A Packet Network Architecture for Integrated Services

Jonathan S. Turner
Leonard F. Wyatt

Bell Laboratories
Naperville, Illinois 60566

Abstract

The Integrated Services Digital Network (ISDN) is a concept of growing popularity that offers the prospect of new voice and data services at a modest cost. Most proposals for ISDN services involve an integrated access method (e.g. the Digital Subscriber Line) and either separate networks for voice and data or hybrid switching systems of various sorts. This paper discusses the concept of using a single internal packet network to support ISDN services, exploring the protocols and performance of the packet network as well as some aspects of its implementation.

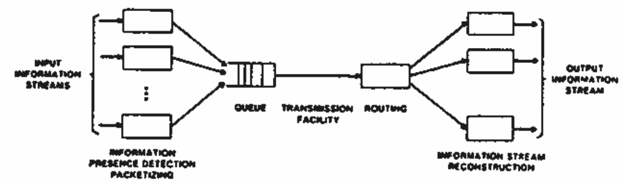
I. Introduction

The Integrated Services Digital Network (ISDN) is a concept of growing popularity that offers the prospect of new voice and data services at a modest cost [1]. Most proposals for ISDN services involve an integrated access method (e.g. the Digital Subscriber Line) and either separate networks for voice and data or hybrid switching systems of various sorts. This paper discuss the use of packet network to support ISDN services. The use of a packet network makes possible a variety of services and achieves high efficiency due to statistical multiplexing. In this paper we will examine this concept and examine some of the problems that must be solved to make it a reality.

Some of the factors that make packet switching an attractive method for offering integrated voice and data services are listed below.

- *Transmission efficiency*. Many data services are characterized by bursty traffic which makes poor use of conventional circuit switched facilities. For example, interactive data users typically use only a few percent of the bandwidth available to them. Although it is less widely recognized, voice is also quite bursty. In the average telephone conversation, less than 40% of the available bandwidth is actually used. Packet switching can exploit this burstiness to allow many users to share the same transmission facility. Packet switching also makes it attractive to consider new coding techniques for information transmission. Most information coding techniques in use today attempt to maintain a smooth flow of information across the transmission facility in order to use it efficiently. With packet switching one can exploit coding techniques which produce bursty information streams.
- *Adaptability to changing traffic*. Packet switching naturally provides each user with exactly the bandwidth required. As new services are developed with different bandwidth requirements, packet switching can adapt to the changing conditions much more easily than circuit switching can.
- *Integrated internal architecture*. Most proposals for ISDN services provide integrated access but require separate switching networks for different types of information. Packet switching can provide both an integrated customer interface and a single network solution for a wide range of different information transport needs, including voice, data and signalling. Substantial cost savings could be possible both in switching and terminal equipment.

The transmission efficiency of packet switching is due to *statistical multiplexing*. In a conventional circuit switched network, each customer is provided with a communications channel that is dedicated to that customer for the duration of the call. If the customer does not make use of the available bandwidth, it is lost; there is no opportunity for making that bandwidth available to another customer. Statistical multiplexing provides a way for customers to share transmission facilities on a demand basis. This is illustrated in Figure 1. Customer information enters at the left, where it is



Statistical Multiplexing
Figure 1.

broken into blocks with a header added. This yields a *packet*. The header contains information needed to route the packet to its destination and reconstruct the original data stream. Packets from different customers are then placed in a queue and sent across a transmission facility. Upon receipt, packets are sent to the proper destinations and the original data streams are reconstructed. As described above, statistical multiplexing is purely a transmission method and as such can be viewed as a generalization of Digital Speech Interpolation. Packet switching extends this idea to include the switching function, which leads to further cost savings.

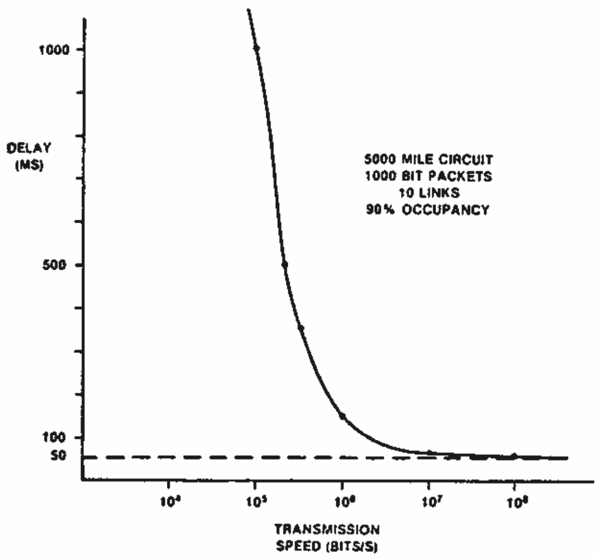
In conventional circuit switched networks, the principal source of delay is the finite propagation speed of signals. This delay is very short for local calls but may be as much as 50 ms for a cross-country call, 100 ms for a terrestrial call between the U.S. and Europe or 250 ms for a call placed over a satellite. In addition, conventional switching and multiplexing equipment introduces a small fixed delay; this rarely exceeds a few milliseconds on an end-to-end basis. Packet switching introduces a variable delay due to the queueing at each outgoing transmission link. The average delay can be approximated by

