Integrated Voice/ Data Switching

Thomas M. Chen
David G. Messerschmitt

Introduction

Historically, voice and data communications have been handled by different communication networks. For example, the primary carrier for voice has been the public switched telephone network whereas data has been handled for the most part by specialized data networks. One reason has been the traditional separation between voice and data applications. Consider a person speaking on the telephone and then sending an electronic message from a computer terminal. The two actions are handled by separate instruments and are perceived as serving separate purposes. Telephony permits immediate, personal, and interactive contact; data messages allow for more pre-meditated, formal, and non-interactive communication. A tremendous need for a unified treatment of voice and data has not existed, so separate special-purpose communication networks have been largely satisfactory.

Another reason for different networks is the fundamentally different characteristics of voice and data signals. Voice is inherently a real-time, analog signal generated by human speakers. Voice signal characteristics, such as spectral density and average activity, are well-known and consistent between different speakers. On the other hand, most data is machine-generated and digital. Data characteristics, such as bit rate and message length, vary widely depending on the particular application.

Interest in "integrating" voice and data communications has been stimulated recently by deregulation of the U.S. telephone industry and international activities in planning the standards for the Integrated Services Digital Network, or ISDN [1-3]. ISDN will be a worldwide digital network offering a wide range of voice and data services based on 64 kbits/s channels. Although ISDN will most likely be comprised of logically separate networks as shown in Figure 1, it will provide subscribers with the functionality of a single, integrated network by offering a standardized, integrated user access to services [4].

Offering voice and data services in a single network promises several benefits. Users achieve convenience, flexibility, and economy. An integrated user interface allows different terminal equipment to be moved and plugged into any interface in the same way that different electrical appliances can use any standard electrical power outlet. Furthermore, services can be customized to individual needs without having to be concerned with the compatibility of different special-purpose networks. For network providers, integration promises benefits in efficiency and economy. Sharing facilities not only increases efficiency, but should also simplify network operations and maintenance, items which will quickly become very complex in a non-integrated network with a proliferation of new services. Reduced network costs should result in lower service prices to users.

This research was supported by a grant from Pacific Bell, with a matching grant from the University of California MICRO Program.

0163-6804/88/0006-0016 $01.00 © 1988 IEEE

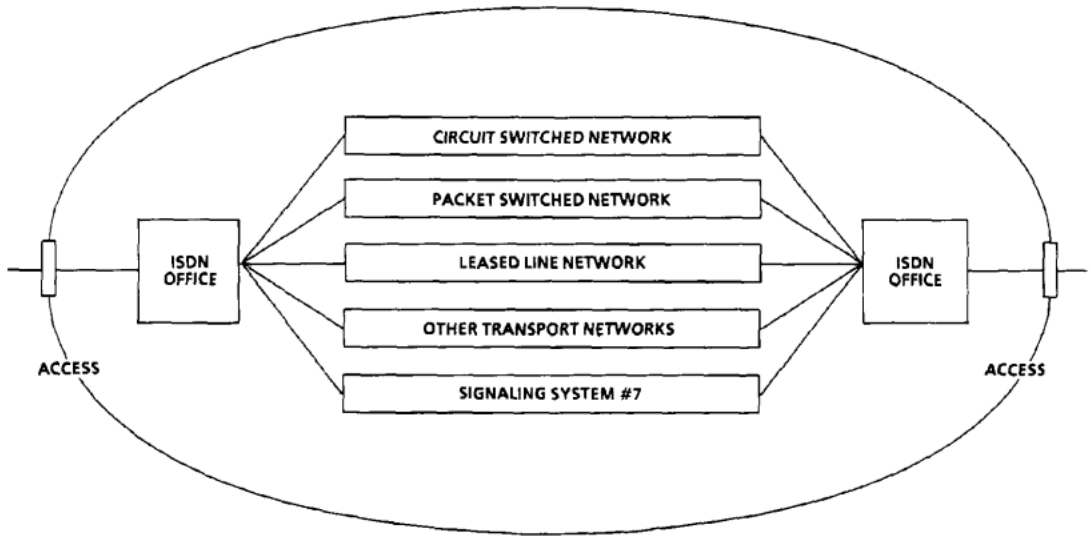


Fig. 1. Probable ISDN architecture.

Integration of voice and data communications is also motivated by advances in computer and communications technology. The telephone system is gradually converting from analog to an entirely digital network because of decreasing costs, increasing data transmission, and increased capability for integration of services. In addition, fiber optics make large bandwidths available at modest cost. This abundant bandwidth combined with increased network capabilities are driving forces in expanding traditional telephone services.

Finally, there is also an expectation that technology will result in the merging of voice and data applications, which have been traditionally separate. For example, artificial intelligence might eventually allow computers to accept and interpret voice commands in person-to-machine conversations. In the futuristic office, messages containing voice and video as well as text might be edited, stored, and transmitted like electronic mail today. Terminal equipment will become more functional and integrate the separate functions of the telephone, terminal, and printer. There will be a need to process and transmit voice and data, and in fact all types of information, in a unified manner.

The purpose of this paper is to provide a basic understanding of the technical problem in integrating voice and data. Although integration is discussed specifically in terms of voice and data, the problem can be generalized to all types of real-time and non-real-time services. First, the different types of traffic found in communication systems are examined. Integration is investigated at different levels. In particular, this paper focuses on the integration of voice and data at the switching level. Different switching approaches are compared and some current integrated switching systems are described.

Types of Traffic

We might think of voice and data information as types of "traffic" to be transported through the communication network. Designing a communication network specifically for either voice or data is relatively uncomplicated; the difficulty in designing a network for both voice and data lies in the different requirements of voice and data traffic. We will describe these requirements by first examining the characteristics of voice and data signals. There are three general classes of traffic in existing communication networks, although more classes might arise in future networks [5].

Voice and video are representatives of the inherently real-time Class I traffic (for our purposes, video can be considered similar to voice except at a higher bandwidth). Voice signals are generated in real-time, person-to-person calls. Due to the conversational nature of speech, only one direction is usually active at any time. Voice traffic can tolerate a certain amount of degradation (e.g., noise, clipping, compression) and occasional blocking (i.e., the connection of a call is refused) without becoming objectionable. However, large transmission delays accentuate subjective problems with echos (unless echo cancelers are used) and themselves disrupt a conversation. Although the exact amount of subjectively acceptable delay is subject to debate, it seems generally agreed that the maximum allowable delay is in the approximate range between 100 and 500 ms [6].

