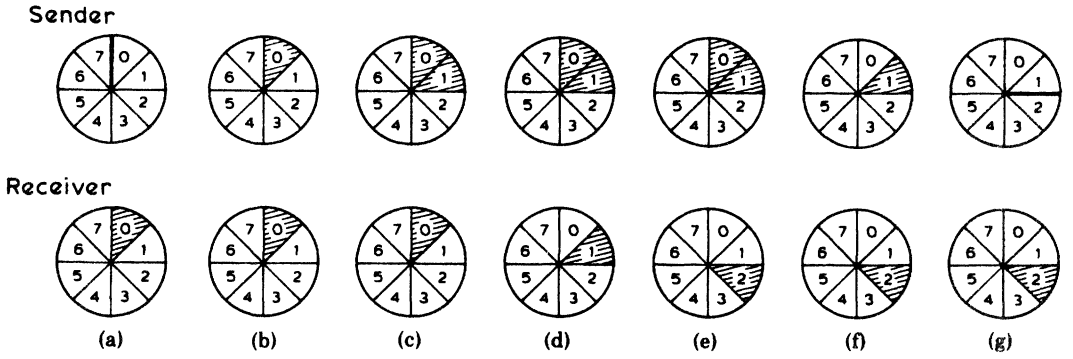


Figure 5. The sliding-window algorithm.

have A put a sequence number in each data link frame header and to have B put in each acknowledgment frame both a sequence number and a bit telling whether the checksum was correct or not. Whenever A received a negative acknowledgment frame (i.e., one announcing a checksum error), it could just repeat the frame.

Unfortunately, this protocol fails if either data or acknowledgment frames can be lost entirely in noise bursts. If a frame is lost, A will wait forever, creating a deadlock. Consequently, A must time out and repeat a frame if no acknowledgment is forthcoming within a reasonable period. Since each frame bears a sequence number, no harm is done if A has an itchy trigger finger and retransmits too quickly; however, some bandwidth is lost.

2.2 Sliding-Window Protocols

Stop-and-wait works well if the propagation time between the hosts is negligible. Consider, for a moment, how stop-and-wait works when 1000-bit frames are sent over a 1-Mbit/s satellite channel:

Time (ms)	Event
0	A starts sending the frame
1	Last bit sent; A starts to wait
270	First bit arrives at B
271	Last bit arrives at B
271	B sends a short acknowledgment
541	The acknowledgment arrives at A

For each millisecond of transmission, A has to wait 540 ms. The channel utilization is thus 1/541, or well below 1 percent. A better protocol is needed.

One such protocol is the *sliding-window* protocol, in which the sender is allowed to have multiple unacknowledged frames outstanding simultaneously. In this protocol, the sender has two variables, S_L and S_U, that tell which frames have been sent but not yet acknowledged. S_L is the lowest numbered frame sent but not yet acknowledged. The upper limit, S_U, is the first frame not yet sent. The current send window size is defined as $S_U - S_L$.

The receiver also has two variables, R_L and R_U, indicating that a frame with sequence number N may be accepted, provided that $R_L \leq N < R_U$. If $R_U - R_L = 1$, then the receiver has a window of size 1, that is, it only accepts frames in sequence. If the receiver's window is larger than 1, the receiver's data link layer may accept frames out of order, but normally it will just buffer such frames internally, so that it can pass frames to the network layer in order.

To keep sequence numbers from growing without bound, arithmetic is done modulo some power of 2. In the example of Figure 5, sequence numbers are recorded modulo 8. Initially (Figure 5a), $S_L = 0$, $S_U = 0$, $R_L = 0$, and $R_U = 1$ (receiver window size is 1 in this example). The current window is shown shaded in the figure. When the data link layer on the sending machine receives a frame to send (from the network layer), it sends the frame and advances the upper edge of its window by 1, as shown in Figure 5b. When it receives the next frame from the network layer, it sends the frame and advances the window again (Figure 5c). When the first frame arrives at the receiver, the receiver's window is rotated by advanc-

ing both edges (Figure 5d), and an acknowledgment is sent back. If frame 1 arrives at the receiver before the acknowledgment gets back to the sender, the state will be as shown in Figure 5e. When the first acknowledgment arrives, the lower edge of the sender's window is advanced (Figure 5f). Figure 5g shows the variables after both acknowledgments arrive.

As with stop-and-wait, the sliding-window protocol uses timeouts to recover from lost frames. The sender maintains a timer for each frame currently in its window. Whenever the lower edge of the window is advanced, the corresponding timer is stopped. Suppose, for example, that frames 0–4 are transmitted, but frame 1 is lost. The receiver will acknowledge frame 0, but discard frames 2–4 as they arrive, because they are outside the receive window (still size 1 in our example). Eventually, frames 1–4 will all time out and be retransmitted.

How many frames may our example sender have outstanding at any instant? The answer is seven, not eight, as might at first appear. To see why, consider the following scenario:

(1) The sender transmits frames 0–7.
(2) All eight frames arrive and are acknowledged.
(3) All eight acknowledgments are lost.
(4) The sender times out and retransmits the eight frames.
(5) The receiver unknowingly accepts the duplicates.

The problem occurs after Step 2, when the receiver's window has rotated all the way around and it is prepared to accept frame 0 again. Unfortunately, it cannot distinguish frame 9 from frame 0, so the stream of frames passed to the network layer will contain undetected duplicate frames.

The solution is to restrict the sender's window to seven outstanding frames. Then, after Step 2 above, the receiver will be expecting frame 7, and will reject all the duplicate frames, informing the sender after each rejection that it expects frame 7 next.

In the above example, whenever a frame is lost, the receiver is obligated to discard subsequent frames, even though they are correctly received. To avoid this inefficiency, we can allow the receiver's window to be greater than 1. Now let us look at the lost frame problem again, with both the sender's and receiver's windows of size 7. When frames 2–4 come in, the receiver keeps them internally. Eventually frame 1 times out and is retransmitted. The receiver replies to the correct receipt of frame 1 by saying that it expects frame 5 next, thereby implicitly acknowledging frames 2–4 and preventing their retransmission. With frames 1–4 now safely in hand, the data link layer can pass them to the network layer in sequence, thus completely shielding the latter from the lost frame and its recovery. This strategy is often called *selective repeat,* as opposed to the *go back n* strategy implied by a receiver window size of 1.

Unfortunately, even with the window settings used above, the protocol can fail. Consider the following scenario:

(1) The sender transmits frames 0–6.
(2) All frames arrive; the receiver's window is now 7, 0, 1, 2, 3, 4, 5.
(3) All seven acknowledgments are lost.
(4) The sender times out and retransmits frames 0–6.
(5) The receiver buffers frames 0–5 and says it wants frame 7 next.
(6) The sender transmits frames 7–13 (sequence numbers 7, 0, 1, 2, 3, 4, 5).
(7) The receiver accepts frame 7 but rejects frames 0–5 as duplicates.

At this point the receiver has frames 7, 0, 1, 2, 3, 4, and 5 buffered. It passes them all to the unsuspecting network layer. Consequently, undetected duplicates sneak through again. To prevent this, the window size must be restricted to not more than half the size of the sequence number space. With such a restriction, the receiver's window after having received a maximum batch of frames will not overlap what it was before having received the frames. Hence no ambiguity arises about whether a frame is a retransmission or an original.

2.3 An Example Data Link Protocol: HDLC

As an example of a data link protocol that is widely used, we now briefly look at HDLC (High-Level Data Link Control). HDLC has many brothers and sisters (e.g., SDLC, ADCCP, LAP, LAPB), each having

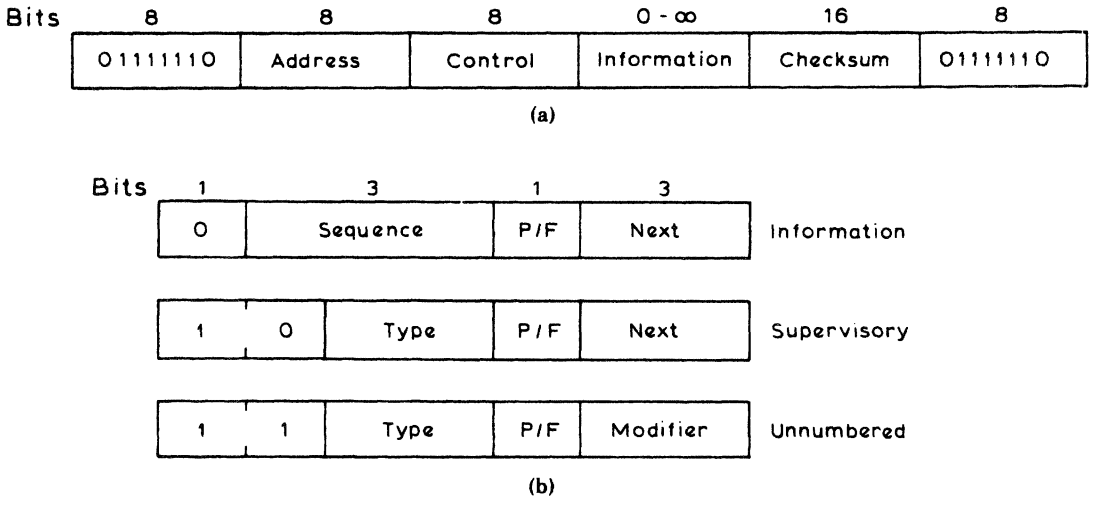


Figure 6. (a) The HDLC frame format. (b) The control byte for the three kinds of frames.

minor, but irritating, differences in the control frames. How this situation came about has to do with how certain large bureaucracies view certain other large bureaucracies, a stone best left unturned here.

HDLC and its friends all use bit stuffing for delimiting frames. Their format is shown in Figure 6a. The *Address* field is used for addressing on multipoint lines (lines connecting more than two computers). The *Control* field is different for each of the three classes of frames (see Figure 6b). In *Information* frames (i.e., ordinary data), the *Sequence* and *Next* fields contain the sequence number of the current frame and of the next frame expected, respectively. When *A* sends a frame to *B*, the *Sequence* field is the number of the frame being sent and the *Next* frame is an acknowledgment to *B* saying that *A* has correctly received all frames sent by *B* up to but not including *Next*. Attaching an acknowledgment field to an outgoing data frame is widely known as *piggybacking*. The practice saves bandwidth by requiring fewer frames. Reducing the number of frames *sent* also reduces the number of frames *received*, and hence reduces the number of I/O interrupts on the receiving machine.

When no reverse traffic is present on which to piggyback acknowledgments, a *Type* = 0 supervisory frame is used. The other types of supervisory frames are for negative acknowledgment, selective repeat, and receiver temporarily not ready. The *P/F* bit stands for *Poll/Final* and has miscellaneous uses, such as indicating polling frames on multipoint lines and the final frame in a sequence.

Unnumbered frames consist of a hodgepodge of control information and comprise the area of greatest difference between the various HDLC-like protocols. Most of these frames are used to initialize the line and to report certain abnormal conditions.

Although Figure 6 depicts HDLC as having a 3-bit sequence number, an alternate format with 7-bit sequence numbers also exists, for use on satellite or other channels where large windows are needed to keep the channel busy. Gelenbe et al. [GELE78] have constructed a mathematical model of HDLC that can be used to calculate the throughput as a function of window size.

2.4 Channel Allocation in Satellite Networks

At this point we switch from the data link layer of point-to-point networks to that of broadcast networks, in particular, satellite and local networks. Broadcast networks are characterized by having a single channel that is dynamically requested and released by hosts for every packet sent. A protocol is needed for determining who may use the

channel when, how to prevent channel overload, and so on. These problems do not occur in point-to-point networks. On the other hand, since every host receives every packet, broadcast networks usually do not have to make any routing decisions. Thus the main function of the network layer is not relevant.

As a consequence of these fundamental differences, it is not really clear where the channel-access protocol should be placed in the ISO OSI Reference Model, which does not mention the issue at all. It could be put in the data link layer, since it deals with getting packets from one machine to the next, but it could equally well be put in the network layer, since it also concerns getting packets from the source host to the destination host. Another argument for putting it in the network layer is that the main task of the access protocol is to avoid congestion on the channel, and congestion control is specifically a network layer function. Last, in most broadcast networks the transport layer is built directly on top of this protocol, or in some cases on top of an internetwork protocol, something lacking in the ISO OSI Reference Model. Nevertheless, we treat the subject as part of the data link layer because the IEEE local network standards committee (802) is probably going to put it there. By analogy, the contention resolution protocol for satellite channels also belongs in the data link layer.

A satellite link can be operated like a terrestrial point-to-point link, providing dedicated bandwidth for each user by time-division or frequency-division multiplexing. In this mode the data link protocols are the same as in point-to-point networks, albeit with longer timeouts to account for the longer propagation delay.

Another mode of operation, however, is to dynamically assign the channel among the numerous competing users. Since their only method of communication is via the channel itself, the protocol used for allocating the channel is nontrivial. Abramson [ABRA70] and Roberts [ROBE73] have devised a method, known as *slotted ALOHA*, that has some interesting properties. In their approach, time is slotted into units of a (fixed-length) packet. During each interval, a host having a packet to send can either send or refrain from sending. If no hosts use a given slot, the slot is just wasted. If one host uses a slot, a successful transmission occurs. If two or more hosts try to use the same slot, a collision occurs and the slot is also wasted. Note that with satellites the hosts do not discover the collision until 270 ms after they start sending the packets. Owing to this long delay, the collision detection principle from CSMA/CD is not applicable here. Instead, after detecting a collision, each host waits a random number of slots and tries again.

Clearly, if few hosts have packets to send, few collisions will occur and the success rate will be high. If, on the other hand, many hosts have packets to send, many collisions will occur and the success rate will be low. In both cases the throughput will be low: in the first case because of lack of offered traffic, in the second case because of collisions. Hence the throughput versus offered traffic curve starts out low, peaks, and then falls again. Abramson [ABRA73] showed that the peak occurs when the mean offered traffic is one packet per slot, which yields a throughput of $1/e$ or about 0.37 packets per slot. Hence the best one can hope for with slotted ALOHA is a 37 percent channel utilization.

Slotted ALOHA has another problem, in addition to the low throughput: stability. Suppose that an ALOHA system has many hosts. By accident, during one slot k hosts transmit and collide. After detecting the collision, each host decides to retransmit during the next slot with probability p (a parameter of the system). In other words, each host picks a random number between 0 and 1. If the number is less than p, it transmits; otherwise it waits until the next slot to pick another random number.

If $kp \gg 1$, many hosts will retransmit during each succeeding slot and practically nothing will get through. Worse yet, these retransmissions will compete with new packets from other hosts, increasing the number of hosts trying to use the channel, which just makes the problem worse. Pretty soon all hosts will be trying to send and the throughput will approach zero, collapsing the system permanently.

The trick to avoid collapse is to set the parameter p low enough that $kp < 1$ for the

k values expected. However, the lower p is, the longer it takes even to attempt retransmission, let alone succeed. Hence a low value of p leads to a stable system, but only at the price of long delay times.

One way to set p is to use a default value on the first retransmission, say 0.5, on the assumption that two hosts are involved in the collision. On each subsequent collision, halve p. Gerla and Kleinrock [GERL77] have another proposal; they suggest that each host monitor the channel all the time, just to measure the collision rate. When the collision rate is low, the hosts can set p high; when the collision rate is high, the hosts can set p low to minimize collisions and get rid of the backlog, albeit slowly.

A completely different way to avoid collisions is to attempt to schedule the slots in advance rather than have continuous competition for them. Crowther et al. [CROW73] proposed grouping slots into n-slot time slices, with the time slice longer than the propagation delay. In their system, contention is used initially, as described above. Once a host has captured (i.e., successfully used) a slot, it is entitled to use the same slot position in the next slice, forbidding all other hosts from trying to use it. This algorithm makes it possible for a host to transmit a long file without too much pain. If a host no longer needs a slot position, it sets a bit in the packet header that permits other users to contend for the slot the next time around.

Roberts [ROBE73] also proposed a method of reducing contention. His proposal also groups slots into time slices. One slot per slice is divided into minislots and used for reserving regular slots. To send a packet, a host must first compete for a minislot. Since all hosts see the results of the minislot contention, they can all keep track of how long the queue is and hence know who gets to send when. In effect, the use of minislots greatly reduces the amount of time wasted on a collision (like the CSMA/CD rule about aborting collisions as soon as they are detected).

2.5 Channel Allocation in Local Networks

As mentioned earlier, when a CSMA/CD host detects a collision, it jams the channel, aborts the current packet, waits a random time, and tries again. How long should it wait? Metcalfe and Boggs [METC76] decided to use a default maximum time interval on the first collision, with the actual waiting time being picked by multiplying a random number in the range 0.0–1.0 by the maximum time interval. On each successive collision the maximum time interval is doubled and a new random number is generated. They called their algorithm *binary exponential backoff*.

Various other algorithms have been proposed for CSMA/CD, including some that prevent all collisions. For example, Chlamtac [CHLA76], Chlamtac et al. [CHLA79], and Scholl [SCHO76] have suggested slotting time into intervals equal to the channel acquisition time (the round-trip propagation delay). After a successful transmission by host n, the next bit slot is then reserved for host $(n + 1)$ (modulo the number of hosts). If the indicated host does not claim its right to use the channel, the next host gets a chance during the succeeding bit slot, and so on. In effect, a virtual baton is passed from host to host, with hosts only allowed to transmit when holding the baton.

Rothauser and Wild [ROTH77] have also proposed a collision-free CSMA/CD protocol. To illustrate their suggestion, we shall assume that there are 1024 hosts, numbered from 0 to 1023 (in binary, although other radices can also be used). After a successful transmission, ten bit slots will be used to determine who goes next. Each host attempts to broadcast its 10-bit number in the ten slots, subject to the rule that as soon as a host realizes that a higher numbered host wants the channel, it must stop trying. If, for example, the first three bits are 011, then some host in the range 368–511 wants the channel, and so hosts below 368 must desist from competing on the current round. No host above 511 wants the channel, as evidenced by the leading 0 bit. In effect, the channel is allocated to the highest numbered contender. Since this system gives high-numbered stations an advantage, it is desirable to make the host numbers virtual, rotating them one position after each successful transmission.

Protocols that allow only a limited number of collisions have also been proposed [CAPE79a, CAPE79b, KLEI78]. Capetanakis'

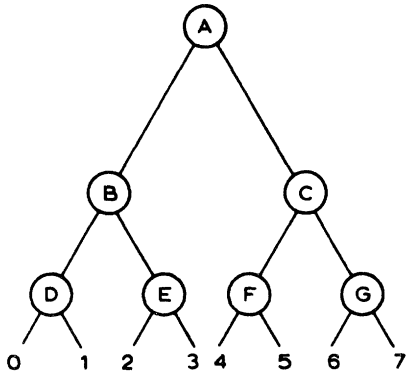


Figure 7. Eight machines organized in Capetanakis' (virtual) tree.

idea is illustrated in Figure 7 for a system with eight hosts. Initially, all hosts may compete. If a collision occurs, only those hosts under node *B* of the tree, namely, 0, 1, 2, and 3, may compete. If another collision occurs, the only descendants of node *D* may try, and so forth. As an example, suppose hosts 2, 3, and 4 all want the channel. After initial collisions for *A* (2, 3, and 4) and *B* (2 and 3), it will be node *D*'s turn and the channel will lie idle. Next comes node *E* and another collision. Then 2 and 3 each get a private slot, followed by *C*. At low load, the algorithm allows pure contention, but under high load it walks the tree looking for hosts that want to send.

Although more could be said about CSMA/CD protocols, we now turn our attention to ring networks. In one of the best-known rings [FARB72], an 8-bit token circulates around the ring when there is no traffic. When a host wants to transmit, it must first capture and destroy the token. Having done so, it may send its packet. When it is finished, it must put the token back, giving the next host downstream a chance to seize it.

If the token is ever lost (e.g., as a result of a ring interface malfunctioning), some mechanism is needed to regenerate it. One possibility is that each host wishing to send must monitor the ring. Having failed to see a token within the worst case interval—namely, all other hosts sending a maximum-length packet—the host generates a new token itself. However, with a little bad luck, two hosts might generate tokens simultaneously. Hence, it appears that token recovery in a ring net is similar to contention in CSMA/CD in systems. Clark et al. [CLAR78] have taken this observation to its logical conclusion and proposed a contention ring that is a hybrid of the token ring and CSMA/CD.

Yet another type of ring is exemplified by the Cambridge Ring [NEED79, WILK79]. This 10-Mbit/s ring contains several small slots around it, each slot consisting of 16 bits of data, an 8-bit source address, an 8-bit destination address, a bit telling whether the slot is full or empty, and a few other control bits. To transmit, a host interface just waits for a free slot and fills it up. When the slot arrives at the destination, the receiving interface sets the control bits telling whether it was accepted or not. About 10 μs after transmission, the slot comes back around again so that the sender can find out what happened to it. The sender is not permitted to reuse the slot immediately, as an antihogging measure. By having such small slots and preventing their immediate reuse, the ring guarantees an extremely short delay for small packets, but at the price of higher overhead than the token ring under heavy load.

Still another design is discussed in LIU78. In Liu's design, each ring interface has a shift register equal in length to the maximum packet size. When a host wants to send a packet, it loads up the shift register and inserts the shift register into the ring between two packets. This mechanism leads to low delay, since a host need only wait until the current packet has passed through. When the shift register becomes empty (through a period of low traffic), it can be removed from the ring.

From the above discussion, it should be obvious that many local network protocols have already been devised, with more being threatened all the time. Without standards, the most likely development would be a proliferation of local networks from various vendors, all incompatible; vendor A's terminal would not talk to vendor B's CPU because they would have different protocols embedded in their hardware. In an attempt to nip this incipient chaos in the bud, the Institute of Electrical and Electronics Engineers (IEEE) set up a commit-

tee in February 1980 to develop a standard for local network protocols. Although the standard, IEEE 802, was not completed at the time of this writing, the general picture looks as if it will probably be as follows:

The standard treats the physical and data link layers. The physical layer allows for base-band, broad-band, and fiber optics communication, and describes the interfacing of the host (DTE) to the cable. The data link layer handles channel access, as mentioned earlier, as well as addressing, frame format, and control.

The data link layer is split up into two sublayers, media access and data link control, with a third optional sublayer for internetworking (whose presence in the data link layer instead of in the network layer is certainly arguable). The media access sublayer handles channel allocation. It is here that a choice had to be made between CSMA/CD and some kind of ring. The arguments for CSMA/CD were that it was fair, easy to implement on a single chip, had six years of operational experience, and had three major companies (DEC, Intel, and Xerox) already publicly committed to it.

The token ring supporters' counterarguments were as follows: rings, unlike CSMA/CD, provide a guaranteed worst case access time (needed for real-time work, such as speech transmission); rings can be logical as well as physical, accommodating various topologies and allowing important hosts better access by inserting them into the logical ring in several places; and ring performance does not degrade at high load, as does CSMA/CD owing to the many collisions. Unfortunately, neither camp had the necessary two-thirds majority required by IEEE rules, and so a compromise was made in which both CSMA/CD and a token ring were included.

The data link control sublayer was designed to be as compatible with HDLC as possible, on the theory that the last thing the world needed was yet another brand-new data link protocol. Two types of service are provided for: connection oriented and pure datagram. In the former, the data link layer times out and retransmits lost frames, guarantees arrival in sequence, and regulates flow using the standard HDLC sliding-window protocol. In the latter, the data link layer guarantees nothing; once sent, the frame is forgotten (at least by the data link layer).

The major difference between the 802 frame and HDLC's is the presence in 802 of two addresses, source and destination, instead of the one address in HDLC, and the use of variable-length addresses (from 1 to 7 bytes), instead of fixed-length, 1-byte addresses. HDLC was designed for two-party, point-to-point lines, where no address is needed, and for multipoint master/slave lines, in which only the slave's address is needed. In contrast, 802 is aimed at multipoint symmetric lines, where any machine can send to any other machine, and so two addresses are required. The decision to have variable-length addresses up to 7 bytes is intended to allow processes to be designated by a worldwide unique address. Three of the 7 bytes are to be administered by an as-yet-unidentified international organization, and 4 are for local use. Most networks will only need 1- or 2-byte addresses for internal traffic.

3. THE NETWORK LAYER

When a frame arrives at an IMP in a point-to-point network, the data link layer strips off the data link header and trailer and passes what is left, called a *packet*, to the network layer. The network layer must then decide which outgoing line to forward the packet on. It would be nice if such decisions could be made so as to avoid having some lines congested and others idle. Hence congestion control is intimately related to routing. We first look at routing and then at congestion control, both for point-to-point networks. With the channel acquisition protocol for broadcast networks in the data link layer, the network layer for these networks is essentially empty.

Two opposing philosophies exist concerning the network layer. In most local networks and some long-haul networks, the network layer provides a service for delivering independent packets from source to destination with a high probability of success (although less than 1.0). Each packet carried is unrelated to any other packet, past or future, and must therefore carry a full destination address. Such packets are called *datagrams*.

The other approach, taken in many public data networks (especially in Europe), is to require a transmitter to first send a setup packet. The setup packet chooses a route for subsequent traffic and initializes the IMPs along the route accordingly. The user chooses, or is given, a *virtual circuit number* to use for subsequent packets going to the same destination. In this organization, data packets belonging to a single conversation are not independent, since they all follow the same route, determined by the virtual circuit number in them.

The advantage of using virtual circuits is that it guarantees that packets will be delivered in order and helps reduce congestion by making it possible to reserve resources (e.g., buffers) along the route in advance. The disadvantage is that a lot of IMP table space is taken up by idle connections and that there is a lot of overhead in setting up and closing down circuits, the latter a great concern in transaction-oriented database systems [MANN78]. With a datagram system, a query-response requires just two packets. With a virtual circuit system it requires six packets: setup, acknowledgment, query, response, close circuit, and acknowledgment.

3.1 Routing in Point-to-Point Networks

Many routing algorithms have been proposed, for example, BARA64, FRAT73, MCQU74, RUDI76, SCHW80, and SEGA81. Below we sketch a few of the more interesting ones. The simplest algorithm is *static* or *directory* routing, in which each IMP has a table indexed by destination, telling which outgoing line to use. When a packet comes in, the destination address is extracted from the network layer header and used as an index into the routing table. The packet is then passed back down to the data link layer (see Figure 2) along with the chosen line number.

A variant algorithm provides two or more outgoing lines for each destination, each with a weight. When a packet arrives, a line is chosen with a probability proportional to its weight. Allowing alternatives eases congestion by spreading the traffic around. Note that when virtual circuits are used *within the subnet*, the routing decision is only made for setup packets, not data packets.

Several proposals have been made for determining the routes to be put in the tables. Shortest path routing, which minimizes the number of hops (IMP-IMP lines), is an obvious candidate. In FRAT73 another method, based on flow deviation, is given.

The problem with static routing is just that—it is static—it does not adapt to changing traffic patterns and does not try to route packets around congested areas. One way to have the network adapt is to have one host function as a routing control center. All IMPs send it periodic reports on their queue lengths and line utilizations, from which it computes the best routes and distributes the new routing tables back to the IMPs.

Although seemingly attractive, centralized routing has more than its share of problems [MCQU74]. To start with, if the routing control center malfunctions, the network will probably be in big trouble. Second, the complete optimal routing calculation for a large network may require a large dedicated host and even then may not be able to keep up with the traffic fluctuations. Third, since IMPs near the routing control center get their new tables before more distant IMPs do, the network will operate with mixed old-new tables occasionally, a situation that may cause traffic (including the new routing tables) to loop. Fourth, if the network is large, the traffic flow into and out of the routing control center may itself get to be a problem.

One of the earliest routing algorithms [BARA64] adapts to changing traffic, but does so without any central control. In *hot-potato routing*, when a packet arrives, it is assigned to the output line which has the shortest transmission queue. This strategy gets rid of the packet as fast as possible, without regard to where it is going. A much better idea is to combine static information about the suitability of a given output line with the queue lengths. This variant is known as *shortest queue plus bias*. It could be parameterized, for example, to use the shortest queue unless the line is going the wrong way, or to use the statically best line unless its queue exceeds some threshold.

Algorithms like this are known as *isolated adaptive* algorithms [McQu74].

Rudin's *delta routing* [Rudi76] combines some features from centralized and isolated adaptive algorithms. In this method, IMPs send periodic status reports to the routing control center, which then computes the k best paths from each source to each destination. It considers the top few paths equivalent if they differ (in length, estimated transit time, etc.) by an amount less than some parameter δ. Each IMP is given the list of equivalent paths for each destination, from which it may make a choice based on local factors such as queue lengths. If δ is small, only the best path is given to the IMPs, resulting in centralized routing. If δ is large, all paths are considered equivalent, producing isolated adaptive routing. Intermediate strategies are obviously also possible.

A completely different approach is distributed adaptive routing [McQu74], first used in the ARPANET, but replaced after ten years owing to the problems with looping discussed below. With this algorithm, each IMP maintains a table indexed by destination giving the estimated time to get to each destination and also which line to use. The IMP also maintains an estimate of how long a newly arrived packet would take to reach each neighbor, which depends on the queue length for the line to that neighbor.

Periodically, each IMP sends its routing table to each neighbor. When a routing table comes in, the IMP performs the following calculation for each destination. If the time to get the neighbor plus the neighbor's estimate of the time to get to the destination is less than the IMP's current estimate to that destination, packets to that destination should henceforth be routed to the neighbor.

As a simple example, consider a five-IMP network. At a certain instant, IMP 2 has estimates to all possible destinations, as shown in Figure 8a. Suddenly the routing table from IMP 3 (assumed to be adjacent to IMP 2) arrives, as shown in Figure 8b. Let us assume that IMP 2 estimates the delay to IMP 3 to be 10 ms, on the basis of the size of its transmission queue for IMP 3. IMP 2 now calculates that the transit

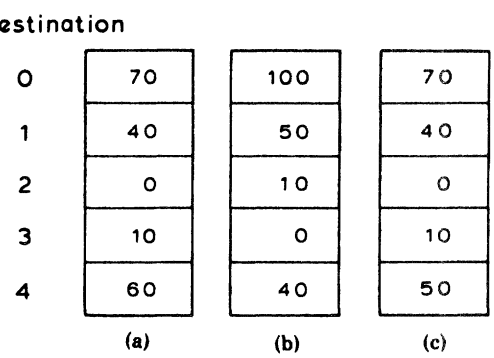


Figure 8. Distributed adaptive routing. (a) An IMP's original routing table. (b) Routing table arriving from a neighboring IMP. (c) The new routing table, assuming a 10-ms delay to the neighbor.

time to IMP 0 via IMP 3 is 10 + 100 ms. Since this time is worse than the 70 ms for its current route, no change is made to entry 0 of the table. Similarly $10 + 50 > 40$, and so no change is made for destination 1 either. However, for destination 4, IMP 3 offers a 40-ms delay, which, when combined with the 10-ms delay to get to IMP 3, is still better than the current route ($10 + 40 < 60$). Therefore, IMP 2 changes its estimate of the time required to get the IMP 4 to 50 ms, and records the line to IMP 3 as the way to get there. The new routing table is given in Figure 8c.

Although this method seems simple and elegant, it has a problem. Suppose that A, B, and C are connected by lines AB and BC. If number of hops is used as a metric, B thinks it is one hop from A, and C thinks it is two hops from A. Now imagine that line AB goes down. B detects the dead line directly and realizes that its delay to A is now infinite via AB. Sooner or later, however, C offers B a route to A of length two hops. B, knowing that line AB is useless, accepts the offer, and modifies its tables to show that A is three hops away via C. At this point B is routing packets destined for A to C, and C is sending them right back again. Having packets loop forever is not considered a good property to have in one's routing algorithm. This particular problem causes great anguish for the transport layer, as we shall see shortly.

To get around the problem of looping packets, several researchers (e.g., Chu78, Sega81) have proposed using the *optimal-*

ity principle to guarantee loop-free routing. This principle states that if B is on the optimal route from A to C, then the best route from B to C falls along the same route. Clearly if there were a better route from B to C, the best route from A to C could use it too. Consequently, the set of best routes to C (or any other destination) from all other IMPs forms a tree rooted at C. By explicitly maintaining all the trees, the routing algorithm can adapt but prevent looping. A good survey of routing algorithms can be found in SCHW80.

3.2 Congestion Control in Point-to-Point Networks

Now we turn to the problem of congestion in point-to-point networks. Actually, little is known about how to deal with it, and all the proposed solutions are rather ad hoc. Davies [DAVI72] suggested starting each network with a collection of special packets called *permits* that would roam about randomly. Whenever a host wanted to send a packet, its IMP would have to capture and destroy a permit before the new packet could be injected into the network.

This mechanism guarantees that the maximum number of packets in the network can never exceed the initial number of permits, which helps somewhat, but still does not guarantee that all the legal packets will not someday end up in one IMP, overloading it. Furthermore, no one has been able to devise a way to regenerate permits lost in IMP crashes (short of deadstarting the whole network, which is unacceptable). If these permits are not generated, carrying capacity will be permanently lost.

Another congestion control scheme is due to Irland [IRLA78]. This scheme calls for IMPs to monitor the utilization of each outgoing line. When a line utilization moves above a trigger value, the IMP sends a *choke packet* back to the source of each new packet needing that line, telling the source to slow down.

Kamoun [KAMO81] has proposed a congestion control scheme based on the observation that when packets must be discarded in an overloaded IMP, some packets are better candidates than others. In particular, if a packet has already made k hops, throwing it away amounts to discarding the investment in resources required to make those k hops. This observation suggests discarding packets with the smallest k values first. A variation of this idea that does not require a hop counter in each packet is to have IMPs discard newly injected packets from local hosts in order to salvage transit traffic with $k \geq 1$.

The limiting case of a congested network is a deadlocked network. If hosts A, B, and C are all full (no free buffers), and A is trying to send to B and B is trying to send to C and C is trying to send to A, a deadlock can occur, as shown in Figure 9.

Merlin and Schweitzer [MERL80a, MERL80b] describe several ways to prevent this kind of *store-and-forward deadlock* from occurring. One way is to provide each IMP with $m + 1$ packet buffers, where m is the longest path in the network. A packet newly arriving in an IMP from a local host goes into buffer 0. At the next IMP along the path it goes in buffer 1. At the following IMP it uses buffer 2. After having made k hops, it goes in buffer k. To see that the algorithm is deadlock free, consider the set of all buffers labeled m. Each buffer is in one of three states:

(1) empty,
(2) holding a packet destined for a local host,
(3) holding a packet destined for a distant host.

In Case 2 the packet can be delivered and the buffer freed. In Case 3 the packet is looping and must be discarded. In all cases the complete set of buffers labeled m can be made empty. Consequently, all packets in buffers labeled $m - 1$ can be either delivered or discarded, one at a time. The process can then be repeated, freeing the buffers labeled $m - 2$, and so on.

Other kinds of deadlocks in computer networks are discussed in GUNT81.

3.3 An Example Network Layer Protocol: X.25

To help standardize public long-haul networks, CCITT has devised a three-layer protocol of its own. The physical layer is X.21 (or X.21 bis, a stopgap analog interface to be used until the digital network arrives).

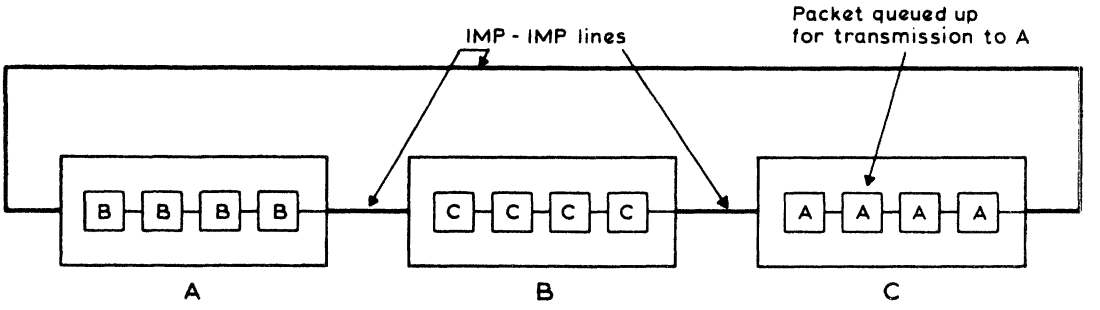


Figure 9. Store-and-forward lockup (deadlock).

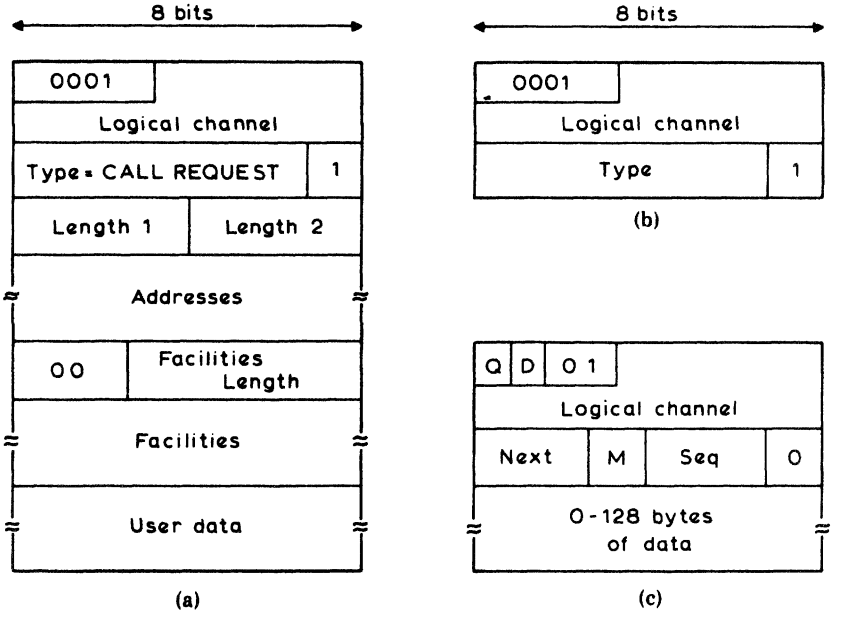


Figure 10. X.25 headers. (a) CALL REQUEST packet. (b) Control packet. (c) Data packet.

The data link layer consists of two variants of HDLC (LAP and LAPB). Whether the next layer is network layer protocol or a transport layer protocol is a matter of some debate in the network community. Let us call it a network layer protocol and discuss it now.

X.25 (which is the collective name for all three layers) is virtual circuit oriented [RYBC80]. To set up a virtual circuit, a host (DTE) sends a CALL REQUEST packet into the network. The remote host can either accept or reject the incoming call. If it accepts it, the virtual circuit is set up; otherwise the circuit is cleared.

Figure 10a shows the format of the CALL REQUEST packet. The first 4 bits are 0001. The next 12 bits are the virtual circuit number chosen by the originating host. The third byte is the type code of CALL REQUEST. The next byte gives the number of decimal digits in the caller's and callee's addresses, followed by up to 30 bytes containing the addresses themselves in binary coded decimal. (The telephone community has been using decimal numbers for 100 years, and old habits die hard.) The *Facilities* field is used to request services such as calling collect. Since the facilities field is variable length, a length field is needed. Finally, the user data field can be used in any way the user chooses, for example, to indicate which process within the called host expects the call.

When the CALL REQUEST packet arrives at the destination, that machine accepts or rejects the call by sending a packet of the form shown in Figure 10b. Acceptance or rejection is indicated in the *Type* field. Once the virtual circuit has been set up, both sides may send data packets at will, which makes the connection, by definition, full duplex. Either side may terminate the call by sending a CLEAR REQUEST packet, which is acknowledged by a CLEAR CONFIRMATION packet.

An ordinary data packet is shown in Figure 10c. The *Sequence* and *Next* fields are analogous to those in HDLC. Like HDLC, X.25 layer 3 also has an optional format with 7-bit sequence numbers. The *M* bit can be used by a host to indicate that more data follow in the next packet, thus partitioning the packet stream into multipacket units.

The meaning of the *Q* bit is not specified, but it is provided to allow the transport layer a means for distinguishing transport layer data packets from control packets. The *D* bit stands for Delivery confirmation. If a host sets it on all the packets sent on a certain virtual circuit, the *Next* field will contain a true acknowledgment from the remote host, producing an end-to-end confirmation. If, however, it is always set to 0, then the *Next* field just means that the local IMP (DCE) received the packet specified, not that the remote host did. Conceivably, when $D = 0$, the local IMP could write all the packets on magnetic tape to be mailed to the remote IMP for delivery in a couple of days (bargain basement service).

In the original version of X.25, only $D = 0$ was provided. That point generated so much controversy that delivery confirmation was added later, as was a pure datagram facility and something called *Fast Select*. With the Fast Select facility, the user data field in the CALL REQUEST packet is extended to 128 bytes and a similar field is added to the CLEAR REQUEST packet (used to reject incoming calls). Thus a host can send a short query in the CALL REQUEST packet and get the reply in the CLEAR REQUEST packet, without having to open a virtual circuit.

Because layers 2 and 3 in X.25 have so much overlap, it is perhaps useful to point out that the layer 2 sequence numbers and acknowledgments refer to the traffic between host and IMP for all virtual circuits combined. If a host sends the IMP seven packets (frames), each one for a different virtual circuit, the host must stop sending until an acknowledgment comes back. The layer 2 protocol is required to keep the host from flooding the IMP. In contrast, in layer 3, the sequence numbers are per virtual circuit and therefore flow control each connection separately.

X.25 layer 3 also has a few control packets. These include RESET and RESET CONFIRMATION, used to reset a virtual circuit; RESTART and RESTART CONFIRMATION, used to reset all virtual circuits after a host or IMP crash; RECEIVER READY, used for acknowledgments; RECEIVER NOT READY, used to indicate temporary problems and stop the other side even though the window is not full; and INTERRUPT and INTERRUPT CONFIRMATION, used to send out-of-band signals, such as breaks. All these control packets use the format of Figure 10b, in some cases augmented with an additional byte or two for additional information.

4. THE TRANSPORT LAYER

The network layer does not necessarily ensure that the bit stream sent by the source arrives intact at the destination. Packets may be lost or reordered, for example, owing to malfunctioning IMP hardware or software. The X.25 standard provides a mechanism (RESET and RESTART packets) for the network to announce to a host that it has crashed and lost track of both the current sequence numbers and any packets that may have been in transit. To provide truly reliable end-to-end (i.e., host-to-host) communication, another layer of protocol is needed: the transport layer. (Note that X.25 with $D = 1$ comes close to being end to end, but is not quite enough since it provides no way to transparently recover from network RESETs and RESTARTs.)

Another way of looking at the transport layer is to say that its task is to provide a network independent *transport service* to

the session layer. The session layer should not have to worry about any of the implementation details of the actual network. They must all be hidden by the transport layer, analogous to the way a compiler must hide the actual machine instructions from the user of a problem-oriented programming language.

4.1 The Transport Station

The program within the host that implements the transport service is called the *transport station.* Its chief functions are to manage connection establishment and teardown, flow control, buffering, and multiplexing. Although a transport station might conceivably offer only datagram primitives to its users, most offer (and emphasize) virtual-circuit primitives. As a bare minimum, the following primitives or their equivalents are normally available:

```
connum = CONNECT(local, remote),
connum = LISTEN(local),
  status = CLOSE(connum),
  status = SEND(connum, buffer, bytes),
  status = RECEIVE(connum, buffer,
           bytes)
```

The primitives for establishing a transport connection, CONNECT and LISTEN, take *transport addresses* as parameters. Each transport address uniquely identifies a specific transport station and a specific *port* (connection endpoint) within that transport station. For example, CCITT has decreed that X.25 will use 14-digit numbers for addressing. The first three identify the country, and the fourth identifies the network within the country. (Multiple country codes have been assigned to countries that expect to have more than ten public networks.) The last ten digits of the X.25 address are assigned by each network operator, for example, five digits to indicate hosts and five digits for the hosts to allocate themselves.

In our example, the LISTEN command tells the transport station that the process executing it is prepared to accept connections addressed to the indicated local address. The process executing the LISTEN is blocked until the connection is established, at which time it is released, with the variable *connum* being set to indicate the connection number. The connection number is needed because multiple connections may be open at the same time and a subsequent SEND, RECEIVE, or CLOSE must be able to tell which connection is meant. If something goes wrong, an error number can be returned in *connum* (e.g., positive for connection established, negative for error).

The CONNECT command tells the transport station to send a message (e.g., X.25 CALL REQUEST) to another host to establish a connection. When the connection has been established (or rejected, for example, due to illegal addresses), the connection number or error code is returned in *connum.*

An important design issue is what should the transport station do if a CALL REQUEST packet comes in specifying a transport address for which no LISTEN is pending? Should it reject the request immediately, or should it queue the request in the hope that a LISTEN will be done shortly? If the request is queued, should it time out and be purged if no LISTEN is forthcoming within a reasonable time? If so, what happens if the LISTEN finally occurs after the timeout?

In the above example, both LISTEN and CONNECT are blocking primitives, that is, the caller is halted until the command completes. Some transport stations use nonblocking primitives. In other words, both calls complete immediately, perhaps only checking the syntactic validity of the addresses provided. When the connection is finally established, or definitively rejected, the respective processes are interrupted. Some transport stations that use nonblocking primitives also provide a way for a process to cancel an outstanding LISTEN or CONNECT, as well as a method for a listening process to inspect an incoming connection request before deciding to accept or reject it.

The primitive CLOSE speaks for itself. The status returned would normally be "OK" if the connection actually existed and "error" if it did not.

The SEND and RECEIVE primitives do the real work of message passing. For the sake of clarity, we refer to the entities exchanged here as "messages," to distinguish

them from the "packets" of the network layer and "frames" of the data link layer. A message will be encased in a packet, which will be inserted into a frame before transmission, of course. SEND specifies the connection on which to send, the buffer address, and the number of bytes. RECEIVE has the same parameters, although here *bytes* might initially contain the buffer size and later be filled in with the size of the received message. Again, both of these could be provided in nonblocking as well as blocking versions.

A more elaborate transport station could offer commands to send and receive datagrams, to send and receive interrupt signals, to reset the connection in the event of error, and to interrogate the status of the other side, a facility particularly useful for recovering from network layer failures.

4.2 Establishing and Closing Connections

As we pointed out earlier, one consequence of adaptive routing is that packets can loop for an indefinite period of time. If a packet gets trapped, the sending transport station will eventually time out and send a duplicate. If the duplicate gets through properly, but the original packet remains trapped for a while, problems can arise when it finally escapes and is delivered. Imagine, for example, what would happen if a message instructing a bank to transfer a large sum of money were stored and later repeated, long after the transaction had already been completed.

A useful first step is to limit the amount of time that a packet can exist in the network. For example, a counter could be put in the packet header. Each time the packet was forwarded, the counter could be decremented. When the counter reached zero, the packet would be discarded. Alternatively, a timestamp in the packet could be used to render it obsolete after a certain interval.

The next step is to have the transport stations use a sequence space so large (e.g., 32 bits) that no packet can live for a complete cycle. As a result, delayed duplicates can always be detected by their sequence numbers. However, if all new connections always start with sequence number 0, packets from previous connections may come back to plague later ones. Therefore, it is necessary to have each new connection initialize its sequence numbers to a value known to be higher than that of any existing packet.

Unfortunately, not even these measures are enough. Since each host has a different range of sequence numbers outstanding, each one must specify the initial sequence number for packets it will send. Assume that sequence numbers are chosen during the call establishment phase. With some bad luck, the following scenario could occur at an instant when *A* wanted to set up a connection with sequence number 100 to *B*:

(1) *A* sends a CALL REQUEST packet with sequence number 100.
(2) The packet is lost.
(3) An old CALL REQUEST with sequence number 50 suddenly arrives at *B*.
(4) *B*'s CALL ACCEPT packet, with sequence number 700, is lost.
(5) An old CALL ACCEPT from *B* with sequence number 650 suddenly arrives at *A*.

At this point the connection is fully established, with *A* about to send packet 100, but *B* expecting packet 50. Similarly, *B* intends to send packet 700, but *A* expects 650. The result is a deadlock.

Tomlinson [TOML75] proposed a connection establishment protocol that works even in the face of delayed control packets. It is called the *three-way handshake*. An example follows (S means sequence, A means acknowledgment):

(1) *A* sends a CALL REQUEST packet with S = 100.
(2) *B* sends a CALL ACCEPTED packet with S = 700, A = 100.
(3) *A* sends a packet with S = 101, A = 700.

Now consider what happens in the face of the same lost and duplicate packets that lead to deadlock above. When *B* receives the CALL REQUEST with S = 50, it replies with S = 700, A = 50. If this packet gets through, *A* sees the bad acknowledgment and rejects the connection. The only way *A* can be spoofed is for an old CALL AC-

CEPTED packet with A = 100 to appear out of the blue. Such a packet could only be generated in response to an *old* CALL REQUEST packet with S = 100, something *A* has not sent for a long time. Sunshine and Dalal (SUNS78] discuss this problem in more detail.

By now you should be convinced that the protocol required to establish a transport layer connection in the face of an unreliable network layer is nontrivial. What about closing a connection? Surely that, at least, is easy: *A* sends *B* a request to close, and *B* sends back a close acknowledgment. Unfortunately, things are not that simple. As an example, let us briefly consider the two-army problem.

Two divisions of the white army are encamped on the opposite walls of a valley occupied by the blue army. If both divisions attack simultaneously, the white army will win; if either division attacks alone, it will be massacred. The white army divisions must synchronize their attack using an unreliable channel (e.g., a messenger subject to capture). Suppose that white's *A* division sends the message: "Let's attack at teatime," and gets the reply "OK." Division *B* has no way of being sure that the reply got back. If it just goes ahead and attacks, it might get slaughtered. Furthermore, *A* is well aware of this line of reasoning and hence may be afraid to attack, even after having received an acknowledgment.