$$q = \frac{bn}{s(1-\rho)} \tag{1}$$

where b is the packet length in bits, n is the number of tandem links, s is the speed of the transmission facility in bits per second, ρ is the average utilization of the facility and q is the queueing delay in seconds. We can decrease the delay by decreasing the packet length, decreasing the number of tandem links, increasing the speed of the transmission facility or decreasing the occupancy of the transmission facility. Decreasing the packet length is helpful up to a point, but can lead to inefficient use of the transmission facility since as packets become shorter, a larger percentage of the bandwidth is used to transmit the header information rather than customer data. Decreasing the link occupancy sacrifices the efficiency of statistical multiplexing for speed. The number of tandem links is a function of the size of the network and the way that the switches are interconnected. As an example of this, the number of toll trunks in a single connection within the Bell System telephone network is currently limited to seven and most long distance connections within the U.S. use five or less. While one could probably design a network with as few as four tandem links in the worst case, it would be difficult to do better than this. The remaining variable in the equation is transmission speed, which can be varied over a wide range. Most packet switched networks today use facilities with speeds of 10 to 56 Kbs. Digital transmission facilities in telephone networks run at speeds of from 1 to 10 Mbs and the newer optical transmission facilities will extend that range to over 100 Mbs. Figure 2 shows how the total delay

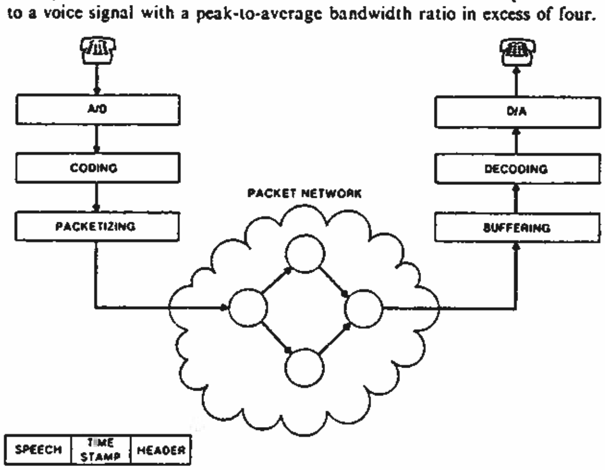
83CH1956-2/83/0000-0045 $1.00 © 1983 IEEE

varies with transmission speed for a ten link connection spanning a distance of 5000 miles. The minimum delay shown in the figure is due to fixed transmission delays that are present even in circuit networks. Although higher transmission speeds can reduce queueing delay to acceptable levels, their exploitation will require a new generation of packet switching equipment.



Transmission Speed vs Delay
Figure 2.

The improvements in transmission efficiency obtainable with packet switching are dependent on the characteristics of the particular application. Packet switching is best suited to applications that are bursty, that is, applications that have a peak bandwidth requirement that is substantially higher than their average bandwidth requirement. While voice is not usually thought of as a bursty data source, closer examination shows that it is quite bursty. Each party in a typical telephone conversation speaks less than 40% of the time, on the average. Certain speech coding algorithms require varying amounts of bandwidth to represent different kinds of sounds. In *linear predictive coding* for example, unvoiced sounds can be adequately represented with about half as many bits as voiced sounds. In addition, various statistical coding algorithms can be used to reduce the average bandwidth even further. A combination of such techniques can lead to a voice signal with a peak-to-average bandwidth ratio in excess of four.



Basic Processes of Packet Voice
Figure 3.

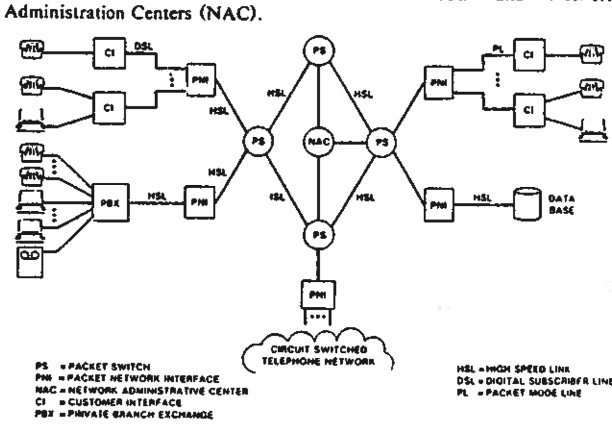
Figure 3 illustrates some of the basic processes in packet voice. The voice signal is first converted to digital form using an A/D converter. It is then coded using an appropriate coding algorithm. Some of the possibilities are linear predictive coding, adaptive differential pulse code modulation, and sub-band coding. The different algorithms vary in cost, quality and bandwidth required. After the voice signal is coded, it is broken into blocks, a header is added, and it is sent through the network. At the destination, the coding process is reversed, the signal is converted to analog and played back. Because the network introduces a variable delay, different packets within the same call (or even the same word) may experience different delays in crossing the network. Consequently, care must be taken to ensure that this effect does not degrade the quality of the reproduced voice signal. This can be ensured by appending a time stamp to each packet at the source and using that time stamp to preserve the cadence of the reproduced speech signal. There are several effective ways that this time stamping can be done, that are discussed in detail in a companion paper [2].