Classes II and III traffic are collectively referred to as "data". Class II traffic consists of person-to-machine (or possibly machine-to-machine) "interactive data", such as videotex. Although not strictly real-time, this traffic has certain delay limitations; a subscriber can wait a fraction of a second, but not minutes, for a

computer to respond to a command. Communication is characteristically "bursty" and asymmetric; that is, this type of traffic takes the form of intermittent bursts of information separated by intervals of silence at unequal rates in the two directions. Class II messages can tolerate short transmission delays but not errors. Class III traffic consists of machine-to-machine "bulk data". Messages are typically unidirectional and relatively long. Not being real-time in nature, messages may be delayed substantially longer than Class II messages and arrive in any random sequential order, but they must arrive without errors.

We can illustrate the differences between traffic classes by placing them in a coordinate system representing their characteristics. For the purpose of illustration, the three coordinates shown in Figure 2 were chosen to represent the traffic's degree of tolerance to: network delay, blocking, or degradation (e.g., source rate reduction, transmission errors, and loss of messages). Class I traffic is relatively tolerant of degradation and blocking but rather intolerant of delay. Class II traffic is more tolerant of delay, less tolerant of blocking, and not tolerant of degradation. Class III traffic is much more tolerant of delay and not tolerant of blocking.

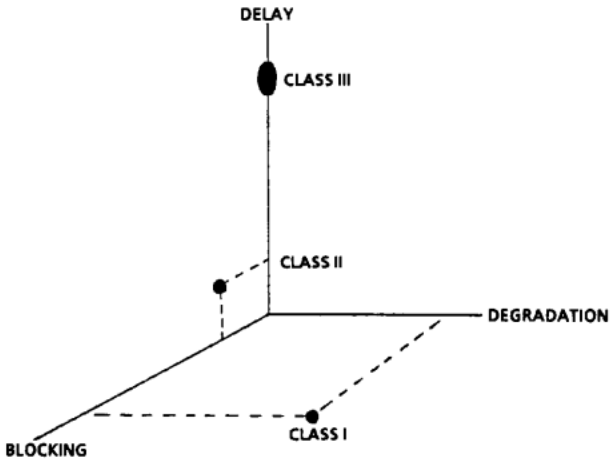


Fig. 2. Traffic tolerances.

These three characteristics were chosen to illustrate the relationship between traffic requirements and switching characteristics. The characteristics of a switching system can be placed in the same coordinate system. An example shown in Figure 3 is a telephone system which has blocking and does not correct for transmission errors. Another example is ARPANET which imposes a variable end-to-end delay but no blocking, and corrects for transmission errors through an elaborate protocol involving error-checking and acknowledgments. If we superimpose the two coordinate systems, it becomes clear that the telephone system most closely meets Class I traffic requirements and ARPANET matches Class II and III traffic requirements. Of course, there are other relevant parameters, such as bandwidth and peak-to-average traffic ratio, that have not been included in this simplified example. Clearly, a switching system that does not impose blocking, degradation, or delay would be ideal for all classes of traffic, but practical systems generally must resort to blocking, delay, or some type of degradation (e.g., discarding messages) to handle the problems of congestion and excessive traffic conditions.

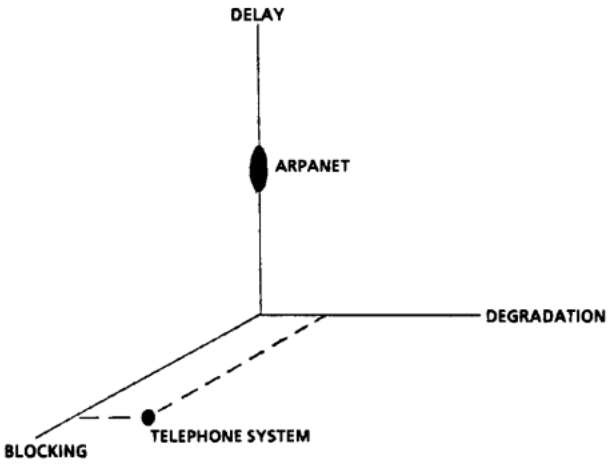


Fig. 3. Switching system characteristics.

Levels of Integration

In a broad sense, integration means that voice and data can be handled by the same communications network. Specifically, voice and data traffic can be integrated at three recognized levels [7], as illustrated in Figure 4. ISDN can be viewed as an example of integration at the first level: integrated access. Voice and data services are accessible through a single user access interface, allowing voice and data terminal equipment to share a common network interface. Voice and data are carried on the same transmission link between the subscriber and network switch called the local loop. However, traffic within the network is routed to different transport networks which are separately optimized for a specific type of service. As planning for ISDN progresses, it is becoming more evident that the real issue at this level is agreement on the standards for the user access interface.

At the second level, called integrated transmission, these transport networks share common transmission facilities between switches but maintain separate switching facilities. Voice and data traffic would share the same transmission link, for example, by Time-Division Multiplexing (TDM) where messages are interleaved in time, or by Frequency-Division Multiplexing (FDM) where messages are sent at separate frequencies. TDM is analogous to the automobile traffic situation where automobiles and pedestrians are allowed use of a road at non-overlapping time intervals. In the FDM case, automobiles and pedestrians are allowed on the same road but in separate lanes. TDM is already used extensively for multiplexing voice signals on long-distance trunks in the telephone network. In the future, transmission links carrying voice and data

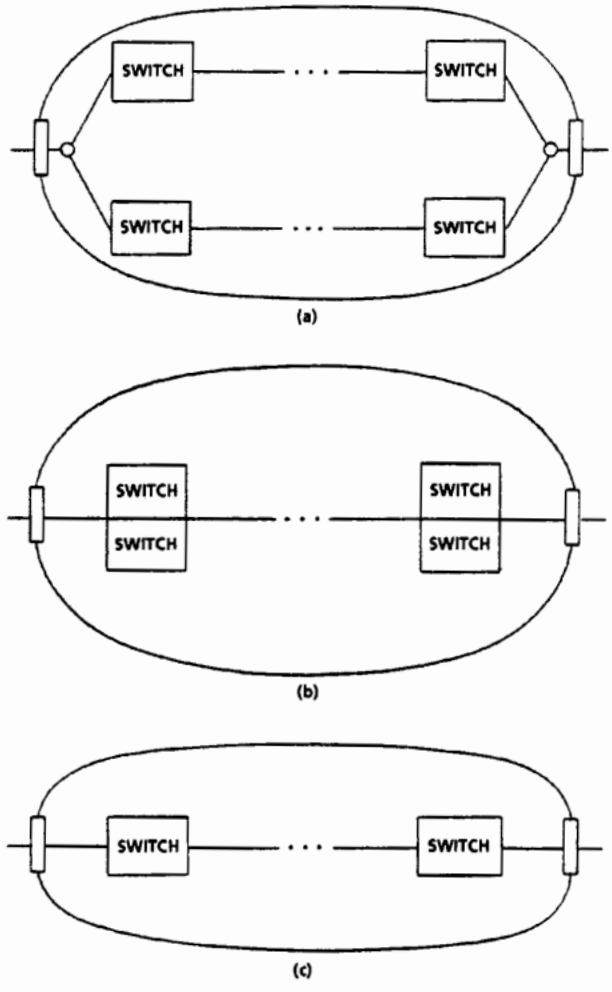


Fig. 4. Levels of integration: (a) Integrated access. (b) Integrated transmission. (c) Integrated switching.

will be predominantly optical fibers and will provide tremendous bandwidth, analogous to 10,000-lane superhighways.