At this point you may be thinking: "Why not use a three-way handshake here?" Unfortunately, it does not work. *A* could confirm receipt of *B*'s acknowledgment, but because this confirmation could get lost, *A* does not know which situation holds:

(1) *B* got the confirmation and the war is on.
(2) The confirmation got lost and the war is off.

How about a four-way handshake? This is no better. An *n*-way handshake? Still no good. In all cases, the sender of the last message cannot tell whether or not it arrived. If its arrival is essential to starting the war, the sender has no way of telling whether the receiver is going to attack or not. If its arrival is not essential to starting the war, one can devise an equivalent protocol not containing it and apply the above reasoning to the new protocol.

The implication of all this is that a closing protocol in which neither side can hang up until it is convinced that the other side also intends to hand up is, at the very least, more complicated than it at first appears. The issue is discussed further in SUNS78 and YEMI79.

4.3 Flow Control and Buffering

An important design issue in the transport layer is flow control. Since no transport station has an infinite amount of buffer space, some way must be provided to prevent a fast sender from inundating a slow receiver. Flow control is well known in other contexts, such as operating systems design, where it is known as the producer-consumer problem. Although it also occurs in the data link and network layers, some new complications are present in the transport layer.

To start with, the data link layer usually has one connection for each adjacent IMP—a handful at most—whereas the transport layer in a large multiprogrammed computer may have many connections open simultaneously. A stop-and-wait protocol in the transport layer is usually undesirable, since both sender and receiver would have to be scheduled and run for every message sent. If each machine had a response time of 500 ms between the moment a process became ready to run and the time it ran, the transport connection could support two messages per second, at best, and probably fewer. Consequently, large windows are needed to achieve high throughput, but the combination of large windows and many open connections necessitates many buffers, most of which are idle most of the time.

An alternative design is not to dedicate buffers to specific connections, but to maintain a buffer pool and pull buffers out of the pool and assign them to connections dynamically, as needed. This strategy entails some risk, since no buffer may be available when one is needed. To avoid this risk, a buffer reservation protocol is needed, increasing traffic and overhead. Furthermore, if the buffers are of fixed size and messages

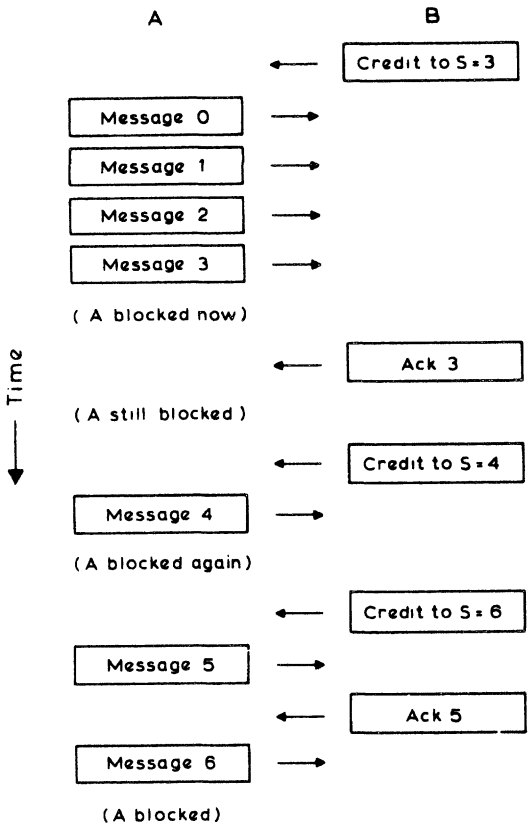


Figure 11. Flow control using credits.

vary from a few characters to thousands of characters, a pool of fixed-sized buffers is not attractive either. No one solution is best. Each transport station must make compromises appropriate to its expected work load.

Another important issue is the relation of flow control to error control. With the sliding-window protocol, an acknowledgment message has two distinct functions: to announce that a message has arrived and to grant the sender permission to send another message. In the transport layer, this coupling is not always desirable.

To see why, consider the dilemma of a transport station that is chronically short of buffer space. What should it do if a message arrives, but the process using the connection has no RECEIVEs outstanding? If it does nothing, the sending transport station will eventually time out and send it again. If it sends an acknowledgment, the other transport station may send yet another message. The problem comes from the fact that the transport station has no control over the rate at which the user does RECEIVEs. Earlier, we more or less assumed that the network layer was always hungry for new packets—a reasonable assumption, since the network code has little else to do.

One way out of this dilemma is to decouple acknowledgments and flow control. To do so, we introduce two kinds of control messages: acknowledgments and credits. An acknowledgment simply says that a certain message (and by implication, all lower numbered messages) has arrived safely. Upon receiving an acknowledgment, the sender may release the buffers containing all the acknowledged messages, since none of them will ever be retransmitted. However, an acknowledgment does *not* imply permission to send any more messages.

Such permission is granted by a credit message. When a connection is established, the receiver grants some credits to the sender. These credits may be for so many messages or so many bits or both. Every time a message is sent, the credits for message count and/or bit count are decremented. When the credits are all used up, the sender must stop sending until more credits arrive. Such credits may be sent as distinct messages or they may be piggybacked onto data or acknowledgment messages. This scheme provides a simple and flexible mechanism for preventing unnecessary retransmissions in the presence of heavy and variable demands on limited buffer space. An example is given in Figure 11.

4.4 Connection Multiplexing

Multiplexing of connections plays an important role in several layers. In lower layers, for example, packets and frames ultimately destined for different hosts are multiplexed onto the same output lines. In the transport layer, two different forms of multiplexing occur. In *upward multiplexing* (shown in Figure 12a) several transport connections are multiplexed onto the same network connection (e.g., the X.25 virtual circuit). Upward multiplexing is often financially better, since some carriers charge

by the packet and also by the second for each virtual circuit that is open.

Now consider the plight of an organization (e.g., an airline) that has 100 telephone operators to handle customer inquiries. If each operator is assigned to a separate virtual circuit, 100 virtual circuits to the central computer will be open all day. The other option would be to use a single virtual circuit to the computer, with the first byte of data being used to distinguish among the operators. The latter has the disadvantage that if the traffic is heavy, the flow control window may always be full, thus slowing down operation. With a dedicated virtual circuit per operator, full windows are much less likely to occur.

The other form of multiplexing, *downward multiplexing* (Figure 12b), becomes interesting when the network layer window is too small. Suppose, for example, a certain network offers X.25, but does not support the 7-bit sequence number option. A user with a large number of data to send might find himself constantly running up against full windows. One way to make an end run around the problem is to open multiple virtual circuits for a single-transport connection. Packets could be distributed among the virtual circuits in a round-robin fashion, first a packet on circuit 0, then a packet on circuit 1, then one on circuit 2, and so on.

Conceivably the two forms could even be combined. For n connections, k virtual circuits could be set up with traffic being dynamically assigned.

5. THE SESSION LAYER

In many networks, the transport layer establishes and maintains connections between hosts. The session layer establishes and maintains connections, called *sessions*, between specific pairs of processes. On the other hand, some networks ignore the session layer altogether and maintain transport connections between specific processes. The ISO OSI Reference Model is exasperatingly vague on this point, stating only that the session layer connects "presentation-entities" and that the transport layer connects "session-entities."

To keep the following discussion from

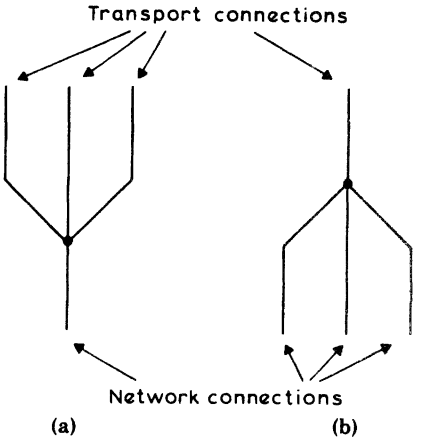


Figure 12. (a) Upward multiplexing. (b) Downward multiplexing.

vanishing in a linguistic fog, we assume that transport connections are between hosts and session connections are between processes. Thus, when a process wants to talk to another process, it makes its desires known to the session layer, which then engages the services of the transport layer to set up a transport connection to the remote host for use by the session.

A principal task of the session layer is to connect two processes together into a session. Since it is inconvenient for users to be aware of hard transport addresses, the session layer could allow them to refer to destinations by symbolic name, with the session layer doing the mapping onto transport addresses. For example, a user could say, in effect, "Give me a phototypesetter process," with the session layer worrying about where such beasts were to be found.

When a session is set up, an activity often call *session binding*, certain conventions about the coming session can be established. Typical conventions are half-duplex versus full-duplex data transfer, character codes, flow control window sizes, the presence or absence of encryption or text compression, and how to recover from transport layer failures.

Another task that the session layer can perform is particularly useful in networks where the user primitives for sending and receiving messages are nonblocking, and where the user may have multiple requests outstanding on the same session at any instant. Under these circumstances, replies

may come back in an order different from that in which the requests were sent. The session layer's *dialog control* function can keep track of requests and replies and reorder them if need be to simplify the design of the user programs.

Another aspect of dialog control is bracketing groups of messages into atomic units. In many database applications it is highly undesirable that a transaction be broken off part way, as a result of a network failure, for example. If the transactions consists of a group of messages, the session layer could make sure that the entire group had been successfully received at the destination before even attempting to start the transaction.

Our discussion of the session layer is now complete. The brevity of this section is directly related to the fact that few networks make much of a distinction between the transport and session layers. In fact, many networks have neither a session layer nor any of the dialog control functions belonging to the session layer. While there are no internationally accepted standards for the transport layer yet, there are at least a few serious proposals that have been under discussion for several years [DEPA76, INWG78]. Session layer protocols have not come as far yet. This situation has occurred because the protocol community has been tackling the layers more or less bottom up and is currently in the vicinity of layer 4. Higher layer standards will no doubt be forthcoming in the future.

6. THE PRESENTATION LAYER

The function of the presentation layer is to perform certain generally useful transformations on the data before they are sent to the session layer. Typical transformations are text compression, encryption, and conversion to and from network standards for terminals and files. We examine each of these subjects in turn.

6.1 Text Compression

Bandwidth is money. Sending thousands of trailing blanks across a network to be "printed" is a good use of neither. Although the network designers could leave the matter of text compression to each user program, it is more efficient and convenient to put it into the network architecture as one of the standard presentation services.

Obvious candidates for text compression are runs of repeated bits (e.g., leading zeros) and repeated characters (e.g., trailing blanks). Huffman coding is also a possibility. Since text compression is such a well-known subject outside the network context (see, e.g., DAVS76), we do not consider it further here.

6.2 Encryption Protocols

Information often has great economic value. As an example, just think about the data transmitted back by oil companies from exploratory sites. With more and more data being transmitted by satellite, the problem of data security looms ever larger. The financial incentive to erect an antenna to spy on competitors is great and the cost is low. Furthermore, privacy legislation in many countries puts a legal requirement on the owners of personal data to make sure such data are kept secret. All these factors combine to make data encryption an essential part of most networks. The December 1979 issue of *Computing Surveys* [COMP79] is devoted to cryptography and contains several introductory articles on it.

An interesting question is: "In which layer does the encryption belong?" In our view, encryption is analogous to text compression: ordinary data go in and compressed or indecipherable data come out. Since everyone agrees that text compression is a presentation service, logically encryption should be too. For historical reasons and implementation convenience, however, it is often put elsewhere, typically the transport layer or the data link layer.

The purpose of encryption is to transform the input, or *plaintext*, into an output, or *ciphertext*, that is incomprehensible to anyone not privy to the secret *key* used to parameterize the transformation. Thus plaintext is converted to ciphertext in the presentation layer of the source machine and reconverted to plaintext in the presentation layer of the destination machine. In neither machine should the user programs be aware of the encryption, other than having specified encryption as an option when the session was bound.

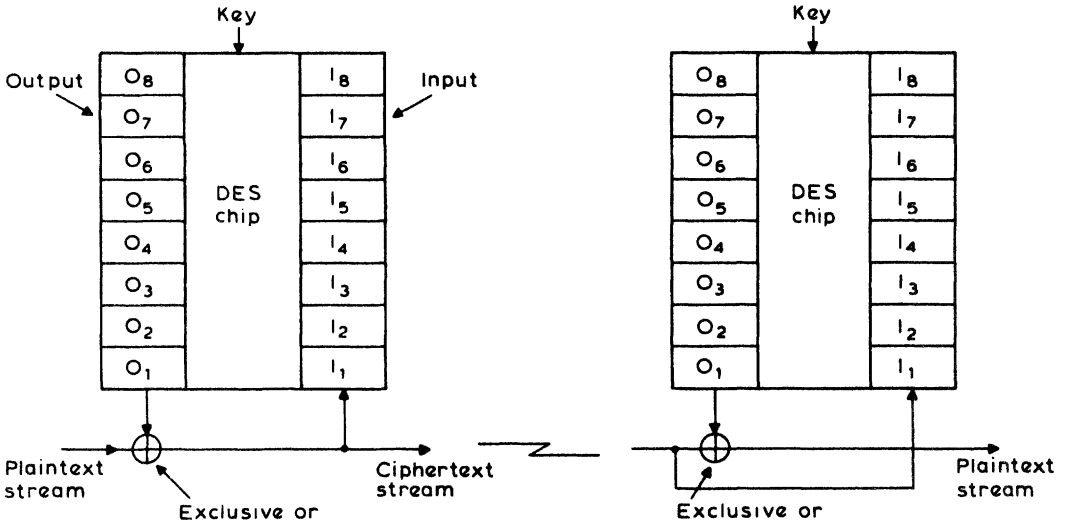


Figure 13. A stream cipher using DES. Data arrives from the left and is encrypted for transmission. The destination machine decrypts it and outputs the plaintext.

One of the best-known encryption methods is the *substitution cipher*, in which a unit of plaintext is converted into a unit of ciphertext. In a *monoalphabetic cipher*, each letter is converted into another letter according to a fixed rule. For example, "a" becomes "M," "b" becomes "R," "c" becomes "G," etc. In this example, the encryption key is MRG ..., that is, the ciphertext corresponding to the plaintext abc.... Although 26! different monoalphabetic substitutions exist, these ciphers can be broken by a clever ten-year-old using the frequency statistics of natural language.

Most computer ciphers use the same principle, but on a larger scale. The U.S. federal government has adopted a substitution cipher that is fast becoming a de facto standard for nongovernmental organizations as well. The *DES* (Data Encryption Standard) cipher takes a 64-bit plaintext input block and produces a 64-bit ciphertext output block. The transformation is driven by a 56-bit key. Conceptually, at least, one could prepare a big table, with 2^{64} columns, one for each possible input, and 2^{56} rows, one for each possible key. Each table entry is the ciphertext for the specified input and key.

DES can also be operated as a *stream cipher*, as shown in Figure 13. The input shift registers on both source and destination machine are initialized to the same 8-byte (random) number, $I_1, \ldots, I_8$. Data are presented for encryption 1 byte at a time, not 8 bytes at a time. When a byte arrives, the DES chip converts the 8 bytes $I_1, \ldots, I_8$ into the output $O_1, \ldots, O_8$. Then O_1 is Exclusive Or'ed with the input to form the ciphertext byte. The ciphertext byte is both transmitted and fed back into I_1, shifting I_2 to I_3, and so on. I_8 is shifted out and lost. Decryption at the other end is similar. Note that feeding back the ciphertext into the DES input register makes subsequent encryption dependent on the entire previous plaintext, and so a given sequence of 8 plaintext bytes will have a different ciphertext on each appearance in the plaintext.

DES has been the subject of great controversy since its inception [DAVA79, DIFF76a, HELL79, HELL80]. Some computer scientists feel that a wealthy and determined intruder who knew, for example, that a certain message was in ASCII, could determine the key by trying all keys until he found one that yielded ASCII plaintext (i.e., only codes 0–127 and not 128–255). If the ciphertext is k bytes long, the probability of an incorrect key yielding ASCII input is 2^{-k}. For even a single line of text, it is unlikely that any key but the correct one could pass the test.

The dispute centers about how much a DES-breaking machine would cost. In 1977, Diffie and Hellman [DIFF77] designed one and computed its cost at 20 million dollars.

The DES supporters say this figure is too low by a factor of 10, although even they concede DES cannot hold out forever against the exponential growth of very large-scale integrated circuits.

To use DES, both the source and destination must use the same key. Obviously the session key cannot be sent through the network in plaintext form. Instead, a master key is hand carried in a locked briefcase to each host. When a session is set up, a key manager process somewhere in the network picks a random key as session key, encrypts it using the master key, and sends it to both parties for decryption. Since the plaintext of this message is a random number, it is hard to break the cipher using statistical techniques. Numerous variations of the idea exist, typically with master keys, regional keys, local keys, and the like.

Shamir [SHAM79] has devised a clever way to share (master) keys in a flexible way among a large group of people, so that n arbitrary people can get together and assemble the master key, but $n - 1$ people can gain no information at all. Basically, each person is given a data point that lies on a degree $n - 1$ polynomial whose y intercept is the key. With n data points, the polynomial, hence the key, is uniquely determined, but with $n - 1$ data points it is not. Modulo arithmetic is used for obfuscatory purposes.

All the master key methods have a significant drawback, though: it is impossible for computers that have not previously had any contact with each other to agree on a session key in a secure way. Considerable academic research has been done on this topic in recent years (not without its own controversy—see SHAP77 and SUGA79), and some interesting results have been achieved. Merkle [MERK78a], for example, has suggested that two strangers, A and B, could establish a key as follows. A sends k ciphertext messages to B with the instruction to pick one of them at random and break it by brute force (i.e., try all possible keys until a plaintext starting with 64 0's appeared). The rest of the message consists of two random numbers, the key number and the key itself. Having broken the cipher, B then sends the key number back to A to indicate which message was broken. Clearly an intruder will have to break $k/2$ messages on the average to find the right one. By adjusting k and the difficulty of breaking a message, A can achieve any degree of security desired.

A completely different approach to key distribution is that of *public key cryptography* [DIFF76b] in which each network user deposits an encryption key E in a publicly readable file. The user keeps the decryption key D secret. The keys must satisfy the property that $D(E(P)) = P$ for an arbitrary plaintext P. (This is essentially the definition of a decryption key.) The cipher system must be such that D cannot be deduced from the publicly known E.

With this background, the encryption system is obvious and trivial: to send a message to a stranger, you just encrypt it with his publicly known key. Only he knows the decryption key and no one can deduce it from the encryption key, so the cipher cannot be broken. The utility of the whole system depends on the availability of key pairs with the requisite properties. Much effort has gone into searching for ways to produce such key pairs. Some algorithms have already been published [MERK78b, RIVE78, SHAM80]. The scheme of Rivest et al. effectively depends on the fact that given two huge prime numbers, generating their product (the public key) is computationally easy, but given the product, finding the prime factors (the secret key) is very hard. In effect, their system takes advantage of the fact that the computational complexity of factorization is high.

Another area where cryptography plays a major role is in authentication. Suppose that a customer's computer instructs a bank's computer to buy a ton of gold and debit a certain account. The bank complies, but the next day the price of gold drops sharply and the customer denies ever having issued any purchase order. How can the bank protect itself against such unscrupulous customers? Traditionally, court battles over such matters have focused on the presence or absence of an authorized handwritten signature on a piece of paper. With electronic funds transfers and similar applications the need for "digital" (i.e., electronic) signatures is obvious.

With a slight additional restriction, pub-

lic key cryptography can be used to provide these badly needed digital signatures. The restriction is that the encryption and decryption algorithms be chosen so that $D(E(P)) = E(D(P))$. In other words, the order of applying encryption and decryption must be interchangeable. The M.I.T. algorithm [RIVE78] has this property.

Now let us reconsider the ton-of-gold problem posed earlier. To protect itself, the bank can insist that a customer C use the following protocol for sending signed messages. First, the customer encrypts the plaintext message P with his secret key; that is, the customer computes $D_C(P)$. Then the customer encrypts this result with the bank's public key E_B, yielding $E_B(D_C(P))$. When this message arrives, the bank applies its decryption key D_B to get

$$D_B(E_B(D_C(P))) = D_C(P).$$

Now the bank applies the customer's public key, E_C, to recover P. The bank also saves P and $D_C(P)$ in case trouble arises.

When the angry letter from the customer arrives, the bank takes both P and $D_C(P)$ to court and asks the judge to decrypt the latter using the customer's public key. When the judge sees that the decryption works, he will realize that the bank must be telling the truth. How else could it have come into possession of a message encrypted by D_C, the customer's secret key? Since the bank does not know any of its customer's secret keys, it cannot forge messages (to generate commissions); hence customers are also protected against unscrupulous banks. While in jail, the customer will have ample time to devise interesting new public key algorithms.

Unfortunately, as Saltzer [SALT78] has pointed out, the public key digital signature protocol suffers from some nontechnical problems. For example, immediately after the price of gold drops, the management of the gold-buying company could run to the police claiming it had just become aware of yesterday's key burglary. Depending on local laws, the company might or might not be able to weasel out of obligations undertaken with the "stolen" key. (As an aside, note that the owner of a stolen credit card usually has only a small liability for its subsequent misuse.)

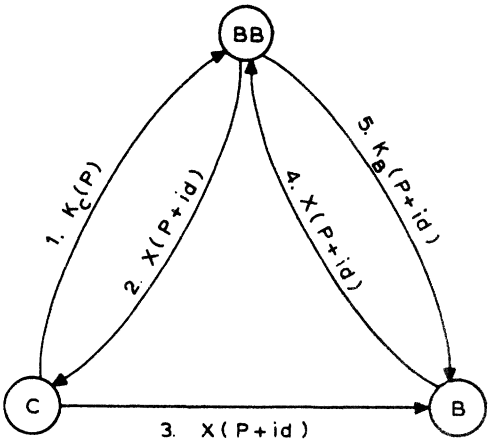


Figure 14. A digital signature protocol with conventional cryptography.

Saltzer also points out that a company is free to change its public key at will. Stronger yet, it may be company policy to do so regularly. If the company changes its key before accusing the bank of fabricating the purchase order, it will be impossible for the bank to convince the judge. This observation suggests that some central key registration authority may be needed. However, if such a central authority, call it Big Brother (BB), exists, conventional cryptography can also be used to achieve digital signatures [NEED78, POPE79].

The signature protocol using DES is illustrated in Figure 14. When a new customer C joins the system, the customer hand carries a secret (DES) key, K_C, to BB. Thus, BB has each user's secret key and can therefore send and receive secure messages from each user. In addition, BB has a secret key of its own, X, that it never discloses to anyone. The protocol for buying gold is as follows (P is the plaintext purchase order):

(1) The customer sends $K_C(P)$ to BB.
(2) BB decrypts the message and returns $X(P$ + identification).
(3) The customer sends $X(P$ + identification) to the bank.
(4) The bank sends $X(P$ + identification) to BB.
(5) BB sends $K_B(P$ + identification) back to the bank.

The "identification" appended to the message by BB consists of the customer's identity, something that BB can guarantee since the incoming message is encrypted by a key only known to one user, plus the date, time, and perhaps a sequence number. Messages encrypted by X can be freely sent through the network, since only BB can decrypt them, and BB is assumed to be trusted. If a dispute arises, the bank can go to the judge with $X(P$ + identification), which the judge can then order BB to decrypt. The judge will then see the identification and know who sent the original message. While in jail, the customer will have ample time to devise interesting new signature protocols using conventional cryptography.

6.3 Virtual-Terminal Protocols

Dozens of brands of terminals are in widespread use, no two of which are identical. Needless to say, a network user who has just been told that the program or host he wishes to use does not converse with his brand of terminal is not likely to be a happy user. For example, if the program treats carriage returns and line feeds as equivalent and the user's terminal only has a "newline" key, which generates one of each, the program will perceive alternate lines as being empty.

To prevent such difficulties, protocols have been invented to try to hide terminal idiosyncracies from application (i.e., user) programs. Such protocols are known as *virtual-terminal protocols*, since they attempt to map real terminals onto a hypothetical network virtual terminal. Virtual-terminal protocols are part of the presentation layer.

Broadly speaking, terminals can be divided up into three classes: scroll mode, page mode, and form mode. Scroll-mode terminals do not have any intelligence. When a key is struck, the character is sent over the line and perhaps printed as well. When a character comes in over the line, it is just displayed. Most hard-copy terminals, and some of the less expensive CRT terminals, are scroll-mode terminals.

Even though scroll-mode terminals are simple, they still can differ in many ways: character set, line length, half duplex/full duplex, overprinting and the way line feed, carriage return, tab, vertical tab, backspace, form feed, and break are handled.

Page-mode terminals are typically CRT terminals with 24 or 25 lines of 80 characters. Most of these have cursor addressing, so that the operator or the program can randomly access the screen. Some of them have a little local editing capability. They have the same potential differences as scroll-mode terminals, and, additionally, problems with screen length, cursor addressing, blinking, reverse video, color, multiple intensities, and the details of the local editing.

Form-mode terminals are sophisticated microprocessor-based devices intended for data entry. They are widely used in airline reservations, banking, and many other applications. In a typical situation the computer displays a form for the operator to fill out using cursor control and local editing facilities. The completed form is then sent back to the computer for processing. Sometimes the microprocessor can perform simple syntax checking, to make sure, for example, that a bank account field contains only numbers.

Two kinds of virtual-terminal protocols are commonly used. The first one is intended for scroll-mode terminals and is based on the ARPANET Telnet protocol [DAVD77]. When this type of protocol is used, the designers invent a fictitious virtual terminal onto which all real terminals are mapped. Application programs output virtual-terminal characters, which are mapped onto the real terminal's character set by the presentation layer at the destination. Supporting a new kind of real terminal thus requires modifying the presentation layer software to effect the new mapping, but does not require changing any of the application programs.

Since most of the current research in virtual-terminal protocols focuses on page- or form-mode terminals, let us move on to them. A general model that has been widely accepted is the data structure model of Schicker and Duenki [SCHI78]. Roughly speaking, protocols based on this approach use the model of Figure 15. Each end of a session has a data structure that represents the state of the virtual terminal.

The data structure consists of a collection

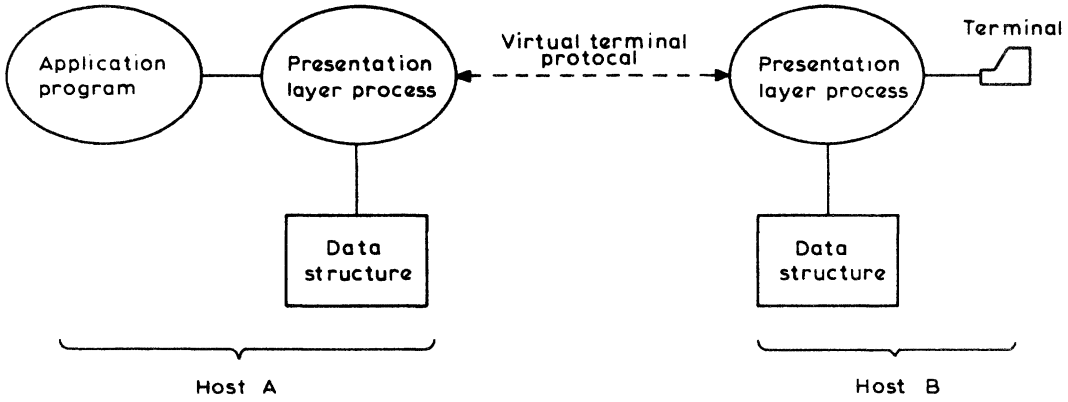


Figure 15. Virtual-terminal protocol model.

of fields, each of which contains certain attributes. Typical attributes are the size of the field, whether it accepts numbers, letters, or both, its rendition (an abstract concept used to model color, reverse video, blinking, and intensity), whether it is protected against operator modification or not, and so forth. The program is written using abstract operations on the data structure. Every time the program changes the data structure on its machine, the presentation layer sends a message to the other machine telling it how to change its data structure. The remote presentation layer is responsible for updating the display on the real terminal to make it correspond to the newly changed data structure. Similarly, changes made to the display by the human operator are reflected in the data structure on the operator's side of the session. Messages are then sent to bring the other side up to date. The protocol used for these messages is the virtual-terminal protocol.

Although a clever presentation layer can come a long way toward hiding the properties of the real terminal from the user program, it cannot work miracles. If the program needs a 24 × 80 screen with cursor addressing and four renditions, the presentation layer will be hard pressed to map everything onto a simple hard-copy terminal. Consequently, virtual-terminal protocols always have an option negotiation facility that is used to establish what each end of the connection is able to provide and what it wants from the other end.

This negotiation can be symmetric or asymmetric. In symmetric negotiation, each end announces its capabilities, inspects its partner's, and sets the parameters to the lowest common denominator. For example, if one end has a 24 × 80 screen, and the other has a 25 × 72 screen, the screen used will be 24 × 72. In asymmetric negotiation, one side makes a proposal and the other side accepts or rejects the proposal. If the proposal is rejected, the proposer may try again. Symmetric negotiation solves the problem of who should go first, but requires more complicated rules to determine what the result of an exchange is. It can also fail, for example, if both sides want to work in alternating (half-duplex) mode, and each wants to go first.

Another important design issue in virtual-terminal protocols is how to handle interrupts (attentions). When a user hits the "break" or "quit" key to terminate an infinite loop with a print statement in it, the presentation layer must purge the pipe of input already queued up; otherwise break will have no apparent effect. It is easy for the presentation layer on the terminal side to begin discarding input upon seeing a break, but it is much harder to determine when to stop discarding. Waiting for the prompt character does not work, since it might occur in the data to be discarded. A special out-of-band signaling protocol is needed. A survey of virtual-terminal protocols is given in DAY80.

6.4 File Transfer Protocols

The most common uses of computer networks at present are for logging onto re-

mote machines and transferring files between machines. These two areas are similar in that just as there is a need for programs to talk to a variety of incompatible terminals, there is a need for programs to read a variety of incompatible files. In principle, the same approach can be used for file transfer as for terminals: define a network standard format and provide a mapping from and to each existing file format.

In practice, this approach seems to work fairly well for terminals, but less well for files, primarily because the differences between terminals are not as great as between file types. Mapping reverse video onto blinking is straightforward compared to mapping 60-bit CDC floating point numbers onto 32-bit IBM floating point numbers, especially when the numbers are strewn randomly throughout the file.

Files are transferred for four primary reasons:

(1) to store a file for subsequent retrieval;
(2) to print a remote file on the local printer;
(3) to submit a file as a remote job;
(4) to use a remote file as data input or output.

Each category of use has its own peculiarities.

When a file is stored for subsequent retrieval, it must be possible to produce an exact, bit-for-bit copy of it upon request. Clearly transmission must be fully transparent, without escape codes that do funny things. The number of bits in the file must be recorded in the stored file, to allow transport between machines with differing word lengths. The last word on the storage machine may be partially full, and so some record of how many bits are in use is required.

When a file is transferred to be printed, problems can arise as a result of different print conventions. Some machines store print files in FORTRAN format, with fixed-length records (with or without some fudge for trailing blanks), and carriage control characters in column 1. Other machines use ASCII style variable-length records, with line feeds and form feeds for indicating vertical motion. When the file is being moved to be used for remote job entry, the same problems are present.

Moving data files containing mixtures of integers, floating point numbers, characters, etc., between machines is nearly impossible. In theory, each data item (e.g., integer, floating point number, character) could occupy one record in a canonical format, with the data type and value both explicitly stored. In practice the idea does not seem to work well, not only because of problems of interfacing existing software to it, but also because of the high overhead and the problems involved in converting floating point numbers from one format to another.

Another aspect of file transfer is file manipulation. Users often need to create, delete, copy, rename, and otherwise manage remote files. Most file transfer protocols tend to concentrate on this aspect of the problem because it is not as hopeless as the conversion aspect. Gien [GIEN78] has described a file transfer protocol in some detail.

7. SUMMARY

Computer networks are designed hierarchically, as a series of independent layers. Processes in a layer correspond with their peers in remote machines using the appropriate protocol, and with their superiors and subordinates in the same machine using the appropriate interface. The ISO OSI Reference Model has been designed to provide a universal framework in which networking can be discussed. Few existing networks follow it closely, but there is a general movement in that direction.

The seven-layer ISO model can be briefly summarized as follows. The physical layer creates a raw bit stream between two machines. The data link layer adds a frame structure to the raw bit stream, and attempts to recover from transmission errors transparently. The network layer handles routing and congestion control. The transport layer provides a network-independent transport service to the session layer. The session layer sets up and manages process-to-process connections. The presentation layer performs a variety of useful conversions. Finally the application layer is up to

the user, although some industry-wide protocols may be developed in the future.

The literature on computer networks is huge. Readers unfamiliar with it, but wishing to continue their study of the subject, may be interested in the textbooks by Davies et al. [DAVI79] and Tanenbaum [TANE81], or the book edited by Kuo [KUO81].

ACKNOWLEDGMENTS

I would like to thank Yogen Dalal, Adele Goldberg, and an anonymous, but artistic, technical editor for their numerous and helpful comments.

REFERENCES

ABRA70 ABRAMSON, N. "The ALOHA system—another alternative for computer communications," in *Proc. 1970 Fall Jt. Computer Conf.*, AFIPS Press, Arlington, Va., pp. 281–285.

ABRA73 ABRAMSON, N. "The ALOHA system," in *Computer-communication networks*, N. Abramson and F. Kuo (Eds.), Prentice-Hall, Englewood Cliffs, N.J., 1973.

BARA64 BARAN, P. "On distributed communication networks," *IEEE Trans. Commun. Syst.* **CS-12** (March 1964), 1–9.

BERT80 BERTINE, H. V. "Physical level protocols," *IEEE Trans. Commun.* **COM-28** (April 1980), 433–444.

BOGG80 BOGGS, D. R., SHOCH, J. F., TAFT, E. A., AND METCALF, R. M. "Pup: An Internet architecture," *IEEE Trans. Commun.* **COM-28** (April 1980), 612–624.

CAPE79a CAPETANAKIS, J. I."Generalized TDMA: The multi-accessing tree protocol," *IEEE Trans. Commun.* **COM-27** (Oct. 1979), 1476–1484.

CAPE79b CAPETANAKIS, J. I. "Tree algorithms for packet broadcast channels," *IEEE Trans. Inf. Theory* **IT-25** (Sept. 1979), 505–515.

CHLA76 CHLAMTAC, I. "Radio packet broadcasted computer network—the broadcast recognition access method," M.S. thesis, Dep. Mathematical Sciences, Tel Aviv Univ., Tel Aviv, Israel, 1976.

CHLA79 CHLAMTAC, I., FRANTA, W. R., AND LEVIN, D. "BRAM: The broadcast recognizing access method," *IEEE Trans. Commun.* **COM-27** (Aug. 1979), 1183–1190.

CHU78 CHU, K. "A distributed protocol for updating network topology information," Rep. RC 7235, IBM Thomas J. Watson Res. Cent., Yorktown Heights, N.Y., 1978.

CLAR78 CLARK, D. D., POGRAN, K. T., AND REED, D. P. "An introduction to local area networks," *Proc. IEEE* **66** (Nov. 1978), 1497–1517.

COMP79 *Computing Surveys* **11**, 4 (Dec. 1979).

CROW73 CROWTHER, W., RETTBERG, R., WALDEN, D., ORNSTEIN, S., AND HEART, F. "A system for broadcast communication: Reservation-Aloha," in *Proc. 6th Hawaii Int. Conf. Systems Science*, 1973, pp. 371–374.

DAVA79 DAVIDA, G. I. "Hellman's scheme breaks DES in its basic form," *IEEE Spectrum* **16** (July 1979), 39.

DAVD77 DAVIDSON, J., HATHAWAY, W., POSTEL, J., MIMNO, N., THOMAS, R., AND WALDEN, D. "The ARPANET Telnet protocol: Its purpose, principles, implementation, and impact on host operating system design," in *Proc. 5th Data Communication Symp.* (ACM/IEEE) (1977), pp. 4.10–4.18.

DAVI72 DAVIES, D. W. "The control of congestion in packet-switching networks," *IEEE Trans. Commun.* **COM-20** (June 1972), 546–550.

DAVI73 DAVIES, D. W., AND BARBER, D. L. A. *Communication networks for computers*, Wiley, New York, 1973.

DAVI79 DAVIES, D. W., BARBER, D. L. A., PRICE, W. L., AND SOLOMONIDES, C. M. *Computer networks and their protocols*, Wiley, New York, 1979.

DAVS76 DAVISSON, L., AND GRAY, R. (Eds.). *Data compression*, Dowden, Hutchinson & Ross, Stroudsburg, Pa., 1976.

DAY80 DAY, J. "Terminal protocols," *IEEE Trans. Commun.* **COM-28** (April 1980), 585–593.

DEPA76 DEPARIS, M., DUENKI, A., GLEN, M., LAWS, J., LEMOLI, G., AND WEAVING, K. "The implementation of an end-to-end protocol by EIN centers: A survey and comparison," in *Proc. 3rd Int. Conf. Computer Communication* (ICCC) (Aug. 1976), pp. 351–360.

DIFF76a DIFFIE, W., AND HELLMAN, M. E. "A critique of the proposed data encryption standard," *Commun. ACM* **19** (March 1976), 164–165.

DIFF76b DIFFIE, W., AND HELLMAN, M. E. "New directions in cryptography," *IEEE Trans. Inf. Theory* **IT-22** (Nov. 1976), 644–654.

DIFF77 DIFFIE, W., AND HELLMAN, M. E. "Exhaustive cryptanalysis of the NBS data encryption standard," *Computer* **10** (June 1977), 74–84.

DOLL78 DOLL, D. R. *Data communications*. Wiley, New York, 1978.

FARB72 FARBER, D. J., AND LARSON, K. C. "The system architecture of the distributed computer system—the communications system," in *Symp. Computer Networks*, Polytechnic Institute of Brooklyn, Brooklyn, N.Y., April 1972.

FOLT80 FOLTS, H. C. "Procedures for circuit-switched service in synchronous public data networks," *IEEE Trans. Commun.* **COM-28** (April 1980), 489–496.

FRAS75 FRASER, A. G. "A virtual channel network," *Datamation* **21** (Feb. 1975), 51–56.

FRAT73 FRATTA, L., GERLA, M., AND KLEINROCK, L. "The flow deviation method: An approach to store-and-forward communication networks," *Networks* 3 (1973), 97–133.

FREE80 FREEMAN, H. A., AND THURBER, K. J. "Updated bibliography on local computer networks," *Comput. Arch. News* 8 (April 1980), 20–28.

GELE78 GELENBE, E., LABETOULLE, J., AND PUJOLLE, G. "Performance evaluation of the HDLC protocol," *Comput. Networks* 2 (Sept.–Oct. 1978), 409–415.

GERL77 GERLA, M., AND KLEINROCK, L. "Closed loop stability controls for S-ALOHA satellite communications," in *Proc. 5th Data Communication Symp.* (ACM/IEEE), (1977), pp. 2-10–2-19.

GIEN78 GIEN, M. "A file transfer protocol (FTP)," *Computer Networks* 2 (Sept.–Oct. 1978), 312–319.

GUNT81 GUNTHER, K. D. "Prevention of deadlocks in packet-switched data transport systems," *IEEE Trans. Commun.* COM-29 (April 1981), 512–524.

HELL79 HELLMAN, M. E. "DES will be totally insecure within ten years," *IEEE Spectrum* 16 (July 1979), 32–39.

HELL80 HELLMAN, M. E. "A cryptanalytic time–memory tradeoff," *IEEE Trans. Inf. Theory* IT-26 (July 1980), 401–406.

INWG78 "A proposal of an Internetwork end to end protocol," in *Proc. Symp. Computer Network Protocols*, University of Liege, Belgium, Feb. 1978, pp. H:5–25.

IRLA78 IRLAND, M. I. "Buffer management in a packet switch," *IEEE Trans. Commun.* COM-26 (March 1978), 328–337.

KAMO81 KAMOUN, F. "A drop and throttle flow control policy for computer networks," *IEEE Trans. Commun.* COM-29 (April 1981), 444–452.

KLEI78 KLEINROCK, L., AND YEMINI, Y. "An optimal adaptive scheme for multiple access broadcast communication," in *Proc. ICC* (IEEE), 1978, pp. 7.2.1–7.2.5.

KUO81 KUO, F. F. (Ed.) *Protocols and techniques for data communication networks*, Prentice-Hall, Englewood Cliffs, N.J., 1981.

LIU78 LIU, M. T. "Distributed loop computer networks," in *Advances in Computers*, M. C. Yovits (Ed.), Academic Press, New York, 1978, pp. 163–221.

MANN78 MANNING, E. G. "On datagram service in public packet-switched networks," *Comput. Networks* 2 (May 1978), 79–83.

MART78 MARTIN, J. *Communications satellite systems*, Prentice-Hall, Englewood Cliffs, N.J., 1978.

MCQU74 MCQUILLAN, J. M. "Adaptive routing algorithms for distributed computer networks," Ph.D. dissertation, Div. Engineering and Applied Sciences, Harvard Univ., 1974.

MCQU77 MCQUILLAN, J. M., AND WALDEN, D. C. "The ARPA network design decisions," *Comput. Networks* 1 (Aug. 1977), 243–289.

MERK78a MERKLE, R. C. "Secure communications over insecure channels," *Commun. ACM* 21 (April 1978), 294–299.

MERK78b MERKLE, R. C., AND HELLMAN, M. E. "Hiding information and receipts in trapdoor knapsacks," *IEEE Trans. Inf. Theory* IT-24 (Sept. 1978), 525–530.

MERL80a MERLIN, P. M., AND SCHWEITZER, P. J. "Deadlock avoidance in store-and-forward networks—I: Store-and-forward deadlock," *IEEE Trans. Commun.* COM-28 (March 1980), 345–354.

MERL80b MERLIN, P. M., AND SCHWEITZER, P. J. "Deadlock avoidance in store-and-forward networks—II: Other deadlock types," *IEEE Trans. Commun.* COM-28 (March 1980), 355–360.

METC76 METCALFE, R. M., AND BOGGS, D. R. "Ethernet: Distributed packet switching for local computer networks," *Commun. ACM* 19 (July 1976), 395–404.

NEED78 NEEDHAM, R. M., AND SCHROEDER, M. D. "Using encryption for authentication in large networks of computers," *Commun. ACM* 21 (Dec. 1978), 993–999.

NEED79 NEEDHAM, R.M. "System aspects of the Cambridge ring," in *Proc. 7th Symp. Operating Systems, Principles* (ACM), 1979, pp. 82–85.

PETE61 PETERSON, W. W., AND BROWN, D. T. "Cyclic codes for error detection," *Proc. IRE* 49 (Jan. 1961), 228–235.

POPE79 POPEK, G. J., AND KLINE, C. S. "Encryption and secure computer networks," *Comput. Surveys* 11 (Dec. 1979), 331–356.

POST80 POSTEL, J. B. "Internetwork protocol approaches," *IEEE Trans. Commun.* COM-28 (April 1980), 604–611.

RIVE78 RIVEST, R. L., SHAMIR, A., AND ADLEMAN, L. "A method for obtaining digital signatures and public key cryptosystems," *Commun. ACM* 21 (Feb. 1978), 120–126.

ROBE73 ROBERTS, L. G. "Dynamic allocation of satellite capacity through packet reservation," in *1973 Nat. Computer Conf.*, AFIPS Press, Arlington, Va., pp. 711–716.

ROTH77 ROTHAUSER, E. H., AND WILD, D. "MLMA—a collision-free multi-access method," *Proc. IFIP Congr. 77*, (IFIP) (1977), 431–436.

RUDI76 RUDIN, H. "On routing and delta routing: A taxonomy and performance comparison of techniques for packet-switched networks" *IEEE Trans. Commun.* COM-24 (Jan. 1976), 43–59.

RYBC80 RYBCZYNSKI, A. "X.25 interface and end-to-end virtual circuit service characteristics," *IEEE Trans. Commun.* COM-28 (April 1980), 500–510.

SALT78 SALTZER, J. H. "On digital signatures," *Oper. Syst. Rev.* 12 (April 1978), 12–14.

SCHI78 SCHICKER, P., AND DUENKI, A. "The vir-

tual terminal definition," *Comput. Networks* **2** (Dec. 1978), 429–441.

SCHO76 SCHOLL, M. "Multiplexing techniques for data transmission over packet switched radio systems," Ph.D. dissertation, Computer Science Dep., UCLA, 1976.

SCHW80 SCHWARTZ, M., AND STERN, T. E. "Routing techniques used in computer communication networks," *IEEE Trans. Commun.* **COM-28** (April 1980), 539–552.

SEGA81 SEGALL, A. "Advances in verifiable fail-safe routing procedures," *IEEE Trans. Commun.* **COM-29** (April 1981), 491–497.

SHAM79 SHAMIR, A. "How to share a secret," *Commun. ACM* **22** (Nov. 1979), 612–613.

SHAM80 SHAMIR, A., AND ZIPPEL, R. "On the security of the Merkle-Hellman cryptographic scheme," *IEEE Trans. Inf. Theory* **IT-26** (May 1980), 339–340.

SHAP77 SHAPLEY, D., AND KOLATA, G. B. "Cryptology: Scientists puzzle over threat to open research, publication," *Science* **197** (Sept. 30, 1977), 1345–1349.

SHOC81 SHOCH, J. "An annotated bibliography on local compuer networks," Xerox Tech. Rep., Xerox PARC, April 1980.

SLOA75 SLOANE, N. J. A. *A short course on error correcting codes,* Springer-Verlag, Berlin and New York, 1975.

SUGA79 SUGARMAN, R. M. "On foiling computer crime," *IEEE Spectrum* **16** (July 1979), 31–32.

SUNS78 SUNSHINE, C. A., AND DALAL, Y. K. "Connection management in transport protocols," *Comput. Networks* **2** (Dec. 1978), 454–473.

TANE81 TANENBAUM, A. S. *Computer networks,* Prentice-Hall, Englewood Cliffs, N.J., 1981.

TOML75 TOMLINSON, R. S. "Selecting sequence numbers," in *Proc. ACM SIGCOMM/SIGOPS Interprocess Communication Workshop* (ACM) (1975), pp. 11–23.

WILK79 WILKES, M. V., AND WHEELER, D. J. "The Cambridge digital communication ring," in *Proc. Local Area Communication Network Symp.*, Mitre Corp and NBS (1979), pp. 47–61.

YEMI79 YEMINI, Y., AND COHEN, D. "Some issues in distributed process communication," in *Proc. 1st Int. Conf. Distributed Computer Systems* (IEEE), 1979, pp. 199–203.

ZIMM80 ZIMMERMANN, H. "OSI reference model—the ISO model of architecture for open systems interconnection," *IEEE Trans. Commun.* **COM-28** (April 1980), 425–432.

Received January 1981; final revision accepted September 1981.

A Layered Network Protocol for Packet Voice and Data Integration

WILLIAM L. HOBERECHT

Reprinted from *IEEE Journal on Selected Areas in Communications*, Vol. SAC-1, No. 6, December 1983, pages 1006-1013. Copyright © 1983 by The Institute of Electrical and Electronics Engineers, Inc. All rights reserved.

Abstract—**This paper discusses considerations in the design of packet protocols suitable for interactive voice and interactive data communication, and then outlines a potential layered protocol architecture for the internal communication of a long haul network that might support packet voice and packet data transport.**

Following the protocol description, the paper compares potential delays for two voice/data packet network architectures: one using only link retransmission, the other using only edge retransmission for data (as included by the voice/data protocol). The underlying data traffic loads offered to the network are the same for the two methods, although they give rise to different traffic patterns. This preliminary analysis shows that the average delay using an edge-to-edge recovery discipline can be made comparable to the delays introduced with a link-by-link recovery discipline, if the network uses high speed transmission facilities (e.g., over 1 Mbits/s) having good error characteristics (e.g., one or less packets corrupted in each 1000), and sends up to 128 bytes of customer data as a single packet.

I. Introduction

THE similarity of packet voice and packet data transport, recognized by many [1]–[4], suggests further study in the area of designing an integrated voice/data network. Unfortunately, along with these similarities are several conflicting communication requirements. Data applications demand essentially error-free transmission and existing protocol standards (e.g., CCITT Recommendation X.25) reflect this need. On the other hand, voice is tolerant of occasional errors. The inherent characteristics of voice allow small amounts of lost or corrupted information to be reconstructed, or even omitted, without a severe degradation in voice quality. Voice, however does have more stringent requirements on the amount of delay permissible between the time an utterance is spoken and the time it is subsequently heard by the receiver. While the exact limits on the amount of acceptable delay are unknown, subjective tests have shown that excessive delays may not be tolerated by telephone users when nondelayed phone service is available [5], [6] (cost of service was not a parameter in these tests).

One recent study [7] addressed the issue of voice/data packet protocols and described a point-to-point protocol meeting the needs of both voice and data. Some earlier studies in the area of protocols for packetized voice [8], [9] have concentrated on end-to-end aspects of the protocol (i.e., peer communication between packet voice transmitters and receivers) and on using current packet data networks for packet voice transport. These efforts included a demonstration of packet voice over Arpanet, reported in 1978 [10], suggesting that it may also be possible to use existing packet data networks (or at least their protocol architecture) for packet voice. While this earlier work considered use of current network designs for packet voice transmission, this paper presents an alternative view of a layered protocol structure for the *internal* operation of a long haul packet network that can support packet voice and packet data transport. These discussions and the description of the protocol architecture represent preliminary thinking on concepts that might be incorporated into an integrated packet voice/data system. The intent is to stimulate further thought on possible protocol architectures for voice/data packet transport.