We have reviewed some of the motivation for constructing an integrated voice and data network using packet switching. In the remainder of this paper we will address some of the problems that must be solved before such a network can be built. We also describe a possible internal packet network architecture having the following characteristics.

- *High speed digital transmission.* Modern digital transmission facilities offer higher speed and better error performance than older transmission technologies. The simplest way to solve the delay problem is to increase the basic transmission speed to over 1 Mbs. For example, in the U.S., the preferred speed might be 1.5 Mbs initially, in Europe it would probably be 2 Mbs. As technology evolves, these speeds can be increased further.
- *Simple internal network protocols.* The use of high speed digital transmission facilities creates opportunities for greatly simplified packet protocols within the network. Simple protocols may also be needed to ensure that fast and inexpensive protocol processors can be built.
- *Hardware switching.* Conventional packet switches rely on general purpose computers or collections of microprocessors. Novel packet switch architectures based on custom VLSI (Very Large Scale Integrated Circuit) components will likely be needed to attain high speed operation and economic implementation.
- *Separation of services from transport.* Future networks will be called upon to provide a tremendous variety of services that can be expected to change often. The internal network should provide basic packet transport services that can serve as a base for these services. The network architecture presented here accommodates different service needs by equipment at the access points to the internal packet network, and by special service nodes attached to but separate from this network.

2. Network Architecture Overview

The basic components of the proposed network are shown in Figure 4. They are Packet Switches (PS), Packet Network Interfaces (PNI), High Speed Links (HSL), Customer Interfaces (CI) and Network Administration Centers (NAC).



Network Architecture
Figure 4.

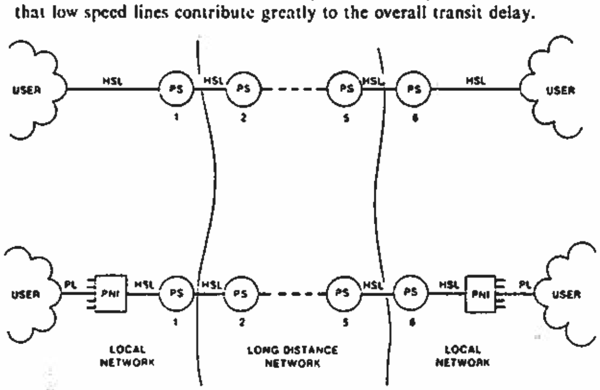
There may be many different kinds of PNIs depending on the nature of the access method and services. Customers may communicate with the PNI using Digital Subscriber Lines (DSL) having a 64 Kbs voice channel and a separate packet mode signalling channel, as proposed for ISDN access [1]. In this case, the PNI would provide voice coding and packetizing, but the far end customer would be given a voice signal as reconstructed by the far end PNI. Signalling would be handled through messages exchanged with

2.1.2

the CI. Another possibility is a pure Packet Line (PL). In this case the voice coding and packetizing would be provided in the CI and the PNI would provide statistical multiplexing and network protection functions. A third possibility is a high speed packet mode line using a protocol similar to the internal network protocol. This would be used primarily for business customers initially, but might ultimately be extended to residential use for wide band services. Finally, one would expect to have interfaces to existing circuit switched telephone networks.

A *reference connection* can be used to study the performance of a network such as this. A worst-case path through the network is chosen as a reference path, and this is used to make performance estimates. Two reference connections (Figure 5) are used here to illustrate how the performance of the network depends on the network size and link speeds. In each model the length is chosen to be 4300 miles, to allow for a coast-to-coast connection in the United States. It is assumed that the speed of the HSLs is 1.5 Mbs.

- *Connection between two high-speed-link customers.* Six packet switches (seven links) are allowed in this connection. This is one less switch than in the longest connection in the present Bell System network, which has become a rare connection due to the connectivity of the existing network. Such a configuration might be used for a connection between two PBX's, or as a long-haul network between gateways. This model yields the best performance results. High speed links are a major contributor to good delay performance.
- *Connection between two low-speed-link customers.* A concentrator and low-speed packet-mode line are added to each end of the above connection. The line is assumed to operate at 64kbps. Here we will see that low speed lines contribute greatly to the overall transit delay.



Reference Connections
Figure 5.