At the third level, called integrated switching, switching facilities are shared as well as transmission links and network access. A network integrated at this level might be considered to be completely integrated since all types of traffic are handled entirely by the same facilities. This level is also the most technically challenging due to the different switching requirements for voice and data in terms of delay, degradation, blocking, and other parameters. Initial approaches have attempted to adapt conventional circuit switching to handle data or conventional packet switching to handle voice. Other approaches have attempted to develop new schemes specifically for both voice and data such as burst switching and hybrid switching. Most recently, the feasibility of an advanced version of packet switching, called wideband or "fast" packet switching, has been demonstrated. It is currently gaining favor as the most promising approach in the telecommunications industry.

Circuit Switching

Conventional switching approaches are circuit switching, packet switching, and message switching. Circuit switching is probably the most familiar since the public telephone system is the primary example of this approach [8]. Circuit switching was developed during the early days of telephony when calls were connected by an operator at a manual switchboard. Although the switching operation is now performed electronically, the principles of circuit switching are still much the same.

The distinguishing feature of circuit switching is the exclusive dedication of a channel of fixed bandwidth between two users for the duration of a call. A circuit or direct electrical path is established during a call set-up procedure, as illustrated in Figure 5. A signaling message indicating a call request is passed through the network to find an available circuit. If a circuit is found and the call is accepted, a signaling message indicating call acceptance is returned. The circuit is held exclusively until the call is disconnected, even if it is not actually utilized for transmission of information.

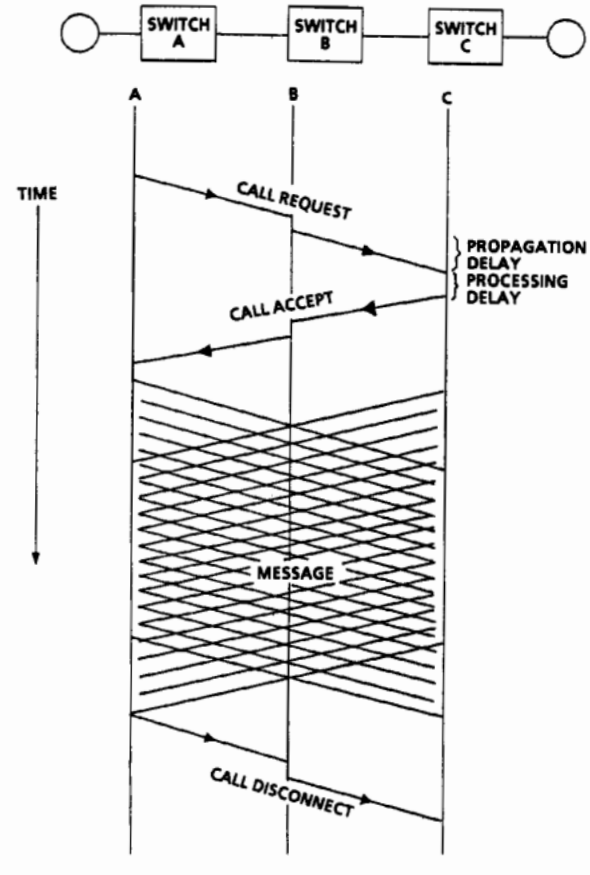


Fig. 5. Timing diagram for circuit switching.

The basic circuit switch, shown in Figure 6, consists of three functional blocks: terminal interface, switching network, and controller. The terminal interface handles signaling functions such as call request and ringing and ensures that the incoming signals are compatible with the electrical characteristics of the switch facilities. The switching network provides the physical transmission paths for the signals. The controller contains the intelligence responsible for the proper operation of the switch hardware.

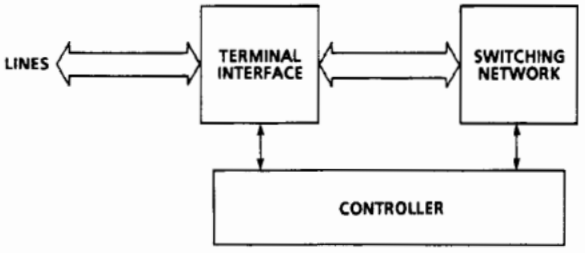


Fig. 6. Basic circuit switch architecture.

Circuit switching has a number of important features. First, traffic is handled on a blocking basis; the network handles excessive traffic conditions by refusing to connect a new call while continuing to hold those calls already connected. The blocked calls may be either delayed for a later time or cleared (e.g., the telephone system clears blocked calls by returning a fast busy signal). Second, this approach is efficient only if the set-up and disconnect times are small compared to the duration or holding time of the call. Third, bandwidth is utilized efficiently only if the circuit is active fairly constantly since the channel is dedicated for the entire call. Fourth, a dedicated channel permits communication with minimal transmission delay, a particularly important consideration for real-time traffic. Finally, traffic is carried by the network without regard to content; that is, information (except for signaling information) is not processed or altered in transit by the network. As a consequence, the network is unable to correct for transmission errors; the responsibility for error detection and correction is relegated to the users.

These properties make conventional circuit switching most suitable for real-time services like voice. Typical conversations last about 300 seconds, much longer than the set-up times, which are approximately a second in the telephone system. During a call, voices are active about 40 to 50 percent of the time, and the network imposes virtually no delay other than propagation time (approximately 20 ms on terrestrial links) from source to destination.

Circuit switching has several disadvantages for most data applications, and in particular for Class II [9]. First, bandwidth is utilized much less efficiently because most data is usually active only about 5 to 15 percent of the time. This means that the channel is nearly always idle. Second, data bursts are often short and require only a brief connection. It is inefficient to circuit-switch messages when the holding time becomes comparable to the call set-up time. Third, circuit switching works well for voice because a 4-kHz bandwidth is characteristic of all speech signals. However, a dynamic allocation of a bandwidth is needed for data communications due to the diversity of bit rates required for various data applications. Finally, circuit switching does not have the capability of node-to-node error-checking since the network is insensitive to the content of the transmitted information. In contrast, if messages are processed within the network, they can be checked for errors before delivery to the destination. This is a desirable feature for error-sensitive data traffic.

