UNIT-I

MULTIPLEXING

Transmission Systems

Functionally, the communications channels between switching systems are referred to as trunks. In the past, these channels were implemented with a variety of facilities, including pairs of wires, coaxial cable, and point-to-point microwave radio links. Except for special situations, trunk facilities now utilize optical fibers.

Open Wire

A classical picture of the telephone network in the past consisted of telephone poles with crossarms and glass insulators used to support uninsulated open-wire pairs. Except in rural environments, the open wire has been replaced with multipair cable systems or fiber. The main advantage of an open-wire pair is its relatively low attenuation (a few hundredths of a decibel per mile at voice frequencies). Hence, open wire is particularly useful for long, rural customer loops. The main disadvantages are having to separate the wires with cross arms to prevent shorting and the need for large amounts of copper. (A single open-wire strand has a diameter that is five times the diameter of a typical strand in a multipair cable. Thus open wire uses roughly 25 times as much copper as does cable.) As a result of copper costs and the emergence of low electronics costs, open wire in rural environments has been mostly replaced with cable systems using (digital) amplifiers to offset attenuation on long loops.

Paired Cable

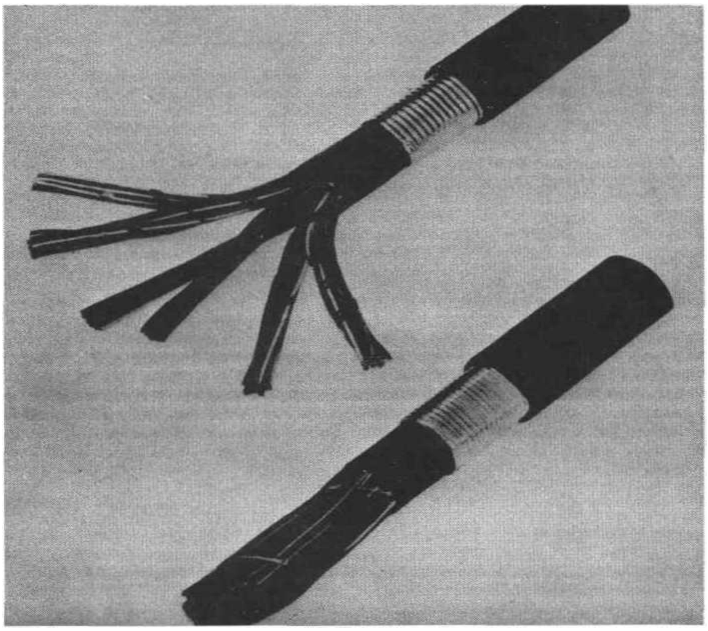


Figure 1.1 Multipair cable.

In response to overcrowded cross arms and high maintenance costs, multipair cable systems were introduced as far back as 1883. Today a single cable may contain anywhere from 6 to 2700 wire pairs. Figure 1.1 shows the structure of a typical cable. When telephone poles are used, a single cable can provide all the circuits required on the route, thereby eliminating the need for crossarms. More recently the preferred means of cable distribution is to bury it directly in the ground (buried cable) or use underground conduit (underground cable).

Table 1.1 lists the most common wire sizes to be found within paired-cable systems. The lower gauge (higher diameter) systems are used for longer distances where signal attenuation and direct-current (dc) resistance can become limiting factors. Figure 1.14 shows attenuation curves for the common gauges of paired cable as a function of frequency. An important point to notice in Figure 1.1 is that the cable pairs are capable of carrying much higher frequencies than required by a telephone quality voice signal (approximately 3.4 kHz).

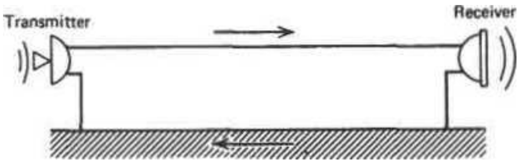


Figure 1.2 Single-wire transmission with ground return.

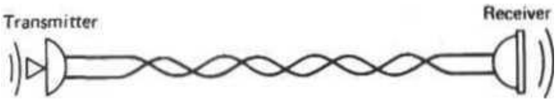


Figure 1.3 Two-wire transmissions.

Two-Wire Versus Four-Wire

All wire-line transmission in the telephone network is based on transmission through pairs of wires. As shown in Figure 1.2, transmission through a single wire (with a ground return) is possible and has been used in the past. However, the resulting circuit is too noisy for customer acceptance. Instead, balanced pairs of wires as shown in Figure 1.3 are used with signals propagating as a voltage difference between the two wires. The electrical current produced by the difference signal flowing through the wires in opposite directions is called a “metallic current.” In contrast, current propagating in the same direction in both wires is referred to as common-mode or longitudinal current. Longitudinal currents are not coupled into a circuit output unless there is an imbalance in the wires that converts some of the longitudinal signal (noise or interference) into a difference signal. Thus the use of a pair of wires for each circuit provides much better circuit quality than does single-wire transmission. Some older switching systems used single-wire (unbalanced) transmission to minimize the number of contacts. Unbalanced circuits were only feasible in small switches where noise and crosstalk could be controlled.