The remainder of this paper describes packet voice communication and a possible internal network packet voice/data protocol structure. The paper first presents some of the factors that have a pronounced effect on the protocol design and the placement of protocol functions, and then describes a layered protocol for the internal communication of a voice/data packet network. The final section presents an analysis of delay for data using the voice/data packet protocol and compares it to delay encountered using a traditional error recovery protocol.

Throughout the paper, a distinction is made between edge-to-edge protocols (completely contained within the network, exercised between packet network interfaces) and end-to-end protocols (exercised between network customers).

II. Packet Voice/Packet Data Network Components

Fig. 1 shows the components of a generic packet network:

1) Packet network interfaces (PNI) through which users (i.e., subscribers and other networks) connect to the network.

2) Interswitch links that carry voice, data, and signaling information.

3) Packet switches (PS) that connect to both packet network interfaces and other packet switches with high-speed lines.

Manuscript received April 1, 1983; revised July 1, 1983. A portion of the paper was presented at GLOBECOM '83, San Diego, CA, November 28–December 1, 1983.

The author is with Bell Laboratories, Naperville, IL 60566.

EH0301-2/91/0000/0042/$01.00 © 1983 IEEE

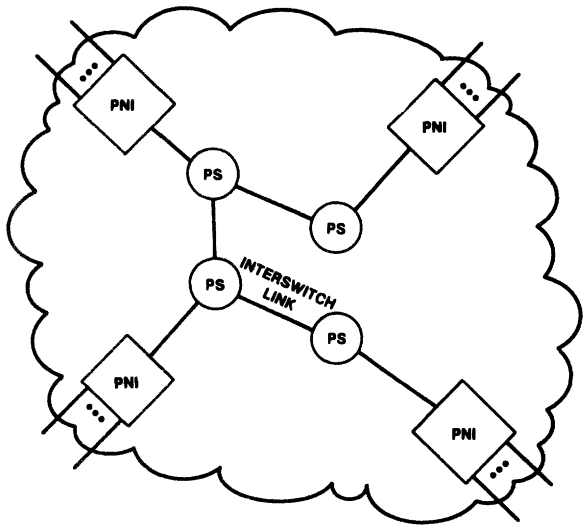


Fig. 1. Packet network components.

III. Factors Influencing the Protocol Design

Several factors, all interrelated, have a pronounced effect on the protocol design. The primary underlying philosophy behind the design of a voice/data packet network and protocol is to consider these factors and design the network to take advantage of them. Three factors have been identified that significantly affect the packet voice/data protocol: characteristics of the applications using the network, performance requirements, and transmission facilities.

A. Characteristics of the Application

The transport of packetized voice is a major application of an integrated packet network. Because of the redundancy in this voice information, it may not require the robust error correction methods commonly found in packet protocols. Exercising conventional flow control on packets containing voice information seems inappropriate. Due to delays that can be introduced through the application of flow control procedures, voice information sent through the network may be rendered unusable. Degrading the level of service for established calls, a characteristic of such flow control procedures, is an undesirable practice in a voice network. Additionally, flow control practices have been developed to allow communication between devices operating at different rates. Because voice coding and decoding equipment would be operating at the same rate, such flow control practices are not necessary. This does not preclude the need for regulating the flow of information into the network to protect network resources.

B. Performance Considerations

The design objectives for a packet voice network place several constraints on the operation of the network. Two categories are particularly influential in the protocol design: loss performance (lost data/voice packets, lost signaling packets) and delay objectives [11].

Delay objectives for the network can be met, in part, through the elimination of *link* error recovery techniques that require retransmission of corrupted information. This has a twofold effect: first, one source of variable delays introduced by packet networks is eliminated, and second, network transmission procedures are simplified.

C. Quality and Speed of Transmission Facilities

The exclusive use of high-quality, high-speed (e.g., over 1 Mbit/s) digital transmission facilities in a packet voice network [1] encourages examination of nontraditional error correction techniques. Current methods, aimed at ensuring error-free information delivery, typically correct errors on each link. While this complexity and expense (in elapsed time, processor resources required, memory, and buffer utilization) can be justified for slow transmission facilities plagued with "high" bit error rates, the supporting reasons become less clear for the high-speed transmission facilities that would be used in a packet voice network.

The performance of the interswitch digital transmission facilities can be expected to provide sufficiently reliable performance to allow the responsibility of error correction to be moved to the packet network interfaces. As a higher level function, the error correction required for data communication would not be performed on each network link, but once across the network. This simplifies processing that must be done by network switches without degrading the performance seen by network users.

IV. Network Access Arrangements

The placement of functions (e.g., voice encoding) in a packet voice/data network is specified by the protocol architecture. For the generic packet voice/data network considered in this paper, two alternatives are possible for voice coder placement. In the first, the interface to the network is a circuit mode interface. The voice stream into the network is a deterministic stream of information. At the PNI, this stream is encoded, placed in packets, and transported across the network. At the distant edge of the network, the packet stream that was carried through the network is converted back into a circuit voice stream. The data interface to the network is packet mode, possibly using the X.25 protocol.

The alternative method uses a packet mode interface to the network for both voice and data traffic. Voice encoding and packetizing functions are performed by customer equipment, before information enters the network. The network does not process (e.g., code and packetize) the voice information, as it did in the first case, but only transports packets between network customers. As with the first type of interface, two grades of network transport are required: voice transport (fast, but errors are permitted) and data transport (error-free service is required). Other grades of network transport may be desirable, but this paper limits discussion to only these two possibilities.

The remainder of this paper focuses on a circuit-mode

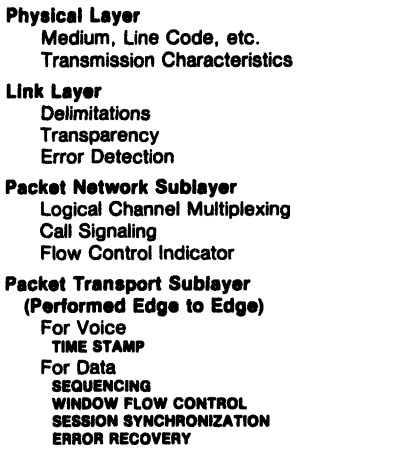


Fig. 2. Protocol functions.

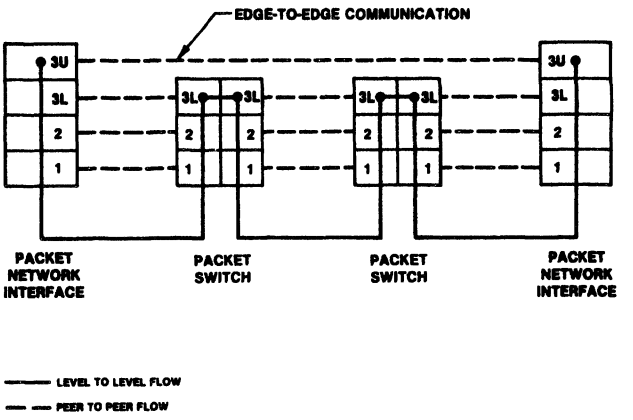


Fig. 3. Internal network protocol layers.

voice interface to the network and a packet-mode data interface. Under this architecture, the packet network interface performs voice encoding and packetizing at the edges of the network. Although this discussion concentrates on internal network operation and functions, these concepts can also be extended, migrating functions from the network to customer premises equipment.

V. A Voice/Data Packet Protocol

Interswitch communication in a voice/data packet network is according to the network transport protocol. Access to the network is not under the protocol specified for interoffice transmission.

Following the spirit of the draft open systems interconnection (OSI) model that the International Organization for Standardization is formulating [12], this partitioning of functions and rules is specified as a hierarchy of layers, each performing specific services, such as error detection. The packet transport protocol outlined here has been designed using the general principles that have been guiding the development of the OSI model. This voice/data packet protocol supports only transport functions (i.e., the transport of packets through the network); the higher level functions (end-to-end) discussed in the OSI model appear as higher layers above this basic transport capability and are not discussed here.

The protocol structure provides the functions necessary for flow control of data within the network and ensures reliable data transmission through the network. Under this structure, the network can be a reliable data-transfer medium. The basic transport mechanism is provided by the lower three levels of the protocol (1-, 2-, and 3-lower) applied to transmissions across each network link. The packet transport sublayer (level 3-upper), employed edge-to-edge within the network between packet network interfaces, provides functions specific to the application (Figs. 2 and 3). In this scheme, the lowest three levels provide packet transmission and routing through the network, but allow the possibility of packet corruption and packet loss. At the edges of the network, any detected errors are corrected with a peer communication between the entry and exit packet network interfaces, thus providing reliable data communication through the network. This protocol structure allows the same link communication protocols to be used for voice and data, and employs an edge-to-edge protocol that is tailored to the specific service used by the network customer. A later section discusses delays experienced by data using this edge-to-edge recovery strategy.

Controlling the flow of information into the network can be accomplished with several mechanisms: call blocking if insufficient network resources are available at the time the call is requested; packet discarding during short periods of congestion; call clearing by the network to face occasional overloads caused by failures of network components (links, switches, etc.); window flow control for data; and other methods.

The following sections present a closer view of the each protocol level.

A. Level 1 — Physical Layer

The physical level of the protocol specifies the electrical characteristics and representation of transmitted bits. Additionally, bit time synchronization and performance characteristics are included in this level. Level 1 is, in practice, similar to the level 1 of conventional packet data protocols and is not discussed further here.

B. Level 2 — Link Layer

The link layer performs several functions necessary for successful transmission between network switches. These functions include frame delimiting, error detection, and bit pattern transparency. As a major departure from HDLC, error recovery procedures are not included in this level. Functions such as link set up, link disconnection, and link resynchronization procedures are not included in the packet voice/data level 2 protocol because no link state information is maintained at level 2.

The link layer protocol procedures for the packet voice network do not include provisions for error recovery or for flow control. When an error is detected in a frame, the corrupted frame is immediately discarded by the switch. No further actions are taken at level 2. These measures are aimed at reducing sources of network introduced packet delay.

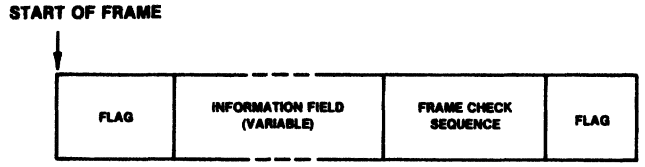


Fig. 4. Level 2 frame format.

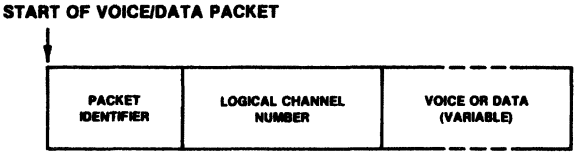


Fig. 5. Level 3-lower packet format.

Each level 2 frame carries a data, voice, or signaling message. The format for information frames (Fig. 4) includes a leading flag that delimits the start of an information frame, a variable-length information field, a frame check sequence for detecting frame errors, and a trailing flag that delimits the end of the information frame. A two byte frame check sequence could probably provide adequate error detection capability.

C. Level 3L — Packet Network Sublayer

Level three of the voice/data protocol has two sublayers: level 3-lower (3L) and level 3-upper (3U). The packet provides the unit of information transfer at the packet network sublayer of this protocol. The primary function provided here is the routing of packets along a path that is established at call-setup time. This path is fixed for the duration of the call.

Two types of packets are included in the level 3L portion of the network protocol: signaling packets and data/voice packets. Signaling packets are used in call establishment, call disconnection, call administration, and link maintenance. These signaling packets are exchanged between adjacent switches (link signaling packets) and between packet network interfaces in the call path (edge signaling packets). Data/voice packets carry information being sent between level 3U entities. Fig. 5 shows a possible format for a voice/data packet. Information contained in the level 3L header includes three fields: a packet identifier field indicating the type of packet and processing required by the switch, a logical channel number field that uniquely identifies a call on a link, and a variable-length voice/data field containing higher level information. Signaling packets would have similar format, with a signaling message indicator and additional parameters carried in the variable-length voice/data field.

D. Level 3U — Packet Transport Sublayers

Multiple types of sublevel 3U allow the network to support different classes of packet transport. The specific messages exchanged and actions taken at level 3U depend on the application being supported by the network (e.g., packet voice transmission and interactive data exchange).

For packet voice communication, level 3U functions are performed at the voice coder location (in the packet network interface). Similarly for data exchange, packet transport sublayer functions are also carried out between packet network interfaces (as an edge-to-edge protocol). However, the concepts presented here do not prelude alternative architectures that place these functions external to the network.

There are two distinct sets of level 3U functions: those for reliable data communication and those for voice communication.

1) Packet Transport Sublayer for Packetized Voice: For packetized voice only, level 3U messages provide synchronization of voice information. Voice that has been encoded and packetized must be time identified, in some manner, relative to either an absolute time reference or to other voice packets (both preceding and subsequent) to allow a smooth play back of voice at the receiver. This level 3U function provides a means of playing out, at the correct time, received voice information. Such a service is necessary because of the variable delays that can be encountered in a long haul packet network. Voice messages are never retransmitted with this protocol architecture.

A time stamp placed in the level 3U header could identify, relative to other voice messages for the call, the time at which the voice message was generated. The information contained in this field would be generated and used only by the two packet network interfaces in the call path. Tandem switching equipment does not include level 3U processing, so this field would not be modified by intermediate switches. Other time stamping schemes are possible [13].

2) Packet Transport Sublayer for Reliable Data Communication: For data communication only, level 3U includes the functions necessary to provide reliable, error-free data transmission. These functions include synchronization of level 3U entities, flow control, and detection and recovery of lost messages. Protocol standards, such as HDLC and CCITT Recommendation X.25, have evolved with the intention of providing reliable communication between devices that do not necessarily process data at the same rate. The packet voice/data protocol supports these functions at level 3U; the packet transport sublayer for data assures error free, sequential message delivery and includes provisions for flow control.

A possible message format for the packet transport sublayer, shown in Fig. 6, includes a message format (MF) field consisting of subfields that can indicate the logical relationship of sequential messages, use of normal or extended-sequence numbering, and other functions necessary to support the access protocol for data; a control field containing flow control and check pointing information; and a variable-length data field.

The set of supervisory messages includes messages necessary to establish and synchronize a level 3U data connection, reliably transfer data across the network, and clear the connection. Level 3U procedures might be thought of as a modified (and enhanced) set of HDLC control procedures exercised across the network (rather than across a single link).

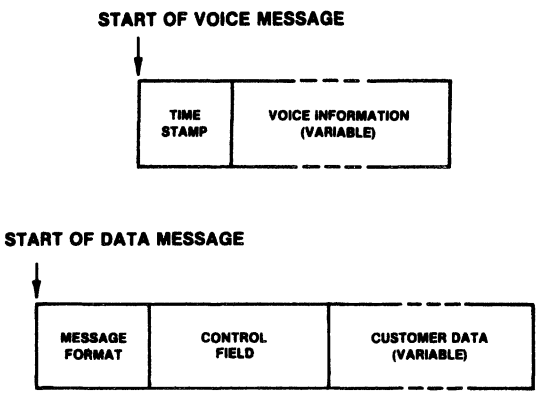


Fig. 6. Level 3-upper message formats.

VI. Delay With Retransmissions for Corrupted Data

This section examines cross-network delay using two different error recovery disciplines. The first uses retransmission between adjacent network switches to recover corrupted data; the interswitch protocol guarantees reliable transmission across each network link. Average cross-network data transmission delays for this protocol are compared to the delays encountered under a second protocol. In this second protocol, lost or corrupted data are recovered by edge-to-edge retransmission procedures between the sending and receiving packet network interfaces (this is the method used in the voice/data packet protocol previously described); the interswitch protocol makes no attempt to recover corrupted data.

A. Link Recovery Model

This section describes a model for analyzing performance using a reliable link protocol to accomplish error-free transport across a network, with a concatenation of reliable links forming the error-free path.

A frame sent across a link may be lost with probability p, which is determined by the error characteristics of the transmission facilities and the size of the frame sent across the link. If this happens, the link is effectively out of service until the sender "pulls back" and retransmits the corrupted frame. The *cycle time*, T_C, is the elapsed time before the sender realizes that it must retransmit the corrupted frame. The cycle time has three components: 1) the time for the frame to cross the link, 2) the interframe arrival time (i.e., the time between reception of the corrupted frame and reception of the next error-free frame), and 3) the time for the reject to be returned to the sender. During this period, the receiver discards all frames it receives. To frames awaiting transmission, the link appears to be transmitting a single, long frame (called an *Xframe* in this paper) during the cycle time. Note that this *X*frame is not an actual frame transmitted on the link, but serves to describe the amount of time the link is unavailable because of error recovery procedures.

As a preliminary effort to gauge the effect of the link unavailability during the cycle time, the link is modeled as an $M/G/1$ queue. For this model, link traffic has two components: frames containing useful data (which also contain embedded data acknowledgments for traffic in the reverse direction), and *X*frames. Using an exponential service time distribution for user data and a deterministic service time for *X*frames (this service time is exactly equal to the cycle time) completely describes all traffic on the link for the model.

Using an $M/G/1$ queue to approximate the average transmission time per link for data (T_L), a weighted first and second moment ($\bar{x}$ and μ_2) describes characteristics of the link traffic. D_{prop} is the propagation delay per link and ρ_L is the total utilization of the link; the utilization of the link attributed to data traffic (ρ_{data}) is slightly lower than ρ_L.

$$T_L = (D_{\text{prop}}) + \bar{x} + \frac{1}{2}\frac{\rho_L}{2(1-\rho_L)}\frac{\mu_2}{\bar{x}}.$$

The average cross-network delay for a path of N_{links} links, for frames containing data, is

$$\text{Average Network Delay} = (N_{\text{Links}})(T_L). \qquad (1)$$

This delay equation will be used later in comparing edge-to-edge and link recovery strategies.

While the effect of *X*frames can be negligible at low link speeds, it can become significant at higher link speeds (e.g., over 1 Mbits/s) because of the increasing size of the *X*frame and the number of frames that must be retransmitted. For example, at 75 percent, total link utilization on a 1.5 Mbits/s link, each frame error causes the link to be unavailable for useful data transmission for over 30 frame transmission times, creating a delay for subsequent frames. The probability of the link corrupting a frame closely describes the amount of link traffic attributed *X*frames. For the model, every time a frame is corrupted this *X*frame appears on the link. Because a frame may be retransmitted several times (additional retransmission is necessary if the frame becomes corrupted when retransmitted), an attempt to transmit a useful data carrying frame across a link causes $p/1-p$ *X*frames to appear on a link. This information is used in determining the probabilities of normal frames and *X*frames being present on a link, which in turn is used to find the weighted $\bar{x}$ and μ_2.

B. Edge Recovery Model

This section describes a model for analyzing performance in a network that uses a nonguaranteed-delivery link protocol, supplemented with a reliable edge-to-edge protocol across the network for error-free packet transport. With an edge-to-edge strategy, error recovery is between the corresponding edges of a call, and is done on a per call basis. In this section, *frames* are sent between adjacent switches on a link. Corrupted frames are simply discarded when a switch detects an error. *Messages* are sent between corresponding edges of a call. If a message is lost in traversing the network, it is retransmitted again from the sending edge.

The probability of having to invoke error recovery procedures is based entirely on the probability of successfully crossing the network. We find this using the probability p of a frame being corrupted on a link. From this the

probability P of, anywhere in the network path, corrupting a frame containing this message is

$$P = 1 - (1 - p)^{N_{\text{links}}} \tag{2}$$

Each link carries a mix of three types of traffic (as contrasted to the link model, in which each link carried only two types of traffic: information frames, which carry customer data, and Xframes). The first type is data traffic, which has the same exponential distribution that is used for the link scenario. To allow a fair comparison with the link recovery model, the amount of traffic attributed to user data, ρ_{data}, is identical for the two models. For the edge recovery model, every data message has an explicit acknowledgment, which has a deterministic service time distribution. This is the second type of traffic. Discarded message traffic enters in as the third factor contributing to link utilization. Discarded messages that the sender must retransmit have exponentially distributed service times. The average service time for all traffic on a link is a weighted average of these three service times. The total link utilization for this model, ρ_E, reflects contributions from each of these three generators of link traffic.

In a call between transmitting edge T and receiving edge R, the delay between the occurrence of an error and discovery of this error by T has three components. These are analogous to the three outlined for link recovery: intermessage arrival time (the time between the moment the corrupted message would have arrived and the time the next uncorrupted message arrives), successful cross-network transit of a data message, and successful cross-network transit of the reject message in the reverse direction.

For a single call, receiver R discards $N_{\text{discarded}} = (T_C)(H) + 1$ messages if the sender continues transmitting messages after the error occurred. H is the user throughput class expressed in the message per second and T_C is the edge-to-edge cycle time (as defined in the link model discussion). Because of retransmissions, the event of corrupting a message can happen more than once in trying to get a message through the network. Each message sent through the network results in $(P/(1-P))N_{\text{discarded}}$ extra messages. These extra messages directly increase the queueing delays experienced packets for other calls in the network.

Recall the definitions for ρ_L and ρ_E. ρ_L is the utilization of each network link in a network using link recovery. ρ_E is the utilization of these links if the network uses edge-to-edge recovery. In general, for a fixed data traffic load (i.e., fixed ρ_D), ρ_E is greater than ρ_L. This is for two reasons: the explicit data acknowledgments under the edge-to-edge protocol (that are not present in the link scenario) and the comparative error rates (2).

1) Cross-Network Delay (Including Effects of Retransmission Traffic): Computing the average transmission time across a link for data (T_{data}), including propagation delay and system time,

$$T_{\text{data}} = (D_{\text{prop}}) + \frac{1}{\mu_{\text{data}}} + \frac{1}{2}\frac{\rho_E}{(1-\rho_E)}\frac{\mu_2}{\bar{x}}.$$

In this equation, $1/\mu_{\text{data}}$ is the service time for frames carrying user data; and the first and second moments (μ_2 and $\bar{x}$) are found using the service time distributions of the three types of link traffic along with their probabilities of occurrence.

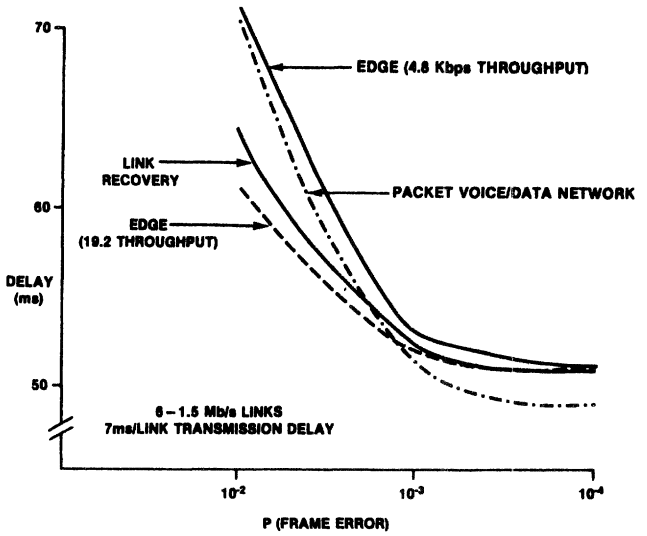


Fig. 7. Average cross-network data delay.

The total average delay incurred in successfully traversing the network one time (without error recovery procedures) for data is $D_{\text{data}} = (N_{\text{links}})(T_{\text{data}})$.

The average intermessage arrival time (I_M) is based entirely on the user transmission speed and amount of data that can be transported in a frame, causing an increase in error detection time at the receiving edge as user throughput class decreases.

For edge-to-edge error recovery, the equation for cross-network delay, including the effects of retransmission procedures, is

$$\text{average cross network delay} = \frac{(1+P)D_{\text{data}} + PI_M}{(1-P)}. \tag{3}$$

This delay equation will be used later in comparing edge-to-edge and link recovery strategies.

C. Delay Comparison

Fig. 7 shows comparative delays for the two recovery disciplines in two types of networks, found using (1) and (3). These delays are for a cross network path of 6 1.5 Mbits/s links, a 7 ms/link propagation and processing delay, uncorrelated frame errors, and frames that contain 128 bytes of customer data not including the frame header. This analysis is not all encompassing; the intent here is to show the effect of Xframes when high-speed facilities are used and show the effect of user throughput on average data delay when using the edge-to-edge recovery mechanism.

In Fig. 7 the amount of data traffic carried is the same for the link and edge recovery curves, ensuring equivalent traffic conditions for the comparison. However, this does not imply that the total link utilization is the same for the two models. (Less data traffic is offered to the voice/data packet network because that network must also carry voice packets.) The method used to generate the link utilizations forces the total link utilization for the link recovery method

to be constant, independent of the probability of corrupting a frame. This means that as the probability of corrupting a frame decreases (moving from left to right in the figure), the ratio of useful data traffic to *X*frames increases, and the amount of data traffic increases. For the two edge recovery curves, the amount of useful data traffic increases to stay the same as for the link curve.

Fig. 7 shows that the delays using link recovery are close to the delays encountered using edge-to-edge recovery under the same data load. Not unexpectedly, as the probability of corrupting a frame decreases, the average delay decreases, because fewer retransmissions are required for either method. In the link case, the *X*frames caused by the retransmission procedures are large and could add significant delays when they appear on a link. With fewer retransmissions (i.e., fewer *X*frames), the total link utilization decreases, diminishing the magnitude of the queueing delays. For the edge-to-edge curves, a major component of cross-network delay is the time to notify the sender that a message must be retransmitted. With this occurring less frequently, the average cross network delay decreases. Additionally, less "extra" traffic (that must be retransmitted) is now appearing on the link, decreasing the queueing delays. However, this is a minor effect for low user throughput classes.

As user throughput increases, two factors affect the amount of "extra" messages present on the link for the edge-to-edge case. First, the number of these extra messages sent during the cycle time (i.e., time to notify the sender that an error has occurred) increases simply because more messages can be sent per second. Along with this, the intermessage arrival time decreases, causing the cycle time to decrease. The net result is that as throughput increases, the number of superfluous messages increases, and for a fixed data load this slightly increases the queueing delays. However, the decrease in time to detect the error significantly decreases the delay introduced by the retransmission protocol. Fig. 7 shows this in the two edge curves, demonstrating that the faster user throughput (19.2 kbits/s) has a lower average delay than the 4.8 kbits/s user throughput. (Note that for user throughputs well over 100 kbits/s, these superfluous messages can cause the link utilization to exceed available bandwidth if the offered data load is held constant. Factors such as link speed and error rate determine the actual customer bandwidth at which this occurs.)

The curve for the voice/data packet network shows average data cross-network delays for an integrated packet network that uses edge-to-edge recovery for data. Although the total link utilizations for this curve are comparable to the other curves, the amount of data traffic is less for this curve because voice traffic is also carried. Corrupted data introduces some extra traffic that must be retransmitted, but corrupted voice messages do not cause any additional network traffic. Because of this, the amount "extra" link traffic caused by retransmission procedures is not as great as it is for the two data-only networks. For this example, frames containing voice messages have a length of 250 bits with a deterministic distribution and outnumber useful data messages by 4 to 1. This ratio does not include discarded messages or acknowledgment messages.

D. Extending the Models

While this paper shows several concerns in finding delay and throughput for the two recovery methods, more can be done in this area. Possible efforts aimed at continuing this study include modifying the model to more accurately describe the probability of corrupting a frame. Additionally, including the effects of corrupted acknowledgment and reject packets will give a more accurate description of network traffic and cross network delay. An enhanced model can include provisions for various message length distributions, increasing the usefulness of the results.

E. Conclusions for the Delay Models

For link correction, two factors influence the effectiveness of the network: 1) unless the network uses selective reject, a frame error *prevents* the sending party on a link from transmitting useful frames for a short period, increasing the cross network delay for subsequent frames; 2) the amount of time required to detect and correct a frame transmission error depends on the link speed and utilization.

When using edge-edge recovery, network protocols are characterized as follows. First, when a message is lost, this prevents useful messages from being sent *for a single call* until the sender retransmits the corrupted message. This causes extra link traffic, increasing queueing delays for all calls on that link. Seoond, the amount of time it takes to correct for a message error is dependent on the user throughput class. (This does not include errors that are detected by the expiration of a timer.)

In a packet network that sends 128 byte messages as a single packet through the network, link recovery and edge-edge recovery methods give comparable performance as seen by a network customer. This conclusion is valid for networks using high speed transmission facilities (e.g., over 1 Mbits/s) having good error characteristics (e.g., corrupted packet rate of 10^{-3}) with reasonable link utilizations for data traffic (e.g., less than 60 percent).

In a combined voice/data packet network, an edge recovery mechanism can provide acceptable delay characteristics for data traffic.

VII. Summary

Several concepts outlining the philosophy of a possible protocol structure for integrated voice, data, and signaling have been presented. The protocol overview covers the rules for communication between adjacent network switches, and across the network between the corresponding packet network interfaces of a call. In this protocol, error correction procedures are not included at the link layer or packet network sublayer for data or voice information. Error recovery for data is performed edge-to-edge, across the network. The average cross-network delays for

data packets under this edge-to-edge error recovery discipline are comparable to the expected delays under a more traditional link-by-link recovery protocol.

While this paper has centered discussion on a possible design for a voice/data *internal* network protocol, the area of accessing such a network with an integrated packet protocol is ripe for study. The concepts outlined in this paper of providing basic transport functions at levels 2 and 3L in the protocol, and tailoring the packet transport sublayer (3U) for the specific application (e.g., packet voice) can be extended to apply to network access protocols.

Acknowledgment

The material presented in this paper has emerged from discussions with members of the Advanced Switching Technology Laboratory at Bell Telephone Laboratories. M. Decina (while at Bell Laboratories on leave from the University of Rome), J. Turner, W. Montgomery, and L. Wyatt have provided their time in discussing packet voice communication, and their help is appreciated.

References

[1] J. W. Forgie and A. G. Nemeth, "An efficient packetized voice/data network using statistical flow control," in *Proc. IEEE Int. Conf. Commun.*, 1977.

[2] G. J. Coviello, O. L. Lake, and G. R. Redinbo, "System design implications of packetized voice," in *Proc. IEEE Int. Conf. Commun.*, 1977.

[3] M. E. Ulug and J. G. Gruber, "Statistical multiplexing of data and encoded voice in a transparent intelligent network," in *Proc. 5th Data Commun. Symp.*, Sept. 1977.

[4] D. Minoli, "Issues in packet voice communication," *Proc. IEEE*, vol. 67, Aug. 1979.

[5] E. T. Klemmer, "Subjective evaluation of transmission delay in telephone conversations," *Bell Syst. Tech. J.*, vol. XLVI, July–Aug. 1967.

[6] ——, "Subjective evaluation of delay and echo suppressors in telephone communications," *Bell Syst. Tech. J.*, vol. XLII, Nov. 1963.

[7] M. Listanti and F. Villani, "An X.25-compatible protocol for packet voice communications," *Comput. Commun.*, pp. 23–31, Feb. 1983.

[8] D. Cohen, "A protocol for packet switching voice communication," in *Proc. Comput. Network Protocols Symp.*, Liege, Belgium, Feb. 1978.

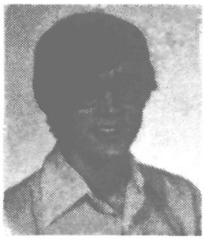
[9] ——, "Issues in transet packetized voice communication," in *Proc. 5th Data Commun. Symp.*, Snowbird, UT, 1977.

[10] S. L. Casner, E. R. Mader, and E. R. Cole, "Some initial measurements of ARPANET packet voice transmission," NTC 1978.

[11] J. G. Gruber, "Delay related issues in integrated voice and data networks," *IEEE Trans. Commun.*, vol. COM-29, June 1981.

[12] H. Zimmermann, "OSI reference model—The ISO model of architecture for open systems interconnection," *IEEE Trans. Commun. (Special Issue on Computer Network Architectures and Protocols)*, vol. COM-28, Apr. 1980.

[13] W. A. Montgomery, "Techniques for packet voice synchronization," *IEEE J. Select. Areas Commun.*, vol. SAC-1, pp. 1022–1028, Dec. 1983.



William L. Hoberecht received the B.S. degree in mathematics from Clarkson College of Technology, Potsdam, NY, in 1979, and the M.S. degree in computer, information, and control engineering from the University of Michigan, Ann Arbor, MI, in 1980.

He is with Bell Telephone Laboratories, Naperville, IL, where his initial assignments were in the areas of packet switching and the design of packet protocols for data and voice communication. He is currently investigating fault-tolerant architectures for advanced packet switching system designs. His technical interests include data communications, packet switching, and operating system design.

An Overview of FDDI: The Fiber Distributed Data Interface

FLOYD E. ROSS

Reprinted from *IEEE Journal on Selected Areas in Communications*, Vol. 7, No. 7, September 1989, pages 1043-1051. Copyright © 1989 by The Institute of Electrical and Electronics Engineers, Inc. All rights reserved.

Abstract—**Fiber Distributed Data Interface (FDDI), which employs an optical fiber medium, is a 100 Mbit/s local area network (LAN) based on a token ring protocol. Updating earlier efforts on this subject by the author and others, this paper presents an overview of this technology and explores the basis for its success. Particular emphasis is placed on the technical specifications for an upwards-compatible version of FDDI, FDDI-II, which adds the capability for circuit-switched services to the packet services of the basic FDDI, thus creating an integrated services LAN.**

I. Introduction

THE Fiber Distributed Data Interface (FDDI) is a 100 Mbit/s local area network (LAN). Using optical fiber as the medium, the FDDI protocol is based on a token ring access method. FDDI is being developed in Accredited Standards Committee (ASC) X3T9—chartered to develop computer input/output (I/O) interface standards. In the late 1970's, ASC X3T9 recognized the need for a new I/O channel standard as an alternative to Federal Information Processing Standards (FIPS) 60-63. By late 1982, work had been started on FDDI in the ASC X3T9.5 technical committee. In mid-1983, two proposals encompassing the Physical layer and the Media Acess Control for FDDI were submitted by Sperry.

Through this same time period, the Institute of Electrical and Electronics Engineers (IEEE) P802 standards project was developing local area network (LAN) standards with data rates up to 20 Mbits/s. FDDI followed the packet data architectural concepts of IEEE P802 and chose the emerging 4 Mbits/s token ring protocol of IEEE P802.5 as the starting point for the FDDI protocol. These choices placed FDDI in an ideal position to be both the backbone network and the follow-on network to the IEEE P802 LAN's.

Meanwhile, in another standards arena, the Open Systems Interconnection (OSI) model had been put in place. This layered the design of computer interconnections, allowing the development of separate standards for the different layers, and provided the proper framework for the development of a set of FDDI standards.

Another factor of significance in the development of FDDI was an increased use of high-performance video workstations for a variety of applications. These brought an emphasis on facilitating low-cost implementations.

Manuscript received May 15, 1988; revised March 1, 1989.
The author is with UNISYS, Malvern, PA 19355.
IEEE Log Number 8929561.

Driven by these forces, FDDI grew from the original Sperry proposals to satisfy the needs of many applications, including the back-end (I/O channel) interface, LAN backbone, and front-end high-performance LAN applications. In the course of its development, each new wave of supporters brought new demands which, in turn, were accommodated by the emerging FDDI standards. As a result, the set of services offered by FDDI is broad enough to allow individual optimization of FDDI networks to satisfy the needs of diverse environments.

One enhancement, FDDI-II, will offer signficantly increased services by integrating circuit-switched data traffic capabilities into what had originally been strictly a packet LAN. The impetus for FDDI-II came from the new generation of digital PBX's of the early 1980's. Their needs were similar to those of many real-time applications including digital voice and video networks as well as sensor and control data streams. All of these disciplines contributed to the emerging FDDI-II design definition which began in late 1984.

A. Outline of Paper

Section II of this paper addresses several issues of overall signficance to FDDI. Section III describes general FDDI characteristics, with the Physical layer, MAC layer, and SMT operation more fully described in Sections IV, V, and VI, respectively. Section VII supplies an overview of FDDI-II concepts, with Section VIII providing details on FDDI-II operation. The status of the FDDI standardiztion effect is covered in Section IX.

II. FDDI as the Solution

The widespread acceptance of FDDI has been due to a number of factors. The IEEE P802 effort provided several medium-speed (1-20 Mbit/s) LAN's. In effect, IEEE P802 popularized the LAN, and by doing so, created a market for a higher speed LAN to perform the backbone function for lower speed IEEE 802 LAN's and to satisfy applications that require a higher performance LAN.

Contributing also to the acceptance of FDDI is the dramatic improvement in the price and performance characteristics of optical fiber and related components such as optical transmitters and receivers. With the many other advantages that the use of optical fiber offers—high data bandwidth, security, safety, immunity to electromagnetic interference, and reduced weight and size—the concept of

EH0301-2/91/0000/0050/$01.00 © 1989 IEEE

an all-fiber LAN was most attractive. Because the FDDI design has been optimized to the use of optical fiber since its inception, FDDI is now in a leadership role in the development of optical fiber LAN technology.

FDDI has maximized the value of standardization. The true value of standards is being recognized in all marketplaces—from the individual customer to multinational corporations and the federal government. With emphasis firmly on standards, FDDI has conformed to the ISO Model. The rich functionality that has been integrated into FDDI to meet the needs of a number of market segments means that it can be the one standard satisfying the requirements of the broad, high-speed LAN marketplace. The value of one high-speed optical fiber LAN standard is immense.

A. Advantages of a Ring Design

No paper on FDDI would seem complete without at least a cursory summary of the advantages that a ring design offers. A ring can be shown to offer superior reliability, availability, and serviceability, even in the face of physical damage to the network. A ring topology can be designed to be capable of continued operation despite any projected failure.

Other advantages include the interconnect simplicity of the physical hardware at the interface level. The point-to-point connections around the ring not only provide an easy focus of standardization, but also allow different ring links to have different characteristics and optimization points. Optical fiber, which does not adapt well to bus configurations, can be easily accommodated. Optical fiber has sufficient bandwidth that bit-serial transmission may be used, thus significantly reducing the size, cost, and complexity of the hardware required by a network.

Ring topologies offer advantages in the ease of initial configuration and reconfiguration as the network requirements change. Failing stations or fiber links can be isolated through the use of appropriate protocols. These protocols also provide for the logical addition and deletion of stations without detrimental effects on existing ring traffic. Actual physical addition or removal of stations from the network is also facilitated because ring initialization, failure isolation, recovery, and reconfiguration mechanisms can provide for continued operation even while the cables are being rearranged.

Ring topologies inherently impose no restrictive logical limit on the length of ring links, the number of stations, or the total extent of the network that can be accommodated.

Ring topologies, and the protocols supported by them, offer significant performance advantages. These include insensitivity to load distribution, the ease of fairly allocating the available bandwidth, low arbitration times, bounded access delay, and no requirement for long preambles.

In today's technology, a ring topology appears to best satisfy the requirements of high-performance networks operating from 20 to 500 Mbits/s where high connectivity and large extents are required. Some of the references, and in particular [10], are recommended for an in-depth review of the advantages that a ring design offers.

III. FDDI Characteristics

The initial version of FDDI uses optical fiber with light-emitting diodes (LED's) transmitting at a nominal wavelength of 1325 nm over multimode fiber. Connections between stations are made with a dual fiber cable employing a polarized duplex connector. A single-mode fiber (SMF) version of Physical Layer Medium Dependent (PMD) uses laser diode transmitters, with two power levels categories specified, the lower of which retains the same receivers as the basic PMD. SMF-PMD will allow individual links to be extended up to 60 or possibly even 100 km.

The data transmission rate is 100 Mbits/s. The effective sustained data rate at the data link layer can be well over 95 percent of this peak rate. The four out of five code used on the optical fiber medium requries a 125 Mbaud transmission rate. The nature of the clocking, which adjusts for accumulated jitter between frames, limits frames to 4500 octets maximum. Multiple frames may, however, be transmitted on the same access opportunity.

A total of 1000 physical connections and a fiber path of 200 km have been used as the basis for calculation of the default values of the recovery timers. Considering reconfiguration requirements, these choices allow a maximum configuration of 500 stations (each station represents two physical connections) linked by 100 km duplex cable. The choice of longer times than the default values for the recovery timers will allow larger networks to be configured. For smaller networks, performance can be optimized by choosing shorter times for the recovery timers. There is no minimum configuration requirement.

A. Station Organization

Fig. 1 shows the component entities necessary for an FDDI station. Identified components, conforming to both the IEEE 802 structure and the OSI concept of layering, are: Station Management (SMT), which specifies the local portion of the network management application process, including the control requried for the proper internal configuration and operation of a station in an FDDI ring; Media Access Control (MAC), which specifies the lower sublayer of the data link layer, including the access to the medium, addressing, data checking, and data framing; Physical Layer Protocol (PHY), which specifies the upper sublayer of the physical layer, including the encode/decode, clocking and framing for transmission; and Physical Layer Medium Dependent (PMD), which specifies the lower sublayer of the physical layer, including power levels and characteristics of the optical transmitter and receiver, interface optical signal requirements, the connector receptacle footprint, the requirements of confirming optical fiber cable plants, and the permissible bit error rates. Two alternative versions of PMD are shown; the basic PMD and SMF-PMD which allows the use of single-mode optical fiber.

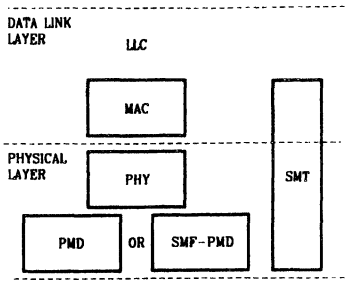


Fig. 1. FDDI relationship to OSI model.

FDDI MAC provides a superset of the services required by the Logical Link Control (LLC) protocol developed by IEEE P802.2. Fig. 1 assumes the use of the IEEE 802.2 LLC as the upper sublayer of the data link layer. Any other appropriate LLC may be used.

These basic component entities allow a variety of station types as shown in the FDDI topology example of Fig. 2. As shown, the FDDI trunk ring consists of a pair of counterrotating rings. Two main classes of stations, depending on whether or not they are allowed to attach directly to the trunk ring, are specified. A dual attachment station has two PHY entities. It may attach directly to the trunk ring or indirectly via a concentrator. It may have one or more MAC entities. In the case of two MAC entities, one may be in each of the counterrotating rings or both may be in the same ring. A dual attachment station may have optical bypass switches to remove it from both rings, at the same time healing them, if the station is powered down or disabled by SMT.

A single attachment station has one PHY and one MAC, and therefore cannot be attached directly into the main FDDI ring. Instead, it must be attached to the ring via a concentrator. A concentrator is a special station that has extra PHY's used to attach other stations that are to be inserted into the main FDDI ring. Varying levels of functionality, including multiple MAC's, are permitted in concentrators.

In Fig. 2 stations 1, 2, and 4 are dual attachment stations. Station 3 is a concentrator connecting stations 4, 5, and 6 into the FDDI ring. Stations 5 and 6 are single attachment stations. Station 4, even though it is a dual attachment station, must perform as a single attachment station insofar as its attachment to the ring via the concentrator is concerned. Operation of the second PHY of station 4 on some other ring is allowed, but there must not be any interconnection of the two rings within this station at a level visible to MAC or PHY.

B. Reliability Provisions

Ring topologies allow for the isolation of failing attachments through several mechanisms. Counterrotating rings, as shown in Fig. 3, are basic to the FDDI structure. The counterrotating ring concept uses two rings connected to each station or concentrator—one clockwise and the other counterclockwise. When a failure in a link occurs, the stations on either side reconfigure internally as shown in

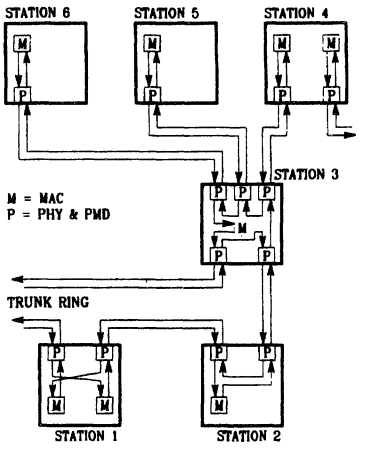


Fig. 2. FDDI topology example.

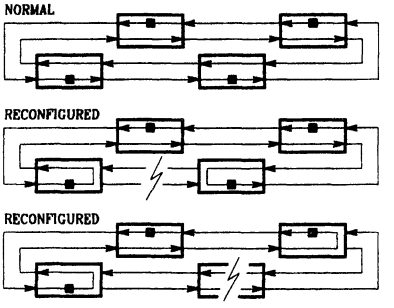


Fig. 3. Reconfiguration of counterrotating rings.

the middle diagram. The functional stations adjacent to the break make use of the connection in the reverse direction to close the ring, thus eliminating the bad link. In this figure, the dark squares represent the logical (MAC) attachment within the stations. Should a station itself fail, as shown in the bottom diagram of this figure, the stations on either side reconfigure to eliminate the failing station and both of the links to it.

Stations may offer a bypass capability, whereby an optical switch is used to bypass a station's receiver and transmitter connections so that the signal from the previous station is passed directly to the next station. Bypassing may be activated by a station itself, at the instigation of its neighbor, by a human operator, automatically at the removal of power, or by some overall network-controlling function.

Yet another approach is the use of concentrators, as shown in Fig. 2, that are attached directly to the trunk ring and, in turn, provide drop connections for a number of stations—in this case, stations 4, 5, and 6. A concentrator may then monitor all of its slave stations and isolate any faulty station. It may also provide for the graceful logical insertion and removal of stations from the ring.

The use of all three techniques allows FDDI networks to be configured to tolerate a variety of station or link failures and physical network configurations without catastrophic consequences. When failures occur, the network automatically reconfigures, eliminating any failing element and maintaining ring operation. Continuous mon-

itoring of the failed link or station allows the network to automatically reconfigure and restore normal operation when repair is effected. Any of these reconfigurations may result in the loss of individual frames, which then need to be retransmitted.

C. Data Encoding

Information on the medium uses a four out of five group code, with each code group called a symbol. As shown in Table I, in the 32-member symbol set, 16 symbols are data symbols, each representing four bits of ordered binary data. Three symbols are used for line-state signaling which is recognized by the physical layer hardware, two are used as control indicators, and four are used for starting and ending delimiters. One of the starting delimiters (*L*) is not used in the basic FDDI, but is reserved for FDDI-II. The remaining seven symbols of the symbol set are not to be transmitted since they violate code run length and dc balance requirements. The QUIET symbol is a necessary member of the line-state symbol set since it is used to indicate the absence of any functional signal.

D. Frame and Token Formats

Information is transmitted on the FDDI ring in frames which are variable in length. Tokens are special short fixed-length "frames" that are used to signify the right to transmit data. Fig. 4 shows the frame and token formats.

The Preamble (PA) field, consisting nominally of 16 IDLE symbols (a maximum frequency signal that is used for establishing and maintaining clock synchronization), precedes every transmission. The Starting Delimiter (SD) field consists of a two-symbol sequence (*JK*) that is uniquely recognizable independent of previously established symbol boundaries. The SD establishes the symbol boundaries for the content that follows.

The Frame Control (FC) is a two-symbol field that defines the type of frame and its characteristics. It distinguishes between synchronous and asynchronous frames, the length of the address field (16 or 48 bits), and the kinds of frame (e.g., LLC or SMT). One set of FC values is reserved for implementer frames that have no defined format and are to be repeated unchanged by all conforming FDDI stations. The FC field also provides for two kinds of tokens, restricted and nonrestricted. The latter is used in a special class of service which provides for extended dialogs among a limited set of cooperating stations. Two Ending Delimiter (ED) symbols (TT) complete a token.

The Destination Address (DA) and Source Address (SA) fields may be either 16 or 48 bits long, depending on the FC value. DA may be either an individual or a group address, the latter of which has the potential to be recognized by more than one station.

The 32-bit Frame Check Sequence (FCS) field is a cyclic redundancy check using the standard polynomial used in the IEEE 802 protocols. The information field of a frame, like the other fields covered by the FCS check, consists only of data symbols. Data symbols are not used in fields not covered by the FCS check.

The ED field of a frame is one delimiter symbol (*T*). It is followed by the Frame Status (FS) field that has a minimum of three control indicator symbols that are modified by the station as it repeats the frame. These indicate, when set, that an error has been detected in the frame by the station, that the addressed station has recognized its address, and that the frame has been copied by the station.

TABLE I
SYMBOL CODING

DECIMAL	CODE GROUP	SYMBOL	NAME	ASSIGNMENT
00	00000	Q	QUIET	LINE STATE SYMBOL
31	11111	I	IDLE	"
04	00100	H	HALT	"
24	11000	J		STARTING DELIMITER
17	10001	K		"
05	00101	L		"
13	01101	T		ENDING DELIMITER
07	00111	R	RESET	CONTROL INDICATOR
25	11001	S	SET	"
30	11110	0		DATA SYMBOL 0000
09	01001	1		" 0001
20	10100	2		" 0010
21	10101	3		" 0011
10	01010	4		" 0100
11	01011	5		" 0101
14	01110	6		" 0110
15	01111	7		" 0111
18	10010	8		" 1000
19	10011	9		" 1001
22	10110	A		" 1010
23	10111	B		" 1011
26	11010	C		" 1100
27	11011	D		" 1101
28	11100	E		" 1110
29	11101	F		" 1111
01	00001	V	VIOLATION	NOT TRANSMITTED
02	00010	V	VIOLATION	"
03	00011	V	VIOLATION	"
06	00110	V	VIOLATION	"
08	01000	V	VIOLATION	"
12	01100	V	VIOLATION	"
16	10000	V	VIOLATION	"

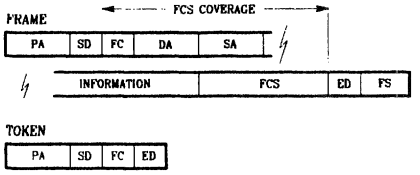


Fig. 4. Frame and token formats.