3. Network Performance

Use of the packet network to carry voice traffic places some delay requirements on network performance which are more stringent than before. The quality of a conversation can be significantly degraded if there is too much delay in the voice signals. Exactly how much is allowable is subject to some debate, and estimates from 200 ms to 600 ms have been forwarded [3]. In the analysis that follows we will assume a maximum delay of 250 ms, as in satellite connections. This is a fairly conservative choice but as we shall see, it appears to be attainable. While voice traffic requires good delay performance, it is much *less* demanding in terms of error performance. It has been estimated that one to five percent of the packets in a voice conversation may be lost without significantly impairing the conversation [4]. This depends in part on the voice coding scheme used. We will assume that the total packet loss rate for voice packets from all possible causes is at most one percent.

There is an important trade-off between the network delay performance and the lost voice packet rate. This will cause some packets to be effectively lost even though they reach their destinations intact. When a packet reaches its destination, it is delayed for some amount of time before being played out. Although the packets were generated at regular intervals, their delay times through the network are variable. The playback delay allows playback to occur at an even pace, even though packets arrive at irregular intervals. This delay cannot be arbitrarily large, however. The network delay plus the playback delay must not exceed 250 ms. A packet with too much network delay has missed the appropriate playback time, and is effectively lost. The voice synthesizer will have filled that time slot with something else, quite possibly silence. Thus packets lost due to excessive delay are included in the lost packet rate for voice. Given the assumptions made above for delay and lost packet rate, we require that at least 99 percent of all voice packets arrive no more than $250ms$ after they were generated.

For data packets, a higher degree of accuracy is desired. Since digital transmission facilities operate with a bit error rate of about 10^{-6}, roughly one packet in 1000 will be lost due to transmission errors (assuming 1000 bit packets). Queues can be engineered to produce roughly the same packet loss rate under worst case load conditions. Lost or corrupted data packets can be recovered by retransmission on an end-to-end basis in our network architecture. Link-by-link retransmission has also been used in other packet networks.

In the remainder of this section we study the network performance in more detail. Section 3.1 focuses on the network delay, while Section 3.2 deals with error performance.

3.1 Delay Performance

In analyzing the delay performance of the packet network, we separate the delays in the packet switch into two nearly independent components. First is the switching delay, which is the amount of time used to route a packet to its destination link, including header processing and error checking. Second is the queueing delay, which is the amount of time the packet spends in an output queue awaiting transmission on the next link. Several sources of delay compose the overall network delay for voice packets. Most of the delays are of fixed duration, with queueing delays being the primary variable delay. The fixed delay components are the coding and packetizing delay (p), the cross-network link transmission delay (t), and the switching delay for all switches in the connection (s). The switching delay is not strictly a constant, but it is small enough in relation to the other variables that we can regard it as such. The queueing time for all queues in the connection will be referred to as q, which is a function of the link occupancy and speed. The end-to-end delay is the sum of all these:

$$d = p + t + s + q$$

We are particularly interested in the 99th percentile delay for voice packets (the time by which 99 percent of the packets will have arrived), and would like this to be less than 250 ms. Queueing delay is the variable in the equation, so:

$$d_{99} = p + t + s + q_{99}$$

The time required for voice coding and packetizing varies according to the technique used. Let us assume that with an LPC coder, 60 ms of speech are represented in each packet. The coding process lags the speaker by 20 ms. Add 10 ms for packetizing, depacketizing at the destination, and playback. Then the coding and packetizing time is 90 ms total. As another example, an ADPCM coder might represent 16 ms of speech in a packet. Negligible delay is encountered at the coder, but another 16 ms is allowed at the destination for interpolation [4]. Thus only 32 ms would be allocated for coding by this scheme, although the resulting longer packets would increase delays in the remainder of the network. Since LPC presents the larger delay, we set $p = 90$ ms.

Transmission delay is the amount of time required to actually move the bits across the links. For long-haul facilities, $10\mu s$ per mile is typical, and one millisecond is allocated for each customer line. Using the 4300 mile reference connection, $t = 45$ ms.

As discussed in Section 6, hardware switching architectures exist which should be able to switch packets in one millisecond or less. This switching time does not include the time spent in the output queue awaiting transmission over a link. By comparison to the other delays, 1 ms is small enough to be regarded as a constant, even though it is not constant in most switching architectures. There are six switches in the reference connection, resulting in $s = 6$ ms. Allowing 1 ms for each multiplexor when low-speed access in involved gives $s = 8$ ms. This should be a safe assumption, given that the multiplexors are made using advanced techniques similar to the switches.

We use an M/M/1 queueing model [5] to approximate the delay due to queueing (q). The average delay through n tandem queues, each with an independent mix of traffic, is

$$q = \frac{bn}{s(1-\rho)}$$

where b is the average packet length in bits, s is the bit rate of the transmission lines (1.5Mbps), and ρ is the utilization of the links. We set $\rho = 0.9$ for all links as a worst-case assumption. In our reference model,

$n = 7$ high speed links.