Modifications to circuit switching have been proposed to overcome these difficulties. Bandwidth can be utilized more efficiently with statistical multiplexing techniques such as Time Assignment Speech Interpolation (TASI) [10] or Digital Speech Interpolation (DSI) [11]. Several sources can share a fewer number of channels by dynamically selecting only active sources. The number of channels is selected such that the probability that a greater number of sources will be active simultaneously is low. In these infrequent cases, signals would be "clipped" (partially cut off) or compressed (quantized more roughly).

The inefficiency in call connections for very brief messages can be improved by reducing call set-up and disconnect times. The idea of fast circuit switching is to develop switches which perform signaling and call set-up so quickly that it becomes efficient to switch very brief data bursts. Set-up times less than 140 ms are commonly presumed, although such fast digital switches are not presently commercially available [9].

Packet Switching

The concept of packet switching originated as a distributed switching system for survivable military communications [12,13]. The first important application was the development of the ARPANET to link together time-shared research computers in a nationwide network. Unlike circuit switching, packet switching was designed particularly for data communications rather than voice communications. After the success of ARPANET, many other packet-switched networks were started for public data services.

Packet switching is based on the idea of message switching, or store-and-forward switching, shown in Figure 7. Message switching resembles the method of mail delivery in the postal system. A message is formed by concatenating the data information with a header and an end-of-message flag, which is similar to putting a letter into an envelope. The header contains all the information necessary for routing the message through the network information such as source, destination, identity number, and checksum for error. The message is stored in a buffer at each switch which decodes the message header and determines the next node in the route. When the appropriate link becomes available, the message is forwarded to the buffer in the next switch. Usually, an acknowledgment is returned by the receiving node if the message is received without error; otherwise, the sender retransmits the message after some time.

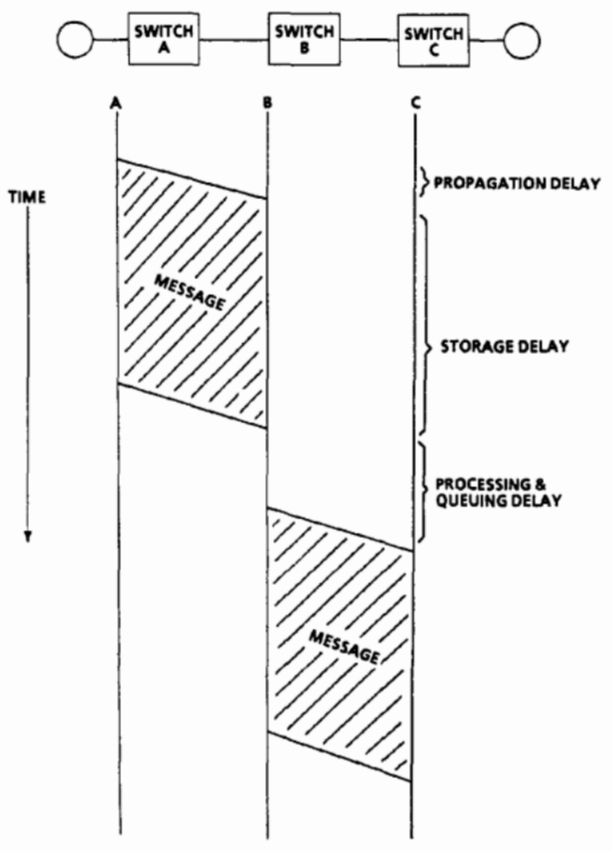


Fig. 7. Timing diagram for message switching.

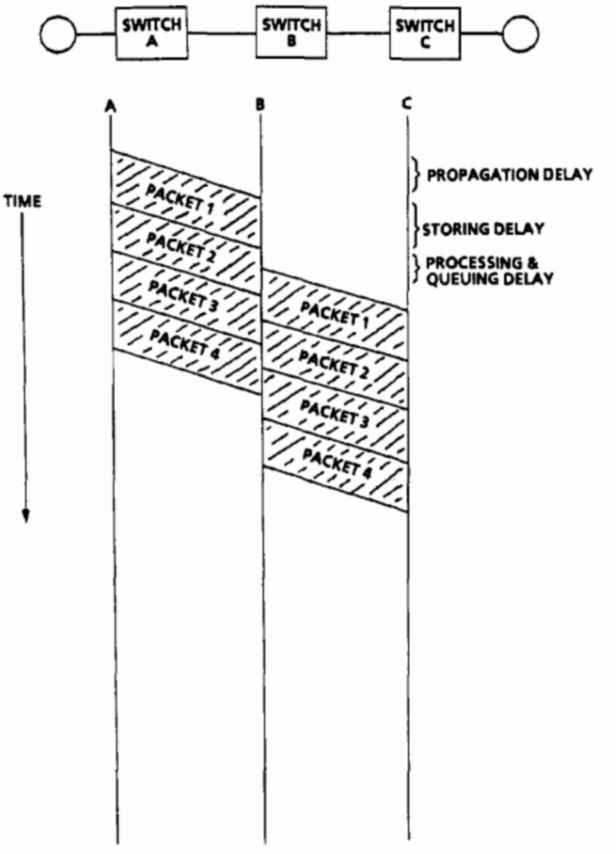


Fig. 8. Timing diagram for packet switching (datagram mode).

Packet switching [14,15] is the same as message switching with one difference; messages are divided into smaller segments of limited length, called packets, each with its own packet header. The shorter packets need less storage time at each switch, as seen in Figure 8. The end-to-end delay for packets is much less than for messages, particularly on routes involving many hops. The exact length of the packet is a trade-off between delay and overhead. Shorter packet lengths decrease the queueing delay at each switch but increase the percentage of the packet taken up by the header bits.

The basic packet switch, shown in Figure 9, consists of four functional blocks: input buffers, output buffers, switch fabric, and controller. Incoming packets are stored in the input buffers, and their headers are decoded. When the appropriate route is determined, they are placed in the proper output buffers through the switch fabric. The switch fabric might be a simple bus or a multi-stage interconnection network. All routing, processing, and control functions are performed by the controller.