Virtually all subscriber loops in the telephone network are implemented with a single pair of wires. The single pair provides for both directions of transmission. If users on both ends of a connection talk simultaneously, their conversations are superimposed on the wire pair and can be heard at the opposite ends. In contrast, wire-line (and fiber) transmission over longer distances, as between switching offices, is best implemented if the two directions of transmission are separated onto separate wire pairs. It is now commonplace to use fiber for the feeder portion of a subscriber loop, but the drop to a residence is a single pair per telephone.

Longer distance transmission requires amplification and most often involves multiplexing. These operations are implemented most easily if the two directions of transmission are isolated from each other. Thus interoffice trunks typically use two pairs of wires or two fibers and are referred to as four-wire systems. The use of two pairs of wires did not necessarily imply the use of twice as much copper as a two-wire circuit.

Sometimes the bandwidth of a single pair of wires was separated into two subbands that were used for the two directions of travel. These systems were referred to as *derived four-wire systems.* Hence, the term *four-wire* has evolved to imply separate channels for each direction of transmission, even when wires may not be involved. For example, fiber optic and radio systems that use separate channels for each direction are also referred to as *four-wire systems.* .

The use of four-wire transmission had a direct impact on the switching systems of the toll network. Since toll network circuits were four-wire, the switches were designed to separately connect both directions of transmission. Hence, two paths through the switch were needed for each connection. A two-wire switch, as used in older analog end offices, required only one path through the switch for each connection.

Two-Wire-to-Four-Wire Conversion

At some point in a long-distance connection it is necessary to convert from two-wire transmission of local loops to four-wire transmission on long-distance trunks. In the past, the conversion usually occurred at the trunk interface of the (two-wire) end office switch. Newer digital end office switches are inherently “four-wire,” which means the two-wire-to-four-wire conversion point is on the subscriber (line) side of the switch as opposed to the trunk side. A generalized interconnection of two-wire and four-wire facilities for a connection is shown in Figure 1.4. The basic conversion function is provided by hybrid circuits that couple the two directions of transmission as shown. Hybrid circuits have been traditionally implemented with specially interconnected transformers.

More recently, however, electronic hybrids have been developed. Ideally a hybrid should couple all energy on the incoming branch of the four-wire circuit into the two-wire circuit, and none of the incoming four-wire signal should be transferred to the outgoing four-wire branch.

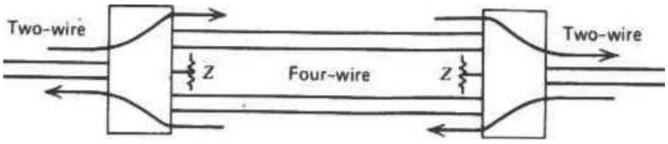


Figure 1.4 Two wire to Four wire conversion

When the impedance matching network Z exactly matches the impedance of the two-wire circuit, near-perfect isolation of the two four-wire branches can be realized. Impedance matching used to be a time-consuming, manual process and was therefore not commonly used. Furthermore, the two-wire circuits were usually switched connections so the impedance that had to be matched would change with each connection. For these reasons the impedances of two-wire lines connected to hybrids were rarely matched. The effect of an impedance mismatch is to cause an echo, the power level of which is related to the degree of mismatch.

Loading Coils

The attenuation curves shown in Figure 1.5 indicate that the higher frequencies of the voice spectrum (up to 3.4 kHz) experience more attenuation than the lower frequencies. This frequency-dependent attenuation distorts the voice signal and is referred to as amplitude distortion. Amplitude distortion becomes most significant on long cable pairs, where the attenuation difference is greatest.

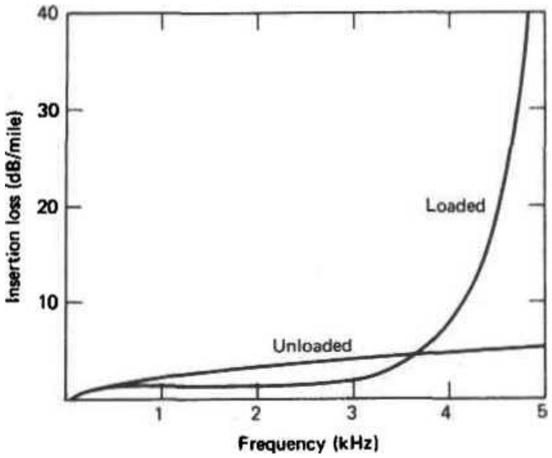


Figure 1.5 Effect of loading on 24-gauge cable pair

The usual method of combating amplitude distortion on intermediate-length (315-mile) wire pairs is to insert artificial inductance into the lines. The extra inductance comes from loading coils that are inserted at 3000-, 4500-, or 6000-ft intervals. Figure 1.5 shows the effect of loading coils on a 24-

gauge loop. Notice that the voiceband response up to 3 kHz is greatly improved, but the effect on higher frequencies is devastating.