The average packet length depends on the mix of traffic in the network. If LPC coders are used, packet lengths of 300 bits or less are likely. In a network combining both voice and data traffic, we expect that many more voice packets than data packets will be present. Since data packets may be longer, we wish to account for the loading they place on the links. As a very conservative assumption, we assume that 20% of the traffic is 1000-bit data packets. Mixed with 80% 300-bit packets, this results in an average packet length of 440 bits. Now the average queueing delay can be computed for the seven high-speed links, and 20.5 ms is the result.

The packet size clearly influences the queueing delay, and is a function of the coders used. For coders with a higher bit rate, the packet size and thus the queueing delay increase. For example, an ADPCM coder would likely generate packets on the order of 600 bits. Using the same traffic mix as above, queueing of 31.7 ms is estimated for the high speed links.

Low speed access links can add a large element to the queueing delay in the packet network. By the same method used above, we compute a delay of 137 ms for two 64 kbps lines at $\rho = 0.9$ with 440-bit average packets. This is too much to fit with the other variables into a 250 ms total delay, and we conclude that the customer line cannot be loaded at this level. Using $\rho = 0.5$, we can compute a 27.5 ms delay for the lines, which is more acceptable.

As described above, the figure of interest for packet voice traffic is the 99th percentile delay. This can be estimated using a tandem queueing model. When n identical M/M/1 queues are arranged in tandem, an Erlang distribution of rank n (E_n) gives the delay distribution. For the reference connection between two high speed link customers we use an E_7 distribution for which the 99th percentile delay is 2.1 times the mean delay, giving $q_{99} = 43$ms. For the low speed access connection we add two 64 Kbs links. The delay over these is given by an E_2 distribution. To get the total delay one must convolve the E_2 and E_7 distributions. Solving numerically we find that $q_{99} = 81\ ms$ for this connection.

The table below summarizes the results from these calculations, including the case of ADPCM coding as well as LPC.

	Voice Delays (99%) LPC	Voice Delays (99%) ADPCM	Data Delays (Avg)
Coding/Packetizing - p	90	32	-
Transmission - t	45	45	45
Switching - s	6	6	6
Queueing - q	43	67	21
Network Delay - d	184 ms	150 ms	72 ms

Although the use of a higher bit rate coder with lower coding delay may decrease the voice delay somewhat, it is important to remember that this is at the expense of call capacity in the network. If each call uses more of the bandwidth on a link, fewer calls can be carried on the link. Adding the low speed access lines adds considerably to the delays:

	Voice Delays (99%) LPC	Voice Delays (99%) ADPCM	Data Delays (Avg)
Coding/Packetizing - p	90	32	-
Transmission - t	45	45	45
Switching - s	8	8	8
Queueing - q	81	171	48
Network Delay - d	224 ms	256 ms	101 ms

3.2 Error Performance

For voice packets, there are three factors contributing to packet losses. Data packets are only subject to the first two types of losses.

- *Packets lost due to transmission errors.* 98% of existing Bell System T1 lines operate with a bit error rate (BER) of less than one in 10^6 [6,7]. If we assume that a new packet switching network would use facilities with error performance at least that good, we can estimate the error performance of the network. An important characteristic of transmission facilities is that errors tend to happen in bursts, where a series of bits is corrupted together. Each bit corrupted contributes to the bit error rate, but not necessarily to the packet error rate. A burst which happens entirely within one packet corrupts only that packet. Nevertheless, we will use the BER to compute a packet error rate (PER), with the understanding that the estimate is conservative.

$$PER = 1 - (1 - BER)^b \approx b \times BER = 1000 \times 10^{-6} = 10^{-3}$$

Again b is the packet length, only here we set it to 1000 bits to obtain the worst error performance.

- *Packets lost due to queue overflows.* Packets which arrive at a queue which is already filled to capacity will be lost. Here we can control the loss rate by engineering the queue length L to an appropriate size. The loss rate also depends on the traffic intensity, for which we assume $I = 0.9$. (Traffic intensity is a measure of traffic arriving at the queue, which may be different from the utilization of the server in the case of a finite length queue.) The probability of loss is given below [5].

$$P[loss] = \frac{1-I}{1-I^{L+2}} \times I^{L+1}$$

There is no reason to reduce queueing losses to extremely small numbers, since the transmission losses would then dominate. Therefore we will set the queue size to result in a loss rate of less than 10^{-3} for all the links combined. If a link is designed with a queue length of 64 packet slots, then $P[loss] \approx 1.06\times10^{-4}$. For all seven high speed links at $I = 0.9$, the cumulative loss rate across the reference connection is:

$$1 - (1 - P[loss])^7 \approx 7 \times P[loss] \approx 7.4\times10^{-4}$$

If the traffic intensity were reduced to 0.8, then a queue length of 32 slots would suffice.

In the reference model with low-speed access links operating at $I = 0.5$, a queue length of 10 slots gives a loss rate of 2.4×10^{-4}.

- *Packets lost due to excessive delay.* As described earlier, packets arriving more than 250 ms after generation cannot be played back. In the previous section, the delay analysis showed that 99 percent of the packets will arrive well within 250 ms when only high speed links are used. When low speed access links are included, 99 percent of the packets arrive within 244 ms. We then expect that slightly less than one percent of all packets will be lost due to excessive delay.