There are two general methods for routing packets through networks: datagrams and virtual circuits. In

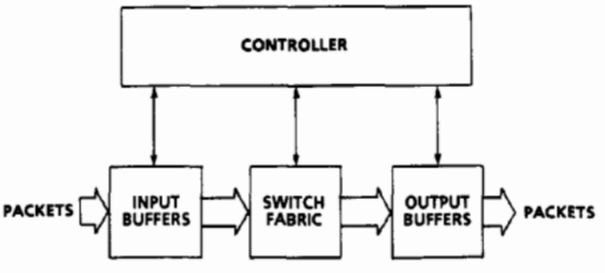


Fig. 9. Basic packet switch architecture.

datagram mode, packets are routed independently. Packets travel any available route to the destination and arrive in a random sequential order. In virtual circuit mode, a "VC request" packet sets up a logical connection between the source and destination. Every subsequent packet between this source and destination will travel this same virtual circuit identified by a number in the header. The virtual circuit is cleared eventually by a "VC disconnect" packet. A virtual circuit is not the same as a dedicated channel in circuit switching

because several virtual circuits can share the same physical circuit. A virtual circuit means only that the routing decision does not need to be performed for every packet. Hence, processing is simpler for multi-packet messages, but every packet still needs to be queued and processed at each switch. Routing by virtual circuit is more vulnerable to node failures and less adaptive to changing traffic conditions than datagrams.

Packet switching has important differences from circuit switching. First, traffic is handled on a delay basis, i.e., the network will accept new packets when traffic is heavy but might impose extremely long queueing delays. The delay at each switch consists of time to store the entire packet, process the header, and wait for an available link. The delay times depend on packet length, header complexity, and network traffic conditions, respectively. Second, there is no signaling involved in call set-up (in datagram mode), but there is overhead associated with the packet headers. These header bits require a portion of the channel capacity and processing at each switch. Third, packet switching allocates bandwidth dynamically instead of pre-allocating bandwidth like in circuit switching. Channel capacity is used only when there is information to be sent. Bandwidth is utilized more efficiently than circuit switching at the expense of increased processing and queueing delays. Finally, and perhaps most importantly, packets are processed as they are being transmitted. This makes it possible to error-check, copy, and even alter packets within the network. Thus, packet switching permits much more direct control over user information than circuit switching.

For its efficiency with "bursty" traffic, packet switching has been used increasingly in commercial data networks. It has not been used for real-time services like voice except in the context of numerous experiments. A basic problem in packet switching voice is the reconstruction of a continuous stream of speech from packets experiencing random and possibly excessive transit delays [16]. It is necessary to impose an additional delay at the speech decoder so that the voice packets are "played-out" with relatively uniform delays. Increasing the delay at the decoder reduces the fraction of "lost" packets (i.e., packets that arrive later than their target playout times), but the delay becomes subjectively annoying beyond a certain limit.

However, for several reasons, packet switching is still attractive for voice. First, packet switching is extremely flexible. For example, a packet can be copied and broadcast to a group of stations as easily as sent point-to-point. Second, packet switching is particularly efficient with bursty traffic, and voice exhibits some "burstiness". Speech actually consists of short, discrete bursts called talkspursts, lasting on the order of milliseconds to seconds. Packetizing these talkspurst and ignoring silence intervals performs a statistical multiplexing function by using channel capacity only when voices are active. Finally, packet switching has the capability of processing and altering speech information within the network, unlike circuit switching, which transports information without processing. This feature is useful, for example, for judiciously abridging or discarding voice packets when traffic becomes congested.

Several modifications have been proposed to reduce the transit delays for voice packets [17-23]. Packet storage delays can be reduced by shorter packet lengths or by "virtual cut-through" switching [24] which allows a packet to be forwarded before it is stored entirely. Packet headers can be simplified by using virtual circuits and not requiring acknowledgments or retransmissions since the reconstructed speech can tolerate a certain amount of degradation. Voice packets can be given higher priority than data packets to reduce queueing delays. This priority can be pre-emptive so that arriving voice packets can interrupt a data packet in progress, or non-preemptive, where arriving voice packets must wait for the packet in progress to finish. When packet delays become excessive, packets can be judiciously abridged or discarded within the network, or the speech source can be instructed to reduce its encoding rate by means of "feedback" control packets. Finally, fast packet switching, an important approach favored by many in the telecommunications industry, is currently being researched and developed. It is discussed in a later section.

Another approach is to minimize the detrimental effect of late voice packets by encoding the speech in a certain manner. In embedded coding [18,19], the speech is encoded at a number of different rates. The encoded information is placed or "embedded" in packets of different priority such that lower priority packets can be discarded without affecting the continuity of the reconstructed speech. The loss of the lower priority packets causes a graceful degradation in the speech quality. For example, a speech encoder operating at the embedded rates of 8, 16, 32, and 64 kbits/s will generate voice packets with four different priority levels. The loss of the lowest priority packets will result in an effective bit rate of 32 kbits/s.

Burst Switching

Burst switching is a form of message switching that combines different features of fast circuit switching [25-27]. The idea is to take advantage of the bursty nature of voice and data traffic. A "burst" is a variable-length message consisting of a 4-byte header followed by an information field and an end-of-burst flag. In a data burst, the information is the data message; in a voice burst, the information is a talkspurt. A third type, a "command" burst, is used to carry network information between burst switches. Like message switching, burst switching uses headers for routing and queueing for bandwidth contention. However, a burst can begin to be forwarded before it is buffered completely because a burst is always transmitted at the same rate, unlike a store-and-forward message that must be buffered and then transmitted at the full rate of the output link. In this way, burst switching resembles fast circuit switching.

Instead of the traditional central control located at the switching center, control in burst switching is partially distributed to numerous small "link switches". Each link switch has limited processing capability. Its own program is capable of exchanging messages with other link switches to execute a service. Link switches handle network access and local switching. Some

processor functions remain in centralized, high-capacity "hub switches" used at points of high concentration. With this distributed architecture, burst switching accommodates network expansion and is less sensitive to node failures and overloads.

Routing information is based on the location of the destination in the network. Local destinations can be handled by the appropriate link switch. Longer distance routes involve accessing a Translation & Routing (T&R) processor for the routing information. The T&R processor returns the appropriate routing information in response to a routing request message. After the routing information is placed in the header, a virtual circuit is made by exchanging a series of messages. The routing information in the header instructs each switch how to route the burst. The virtual circuit is held until it is broken down by a disconnect message at the end of a call.

Although bursts are handled by the same switching facilities, different types of burst are switched differently. Command bursts have highest priority, voice second priority, and data lowest priority. When bursts contend for a link, the queued burst with highest priority is sent first. The protocol for command and data bursts involves aborting and retransmitting the bursts in case of incorrect delivery. Voice bursts are not retransmitted because speech information is extremely time-sensitive. In addition, when 2 ms of speech samples accumulate at a burst switch before an output link becomes available, the information is discarded by the switch, resulting in clipping of the reconstructed speech.