Prior to the introduction of wire-line and fiber carrier systems, loading coils were used extensively on exchange area interoffice trunks. Loading coils are also used on the longer, typically rural, subscriber loops. Here, too, carrier systems have displaced most of the single pairs of wires being used on long routes. Multiplexing technique for the time. The same basic technique has since been used in numerous applications with digital speech for satellite and land-line applications. These systems are generally called digital speech interpolation (DSI) systems,

FDM Multiplexing and Modulation

The introduction of cable systems into the transmission plant to increase the circuit packing density of open wire is one instance of multiplexing in the telephone network. This form of multiplexing, referred to as *space division multiplexing,* involves nothing more than bundling more than one pair of wires into a single cable. The telephone network uses two other forms of multiplexing, both of which use electronics to pack more than one voice circuit into the bandwidth of a single transmission medium. Analog frequency division multiplexing (FDM) has been used extensively in point-to-point microwave radios and to a much lesser degree on some obsolete coaxial cable and wire-line systems. FDM is also utilized in fiber optic transmission systems, where it is referred to as wavelength division multiplexing (WDM). Digital time division multiplexing (TDM) is the dominant form of multiplexing used in the telephone networks worldwide.

Frequency Division Multiplexing

As indicated in Figure, an FDM system divides the available bandwidth of the transmission medium into a number of narrower bands or subchannels. Individual voice signals are inserted into the subchannels by amplitude modulating appropriately selected carrier frequencies. As a compromise between realizing the largest number of voice channels in a multiplex system and maintaining acceptable voice fidelity, the telephone companies established 4 kHz as the standard bandwidth of a voice circuit. If both sidebands produced by amplitude modulation are used (as in obsolete N1 or N2 carrier systems on paired cable), the subchannel bandwidth is 8 kHz, and the corresponding carrier frequencies lie in the middle of each subchannel. Since doublesideband modulation is wasteful of bandwidth, single-sideband (SSB) modulation was used whenever the extra terminal costs were justified. The carrier frequencies for single-sideband systems lie at either the upper or lower edge of the corresponding subchannel, depending on whether the lower or upper sideband is selected. The A5 channel bank multiplexer of AT&T used lower sideband modulation. '?

FDM Hierarchy

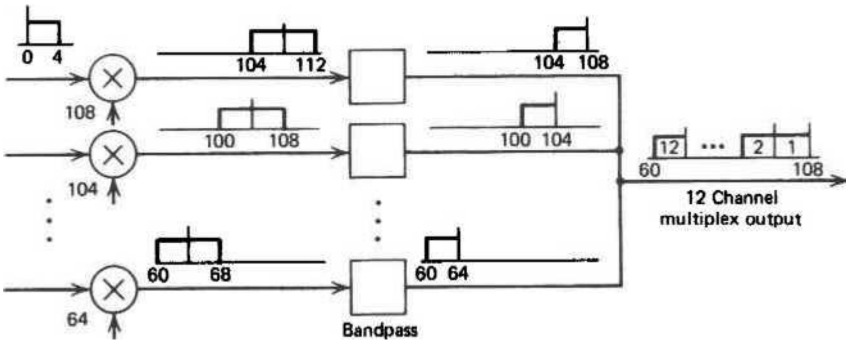
In order to standardize the equipment in the various broadband transmission systems of the original analog network, the Bell System established an FDM hierarchy as provided in Table 1.2. CCITT recommendations specify the same hierarchy at the lower end. Optical technology is customarily defined in terms of the wavelength of the optical signal as opposed to the corresponding frequency. Actually, the usable bandwidth of an FDM voice channel was closer to 3 kHz due to guard bands needed by the FDM separation filters.

TABLE 1.2 FDM Hierarchy of the Bell Network

Multiplex Level	Number of Voice Circuits	Formation	Frequency Band (kHz)
Voice channel	1		0-4
Group	12	12 voice circuits	60-108
Supergroup	60	5 groups	312-552
Mastergroup	600	10 supergroups	564-3,084
Mastergroup	1,200-3,600	Various	312,564-17,548
Jumbogroup	3,600	6 master groups	564-17,548
Jumbogroup Mux	10,800	3 jumbo groups	3,000-60,000

levels. Each level of the hierarchy is implemented using a set of standard FDM modules. The multiplex equipment is independent of particular broadband transmission media.

All multiplex equipment in the FDM hierarchy used SSB modulation. Thus, every voice circuit required approximately 4 kHz of bandwidth. The lowest level building block in the hierarchy is a channel group consisting of 12 voice channels. A channel group multiplex uses a total bandwidth of 48 kHz. Figure 1.20 shows a block diagram of an A5 channel group multiplexer, the most common A-type channel bank used for first-level multiplexing. Twelve modulators using 12 separate carriers generate 12 double-sideband signals as indicated. Each channel is then bandpass filtered to select only the lower sideband of each double-sideband signal. The composite multiplex signal is produced by superposing the filter outputs. Demultiplex equipment in a receiving terminal uses the same basic processing in reverse order.