For data packets, we then expect that about two packets out of 1000 will be "lost" in the internal network. These losses can be detected by the network protocol and recovered by retransmitting the packets. About one voice packet in 100 will be lost. These will not be recovered; rather, the voice synthesizer at the destination will be forced to fill the time with something. Repetition of the last packet, interpolation and silence fill are all possibilities.

4. Protocol

The design of a communications protocol must take many different factors into account. Perhaps the most important are the nature of the applications that will use the protocol and the design of the network equipment that must implement it. Some of the characteristics of the applications that will affect protocol design are listed below.

- *Delay.* Voice imposes strict requirements on end-to-end delay; voice packets must be received within the maximum allowable network delay (assumed to be 250 ms) if they are to be useful. As discussed in section 3, most of this allocation is used up in coding, propagation delay and access delay, leaving about 100 ms for the network. This leaves no time for retransmission of packets that are lost. Most data applications, on the other hand, can tolerate some delay. They are also less sensitive to the variation in the delay than voice.

- *Errors.* Voice is fairly tolerant of errors in the signal. A few bits in error, or even an occasional lost packet, will not seriously degrade the quality of the voice signal. This is in contrast to most data applications, which require error free delivery.

- *Flow Control.* End-to-end flow control is not needed for voice, since both the sender and receiver operate at the same speed. In data applications, the sender and receiver may operate at very different speeds, making flow control essential.

- *Traffic Characteristics.* The statistical properties of voice signals have been thoroughly studied and are quite predictable. Data is less well-behaved; there are many different applications with very different characteristics and some applications can display very different traffic characteristics within the course of a single call. For example, the percentage of time that a speaker is talking during a voice call (the *activity*) is about 40%. For data, the activity may range from less than 1% to over 80%. A typical voice call would require the transmission of a few thousand packets, while a data call may involve just a few packets, or as many as ten thousand.

The design of network equipment can affect the protocol in two ways. It can provide opportunities that aren't available with other network designs and it can constrain the possibilities for the protocol designer. Modern digital transmission facilities have two very attractive characteristics. They are fast and they have very low error rates. This makes it possible to simplify protocols to a surprising degree. Many functions that are performed on each transmission link with conventional communications protocols can be performed on an edge-to-edge basis if the network is fast enough and reliable enough to permit it. This has several benefits. First, it permits the network to use a simplified protocol, which improves performance and eases implementation. Since the network must operate at high speeds, it is important to avoid excessive complexity. Second, it permits the use of different edge-to-edge protocols that are tailored to specific applications. For example, voice does not require re-transmission of lost packets (they would be too late to be useful anyway), while many data applications do. By providing these functions on an edge-to-edge basis, they can be supplied only where they are needed, resulting in a reduction of the resources (transmission bandwidth and processing) expended on transporting packet voice. Finally, changes in technology are making it easier and economically attractive to provide these functions at the access points to the internal network rather than providing them throughout the internal network. VLSI technology is better suited to the production of many copies of complex but moderate performance devices to be used at the packet network interface to provide service to individual customers than it is to the production of a few very high performance devices that could be shared by a large number of customers. These trends parallel what is happening in computing, with the increasing availability and popularity of small dedicated machines over large shared systems.

The above considerations lead to an internal network protocol having the following characteristics.

- *Simplified link layer.* The link layer protocol provides for transmission of information frames across a data link and includes frame delimiting, bit pattern transparency and error detection, using mechanisms similar to HDLC [8]. The link layer does not include error recovery and flow control procedures. This eliminates the need for state information in the protocol processors and allows link initialization procedures to be dropped as well.
- *Virtual call routing.* Virtual calls are allocated a path that is used by all packets in the call. Each call on a particular trunk is assigned a logical channel number, which is part of the packet network sub-layer of the protocol. Virtual call routing greatly facilitates the design of special purpose switching hardware to route the individual packets of a call and greatly simplifies overload control.
- *Multiple packet transport sub-layers.* The packet transport sub-layer is performed between the ends of the internal network, and different protocols are provided to support different services. For example, the packet transport sub-layer for voice would include some form of time stamp to permit proper timing of the playout of speech, while the transport sub-layer for data would include retransmission to correct errors.

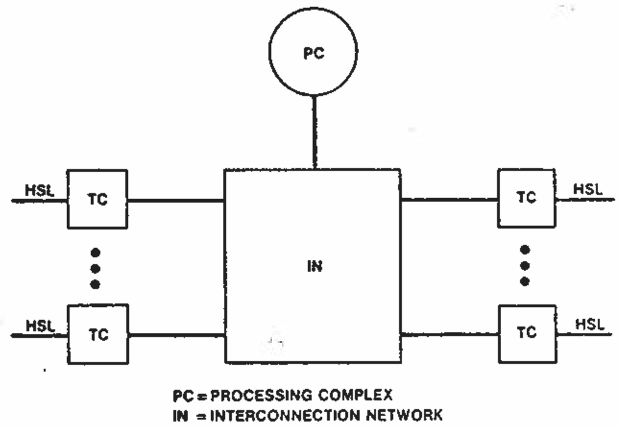
A more detailed description of the internal network protocol is given in a companion paper [9].