Hybrid Switching

Hybrid switching attempts to provide both circuit and packet switching features [5,28,29]. This is accomplished by time-multiplexing voice and data as shown in Figure 10. A master time frame is established on each link consisting of a synchronizing Start-Of-Frame (SOF) marker and a number of time slots. Voice traffic is allotted a certain number of slots while data is allowed on the remaining slots. The voice slots are circuit switched while the data slots are packet switched.

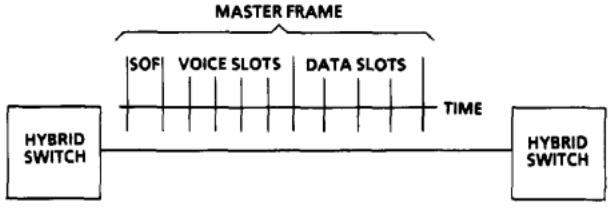


Fig. 10. Hybrid switching.

In the basic scheme, the frame length and voice/data boundary are fixed, and time slots are equal lengths. For example, a master frame of 10 ms on 1.544 Mbits/s links would provide frames of 15,440 bits at a rate of 100 frames/s. Voice sources at 16 kbits/s would need 160-bit time slots. However, a fixed voice/data boundary is inefficient due to the statistical fluctuations in voice and data traffic. Temporarily unused voice slots cannot be used for data nor unused data slots for voice. It is difficult to place the voice/data boundary for maximum efficiency under changing traffic patterns. For more efficiency, the voice/data boundary can be "movable". Voice calls can be allotted up to a maximum number of time slots; data is allowed to use temporarily silent voice slots as well as slots allocated just to data. Furthermore, the frame length and time slots can be dynamic for more adaptability. As might be expected, increased efficiency is achieved at the cost of complicated analysis and switch operations.

The idea of hybrid switching is to combine the features of circuit and packet switching in order to handle each type of traffic in the conventional manner, i.e., voice is handled on a blocking basis and data on a delay basis. Although voice and data traffic must be separated to different fabrics within the switch, the hybrid switch presents the functionality of a single, integrated switch at each link. With both circuit and packet switching capabilities, hybrid switching will effectively accommodate any mix of traffic. Furthermore, hybrid switching is more compatible with the present telephone system than packet or burst switching because existing circuit switches can be updated to handle the data switching requirement. Thus, hybrid switching allows for a more graceful evolution of the telephone network.

Fast Packet Switching

Fast packet switching, also called Asynchronous Time Division (ATD) or Asynchronous Transfer Mode (ATM), is an advanced version of packet switching based on fiber optic links and simplified protocols [30-37]. A protocol is the set of rules that implement the various functions involved in the transfer of data between two users (e.g., routing, error control, flow control, etc.). Due to their high degree of complexity, conventional protocols are structured hierarchically as a set of layers referred to as the network architecture [15,38,39]. Each layer performs specific functions of the protocol and provides certain functions or "services" to higher layers while relying on the services provided by lower layers.

The most well-known network architecture is the Open Systems Interconnection (OSI) reference model developed by the International Standards Organization (ISO) as a framework for standardization of protocols [40]. Its seven layers are shown in Figure 11. When an "application", (the highest level entity), communicates with another "application", its data message is passed down the layers with each layer appending its own header and containing information to be used in the same layer at the destination. At the destination, the appropriate headers are removed as the data is passed up the layers. Each layer communicates with its "peer" in the same layer.

The lowest layer, called the physical layer, covers all the physical aspects of transmitting a bitstream between adjacent nodes. Above the physical layer, the data link layer is primarily responsible for making the

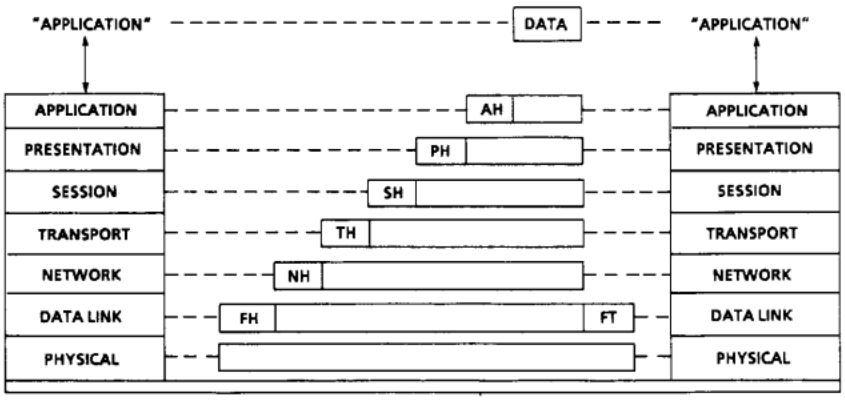
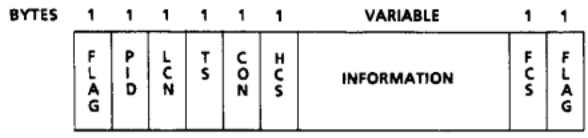


Fig. 11. OSI protocol reference model.

physical link reliable. This is accomplished through a usually complicated procedure involving error detection, acknowledgments, and retransmissions. The next higher layer is the network layer, which encompasses the functions of routing and congestion control. The four remaining layers, above the network layer, involve end-to-end protocols that are less relevant to the switching network.

The key aspect of fast packet switching is simplification of the lower layer protocols. Specifically, the functions of error correction and flow control can be removed from the lower layers because of the abundant bandwidth and low error rates possible with fiber optic links. Error correction and flow control are provided on an end-to-end basis by higher layer protocols as needed (i.e., for data but not for voice). The philosophy is to handle all types of traffic in a common manner in the lower layers and to overlay more traffic-specific protocols in the higher layers. In addition, routing is simplified. The simplified protocols and routing result in a concise header format, similar to the example shown in Figure 12, which enables the processing of packets to be performed entirely in hardware. The combination of high-speed links and hardware processing makes possible fast packet switches with low delay and high throughput sufficient for voice and possibly video services [31,41-44].



FLAG : PACKET DELIMITER
PID : PACKET IDENTIFICATION
LCN : LOGICAL CHANNEL NUMBER
TS : TIME STAMP
CON : CONTROL FIELD DEPENDENT ON APPLICATION
HCS : HEADER CHECK SEQUENCE
FCS : FRAME CHECK SEQUENCE

Fig. 12. A fast packet format.

A Comparative Discussion of Switching Approaches

Comparisons of switching approaches have focused on circuit and packet switching for many reasons. These technologies are the predominant ones today and will continue to dominate because of the enormous investment in existing telephone and data networks. At least, any future switching techniques will still be based on a circuit or packet switched backbone network. It is more practical to adapt a proven technology than to implement a new technology. Another reason for their dominance is their proven efficiency and effectiveness for voice or data. The final reason is that circuit and packet switching will soon be implemented in the worldwide ISDN. If successful, ISDN will evolve into the broad-band-ISDN [45,46]. The broad-band-ISDN will also be circuit and packet-switched and will provide the user with much higher bit rates at the interfaces to the network.