IV. Physical Layer (PHY and PMD) Operation

PHY provides the protocols and PMD the optical fiber hardware components that support a link from one FDDI station to another. PHY simultaneously receives and transmits. The transmitter accepts symbols from MAC, converts these to five-bit code groups, and transmits the encoded serial data stream on the medium.

The receiver recovers the encoded serial data stream from the medium, establishes symbol boundaries based on the recognition of a start delimiter, and forwards decoded symbols to MAC. Additional symbols (QUIET,

IDLE, and HALT) are interpreted by PHY and used to support SMT functions.

PHY also provides the bit clocks for each station. The total ring, including all stations and links, must remain the same apparent bit length (i.e., no bits may be created or deleted) during the transmission of a frame around the ring. Otherwise, an error would be generated in the frame as it is repeated around the ring. In the face of jitter, voltage, temperature, and component aging effects, such stability can only be realized through special provision. PHY provides an elasticity buffer which is always inserted between the receiver and the transmitter. The receiver employs a variable frequency clock, using standard techniques such as a phase-locked loop oscillator, to recover the clock of the previous transmitting station from the received data. The transmitter, in contrast, uses a local fixed-frequency clock. The elasticity buffer in each station compensates for the difference in frequency between the local clock and that of the upstream station by adjusting the bit delay through the station. The elasticity buffer in each station is reinitialized to its center position during the preamble (PA) that precedes each frame or token. This has the effect of increasing or decreasing the length of the PA, which is initially transmitted as 16 or more symbols, as it proceeds around the ring.

The transmitter clock has been chosen with 0.005 percent stability. With a minimum elasticity buffer of 10 bits, frames of up to 4500 octets in length can be transmitted without exceeding the limits of the elasticity buffer.

Simulations of early FDDI designs, presented to the technical committee, showed that rather than maintaining a nominal 16 symbol PA length, there was a tendency for the PA lengths in long rings to move toward a flat unbounded distribution. Longer PA's created no problem by themselves, but came at the expense of shorter, or even negative, length PA's. A negative PA indicates that the PA has been so severely shortened that it has completely disappeared and that symbols have consequently been lost off the end of a frame, resulting in the loss of the entire frame. This problem was solved by means of a smoothing buffer function incorporated into PHY. This examines the PA length between frames and either inserts or deletes, as the case may be, preamble symbols (or bytes) in order to maintain the PA near the nominal 16 symbol length, thus ensuring that the preamble never decreases below 4 or 5 bytes. Simulation work presented to the technical committee showed that the algorithm chosen, even under worst case conditions, reduced the probability of frame loss to less than 10^{-12}. Later work presented to the technical committee has reaffirmed this result with extended testing of a large physical ring configuration.

V. Token MAC Functional Operation

A major function of any station is deciding which station has control of the medium. MAC schedules and performs all data transfers on the ring.

The basic concept of a ring is that each station repeats the frame that it has received from its upstream neighbor to its downstream neighbor. If the destination address (DA) of the frame matches that MAC's address and there is no error indicated, then the frame is copied into a local buffer with MAC notifying LLC (or SMT) of the frame's arrival. MAC modifies the indicator symbols in the FS field of the frame as it repeats it to indicate the detection of an error in the frame, the recognition of its own address, and the copying of the frame. The frame propagates around the ring to the station that originally placed it on the ring. The transmitting station may examine the indicator symbols in the FS field to determine the success of the transmission. The MAC of this transmitting station is responsible for removing from the ring all of the frames that it has placed on the ring (a process termed stripping). MAC recognizes these frames for stripping by the fact that the SA contained in them is its own address. IDLE symbols are placed on the medium during stripping.

If MAC has a frame from LLC (or SMT) to transmit, it may do so only after a token has been captured. A token is a special frame that indicates that the medium is available for use. Priority requirements, necessary to assure the proper handling of frames, are implemented in the rules of token capture. Under these rules, if a given station is not allowed to capture the token, then it must repeat it (or in certain cases reissue a token) to the next station in the ring. Only after having captured a token and stripping it from the ring is MAC allowed to transmit a frame or frames. When finished, MAC issues a new token to signify that the medium is available for use by another station.

The FDDI MAC uses a Timed Token Rotation (TTR) protocol to control access to the medium. Under this protocol, each station measures the time that has elapsed since a token was last received. The initialization procedures establish the Target Token Rotation Time (TTRT) equal to the lowest value that is bid by any of the stations. Two classes of service are defined. Synchronous service allows use of a token whenever MAC has synchronous frames queued for transmission. Asynchronous service allows use of a token only when the time since a token last was received has not exceeded the established TTRT. Multiple levels of priority for asynchronous frames may be provided within a station by specifying additional (more restrictive) time thresholds for token rotation.

The use of the TTR protocol imparts some useful operational characteristics. It allows stations to request and establish (via SMT procedures) guaranteed bandwidth and response time for synchronous frames. It establishes a guaranteed minimum response time for the ring because, in the worst case, the time between the arrival of two successive tokens will never exceed twice the value of TTRT. It also provides a guaranteed level of ring utilizations equal to (TTRT–RL)/TTRT where RL is the physical ring latency—essentially the time for a token to propagate around the ring under no load conditions. Reference [15] and others by the same author have dealt extensively with these aspects of FDDI operation.

Low values of TTRT (e.g., 4 ms) may be used to es-

tablish an average token rotation time of 4 ms and a guaranteed response time not exceeding 8 ms. This would be useful in a time-critical application (e.g., packetized voice). Larger value of TTRT may be used to establish very high ring utilizations under heavy loads. For instance, using TTRT of 50 ms and a ring latency (RL) of 0.25 ms (reasonable for a ring consisting of 75 stations and 30 km of fiber), the above formula shows that a utilization of 99.5 percent can be achieved.

VI. Station Management (SMT) Operation

SMT is the local portion of the network management application process, including the control required for proper operation of an FDDI station in an FDDI ring. SMT monitors activity and exercises overall control of station activity. These functions include control and management within a station for such purposes as initialization, activation, performance monitoring, maintenance, and error control. Additionally, SMT communicates with other SMT entities on the network for the purpose of controlling network operation. Examples of these SMT functions include the administration of addressing, allocation of network bandwidth, and network control and configuration.

The SMT Connection Management (CMT) function establishes the physical connections between adjacent stations. For this function, CMT uses streams composed of QUIET, HALT, and IDLE symbols in low-level signaling protocols. Once a physical connection is established, CMT creates a logical configuration within the station by activating the appropriate paths between the PHY and MAC entities within that station. A large degree of flexibility is provided in the logical configuraiton, which may be established consistent with station functionality and desired station personality. This flexibility allows FDDI to support a wide range of topologies and applications.

VII. FDDI-II Concepts

FDDI-II is an upward-compatible enhancement of the basic FDDI that adds a circuit-switched service to the existing packet capability.

A packet service is a service where the elements of data to be transferred are placed in frames. Packets may vary in length and are self-defining in that each contains delimiters that mark its beginning and end and an address that specifies the target station. FDDI packets are called frames.

In contrast, a circuit-switched service provides a continuous connection between two or more stations. Instead of using addresses, the connection is established based upon some prior agreement, which may have been negotiated using packet messages or established by some other suitable convention known to the stations involved. This prior agreement typically takes the form of knowing the location of a time slot, or slots, that occur regularly relative to a readily recognizable timing marker.

A common timing marker used in North America is the Basic System Reference Frequency (BSRF), a 125 μs clock used by the public networks. Use of this clock is assumed for FDDI-II. In local FDDI usage, this is referred to as the cycle clock and is signaled by the *JK* starting delimiter of the FDDI-II cycle format.

In FDDI-II, a circuit-switched connection is described as N bits beginning at byte M after the cycle clock marker in Wideband Channel (WBC) number X. The last descriptor is necessary because FDDI-II has 16 WBC's that may be independently assigned to either packet-switched or circuit-switched data. This definition allows connections at data rates of all multiples of 8 kbits/s (i.e., $N = 1$) up to the 6.144 Mbit/s data rate of a WBC. If need be, multiple WBC's may be used to accommodate higher data rates.

The data transferred in a circuit-switched mode is best described as a stream of data. The data rate is appropriate to the service being provided—with, for example, 64 kbits/s being used for a digital voice data stream. Other data stream rates, even up to many Mbits/s in the case of video, are used for other applications. Once a connection is established, the data rate remains constant.

The contrasting nature of packet-switched and circuit-switched data is of interest. Most packet data traffic occurs in random quantities and at random times. This is referred to as asynchronous traffic. Other packet traffic, more regular in nature and occurring in relatively predictable quantities on a regular time basis, is referred to as synchronous (packet) traffic. In contrast, isochronous data occur in precise amounts on a precise time basis. They typically represent a sequence of digital samples from a sensor (e.g., voice or video). More importantly, isochronous data must by synchronized with clock information to ensure the accurate regeneration of the sampling clock (as distinct from the bit clock) to minimize distortion in data reconstruction. Isochronous data are more easily transferred in a circuit-switched network.

Networks that carry isochronous data must maintain precise synchronism with the cycle clock. For the FDDI ring, this means that one station (called the cycle master) must insert a delay for all isochronous data so that the ring appears to be an exact multiple of 125 μs in length. FDDI incorporates this delay in the cycle master in such a way that it does not cause any delay in the packet traffic. This is essential in providing an integrated services network with acceptable packet service.

VIII. FDDI-II Operation

Fig. 5, much like Fig. 1, shows how FDDI-II is implemented using one additional standard—Hybrid Ring Control (HRC). HRC becomes the new lowest sublayer of the data link layer, taking its place between MAC and PHY. HRC multiplexes data between the (packet) MAC and the isochronous MAC (I-MAC). This requires that the (packet) MAC be able to transmit and accept data on a noncontinuous basis because packet data are interleaved with isochronous data.

FDDI-II is a network with 100 Mbits/s of bandwidth available. This bandwidth may be devoted totally to op-

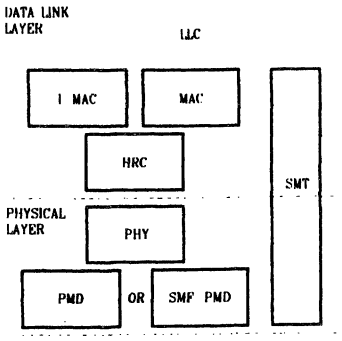


Fig. 5. FDDI-II relationship to OSI model.

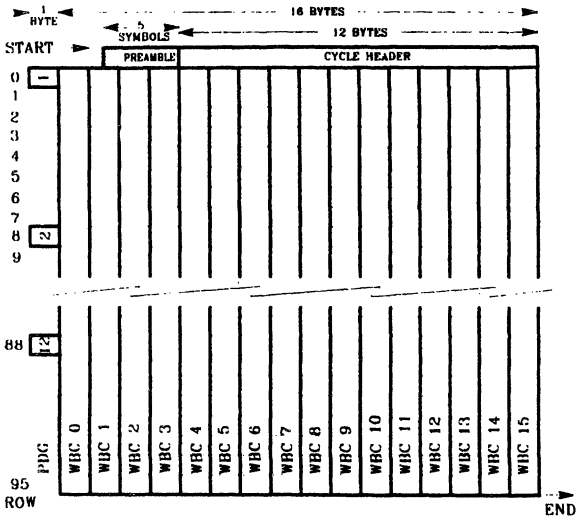


Fig. 6. FDDI-II cycle format.

eration as a packet network. Alternatively, portions of this bandwidth, in units of WBC's, may be dynamically separated for use with circuit-switched data. Up to 16 WBC's may be assigned. Each WBC is 6.144 Mbits/s, which is four times the North American and three times the European basic access rate to the telephone network. WBC's are full duplex and are independently allocatable and deallocatable. In effect, a broadband circuit capability has been provided with 16 available channels.

The WBC's provide a bandwidth division mechanism between the packet and isochronous traffic with a granularity of 6.144 Mbits/s. The allocation of virtual services within the isochronous traffic is allowed with an 8 kbit/s granularity. Once a station has been assigned a WBC or a number of WBC's, that station may suballocate the combined bandwidth of these WBC's as required. This suballocation may be in terms of any multiples of 8 kbit/s subchannels, including the commonly used 16, 32, 64 (*B* channel), 384, 1536, 1920, and 2048 kbit/s subchannels. Mixtures of these data rates in the same WBC are allowed. If preferred, the aggregate of any or all of the allocated WBC's may be used as one virtual service, satisfying the needs of such applications as high-resolution video. Thus, a multiplicity of virtual circuits may be provided within the same FDDI-II ring.

Assignment of all 16 WBC's, each at 6.144 Mbits/s, yields a total bandwidth of 98.304 Mbits/s. After allowance for the preamble and the cycle header, if all WBC's were allocated, a residual 768 kbit/s channel would then be left for packet traffic. This bandwidth, consisting of 12 bytes every cycle (125 ms), known as the Packet Data Group (PDG), is interleaved with the 16 WBC's as shown in Fig. 6. The order of transmission is left to right by row starting with the top row. Data steering logic in HRC augments the PDG with the bandwidth of any WBC's that are not assigned. Each WBC is one of the columns in Fig. 6 and represents a bandwidth of 6.144 Mbits/s or 96 bytes per cycle. This is an efficient system that allows the bandwidth of all unallocated WBC's to be used by the packet channel. Thus, for example, with eight (i.e., one half) of the WBC's assigned to isochronous service, 49.92 Mbits/s of bandwidth would be available for packet traffic.

A ring is initialized in basic (token) mode and switched to a hybrid mode of operation, combining both packet-switched and circuit-switched data capabilities, only after a station has negotiated for and won the right to be cycle master and has the synchronous bandwidth allocation required to support it. This cycle master then generates cycles at an 8 kHz rate (every 125 μs) and inserts the latency required to maintain an integral number of cycles synchronously on the ring. Alternative designs may allow the ring to be initialized directly in hybrid mode.

The cycle header format is shown in Fig. 7. It follows the Preamble (PA) which is nominally five symbols long. The Starting Delimiter (SD) is the same symbol pair (*JK*) used as the SD for frames when FDDI is operating in basic mode. In hybrid mode, frames use an IL symbol pair, readily recognized in the rigidly formatted context of hybrid mode, as their starting delimiter. The Cycle Sequence (CS) byte provides a modulo 192 cycle sequence count. The Maintenance Voice Channel (MVC) byte provides a 64 kbit/s voice channel for maintenance purposes.

The 16 symbols of programming information (*P*0–*P*15) determine whether the corresponding WBC (0–15) is allocated to packet or isochronous traffic. Thus, each of these (*Pn*) controls the multiplexing of one of the columns (WBC*n*) shown in Fig. 6. The *C*1 and *C*2 symbols are used for synchronization control in the transfer of this programming information to the other stations by the cycle master.

A. FDDI-II Priority Levels

Four kinds of traffic may coexist in an FDDI-II ring. Once WBC's have been allocated, the isochronous traffic within them has the highest priority.

Second highest priority is given to synchronous packet traffic where predictable units of data are to be delivered at regular intervals. Delivery is guaranteed with a delay not exceeding twice TTRT. These data may be transmitted following the capture of either a restricted or nonrestricted token.



Fig. 7. Cycle header format.

The bandwidth required for both isochronous and synchronous traffic is allocated from the available FDDI bandwidth. The allocation algorithm must ensure that the total allocation does not exceed 100 percent. Unallocated bandwidth is used on an as-available basis for asynchronous packet traffic.

Third highest priority is given to asynchronous traffic operating in restricted token mode. Such traffic may be transmitted upon the capture of either a restricted or a nonrestricted token. Cooperating stations may enter a restricted token mode of operation, which allows them to issue and use restricted tokens, only after negotiating an agreement using nonrestricted tokens. Restricted token mode operation allows stations to vie for available asynchronous bandwidth on a dialog basis.

Lowest priority is given to asynchronous traffic that may be transmitted only by capturing a nonrestricted token. This mode of operation allows stations to vie for the available asynchronous bandwidth on a single frame basis.

B. FDDI-II Applications

FDDI-II considerably expands the range of applications that may be addressed by FDDI rings. An FDDI-II ring may connect high-performance processors, mass storage systems, high-performance workstations, and perform the backbone function to a number of lower performance LAN's. The same ring may have some of its bandwidth allocated to isochronous services provided by the WBC's. This isochronous bandwidth may in turn be suballocated into a variety of virtual circuit services such as video, voice, and quite possibly, control or sensor data streams. The division of bandwidth between these two kinds of services may be adjusted based on the time of day or other requirements. In practice, no single instance of FDDI connects to all these types of equipment, and certainly not to all the equipment of a large site. Instead, multiple instances of FDDI-II rings will coexist with high traffic units, such as processors, attaching to multiple FDDI networks.

In a packet-switched mode, FDDI can be a backbone for bridges to a variety of other lower-speed LAN's, for example, the various IEEE 802 media access methods. It may also provide a backbone for gateways to the public data networks. In both cases, connections to processors and mass storage subsystems can be provided. Connections to high-performance workstations are likewise provided.

It is anticipated that FDDI-II implementations will follow those of the basic FDDI by one to two years. Committee X3T9.5 has specified the rules of coexistence of these two FDDI implementations to ensure interconnectability and interoperability of FDDI-II implementations operating in basic mode with the basic FDDI implementations. This also allows the use of FDDI-II chips for all applications once they become available.

IX. FDDI Standards Effort and Status

The FDDI standardization effort is taking place in the ASC X3T9.5 committee which meets bimonthly with an attendance of well over 100. The official voting membership, limited to one per corporation, is approximately 80. Ad·hoc working meetings, scheduled as required, are presently focusing on SMT, FDDI-II, and SMF-PMD detail design definitions as well as issues concerning MAC-level bridging.

After completion by ASC X3T9.5, a draft proposed American National Standard (dpANS) is forwarded to ASC X3T9 for a technical letter ballot which approves forwarding to X3. X3 conducts a four-month public review, followed by an X3 letter ballot. Finally, the dpANS is forwarded to the Board of Standards Review (BSR) for formal approval. Upon approval, the standard is given a final editing to conform to ANSI requirements and published.

MAC (Rev. 10) is an American National Standard (ANSI X3.138-1987) and was published in July 1987. Current activities are focused on potential enhancements to MAC, including formalization of the 48-bit addressing structure used in IEEE 802 LAN's, reliability enhancement in the criteria for frame recognition, and incorporation of the FDDI-II requirements.

PHY (Rev. 15) is also an American National Standard (ANSI X3.148-1988) and was published in December 1988.

PMD (Rev. 8), X3.166-198x, completed the X3 four-month public review in late 1987. This initial forwarding had been intended to allow a full set of comments on the optical parameters documented in it. Five sets of comments received resulted in a number of refinements to the optical specifications of PMD. There were also a number of comments on the connector choice and its documentation. In April 1988, X3T9.5 finalized the optical specifications and reaffirmed its connector choice. PMD, having completed a second (two-month) public review in December 1988, is now being processed for final approval as a standard.

Work has progressed smoothly on the SMF-PMD project which was started in mid-1987. An X3T9 technical letter ballot approved the SMF-PMD document (Revision 4) and it has been forwarded to X3 for the four-month public review and final approval as a standard.

The technical definition of FDDI-II, long maintained in a working paper, has now been completed and is contained in Revision 5 of the Hybrid Ring Control (HRC) document. This document also passed an X3T9 technical letter ballot and has also been forwarded to X3 for public review and approval.

Participation in the working meetings on SMT has been high through much of 1987 and into 1989, with well over 100 attendees at the most recent meetings. Representing

the meeting ground between the widely differing FDDI implementation and usage philosophies, SMT has proved a difficult task. Contentious items have included the use of concentrators without a MAC, link confidence test duration, physical layer link error monitoring, and physical layer signaling. But the committee's determination to produce an SMT stnadard assuring interoperability has prevailed and SMT is progressing toward a technical letter ballot in the latter part of 1989.

In a similar process designed to produce technically equivalent ISO standards for FDDI, the FDDI documents are being processed by ISO/IEC JTC1/SC 13. This process has resulted in the approval of MAC (IS 9314-2) and PHY (IS 9314-1) as ISO standards. PMD (DIS 9314-3) has passed the letter ballot for approval as an International Standard with publishing expected in late 1989. SMF-PMD and HRC are currently being processed for submission to ISO.

Led by the ASC X3T9.5, this FDDI standards effort has continued to reflect the FDDI product implementations of its many supporters, including a number of semiconductor manufacturers working towards FDDI chip sets. Several FDDI chip sets available either now or in the near future promise cost-effective implementations of these FDDI standards in the near future.

X. Conclusion

FDDI has proved a very successful standardization effort, far more successful than those who conceived it could ever have imagined. Users, system houses, chip designers, and component vendors have come together to produce a set of standards of lasting significance. With the widespread use of the basic FDDI will come a broad based market for FDDI-II which satisfies the much wider range of requirements typical of voice, video, and sensor/control based data streams.

References

[1] ANSI/IEEE Standard, 802.5-1985, "Token Ring Acess Method and Physical Layer Specifications."

[2] American National Standard, "FDDI Token ring Media Access Control (MAC)," ANSI X3.139-1987.

[3] American National Standard, "FDDI Token Ring Physical Layer Protocol (PHY)," ANSI X3.148-1988.

[4] Draft Proposed American National Standard, "FDDI Token Ring Physical Layer Medium Dependent (PMD)," ACS X3T9.5 Rev. 9, Mar. 1989 (ANSI X3.166-198x).

[5] Draft Proposed American National Standard, "FDDI Hybrid Ring Control (HRC)," ASC X3T9.5 Rev. 5, Apr. 1989.

[6] Draft Proposed American National Standard, "FDDI Token Ring Station Management (SMT)," ASC X3T9.5, Rev. 5, May 1989.

[7] Draft Proposed American National Standard, "FDDI Token Ring Single-Mode Fiber Physical Layer Medium Dependent (SMF-PMD)," ASC X3T9.5 Rev. 4, Apr. 1989.

[8] F. Ross, "FDDI—A tutorial," *IEEE Commun. Mag.*, May 1986.

[9] V. Iyer and S. Joshi, "FDDI's 100-Mbps protocol improves on 802.5 spec's 4-Mbps limit," *EDN*, May 2, 1985.

[10] F. E. Ross, "Rings are 'round for good," *IEEE Network Mag.*, Jan. 1987.

[11] M. J. Johnson, "Reliability mechanisms of the FDDI high bandwidth token ring protocol," *Comput. Networks ISDN Syst.*, vol. II, no. 2, pp. 121-131, 1986.

[12] H. Salwen, "In praise of ring architecture for local area networks," *Comput. Design*, pp. 183-192, Mar. 1983.

[13] R. M. Grow, "A timed token protocol for local area networks," presented at Electro'82, Token Access Protocol (17/3), May 1982.

[14] M. J. Johnson, "Fairness of channel access for non-time-critical traffic using the FDDI token ring protocol," in *Proc. Seminar Real-Time Local Area Networks*, Bandol, France, Apr. 1986, pp. 145-157.

[15] ——, "Proof that timing requirements of the FDDI token ring protocol are satisfied," *IEEE Trans. Commun.*, vol. COM-35, pp. 620-625, 1987.

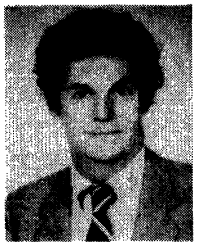
[16] F. E. Ross, "FDDI—A perspective," in *1988 Fiber Optics Sourcebook*, 3rd annu. ed., pp. 193-210.

[17] M. J. Johnson, "Analysis of FDDI synchronous traffic delays," in *Proc. Syst. Design Networks*, Apr. 1988, pp. 65-72.

[18] ——, "Performance analysis of FDDI," presented at EFOC/LAN'88, June 1988.

[19] J. Hamstra, "FDDI design tradeoffs," presented at the 13th Conf. Local Comput. Networks, Oct. 1988.

[20] W. E. Burr, "The FDDI optical data link," *IEEE Commun. Mag.*, May 1986.



Floyd E. Ross received the B.A. degree in mathematics and physics from Mount Allison University in 1955 and did postgraduate work at McMaster University, Hamilton, Ont., Canada.

He is a Staff Engineer with UNISYS, Devon, PA, with responsibilities in communications networks and in LAN definition and implementation. He is an active contributor to accredited standards committee X3T9 on I/O Interfaces and to IEEE 802. He is Vice Chairman of X3T9.5 and has been the leader of the FDDI standards development effort since its inception in that committee. He has chaired many of the FDDI working meetings including those on SMT and FDDI-II and is technical editor of the four basic FDDI standards. He has also presented and authored a number of papers on both local area networks and FDDI.

The QPSX Man

R. M. Newman
Z. L. Budrikis
J. L. Hullett

Reprinted from *IEEE Communications*, Vol. 26, No. 4, April 1988, pages 20-28. Copyright © 1988 by The Institute of Electrical and Electronics Engineers, Inc. All rights reserved.

Introduction

Public communications are catching up with demands set by information technology developments and will soon lead them. A number of standards forums, the CCITT, T1, the IEEE, the ISO and others are working on interrelated projects to define interfaces, networks and protocols that promise to make data communications over a wide spectrum as ubiquitous and as universally switchable as the telephone. More than that, from the user's prospective all communications will be fully integrated flowing over the one interface and access loop, with telephone just one of the many services.

The CCITT has already defined the integrated services digital network (ISDN) and is defining the broadband ISDN, or B—ISDN. T1 is defining a synchronous optical network (SONET), capable of carrying the large traffic within the network. ISO is standardising protocols and procedures so that different equipments can communicate with each other. The IEEE in P802 is bridging the gap between private and public networks. It is defining local area networks (LANs) suitable for customer premise or campuses, and metropolitan area network (MAN) that will interconnect LANs and provide other integrated services as part of the public network. In fact, the IEEE defined MAN can be a significant stepping stone on the path to a full B—ISDN.

The IEEE MAN is being defined in 802.6, a Working Group of P802. The project is actively supported by Telcos which gives assurance that it will measure up to public network requirements. A technology proposal that has particularly wide Telco support is one put forward from Australia. It is the Dual Bus QPSX MAN [1],[2].

QPSX (the queued packet and synchronous switch) is a distributed switch/network that will fulfill the requirements of a public MAN. The switch architecture of QPSX is based on two contra-directional buses as shown in Fig. 1. This allows full duplex communications between each pair of nodes of the switch. The communications on QPSX are integrated, providing isochronous (circuit switched) and non-isochronous (packet switched) services. The two switch modes function independently and the total capacity of the switch is shared flexibly and completely between them.

The bus architecture of QPSX is inherently highly reliable. The operability of the network is to all intents independent of the operation of the individual nodes. Furthermore, fault handling mechanisms have been developed that permit the isolation of potentially catastrophic faults with no service degradation in the case of a single failure, and minimal disruption even with multiple failures.

The access of packets to the buses of QPSX is controlled by a unique MAC (media access control) protocol referred to as distributed queueing [3]. The distributed queueing protocol provides near perfect access characteristics independent of network size and speed. In particular, the protocol supports 100 percent throughput and can provide average packet access delay equal to that of a perfect scheduler, the ultimate queueing system.

EH0301-2/91/0000/0059/$01.00 © 1988 IEEE

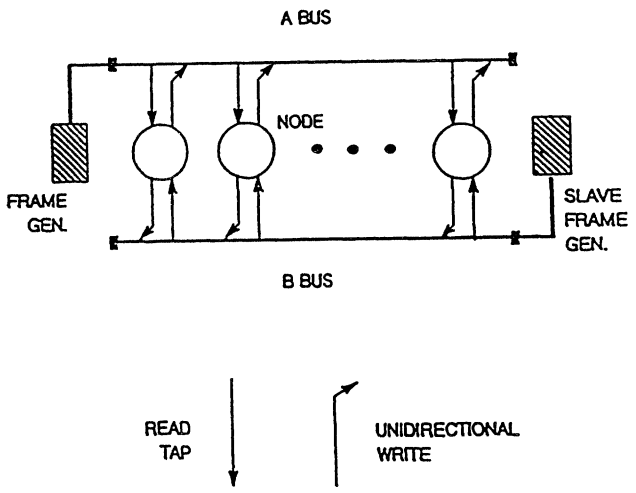


Fig. 1. QPSX dual bus architecture.

The operation of the protocol is based on two control bits, the busy bit which indicates whether a slot on the network is used and the REQ (request) bit which is sent whenever a node has a packet waiting for access. Each node by counting the number of REQs it receives and non-busied packets that pass it, can determine the number of packets queued (i.e., in line) ahead of it. This counting establishes a single ordered queue across the network for access to each bus. With such queued access, levels of priority can be established by operating a number of queues, one for each level. Within each level the perfect performance characteristics are maintained: the packets will gain access as soon as capacity becomes available, but priority is always given to packets in higher level queues.

QPSX as a Public Network MAN

The P802 forum started with development of LAN standards and much of the MAN work has evolved from this base. However, the MAN has very much greater functionality and a different role. The increased functionality includes the ability to cover much larger geographical areas, to operate at much higher speeds, and to provide switching for a mix of traffic services. These capabilities take the MAN well beyond the realm of in-house or building networks and into the domain of the public network.

QPSX MAN is particularly well suited to application in the public network. It has the necessary features of outstanding reliability and independence of network size and speed, to meet the requirements of all public network services. As well, it has the features, such as security, network administration, bandwidth control and charging facilities, which are also essential in the public network. A further advantage in relation to its application in the public network is that it has a frame structure that will be compatible with the evolving broadband ISDN. Hence, QPSX can be seen as an evolution path to this future public network technology.

The Dual Bus Architecture of QPSX

The principal components of the QPSX architecture are a head station, two unidirectional buses, and a multiplicity of access units (AUs). These are configured in the unidirectional bus structure shown in Fig. 1. The head station generates the frame synchronization on the forward bus and the end station generates the frame pattern at the same rate on the reverse bus.

Access units attach to both buses via read and write connections. The writing on to the bus is by logical OR of the data from upstream with the data from the unit. The read connection is placed ahead of the write connection and allows all data to be copied from the bus unaffected by the unit's own writing. The digital bus implementation of the AU connection is shown in Fig. 2.

The bus architecture of QPSX has been chosen for its reliability. As can be seen from the Fig. 2, the data bus passes directly by each node in the network. Data does not pass through the node as is the case in ring networks. Hence, in QPSX, nodes can fail or even be removed from the network with no consequence on the operation of the rest of the network. The network does not even need to reconfigure in this case.

The only fault conditions in QPSX that possibly may affect the operation of the network are a "jabbering" node or failure of the bus. The first fault is handled by a fault detection and isolation scheme which may be instituted, again without causing reconfiguration of the network. Failure within the bus is the only condition handled by network reconfiguration. The QPSX reconfiguration scheme is simple and very fast acting.

The Looped Bus Architecture

An extension of the dual bus architecture is the looped bus shown in Fig. 3. The only change in this from the original architecture is that the end points of the buses are co-located. Data do not flow through the head point of the loop. Thus, while the looped bus architecture may appear similar to a ring, it has no log-

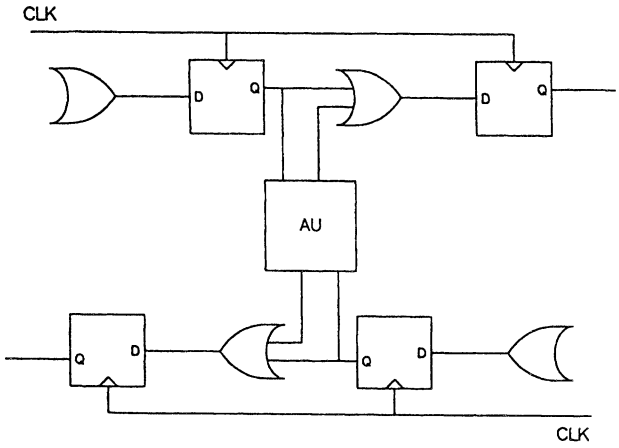


Fig. 2. Digital bus realization.

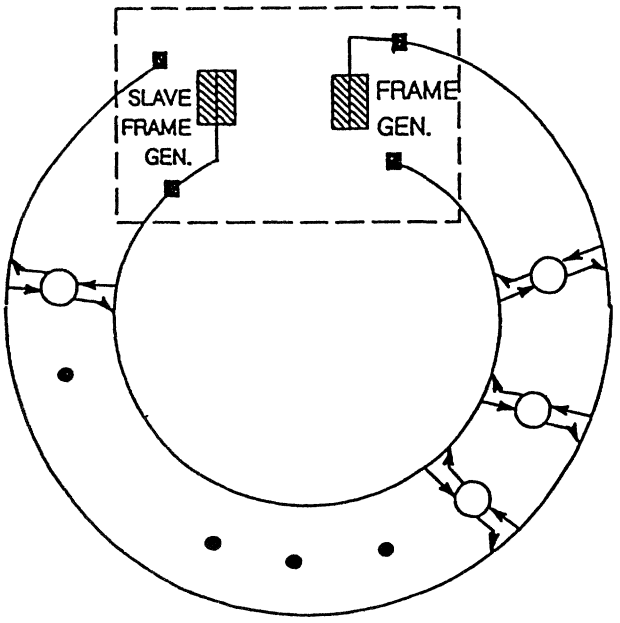


Fig. 3. QPSX looped bus architecture.

ical similarity to it. Hence, it provides all the reliability advantages associated with the QPSX bus architecture.

There are two purposes in looping the QPSX bus architecture. One is that the framing generator for the two buses is common and this point provides the entry point for the PSTN clock into the network. The second and more important reason for looping the buses is that it provides an automatic route diversity within the switch. In the case of a bus fault, the network can isolate the fault and close the data buses through the head point of the loop. Hence, the network is reconfigured and fully operational in the reconfigured state. This is achieved without the need for any redundant transmission hardware, which is in direct contrast with ring networks that require a second redundant ring to institute network reconfiguration.

The Architecture of the Public MAN

The role of the QPSX MAN in the public network is to provide interconnectability for a range of communication services not yet provided. The geographical scope of the MAN is such that it is planned to cover complete cities. In practice it is most sensible if the MAN covers an area that corresponds to the telephone number plan area, NPA. While the dual bus architecture of QPSX is such that a single network would operate over very large geographical areas, it will not be long before the capacity requirements of the attached customers will exceed 2×150Mbits/s capacity of the network. In fact the number of networks required in large cities could well approach the hundreds and even thousands. As a result, there is a need for bridged interconnection of networks to form a single large MAN.

In order to permit a manageable and efficient interconnection of the networks within the MAN, a simple interconnection architecture and bridging algorithm is required. The architecture of the public network based on QPSX is very similar to that of the public telephone network and is illustrated in Fig. 4. It is based on a hierarchy with possible cross links between switches at a common level of the hierarchy. The cross links provide route diversity in the case of failure and also provide a short path between switches which have a high degree of cross traffic.

At the lowest level of the MAN hierarchy is the access network, shown as level one in the Fig. 4. This level provides for the customer's connection into the public network. The protocol operated over the access network is the same QPSX protocol. The access network will go from the customer to the nearest site owned by the public network provider. In terms of utilizing the cabling efficiencies associated with distributed switching, if the network provider has space leased at sites away from the central office, then connection to the public network switch will be at those points. Otherwise, all connections to the public switch will be in the central office. In this latter case, the length of the access network loops generally will be longer.

The point where the access network reaches the office level network is the edge of the public network. This point, called the network connection point (NCP), is responsible for maintaining and administering the customer's connection. The functions performed at NCP are illustrated in the Protocol architecture diagram, Fig. 5. There are two principal sets of functions performed at the NCP. These are routing and edge-to-edge services. The routing layer performs the function of converting the destination E.164 address indicated in packets sent by the customer to determine where the packet is to be routed. The routing layer will translate that address into an internal address used within the QPSX MAN. The edge-to-edge

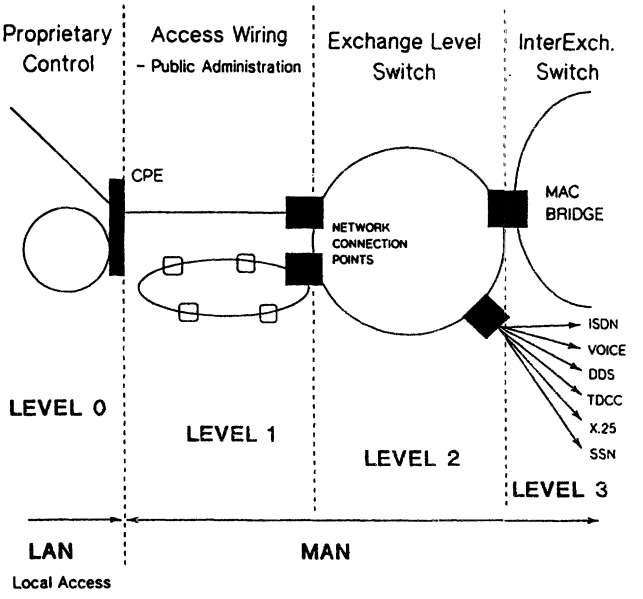


Fig. 4. QPSX hierarchy (physical architecture).

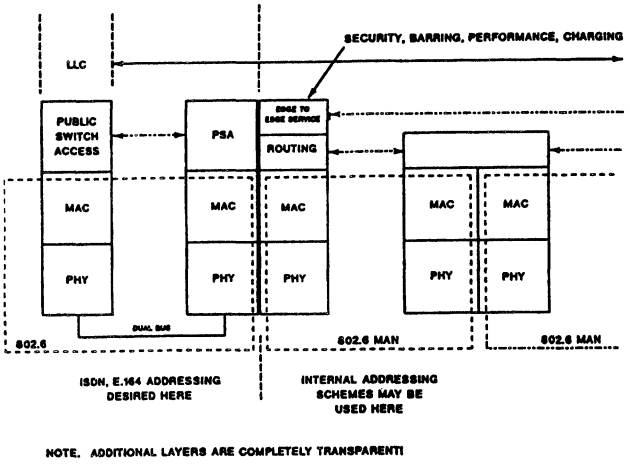


Fig. 5. Protocol architecture.

services layer will perform functions related to the network provider's management of the customer's connection, such as security provision, charging, performance monitoring barring, etc.

The NCP represents the point where the customer is physically and logically connected to the switch hierarchy of the QPSX MAN. The switch hierarchy consists of office level switches at the lowest level, interoffice switches at the next higher level and so on upwards to whatever depth is required. All switches are interconnected by bridges which can provide up to 150Mbits/s throughput. The same system hardware can be used at all levels of the hierarchy, though ultimately higher rate switches will be required at the higher levels. Nodes at all switch levels are connected to the QPSX bus at cluster points, and clusters may be either widely distributed, or adjacent thus forming a centralized switch realization.

Communications across MANs will be routed through the source MAN to a gateway to the wide area network (WAN). This gateway will again perform the necessary routing layer functions using the E.164 address contained in the packet to determine the path through the WAN to the destination MAN. It is not intended, at least in the first realizations of QPSX, for synchronous communications to be carried in the WAN or even in the interoffice switches of the QPSX hierarchy. Rather this traffic will be given over to the existing switch services provided for it.

The routing within the MAN is based on a simple protocol which permits very high throughput in the bridges of the MAN. In fact, a bridge can switch between two networks up to the full capacity of the QPSX dual bus. Route diversity within the bridging algorithm is also provided to circumvent bridge faults.

Frame Format and Synchronization

Communications on the QPSX buses are within the frame format shown in Fig. 6. The frame interval is 125 microseconds, matching that of digital telephony. The frame is subdivided into a fixed number of equal size units called slots. Slots provide the subdivision between the isochronous and non-isochronous traffic. A slot may be allocated to contain either type of communication. Slots are allocated to isochronous use commensurate with the demand for isochronous capacity. The slots not allocated to isochronous use are available for packet communications.

For isochronous communications, each octet within a slot provides a 64kbit/s channel. By using multiple octets per frame higher rate circuits can be constructed. By the use of a multi-frame count in the frame header, circuits at sub-64kbps can also be constructed.

For non-isochronous communications, a slot once claimed for access by a node, is used to transfer a single segment of a packet. Access to slots is handled by the Distributed Queueing protocol. The segmentation of packets into slots and the reassembly of the packet at the destination is by a source-identifier-linked segmentation and reassembly protocol. Both protocols are discussed in later sections.

The frame structure for QPSX is very similar to that which is currently under consideration in CCITT as the structure for B-ISDN. It has already been decided that the B-ISDN frame will consist of fixed sized cells, these cells being a similar concept to QPSX slots. In the standardization effort of QPSX, it is expected that slot size and slot header size will be aligned with that adopted by CCITT for B-ISDN cell size and cell header. Further levels of compatibility such as the format of the header will be a future issue. However, it is expected that IEEE 802.6 will develop a MAN standard well in advance of the B-ISDN standard by the CCITT.

The Distributed Queueing Protocol

Distributed Queueing is a media access control (MAC) packet switching scheme that controls the access of fixed length data segments to the slots on the QPSX bus. The protocol overcomes many of the limitations of other media access protocols. The maximum throughput of the network is not constrained by network speed or size and the protocol operates on the highly reliable dual bus architecture.

The operation of Distributed Queueing is fundamentally different from all existing MAC protocols. In all other access schemes, no continuous record is kept in the nodes that explicitly indicates the state of the network. Hence, when in these protocols a node has a packet to transmit, it must first derive the information from the network as to when to access. This leads directly to a sensitivity in almost all protocols to the size of the network. The larger the network, the longer the delay in obtaining the access information. This limits the size over which the protocols may operate efficiently.

By contrast, with Distributed Queueing a current state record is kept in every node which holds the number of segments awaiting access to the bus. When a node has a segment for transmission, it uses this

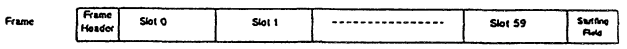


Fig. 6. Frame format for a QPSX man.

count to determine its position in the distributed queue. If no segments are waiting, access is immediate, otherwise deference is given only to those segments that queued first.

With the queueing of segments for the bus, slots are never wasted. This guarantees minimum access delay at all levels of loading right up to the maximum network utilization of 100 percent. This performance is achieved with negligible control overhead, in fact only two bits per slot, and is effectively independent of network size and speed.

The Basic Distributed Queueing Algorithm

The operation of the basic Distributed Queueing algorithm is particularly simple, as illustrated in Fig. 7. The protocol uses just two control bits, busy and REQ, to control access to the slots on the forward bus. An identical, but independent arrangement applies for access to the opposite bus.

The Busy control bit is a marker at the head of each slot which indicates whether the slot is full and hence not available for access. The REQ control bit is the crux of the queueing mechanism and is used to signal when a segment has queued.

When a node has a segment for transmission on the forward bus it will issue a single REQ bit on the reverse bus. This bit will pass to all upstream nodes, where upstream is defined in relation to flow on the forward bus. This REQ bit serves as an indicator to the upstream nodes that an additional segment is now queued for access.

Each node keeps track of the number of segments queued downstream from itself by counting the REQ bits as they pass on the reverse bus, as shown in Fig. 8a). For each REQ passing on the reverse bus, the request (RQ) counter is incremented. One REQ in the RQ counter is canceled each time an empty slot passes on the forward bus. This is done since the empty segment that passes the node will be used by one of the downstream queued segments. Hence, we see that with these two actions the RQ counter keeps a precise record of the number of segments queued downstream.

When a node has a segment for access to the bus, it transfers the current value of the RQ count to its second counter, the countdown (CD) counter. This action loads the CD count with the number of downstream segments queued ahead of it. This, along with

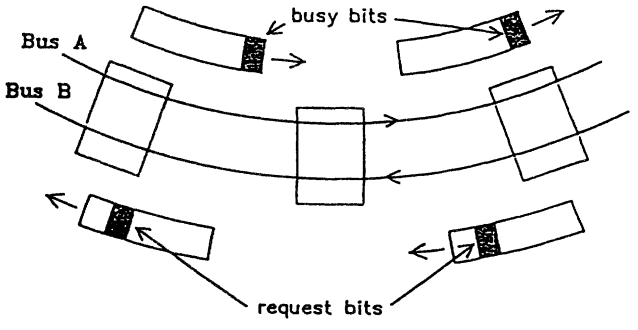


Fig. 7. Distributed queueing. Queue formation on Bus A.

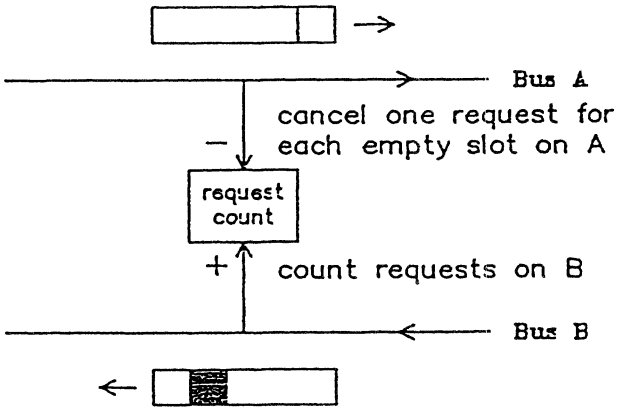


Fig. 8a). Distributed queueing (Bus A). Node not queued to send.

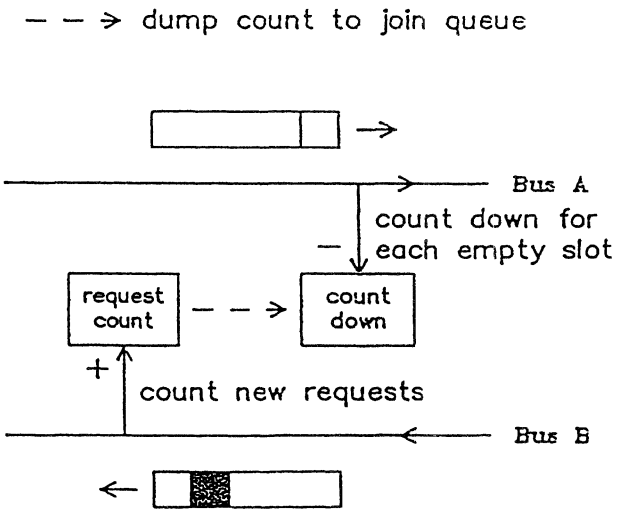


Fig. 8b). Distributed queueing (Bus A). Node queued to send.

the sending of the REQ for the node's segment, effectively places the segment in the distributed queue.

To ensure that the segments registered in the CD counter gain access before the newly queued segment in the given node, the CD counter is decremented for every empty slot that passes on the forward bus. This operation is shown in Fig. 8b). The given node can then transmit its segment in the first empty slot after the CD count reaches zero. The claiming of the first free slot ensures that no downstream segment that queued after the given segment, can push in and access out of order.

During the time that the node is waiting for access for its segment, any new REQs received from the reverse bus are added to the RQ counter, as shown in Fig. 8b). Hence, the RQ count still tracks the number of segments queued downstream and the count will be correct for the next segment access.

Thus, with the use of two counters in each node, one counting outstanding access requests and the other

counting down before access, a first-in-first-out queue is established for access to the forward bus. The queue formation is also such that a slot is never wasted on the network if there is a segment queued for it. This is guaranteed since the CD count in the queued nodes represents the number of segments queued ahead. Since at any point in time one segment must have queued first, then at least one node is guaranteed to have a CD count of zero. It is that node that will access.

Priority Distributed Queueing

The Distributed Queueing protocol can be extended to permit the assignment of priority to segment access. The priority access control is absolute in that segments with a higher priority will always gain access ahead of segments at all lower levels. This is achieved by operating separate distributed queues for each level of priority.

The operation of separate distributed queues for each level of priority consists of the use of a separate REQ bit on the reverse bus for each level of priority and separate RQ counters for each priority level. The counters operate much as before, except that account must be taken of REQs at higher levels. That is, a RQ counter operating at a particular level will count REQs at the same and higher priority levels. Thus the RQ count records all queued segments at equal and higher priorities.

The operation of the CD counter is also slightly altered. A CD counter operating at a particular priority level will, in addition to counting down the passing empty segments, be incremented for REQs received at higher priority levels. This allows the higher priority segments to claim access ahead of already queued segments.

With such a priority scheme, the access performance for high priority traffic is ideal. Immediate access can be gained without any delay due to already queued traffic. Such a feature is very important for network signaling. It would also be important for packet voice, should that be implemented.

Performance

The operation of the Distributed Queueing protocol provides two key advantages over the operation of other existing protocols. One is that, through its use of REQ counters, an ordered queue is established for access where capacity on the network is never wasted. The second advantage is through the use of segmented access, which allows finer granularity and control in the allocation of the bandwidth resource.

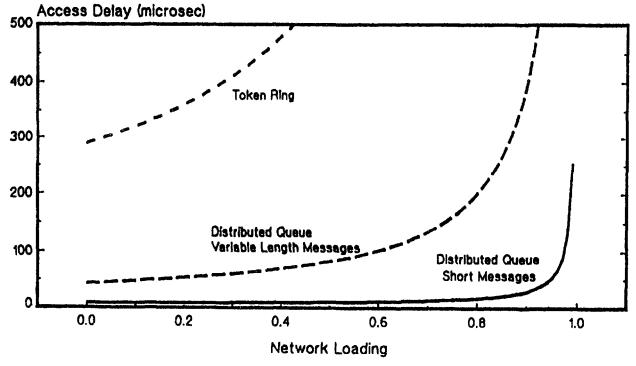


Fig. 9. Delay for distributed queueing and token ring (100 Mbits/s, 100km).