5. Packet Switch Architecture

In this section, we outline some of the basic objectives for the switching components of the internal packet network and describe a high level architecture. First, let's briefly review some of the basic parameters of typical circuit and packet switching systems today. A large telephone switching system can terminate up to 100,000 trunks, permitting it to handle close to 50,000 simultaneous voice conversations. This corresponds to a throughput of over 6 Gbs. The delay through a circuit switch is at most a few milliseconds and the cost of such a system is typically between one and two hundred dollars per trunk. Typical packet switches present a very different picture. They typically can handle about 100 trunks (at 56 Kbs or less), have a throughput of less than 5 Mbs, incur delays of twenty milliseconds or more and cost a few thousand dollars per trunk. From this very crude comparison it is clear that if packet switching is to be competitive with circuit switching in an application like voice, new approaches are needed. Indeed, we require an improvement of an order of magnitude in both delay and cost, and two or three orders of magnitude in throughput.

The number of voice conversations that can be carried on a 1.5 Mbs HSL is a function of the coding rate. For 32 Kbs ADPCM this will be between and 100 and 120 conversations, for LPC it could be as high as 500. A packet switch that can handle 50,000 voice conversations will then have to terminate between 200 and 1000 HSLs. It will need a throughput between .3 and 1.5 Gbs or from .7 to 2.5 million packets per second.

The natural way to achieve the high performance switch implied by these numbers is to make extensive use of parallelism and distribute functions rather than centralize them. Packet switching is ideally suited to such parallel processing, because the control information needed to process a packet can be included as part of the packet header, and because the individual processing operations are simple and must be performed on many different streams of data. In addition, such a parallel architecture can best exploit trends in electronics technology that favor systems that can be built up of many copies of a small number of components.



Packet Switch Block Diagram
Figure 6.

Figure 6 is a very high level block diagram of a packet switch for an internal packet network. The Link Interfaces (LI) perform the lower levels of the internal protocol discussed in Section 4. Essentially, any processing that must be done for every packet is done by the LI. The functions performed include framing, error detection and routing (logical channel translation). In addition, the LIs route special packets (eg. call setup packets or datagrams) to the Processing Complex. The Interconnection Network (IN) is a switching fabric that must be able to accept packets containing some routing information in their headers and deliver the packets to the specified destination. The possibilities include a set of interconnected rings, busses, and various multi-stage networks such as the buffered delta network, various kinds of shuffle networks and the multi-stage cube network [10,11,12]. The Processing Complex (PC) consists of one or more processors that handle all call processing, maintenance and system administration. Essentially, it performs the functions handled by the processor in a telephone switch. Unlike many packet switches, however, it does no per-packet processing.

Many specific packet switch designs can fit within the broad structure described here. The important point about this structure is that the simple but highly repetitive task of processing packets associated with individual calls is performed by the LIs and the IN which rely on parallelism to achieve the required performance, while the complex but less frequently performed tasks of call processing and maintenance are performed by a central Processing Complex.

Some of the architectural concepts suggested here have been used in the Voice Funnel project [13], a packet voice multiplexor used in a satellite communications system.

6. Separation of Transport and Services

One of the most attractive features of a high speed packet network is that it makes it very easy to provide services without embedding them within the internal portions of the network. The speed and economy of message communications in such networks makes them well-suited to services that are implemented in the Packet Network Interfaces, the terminals or special service nodes that are connected to the internal network. Some of the factors that make this separation attractive are listed below.

* *Rapid introduction of new services.* Services that are embedded in the internals of a communications network can be difficult to change because they can affect all customers, often in unexpected ways. This creates obstacles that must be overcome when introducing new services or changing existing services. Separating the services from the internal network reduces the interactions among different services and between services and the internal network, eliminating some of the obstacles. This can reduce development costs which are becoming a major portion of the cost of communications services, and more important can reduce development time which is often critical from a marketing standpoint.
* *Extend internal network longevity.* Communications networks suffer from a fundamental 'chicken and egg' problem. Customers are needed to pay for the investment in network equipment, but the network will be attractive to customers only if it interconnects many customers. This is an intrinsic characteristic that makes it difficult to introduce new communications networks and to change existing ones. When services are separated from the internal transport mechanisms of the network, services can be changed rapidly without changing the internal network. An internal network that provides basic transport capabilities is thus not subject to the rapid obsolescence that can plague networks that attempt to do everything.

The following list contains several examples of communications services and outlines how they could be provided by the network interfaces and special service nodes. It also indicates some of the capabilities the network would have to provide in order to make these services possible.