Notice that a sideband separation filter not only removes the upper sideband but also restricts the bandwidth of the retained signal: the lower sideband. These filters therefore represented a basic point in the analog telephone network that defined the bandwidth of a voice circuit. Since FDM was used on all long-haul analog circuits,long-distance connections provided somewhat less than 4 kHz of bandwidth. (The loading coils discussed previously also produce similar bandwidth limitations into a voice circuit.)

As indicated in Table 1.2, the second level of the FDM hierarchy is a 60-channel multiplex referred to as a supergroup. Figure shows the basic implementation of an LMX group bank that multiplexes five first-level channel groups. The resulting 60- channel multiplex output is identical to that obtained when the channels are individually translated into 4-kHz bands from 312 to 552 kHz. Direct translation requires 60 separate SSB systems with 60 distinct carriers. The LMX group bank, however, uses only five SSB systems plus five lower level modules. Thus two-stage multiplexing, as implied by the LMX group bank, requires more total equipment but achieves economy through the use of common building blocks.

Because a second-level multiplexer packs individual first-level signals together without guard bands, the carrier frequencies and bandpass filters in the LMX group bank must be maintained with high accuracy. Higher level multiplexers do not pack the lower level signals as close together. Notice that a master group, for example, does not provide one voice channel for every 4 kHz of bandwidth. It is not practical to maintain the tight spacing between the wider bandwidth signals at higher frequencies. Furthermore, higher level multiplex signals include pilot tones to monitor transmission link quality and aid in carrier recovery.

1.3.2 Time Division Multiplexing

Basically, time division multiplexing (TDM) involves nothing more than sharing a transmission medium by establishing a sequence of time slots during which individual sources can transmit signals. Thus the entire bandwidth of the facility is periodically available to each source for a restricted time interval. In contrast, FDM systems assign a restricted bandwidth to each source for all time. Normally, all time slots of a TDM system are of equal length. Also, each subchannel is usually assigned a time slot with a common repetition period called a frame interval. This form of TDM is sometimes referred to as synchronous time division multiplexing to specifically imply that each subchannel is assigned a certain amount of transmission capacity determined by the time slot duration and the repetition rate. With this second form of multiplexing, subchannel rates are allowed to vary according to the individual needs of the sources. The backbone digital links of the public telephone network (T-carrier, digital microwave, and fiber optics) use a synchronous variety of TDM.

Time division multiplexing is normally associated only with digital transmission links. Although analog TDM transmission can be implemented by interleaving samples from each signal, the individual samples are usually too sensitive to all varieties of transmission impairments. In contrast, time division switching of analog signals is more feasible than analog TDM transmission because noise and distortion within the switching equipment are more controllable.

T-Carrier Systems

The volume of interoffice telephone traffic in the United States has traditionally grown more rapidly than local traffic. This rapid growth put severe strain on the older interoffice transmission facilities that are designed for lower traffic volumes. Telephone companies were often faced with the necessary task of expanding the number of interoffice circuits. T-carrier systems were initially developed as a cost-effective means for interoffice transmission: both for initial installations and for relief of crowded interoffice cable pairs.

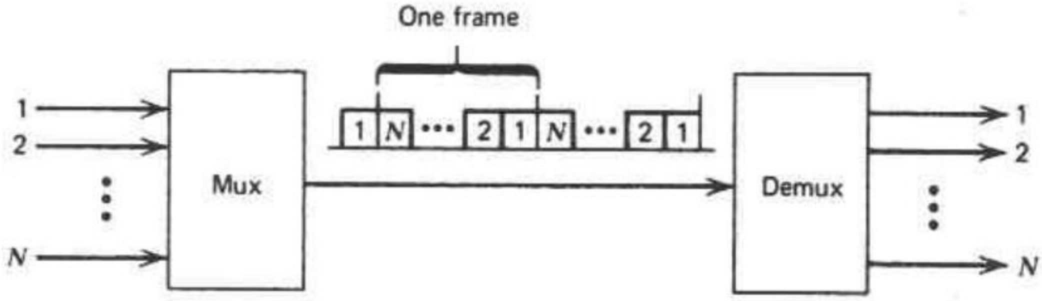


Figure 1.39 Time division multiplexing.

Despite the need to convert the voice signals to a digital format at one end of a T1 line and back to analog at the other, the combined conversion and multiplexing cost of a digital TDM terminal was lower than the cost of a comparable analog FDM terminal. The first T-carrier systems were designed specifically for exchange area trunks at distances between 10 and 50 miles.

A T-carrier system consists of terminal equipment at each end of a line and a number of regenerative repeaters at intermediate points in the line. The function of each regenerative repeater is to restore the digital bit stream to its original form before transmission impairments obliterate the identity of the digital pulses. The line itself, including the regenerative repeaters, is referred to as a span line. The original terminal equipment was referred to as D-type (digital) channel banks, which came in numerous versions. The transmission lines were wire pairs using 16- to 26-gauge cable. A block diagram of a T-carrier system is shown in Figure 1.40.