The access performance advantage for the Distributed Queueing protocol is best demonstrated by the graph in Fig. 9. This figure shows the access delay for packets on a typical MAN. To represent the delay best it has been normalized to the average length of the packets. Also, shown on the figure is the token ring protocol. Distributed Queueing has a very clear advantage in terms of access delay.

The advantage demonstrated in Fig. 9 is even further exaggerated when either the speed or the size of the network increases. The normalized delay performance for the Distributed Queue is unaffected by these network parameters, whereas the token ring normalized delay will continue to increase with the network size and speed. The increase in delay is clearly undesirable.

The limitation of the token ring is the delay in the transfer of the token. Even at very low utilization, a node wishing to access must wait for the token to pass to it. On average this delay is at least half of the ring latency at low loading and increases beyond this as load increases. Such a delay eventually limits the usable size of a token ring network.

The second performance advantage associated with distributed queue access is that of segmentation. Packets sent over the QPSX network are not sent as a whole but rather in segments to suit the slot size on the bus. Segmented transfer versus whole packet transfer is much like the comparison of time shared processing versus batch processing.

With whole packet transfer and likewise batch processing, the single processor is allocated to serve each packet unit at a time. Consequently, very large packets cause long delays for all other packet transfers. Most affected by this are short packets and high priority packets. Both must wait while the large packet completes access.

By contrast to this operation, with segmented access such as in the distributed queue the access is shared on a segment by segment basis. This means that even during the transfer of a very large packet, higher priority packets need only wait at most one segment before gaining access. Likewise, other packets at the same priority will be able to obtain immediately their share of the capacity. Demonstration of the access advantage for the short packets is illustrated also in Fig. 9.

Thus, segmentation provides a much finer granularity for bandwidth sharing and aligns communications switching with the performance expectations now commonly accepted in the computer industry.

The Segmentation and Reassembly Scheme

We have noted the performance advantages associated with segmented access of packets. Now we describe the scheme used to control the segmentation of packets at the source and the reassembly at the ultimate destination.

The segmentation process is straightforward and involves the fragmentation of the original packet to match the slot size on the bus. This process is shown in Fig. 10, where the segments of the packet are formed. This includes also the forming of a packet header segment which is necessary in routing the packet and instituting the reassembly process.

All segments consist of a header field along with the actual data as shown in Fig. 11. The header field consists of two sub-fields. The first is a single bit, the C/D subfield, which is used to indicate whether the segment is a control or a data segment. The second field is the source identifier (SI). This is used to provide the logical linking between slots of the same packet.

The SI is a fifteen bit field. Each node in a QPSX dual bus will have one or more unique SI number(s) related to its physical location along the bus. The SI is in addition to, and independent of, any network layer addressing. The allocation of the SI numbers is locally administered and only has significance within a single network. The purpose of the SI is to identify segments of a single packet. This information is used to reassemble the segmented message at the destination. To describe the operation of the packet transfer scheme, the segmentation of the packet at the source is described first and the action of the receiver is considered later.

The train of segments sent by the source is shown in Fig. 10. The first segment of a multi-segment message is a control segment. Within this segment a control field immediately after the segment header will indicate the beginning of message (BOM) code. This code signifies the start of a new packet transfer. The SI field of this segment is set to the source's value and the information field of the segment contains additional control information amounting to a packet header. Part of this information is routing information required to transfer the packet to the destination. All routing within the MAN is based on physical access unit addresses, not on the network layer addresses. Also, contained within the information field of the first segment is the length of the packet.

All segments of the packet until the last are each placed in the information fields of slots following the

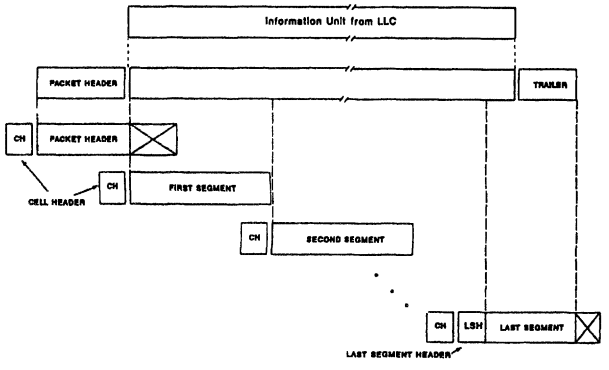


Fig. 10. Packet segmentation scheme.

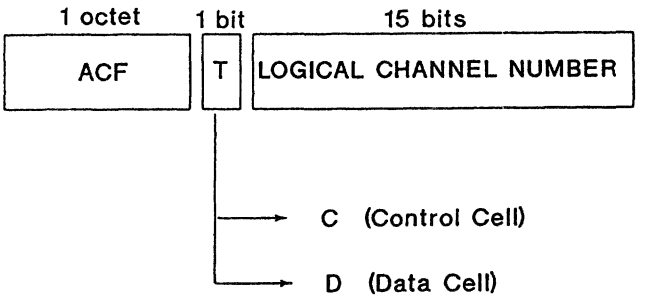


Fig. 11. QPSX segment header.

first control slot. In the header of each of these slots the C/D bit is reset to indicate data and the SI field is the same value as was sent in the first or control segment. The transfer of the multi-segment packet is completed by sending a further control slot which contains the end of message (EOM) code in the control field after the segment header.

For the transfer of a message that only requires a single segment, the SSM (single segment message) code is used in the control field of the first segment. The SI is not required in this case, however it is still used for consistency in operation.

To receive the packets segmented as described above, each AU will implement the state machine in Fig. 12. This state machine will handle the receipt of a packet from a single source at a time. To handle concurrent receipt of more than one message, multiple state machines and receiver buffers are used in the AU. In the diagram, the condition for a state transition is shown at the start of the transition line and the action taken is at the arrowhead of the transition line.

The receive machine consists of two states: IDLE and REASSEMBLE. In the IDLE state the machine is not currently receiving any message. Thus, in this state the machine will check for control slots with the control field equal to BOM or SSM. In the case a BOM code is received the machine will check the destination identifier in the first part of the information field. If the destination identifier is its own, then this indicates the packet is intended for that station and the state machine enters the REASSEMBLE state. In the case that SSM is detected in the IDLE state and the destination identifier field matches, the length and information fields are copied and the AU will indicate to the higher layers that a packet is received. The receive

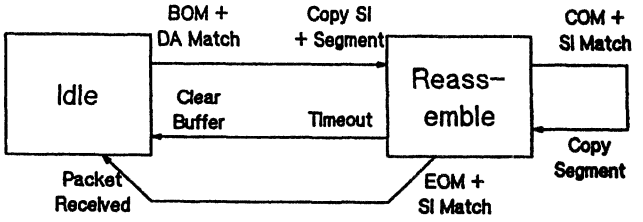


Fig. 12. Reassembly machine.

machine remains in the IDLE state after copying a SSM segment.

The REASSEMBLE state is used to receive the segments following the BOM control segment of a multi-segment message. In this state, the segments with the SI equal to that copied from the BOM segment will be received. The information fields of these following segments are appended together to form the complete packet. New packets addressed to the given station in this state are ignored by the state machine. Other idle receive machines are used to receive such messages. When the EOM segment of the packet is detected by the receive state machine, again with the same SI, the machine will pass the reassembled packet to the station and return to the IDLE state. This completes the receipt of the packet.

To guard against the loss of the EOM segment, which would cause the receive state machine to be locked in the REASSEMBLE state, a timer is used. This timer is started after each segment is received. If the timer expires before the next segment is received, a failure in the transfer is assumed. The machine will then clear all copied slots and return to the IDLE state. Loss of segments of the packet can be detected by comparing the length field given in the BOM slot with the number of slots received.

The SI can be reused by the source for its next packet transmission immediately after the EOM segment is sent. Even if the destination of the first packet did not receive its EOM slot, the immediate reuse of the SI does not present a problem. The BOM issued for the new packet can serve to terminate the receipt of slots by the first destination. As a further safeguard, knowledge of packet length can prevent the receipt of further segments not intended for the first destination.

Stations that need to send more than one message concurrently, such as bridges, may be allocated more than one SI. The number of packets that may be sent concurrently equals the number of SIs allocated.

Reliability Features of the QPSX Network

The reliability of QPSX stems both from its architecture and its Distributed Queueing medium access control protocol or MAC. There are two reasons for the inherent reliability of the bus architecture. First, the network nodes are logically adjacent to the bus not serially and logically connected as in a ring. As a consequence data do not pass through the nodes and passive MAC failures have no effect. Secondly, because the node writing function is by logical "OR," i.e., there is no

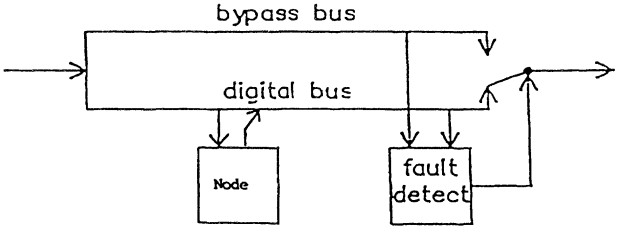


Fig. 13. Node fault detection.

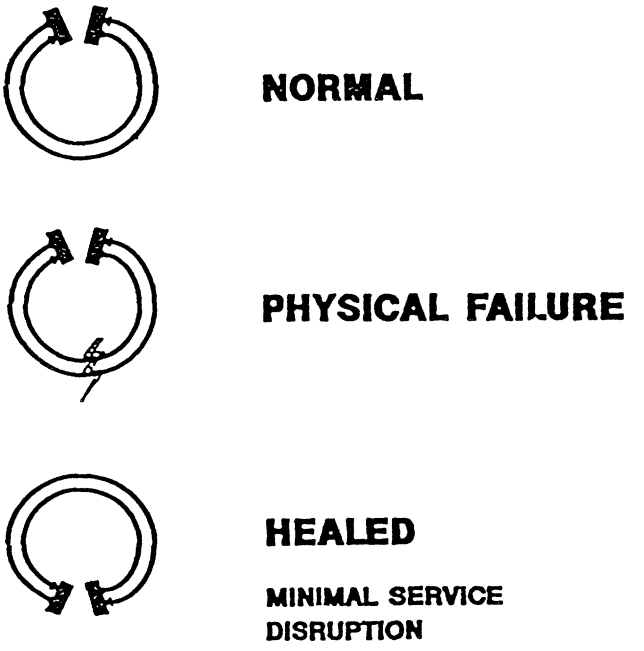


Fig. 14. Healing mechanism. Physical breaks are healed by repositioning the natural break in the loop.

overwriting, active MAC failures and bus failures are easily detected by input to output coincidence checking. An arrangement to implement this is shown in Fig. 13. Such failure points when they occur are bypassed at the node. The operation can be totally hardware based, with immediate response and causes no loss of synchronism, and minimal network disruption. With QPSX no node monitoring functions are required to check the MAC.

In this context a comparison with ring based networks is pertinent. With rings [4] both active and passive node MAC failures are catastrophic in that both invoke healing mechanisms associated with redundant rings. Such action involves considerable network disruption because of the need to reconfigure with attendant loss of synchronization. Any additional node failure can then produce isolated network islands.

With the QPSX architecture those faults involving node failure do not need to invoke any network healing mechanism which reconfigures the network. Faulty nodes are easily and non-disruptively bypassed.

Failures in the physical transport however do involve reconfiguration. But on healing the network captures the same synchronization and therefore there is minimal disruption to all synchronous communications. The healing mechanism is depicted in Fig. 14 and shows how physical faults or line breaks are healed by repositioning the natural break in the loop to the position of the break due to failure. Significantly there is no loss of network capacity involved in network reconfiguration. Here again the contrast with ring type networks is sharp. If a ring network carries traffic on a back-up ring then network reconfiguration, due to either node or transport failure, introduces a 50 percent reduction in packet data capacity. Alternatively if the back-up ring carries no data traffic, there is

always a 50 percent inefficiency factor relative to the QPSX bus network. This is a critical factor in relation to cost efficiency, more particularly so in the public environment.

With QPSX it takes a real disaster, such as a backhoe through a physical transport, to invoke network healing. As explained, node failures do not invoke reconfiguration since faulty nodes are excised using very simple low complexity and low cost hardware. With rings however node failure invokes global healing.

Summary

QPSX is being developed as a public metropolitan area network and is in the IEEE Standards arena for 802.6 MAN Standardization. It has the inherent robustness and reliability necessary in a public network and it is being developed with management and healing features that materially enhance that reliability.

Next after reliability, an important issue is performance and in this respect QPSX is again quite outstanding. It uses a unique distributed queueing access control that permits fair access with the smallest possible delay and up to 100 percent loading. Moreover, the performance is independent of network rate or size.

The basic QPSX subnetwork is a pair of oppositely directed buses with station, bridge and gateway access units attached to both. Bus rates are up to 150 Mbits/s, with the streams divided into fixed length slots recurrently at the frame rate of eight kilohertz. Slots can be reserved for synchronous communications and those not reserved are available for packet-switched transfers.

A metropolitan area network will consist of many, possibly hundreds or even thousands, of QPSX subnetworks. These will be interconnected in an approximately hierarchical manner using bridges. MANs will be interconnectable to each other via gateways over a Wide Area Network. Customer premises networks or LANs will connect to a MAN also via gateways. Routing over the MAN and WAN will be by physical addresses of gateways, bridges and stations, related to E.164 addresses. Data packets up to eight kbytes in length will be transferred in segments over virtual circuits, a circuit normally existing only for the duration of a packet. Thus, with respect to packets the network will give a datagram or connectionless service. A connection-orientated packet service will also be possible.

Data transfer in QPSX resembles closely the transfer mode that has been decided by CCITT SGXVIII for B—ISDN making alignment of the two standards feasible. Complete alignment requires that the slot or cell as well as the segment or cell header sizes be equal in the two and that the headers encompass the product set of functions, expressed by identical codes and formats. Thus aligned, QPSX MAN would greatly speed the advent of B—ISDN.

References

[1] Draft proposed, IEEE Standard 802.6 metropolitan area network (MAN), QPSX dual bus media access control and physical layer protocol documents. Submitted July 1987.

[2] Z. L. Budrikis, J. L. Hullett, R. M. Newman, D. Economou, F. M. Fozdar, R. D. Jeffery, "QPSX: A Queued Packet and Synchronous Circuit Exchange," *Proc. 8th Int. Conf. Comp. Comm.*, Munich, 1986, pp. 288-293.

[3] R. M. Newman, J. L. Hullett, "Distributed Queueing: A Fast and Efficient Packet Access Protocol for QPSX," *Proc. 8th Int. Conf. Comp. Comm.*, Munich, 1986, pp. 294-299.

[4] F. E. Ross, "FDDI—a tutorial", *IEEE Comm. Mag.*, vol. 24, no. 5, pp. 10-17, May 1986.

Robert M. Newman was born on September 15, 1963. He received his Bachelor of Engineering with first class honors from the University of Western Australia in 1985, and has completed his Ph.D. research studies at the University of Western Australia. In both or these degrees Mr. Newman was involved in research work that led to the initial concept of QPSX switching and has also performed detailed modelling of the switch's operational characteristics.

Since April 1987, Mr. Newman has been employed as Projects Director, Network Systems for QPSX Communications Pty Ltd. In this role he has been responsible for the development of the initial QPSX switch products and has been actively involved in the standardization of dual bus technology in the IEEE 802.6 Committee. Mr. Newman is the co-inventor of the core QPSX technology.

Zigmantas L. Budrikis is a Director of QPSX Communications Pty Ltd and is on the Electrical Engineering Faculty in the University of Western Australia. He was born in Lithuania on September 29, 1932. He graduated B.Sc. majoring in physics in 1955 and B.E. with first class honors in electrical engineering in 1957, both from the University of Sydney, and Ph.D. in 1970 from the University of Western Australia.

Dr. Budrikis worked in the research laboratories of the PMG's Department (now Telecom Australia) from 1958 to 1960 and in Australia's Aeronautical Research Laboratories in 1961. Since 1962 he has been on the academic staff of the University of Western Australia. He was a visiting lecturer at U.C. Berkley in 1968, a visitor at Vilnius University at Technical University Munich in 1976-1977, an MTS at Bell Laboratories Holmdel in 1972 and 1974, and a consultant on four separate occasions at AT&T Bell Laboratories at Holmdel and Murray Hill over the period 1981 to 1985. He has worked extensively on problems of TV signal coding, human vision modeling, electromagnetic theory foundations, data communications and computer graphics. He is author of some forty papers and holder of several patents.

Dr. Budrikis is a Fellow of the Institution of Engineers, Australia and a member of IEEE, the New York Academy of Science, and the Optical Society of America.

John L. Hullett was born in Perth, Western Australia on December 20, 1942. He received his Bachelor of Engineering with first class honors in electrical engineering in 1965 and his Ph.D. in 1970 from the University of Western Australia.

Dr. Hullett worked in the research laboratories of Telecom Australia from 1969 to 1972 where from 1971 he was head of the Data Systems Division. Since 1972 he has been a member of the academic staff of the University of Western Australia and a member of the faculty of engineering. He has worked extensively in digital, data and optical communications, and networking and has published some forty five papers in these fields. He is a member of the IEEE.

Dr. Hullett is the co-inventor of the core QPSX technology and acts now as the Director of Strategic Planning and the Technical Division of QPSX Communications Pty Ltd. ■

Switched Multi-megabit Data Service and Early Availability Via MAN Technology

Christine F. Hemrick
Robert W. Klessig
Josephine M. McRoberts

Reprinted from *IEEE Communications*, Vol. 26, No. 4, April 1988, pages 9-14. Copyright © 1988 by The Institute of Electrical and Electronics Engineers, Inc. All rights reserved.

EH0301-2/91/0000/0068/$01.00 © 1988 IEEE

The widespread penetration of local area networks (LANs) has firmly established very high-speed data networking (1-10Mbits/s) in the local premises environment. The success of LANs has been synergistic with the rapidly growing use of more powerful PCs and workstations for high performance applications such as remote file serving and distributed processing. These market and technology trends portend an emerging market for multi-megabit communications services that extend beyond the local premises, across metropolitan and wide area environments.

This paper discusses work currently proceeding at Bell Communications Research in the field of novel high-speed data services and supporting network requirements [1]. The service described herein, switched multi-megabit data service (SMDS), is one which could be offered by carrier networks to meet emerging customer high-speed data needs. The use of Metropolitan Area Network (MAN) technology to provide high-speed packet switching in carrier networks appears to provide a promising opportunity for early availability of SMDS.

In this article there follows a section providing an overview of the motivations behind the service development, followed by a discussion of the importance of the embedded customer base and the factors that significantly influence the nature of the service. Next there is a description of specific SMDS features and characteristics. Finally, there is a discussion of the use of MAN technology for the support of SMDS.

Motivations for SMDS

Multi-megabit data communications have rapidly penetrated the workplace because of two complementary technology trends: 1) the availability of local area network (LAN) technology that provides high capacity, low latency performance and a low per-port interconnection cost; and 2) the dramatic, continuing decline in the cost of powerful computing devices.

These trends have led to the development of distributed processing techniques, providing real-time, shared access to files and executable software. The growth of distributed processing applications, such as desktop publlishing and CAD/CAM (which, prior to these technology advancements, was restricted to "stand-alone" type configurations) is expected to continue, due to the real benefits they bring to users in the areas of productivity, management and control, and "user-friendliness." However, the key to user acceptance and satisfaction with such systems is the very high-speed communication between devices that allows functions to be performed either locally or remotely without the user perceiving a difference in performance.

The need to respond to these market and technology trends has provided the principal motivation behind our work to define SMDS and the means for its early availability. The initial market for wide-area multi-megabit data communications is emerging now among larger and more sophisticated business and public sector customers. Thus, there is a need to make SMDS available early to meet key customer demands. For early deployment, the use of MAN technologies appears to offer a promising approach. The MAN trials recently conducted by NYNEX/New England

Telephone [2] and U.S. West/Northwestern Bell [3] have demonstrated the market and technical feasibility of delivering high-speed data service by employing MAN technology in the carrier network environment. While early availability is a key objective, attention must also be paid to the continuing growth in the wide-area, high-speed data market that is expected to occur in concert with progress toward the addition of Broadband capabilities for multiple services (for example, high quality video, high-speed data, etc.) into carrier networks. Thus, SMDS is also planned to be an initial offering of "multi-service" Broadband carrier networks. In fact, a key aspect of the service is that it has been defined independent of the underlying architecture and technology used to support it in the network. This is essential so that customers do not incur service discontinuity as the supporting network evolves from the use of early availability technology for SMDS to the introduction of technology supporting multiple Broadband services.

In addition to being a timely service concept from a market perspective, SMDS is intended to take advantage of two of the carrier networks' unique strengths: the wide-spread, rapidly growing deployment of single mode optical fiber and an extensive operations infrastructure. Use of this fiber to support SMDS alongside existing services has the potential for significant economies, over the situation where facilities must be dedicated to a single customer or service. The existence of the operations infrastructure is the basis for the economical provision of quality service, consistent with carrier network standards and customer expectations.

The Importance of the Target Customer Environment and Factors Influencing Service Characteristics

The potential SMDS customer is likely already to be a user of high-speed data communications on the local premises and will already have a substantial investment in hardware (hosts, terminals, workstations, LANs), software, applications and operations support. This substantial investment by the customer dominates the cost of related communications facilities and services. To introduce successfully a new data communication service to such customers, it is absolutely essential that the service not require substantial changes in the customer's existing communications environment. Rather, the service must be compatible with, and enhance the value of, the customer's embedded base. These are key factors that have influenced our service decisions.

Figure 1 represents a typical scenario in which SMDS might be used. The typical SMDS customer environment is likely to contain multiple networks, which may be either closely located or geographically separated. Each network constitutes a "subnetwork" in the customer's overall multiple network configuration. Typically today, such subnetworks will include local area networks, sometimes interconnected using wide-area subnetworks such as leased lines or X.25 packet-switched networks. Multiple LANs are usually inter-

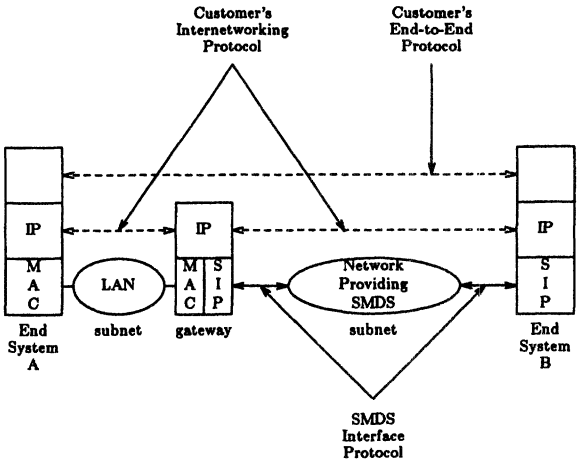


Fig. 1. Example scenario of the use of SMDS.

connected by means of "bridges" or "gateways." To achieve integration of the subnetworks into a larger internetwork, the customer equipment implements internetworking and transport protocols above the subnetworks in "gateways" and end systems. Commonly, the internetworking protocol will operate in a "connectionless" fashion, while the transport protocol will provide end-to-end reliability and control functions. Taken together, these provide a robust, error-controlled, flow-controlled service to the customer's distributed systems applications, which run over these protocols.

The most popular such protocol coupling, used for network interconnection, is one based on the DARPA-developed[1] internetworking protocols, popularly known as TCP/IP[2] [4], [5]. The TCP/IP protocol suite has long been the de facto standard for use in university and scientific and engineering environments, and in the Department of Defense networks for which it was originally developed. More recently, it has become an important factor in the commercial marketplace, emerging as the basis for building "multi-vendor" networks [6]-[8].

The TCP/IP architecture is very similar to the architecture for internetworking that comprises a part of the open systems interconnection (OSI) standards [9], [10]. In the OSI architecture an internetworking protocol, ISO 8473 [11], [12] positioned in the layer 3, network layer [13], [14] corresponds to the IP in TCP/IP. The OSI Transport protocol, ISO 8073 [15] includes one protocol class, "Class 4,"[3] that generally corresponds in functionality to TCP.

---

[1] DARPA is the Department of Defense Advanced Research Projects Agency

[2] "TCP" is an acronym for "transmission control protocol." "IP" is "internetwork protocol" or "internet protocol."

[3] The OSI transport protocol includes five classes of protocol, each of which provides a different degree of functionality and robustness. "Class 4" is the most robust, and includes the mechanisms for error detection and recovery, flow control, etc. that are required to provide an end-to-end, reliable transfer capability over a connectionless network environment.

A number of vendor-proprietary network architectures also closely correspond to the TCP/IP architecture and the OSI standards for internetworking. These include Digital Equipment Corp.'s DECNET/DNA, Xerox's XNS and Unisys' BNA. The type of architecture that these products reflect represents a now well-established and growing trend for internetworking, particularly where high-speed LANs are involved.

Role of SMDS in the Customer Environment

For SMDS to function in a subnetwork role as depicted, it must have a high degree of compatibility with the communications protocol architecture and the gateway philosophies presently used by its target customers. For example, in such environments it is important that gateways be relatively simple, and minimize the amount of resources (for example, buffers, processor cycles) used. Thus, we conclude that SMDS should offer a simple connectionless transfer mode, like a LAN, and that the interface to the network providing SMDS have simple procedures that are efficient for a gateway to utilize. Like the media access control (MAC) protocol of most LANs, the SMDS Interface Protocol should be easy to implement, by installing a new board interface and a simple software driver, for example.

Extension through SMDS

Not only should SMDS fit into the target customer's present environment, but its introduction into this environment has the primary goal of extending over a wide area the set of capabilities and performance the customer experiences in the local area.

The "end-user" applications found in the typical SMDS customer environment are very diverse. The availability of high-bandwidth communications on the local premises provides opportunities for functionality to be shared, with significant, positive consequences on the cost and method of doing business. For example, as mentioned earlier, applications that previously were accomplished by direct access to powerful processors (such as CAD/CAM, sophisticated graphics for publication layouts), can now be done using desktop workstations that communicate, via high-speed local networks, to supporting processors and peripherals. Files and executable programs can be stored on a powerful central "server" machine that can be accessed by multiple workstations. Additionally, if the user needs to access other software or data files, these can be quickly downloaded for use in the workstation, so that from a user perspective, they are as convenient and efficient as if they were resident in the workstation itself.

In today's environment, customers have limited choices, to extend the use of these applications beyond the local premises. One alternative is to resort to wide-area service offerings that support throughput at lower-than-LAN speeds, for example, leased line facilities at 56Kbits/s or 1.544 Mbits/s (T1). However, for the most part, these cannot meet the low delay requirements of high performance applications such as distributed processing. Another alternative is to acquire dedicated facilities that offer higher throughput and better delay characteristics but have equally higher cost, for example, private fiber optic networks or 44 Mbit/s (DS3) facilities. The objective of SMDS is to provide a more economical, shared network solution that would allow the use of high performance applications to be extended over a wide-area, and thus, no longer be limited by performance or by the feature/functionality discontinuity between widely separated subnetworks.

SMDS Requirements Summary

In summary, based on our view of the target customer, SMDS must exhibit the following general characteristics:

- features and characteristics similar to LANs;
- a simple, efficient interface; and
- high throughput and low delay performance.

Specific Features and Characteristics of SMDS

From the assumptions and objectives described above, we have derived a view of the specific features and characteristics that SMDS should exhibit in order to meet the target market.

Fundamentally, SMDS is a very high-speed connectionless or "datagram" packet-switched service. Typical of datagram services, it provides for the transfer of individual "units" of data, termed Service Data Units (SDUs), each of which has associated source and destination addresses. To efficiently serve high-speed data applications, SDUs have a variable length, up to a maximum of 8191 octets. Thus, the service can be used for applications which perform bridging between IEEE 802 standard LANs without the need to segment and reassemble MAC frames.

Source and destination addresses are delivered with each SDU, as specified by the sender. The network supporting SMDS validates all source addresses to ensure that the specified address is legitimately assigned to the interface from which the SDU was sent. Any SDU with a fraudulent source address is discarded.

Consistent with the support of SMDS by B-ISDN, addressing based on CCITT Recommendation E.164, the ISDN numbering scheme, is used to identify customer interfaces, with an additional provision for "group addressing."

Group addressing is analogous to the multi-casting feature of many LAN technologies, which is exploited by many equipment vendors in their communications software and applications. Its usage allows a single address to identify collectively a set of addresses. Incorporation of a group addressing feature in SMDS allows such vendor systems to adapt easily to using SMDS.

SMDS address screening can be used to allow a customer to restrict the reception of SDUs or the delivery of SDUs. Specifically, in *destination address screening*, all SDUs sent by the customer will have their destination addresses checked against a set of candidate destination addresses. In *source address*

screening, all SDUs to be delivered to the customer will have their source addresses checked against a set of candidate sources. Any SDU that doesn't match one of the candidate source or destination addresses, respectively, is discarded. Alternately, the screening features can each be provisioned to discard SDUs when screened addresses *do* match the candidate addresses in the screening table.

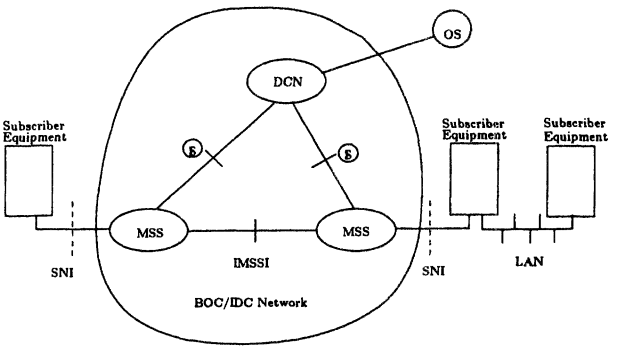
When the destination address screening and source address screening features are used in conjunction, a "logical private network service" may be offered. All customers in the "logical private network service" will subscribe to both screening features and specify in the address screening tables the addresses with which they are allowed to communicate. By tailoring the screening tables and permissions, a customer can also create variants of this "logical private network service."

Other addressing features included in SMDS are the ability to assign multiple addresses to a single customer interface, and some degree of mobility for addresses.

Another key feature of SMDS is the provision and enforcement of "Access Classes." Since most transport layer implementations provide for end-to-end flow control, it is unnecessary to include an explicit flow control mechanism across the interface to the network supporting SMDS. Moreover, such explicit flow control techniques impose penalties on the high performance of the service. However, since SMDS would be offered over very high-speed access paths, it is necessary to enforce limits on the rate of information transfer on the access, both in the subscriber-to-network direction, and in the network-to-subscriber direction. To provide for this enforcement, SMDS has various access classes and a "credit manager" mechanism. The credit manager is an algorithm implemented by the network which observes an implicit credit window based on predefined parameter values. The parameter values can be chosen to permit varying rates of sustained information transfer and different degrees of "burstiness." SDUs transferred from the subscriber to the network that violate the access class are discarded. In the network-to-subscriber direction, the network credit manager ensures that transfers always obey the access class. To avoid unnecessary data unit loss, the subscriber equipment can also implement and observe the access class parameters for transfers into the network.

The Use of MAN Technology for Early Availability of SMDS

To provide SMDS in a time-frame that will meet the emerging market, we have identified an approach which uses the BOC/IDC embedded base of multi-megabit fiber-based transmission systems, complemented by the use of MAN technology to provide very high-speed packet-switching. Figure 2 illustrates how MAN switching systems (MSSs) such as, switching systems based on MAN technology would be used as part of a network configuration to offer SMDS. The SMDS interface protocol (SIP) is defined across the subscriber-network interface (SNI). Within the network, MSSs are interconnected across an Inter-MSS Interface (IMSSI). Each MSS communicates with BOC/IDC Operations Systems (OSs) via a generic interface (*$*), across an operations Data Communications Network (DCN).



KEY

SNI: Subscriber-network interface
IMSSI: Inter-MAN switching system interface
(S) Generic interface for operations
MSS: MAN switching system
OS: Operations system
DCN: Data communications network
LAN: Local area network

Fig. 2. Network in support of SMDS.

Subscriber systems' access to the network supporting SMDS is by means of a dedicated DS3 path from the subscriber system to an MSS. Because a standard DS3 rate channel is used, the location of the MSS that terminates the access may be in the subscriber's serving central office (CO) or may be located in another CO (perhaps at a fiber hub point) in the network. This flexibility in locating MSSs is a key factor in efficient early deployment of SMDS since the customer base is likely to initially be small, but potentially geographically dispersed. The transmission facilities used for the access will be optical fiber based, and although the initial interface is being defined as an electrical DS3 interface (DSX-3), it is planned to define an optical SONET (Synchronous Optical Network) interface as soon as equipment conforming to the SONET standard is available. The use of digital transmission hierarchy standards such as DS3 allows existing operations capabilities for BOC/IDC transmission systems to be exploited, resulting in significant advantages for overall SMDS operations support.

The DS3 electrical interface represents the first level in the three level interface protocol, referred to as the SIP, SMDS interface protocol, as depicted in Fig. 3. Level 3 of the SIP associates appropriate addressing information with each data unit of subscriber information that traverses the interface. Level 3 also contains the algorithm for enforcing the subscribed-to "access class" thereby enforcing the agreed to limits on sustained information transfer and burstiness. Since

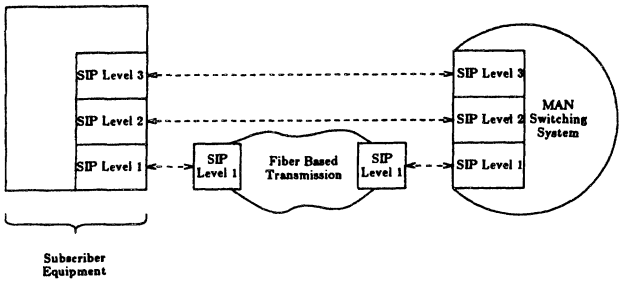


Fig. 3. SIP protocol stack.

this algorithm does not require explicit state synchronization between level 3 entities, the level 3 protocol is very simple. Framing of the level 3 protocol unit is performed by the level 2 of SIP which also includes a 32 bit CRC for error detection purposes.

Since the service (SMDS) expressed by SIP is "connectionless" or "datagram" in nature, no explicit flow control, sequencing, or retransmission procedures for errored data units are defined in the protocol. In fact, the interface that SIP provides to higher layer communications protocols in the subscriber systems is very similar to that available from most LANs. As a result, it is expected that existing and emerging CPE developed for use in high-speed LAN environments will be able to easily integrate and support the SIP for access to SMDS.

Inside the MSS, any one of several technologies could be used to meet the functional and performance criteria needed to offer SMDS. The network configuration illustrated in Fig. 2 is used to define the generic functions that an MSS must perform and the necessary interfaces it must meet, in particular the SNI and its protocol SIP, the IMSSI, and the generic interface for operations. No particular technology or architecture is prescribed for use inside an MSS and, in fact, the existence of a well-defined IMSSI provides for the deployment of MSSs that use different internal technologies. As an example, Fig. 4 illustrates two possible MSSs, one of which utilizes a shared medium ring topology and the other based on a looped bus arrangement. In each case, functionality of the MSS is distributed among components acting as stations on the shared media. (It is important to note that the shared medium which is used to transport the information of multiple subscribers is never directly accessible to any subscriber system. Components of the MSS terminate the shared medium MAC and deliver across the SNI only the information intended for the particular subscriber. It should also be noted that the focus of the MSS is solely on providing "data only" service and thus any capability of the MAN technology for supporting voice traffic is not exploited in providing SMDS.

As described here, our detailed work to date prescribing early availability of SMDS has focused on a precise technical service description, the framework architecture for describing generic functions and interfaces as shown in Fig. 2, details of the SIP, and use of the

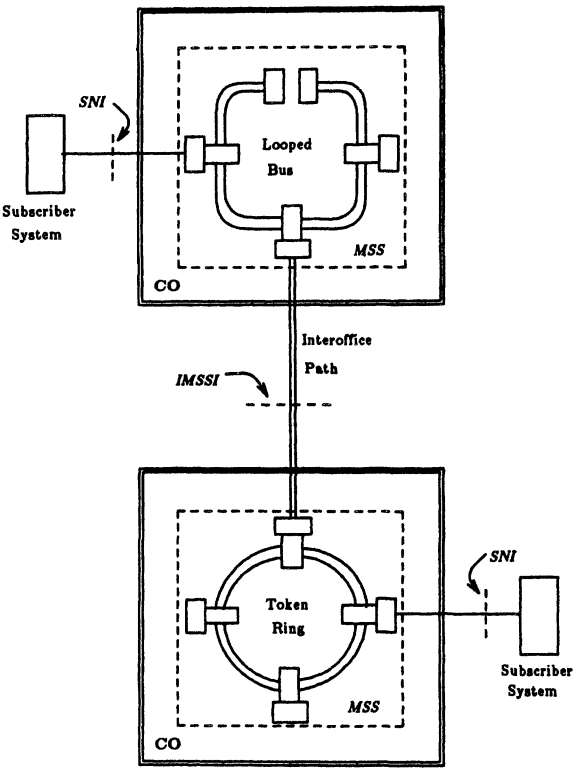


Fig. 4. Examples of MAN technology in MSSs.

existing BOC/IDC embedded transmission plant. Work is proceeding at Bellcore in additional areas such as the definition of the protocol between MSSs across the IMSSI and the operational requirements associated with the introduction of very high-speed packet-switching capabilities and the MSSs.

Summary

High-speed data networking across metropolitan and wide area environments represents an emerging market and a potential service opportunity for carrier networks. SMDS is a service concept that is intended to address this market by respecting the embedded and growing customer base of hardware, software and applications and extending the use of high performance applications beyond the limits of the local premises. The SMDS service concept has been defined so as to be suitable for deployment with different supporting network architectures and technology bases. Thus, the use of MAN technology to bring very high-speed packet switching to the carrier networks appears to provide a promising approach for early availability of SMDS.

Acknowledgments

The authors wish to acknowledge the other members of the technical service description team for SMDS, Jon Shantz, Kiho Yum, Patrick Sher, and Joe Lawrence.

References

[1] "Metropolitan area network generic framework systems requirements in support of switched multi-megabit data service," Technical Advisory TA-TSY-000772, Bell Communications Research, Inc., issue 1, Feb. 1988.

[2] NYNEX tests metro area network," *Telephony*, June 30, 1986.

[3] "Metro-area net to be tested," *Network World*, Sept. 22, 1986.

[4] "Internet Protocol," MIL-STD 1777, *DDN Protocol Handbook*, vol. 1, DCA, Dec. 1985.

[5] "Transmission Control Protocol," MIL-STD 1778, *DDN Protocol Handbook*, vol. 1, DCA, Dec. 1985.

[6] "COMNET '87: The 'Network Solution' show," *LAN: The Local Area Network Magazine*, Apr. 1987.

[7] "Big turnout expected at TCP/IP conference," *Communications Week*, March 9, 1987.

[8] "IBM embraces TCP/IP," *Network World*, May 25, 1987.

[9] ISO 7498-1984(E), Information processing systems—Open Systems Interconnection—Basic reference model, 1984.

[10] "Recommendation X.200, reference model of Open Systems Interconnection for CCITT applications," *CCITT Red Book*, vol. VIII.5, 1985.

[11] ISO 8473-1987(E), Information processing systems—Telecommunications and information exchange between systems—Protocol for providing the connectionless-mode network service, 1987.

[12] A. L. Chapin and D. M. Piscitello, "An International Internetwork Protocol Standard," *J. Telecommunications Networks*, vol. 3, no. 3, Fall 1984.

[13] ISO 8648-1987(E), Information processing systems—Telecommunications and information exchange between systems—Internal organization of the network Layer, 1987.

[14] C. Hemrick, "The internal organizzation of the Network Layer: Concepts, Applications and Issues," *J Telecommunications Networks*, vol. 3, no. 3, Fall 1984.

[15] ISO 8073-1984(E), Information processing systems—Open Systems Interconnection—Transport protocol, 1984.

Christine Hemrick is District Manager of Data Services Concepts in the Network Planning Area at Bell Communications Research. This district has responsibility for new service concept development, with a particular focus on high-speed data and services for broadband networks. She received a B.S. degree in mathematics from North Carolina State University in 1974.

Prior to joining Bellcore in 1985, Ms. Hemrick was a senior staff member with the National Telecommunications and Information Administration's (NTIA) Institute for Telecommunication Science (ITS) in Boulder, Colorado. While with NTIA, she served on behalf of the U.S. State Department as coordinator of U.S. participation in CCITT Study Group VII, and as head of the U.S. delegation to that study group. Previous to her NTIA experience, she held positions at Digital Equipment Corporation and GTE Telenet.

Robert Klessig is District Manager of Protocol Architecture Planning at Bell Communications Research. The district is involved in formulating internal network protocol architectures for both data transport and network control. He is also an active participant in IEEE Project 802.6, Metropolitan Area Network Standards. He received his B.S. in Engineering from Case Western Reserve University, and his M.S. in electrical engineering and Ph.D. in electrical engineering and computer sciences, both from the University of California, Berkeley.

Previous to the formation of Bellcore, he was a member of the technical staff and then supervisor at Bell Telephone Laboratories working in the areas of network facility planning, special services planning and data communications network planning. He has also been an acting assistant professor and visiting lecturer in the Electrical Engineering and Computer Sciences Department at the University of California, Berkeley.

He has published papers in the areas of nonlinear programming, optimal control, facility planning, and data communications protocols. He is a member of IEEE, SIAM, ORSA, and the Mathematical Programming Society.

Josephine McRoberts is a member of technical staff at Bell Communications Research, in the Data Services Concepts District. She has been involved in broadband services research, including video and high speed data services, with an emphasis on target applications. She attends the Broadband subworking groups of T1S1 and CCITT SGXVIII. She received her BSc in applied physics from the University of Strathclyde, Scotland in 1984 and a MSE in systems engineering from the University of Pennsylvania in 1985.

She is a member of IEEE and the U.K. Institute of Physics. ■

Packet mode services: from X.25 to frame relaying

Wai Sum Lai discusses the relationship and interworking between frame relaying and X.25

Since its inception, the X.25 standard has become a widely accepted and deployed data service in both public and private networks. X.25 has also been defined as the packet mode service in Integrated Services Digital Networks, because of its current utility and the need to ensure compatibility with existing public data networks. The move to out-of-band call control with ISDN creates possibilities for the provision of additional packet mode services such as frame relaying. The relationship and interworking between X.25 and frame relaying services are discussed here.

Keywords: computer networks, packet switching, Integrated Services Digital Networks, X.25

The only standardized packet mode service currently being offered in an Integrated Service Digital Network (ISDN) is the X.25-based virtual circuit. Also under development at the CCITT, and various other standards bodies, is the provision of additional ISDN packet mode services.

The historical development of packet mode services at the CCITT is briefly traced, then the architectural and functional characteristics of the newly proposed additional ISDN packet mode services, with particular emphasis on frame relaying service, is presented. Finally, how frame relaying can interwork with the existing large installed base of X.25 is discussed.

HISTORICAL PERSPECTIVE

The utility and usage of computer equipment in the workplace has changed dramatically over the past two or three decades — from a centralized model of a large, single central processing unit with small numbers of appending input/output devices, to a distributed model in a multivendor environment. As these changes occurred, the need for a cost-effective method of communication between dispersed machines arose. Packet switching addressed this need very well, with the inherent benefits of:

- interconnection flexibility between dissimilar types of equipment;
- data transmission integrity through error detection/correction;
- speed matching; and
- providing a shared communications facility over wide geographic areas.

Ever since the potential for packet switching was demonstrated by the Advanced Research Projects Agency (ARPA) network in the late 1960s and early 1970s, much international activity for the standardization of packet switching services was followed. As a result, in 1976, the CCITT published the then new X.25 Recommendation (*Interface between Data Terminal Equipment and Data Circuit-Terminating Equipment for Terminals Operating in the Packet Mode on Public Data Networks and Connected to Public Data Networks by Dedicated Circuits*) for packet mode communications based on virtual circuit service.

X.25 has gained wide acceptance. Since the mid 1970s, numerous X.25-based Packet Switched Public Data Networks (PSPDN) have been implemented all over the world. To ensure the global ubiquity of packet mode services, it is critical that these networks be interconnected in a standard way. To address this need, the X.75 Recommendation (*Terminal and Transit Call Control Procedures and Data Transfer System on International Circuits between Packet-Switched Data Networks*) was produced in 1980, where the internetwork signalling procedures between PSPDNs are specified.

With the advent of ISDN[1] in the 1980s, attention has focused on the provision of useful packet mode services

AT & T Bell Laboratories, Crawfords Corner Road, Holmdel, NJ 07733, USA

"Packet Mode Services: From X.25 to Frame Relaying" by W.S. Lai from *Computer Communications*, Vol. 12, No. 1, February 1989, pages 10-16. Copyright © 1989 by Butterworth & Heinemann Publishers Ltd.

in that environment. Because of the necessity to provide interworking with current packet data services, and the fact that X.25-based PSPDNs are well established worldwide, X.25 has been standardized as the packet mode service that can be offered in an ISDN (Recommendation X.31/I.462: *Support of Packet Mode Terminal Equipment by an ISDN*).

To ease the provision and use of different telecommunications services, out-of-band call control is employed in ISDNs. This control method is based on the principle of separation of the control and user planes in the ISDN protocol reference model (Recommendation I.320: *ISDN Protocol Reference Model*). The concept of separate control and user planes is used to distinguish interactions associated with control and signalling functions from those needed to transfer user data.

However, X.25 in the ISDN environment as specified in Recommendation X.31 uses in-band, rather than out-of-band, procedures for the control of virtual circuits. Such an approach is being taken for pragmatic reasons — to offer access to packet data services through an integrated physical interface while minimizing deployment and interworking difficulties.

The evolution of packet mode services in ISDN is currently under investigation in various national and international standards forums. In CCITT Recommendation I.122 (*Framework for Providing Additional Packet Mode Bearer Services*), the provision of additional packet mode services utilizing out-of-band call control has been proposed. Before discussing these new services, let us first review the current ISDN packet mode services of X.31, and the applicability of out-of-band call control.

X.31 PACKET MODE SERVICES

Recommendation X.31 defines the procedures for the support of existing packet mode terminals accessing an ISDN. In this support of X.25 Data Terminal Equipment (DTE), two scenarios are specified:

1 In the so-called *minimum integration scenario* (see Figure 1a), X.25 packet calls are handled transparently through an ISDN. In effect, the ISDN provides only a physical digital pipe between the X.25 user terminal adapter at the customer's premises, and an appropriate interworking port in an existing PSPDN. With the ISDN being operated as a Layer 1 (physical layer) relay, X.25 Layer 2 (link layer*) and Layer 3 (packet layer) functions are performed outside the ISDN.

2 In the *maximum integration scenario* (see Figure 1b), a packet handling function is provided within an ISDN. In conjunction with such a Packet Handler (PH), an ISDN appears to a PSPDN either as another PSPDN, or as an extension of the same PSPDN. As a result, once a physical connection is established between an X.25 terminal and a PH in an ISDN, the complete processing of the X.25 packet call can be carried out within the ISDN.

Thus, in the minimum integration case, the X.25 DTEs can be considered to be only physically connected to an ISDN, but logically connected to a PSPDN. In the maximum integration case, the DTEs are both physically and logically connected to the ISDN where the X.25 call is

*In Open Systems Interconnection (OSI) terms, Layer 2 is called the data link layer. Here it is referred to simply as the link layer.

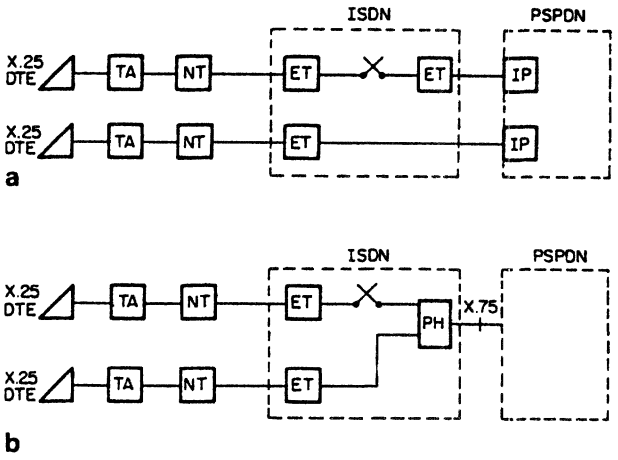


Figure 1. X.31 packet mode services — access arrangements: (a) access to PSPDN services, (b) access to ISDN virtual circuit bearer service. TA: terminal adapter; NT: network termination; ET: exchange termination; IP: ISDN interworking port; PH: packet handler

processed. In view of their different capabilities, these two scenarios have now been renamed as *access to PSPDN services* and *ISDN virtual circuit bearer service*, respectively.

In both types of service, existing in-band call control procedures in X.25 continue to be employed for the establishment and release of virtual circuits†, i.e. the complete packet layer (Layer 3) of X.25 is used for both call control and data transfer phases of a packet call. What X.31 does is really just the encapsulation of existing X.25 packet mode services in an ISDN envelope at the physical layer.

With the X.25 packet layer procedures remaining intact, the number of modifications required to accommodate existing X.25 DTEs in an ISDN is minimized. In this way, a large population of PSPDN users can easily migrate to ISDNs as these networks are deployed. A smooth transition for both carriers and users is thus ensured as a result of the consistency and uniformity between PSPDN and ISDN packet mode services.