* *Call forwarding.* This is a service that is typically performed by the network equipment in circuit switched networks, but that can be effectively implemented at the edges of the internal packet network used to transport the call. If a customer wants a call forwarded to another location, a reject message is sent to the internal network in response to a call setup request, but now the reject message would also contain a field that would inform the caller where the call should be re-directed to. All that is required of the internal network is that it permit a call reject message to have an embedded data field, and deliver the data to the source of the call.
* *Conferencing.* Multi-way conferencing requires a conference bridge which receives information streams from all the participants and either chooses one stream or combines the streams in some way and then broadcasts the resulting stream to the participants. For small conferences, this function could be provided by the network interface of one of the participants, using several virtual circuits in the internal network to gather voice packets and distribute then to the other parties. For larger conferences, special service nodes attached to the internal network can provide the bridge. For very large conferences, it might be desirable to use a multi-level hierarchy of special service nodes to reduce the communications costs. Broadcast could be handled as a special case of this structure. Note that the internal network itself need not know about the conference structure.
* *Private networks.* A popular feature for computer networks is the virtual private network or closed user group. This feature can be provided by having the internal network supply the identity of the calling party in each call setup packet. The screening function can be done at the edges of the internal network, or by a a special service node which screens all calls to a particular address and then uses the call forwarding mechanism to re-direct the call.

One can easily construct other examples of services that have traditionally been implemented within the internal network that in future may be more effectively handled at its edges. The key here is that the internal network must be capable of fast and inexpensive message transmission. This, in combination with intelligent service nodes and network interfaces offers new and powerful ways of implementing sophisticated communications services.

7. Summary

We have described a packet network architecture for supporting integrated services. The important characteristics of this architecture are:

* *High speed digital transmission.* This provides the necessary delay and error performance while making efficient use of network resources.
* *Simple internal network protocols.* The new protocols exploit the high speed and good error performance of modern digital transmission facilities and simplify the design of high performance packet switches.
* *Hardware switching.* Special purpose switching hardware provides high performance and exploits the characteristics of VLSI for economical implementation.
* *Separation of services from transport.* Separation of services from transport insulates the internal network from changes in services, making it easier to add new services and avoiding rapid obsolescence of the internal network.
* *Usage-based charging.* In architectures where services are embedded within the internal network, it is often difficult to charge customers for the services that they use, because of the large overhead costs that must be borne by all. If services are controlled from the edges, there is less embedded overhead and it becomes easier to base charges on the resources acutally used.

Of course, these advantages don't come for free. There is somewhat less opportunity for equipment sharing which can lead to possibly higher costs; however, the rapidly falling cost of integrated circuits makes this less of an issue than it would otherwise be. In fact, the cost of providing shared access to a common resource and then protecting that resource against failures can exceed the cost of replication. In addition, rising development costs favor replication. The more important issue is that access protocols to the internal network must be carefully designed to permit effective implementation of services outside of the internal network. The signaling protocol is particularly critical. It must provide a set of primitive capabilities that can be combined to offer a variety of services, and it must be extensible so that as new needs arise, they can be provided for.

References

[1] Decina, Maurizio. "Progress Towards User Access Arrangements in Integrated Services Digital Networks". In *IEEE Transactions on Communications 30*, 2117-2130, 9/82.

[2] Montgomery, Warren A. "Techniques for Packet Voice Synchronization". In *IEEE Transactions on Communications*, this issue.

[3] Gruber, J. G. "Performance Considerations for Integrated Voice and Data Networks". In *Computer Communications*, 6/81.

[4] Jayant, N. S. "Effect of Packet Losses in Waveform Coded Speech". In *International Conference on Computer Communications*, 10/80.

[5] Kleinrock, Leonard. *Queueing Systems*. John Wiley & Sons, Inc., 1975.

[6] AT&T "Impairments in Digital Networks", CCITT-COMXVII-No. 33, 3/81.

[7] Ritchie, G. R., P. E. Scheffler. "Projecting the Error Performance of the Bell System Digital Network". In *IEEE International Conference on Communications*, 2D.2.1-2D.2.6, 6/82.

[8] Tanenbaum, Andrew S. *Computer Networks*. Prentice-Hall, 1981.

[9] Hoberecht, William L. "A Layered Transport Protocol for Packet Voice and Data Integration". In *IEEE Transactions on Communications*, 12/83.

[10] Dias, Daniel M., J. Robert Jump. "Packet Switching Interconnection Networks for Modular Systems". In *Computer 14*, 43-54, 12/81.

[11] Chen, P-Y., D. H. Lawrie, D. A. Padua, P-C. Yew. "Interconnection Networks Using Shuffles". In *Computer 14*, 55-64, 12/81.

[12] Siegel, Howard Jay, Robert J. McMillen. "The Multistage Cube: A Versatile Interconnection Network". In *Computer 14*, 65-76, 12/81.

[13] Rettberg, R., C. Wyman, D. Hunt, M. Hoffman, P. Carvey, B. Hyde, W. Clark, M. Kraley. "Development of a Voice Funnel System: Design Report", Bolt Beranek and Newman, Report No. 4098, 8/79.