The first T1 systems used D1A channel banks for interfacing, converting, and multiplexing 24 analog circuits. A channel bank at each end of a span line provided interfacing for both directions of transmission. Incoming analog signals were time division multiplexed and digitized for transmission. When received at the other end of the line, the incoming bit stream was decoded into analog samples, demultiplexed, and filtered to reconstruct the original signals. Each individual TDM channel was assigned 8 bits per time slot. Thus, there were (24)(8) = 192 bits of information in a frame. One additional bit was added to each frame to identify the frame boundaries, thereby producing a total of 193 bits in a frame. Since the frame interval is 125 msec, the basic T1 line rate became 1.544 Mbps. This line rate has been established as the fundamental standard for digital transmission in North America and Japan. The standard is referred to as a DS1 signal (for digital signal 1).

A similar standard of 2.048 Mbps has been established by ITU-T for most of the rest of the world. This standard evolved from a Tl-like system that provides 32 channels at the same rate as the North American channels. Only 30 of the channels in the El standard, however, are used for voice. The other two are used for frame synchronization and signaling. The greatly increased attenuation of a wire pair at the frequencies of a DS1 signal (772 kHz center frequency) mandates the use of amplification at intermediate points of a T1 span line. In contrast to an analog signal, however, a digital signal can not only be amplified but also be detected and regenerated. That is, as long as a pulse can be detected, it can be restored to its original form and relayed to the next line segment. For this reason T1 repeaters are referred to as regenerative repeaters. The basic functions of these repeaters are:

1. Equalization
2. Clock recovery
3. Pulse detection
4. Transmission

Equalization is required because the wire pairs introduce certain amounts of both phase and amplitude distortion that cause intersymbol interference if uncompensated. Clock recovery is required for two basic purposes: first, to establish a timing signal to sample the incoming pulses; second, to transmit outgoing pulses at the same rate as at the input to the line.

Regenerative repeaters are normally spaced every 6000 ft in a T1 span line. This distance was chosen as a matter of convenience for converting existing voice frequency cables to T-carrier lines. Interoffice voice frequency cables typically used loading coils that were spaced at 6000-ft intervals. Since these coils were located at convenient access points (manholes) and had to be removed for high-frequency transmission, it was only natural that the 6000-ft interval be chosen. One general exception is that the first regenerative repeater is typically spaced 3000 ft from a central office. The shorter spacing of this line segment was needed to maintain a relatively strong signal in the presence of impulse noise generated by older switching machines.

The operating experience of T1 systems was so favorable that they were continually upgraded and expanded. One of the initial improvements produced TIC systems that provide higher transmission rates over 22-gauge cable. A TIC line operates at 3.152 Mbps for 48 voice channels, twice as many as a T1 system.

Another level of digital transmission became available in 1972 when the T2 system was introduced. This system was designed for toll network connections. In contrast, T1 systems were originally designed only for exchange area transmission. The T2 system provided for 96 voice channels at distances up to 500 miles. The line rate was 6.312 Mbps, which is referred to as a DS2 standard. The transmission media was special low-capacitance 22-gauge cable. By using separate cables for each direction of transmission and the specially developed cables, T2 systems could use repeater spac- ings up to 14,800 ft in low-noise environments. The emergence of optical fiber systems made copper-based T2 transmission systems obsolete.

TDM Hierarchy

In a manner analogous to the FDM hierarchy, AT&T established a digital TDM hierarchy that has become the standard for North America. Starting with a DS1 signal as a fundamental building block, all other levels are implemented as a combination of some number of lower level signals. The designation of the higher level digital multiplexers reflects the respective input and output levels. For example, an M12 multiplexer combines four DS1 signals to form a single DS2 signal. Table lists the various multiplex levels, their bit rates, and the transmission

Digital TDM Signals of North America and Japan

Digital Signal Number	Number of Voice Circuits	Multiplexer Designation	Bit Rate (Mbps)	Transmission Media
DS1	24	D channel bank (24 analog inputs)	1.544	T1 paired cable
DS1C	48	M1C (2 DS1 inputs)	3.152	T1C paired cable
DS2	96	M12 (4 DS1 inputs)	6.312	T2 paired cable
DS3	672	M13 (28 DS1 inputs)	44.736	Radio, Fiber
DS4	4032	M34 (6 DS3 inputs)	274.176	T4M coax, WT4 waveguide, radio

media used for each. Notice that the bit rate of a high-level multiplex signal is slightly higher than the combined rates of the lower level inputs. A similar digital hierarchy has also been established by ITU-T as an international standard.