Harrington [9] analyzed circuit switching techniques for voice and data, and concluded that circuit switching is not the preferable approach for data services. Coviello [47] compared circuit and packet switching for voice and concluded that conventional packet switching was inadequate without major modifications. The general consensus in comparisons of circuit and packet switching [5,48,49] is that each technology has a different area of usefulness depending on parameters such as message length, network topology, traffic patterns, etc. Thus, both circuit and packet switching are commonly expected to be useful in communication networks carrying a mix of different types of traffic, for at least the near future, which is encouraging to the proponents of the hybrid switching approach.

Performance evaluations of burst switching have been limited. An analytical comparison between burst switching and fast packet switching [50] indicated that both techniques performed roughly equivalently for

both voice and data. However, burst switching distinguishes between voice and data at the switching level whereas fast packet switching does not. Given that packet switching is a proven technology and will be implemented more widely in the near future, burst switching seems to be disadvantageous in terms of implementation without offering significant performance advantages.

Hybrid switching seems to be a practical short-term approach because present network facilities are based on circuit and packet technologies and hybrid switching is capable of effectively handling a wide mix or different types of traffic. However, hybrid switching is essentially a packaging of two switching fabrics into a single functional switch, and it is not clear that the potential benefits derived from integration can be achieved with hybrid switching. It has the disadvantages of complicated network design and analysis [29] and switch architectures [51,52].

On the other hand, the fast packet switching approach is leading to switches capable of extremely low delays and high throughput. The concept overcomes the drawbacks usually presented in arguments against packet switching while retaining the inherent advantages of flexibility and efficiency. Also, voice and data are truly transported in an integrated manner at the switching level. Consequently, fast packet switching is rapidly emerging in the telecommunications industry as the favored long-term approach.

Conclusions

We have presented a discussion of the technical problem of integrating voice and data, while focusing on the switching level. In particular, we have considered modifications of circuit switching and packet switching, which are conventionally used separately for voice and data, plus burst and hybrid switching.

As might be expected, conventional circuit and packet switching have different regions or usefulness, and neither is suitable without significant modifications. Burst switching suffers from the disadvantages of incompatibility with present networks, without appearing to offer any major performance advantages. Hybrid switching is promising as a short-term approach in terms of network evolution and performance, but presents some significant technical difficulties. On the other hand, fast packet switching is a promising solution to the integration problem, but more research and development is needed.

An example of a problem that lacks thorough understanding is congestion control. Although flow control is exercised end-to-end (e.g., sliding window for data and call blocking for voice), congestion control might still be necessary on a node-to-node basis due to the random nature of traffic flows in packet networks. We speculate that congestion control can be implemented by simply discarding packets, in which case a mechanism for choosing which packets to discard might be needed. A possible mechanism could be a priority system based on an implicit expiration time associated with each packet. However, the implications of this conjectured scheme need to be studied further.

Acknowledgment

The authors are grateful to Dr. Patrick White of Bell Communications Research for his helpful comments on an earlier version of this manuscript.

References

[1] I. Dorros, "ISDN," *IEEE Commun. Mag.*, vol. 19, pp. 16-19, Mar. 1981.

[2] I. Dorros, "Telephone nets go digital," *IEEE Spectrum*, vol. 20, pp. 45-53, Apr. 1983.

[3] W. Tang, "ISDN- new vistas in information processing," *IEEE Commun. Mag.*, vol. 24, pp. 11-16, Nov. 1986.

[4] W. Gifford, "ISDN user-network interfaces," *IEEE J. on Sel. Areas in Commun.*, vol. SAC-4, pp. 343-348, May 1986.

[5] M. Ross, "Circuit versus packet switching," in *Fundamentals of Digital Switching*, (J. McDonald, ed.). NY: Plenum, 1983.

[6] J. Turner, "Design of an integrated services packet network," *IEEE J. on Sel. Areas in Commun.*, vol. SAC-4, pp. 1373-1380, Nov. 1986.

[7] M. Gerla, R. Pazos-Rangel, "Bandwidth allocation and routing in ISDN's," *IEEE Commun. Mag.*, vol. 22, pp. 16-26, Feb. 1984.

[8] A. Joel, Jr., "Circuit-switching fundamentals," in *Fundamentals of Digital Switching*, (J. McDonald, ed.). NY: Plenum, 1983.

[9] E. Harrington, "Voice/data integration using circuit-switched networks," *IEEE Trans. on Commun.*, vol. COM-28, pp. 781-793, June 1980.

[10] K. Bullington, J. Fraser, "Engineering aspects of TASI," *Bell Sys. Tech. J.*, vol. 38, pp. 353-364, Mar. 1959.

[11] S. Campanella, "Digital speech interpolation," *COMSAT Tech. Rev.*, vol. 6, pp. 127-158, Spring 1976.

[12] L. Roberts, "The evolution of packet switching," *Proc IEEE*, vol. 66, pp. 1307-1313, Nov. 1978.

[13] P. Green, Jr., "Computer communications: milestones and prophecies," *IEEE Commun. Mag.*, vol. 22, pp. 49-63, May 1984.

[14] H. Heggestad, "An overview of packet-switching communications," *IEEE Commun. Mag.*, vol. 22, pp. 24-31, Apr. 1984.

[15] A. Tanenbaum, *Computer Networks*. NJ: Prentice-Hall, 1981.

[16] W. Montgomery, "Techniques for packet voice synchronization," *IEEE J. on Sel. Areas in Commun.*, vol. SAC-1, pp. 1022-1028, Dec. 1983.

[17] B. Gold, "Digital speech networks," *Proc. IEEE*, vol. 65, pp. 1636-1658, Dec. 1977.

[18] T. Bially, *et al.*, "Voice communications in integrated digital voice and data networks," *IEEE Trans. on Commun.*, vol. COM-28, pp. 1478-1490, Sept. 1980.

[19] T. Bially, *et al.*, "A technique for adaptive voice flow control in integrated packet networks," *IEEE Trans. on Commun.*, vol. COM-28, pp. 325-333, March 1980.

[20] G. Barberis, *et al.*, "Coded speech in packet-switched networks: models and experiments," *IEEE J. on Sel. Areas in Commun.*, vol. SAC-1, pp. 1028-1038, Dec. 1983.

[21] C. Weinstein, J. Forgie, "Experience with speech communications in packet networks," *IEEE J. on Sel. Areas in Commun.*, vol. SAC-1, pp. 963-980, Dec. 1980.