OUT-OF-BAND CONTROL

While in-band call control is being specified in Recommendation X.31 for the support of X.25 to minimize deployment and interworking difficulties, out-of-band call control has been adopted for the additional ISDN packet mode services currently under investigation by the CCITT, and described in Recommendation I.122.

As detailed in I.122, this use of out-of-band control is made in recognition of the fundamental principle of separation of the control and user planes for all telecommunications services in ISDNs. Adherence to this principle enables ISDNs to achieve the goal of service independence: the same signalling mechanisms as other ISDN bearer services will be employed by the additional

† In the case of circuit-switched B-channel access to packet mode service, out-of-band control based on Q.931 message exchange (*ISDN User-network Interface Layer 3 Specification*) over the signalling D-channel is first used for the establishment of a circuit-switched physical connection. After this, the in-band X.25 procedure is used for the establishment of a virtual circuit over the physical connection.

ISDN packet mode services in order to completely set up packet calls.

In most existing transport networks there is no clear distinction between the paths for the control and user information streams. The significant advantage of this approach is that, since network control tends to be intimately tied in with the associated information carriage, the resulting network design can have a high degree of optimization. The consequence of this is that different information carriages will require different forms of network control, because of different control requirements. A close coupling between information and control also leads to a lack of the flexibility needed to cater for new signalling and control needs.

With the ever-increasing demand for data communications within the business, and to some extent the residential, environment the role of the network is changing from that of a predominantly voice network to one which is more responsive to other new services. The acceptance and deployment of digital switching and digital transmission throughout the public telephone networks has primed the network with the basic capabilities to handle traffic other than voice. To ensure maximum efficiency of the network, separation of signalling control from the information carriage is desirable.

Major benefits derived from this separation include:

- there is the potential for the integration of control for voice, data and other new services. As a result, there can be sharing across all bearer services of functional units, providing, e.g. call control, supplementary services and higher layer capabilities. So, as new bearer services are added, the proliferation of signalling types and procedures can be avoided. Also, as new supplementary services are added, they could possibly be applied to all bearer services;
- since the information path no longer needs to worry about control, its logic can be substantially simplified — this implies that hardware implementation can now be exploited more fully;
- with the separation of signalling, the provision of multimedia calls can be accommodated — this type of call can have very different attributes associated with different portions of a single call, or for the two directions of a single call;
- different qualities of service are usually required by information transfer and signalling control. Their separation therefore allows for separate network provisioning to cater for their different needs, e.g. the design of functional units used for signalling functions are no longer constrained by the bearer channels with which they may be associated for call supervision. In the same vein, the attributes of requested bearer services can be based only on information transfer requirements, and not on signalling;
- in addition to independent optimization, independent evolution of control and transport networks can also be accommodated.

I.122 ADDITIONAL PACKET MODE SERVICES

In Recommendation I.122 four types of additional ISDN packet mode bearer services are defined: frame relaying 1; frame relaying 2; frame switching; and X.25-based additional packet mode.

From a high-level architectural viewpoint, this family of new services is built upon a common framework that aims at service independence by using the concepts of out-of-band signalling and link layer multiplexing. What differs among them is the extent of functionality provided by a network during the data transfer phase.

Control and user planes

As discussed above, all I.122 services adopt out-of-band signalling. Specifically, for the establishment and release of packet-mode calls, Q.931 messages over the signalling D-channel are used. These procedures are in the Control plane (C-plane). After call establishment, all procedures for the transfer of user data are in the User plane (U-plane).

According to Recommendation I.122, such a separation of C-plane procedures from those in the U-plane can occur in one of several ways: on a physically separate interface; on another channel (time slot) within the same interface; or on a separate logical link within the same channel (e.g. the D-channel).

Link layer multiplexing

In the U-plane, in order to facilitate the sharing of bandwidth resources among different user data streams, the concept of link layer multiplexing is used.

In a conventional packet network, a link entity typically does not provide any multiplexing function. Multiplexing in X.25 is achieved through the use of logical channels in the packet layer. So, effectively it is Layer 3 that provides the switching function.

In I.122 services, switching is done in the link layer through the statistical multiplexing of different *data link connections* on the same physical channel. This support of multiple data link connections in the link layer is based on the multiplexing operation specified in Recommendation Q.921/I.441 (*ISDN User-network Interface Data Link Layer Specification*) (Link Access Protocol on D-Channel (LAPD)). With LAPD as a basis, the creation of a new protocol is therefore not required.

Network functionality

With link layer multiplexing in the U-plane during the data transfer phase as a foundation, the family of I.122 services is offered by a network through different degrees of protocol support, i.e. different levels of protocol termination at the network edges after call establishment. Figure 2 shows the different protocol break points, or points at which a network can terminate the protocol(s) in order to provide the requested bearer services. It can be seen from this figure that, in terms of the network functionality provided, frame relaying requires the least, while X.25-based additional packet mode the most.

In *frame relaying service* (see Figures 3a and b), the network supports only the minimal, or the so-called *core*, set of functions at Layer 2 that are needed to transfer frames between communicating terminals during the data transfer phase. It suffices for now to note that these functions essentially ensure that only frames with a valid format and address (i.e. data link connection identifier) are delivered. All other Layer 2 functions, as well as those

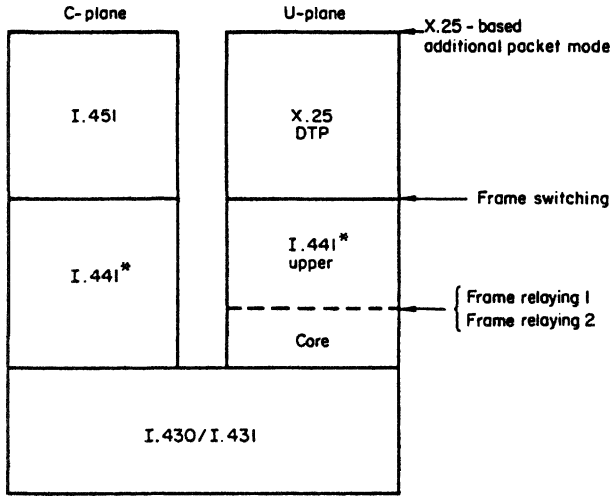


Figure 2. Functionality of an ISDN node at the user-network interface for the provision of I.122 services. Core: I.441 core functions; I.441: I.441 extended to cover I.122 requirements; I.441* upper: elements of procedures specified in I.441* (i.e. upper part of I.441*); DTP: data transfer part of X.25 packet layer procedures*

at Layer 3 and above, are transparent to the network. They are therefore operated on an end-to-end basis by the users themselves. (In addition to link layer multiplexing which is provided by the network, a user may also choose to perform Layer 3 multiplexing within a particular data link connection. But such an operation will be transparent to the network.)

There are two types of frame relaying service. In *frame relaying 1 service* (see Figure 3a), any user-specified link layer end-to-end protocol above the core functions can be used. For *frame relaying 2 service* (see Figure 3b), the elements of procedures as specified in Recommendation Q.921/I.441 are to be used end-to-end.

As a result of the fact that the network provides only the core functions of the link layer, a user's link layer can be viewed as split into two parts: the lower core functions and the upper end-to-end procedures. This decomposition is conceptually very similar to the Medium Access Control (MAC) and the Logical Link Control (LLC) sublayers of the link layer in the IEEE 802 Local Area Networks (LAN).

In *frame switching service* (see Figure 3c), a user's link layer protocol in the U-plane is specified to be Q.921, and this is fully terminated by the network. So, only Layer 3 and above are considered to be end-to-end.

In *X.25-based additional packet mode service* (see Figure 3d), Q.921 and the data transfer part of the X.25 packet layer procedures are specified to be a user's Layer 2 and 3 protocols, respectively, in the U-plane. Both these protocols are fully terminated by the network. Observe that only the data transfer part of the X.25 packet layer procedures is required here. This is because the Q.931-based out-of-band signalling method eliminates the need for the call control part of the X.25 packet layer procedures.

Relationships between different services

A closer examination of Figures 3a, b and c reveals some interesting relationships between the frame relaying 1,

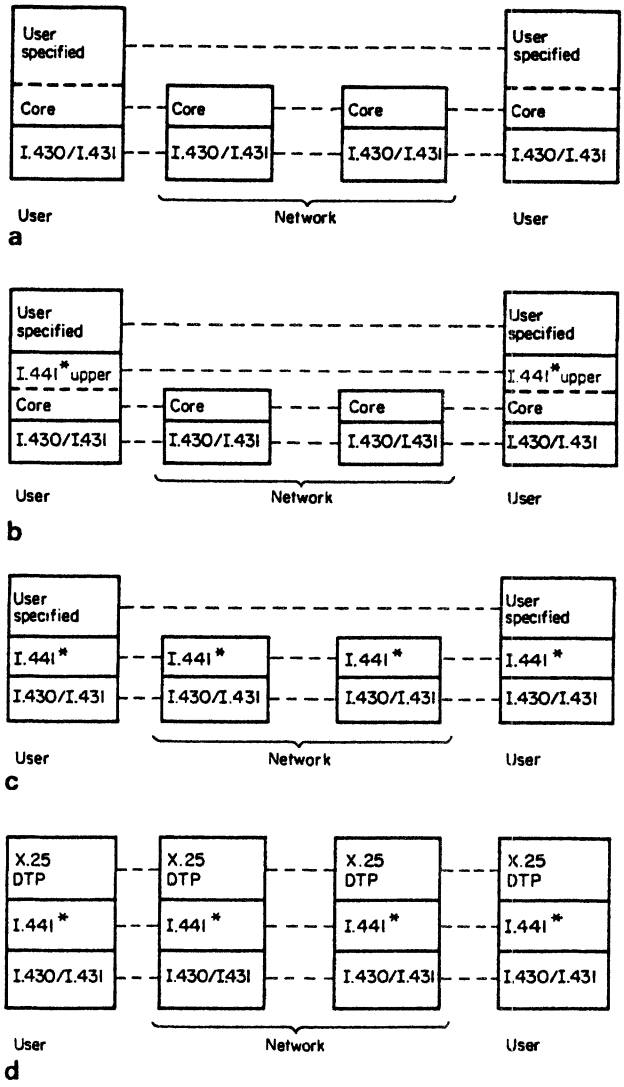


Figure 3. I.122 additional packet mode services: U-plane procedures. (a) Frame relaying 1 service, (b) Frame relaying 2 service, (c) Frame switching service, (d) X.25-based additional packet mode service. Core: I.441 core functions; I.441: I.441 extended to cover I.122 requirements; I.441* upper: elements of procedures specified in I.441* (i.e. upper part of I.441*); DTP: data transfer part of X.25 packet layer procedures*

frame relaying 2 and frame switching services. To see these we take two orthogonal viewpoints in the U-plane: a vertical cut across the protocol stack at the user-network interface; and a horizontal split along the boundary between Layers 2 and 3 of a user.

At the user-network interface, from the perspective of a network there is really no significant difference between frame relaying 1 and frame relaying 2, since only core functions are provided by the network in both cases. The difference between them mainly stems from the standpoint of OSI layer service. This is because, depending on the specific Layer 2 protocol being used, a different Layer 2 service will be provided to Layer 3. So, with the specification of Q.921 as the Layer 2 protocol, the Layer 2 service of frame relaying 2 is completely specified by I.122. For frame relaying 1 with the user-specified Layer 2 protocol, the Layer 2 service is defined, not by I.122, but

by the users themselves. So, it can be different from one user connection to another.

In this sense, one can say that both frame relaying 2 and frame switching provide the same Layer 2 service, since Q.921 is specified as the Layer 2 protocol in both cases. However, as noted earlier, their difference arises from the different levels of protocol termination at the user-network interface. As a result of such a difference, even though the Layer 2 services provided by frame relaying 2 and frame switching respectively are the same in terms of functionality, there can be differences in the performance between the two Layer 2 services in terms of delay or throughput.

Why different services?

Having described the different I.122 services, it is now appropriate to discuss the rationale for defining the range of services as above. One reason is that such a family can probably be tailored appropriately to the needs of the growing diversity of user application. Indeed, the original intention behind developing this family was to offer not only an 'X.25-like' service to complement the existing X.31 packet mode services, but also to support new applications such as the interconnection of LANs, large file transfers, high resolution graphics, etc.

By defining a spectrum of services it can be envisioned that the users are given the 'flexibility' to choose, according to the needs of different applications, different terminations of protocols by the network in order to obtain the different grades of service desired. However, such a freedom can potentially lead to a complex interface with significant management problems for both network providers and users. The different types of protocol termination can also result in potential confusion on the user's part, and possibly inefficient use of the services provided by an ISDN.

It should be noted that one of the principles of ISDN is that a broad spectrum of applications should be supported by the same network using a limited number of connection types and multipurpose user-network interface arrangements.

With these considerations in mind, it is desirable to limit the number of bearer services provided by an ISDN. The current view is that not all of these services may need to be provided by a network. A new question for the CCITT in the next Study Period (1989–1992) is to determine which service(s) should be retained.

FRAME RELAYING SERVICE

The relative merits of the above services notwithstanding, one type of service which stands out most is frame relaying. When compared to the others, the protocol functionality required at the user-network interface for frame relaying is much reduced. This service would therefore permit less per-frame processing by a network. As a result, less delay and higher throughput could be expected. Because of this promise, frame relaying is currently attracting much attention.

Following this emphasis, the focus in the rest of the paper is on the frame relaying service. Furthermore, the discussion is based on the frame relaying 1 service in view of its generality in terms of the user link layer end-to-end protocol.

To simplify subsequent description, the term 'frame relaying 1 service' is abbreviated to *FR service*. By the term *Frame Relaying network (FR network)* we mean an ISDN that provides, on a per-call basis, FR service.

For the support of FR service, the link layer within a FR network provides only the so-called *core functions of I.441*. For discussion purposes, these core functions can be logically separated into two groups: frame integrity control and frame multiplexing.

Frame integrity control includes the following basic functions associated with an individual frame: frame delimitation, e.g. by High-level Data Link Control (HDLC) flags; transparency of user information, e.g. through the use of HDLC zero insertion/deletion; and frame integrity protection, e.g. by HDLC Frame Check Sequence (FCS).

These types of processing are done on a frame-by-frame basis. Building on this, the frame multiplexing function is responsible for the interleaving of, and discrimination between, the frames from different data link connections on the same physical channel.

To achieve multiplexing, each data link connection within a given physical channel is being assigned a unique identifier, called the *Data Link Connection Identifier* (DLCI). Such a DLCI only has a local significance, i.e. its value is only unique within a physical channel.

Switching in the link layer is achieved by binding the DLCIs to routing information at intermediate nodes to form a set of network edge-to-edge logical paths[2]. These bindings at intermediate nodes are established at call setup before the start of data frame transfer, and released at call clearing.

In this way, a connection-oriented link layer service is offered to the users of a FR network. The characteristics of this service are: preservation of the order of frame transfer from one network edge to another; non-duplication of frames; and a (very) small probability of frame loss.

INTERWORKING WITH X.25

To prevent isolated islands of subscribers from being created, the provision of connectivity to the existing user base must be a key aspect of any new service. So, a complete definition of a new packet mode service such as FR should include a specification not only of the services to be provided at the network access interface, but also of that provided when interworking with existing networks.

FR and X.25 are two different network access protocols: they provide different types of service to a user. The numbering plans for the two networks are also potentially different, since FR users can be identified according to the ISDN numbering plan E.164 (*Numbering Plan for the ISDN Era*) rather than the public data network numbering plan X.121 (*International Numbering Plan for Public Data Networks*).

An *Interworking Function (IWF)* is therefore required to resolve the differences so that a user on a FR network can communicate with a DTE on an X.25 network (by the term X.25 network either a PSPDN or an ISDN offering X.25 service based on X.31 is meant). Such an IWF can be provided either by a FR network or an X.25 network.

Conceptually, an IWF can also be configured as an intermediary between a FR network and an X.25 network. In this case, the IWF is separate from either network. However, such a non-integrated approach may cause complication in the management of internetwork connections, especially in the procedures for maintenance,

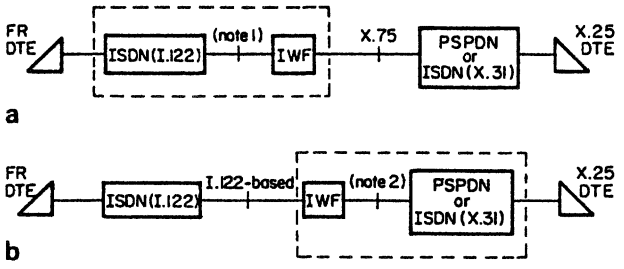


Figure 4. FR-X.25 interworking arrangements. (a) IWF logically belongs to ISDN (I.122), (b) IWF logically belongs to PSPDN or ISDN (X.31). (Note 1: In the control plane, signalling system No 7, I.451 with appropriate extensions, or proprietary protocols with equivalent functions may be used. Note 2: X.25 or equivalent proprietary protocols may be used.) -----: logical grouping

traffic planning and accounting. This strategy is therefore not considered any further.

FR-X.25 INTERWORKING ARCHITECTURES

We first discuss the case when the IWF logically belongs to a FR network (see Figure 4a). As described above, such an IWF must bridge the difference in layer services between a FR network and an X.25 network.

Using the terminology of Recommendation X.300 (*General Principles and Arrangements for Interworking between Public Data Networks and between Public Data Networks and other Public Networks*), the association of a FR network with an IWF could be considered globally as a *Type I subnetwork**. Such an association should offer the full capability of OSI (Open Systems Interconnection) connection-oriented network layer service. (An X.25 network is by definition a Type I subnetwork.)

As specified in X.300, the interworking arrangement between two Type I subnetworks should be based on Recommendation X.75, i.e. if a FR network is to provide the IWF, then the FR-X.25 interworking procedure should be X.75-based.

Alternatively, an X.25 network can offer an adapter which provides the IWF (see Figure 4b). Such an adapter would resemble a user requesting FR service from a FR network. Hence, I.122-based procedures, appropriately extended to cover network-to-network interface requirements, can form the basis for FR-X.25 interworking.

In this interworking arrangement, a FR terminal attached to a FR network first accesses an X.25 network by using such I.122-based procedures to set up a FR call via the FR network to the adapter in the X.25 network†.

After the establishment of such a FR connection, the data path between the FR terminal and the adapter will be totally transparent to the FR network. As described above, any user-specified Layer 2 and Layer 3 end-to-end protocols can run atop the core functions provided by the FR network. In particular, the FR terminal can now establish an X.25 virtual circuit to a remote DTE on the X.25 network via the existing in-band X.25 packet layer procedures for call control.

Similar two-stage calling procedures apply for calls originating from an X.25 DTE to a FR terminal. A FR connection from the adapter to the FR terminal must first be established over the FR network before any X.25 packet layer call requests can be passed over to the called FR terminal.

Note that under this scheme, X.25 virtual circuits are being multiplexed on a single FR connection. However, it is possible for both Layer 2 and Layer 3 multiplexing to co-exist, e.g. a FR terminal may simultaneously have multiple FR connections to the same or different X.25 networks on a single bearer channel, with each FR connection carrying one or many X.25 virtual circuits. Similarly, an X.25 network may have multiple FR connections to multiple FR terminals, over one or many bearer channels.

COMPARING X.75/I.122 INTERWORKING ARRANGEMENTS

To recapitulate, the strategies that can be used for FR-X.25 interworking are: OSI network layer interworking through X.75; and I.122-based interworking. Under X.75 interworking, a FR-X.25 internetwork connection consists of three segments: a FR segment inside the FR network; a virtual circuit segment through the X.75 internetwork link; and another virtual circuit segment inside the X.25 network. In contrast, there are only two segments in I.122-based interworking: a FR segment from the FR network user to the adapter, and a virtual circuit segment from the adapter to the X.25 DTE. Obviously, this makes the data transfer phase more efficient.

Another aspect is that X.75 is based on in-band signalling. I.122-based procedures provide out-of-band call control, at least for the establishment of FR connections. It is therefore in line with ISDN philosophy.

While there are advantages in using I.122 in the long-term, X.75 provides a number of immediate benefits. In particular, the X.75 strategy provides:

- backward compatibility: in principle, no changes to the installed X.25 networks will be required;
- an interworking protocol that is proven and widely supported (furthermore, E.164/X.121 interworking has essentially been resolved for X.75); and
- a basis for ensuring consistency of the OSI network layer services.

Therefore, X.75 should also be the basis of interconnection. This will ensure the interworking of any new FR networks with the large established base of X.25 networks.

In the I.122-based interworking scheme, X.25 networks are being accessed by multiplexing X.25 virtual circuits over a FR connection. Conceptually, such an access mechanism is very similar to the use of X.31 to access PSPDN services (i.e. the minimum integration scenario, see above). The essential differences between these two methods, when the X.25 calls are being carried on bearer channels, are the use of Q.931 to establish a FR connection rather than just a bearer channel, and the use of LAPD on the bearer channel instead of LAP on B-channel (LAPB). Therefore, with minor extensions to X.31, existing packet-mode terminals can use FR as an additional mechanism for accessing X.25 networks.

*Based on the network support for the OSI connection-oriented network layer service, four basic categories of subnetworks are considered in X.300. These are referred to as Types I, II, III and IV subnetworks. A Type I subnetwork is one that provides functionality equivalent to that of a PSPDN.

†In X.300, this method of interworking is called *port access*.

CONCLUSION

The evolution of packet mode services from X.25 with in-band control and packet layer multiplexing to FR with out-of-band control and link layer multiplexing has been presented.

FR is still currently under development at the CCITT and various other standards bodies. To ensure the success of FR, its interworking with the existing X.25 services must be developed, and for this purpose both X.75- and I.122-based interworking alternatives can be used.

ACKNOWLEDGEMENTS

The author would like to thank Wing-Man Chan, Ian Cunningham, David Delaney, Ken Haywand, Bam Liem, Stephen Ng, Andy Nichol and John O'Connel for their help in this work.

REFERENCES

1 **Rutkowski, A** *Integrated Services Digital Networks* Artech House, USA (1985)

2 **Lai, W S** 'Packet forwarding' *IEEE Commun. Magazine* Vol 26 No 7 (July 1988) pp 8–17



Wai Sum Lai *received his BSc degree in electrical engineering from the University of Hong Kong in 1970, his MSc degree in information and computer sciences from the University of Hawaii in 1974, and his PhD degree in systems and computer engineering from Carleton University, Ottawa, Canada in 1987. From 1970 to 1971 he was a consumer radio engineer for TransWorld Electronics Ltd., Hong Kong. Between 1971 and 1975 he was a research assistant for the ALOHA System at the University of Hawaii. From 1975 to 1988 he worked with Bell-Northern Research, Ottawa, Canada. He has recently moved to AT&T Bell Laboratories, Holmdel, NJ, USA.*

Reprinted from *IEEE Communications*, Vol. 27, No. 9, September 1989, pages 17-24. Copyright © 1989 by The Institute of Electrical and Electronics Engineers, Inc. All rights reserved.

Broadband ISDN and Asynchronous Transfer Mode (ATM)

Steven E. Minzer

THE EMERGENCE OF HIGH-SPEED TELECOMMUNICATIONS technologies has provided an impetus for the definition of Broadband ISDN (BISDN) [1-5]. BISDN is conceived as an all-purpose digital network. It will provide an integrated access that will support a wide variety of applications for its customers in a flexible and cost-effective manner [6]. The network capabilities will include support for:

- *Interactive and distributive services*—The network serves as a common carrier of both interactive and distributive communications, which may include audio, video, and data.
- *Broadband and narrowband rates*—Switched services at rates of up to about 140 Mb/s, with the possibility of raising this limit in the future.
- *Bursty and continuous traffic*—Some services (e.g., voice and video) may need guaranteed bandwidth to meet performance requirements (low information loss and delay); on the other hand, bursty data services may be provided more cost-effectively with a lower grade of service, using resources that can be shared statistically.

BISDN is conceived as an all-purpose digital network.

- *Connection-oriented and connectionless services*—Some applications, including most continuous-bit-rate communications, are best served by connection-oriented services. Connection-oriented services are characterized by separation of the procedures for connection establishment and the end-to-end transfer of user information. Connection establishment procedures, which must precede information transfer, determine a route and set up a path between users. Other services, including mail and other data-oriented communications, may not warrant separate connection establishment and information transfer phases, with route-related and user information carried in the same message.
- *Digital signal processing*—Some services require transparent transport of digital bit streams through the network; others may take advantage of code compression capabilities used to conserve internal network bandwidth or code conversion capabilities provided to overcome terminal incompatibilities.
- *Point-to-point and complex communications*—Some services require a single point-to-point connection, either unidirectional or bidirectional, between two end points; other "multimedia" applications benefit from parallel connections between end points, connections among multiple users, and even combinations of multiple-connection, multi-user calls. Some multi-user services may involve digital signal processing.

The goal of BISDN is to define a user interface and network that will meet these varied requirements. An objective of the network is to be able to accommodate volatile changes in service mixes, both at the level of the individual interface and the system as a whole. High-capacity and high-performance fiber-based transmission facilities are generally assumed to be required to support this environment. In addition, the network must have flexible switching capabilities and special-purpose service modules. This paper focuses primarily on the switching and multiplexing capabilities of BISDN, though the transmission aspects of the BISDN interface cannot be ignored. CCITT Study Group XVIII, the current focus for international standards on BISDN, generically calls the switching and multiplexing aspects the "transfer modes." The transfer mode chosen by the CCITT Study Group XVIII Task Group on ISDN Broadband Aspects (BBTG) as the basis of BISDN is called the Asynchronous Transfer Mode (ATM). ATM is a high-bandwidth, low-delay, packet-like switching and multiplexing technique. It is essentially a connection-oriented technique, though it is envisioned as a basis for supporting all services, connectionless as well as connection-oriented.

In retrospect, the choice of the term "Asynchronous" may have been unfortunate, for it has often been confused with asynchronous transmission. ATM is not an asynchronous transmission technique. Figure 1 depicts the hierarchical relationships of some of the functional layers required for information transfer across BISDN.[1]

The transmission layer, also referred to as the physical-medium-dependent layer, underlies the transfer modes. A wide variety of service modules can be supported as higher-layer functions above the transfer mode layer. Connectionless services, multimedia, and digital signal processing capabilities can all be included in the BISDN as service modules. User-network signaling for control of connection-oriented and complex services also belongs at this level. A point-to-point connection-oriented service may not require a service module above the transfer mode layer, since ATM can be used to perform the relaying functions required to transport the user information through the network.

[1]This illustration is not a complete picture of the BISDN architecture. It excludes the "adaptation" layer, which would appear directly above the transfer mode layer, its primary function being conversion between service-specific information units and the ATM format.

EH0301-2/91/0000/0081/$01.00 © 1989 IEEE

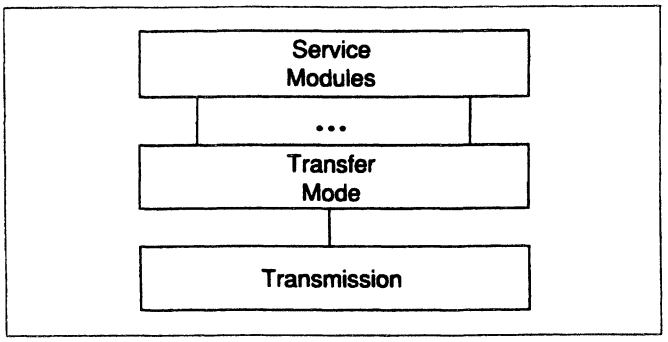


Fig. 1. BISDN functional layers.

The choice of ATM as the transfer mode for BISDN signifies a fundamental shift in principles. Two years ago, there were major disagreements on the most appropriate transfer mode for BISDN. The Synchronous Transfer Mode (STM), based on time division switching and multiplexing, was initially assumed to be the appropriate transfer mode for BISDN by many, though there was a relatively small but vocal group of advocates for "new" ATM techniques. Hybrid structures were, for a time, also considered as an option. Studies on these approaches were compartmentalized, with groups of CCITT Study Group XVIII experts working almost independently on each. There was no coherent overall plan to coordinate their efforts [7].

Fortunately, since then, the situation regarding the choice of transfer mode has improved. Over the next several meetings, the focus gradually shifted in the direction of ATM. CCITT Recommendation I.121 [8], a guideline for future BISDN standardization, designates ATM as the "target transfer mode solution for implementing a BISDN." It recognizes that ATM "will influence the standardization of digital hierarchies and multiplexing structures, switching and interfaces for broadband signals." There are only a few scattered references to other transfer modes, primarily with respect to network evolution. For example, I.121 states:

> In the early stages of the evolution of BISDN, some *interim* user-network arrangements (e.g., combinations of Synchronous Transfer Mode [STM] and ATM Techniques) may need to be adopted *in certain countries* to facilitate early penetration of digital service capabilities. (Italics are mine).

I.121 contains no sentiment to sanction international interfaces other than those based exclusively on ATM.

Although there is a virtual consensus on ATM, a number of very significant issues remain unresolved. Failure to agree on these could derail the effort to define a unique international standard. These issues are related to ATM parameters and the underlying transmission system for the User-Network Interface (UNI), the interface connecting a customer to a BISDN public network.

One major issue is the degree to which voice service considerations should be a factor in determining ATM parameters. While it is recognized that voice will have to be interworked with existing network facilities for a considerable period of time, views differ on the implications of this requirement on the specification of ATM parameters. There are also other factors that must be taken into account in setting these parameters; these may be more significant for the long-term utility of the network (e.g., suitability for high-quality video and high-speed data).

Another key issue pertains to the use of a new optical transmission system, popularly referred to as SONET [9]. I.121 contains a general statement that "ATM can be supported by any digital transmission hierarchy or system" and specifically refers to the SONET recommendations (G.707 [10], G.708 [11], and G.709 [12]) as an example. Flexibility in choice of internal transmission facilities is an advantage (e.g., existing DS3 facilities might be used to carry some ATM-based services), but a single standard is desirable at the UNI for user equipment portability. The choice between an interface with all of its physical bandwidth organized as an ATM structure and one in which an ATM structure is embedded within another framework such as a SONET frame has been left unresolved in I.121.

Complicating matters even further is the fact that these issues cannot be settled independently of each other. Both transmission and ATM parameters affect the usable capacity on an interface, limiting service capabilities that can be provided. A BISDN interface should be thought of as a system constrained by a set of higher-level service capabilities at one end, lower-level transmission capabilities at another, and ATM capabilities sandwiched in between.

This paper is divided into two parts. In the first, the basic transfer mode concepts are explained and reasons for choosing ATM over STM are given. In the second part, the main technical issues having an impact on the ATM specifications are discussed.

Paradigms and Evolution

A paradigm [13] consists of the fundamental concepts, rules, and procedures used in circumscribing a system. A change of paradigm, which should occur relatively infrequently, involves the redefinition of familiar concepts. It happens when the existing paradigm has difficulty dealing with important problems. Frequently, a definition of a new paradigm is based on assumptions about the future, which are taken largely on faith, for most current needs are supported within the existing paradigm.

The choice of ATM over STM can be viewed as the selection of a new paradigm. Both STM and ATM deal with the rules for dividing up the usable capacity (bandwidth) of an interface and allocating it to user services. The fundamental concept in the division and allocation is the channel. A channel associates a portion of the usable capacity with a service. ATM redefines this fundamental concept.

Interworking allows for an extended migration, a mandatory requirement since the entire network cannot change at once.

The elaboration of the ATM concept is in part motivated by the perceived need to accommodate the wide variety of new services mentioned earlier. On the other hand, the substantial investment in the existing plant of a complex and expensive network must be a major consideration in any scenario, since a paradigm shift cannot succeed without a smooth migration strategy to lead from the network of today to the network of the future. I.121 includes a section entitled "Evolution Steps," recognizing the need for such a strategy; however, the recommendation does not indicate any specific steps. There is surprisingly little attention devoted to the subject of evolution in the text of the recommendation. The prevailing sentiment is that network administrators will follow their own course, as dictated by national experience and needs. The only identifiable "evolutionary requirement" is that BISDN services be interworked with services in other networks, which may be a tautology since BISDN, by definition, should be able to support any service.

Interworking allows for an extended migration, a mandatory requirement since the entire network cannot change at once. But network evolution, consisting of gradual migration of geographical sections and functional elements of the network (i.e., transmission, switching, services, signaling, and control modules), should be distinguished from planning to introduce a succession of devices on a step-by-step basis to perform a particular function (e.g., an STM switch to be replaced by a hybrid switch to be replaced by an ATM switch). A series of small steps is not necessarily the best way to reach a known target,

though undoubtedly some catering will be needed to meet short-term demand. Since major network overhauls take a long time and are expensive, it is best to take as few major steps as possible.

The Limits of STM

STM, the old paradigm, was once a new paradigm. It superseded Frequency Division Multiplexing (FDM). The rule for subdivision and allocation of bandwidth using STM is: allocate time slots within a recurring structure (frame) to a service for the duration of a call. An STM channel is identified by the position of its time slots within a synchronous structure, replacing identification by frequency band of the older FDM paradigm.

FDM primarily supports analog communication systems. STM thrives in an environment dominated by digital telephony. It is the foundation of the digital transmission hierarchies and was introduced into ISDN with the $2B + D$ channel structure of the basic ISDN interface and the $nB + D$ ($n = 23$ in North America, $n = 30$ in Europe) of the primary rate interfaces.[2]

Initially, it was generally assumed by most experts in the standards community that STM would serve as the paradigm for BISDN as well. The interface structure for a BISDN could be expressed as:

$$iH_4 + jH_3 + kH_2 + lH_1 + mH_0 + nB + D$$

The coefficients $i,...n$ indicate the number of occurrences of a particular type of bearer channel on an interface. A bearer channel can carry user information end to end. There is only one signaling channel, the D-channel, which is used primarily for carrying user-network control messages, though it may also be used for user information but not at the full channel rate. Proposed and standardized rates for the bearer channels appear in Table I. Values either standardized or firmly agreed on appear in bold numerals.[3]

The STM interface structure expressed in Table I is an extension of the narrowband ISDN structure, obtained by adding higher-bit-rate channels to the previously standardized $nB + D$ structures. It was thought that STM structures would facilitate early deployment of BISDN systems, since they could be based on familiar concepts associated with existing technology. Compatibility with existing systems was a primary objective and it was envisioned that newer services would be carried via overlay structures and networks. The decision to specify channel rates for BISDN, though channel rates cease to have the same significance in the ATM paradigm, was in part based on a feeling that we should not burn all of our gateways behind us, facilitating possible interworking with STM-based fabrics.

The suitability of the STM paradigm for BISDN was first questioned because of a concern about its flexibility in meeting the needs of the future. Existing STM-based interfaces tend to be rigidly structured. In the digital hierarchies, primarily used for interoffice transport, interfaces aggregate a fixed number of identical channels of the next lower level of the hierarchy. The basic $2B + D$ ISDN interface has little flexibility in its channel structure, largely because of its limited capacity. The primary rate interface can potentially carry a mix with other channel types (H_0 and possibly H_1 channels). But demand circuit-switched services at these higher rates are not essential services of an ISDN. Many digital switches, originally designed before ISDN was conceived, support only 64-kb/s circuits, making it highly unlikely that higher-rate circuit switching fabrics will become ubiquitous.

Flexible assignment of sets of time slots to channels for switched services, with each channel consisting of one or more slots per frame, is possible. But it requires coordination of relatively complex mapping functions by both the user and network sides of an interface. There are approximately 2,000 usable 8-bit time slots per frame in a 150-Mb/s interface, and even more slots in higher-rate interfaces. Given a large number of slots and even the modest variety of channel types that were being discussed, it would require relatively sophisticated functionality to manage the mapping of all possible combinations of slot assignments for a dynamically changing mix of services (all the sets of values for $i,...n$).

In order to somewhat simplify the mapping function, STM advocates divided the usable capacity into a limited number of fixed STM-based partitions, called containers. Each container would be permanently assigned a set of time slots. For example, a proposed interface at about 150 Mb/s may be based on an interface structure of:

$$H_4 + 4H_1 + nB + D$$

In order to increase flexibility, the containers for the H_4 and H_1 channels may be further subdivided (e.g., the H_4 container could be divided into H_2 containers, which in turn could be subdivided to carry more H_1 channels). Channels could not span containers. Strict specification of the rules for subdivision and assignment of slots to channels might provide some flexibility without all of the complexity associated with unconstrained assignment of slots to channels.

Table I. Bearer Channel Rates (kb/s)

Channel	2.048 kb/s Hierarchy	1.544 kb/s Hierarchy
B	64	
H_0	384	
H_1	1,920	1,536
H_2	32,768	about 43,000-45,000
H_4	132,032-138,240	

However, the subdivision of containers can have some undesirable consequences. For example, given the above partitioned structure, a user might request a call needing a fifth H_1 channel, which can only be completed if the capacity for the available H_4 container is carved up. After allocating the fifth H_1 channel from within the H_4 container, some of the original H_1 calls might be disconnected. Even though the total idle usable capacity on the interface is sufficient, an H_4-rate call cannot be completed because the capacity for the H_4 channel does not exist within the H_4 container. If lower rate connections using the H_4 partition have long holding times, they can effectively lock out H_4 service. To remedy this situation, it is conceivable for these slots to be reassigned dynamically to an H_1 container during a call, though this has not normally been part of the functionality of an STM-based system. Historically, STM systems never needed to reassign slots because all channels at a given level of the digital hierarchy were the same size.

Multiple-rate STM also complicates the switching system. In terms of utilization of switching bandwidth on a per-connection basis, it would be more efficient to deploy a separate switching fabric for each channel rate. However, deployment of multiple fabrics is undesirable from the perspective of the network as a whole, complicating management, provisioning, and maintenance. A multi-rate switching fabric, though more complex than a single-rate fabric, is conceivable, but it is difficult to build an efficient broadband switching capability out of 64-kb/s modularity. Thus, even with multi-rate switching capability, deployment of several fabrics may be the ultimate fate of an STM-based network.

[2] A B-channel is a 64-kb/s bearer channel, with the capacity required to carry Pulse Code Modulated (PCM) voice. The D-channel, which is 16 kb/s in the basic and 64 kb/s in the primary rate interface, carries signaling and possibly other packetized services.

[3] H_3 has been omitted because there is currently little interest in defining a channel in the 60–70-Mb/s range.

So far, we have focused on problems that may arise if STM-based solutions are used to carry a dynamically changing mix of services at a variety of fixed channel rates. STM is at its best when it comes to fixed-rate services. BISDN, however, needs to support an entirely different class of services—bursty services—as well. STM techniques cannot transport these services efficiently. STM channels must accommodate peak transfer rates. Higher-layer multiplexing and switching (yet another type of switching fabric) would be required to efficiently use an STM channel.

The Flexibility of ATM

ATM techniques attempt to eliminate these limitations. Usable capacity can be dynamically assigned on demand. A single fabric could conceivably be used to switch all services. And ATM-based networks may be engineered to take advantage of statistical gain among bursty services while guaranteeing acceptable performance for continuous-bit-rate services.

The ATM paradigm did not, strictly speaking, originate with BISDN. Some of its fundamental concepts can be traced back to packet switching systems. But ATM differs significantly from older store-and-forward networks. It supports a wider variety of services by adopting procedures that ensure higher grades of performance.

STM and ATM have different rules for allocating usable capacity to channels. In ATM, specific periodic time slots are not assigned to a channel. Usable capacity (bandwidth) is segmented into fixed-size[4] information-bearing units called cells. Each cell consists of a header and an information field (see Figure 2). These cells can be allocated to services on demand.

The header contains a Virtual Channel Identifier (VCI) field. The VCI is a label that is used like the STM time slot position for channel identification. Like a time slot number, a VCI value is locally significant to an interface and a cell may undergo VCI translation, analogous to time slot interchange, before it is transported over another interface.

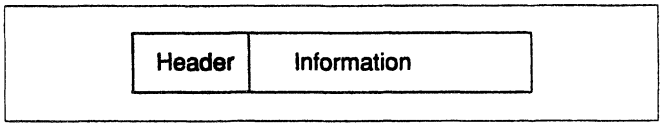


Fig. 2. ATM cell.

An ATM interface structure consists of a set of labeled, not positioned, channels. Labeled channels do not have to be restricted to a small set of fixed-rate values. The service mix and information transfer rates are decoupled from characteristics of the switching fabric. Mapping of time slots to channels at call establishment is not performed. Partitioning of the interface into containers is unnecessary, low- and high-bit-rate services sharing the same homogeneous "container."[5] Thus, ATM can be responsive to short-term fluctuations and long-term changes in service demand.

[4]Variable-size units at the ATM level were considered, but relatively small, fixed-size units were chosen. Small fixed-size cells may reduce queuing delay for a high-priority cell, since it waits less if it arrives slightly behind a lower-priority cell that has gained access to a resource. It was also thought, based on the state of existing experimental fast packet switching technology, that fixed-size cells could be switched more efficiently.

[5]While not necessary, an interface could still be partitioned using synchronous multiplexing for transmission purposes. It may be desirable to multiplex several ATM streams using STM techniques to achieve transmission interfaces exceeding the rate of operation of ATM switches and multiplexers. SONET is designed to provide this STM multiplexing capability.

In comparison to STM equipment, ATM multiplexers and switches are less dependent on considerations of bit rates for particular services. ATM equipment can flexibly support a wide variety of services with different information transfer rates without the complications associated with terminating a multiplicity of positioned channels. Consequently, multiple overlay rate-dependent circuit fabrics are unnecessary, enhancing network integration.

ATM may be suitable for both bursty and continuous-bit-rate services. ATM, operating in a deterministic mode, can support real-time continuous-bit-rate services by circuit-mode emulation, provided that sufficient switching and transport bandwidth is guaranteed. Bandwidth is allocated as part of a connection establishment signaling procedure that precedes end-to-end information exchange. The information transfer mode of a deterministic service must meet minimal end-to-end performance requirements for delay and loss so that a receiver can reconstitute a periodic signal in which the time interval separating instances of the signal are equal.

ATM may also be able to operate in a statistical mode to concentrate bursty data. Bursty traffic may be assigned a lower priority so that it does not adversely affect deterministic services. While ATM may not be as "efficient" as STM in terms of bandwidth utilization and total delay for a specific continuous-bit-rate service (because of additional header overhead and the time to collect a full cell of information), these factors may not significantly impair resource utilization or performance, and may be offset by ATM's overall advantages. ATM may be more efficient in carrying bursty services. Each service may not require continuous allocation of its peak bandwidth, allowing more services to statistically share the resource. There are still significant technical issues [14] and concerns over the tradeoff between statistical gain and quality of service. Some feel that the basic ATM capabilities should be deterministic, requiring peak bandwidth allocation for bursty services, with a possibility for statistical multiplexing above ATM [15]. ATM, in their view, is justified solely in terms of its support of deterministic services at any bit rate. Others are trying to deal with the statistical gain problem in order to optimize resource utilization and avoid introducing an additional level of multiplexing and switching.

In comparison to STM equipment, ATM multiplexers and switches are less dependent on considerations of bit rates for particular services.

Hybrid Solutions

Even as the merits of the new approach became increasingly apparent, it was still common to view STM as a necessary transitional link. It was assumed that a hybrid structure might facilitate gradual change from an STM-based to an ATM-based network, though the technical basis for this belief was rarely closely scrutinized.

The term hybrid has been used in several different contexts in standards discussions. There is the "hybrid network," which means that the network might contain internally separate STM-based and ATM-based fabrics. Individual interfaces could still be either ATM or STM. The UNI might be based on ATM, but some services may be segregated within the network and transported over STM overlay fabrics. Internal segregation of different types of traffic may be advantageous.

The term hybrid has mostly been used in discussions of interfaces, not networks, but even in this context, its usage has

not always been clear. The distinction between hybrids on the horizontal and vertical plane has been frequently glossed over. Employing STM and ATM techniques at different levels of an interface is not the same thing as having both techniques coexist at the same level. An example of a vertical hybrid would be an interface that carries ATM cells within the payload capacity of a SONET transmission frame. It is hybrid in the sense that SONET uses STM techniques to allocate time slots between transmission overhead and usable payload. At the transfer mode level, ATM techniques are used for multiplexing user channels. A vertical hybrid permits use of the STM multiplexing capabilities of SONET to build higher-rate interfaces consisting of several ATM streams.

An example of a horizontal hybrid is an interface supporting user services directly on both STM and ATM channels. Some channels are identified by time slot, others by labels. In the past, a variety of arguments had been made in favor of this approach, some based on alleged limitations of ATM. For example, it was argued that it might not be possible to switch video services at the H_4 rate. Even if this were true, and recent evidence indicates that it is not, video could be circuit switched in cell format and the UNI payload could still be based entirely on an ATM multiplexing structure. It was also argued that time slots for B-channels were needed for voice in order to avoid the delay incurred in collecting a full cell of voice samples; however, other means for dealing with voice delay, which are to be discussed in the next section of this paper, are now being favored.

The reluctance, internationally, to embrace horizontal hybrid interfaces indicates a commitment to looking forward to the full ATM access and integrated ATM fabric, avoiding the institution of aspects of the STM paradigm into the user payload structure of BISDN. It eliminates the prospect of two modes of voice service, which conceivably might coexist on a single interface, one for STM voice and another for ATM voice. Perpetuation of a duality complicates termination of information-bearing and signaling channels.

Some Unresolved Issues

The selection of ATM has resulted in a common paradigm for the transfer mode to be used at the UNI. But it has not resolved all issues concerning the elaboration of the ATM concept. ATM specifications are influenced by considerations stemming from other functional layers of the BISDN architecture. At this time, the most salient differences confronting standardization of the UNI for BISDN are concerned with ATM parameters and the transmission structure underlying the ATM layer.

I.121 indicates that two BISDN interface rates will be standardized, one at about 150 Mb/s, the other at about 600 Mb/s. This section focuses on the 150-Mb/s interface because the constraints on interface parameters are more severe at this rate. Use of an ATM-like cell structure is not necessarily restricted to BISDN interfaces. IEEE 802.6 is defining a protocol for the Media Access Control (MAC) layer for a Metropolitan Area Network (MAN) called Distributed Queue Dual Bus (DQDB) (originally QPSX) [16], with a cell structure that is intended to conform with that of ATM. While DQDB will initially be used over a DS3 rate (44.736 Mb/s) physical interface, conforming to the ATM structure is intended to facilitate migration to BISDN.

The discussion of the technical issues is organized for presentation purposes into two "positions." One position is being formulated within T1S1, a body commissioned by the American National Standards Institute (ANSI) to develop positions on ISDN standards for North America. The other characterizes a position defined by members of the European Telecommunications Standards Institute (ETSI),[6] a regional organization that coordinates telecommunications policies in Europe. These two positions were chosen to provide a vehicle for understanding technical issues. The standards community is not divided into well-defined, monolithic camps. There are other significant actors, including participants from Japan and Australia, that are also involved.

Table II presents system parameters pertaining to the 150-Mb/s interface for each of these two positions. The row headings in the table and the next three subsections of this paper each correspond to a layer in the architecture depicted in Figure 1. The service mix and usable transmission parameters apply only to the 150-Mb/s interface, but the ATM parameters carry over to other BISDN interfaces, i.e., the 600-Mb/s interface.

Parameter definitions for Table II are:

- C_{info}: Total information payload provided to the user by ATM in Mb/s
- H_4: Capacity in Mb/s for an H_4-rate service
- D: Capacity in Mb/s for user-network signaling
- x: Capacity in Mb/s for some narrowband services
- S_h: Length of the header portion of a cell in octets
- S_i: Length of the information portion of a cell in octets
- I: Information payload capacity in 125 μs in octets
- n: A positive integer

Table II. Interface Parameters

	T1S1	ETSI
Service Mix	$C_{info}=H_4+D+x$	
ATM Parameters	$S_h=5$	$S_h=4$
	$S_i=64$	$S_i=32$
Usable Transmission Capacity	$I=2,346$	$I=n*[S_h+S_i]$
	$C_{info}=10^{-6}*8,000*[8*S_i]\left[\frac{I}{S_h+S_i}\right]$	

Service Demand

The service mix requirements for the 150-Mb/s interface are expressed in terms of a set of fixed-rate channels (see Table II). As previously noted, the specification of BISDN channel rates is in a sense a throwback to the STM paradigm. But channel rates still have value as a guideline for service rates. These rates may serve as a basis for internally deployed overlay networks. The H_4-rate may also serve as a peak value for services, ensuring that some capacity remains available for multiplexing other services and control messages.

The service mix equation in the table gives the total ATM information payload capacity available to the user. The right side of the equation includes some narrowband services, x, which take up the slack left over by an H_4-rate service and signaling. Precise values have not been selected for any of the terms of the right side. The range of values being considered for H_4 appears in Table I. Some groups primarily interested in high-quality video services, such as Europe's Joint Committee on Television Transmission (CMTT), want H_4 as high as possible; however, selection of a smaller H_4 increases x and/or makes a lower value of C_{info} feasible. The capacity requirement for the D-channel is generally thought to be relatively low, and other lower-priority traffic may statistically share it.

There have been few recent discussions on the value of x. Arguments for letting $x = 4H_1 + 30B$, popular among STM

[6]Until recently, the European Conference of Postal and Telecommunication Administrations (CEPT) performed this function.

advocates, are not as compelling for an ATM-based interface. In ATM, the fifth H_1 anomaly discussed previously disappears. Thus, it may be possible to allocate all broadband and narrowband services from a slightly smaller amount of undifferentiated usable payload. The value of x is not likely to be a major stumbling block.