Digital Pair-Galn Systems

Following the successful introduction of T1 systems for interoffice trunks, most major manufacturers of telephone equipment developed digital TDM systems for local distribution. These systems are most applicable to long rural loops where the cost of the electronics is offset by the savings in wire pairs. No matter what the distance is, unexpected growth can be most economically accommodated by adding electronics, instead of wire, to produce a pair-gain system. The possibility of trailer parks, apartment houses, or Internet service providers springing up almost overnight causes nightmares in the minds of cable plant forecasters. Pair-gain systems provide a networking alternative to dispel those nightmares. Digital pair-gain systems are also useful as alternatives to switching offices in small communities. Small communities are often serviced by small automatic switching systems normally unattended and remotely controlled from a larger switching office nearby. These small community switches are referred to as community dial offices Because T2 transmission systems have become obsolete, the Ml 2 function exists only in a functional sense within Ml3 multiplexers, which multiplex 28 DS1 signals into 1 DS3 signal. (CDOs). A CDO typically provides only limited service features to the customers and often requires considerable maintenance. Because digital pair-gain systems lower transmission costs for moderate-sized groups of subscribers, they are a viable alternative to a CDO: Stations in the small community are serviced from the central office by way of pair-gain systems. A fundamental consideration in choosing between pair gain systems and remote switching involves the traffic volumes and calling patterns within the small community.

The first two digital pair-gain systems used in the Bell System were the subscriber loop multiplex (SLM) system and, its successor, the subscriber loop carrier (SLC-40) system. Although these systems used a form of voice digitization (delta modulation) different from that used in T-carrier systems (pulse code modulation), they both used standard T1 repeaters for digital transmission at 1.544 Mbps. Both systems also converted the digitized voice signals back into individual analog interfaces at the end office switch to achieve system transparency. Notice that the SLM system provided both concentration and multiplexing (80 subscribers for 24 channels) while the SLC-40 was strictly a multiplexer (40 subscribers assigned in a one-to-one manner to 40 channels).

ITU Digital Hierarchy

Level Number	**Number of Voice Circuits**	**Multiplexer Designation**	**Bit Rate (Mbps)**
E1	30		2.048
E2	120	M12	8,448
E3	480	M23	34.368
E4	1920	M34	139.264
E5	7680	M45	565.148

The SLM and SLC-40 systems used delta modulation voice coding because it was simpler than pulse code modulation as used in T1 systems and was therefore less costly to implement on a per-channel basis a desirable feature for modular system implementations. The original T1 systems, on the other hand, minimized electronics costs by using common encoders and decoders, which precluded implementation of less than 24 channels (an unnecessary feature in an interoffice application). By the late 1970s low-cost, integrated circuit implementations of standard pulse code modulation became available that led the way to the first (1979) installation of the SLC-96, a subscriber carrier system using voice coding that was compatible with T1 systems and the emerging digital end office switching machines.

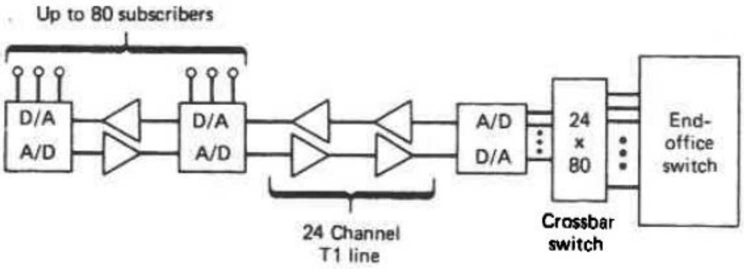


Figure 1.41 Subscriber loop multiplexer.

The SLC-96 system (which is functionally equivalent to four T1 lines) can interface directly with a digital end office and not be demultiplexed into 24 distinct analog interfaces. Thus this capability, which is referred to as integrated digital loop carrier (IDLC), greatly reduces the prove-in distance where the digital carrier becomes less expensive than separate subscriber pairs.

Data under Voice

After the technology of T-carrier systems had been established, AT&T began offering leased digital transmission services for data communications. This service, known as Dataphone Digital Service

(DDS), uses T1 transmission links with special terminals (channel banks) that provide direct access to the digital line. An initial drawback of DDS arose because T-carrier systems were originally used only for exchange area and short toll network trunks. Without some form of long-distance digital transmission, the digital circuits in separate exchange areas could not be interconnected. AT&T's original response to long-distance digital transmission was the development of a special radio terminal called the 1A radio digital terminal (1A-RDT). This terminal encoded one DS1 signal (1.544 Mbps) into less than 500 kHz of bandwidth. As shown in Figure, a signal of this bandwidth was inserted below the lowest frequency of a master group multiplex. Since this frequency band is normally unused in TD or TH analog radio systems, the DS1 signal could be added to existing analog routes without displacing any voice channels. The use of frequencies below those used for voice signals leads to the designation "data under voice" (DUV).

It is important to point out that DUV represented a special development specifically intended for data transmission and not for voice services. In fact, DUV was used only to provide long-distance digital transmission facilities for DDS.