[22] J. Forgie, A Nemeth, "An efficient packetized voice/data network using statistical flow control," *ICC*, pp. 38.2.44-48, June 1977.

[23] M. Listanti, F. Villani, "An X.25-compatible protocol for packet voice communications," *Computer Commun.*, vol. 6, pp. 23-31, Feb. 1983.
[24] P. Kermani, L. Kleinrock, "Virtual cut-through: a new computer communication switching technique," *Computer Networks*, vol. 3, pp. 267-286, 1979.
[25] S. Amstutz, "Burst switching — an introduction," *IEEE Commun. Mag.*, vol. 21, pp. 36-42, Nov. 1983.
[26] E. Haselton, "A PCM frame switching concept leading to burst switching network architecture," *IEEE Commun. Mag.*, pp. 13-19, Sept. 1983.
[27] J. Haughney, "Application of burst-switching technology to the defense communication systems," *IEEE Commun. Mag.*, vol. 22, pp. 15-21, Oct. 1984.
[28] G. Coviello, P. Vena, "Integration of circuit/packet switching in a SENET (slotted envelope network) concept," *NTC '75*, pp. 42.12-42.17.
[29] I. Gitman, *et al.*, "Analysis and design of hybrid switching networks," *IEEE Trans. on Commun.*, vol. COM-29, no. 9, pp. 1290-1300, 1981.
[30] J. Turner, "New directions in communications (or which way to the information age?)," *IEEE Commun. Mag.*, vol. 24, pp. 8-15, Oct. 1986.
[31] R. Muise, *et al.*, "Experiments in wideband packet technology," *1986 Zurich Sem. on Dig. Commun.*, pp. D4.1-5.
[32] M. Dieudonne, M. Quinquis, "Switching techniques for asynchronous time division multiplexing (or fast packet switching)," *ISS '87*, pp. B5.1.1-6.
[33] P. Gerke, "Fast packet switching- a principle for future system generations?," *ISS '87*, pp. B5.2.1-7.
[34] P. Kirton, *et al.*, "Fast packet switching for integrated network evolution," *ISS '87*, pp. B6.2.1-7.
[35] G. Luderer, *et al.*, "Wideband packet technology for switching systems," *ISS '87*, pp. B6.1.1-7.
[36] W. Standish, S. Sistla, "Network switching in the 1990's," *ISS '87*, pp. C5.1.1-9.
[37] K. Takami, T. Tatenaka, "Architectural and functional aspects of a multi-media packet switched network," *ISS '87*, pp. B6.4.1-5.
[38] W. Stallings, *Data and Computer Communications*. NY: MacMillan, 1985.
[39] M. Schwartz, *Telecommunication Networks: Protocols, Modeling and Analysis*. MA: Addison-Wesley, 1987.
[40] H. Zimmermann, "OSI reference model- the ISO model of architecture for Open System Interconnection," *IEEE Trans. on Commun.*, vol. COM-28, pp. 425-432, Apr. 1980.
[41] J. Hui, E. Arthurs, "A broadband packet switch for integrated transport," *IEEE J. on Sel. Areas in Commun.*, vol. SAC-5, pp. 1264-1272, Oct. 1987.
[42] H. Ichikawa, *et al.*, "High-speed packet switching systems for multimedia communications," *IEEE J. on Sel. Areas in Commun.*, vol. SAC-5, pp. 1336-1345, Oct. 1987.
[43] S. Nojima, *et al.*, "Integrated services packet network using bus matrix switch," *IEEE J. on Sel. Areas in Commun.*, vol. SAC-5, pp. 1284-1291, Oct. 1987.
[44] Y. Yeh, *et al.*, "The Knockout switch: a simple, modular architecture for high-performance packet switching," *IEEE J. on Sel. Areas in Commun.*, vol. SAC-5, pp. 1274-1283, Oct. 1987.
[45] C. Hoppitt, "ISDN evolution: from copper to fiber in easy stages," *IEEE Commun. Mag.*, vol. 24, pp. 17-22, Nov. 1986.
[46] B. Schaffer, "Switching in the broad-band ISDN," *IEEE J. on Sel. Areas in Commun.*, vol. SAC-4, pp. 536-541, July 1986.
[47] G. Coviello, "Comparative discussion of circuit vs packet switched voice," *IEEE Trans. on Commun.*, vol. COM-27, no. 8, pp. 1153-1159, 1979.
[48] P. Kermani, L. Kleinrock, "A trade-off study of switching systems," *ICC '79*, pp. 20.4.1-20.4.8.
[49] K. Kummerle, H. Rudin, "Packet and circuit switching: cost and performance boundaries," *Computer Networks*, vol. 2, pp. 3-17, 1978.
[50] P. O'Reilly, "Burst and fast-packet switching: performance comparisons," *Infocom '86*, pp. 653-666.
[51] H. Rudin, "Studies on the integration of circuit and packet switching," *ICC '78*, pp. 20.2.1-20.2.7.
[52] N. Keyes, M. Gerla, "Report on experience in developing a hybrid packet and circuit switched network," *ICC '78*, pp. 20.6.1-20.6.6.

Thomas M. Chen (S '81) received his B.S. and M.S. degrees in 1984 from Massachusetts Institute of Technology, Cambridge.

He has worked on image compression at the IBM San Jose Research Laboratory and on ISDN at Pacific Bell. He is currently pursuing his Ph.D. in electrical engineering at the University of California, Berkeley.

David G. Messerschmitt (IEEE Fellow) received a B.S. degree from the University of Colorado in 1967, and an M.S. and Ph.D. from the University of Michigan in 1968 and 1971, respectively. From 1968 to 1977 he was a member of the technical staff and later, Supervisor at Bell Laboratories, Holmdel N.J., where he did systems engineering, development, and research on digital transmission and digital signal processing (particularly relating to speech processing). He is currently a Professor of Electrical Engineering and Computer Sciences at the University of California at Berkeley. Current research interests include applications of digital signal processing, adaptive filtering, digital communications (on the subscriber loop and fiber optics), architecture and software approaches to programmable and dedicated hardware digital signal processing, communication network design and protocols, and computer-aided design of communications and signal processing systems. He has published over 100 papers and has ten patents issued or pending in these fields. Since 1977, he has also served as a consultant to a number of companies.

Dr. Messerschmitt is a member of Eta Kappa Nu, Tau Beta Pi, Sigma Xi, and has several best paper awards. He has served as a Senior Editor of the *Communications Magazine* and as Editor for Transmission of the *Transactions on Communications*. Also, he was a member of the Board of Governors of the Communications Society. He has organized and participated in a number of short courses and seminars devoted to continuing engineering education.