ATM Parameters

Table II shows differences in proposed values for ATM parameters. The discrepancies in the header size (S_h) can be attributed to different concepts of ATM functionality and differences in some field sizes. The difference in the information field size (S_i) stems from two views on the implications of interworking voice service between ATM and existing telephony fabrics.

The ATM Cell Header

In theory, the label is the only field that is essential in the ATM cell header in order to perform switching and multiplexing functions. However, other considerations, some related to enhanced performance, warrant the inclusion of other fields.

The five-octet cell header format shown on the left side of Figure 3 (each line is one octet) is the product of a coordinated effort by T1S1 and IEEE 802.6. The header contains an 8-bit Access Control Field (ACF) for multi-access interfaces, a 20-bit label, all of which is VCI, a 2-bit payload type field (TYPE) for maintenance, a 2-bit priority field (PRIORITY) and an 8-bit Header Check Sequence (HCS) field. The right side of Figure 3 shows a four-octet header format conforming to the ETSI position. It calls for a 24-bit label field containing Virtual Path Identifier (VPI) and VCI subfields, a 5-bit HCS, a 1-bit TYPE, and 2 bits of "spare capacity" (SPARE).

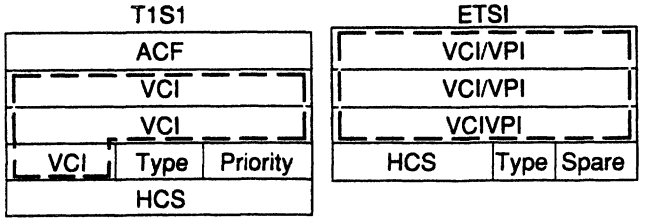


Fig. 3. ATM header formats.

One significant difference is the appearance of the ACF field in only the T1S1 format. The purpose of the ACF is to provide an efficient and reliable mechanism for multiple BISDN terminals to be connected to a shared medium. This option makes it possible to define BISDN interface configurations similar to those employed for Local Area Networks (LANs) and MANs. While T1S1 favors having the shared medium option at the UNI, many CCITT delegations want to restrict BISDN interfaces to point-to-point interfaces only. This would require multiplexing and switching of BISDN terminals on a single UNI to be performed as a centralized function. According to this view, a shared medium interface is not subject to standardization. ACF functionality for a multi-access interface is dependent on the terminal configuration, they contend, and it is undesirable to define an ACF in the ATM layer that is geared to a particular topology.

Another significant difference is the inclusion of the VPI in the ETSI label. The virtual path concept, originally conceived for cell-based cross-connect systems internal to the network, is now being considered for the UNI as well. The VPI defines an additional layer of multiplexing on a per-cell basis underneath the VCI, making it possible to group VCIs into a virtual path that can be switched as a unit. The label field, which combines VCI and VPI, is 24 bits. The size of the subfields may vary at different interfaces but will be fixed at a particular interface, such as the UNI. Sizes for the VCI and VPI subfields, the role for a VPI at the UNI, and the relationship between virtual paths and BISDN signaling and control functions need more study.

In theory, the label is the only field that is essential in the ATM cell header in order to perform switching and multiplexing functions.

The selection by T1S1 of a 20-bit label over a 16-bit label, the first value under consideration, was based less on a firm belief that all the extra bits are needed at the UNI, but on a feeling that they provide a margin of safety. 16 bits may be sufficient for communications passing between a single subscriber and the network at the UNI. However, the additional space may be needed for other purposes. For example, some of the label space may be used exclusively for local communications, or the network may require a larger number space between remote electronics and the central office. All 20 bits are designated as VCI, but it is conceivable that some bits may be assigned for VPI functions if it is agreed that the virtual path concept is the appropriate way to cross-connect virtual channels.

The role of priority in the ATM layer is another point that remains unsettled and has caused some confusion. The confusion is partly semantic, for priority can pertain to both delay and loss. Delay priority is related to the order in which queued cells are served. Higher-priority cells would, on the average, incur less delay between two given end points, since they are served first. Changing the delay priority within a given VCI can result in resequencing of cells, which contradicts the common notion of circuit.

A second type of priority, referred to as loss priority, may not be related to the order in which cells are delivered. Loss priority determines the order in which the network discards cells in overload situations when buffer overflow is impending. Some application may be designed to take advantage of such a capability, changing the loss priority for cells belonging to a single VCI on a per-cell basis. For example, a video application may use subband coding and may be willing to accept a higher cell loss rate for some information. Some loss algorithms may be more complicated than those for delay priority. The implications of instituting loss priority on services not requiring this feature must also be studied.

The ETSI header does not include an explicit field for priority. Priority can be selected when a connection is established, with its value stored where needed (e.g., in routing tables). Loss priority cannot be changed on a per-cell basis. When the value is not carried explicitly, tables containing priority for each circuit might need to be maintained at several points throughout the network. An explicit field can reduce the number of locations; however, the network may not be able to trust a user-supplied value, especially in the delay priority case, and may need to verify it at some point against the value specified at call establishment, if applicable.

An HCS field that provides one-bit error correction capability on the label significantly improves the quality of services for virtual connections emulating circuit-mode connections. Correction of random bit errors in the VCI can

prevent the loss of an entire cell. Loss of a cell, in turn, could result in loss of frame in an emulated circuit service. The HCS can also be used to detect burst errors. A 5-bit HCS can provide single-bit correction over a 24-bit field; a 6-bit HCS would be required to protect 32 bits. A longer HCS buys additional error detection capability [17]. ETSI proposed a 5-bit HCS, France recommended raising it to 6 bits, and most other delegations advocate an 8-bit field.

The Payload type field can be used for maintenance purposes. Choice of size is not likely to become a significant issue.

The Cell Information Field

The effect of the delay incurred in collecting a full cell of voice samples has been a major consideration in discussions on cell size. The current performance standards of the existing network do not tolerate much additional delay in the local access portion of a voice connection without possible degradation in the quality of service in the guise of signal loss or echo. If additional delay is incurred, measures such as deploying echo cancellation devices must be taken.

ETSI favors a 32-octet information field, as shown in S_i of Table II. This cell size would incur an additional 4 ms to collect 32 1-octet samples using 125-μs-based voice protocols. Their view is that this additional delay can be compensated for by reducing the maximum distance of connections that do not require echo cancelers. Larger information-field sizes would require echo cancelers for shorter distances.

In the telephone network in the United States and elsewhere, echo cancelers would be required in interworking units even if more than 16 octets of information are collected. The United States has stricter standards than many European countries, limiting signal loss in calls over short distances, so there would be a problem even in local calls. Consequently, an alternative method for dealing with delay, independent of the information field size, is needed. Either cells are not completely filled with voice samples of a single connection or echo cancelers must be used. Partial filling of cells may inflate the required value of x in the service mix equation of Table II. Various techniques, sometimes called composite techniques, may allow sharing of a cell among several connections. But these add an additional level of multiplexing and run counter to the objective of basing all network services on a single, homogeneous transfer mode.

One advantage of a longer cell is that it gives ATM processing units more time to process a header, though it is technically feasible to build equipment based on either size. Another is that, given the header sizes under consideration, studies [17] have shown some improvement in capacity utilization for 64 octets over the smaller-size information fields. Studies confined to the ATM layer may in fact be underestimating the cost of small cell sizes since there may be additional per-cell overhead for segmentation/reassembly and other functions in an "adaptation layer" above ATM.

Transmission and Its Impact on Aggregate Usable Capacity

The usable transmission capacity rows of Table II define the aggregate usable capacity available in a 150-Mb/s interface. The different forms of the expressions on the first line of these rows reflects an underlying disagreement on the fundamental transmission structures. Figure 4 depicts the structure of the two approaches being considered, the SONET approach favored by T1S1 and the Asynchronous Time Division (ATD) approach favored by some in ETSI.

Each rectangle in Figure 4 represents a 125-μs snapshot of a transmission signal. Transmission occurs from left to right, row by row. Cells are bounded by solid lines, and in the case of the structure in Figure 4a, a single cell can be divided by SONET transmission Overhead (OH). If SONET is used, a framing pattern would not be required in ATM cells since cell boundaries can be derived from the 125-μs framing (ff) in the SONET overhead. All cells can be used to carry user information. In contrast, ATD, a physical interface consisting solely of cells, requires a framing pattern within the ATM stream. Since framing is essential for delimiting cell boundaries, 100% occupancy of cells is not possible. Any additional transmission and operations OH required must also appear in cells, making it difficult to determine a precise value for the usable capacity. The C_{info} value computed for an ATD interface is slightly inflated, because it does not account for OH occupying information portions of cells. One method of framing in ATD fills empty cells with a special pattern (depicted as ffffff...). When the interface is overloaded, that is, there is a 100% offered load, a framing cell would have to be periodically "forced."

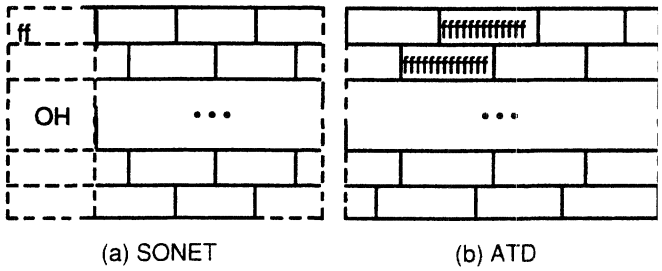


Fig. 4. Framing structures.

T1S1 favors SONET over ATD because:

- It can be used to carry either ATM-based or STM-based payloads, making it possible to initially deploy a high-capacity fiber-based transmission infrastructure for aggregated narrowband and data services. This early deployment, which can be justified purely in pre-BISDN terms, preadapts the network for BISDN and ATM. Assuming widespread penetration of SONET for these purposes, it is sensible to continue to use SONET for BISDN rather than a new structure to perform the same functions.
- SONET offers cross-connect capabilities that can be used to transport aggregations of ATM connections within the network. This is an alternative to the VPI-based cross-connect function mentioned previously. A SONET-based cross-connect, in contrast to a virtual path cross-connect, may not be quite as flexible in meeting instantaneous changes or in statistically allocating bandwidth. These features, however, may not be desirable in a cross-connect system, since the objective in its design is to avoid complexity and potential complications if congestion is allowed to occur.
- The cells of a specific connection may be "segregated" and circuit switched using a "SONET channel." For example, an H_4-rate video connection may be demultiplexed and then mapped into its own exclusive payload envelope of the SONET STS-3c (155 Mb/s) signal, which can be circuit switched in the network. It is possible to define payload envelopes at a variety of rates and the channels can be subdivided using SONET's virtual tributary capabilities [9]. The payload can remain in cell format, making STM/ATM format conversion unnecessary, or it may be converted to a more STM-oriented format. STM-like switches of this type can be used internally as an interim measure, before ATM switches are available. It may even prove to be more cost effective than ATM switching for handling some services, such as commercial video distribution.
- SONET synchronous multiplexing capabilities can be used to combine several ATM streams to build interfaces at rates greater than those supported by the ATM layer. This technique may be employed inside the network and at the UNI. For example, a 622-Mb/s (STS-12) interface can be built from four 155-Mb/s (STS-3c) modules, each one carrying a

separate ATM multiplex. This arrangement may be more cost effective than one using a single 600-Mb/s ATM stream.

ATD proponents argue that basing both transmission and transfer mode functions on a common structure simplifies the interface. This contention must be weighed against the versatility of using SONET in the network and the benefits of a UNI that has commonality with SONET network node interfaces.

The differences in the first line of the usable transmission capacity rows of Table II can be explained in terms of the structures in Figure 4. SONET provides a fixed-rate payload envelope, I, which is computed by subtracting SONET overhead from the SONET interface rate. In ATD, there is greater freedom in determining I. In the past, it has been suggested that an integer number of cells occur in a 125-μs period, making I a multiple of the cell size and an integer value n. This condition is not essential for ATM, since there is no requirement that synchronization cells be transmitted periodically, let alone fall on a 125-μs boundary; however, periodic framing makes it possible to align the framing patterns for synchronous multiplexing without introducing an external framing pattern and 125-μs periodicity makes it simpler to derive 125-μs clock from the signal, which may simplify provision of some services.

The second line of the usable transmission capability row computes C_{info}, the total usable capacity provided by ATM to its users. Adopting a SONET-based pointer scheme proposed by AT&T for delimiting ATM cells [18], T1S1 uses all 2,346 octets of payload in a SONET STS-3c frame for carrying ATM cells, yielding a C_{info} of 139.264 Mb/s. The ETSI ATD approach, in contrast, makes it possible to derive I from the other parameters. Thus, issues dealing with service mix and ATM parameters may be settled first. The ATM parameters favored by ETSI, $S_h = 4$ and $S_i = 32$, incur a relatively high percentage of ATM header overhead, yielding a C_{info} of only 133.120 Mb/s in a SONET STS-3c.

The ETSI system is molded by their perspective on interworking voice communications with shorter cells. By and large, they have ignored the implications of choosing SONET for a UNI on their selection of ATM parameters. T1S1, on the other hand, takes SONET at the UNI as a given and looks beyond the voice delay problem, seeing no significant difference in the implications of 32- versus 64-cell information fields for voice and emphasizing other applications for its selection of ATM parameters.

Conclusions

Some substantial issues stand in the way of achieving the CCITT objective of a unique worldwide interface for BISDN. However, standards organizations in the past have pulled things off when least expected. The agreement on the SONET network node interface is a good example. CCITT Study Group XVIII Working Party VII overcame many obstacles to make it acceptable to the international community [9].

The CCITT Study Group XVIII Broadband Task Group has laid the foundation for an agreement on a unique international standard for the BISDN UNI by reaching consensus on the selection of the ATM paradigm, but significant differences remain. The standards organizations are working against the clock. Business pressure to provide high-bandwidth communications over wider areas is growing. Inability to agree in a timely fashion increases the likelihood of regional solutions. CCITT called an interregnum meeting for January 1989 to try to expedite international agreement, hoping to achieve a standard in 1990. Compromise solutions were explored at the January meeting, but not much progress was made. This too is reminiscent of the SONET experience.

The focus of discussions in the Broadband Task Group have tended to emphasize transmission, switching, and multiplexing. Discussions of other aspects of BISDN are in the embryonic stage. In the future, the emphasis is likely to shift more in the direction of higher-layer capabilities, including adaptation of higher-layer functions to ATM, new service-specific functions, and signaling. There has been little discussion of multimedia calls and services, which have a tremendous potential for thriving in a BISDN environment. Multimedia services may require rethinking some of our concepts [19].

Reaching agreement on the fundamental transmission, multiplexing, and switching aspects of BISDN is an important step in building the intelligent broadband network of the future.

Reaching agreement on the fundamental transmission, multiplexing, and switching aspects of BISDN is an important step in building the intelligent broadband network of the future. Defining the transport capabilities for BISDN is a complex problem, requiring a solution taking transmission, switching, and services factors into account. Involvement of many different organizations in this process has complicated the situation further. The task of achieving a solution that is both forward-looking and backward-compatible is not trivial. The payoff for coming up with a workable solution could be immense.

Addendum

The results of the latest CCITT Study Group XVIII meeting, held in June 1989, include agreements on some ATM and transmission parameters. ATM cells will consist of a 5-octet header (S_h) and a 48-octet information field (S_i). For the UNI, the header includes:

- A 4-bit "generic flow control" field, which may be used to perform some of the functionality of an ACF field, though it can also be used for point-to-point
- A 24-bit label, with up to 12 bits for VPI and the remaining 12 to 16 bits for VCI
- A 2-bit payload type field
- A 2-bit reserved field
- An 8-bit header error check field

There is no explicit priority indication.

CCITT will standardize two physical interfaces to BISDN, one based on SONET, the other a variation of ATD. The UNI interface rate for both of these was set at 155.520 Mb/s, with an interface transfer capacity equivalent to 149.760 for carrying cells containing user information. Thus, the value of I is 2,346 octets on both interfaces, and C_{atm} is about 135.979 Mb/s. The H_4 rate may be further constrained by additional per-cell overhead at the adaptation layer. If a 1-octet-per-cell adaption layer header for continuous-bit-rate services is added, C_{atm} is effectively reduced to about 133.146 Mb/s.

References

[1] J. L. Gimlett, M. Stern, R. S. Vodhanel, and N. K. Cheung, "Transmission Experiments at 560 Mb/s and 140 Mb/s Using Single-Mode Fiber and 1300-nm LEDs," *Electron. Lett.*, vol. 21, pp. 1,198–1,200, Dec. 1985.

[2] G. Hayward, L. Linnell, D. Mahoney, and L. Smoot, "A Broadband ISDN Local Access System Using Emerging-Technology Components," *Proc. of Int'l. Switching Symp. '87*, vol. 3, pp. 597—601, 1987.
[3] J. S. Turner, "New Directions in Communications," *IEEE Commun. Mag.*, vol. 24, no. 10, pp. 8–15, Oct. 1986.
[4] J. P. Coudreuse and M. Servel, "Prelude: An Asynchronous Time-Division Switched Network," *Proc. of the IEEE Int'l. Conf. on Commun. '87*, vol. 2, pp. 769–773, 1987.
[5] C. Day, J. Giacopelli, and J. Hickey, "Applications of Self-Routing Switches to LATA Fiber Optic Networks," *Proc. of Int'l. Switching Symp. '87*, vol. 3, pp. 519–523, 1987.
[6] D. Spears, "Broadband ISDN—Service Visions and Technological Realities," *Int'l. J. for Analog and Digital Cabled Syst.*, vol. 1, no. 1, pp. 3–18, Jan. 1988.
[7] S. Minzer, "Broadband User-Network Interfaces to ISDN," *Proc. of the IEEE Int'l. Conf. on Commun. '87*, vol. 1, pp. 364–369, June 1987.
[8] CCITT Recommendation I.121, "Broadband Aspects of ISDN," Blue Book, Geneva, Switzerland, 1989.
[9] R. Ballart, Y. C. Ching, "SONET: Now It's the Standard Optical Network," *IEEE Commun. Mag.*, vol. 29, no. 3, pp. 8–15, Mar. 1989.
[10] CCITT Recommendation G.707, "Synchronous Digital Hierarchy Bit Rates," Blue Book, Geneva, Switzerland, 1989.
[11] CCITT Recommendation G.708, "Network Node Interface for the Synchronous Digital Hierarchy," Blue Book, Geneva, Switzerland, 1989.
[12] CCITT Recommendation G.709, "Synchronous Multiplexing Structure," Blue Book, Geneva, Switzerland, 1989.
[13] T. Kuhn, *The Structure of Scientific Revolutions*, 2nd ed., Chicago, IL: University of Chicago Press, 1970.
[14] J. Gechter and P. O'Reilly, "Conceptual Issues for ATM," *IEEE Network*, vol. 3, no. 1, pp. 14–16, Jan. 1989.
[15] B. Eklundh, I. Gard, and G. Leijonhufvud, "A Layered Architecture for ATM Networks," *Proc. of IEEE Globecom '88*, pp. 409–413, Nov./Dec. 1988.
[16] R. M. Newman, Z. L. Budrikis, and J. L. Hullett, "The QPSX MAN," *IEEE Commun. Mag.*, vol. 26, no. 4, pp. 20–28, Apr. 1988.
[17] D. P. Hsing, F. Vakil, and G. H. Estes, "On Cell Size and Header Error Control of Asynchronous Transfer Mode (ATM)," *Proc. of IEEE Globecom '88*, pp. 394–399, Nov./Dec. 1988.
[18] T1S1.1/88-390, "Mapping BISDN ATM Cells into SONET STS-3c Payload," AT&T Technologies, Oct. 1988.
[19] S. E. Minzer and D. R. Spears, "New Directions in Signaling for Broadband ISDN," *IEEE Commun. Mag.*, vol. 27, no. 2, pp. 6–14 and 52, Feb. 1989.

Biography

Steven E. Minzer is a Member of Technical Staff in the Multimedia Communications Research Division of Bellcore, Morristown, New Jersey. He has been involved in Broadband ISDN standards activities since 1985, participating in Committee T1 and CCITT Study Group XVIII meetings. He has also been involved in protocol design for an experimental broadband network at Bellcore. Before joining Bellcore, he was with AT&T Bell Laboratories, designing and implementing software for the DMERT operating system. He also worked in database administration and software development for GTE Automatic Electric. He obtained degrees in political science from Cornell University (B.A.) and Ohio University (M.A.) and in computer science from Northern Illinois University (M.S.). His publications include several papers and a Bellcore Special Report on BISDN and articles in *ISS '87* and *ICC '87* on protocols for the experimental broadband network.

The Evolution of Broadband Network Architectures

D. V. Batorsky, D. R. Spears and A. R. Tedesco

Bellcore
Red Bank, NJ 07701

ABSTRACT

Market pull and technology push are the major forces driving future broadband networks. Firstly, this paper examines these two driving forces. Secondly, it describes a long-term broadband network architecture. Finally, it describes possible evolutionary paths from early-availability architectures through the longer-term.

1. BROADBAND SERVICES MARKET PULL

New service opportunities are a major driving force for future broadband networks [1] [2] [3]. Business customers have a growing need for information technology and increasingly sophisticated telecommunications. Local Area Networks (LAN) have increased the intrapremises data rates from kbit/s to Mbit/s over the past decade, but interconnection of these LANs is still a problem.

Residential customers are demanding more selectivity in video entertainment, evidenced by the increasing number of channels demanded on cable television systems and the increasing rentals of video cassettes. Future entertainment services may range from viewing normal broadcast video sources, to viewing special events such as first-run movies on a pay-per-view basis, to video-on-demand in which a user is connected directly to a remotely-located video source with complete control over that source.

This section discusses some of the existing and future services that a broadband network should support.

1.1 Existing Services

A broadband network should support existing voice telecommunications services. Other voice services, such as 7 kHz and 15 kHz speech communications, may also become important, especially for business customers.

A broadband network should support existing data communications services, such as those that provide features for X.25 and SNA applications. In addition, higher-speed data services, discussed in the next section, will become increasingly important.

A broadband network should support existing private line services, including the DS-0, DS-1 and DS-3 rates. New innovations in the signaling protocol for broadband networks may enhance these capabilities by allowing private virtual networks to be defined dynamically.

1.2 High-Speed Data Services

The widespread penetration of LANs has firmly established very high-speed networking (1-10 Mbit/s) in the local premises environment. The success of LANs can be attributed to the growing use of more powerful PCs and workstations for high performance applications such as remote file serving and distributed processing. The growing need for LAN interconnection and such applications as desktop publishing and CAD/CAM suggest an emerging market for multi-megabit communications services that extend beyond the local premises, across metropolitan and wide area environments. Switched Multi-megabit Data Service (SMDS) is a high-speed connectionless data service proposed for broadband networks to meet these future data communications needs [4].

1.3 Video and Image Services

The availability of high-speed communications can completely change the character of business interactions. In desktop teleconferencing applications, windows on the user's terminal can display full-motion images of the other parties in a conference. Viewgraphs, sketches, photographs, and text files can be exchanged and either displayed or printed. Other parties can be added to a conference as needed for consultation, and subconferences created for private conversations.

Electronic mail, commonly used today for transmitting text messages, can be stimulated by the availability of broadband networks. Integrating multiple media, including text, voice, image and full-motion video, can increase the effectiveness of these systems.

Switched access television service, the delivery of entertainment video to residential customers over a switched broadband network, can tap a large existing market and offer several advantages over current delivery alternatives. These advantages include access to a wider variety of programming, elimination of the need for scrambling equipment, opportunities for new services such as pay-per-view, and higher quality video such as high definition television (HDTV).

Recent increases in the sale and rental of video recorders, players and cassettes show that users want to control their own programming. Video-on-demand service provides the user with ultimate control over video programming without having to purchase and maintain a video cassette player and without having to pick up and return video cassettes. Instead, a user calls a video vendor and selects a program from a list of available video sources. Video information from the vendor is switched to the user via the broadband

Reprinted from the Proceedings of Globecom, 1988, pages 367-373. Copyright © 1988 by The Institute of Electrical and Electronics Engineers, Inc. All rights reserved.

EH0301-2/91/0000/0090/$01.00 © 1988 IEEE

network, and control signals from the user (play, stop, pause, rewind, etc.) are transmitted to the vendor over a data connection.

1.4 Other Future Services

Other future broadband services include access to reference information. Text, graphics and images can be viewed on a student's terminal, printed on a local printer, or stored in a disk file on the student's PC. Some people predict that broadband networks will completely change the way data is viewed. Just as the telephone system displaced the telegraph in the nineteenth century, *telesophy*[1] may displace today's simplistic data communications [5]. Telesophy advocates predict a new mass communications system that will explicitly support the sharing of knowledge within a global information community.

Video browsing may become popular for shopping. For example, by calling a vendor via a broadband network, a shopper can view a full-motion demonstration of a riding lawn mower illustrating its maneuverability around trees and its ease of operation. With interactive capability, the seller can tailor the presentation to the level of detail desired by the buyer. A complete home merchandising system can provide cost and availability information to the buyer, allow the buyer to order the product and specify shipping information, and allow the seller to update an inventory system automatically. Real estate sales is another possible application. A person transferring to a different city can access video information remotely about homes for sale, schools, churches, shopping centers or maps of the new community.

2. BROADBAND TECHNOLOGY PUSH

Over the past few years, broadband technology has emerged as an attractive alternative for future telecommunications networks. It promises an integrated access structure with flexible allocation of bandwidth, capable of providing existing services and positioned to serve a broad range of future services. It promises a more integrated switching fabric, eliminating the need for a switching fabric for each service type and the associated complexity and high operations costs. Finally, it promises to simplify interoffice transmission and switching facilities.

Several parallel advances are responsible for the shift in economics of integrated broadband networks.

2.1 Optical Fiber

Optical fiber is used extensively in the interoffice plant. It is now starting to be used in the loop between the Central Office (CO) and subscriber carrier locations. The cost of single-mode fiber, which offers virtually unlimited bandwidth, dropped from 5 dollars/m in 1982 to about 23 cents/m in 1987. With continued progress some experts predict that the cost will drop to 4 cents/m by 1992.

1. Telesophy is a word coined by Bruce Schatz of Bellcore. It comes from the words telephony and philosophy and means "wisdom at a distance". The "tele" part indicates that the access can be made independent of physical location in the communications network. The "sophy" part indicates that filtering can be done independent of physical data type.

2.2 Low-Cost Optical Devices

Until recently, single-mode fiber implied the need for laser transmitters. However, single-mode laser sources had inadequate reliability for use in the loop and were not compatible with near-term cost objectives. On the other hand, lower cost light-emitting diodes (LED) as used in low-bit-rate multi-mode systems were known to meet loop reliability requirements. It has now been demonstrated that light emitting diodes can be used with significant lengths of single-mode fiber at bit rates in the 600 Mbit/s range [6].

More recent work has been aimed at reducing the cost of laser sources through simple packaging and relaxed yield criteria [7]. As laser technology for loop applications matures and as production quantities reach levels characteristic of the loop market, laser costs and reliability should improve. Thus, both LED and laser sources now show the potential for use as low-cost optical sources for integrated broadband networks.

2.3 Low-Cost Broadband Circuit Switches

Much progress has been made in broadband circuit switching technology. Just a few years ago, video switches used discrete relays or *pin* diodes to implement an analog switching fabric and were both large and costly.

Several experimental devices, using both CMOS and ECL technologies, have been fabricated for space-division circuit switching of broadband channels. Using such VLSI technologies and new packaging methods, it is now possible to create a compact 64x64 switching subsystem with a 4x4 matrix of 16x16 CMOS crosspoint chips on a single card measuring four inches square [8]. In addition to the advantages of digital switching and smaller size, this board represents about a 10:1 reduction in the size and the cost per crosspoint over that of the analog switching fabric.

2.4 Synchronous Optical NETwork (SONET)

The advances in synchronous multiplexing technology, the widespread deployment of incompatible fiber systems, and the need for flexible management of the bandwidth of fiber transmission systems led to the formulation of the Synchronous Optical NETwork (SONET) concept [9]. SONET is a network hierarchy of octet-interleaved synchronous signals. This hierarchy forms a family of standard electrical synchronous transport signals (STS) that can be simply converted to a family of standard optical carriers (OC). Through these STS and OC interfaces, SONET provides a set of standard transmission interfaces for customer premise equipment, switches, distribution and interoffice transport systems.

The STS rates are integer multiples of 51.730 Mb/s. The interface structure contains separate overhead and information payload. As a result operations, administration and maintenance capabilities are an integral part of the transport systems. Currently, there is a significant interest in introducing SONET based transport in exchange carrier networks when the equipment becomes available over the next several years. Moreover, SONET is proposed as a basis for defining broadband user-network interfaces for ISDN.

2.5 Label Multiplexing

Position multiplexing, also called Synchronous Transfer Mode (STM), is used for multiplexing channels into a common transmission stream, such as DS-1 or DS-3. An STM stream has a fixed channel structure; each bit within the payload is associated with a specific channel, which has a specific information transfer rate.

With label multiplexing, also called Asynchronous Transfer Mode (ATM), each frame is divided into cells that are similar to small packets [10] [11]. Each cell contains a fixed-length header and a fixed-length payload. A label in each header explicitly identifies the channel. Therefore, multiplexing and demultiplexing hardware is simplified and the structure of the transmission stream can be changed dynamically. Furthermore, due to the fixed size of the cells, STM can be emulated by ATM. ATM has been proposed for use in the SONET payload of integrated broadband networks.

2.6 ATM Cell Mergers and Sorters

Cell mergers and sorters provide flexible handling of ATM cells and have a variety of uses [12]. For example, cell mergers in the Interface Module (IM) can multiplex information from many different users onto a single STS-3c signal for transmission to the CO; a cell sorter in the CO can provide the demultiplexing function. A cell sorter in the IM can separate video information from lower speed signals and route it to a circuit switch. A cell merger in the IM can combine a circuit-switched video stream with lower-speed voice and data streams for delivery over a single fiber to the customer.

2.7 ATM Cell Switching

The technology required for broadband cell switching is much different from that used in lower speed packet switching networks. One objective of conventional store-and-forward packet switching networks is improving the error performance of noisy facilities. Copies of packets are held at the transmitting side of a link until acknowledged by the receiving side of the link. Using optical fiber transmission, the error performance of a broadband network is expected to be much better than that of current networks, so retransmission is much less likely. The primary objectives for broadband cell switching are low delay for traffic such as speech, high throughput for traffic such as still frame video, low cell loss rate, and low internal buffer requirements for minimizing equipment costs.

One promising technology for broadband cell switching is the banyan routing network. A cell processor inserts a routing field, with the most significant bit first, into each cell delivered to the input of the banyan switch. The first stage of the banyan switch examines the first bit of the routing field and routes to the top or the bottom half of the second stage based on the value of this bit. The second stage examines the second bit of the routing field and routes to the corresponding fourth of the network. This process continues through each of the N stages. A switch with N stages has 2^N inputs and 2^N outputs and is composed of $N * 2^{N-1}$ switching nodes, with two inputs and two outputs each.

If a banyan routing network is preceded by a Batcher sorting network, the combination is non-blocking internally. A 32x32 Batcher sorting network and a 32x32 banyan routing network have been implementing in VLSI; the current chips operate at over 100 Mbit/s per port with the next version expected to operate at greater than 155 Mbit/s [13].

2.8 Broadband System Prototypes

Several broadband system experiments are proceeding, including The Prelude switch in France[14], the Elastic Basket switch in Japan [15], the Knockout switch at Bell Laboratories [16], the Experimental Research Prototype at Bellcore [17] [18] [12], and the PARIS switch at IBM [19]. Each of these experiments focuses on different aspects of broadband packet switching - the interface structure, the services, the switching fabric, the protocols and the addressing - and each experiment represents a different solution to broadband packet switching.

3. A LONG-TERM BROADBAND ARCHITECTURE

Figure 1 shows a view of a long-term target architecture for a broadband network. An interworking unit (IWU) at the customer's premises provides the conversion between various customer interfaces and an integrated user-network interface. This user-network interface uses SONET transmission at the STS-3c basic rate of 155.52 Mbit/s or at a higher rate such as 622.08 Mbit/s, depending on the service needs of the customer. ATM cells within the SONET payload provide flexible multiplexing in order to support a wide range of existing and future services.

In the distribution and drop plant between the Remote Electronics (RE) and the customer, single-mode fiber is dedicated to the customer, forming a star topology[2]. Two fibers can be provided to each customer, one for upstream and one for downstream transmission, or both directions of transmission can share the same fiber using WDM techniques. The cost difference between these two alternatives is slight, with short loops favoring two fibers and longer loops favoring one fiber [20]. Operations costs, while not quantified, may favor the use of a single fiber.

Single-mode fiber is also used in the feeder plant between the RE and the CO, but operates at a higher rate near 2.4 Gbit/s. Although the RE-to-CO distances are longer than the customer-to-RE distances, the feeder costs are offset by the concentration provided by the RE and by the higher transmission rate in the feeder and represent less than 5% of the total cost [20].

Customers near the CO will be served directly from an IM in the CO. Customers further from the CO may be more economically served from an IM located remotely [20], trading off the additional cost of placing the IM in a remote location against the cost of longer lengths of dedicated fiber.

Several topologies are possible for interconnecting the RE locations to the CO. In a star topology, dedicated fiber connects each RE to the CO. In a ring topology, a fiber ring

2. A recent T1S1.1 contribution proposed another level, called a Remote Multiplexer (RM), between the RE and the customer in order to increase sharing of the distribution plant. The fiber between the RE and the RM is called the sub-feeder plant and can use either a star, bus, or ring topology.

starts at the CO and passes through each RE using add/drop multiplexers to drop and insert channels. A combination star/ring is also possible for the feeder topology in which switched channels are interconnected in a star topology and broadcast channels in a bus topology. The latter alternative may offer some savings, but the effect on the overall cost is slight since the feeder cable represents only a small portion of the total cost.

The physical implementation of a Next Generation Switch (NGS) [21] may incorporate different switch fabrics for different traffic types. For example, ATM cells carrying video information may be separated from cells carrying lower-speed information; video cells may routed to a circuit switching fabric while cells of other traffic are concentrated by ATM mergers in the IM and routed to a self-routing Batcher-banyan switching fabric. Although the physical implementation may incorporate multiple switching fabrics, the fabric appears to be integrated from the point of view of switch control and from the customer's perspective.

User-network and network-network signaling are integrated, with the same signaling protocol being used for both circuit and packet communications. Multi-media and multi-party communications are an integral part of these signaling procedures, facilitating the development of future interactive applications.

Hub offices provide the interconnections between CO locations. Like the feeder plant the interoffice plant connecting the CO locations to the Hub office may be connected in various topologies, such as star, ring or bus. If the community of interest is high enough, two COs may also have direct trunk groups in addition to connections to the Hub office.

Narrowband networks will continue to exist for many years. Interworking units (IWU) in central offices and hub offices provide the necessary conversions for interconnecting to other networks. In addition, fiber loops may be installed in exchanges with existing narrowband offices. An IWU in these offices provides conversion between the SONET interface on the fiber and an Integrated Digital Loop Carrier (IDLC) interface to the existing switch.

Integrated network operations and maintenance systems provide continuous surveillance of the network, reconfiguring the network during failure conditions to provide continuous service, isolating the cause of the failure, and dispatching personnel to repair the failed component. Integrated administration systems provide rapid service provisioning and accurate billing.

If coherent optics or dense WDM systems become practical and low-cost, Fig. 1 may change as follows. The active electronics in the RE may be moved back to the CO and multiple wavelengths used on the RE-to-CO fibers. This provides dedicated channels from the CO to the RE without increasing the number of fibers in the feeder plant. From a first cost standpoint, this alternative trades off the added cost of WDM equipment on a per-line basis against the cost of 2.4 Gbit/s multiplexers on a per-feeder basis. Also, if only passive components remain in the RE, the operations costs may be lower. In either case, the physical architecture of the installed fiber is identical independently of whether the RE contains active electronics.

4. EVOLUTIONARY PATHS TOWARD A BROADBAND NETWORK ARCHITECTURE

The evolution toward the broadband network architecture described in the previous section will be constrained by several factors. First, standards for broadband are still being discussed, although much progress has been made over the past year, particularly in standardizing SONET. Second, even after broadband standards are complete, the development of a next generation switch (NGS) is likely to require several years. During this time, some components of the NGS will become available earlier than others, resulting in network architectures using combinations of old and new technology. Third, initial broadband systems will be more expensive since they are at the beginning of the cost learning curve. Finally, even after the cost of broadband systems reach the lower part of the learning curve, capital constraints will limit the rate at which broadband systems can be installed.

In light of these factors, an evolutionary strategy for introducing broadband network elements is needed that satisfies several conditions. First, it should allow some broadband services that are already in demand, such as high-speed data services, to be offered using early availability technology. Second, it should allow fiber to be installed for growth in the loop plant instead of copper, at a cost competitive with copper, in order to be prepared for the provision of future broadband services. Finally, the evolutionary path should allow cost effect upgrades as NGS systems become available.

The next two sections outline a possible evolutionary strategy for residential and business network architectures.

4.1 Evolution of Network Architectures for Business Services

The desire to provide early availability of high speed data services coupled with current development of Metropolitan Area Networks (MAN) equipment, leads to the possible architecture implementation shown in Fig. 2. A gateway, G, provides the conversion from a LAN protocol to the Switched Multimegabit Data Service (SMDS) protocol carried in a DS-3 rate transmission stream. The DS-3 electrical signal is carried to the CO by an optical transmission system with a proprietary interface. In the CO, the DS-3 signal is cross connected to a multiplexer and carried to a hub office, again using a transmission system with a proprietary interface. At the hub office the DS-3 signals are connected to a MAN switching system (MSS), where each SMDS packet is segmented into cells and routed to the destination port.

As SONET transmission systems become available, the architecture of Fig. 2 allows for evolution of the network to include SONET transmission systems with standard interfaces. Furthermore, when ATM mergers and sorters become available, the architecture shown in Fig. 2 evolves to that shown in Fig. 3. Standard SONET transmission is used on the optical fibers with ATM cells carried in the SONET payload. Segmentation and reassembly of SMDS packets into ATM cells is done at the customer's premises. ATM mergers and sorters in the COs reduce the interoffice transmission requirements. Finally, the MSS can be based on an ATM merger/sorter pair operating in a MAN emulation mode with static translation tables.

As the demand for SMDS service increases, the throughput capacity of the ATM merger in the hub office is reached. At that point, a higher capacity ATM cell switch can be placed between the ATM sorter and merger, as shown in Fig. 4. A cell switch with N ports can route cells from any of the N mergers at the input ports to any of the N sorters at the output ports, increasing the capacity of the system by a factor of N. As the data traffic further increases, cell switches are placed in the COs to handle intra-CO traffic, reducing the load on the interoffice links. Cell switches in the COs also allow for direct trunks between COs with a high community of interest, further reducing the load on the interoffice links.

Voice services can also be provided over the same fibers in each of the three architectures shown above. In Fig. 2 the proprietary transmission can provide multiple DS-3 channels over the fiber. For example, one DS-3 channel may carry voice traffic to the existing voice switch via an Integrated Digital Loop Carrier (IDLC) interface. In Fig. 3 the voice information is carried in ATM cells on the user-network interface, split off from the data traffic in the CO using ATM sorters, and routed to the existing voice switch. When ATM cell switches are installed in a CO, the voice traffic may be carried by the cell switch, with the interworking unit moved to the trunk side of the office. This evolution is discussed further in the following section.

4.2 Evolution of Network Architectures for Residential Services

For growth of voice services in the loop, optical fiber is becoming a cost-effective alternative to copper. In addition, it positions the loop for providing future broadband services.

Several suppliers are now developing equipment for providing POTS service over fiber using proprietary interfaces. After completion of the broadband standards for SONET and ATM, standard equipment will become available for providing POTS service using the architecture shown in Fig. 5. The IWU provides the BORSCHT functions and places speech samples into ATM cells. At the RE, the cells from several users are merged together into a common stream, reducing the transmission requirements between the RE and the CO. At the CO, the aggregated SONET/ATM interface is converted to an IDLC interface compatible with the existing voice switch.

As the amount of fiber in an exchange increases, the existing voice switch may be replaced with an ATM cell switch capable of switching both voice and data services. The architecture shown in Fig. 5 is easily evolvable to that shown in Fig. 6, since the electronics in the RE are unchanged. When the cell switch is installed in the CO, an IWU on the trunk side of the office replaces the IWU on the line side.

A loop based on optical fiber has the capability to support video distribution services in addition to voice and data services. The architecture shown in Fig. 7 is one alternative for early availability video distribution. This architecture is similar to that shown in Figs. 5 and 6, except a second wavelength is multiplexed onto the fiber using WDM techniques. This second wavelength may carry several analog video signals using techniques such as frequency-modulated (FM) microwave subcarriers [22]. The incremental cost of adding video service in this architecture is the cost of the optical source and the WDM equipment in the RE, and the cost of the tuner at the customer's premises.

A second alternative for providing video distribution service is shown in Fig. 8. A channel-change message from the user results in a crosspoint being closed in the circuit switch. Thus, the selected channel is switched to the input of the ATM merger. This alternative uses ATM for multiplexing the voice, video, and data signals into a single transmission stream carried by a single wavelength. The incremental cost for providing video service with this architecture is the cost of the merger and the allocated cost of the circuit switch in the RE, and the cost of an ATM cell sorter and a codec at the customer's premises.

The architectures shown in Figs. 7 and 8 provide different service capabilities. The first architecture provides broadcast access to a limited number of video signals, but does not constrain the number of channels simultaneously accessible from that limited set. The second architecture provides switched access to an unlimited number of video sources, but limits the number of sources available simultaneously. The incremental cost of accessing an additional channel with the first architecture is the cost of an additional tuner. The incremental cost of accessing an additional channel with the second architecture is the allocated cost of an additional port on the circuit switch and the cost of an additional codec. Also, depending on the video coding algorithm, the line rate requirements may change. For example, a 155.52 Mbit/s signal is expected to carry one HDTV signal or three NTSC signals, while a 622.08 Mbit/s signal carries up to four HDTV signals.

Other early-availability architectures are also possible for carrying video signals to residential customers, but are not discussed here.

5. CONCLUSIONS

This paper described possible services providing a market pull for broadband networks and recent technological advances providing a technology push. A long term broadband network architecture was described based on a double star topology, optical transmission, and high-speed circuit and packet switching. Finally, evolutionary paths for broadband network architectures were described, which gradually introduce broadband network components and capabilities as they become available in order to provide early-availability of some broadband services.

6. ACKNOWLEDGEMENTS

We would like to acknowledge the work of Tom Helstern, Craig Valenti and Mih Yin and the helpful comments of Howard Bussey, Harry Cykiert, Lloyd Linnell, Stagg Newman, Howard Sherry, and Steve Walters.

REFERENCES

1. P. E. White, "The Broadband ISDN - the Next Generation Telecommunications Network," *Proc. ICC*, June 1986.

2. D. R. Spears, "Broadband ISDN Switching Capabilities from a Services Perspective," *J. Select. Areas in Commun.*, Oct. 1987.

3. D. R. Spears, "Broadband ISDN - Service Visions and Technological Realities," *Int. J. Digital and Analog Cabled Systems*, Vol. 1, April 1988.

4. C. F. Hemrick, R. W. Klessig and J. M. McRoberts, "Switched Multi-megabit Data Service and Early Availability Via MAN Technology," submitted to *IEEE Communications Magazine*.
5. B. R. Schatz, "Telesophy - System for Manipulating Knowledge," *Proc. Globecom*, Nov. 1987.
6. J. L. Gimlett, M. Stern, R. S. Vodhanel and N. K. Cheung, G. K. Chang, H. P. Leblanc, P. W. Shumate and A. Suzuki, "Transmission Experiments at 560 Mbit/s and 140 Mbit/s Using Single-Mode Fiber and 1300nm LEDs," *Electron. Lett.*, Vol. 21, Nov. 1985.
7. L. A. Reith, P. W. Shumate and Y. Koga, "Laser Coupling To Single-Mode Fibre Using Graded-Index Lenses And Compact Disc 1-3 μm Laser Package," *Electron. Lett.*, Vol. 22, July 1986.
8. G. Hayward, A. Gottlieb, S. Jain and D. Mahoney, "CMOS VLSI Applications in Broadband Circuit Switching," *IEEE J. Select. Areas Commun.*, Oct. 1987.
9. "American National Standard for Optical Interface Rates and Formats," ESCA Committee, T1 Draft, Sept. 1987.
10. M. W. Beckner, T. T. Lee and S. E. Minzer, "A Protocol and Prototype for Broadband Subscriber Access to ISDN's," *Proc. ISS*, March 1987.
11. L. T. Wu, S. H. Lee and T. T. Lee, "Dynamic TDM: A Packet Approach to Broadband Networking," *Proc. ICC*, June 1987.
12. H. E. Bussey and F. D. Porter, "A Second Generation Prototype for Broadband Integrated Access and Packet Switching," *Proc. Globecom*, Nov. 1988.
13. C. Day, J. Giacopelli and J. Hickey, "Applications of Self-Routing Switches to LATA Fiber Optic Networks," *Proc. ISS*, March 1987.
14. J. Coudreuse, M. Servel, "Prelude: An Asynchronous Time-Division Switched Network", *Proc. ICC*, June 1987.
15. S. Morita, T. Katsuyama, I. Kazuhiko, and H. Hayami, "Elastic Basket Switching - Application to Distributed PBX", *Proc. ISS*, March 1987.
16. K. Eng, M. Hluchyj and Y. Yeh, "A Knockout Switch for Variable-Length Packets", *Proc. ICC*, June 1987.
17. G. Hayward, L. Linnell, D. Mahoney and L. Smoot, "A Broadband ISDN Local Access System using Emerging-Technology Components," *Proc. ISS*, March 1987.
18. L. R. Linnell and D. R. Spears, "A Broadband System Prototype," *Proc. NCF*, Sept. 1987.
19. I. Cidon, I. Gopal and H Heleis, "PARIS - An approach to integrated private networks", *Proc. ICC*, June 1987.
20. K. W. Lu and R. S. Wolff, "Cost Analysis for Switched Star Broadband Access", *Proc. Globecom*, Nov. 1988.
21. "Next Generation Switch Symposium," Bellcore Special Report, SR-TSY-000755, Issue 1, June 1987.
22. R. Olshansky and V. A. Lanzisera, "60-Channel FM Video Subcarrier Multiplexed Optical Communications System," *Electron. Lett.*, Vol. 23, No. 22, Oct. 1987.

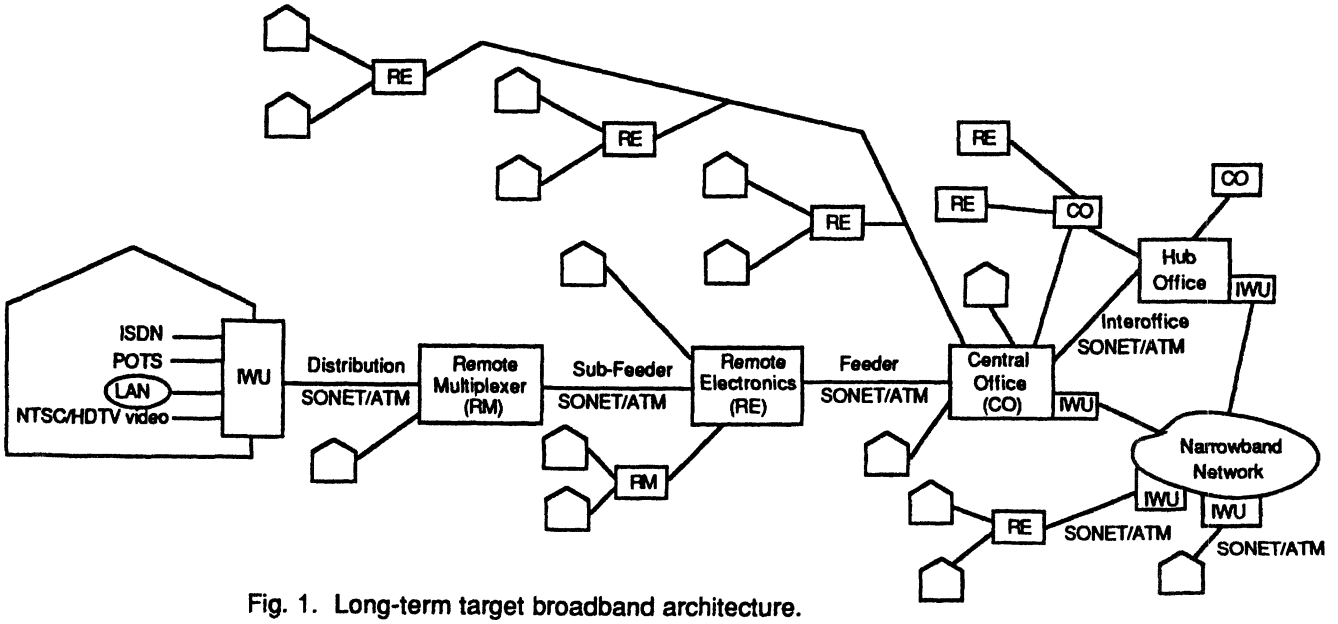


Fig. 1. Long-term target broadband architecture.

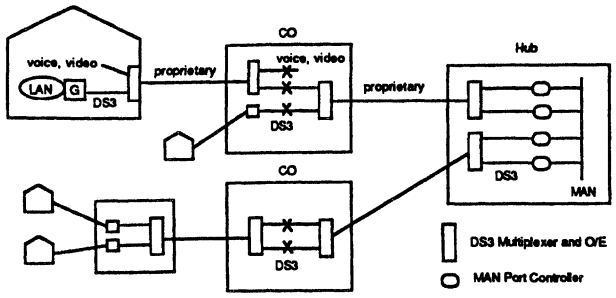


Fig. 2. Early availability architecture for providing SMDS service using MAN technology.