PULSE TRANSMISSION

All digital transmission systems are designed around some particular form of pulse response. Even carrier systems must ultimately produce specific pulse shapes at the detection circuitry of the receiver. As a first step, consider the perfectly square pulse shown in Figure 4.1. The frequency spectrum corresponding to the rectangular pulse is derived in Appendix A and shown in Figure 4.2. It is commonly referred to as a sin(x)/x response:

$$F(\omega) = (T)\,\frac{\sin(\omega T/2)}{\omega T/2}$$

Where w = radian frequency *2nf,T* -duration of a signal interval

The high percentage of energy within this band indicates that the signal can be confined to a bandwidth of 1/2" and still pass a good approximation to the ideal waveform. In theory, if only the sample values at the middle of each signal interval are to be preserved, the bandwidth can be confined to *HIT.* From this fact the maximum baseband signaling rate in a specified bandwidth is determined as

$$\textbf{Rmax=2BW} \qquad \textbf{(4.2)}$$

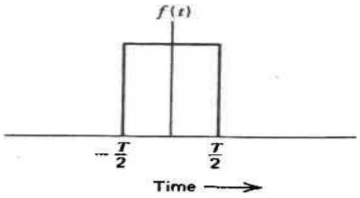


Figure 4.1 Definition of a square pulse

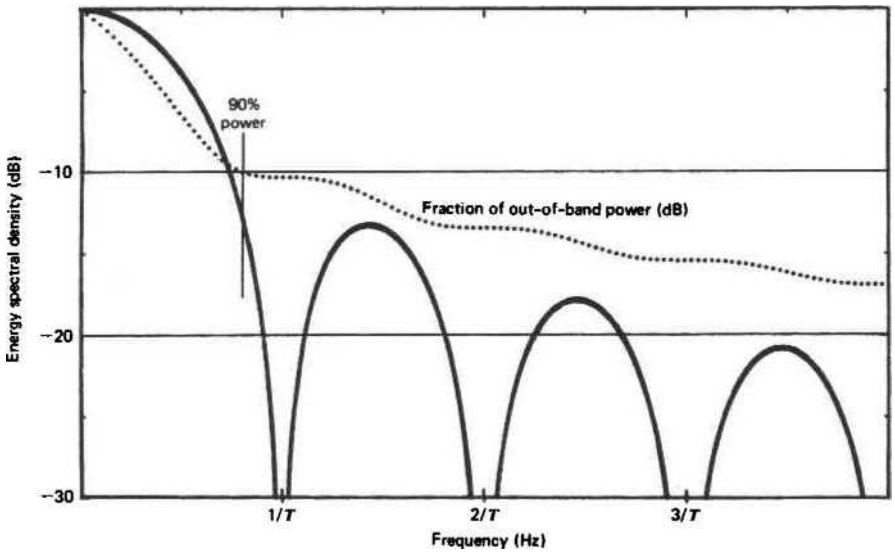


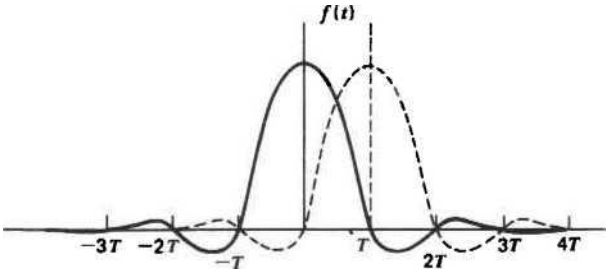
Figure 4.2 Spectrum of square pulse with duration *T*.

Where, R = signaling rate, = 1 /T

BW - available bandwidth

The maximum signaling rate achievable through a low-pass bandwidth with no intersymbol interference is equal to twice the bandwidth. This rate $R_{max}$is sometimes referred to as the *Nyquist rate.*

Although discrete, square-shaped pulses are easiest to visualize, preservation of the square shape requires wide bandwidths and is therefore undesirable. A more typical shape for a single pulse is shown in Figure. The ringing on both sides of the main part of the pulse is a necessary accompaniment to a channel with a limited bandwidth.



Normally, a digital transmission link is excited with square pulses (or modulated equivalents thereof), but bandlimiting filters and the transmission medium itself combine to produce a response like the one shown. Figure 4.3 shows pulse output in negative time so the center of the pulse occurs at *t* = 0. Actually, the duration of the preringing is limited to the delay of the channel, the filters, and the equalizers.

An important feature of the pulse response is that, despite the ringing, a pulse can be transmitted once every *T* seconds and be detected at the receiver without interference from adjacent pulses. Obviously, the sample time must coincide with the zero crossings of the adjacent pulses. Pulse responses like the one shown in Figure can be achieved in channel bandwidths approaching the minimum (Nyquist) bandwidth equal to one-half of the signaling rate.