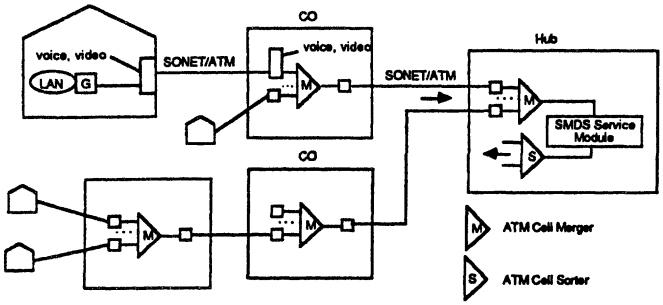


Fig. 3. Architecture for providing SMDS service using ATM technology.

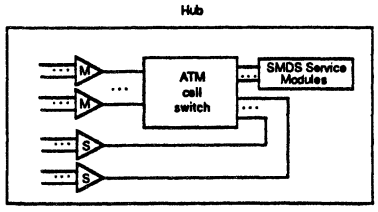


Fig. 4. Expanding the capacity of a switch for providing SMDS service.

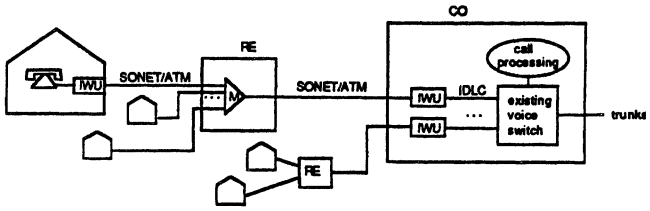


Fig. 5. Architecture for POTS-on-fiber using ATM and existing voice switches.

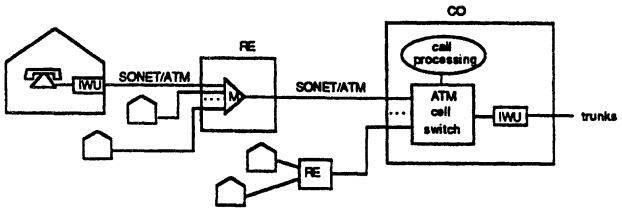


Fig. 6. Architecture for POTS-on-fiber using ATM cell switch.

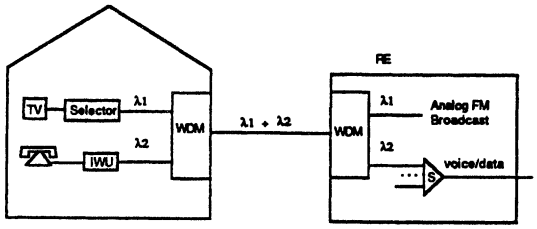


Fig. 7. Architecture for adding broadcast analog video.

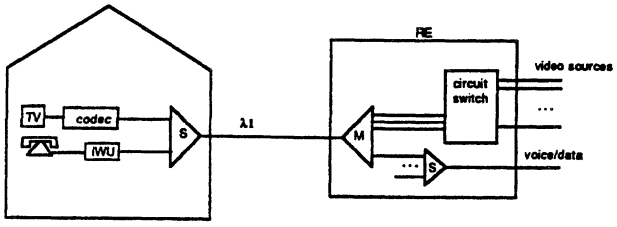


Fig. 8. Architecture for adding switched digital video.

Reprinted from *IEEE Communications*, Vol. 29, No. 3, March 1989, pages 8-15. Copyright © 1989 by The Institute of Electrical and Electronics Engineers, Inc. All rights reserved.

SONET: Now It's the Standard Optical Network

Ralph Ballart
Yau-Chau Ching

SONET (SYNCHRONOUS OPTICAL NETWORK) IS the name of a newly adopted standard, originally proposed by Bellcore (Bell Communications Research) for a family of interfaces for use in Operating Telephone Company (OTC) optical networks. With single-mode fiber becoming the medium of choice for high-speed digital transport, the lack of signal standards for optical networks inevitably led to a proliferation of proprietary interfaces. Thus, the fiber optics transmission systems of one manufacturer cannot optically interconnect with those of any other manufacturer, and the ability to mix and match different equipment is restricted. SONET defines standard optical signals, a synchronous frame structure for the multiplexed digital traffic, and operations procedures.

SONET defines standard optical signals, a synchronous frame structure for the multiplexed digital traffic, and operations procedures.

SONET standardization began during 1985 in the T1X1 subcommittee of the ANSI-accredited Committee T1 to standardize carrier-to-carrier (e.g., NYNEX-to-MCI) optical interfaces. Clearly, such a standard would also have an impact on intracarrier networks, and for that reason has been a subject of great interest for many carriers, manufacturers, and others. Initial T1 standards for SONET rates and formats and optical parameters have now been completed. The history and technical highlights of the SONET standard and its applications are the subject of this paper.

Since it began in the post-divestiture environment, SONET standardization can be thought of as a paradigm for the development of new transmission signal standards. Bellcore's original SONET proposal was not fully detailed because all the technical questions were not yet answered. However, some aspects of the proposal have been carried through the entire process and are now part of the final standards. These include:

- The need for a family of digital signal interfaces, since the march of technology is going to continually increase optical interface bit rates
- The use of a base rate SONET signal near 50 Mb/s to accommodate the DS3 electrical signal at 44.736 Mb/s
- The use of synchronous multiplexing to simplify multiplexing and demultiplexing of SONET component signals, to obtain easy access to SONET payloads and to exploit the increasing synchronization of the network
- Support for the transport of broadband (> 50 Mb/s) payloads
- Specification of enough overhead channels and functions to fully support facility maintenance

As standardization progressed, two key challenges emerged, the solution of which gave SONET universal application. The first was to make SONET work in a plesiochronous[1] environment and still retain its synchronous nature; the solution was the development of payload pointers to indicate the phase of SONET payloads with respect to the overall frame structure (see "SONET Signal Standard—Technical Highlights"). The second was to extend SONET to become an international transmission standard, and thereby begin to resolve the incompatibilities between the European signal hierarchy (based on 2.048 Mb/s) and the North American hierarchy (based on 1.544 Mb/s). Toward the latter goal, the International Telegraph and Telephone Consultative Committee (CCITT) standardization of SONET concepts began in 1986 and the first Recommendations (standards) were completed in June 1988.

This paper will not present a full technical picture of the national and international SONET standards. Instead, we will concentrate on those aspects of the standards and standardization process that are of particular interest. In the next section, a brief and instructional history of the SONET standard is presented. As philosopher George Santayana said, "Those who cannot remember the past are condemned to repeat it." We will then discuss key technical aspects of the SONET standard. Finally, an outline of future work is given in the final section.

[1] As defined in CCITT, corresponding signals are plesiochronous if their significant instants occur at nominally the same rate, any variation in rate being constrained within specified limits.

EH0301-2/91/0000/0097/$01.00 © 1989 IEEE

A History of SONET in T1 and CCITT

The standardization of SONET in T1 started in two different directions and in three areas. First, the Interexchange Carrier Compatibility Forum (ICCF), at the urging of MCI in 1984, requested T1 to work on standards that would allow the interconnection of multi-owner, multi-manufacturer fiber optic transmission terminals (also known as the mid-fiber meet capability). Of several ambitious tasks that ICCF wanted addressed to ensure a full mid-fiber meet capability, two were submitted to T1. A proposal on optical interface parameters (e.g., wavelength, optical power levels, etc.) was submitted to T1X1 in August 1984 and, after three and a half years of intensive work, resulted in a draft standard on single-mode optical interface specifications [1]. The ICCF proposal on long-term operations was submitted to T1M1 and resulted in a draft standard on fiber optic systems maintenance [2].

In February 1985, Bellcore proposed to T1X1 a network approach to fiber system standardization that would allow not only the interconnection of multi-owner, multi-manufacturer fiber optic transmission terminals, but also the interconnection of fiber optic network elements of varying functionalities. For example, the standard would allow the direct interconnection between several optical line terminating multiplexers, manufactured and owned by different entities, and a digital cross-connect system. In addition, the proposal suggested a hierarchical family of digital signals whose rates are integer multiples of a basic module signal, and suggested a simple synchronous bit-interleaving multiplexing technique that would allow economical implementations. Thus, the term Synchronous Optical NETwork (SONET) was coined. This proposal eventually led to a draft standard on optical rates and formats [3]. For the remainder of this paper, the focal points are the history and highlights of the rates and formats document. However, one should always be reminded that this document is only one part of the inseparable triplet: optical interface specifications, rates and formats specifications, and operations specifications.

As it turned out, the notions of a network approach and simple synchronous multiplexing had been independently investigated by many manufacturers. Some of them were already developing product plans, thus complicating the standards process. With the desire of the network providers (i.e., the OTCs) for expedited standards, SONET quickly gained support and momentum. By August 1985, T1X1 approved a project proposal based on the SONET principle. Because the issues on rates and formats were complex and required diligent but timely technical analyses, a steady stream of contributions poured into T1X1. Several ad hoc groups were formed and interim meetings were called to address them. The contributions came from over thirty entities, representing the manufacturers and the network providers alike.

In the early stage, the main topic of contention was the rate of the basic module. From two original proposals of 50.688 Mb/s (from Bellcore) and 146.432 Mb/s (from AT&T), a new rate of 49.920 Mb/s was derived and agreed on. In addition, the notion of a Virtual Tributary (VT) was introduced and accepted as the cornerstone for transporting DSl services. By the beginning of 1987, substantial details had been agreed upon and a draft document was almost ready for voting. Then came CCITT.

The SONET standards were first developed in T1X1 to serve the U.S. telecommunications networks. When CCITT first expressed its interest in SONET in the summer of 1986, major procedural difficulties appeared. According to the established protocol, only contributions that had consensus in T1X1 were forwarded, through U.S. Study Group C, to CCITT. As a result, some aspects of U.S. positions in CCITT appeared to lack flexibility without input from T1X1. Additionally, the views of other administrations in CCITT were not thoroughly understood in T1X1. There were also differences in schedule and perceived urgency. CCITT runs by a four-year plenary period and their meetings are six to nine months apart, while T1 approves standards whenever they are ready and its technical subcommittees meet at least four times a year. While T1X1 saw the SONET standard as a way to stop the proliferation of incompatible fiber optic transmission terminals, no such need was perceived by many other nations whose networks were still fully regulated and non-competitive.

TABLE I. CCITT Rec. G.702 Asynchronous Digital Hierarchies (in Mb/s)

Level	North America	Europe	Japan
1	1.544 (DS1)	2.048	1.544
2	6.312 (DS2)	8.448	6.312
3	44.736 (DS3)	34.368	32.064
4	—	139.264	97.728

The procedural difficulties were partially resolved when representatives from the Japanese and British delegations started to participate in T1X1 meetings in April 1987. These representatives not only gave to T1X1 the perspectives of two important supporters of an international SONET standard, they also served as a conduit between T1X1 and CEPT, the European telecommunications organization.

Separately, interests in an international SONET standard also gained support in the US. Spearheaded by Bellcore, informal discussions in search of an acceptable solution took place in a variety of forums, and contributions in support of this standard were submitted to both T1 and CCITT. Many of these informal discussions had the highest level of corporate support from several U.S. companies, including manufacturers and network providers.

In July 1986, CCITT Study Group XVIII began the study of a new synchronous signal hierarchy and its associated Network Node Interface (NNI). The NNI is a non-media-specific network interface and is distinct from the user-network interface associated with Broadband ISDN. The interaction between T1X1 and CCITT on SONET and the new synchronous hierarchy was fascinating to the participants and will probably alter the way international standards are made in the future. The U.S. wanted an international standard, but not at the price of scrapping SONET or seriously delaying an American national standard upon which OTC networks were planned. The CCITT was not used to working so quickly on so complicated an issue, but was concerned about being supplanted by the T1 committee in the development of new standards.

The U.S. first formally proposed SONET to CCITT for use in the NNI at the February 1987 Brasilia meeting; this proposal had a base signal level (rate) near 50 Mb/s. Table I shows that the European signal hierarchy has no level near 50 Mb/s, and therefore CEPT wanted the new synchronous hierarchy to have a base signal near 150 Mb/s to transport their 139.264 Mb/s signal.

Thus, the informal European response was that the U.S. must change from bit interleave to byte interleave multiplexing to provide a byte organized frame structure at 150 Mb/s. However, there was still no indication from many administrations that an international standard was either desirable or achievable. It took T1X1 three months and three meetings to agree to byte-interleaving and the results were submitted to CCITT as a new T1X1 draft standard document. Thus, T1X1 never gave up the responsibility of developing a SONET standard for the U.S. and, while conceding changes to CCITT wishes, progress was made in other areas of the U.S. standard.

After CCITT met again in Hamburg in July 1987, a formal request was made to all administrations to consider two alternative proposals for an NNI specification near 150 Mb/s. The U.S. proposal was based on the SONET STS-3 frame structure; the STS-3 frame could be drawn as a rectangle with 13 rows and 180 columns of bytes. CEPT proposed, instead, a new STS-3 frame with 9 rows and 270 columns. (Commonly referred to as the 9-row/13-row debate, this prompted one amateur poet to chide that neither conforms to the correct SON(N) ET format of 14 lines.) An NNI near 150 Mb/s received unanimous support because it was assumed that future broadband payloads would be about that size. A North American basic module near 50 Mb/s could be easily derived in both proposals, with a frame structure of either 13 rows and 60 columns or 9 rows and 90 columns.

The Europeans wanted a 9-row frame structure to accommodate their 2.048 Mb/s primary rate signal. This signal has 32 bytes per 125 μs, but in the 13-row proposal could only be accommodated in the most straightforward way using three 13-byte columns, or 39 bytes. The Europeans decried this waste of bandwidth, and refused to consider any alternative (and more efficient) mapping of the 2.048 Mb/s signal into the 13-row structure. Their 9-row frame structure could carry the U.S. 1,544 Mb/s primary rate signal (requiring about 24 bytes/125 μs) in 3 columns of 9 bytes and the 2.048 Mb/s signal in 4 columns of 9 bytes.

The CEPT 9-row proposal called for changes in both the rate and format in the U.S., just as T1X1 was about to complete the SONET standard. However, the request also carried an attractive incentive from a CEPT subcommittee, who stated in a letter that these were the only changes necessary for an international agreement. In addition, the text of the CEPT proposal was based largely on the T1X1 draft document, so that it was complete. Therefore, after the Hamburg meeting, there was tremendous international pressure on the U.S. to accept the 9-row proposal. After some intense debates in T1X1, the U.S. agreed to change.

Unfortunately, the CEPT proposal did not have unanimous support from all CEPT administrations. While some administrations were anxious to get an international standard, a few became concerned that the 9-row proposal favored the U.S. DS3 signal over the CEPT 34.368 Mb/s signal. A CEPT contribution to the November 1987 CCITT meeting stated that it was too early to draft Recommendations on a new synchronous hierarchy. Little progress was made at that meeting and the international SONET standard was in serious jeopardy.

Many T1X1 participants were upset at the apparent change in CEPT's position. Since there were no alternative proposals from CCITT at its November meeting, T1X1 decided to approve the two SONET documents for T1 letter balloting. However, the balloting schedule was deliberately set such that it fell between the CCITT meeting at the beginning of February 1988 and the T1X1 meeting at the end of February 1988. This scheduling allowed a last ditch attempt for an international agreement. In CCITT, a mad rush to rescue the international standard also took place. In addition to a series of informal discussions, a pre-CCITT meeting was held in Tokyo to search for a compromise. Under the skillful helmsmanship of Mr. K. Okimi of Japan, the CCITT meeting in Seoul proposed one additional change to the U.S. draft standards. The new proposal called for a change in the order that 50 Mb/s tributaries are byte-multiplexed to higher SONET signal levels. It also put more emphasis on the NNI as a 150 Mb/s signal by including optional payload structures to better accommodate the European 34.368 Mb/s signal. The U.S. CCITT delegates eventually viewed this proposal as a minor change to the U.S. standard (minor to the extent that equipment under development would probably not require modification) and agreed to accept it. An extensive set of three CCITT Recommendations was drafted and approved by the working party plenary. The U.S. acceptance of these changes was predicated on the understanding that no additional changes of substance would be considered in approving the final versions of the Recommendations.

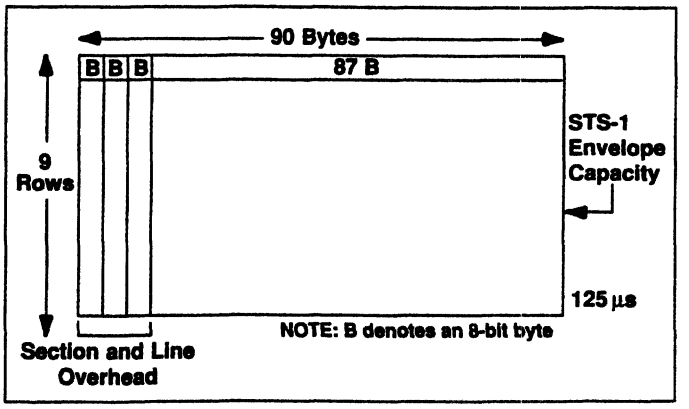


Fig. 1. STS-1 frame.

In February 1988, T1X1 accepted the new changes at its meeting in Phoenix. T1 default balloting based on the change was completed in May and the final passage of the American national standard is expected this summer. Editorial corrections to the CCITT Recommendations [4–6] were completed in June during the Study Group XVIII meeting and with their final approval later this year, the international SONET standard will be born!

SONET Signal Standard—Technical Highlights

In this section, we describe the technical highlights of the American national standards related to SONET. We use U.S. rather than CCITT terminology, although everything described is consistent with both the American national standards and the CCITT Recommendations.

SONET Signal Hierarchy

The basic building block and first level of the SONET signal hierarchy is called the Synchronous Transport Signal—Level 1 (STS-1). The STS-1 has a bit rate of 51.84 Mb/s and is assumed to be synchronous with an appropriate network synchronization source. The STS-1 frame structure can be drawn as 90 columns and 9-rows of 8 bit bytes (Figure 1). The order of transmission of the bytes is row by row, from left to right, with one entire frame being transmitted every 125 μs. (125 μs frame period supports digital voice signal transport, since these signals are encoded using 1 byte/125 μs = 64 kb/s.) The first three columns of the STS-1 contain section and line overhead bytes (see the following subsection). The remaining 87 columns and 9-rows are used to carry the STS-1 Synchronous Payload Envelope (SPE); the SPE is used to carry SONET payloads including 9 bytes of path overhead (see next section). The STS-1 can carry a clear channel DS3 signal (44.736 Mb/s) or, alternatively, a variety of lower-rate signals such as DS1, DS1C, and DS2.

No physical interface for the STS-1 signal has been defined as yet; the Optical Carrier—Level 1 (OC-1) is obtained from the STS-1 after scrambling (to avoid long strings of ones and zeros and allow clock recovery at receivers) and electrical-to-optical conversion. The OC-1 is the lowest-level optical signal to be used at SONET equipment and network interfaces.

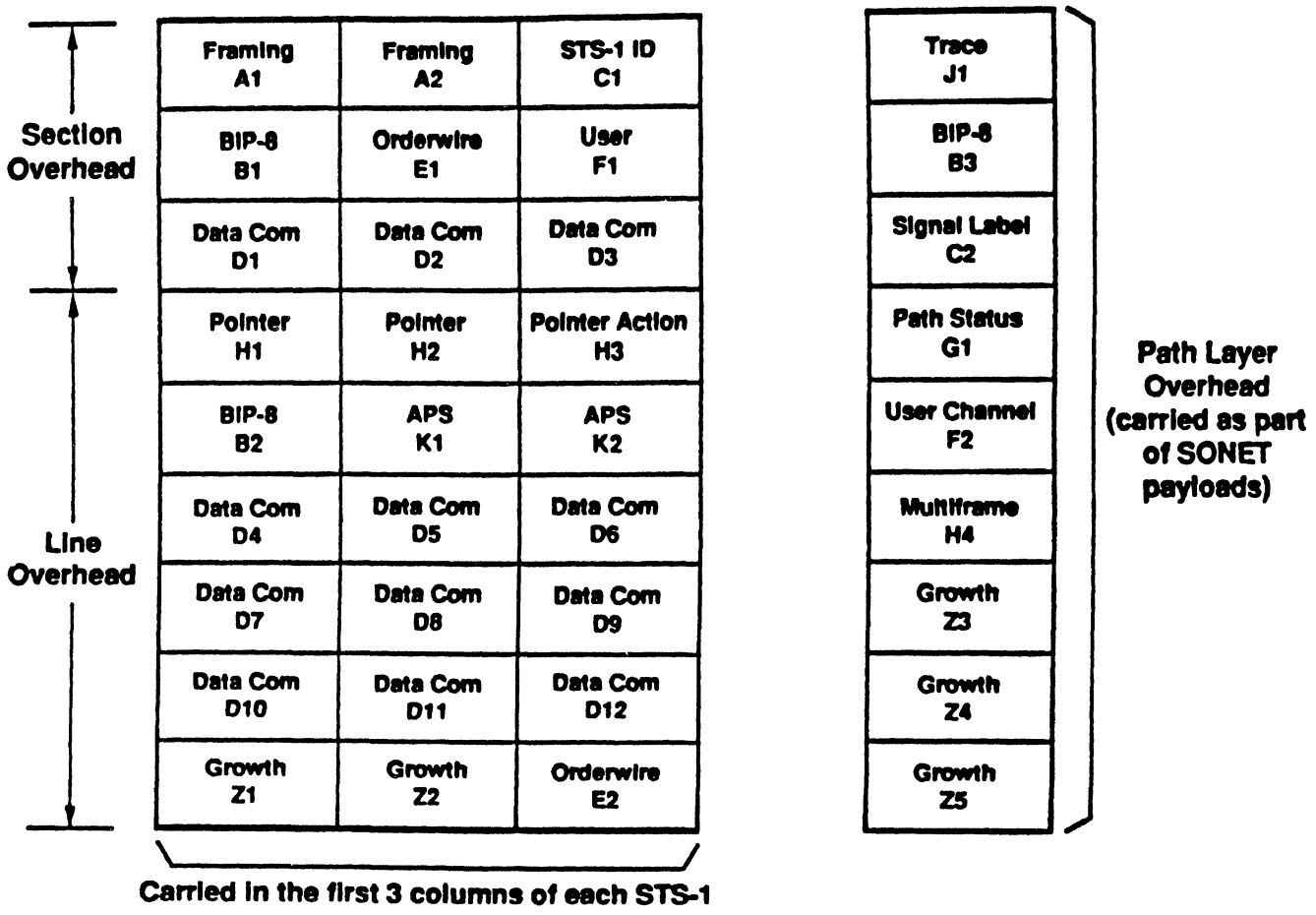


Fig. 2. SONET overhead bytes.

SONET Overhead Channels

The SONET overhead is divided into section, line, and path layers; Figure 2 shows the overhead bytes and their relative positions in the SONET frame structure. This division clearly reflects the segregation of processing functions in network elements (equipment) and promotes understanding of the overhead functions. The section layer contains those overhead channels that are processed by all SONET equipment including regenerators. The section overhead channels for an STS-1 include two framing bytes that show the start of each STS-1 frame, an STS-1 identification byte, an 8-bit Bit-Interleaved Parity (BIP-8) check for section error monitoring, an orderwire channel for craft (network maintenance personnel) communications, a channel for unspecified network user (operator) applications, and three bytes for a section level data communications channel to carry maintenance and provisioning information. When a SONET signal is scrambled, the only bytes left unscrambled are the section layer framing bytes and the STS-1 identification bytes. The second (link) layer of the section data communications channel protocol is LAPD while ISO 8473 is under study for the third (network) layer; higher layers of the protocol will be defined in future updates of the standard.

The line overhead is processed at all SONET equipment except regenerators. It includes the STS-1 pointer bytes (discussed below), an additional BIP-8 for line error monitoring, a two-byte Automatic Protection Switching (APS) message channel (both *1* + *1* and *1* by *N* protection are supported), a nine-byte line data communications channel, bytes reserved for future growth, and a line orderwire channel. The higher layers of the line data communications channel are not specified in the current version of the SONET standard.

The path overhead bytes are processed at SONET STS-1 payload terminating equipment; that is, the path overhead is part of the SONET STS-1 payload and travels with it. The path overhead includes a path BIP-8 for end-to-end payload error monitoring, a signal label byte to identify the type of payload being carried, a path status byte to carry maintenance signals, a multiframe alignment byte to show DS0 signaling bit phase, and others.

Multiplexing

Higher rate SONET signals are obtained by first byte-interleaving *N* frame-aligned STS-1s to form an STS-*N* (Figure 3). Byte-interleaving and frame alignment are used primarily to obtain a byte-organized frame format at the 150 Mb/s level that is acceptable to the CCITT; as discussed below, frame alignment and byte-interleaving also help an STS-*N* to carry broadband payloads of about 150 or 600 Mb/s. All the section and line overhead channels in STS-1 #1 of an STS-*N* are used; however, many of the overhead channels in the remaining STS-1s are unused. (Only the section overhead framing, STS-1 ID, and BIP-8 channels and the line overhead pointer and BIP-8 channels are used in all STS-1s in an STS-*N*.) The STS-*N* is then scrambled and converted to an Optical Carrier—Level *N* (OC-*N*) signal. The OC-*N* will have a line rate exactly *N* times that of an OC-1. Table II shows the OC-*N* levels allowed by the American national standard.

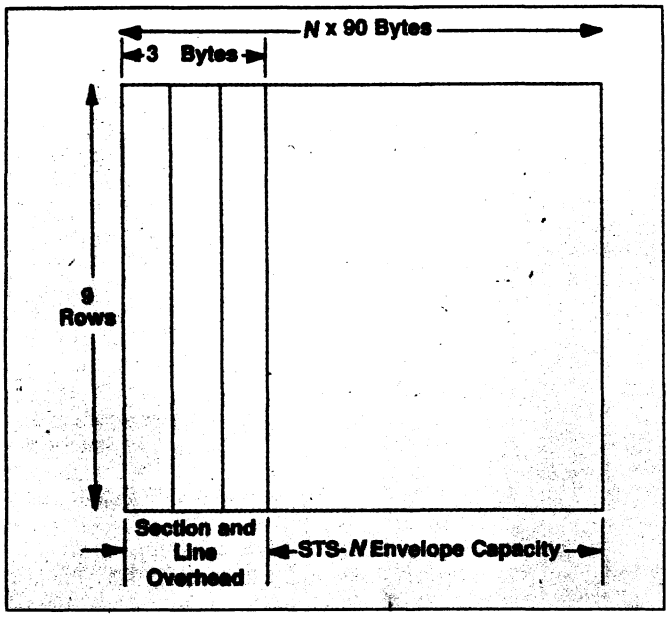


Fig. 3. STS-N frame

SONET STS-1 Payload Pointer

Each SONET STS-1 signal carries a payload pointer in its line overhead. The STS-1 payload pointer is a key innovation of SONET, and it is used for multiplexing synchronization in a plesiochronous environment and also to frame align STS-*N* signals.

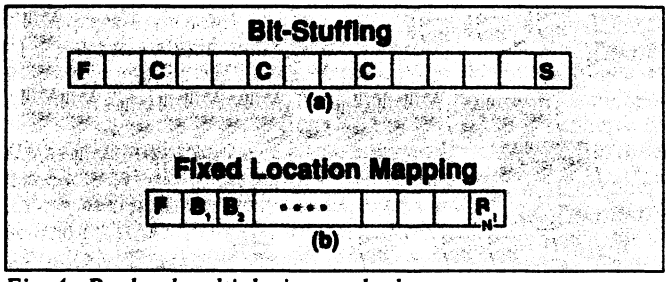


Fig. 4. Payload multiplexing methods.

Pointers and Multiplexing Synchronization

There are two conventional ways to multiplex payloads into higher-rate signals. The first is to use positive bit-stuffing to increase the bit rate of a tributary signal to match the available payload capacity in a higher-rate signal. As shown in Figure 4a, bit-stuffing indicators (labeled C) are located in fixed positions with respect to signal frame *F* and indicate whether the stuffing bit *S* carries real or dummy data in each higher-level signal frame. Examples of bit stuffing are the multiplexing of four DS1 signals into the DS2 signal and the multiplexing of seven DS2 signals into the asynchronous DS3 signal. Bit-stuffing can accommodate large (asynchronous) frequency variations of the multiplexed payloads. However, access to those payloads from the higher-level multiplexed signal is conceptually difficult, since the tributary signal must first be destuffed (real bits separated from the dummy bits) and then the framing pattern of the payload must be identified if complete payload access is required.

TABLE II. Levels of the SONET Signal Hierarchy

Level	Line Rate (Mb/s)
OC-1	51.84
OC-3	155.52
OC-9	466.56
OC-12	622.08
OC-18	933.12
OC-24	1244.16
OC-36	1866.24
OC-48	2488.32

The second conventional method is the use of fixed location mapping of tributaries into higher-rate signals. As network synchronization increases with the deployment of digital switches, it becomes possible to synchronize transmission signals to the overall network clock. Fixed location mapping is the use of specific bit positions in a higher-rate synchronous signal to carry lower-rate synchronous signals; for example, in Figure 4b, frame position B2 would always carry information from one specific tributary payload. This method allows easy access to the transported tributary payloads, since no destuffing is required. The SYNTRAN DS3 signal is an example of a synchronous signal that uses fixed location mapping of its tributary DS1 signals. However, there is no guarantee that the high-speed signal and its tributary will be phase-aligned with each other. Also, small frequency differences between the transport signal and its tributary signal may occur, due to synchronization network failures or at plesiochronous boundaries. Therefore, multiplexing equipment interfaces require 125-μs buffers to phase-align and slip (repeat or delete a frame of information to correct frequency differences) the tributary signal. These buffers are undesirable because of the signal delay that they impose and the signal impairment that slipping causes.

In SONET, payload pointers represent a novel technique that allows easy access to synchronous payloads while avoiding the need for 125-μs buffers and associated slips at multiplexing equipment interfaces. The payload pointer is a number carried in each STS-1 line overhead (bytes H1, H2 in Figure 2) that indicates the starting byte location of the STS-1 SPE payload within the STS-1 frame (Figure 5). Thus, the payload is not locked to the STS-1 frame structure as it would be if fixed location mapping was used but instead floats with respect to the STS-1 frame. (The STS-1 section and line overhead byte positions determine the STS-1 frame structure; note in Figure 5 that the 9-row-by-87-column SPE payload maps into an irregular shape across two 125-μs STS-1 frames.)

Any small frequency variations of the STS-1 payload can be accommodated by either increasing or decreasing the pointer value; however, the pointers cannot adjust to asynchronous frequency differences. For example, if the STS-1 payload data rate is high with respect to the STS-1 frame rate, the payload pointer is decremented by one and the H3 overhead byte is used to carry data for one frame (Figure 6). If the payload data rate is slow with respect to the STS-1 frame rate, the data byte immediately following the H3 byte is nulled for one frame and the pointer is incremented by one (Figure 7). Thus, slips and their associated data loss are avoided while the phase of the STS-1 synchronous payload is immediately known by simply reading the pointer value. Thus, SONET pointers combine the best features of the positive bit-stuffing and fixed location mapping methods. Of course, these advantages come at the cost of having to process the pointers; however, pointer processing appears readily implementable in today's Very Large Scale Integration (VLSI) technologies.

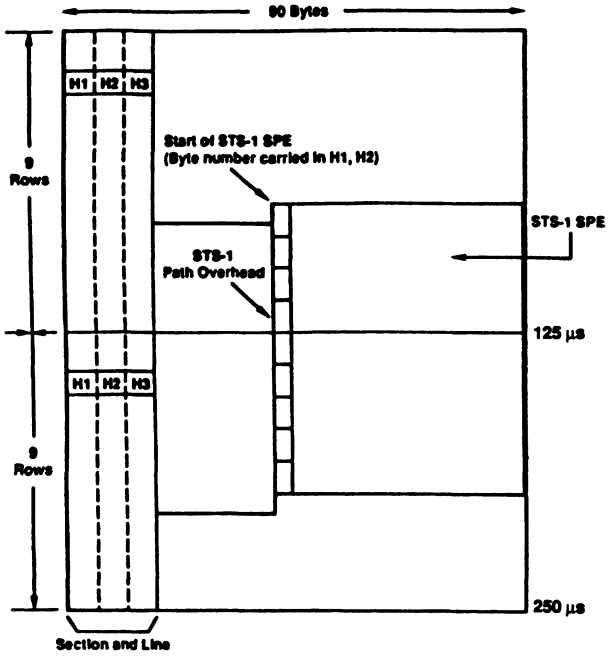


Fig. 5. STS-1 SPE in interior of STS-1 frame.

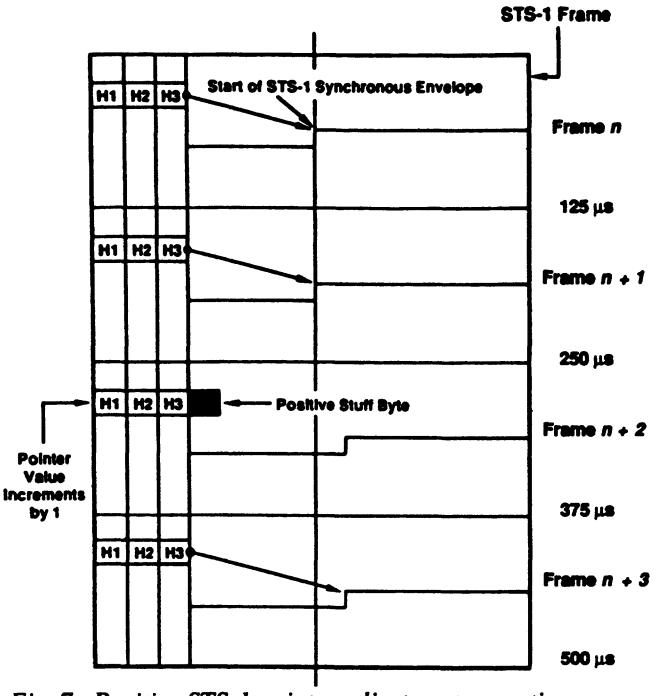


Fig. 7. Positive STS-1 pointer adjustment operation.

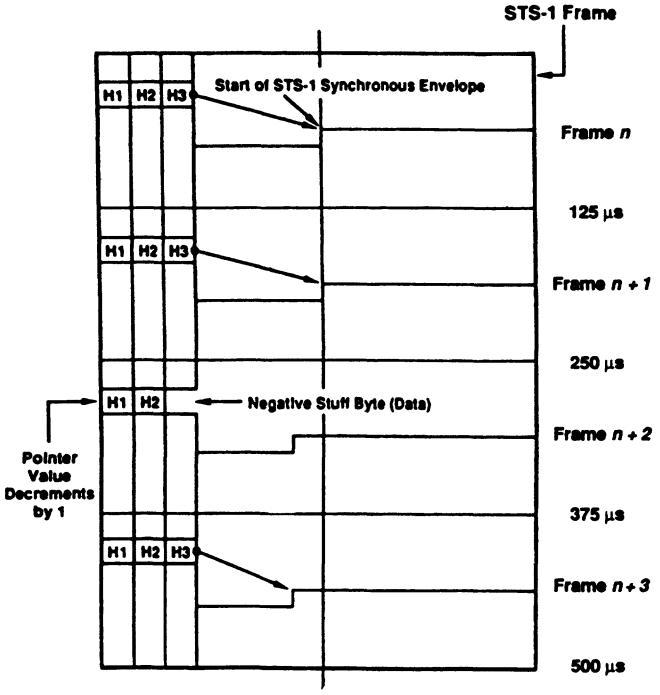


Fig. 6. Negative STS-1 pointer adjustment operation.

Broadband Payload Transport with Payload Pointers

As discussed above, STS-1 payload pointers can be used to adjust the frequencies of several STS-1 payloads in multiplexing to the STS-*N* signal level. As this is done, the various STS-1 section and line overhead bytes are frame-aligned. In Figure 8, two hypothetical and simplified SONET frames (A and B) are out of phase with respect to the arbitrary, outgoing (multiplexed) SONET signal phase. By recalculating the SONET pointer values and regenerating the SONET section and line overhead bytes, two phase-aligned signals (A and B) are formed. A and B can then be byte-interleaved to form a higher level STS-*N* signal. As shown, this can be done with minimum payload buffering and signal delay.

With frame alignment, the STS-1 pointers in an STS-*N* are grouped together for easy access at an OC-*N* receiver using a single STS-*N* framing circuit. If it is desired to carry a broadband payload requiring, for example, three STS-1 payloads, the phase and frequency of the three STS-1 payloads must be locked together as the broadband payload is transported through the network. This is easily done by using a "concatenation indication" in the second and third STS-1 pointers. The concatenation indication is a pointer value that indicates to an STS-1 pointer processor that this pointer should have the same value as the previous STS-1 pointer. Thus, by frame aligning STS-*N* signals and using pointer concatenation, multiple STS-1 payloads can be created. The STS-*N* signal that is locked together in this way is called an STS-*N*c, where the "c" stands for concatenated. Allowed values of STS-*N*c are STS-2c, STS-3c, STS-6c, STS-9c, etc. For broadband User Network Interfaces (UNI), STS-3c and STS-12c are of particular interest.

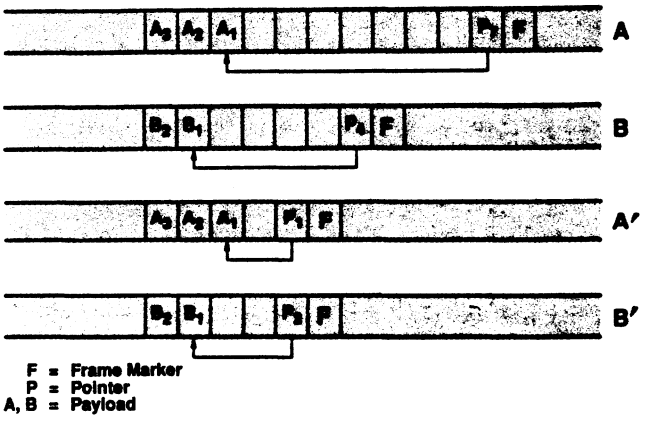


Fig. 8. Frame alignment using pointers.

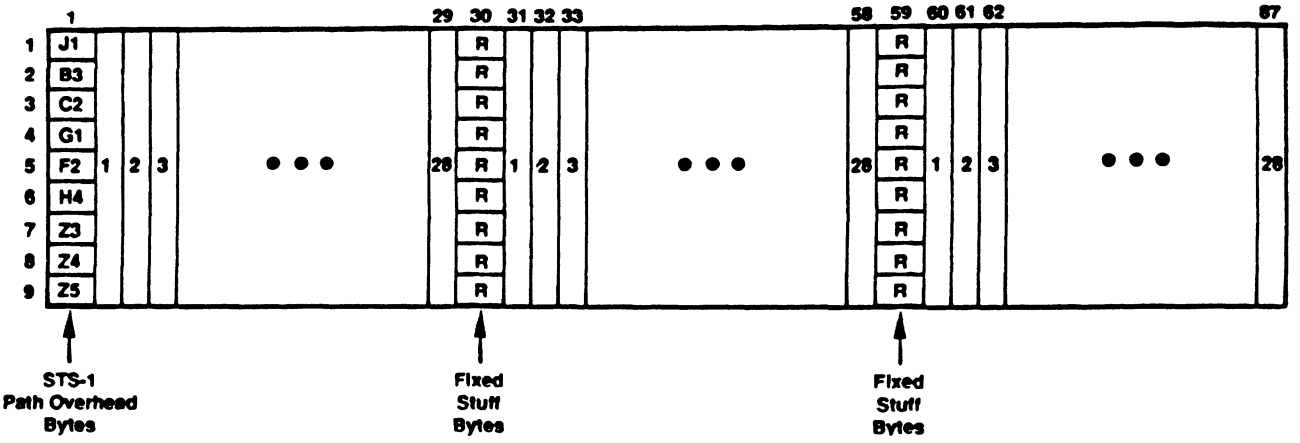


Fig. 9. VT-structured STS-1 SPE: all VT1.5.

As discussed in the section on the history of SONET standards, the Europeans had no interest in using the SONET STS-1 signal. Instead, they were interested in using a base signal of about 150 Mb/s to allow transport of their 139.264 Mb/s electrical signal and for possible Broadband ISDN applications. As the above discussion shows, the technical solution to this problem is the use of the STS-3c signal. In the U.S., we can continue to think of this signal as three concatenated STS-1 signals. In Europe and the CCITT, the STS-3c is considered as the basic building block of the new synchronous hierarchy and is referred to as the Synchronous Transport Module—Level 1 (STM-1).

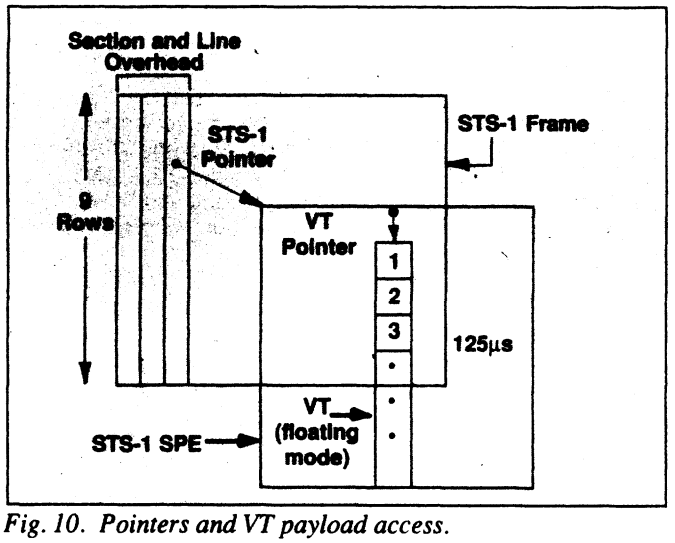


Fig. 10. Pointers and VT payload access.

Sub-STS-1 Payloads

To transport payloads requiring less than an STS-1 payload capacity, the STS-1 SPE is divided into payload structures called virtual tributaries (VTs). There are four sizes of VTs: VT1.5, VT2, VT3, and VT6, where each VT has enough bandwidth to carry a DS1, CEPT-1 (2.048 Mb/s), DS1C, and DS2 signal, respectively. Each VT occupies several 9-row columns within the SPE. The VT1.5 is carried in three columns (27 bytes), the VT2 in 4 columns (36 bytes), the VT3 in six columns (54 bytes), and the VT6 in twelve columns (108 bytes).

A VT group is defined to be a 9-row-by-12-column payload structure that can carry four VT1.5s, three VT2s, two VT3s, or one VT6. Seven VT groups (84 columns), one path overhead column, and two unused columns are byte-interleaved to fully occupy the STS-1 SPE. Figure 9 shows the STS-1 SPE configured to carry 28 VT1.5s. VT groups carrying different VT types can be mixed within one STS-1. As discussed in the section on history, the ability of the 9-row format structure to flexibly carry both the 1.544 and the 2.048 Mb/s signals was a necessary step in reaching an international agreement on SONET.

Two different modes have been adopted for transporting payloads within a VT. The VT operating in the "floating" mode improves the transport and cross-connection of VT payloads. A floating VT is so called because a VT pointer is used to show the starting byte position of the VT SPE within the VT payload structure. In this sense, the operation of the VT pointer is directly analogous to that of the STS-1 pointer, and has the same advantages of minimizing payload buffers and associated delay when mapping signals into the VT. Figure 10 shows conceptually how the STS-1 and VT pointers are used to locate a particular VT payload in an STS-1. The other VT mode is the "locked" mode. The locked VT does not use the VT pointer, but instead locks the VT payload structure directly to the STS-1 SPE. (Of course, the STS-1 SPE still floats with respect to the STS-1 frame.) The locked mode improves the transport and cross-connection of DS0 signals by maintaining the relative phase and frequency of DS0 signals carried in multiple locked VT's. When VT-organized, each STS-1 SPE carries either all floating or all locked VTs.

More than one specific payload mapping is possible with each of the VT modes described above. Asynchronous mappings are used for clear channel transport of nominally asynchronous signals using the floating mode of operation; conventional positive bit-stuffing is used to multiplex these signals into the VT SPE. "Byte synchronous" mappings have been defined in both the locked and floating modes for the efficient, synchronous transport of DS0 signals and their associated signaling; conventional fixed position mappings are used to carry the DS0's in the VT SPE (floating mode) or VT (locked mode). "Bit synchronous" mappings are used in both the locked and floating modes for the clear channel transport of unframed, synchronous signals. The VT mappings that have been defined in the current version of the American national standard are given in Table III.

TABLE III. Sub-STS-1 Mappings

Mappings	VT (Virtual Tributary) Modes	
	Floating	Locked
Asynchronous	DS1 CEPT-1 DS1C, DS2	—
Byte Synchronous	DS1 CEP-1	DS1 CEP-1 SYNTRAN
Bit Synchronous	DS1 CEP-1	DS1 CEP-1

Optical Parameters

The SONET optical interface parameters were developed in parallel with the SONET rates and formats. The optical parameters specified in the American national standard include spectral characteristics, line rate, power levels, and pulse shapes; jitter specifications will be developed in the next phase of the standard. The current optical specifications extend up to OC-48 (see Table II). It is expected that as more experience is gained with high data rate systems, the optical parameters associated with OC-18, OC-24, OC-36, and OC-48 will be updated.

The intent of this first optical interface standard is to provide specifications for "long reach" fiber transmission systems, i.e., systems using lasers. The second phase of SONET standardization in T1 will address "short reach" specifications for fiber transmission systems based on LEDs and low-power, loop lasers.

Conclusion and Future Work

The Synchronous Optical Network concept was developed to promulgate standard optical transmission signal interfaces to allow mid-section meets of fiber systems, easy access to tributary signals, and direct optical interfaces on terminals, and to provide new network features. The basic SONET signal format can transport all signals of the North American hierarchy up to and including DS3, and also future broadband signals. SONET will soon be an American national standard and a CCITT transmission signal hierarchy standard. The second phase of SONET T1 standardization will fully specify the data communications channel protocols, specify short-reach SONET optical interfaces for use in intra-office applications, and update SONET optical parameters for selected levels above OC-12.

SONET represents a successful test case for standards-making in the post-divestiture environment. Of course, the ultimate test for any standard is the development of products and services that are compliant with the new standard. For specific implementations, requirements beyond those contained in the standard are often needed. For example, Bellcore has issued a series of Technical Advisories giving additional requirements for SONET multiplexes, digital cross-connect systems, and digital switch interfaces. The first field trials of SONET equipment are expected in 1989.

References

[1] ANSI T1.106-1988, "American National Standard for Telecommunications—Digital Hierarchy Optical Interface Specifications, Single Mode," to be issued.

[2] T1M1.2/87-37R2 "Functional Requirements for Fiber Optic Terminating Equipment."

[3] ANSI T1.105-1988 "American National Standard for Telecommunication—Digital Hierarchy Optical Rates and Formats Specification," to be issued.

[4] CCITT Recommendation G.707, "Synchronous Digital Hierarchy Bit Rates," to be issued.

[5] CCITT Recommendation G.708, "Network Node Interface for the Synchronous Digital Hierarchy," to be issued.

[6] CCITT Recommendation G.709, "Synchronous Multiplexing Structure," to be issued.

Biography

Ralph Ballart is District Manager of Network Node Equipment Requirements at Bell Communications Research, Inc. (Bellcore), responsible for the development of requirements for SONET digital cross-connect systems and trunk side digital switch interfaces. Ralph is also Chairman of the T1X1.5 U.S. CCITT Drafting Sub-working Group that has responsibility for drafting contributions related to SONET to CCITT Study Group XVIII.

He received his B.S. degree in physics from the then Polytechnic Institute of Brooklyn in 1973 and his Ph.D. in physics from the University of Arizona in 1980. He joined Bell Telephone Laboratories in 1980 and worked on fundamental network planning tools and network applications for digital cross-connect systems. In 1984, he joined Bellcore and worked on requirements for SYNTRAN equipment. He was promoted to District Manager in 1985; his district participated in the technical development of SONET and its international standardization in CCITT. Dr. Ballart is a member of IEEE Communications Society.

Yau-Chau Ching is District Manager of SONET Interface Standards, responsible for strategic planning to support the SONET project in its initial applications as well as its later extension to Broadband ISDN. He is the Co-Chairman of T1X1.5 Working Group (Optical Hierarchy), where much of the SONET standardization work takes place. Before the inception of T1X1.5, he was the Co-Chairman of T1X1.4 Subworking Group on Optical Rates and Formats and an active member of T1X1.4 Subworking Group on Optical Interface. These latter groups were the predecessors of T1X1.5.

He received his B.E.E.E. degree from City College of New York in 1966 and his Ph.D. from New York University in 1969 under a National Science Foundation fellowship. From 1969 to 1984, he was with Bell Laboratories, where he did a series of exploratory development work in data compression, such as interframe video coding, digital speech interpolation, embedded ADPCM coding, and Fast Packet Network. From 1980 to 1983, he supervised a group responsible for the implementation of the Access Interface to the Fast Packet Network.

Since the divestiture of the Bell System, he has been with Bellcore, where his district was responsible for the generic requirements of fiber optic transmission systems. His district was also the originator of the SONET concept in Bellcore and its strong advocate in T1X1. Dr. Ching is a member of IEEE, Tau Beta Pi, and Eta Kappa Nu, and he holds 11 patents.