Intersymbol Interference

As the signaling rate of a digital transmission link approaches the maximum **ra**te for a given bandwidth, both the channel design and the sample times become more critical. Small perturbations in the channel response or the sample times produce nonzero overlap at the sample times called intersymbol interference. The main causes of intersymbol interference are:

1. Timing inaccuracies
2. Insufficient bandwidth
3. Amplitude distortion
4. Phase distortion

Timing Inaccuracies

Timing inaccuracies occurring in either the transmitter or the receiver produce intersymbol interference. In the transmitter, timing inaccuracies cause intersymbol interference if the rate of transmission does not conform to the ringing frequency designed into the channel. Timing inaccuracies of this type are insignificant unless extremely sharp filter cutoffs are used while signaling at the Nyquist rate.

Since timing in the receiver is derived from noisy and possibly distorted receive signals, inaccurate sample timing is more likely than inaccurate transmitter timing. Sensitivity to timing errors is small if the transmission rate is well below the Nyquist rate.

Insufficient Bandwidth

The ringing frequency shown in Figure is exactly equal to the theoretical minimum bandwidth of the channel. If the bandwidth is reduced further, the ringing frequency is reduced and intersymbol interference necessarily results.

Some systems purposely signal at a rate exceeding the Nyquist rate, but do so with prescribed amounts of intersymbol interference accounted for in the receiver. These systems are commonly referred to as partial-response systems so called because the channel does not fully respond to an input during the time of a single pulse. The most common forms of partial-response systems are discussed in a later section.

Amplitude Distortion

Digital transmission systems invariably require filters to bandlimit transmit spectrums and to reject noise and interference in receivers. Overall, the filters are designed to produce a specific pulse response. When a transmission medium with predetermined characteristics is used, these characteristics can be included in the overall filter design. However, the frequency response of the channel cannot always be predicted adequately. A departure from the desired frequency response is referred to as amplitude distortion and causes pulse distortions (reduced peak amplitudes and improper ringing frequencies) in the time domain. Compensation for irregularities in the frequency response of the channel is referred to as *amplitude equalization.*

Phase Distortion

When viewed in the frequency domain, a pulse is represented as the superposition of frequency components with specific amplitude and phase relationships. If the relative amplitudes of the frequency components are altered, amplitude distortion results as above. If the phase relationships of the components are altered, phase distortion occurs. Basically, phase distortion results when the frequency components of a signal experience differing amounts of delay in the transmission link. Compensation of phase distortion is referred to as *phase equalization.*

ASYNCHRONOUS VERSUS SYNCHRONOUS TRANSMISSION

There are two basic modes of digital transmission involving two fundamentally different techniques for establishing a time base (sample clock) in the receiving terminal of a digital transmission link. The first of these techniques is asynchronous transmission, which involves separate transmissions of groups of bits or characters. Within an individual group a specific predefined time interval is used for each discrete signal. However, the transmission times of the groups are unrelated to each other. Thus the sample clock in the receiving terminal is reestablished for reception of each group. With the second technique, called synchronous transmission, digital signals are sent continuously at a constant rate. Hence the receiving terminal must establish and maintain a sample clock that is synchronized to the incoming data for an indefinite period of time.

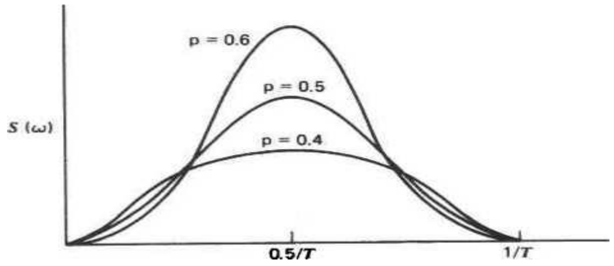


Figure 4.10 Spectral density of bipolar coding.

Code Space Redundancy

In essence, bipolar coding uses a ternary code space but only two of the levels during any particular signal interval. Hence bipolar coding eliminates dc wander with an inefficient and redundant use of the code space. The redundancy in the waveform also provides other benefits. The most important additional benefit is the opportunity to monitor the quality of the line with no knowledge of the nature of the traffic being transmitted. Since pulses on the line are supposed to alternate in polarity, the detection of two successive pulses of one polarity implies an error. This error condition is known as a *bipolar violation.* No single error can occur without a bipolar violation also occurring. Hence the bipolar code inherently provides a form of line code parity. The terminals of T1 lines are designed to monitor the frequency of occurrence of bipolar violations, and if the frequency of occurrence exceeds some threshold, an alarm is set.

In T-carrier systems, bipolar violations are used merely to detect channel errors. By adding some rather sophisticated detection circuitry, the same redundancy can be used for correcting errors in addition to detecting them. Whenever a bipolar violation is detected, an error has occurred in one of the bits between and including the pulses indicating the violation. Either a pulse should be a 0 or an intervening 0 should have been a pulse of the opposite polarity. By examining the actual sample values more closely, a decision can be made as to where the error was most likely to have occurred. The bit with a sample value closest to its decision threshold is the most likely bit in error. This technique belongs to a general class of decision algorithms for redundant signals called maximum likelihood or Viterbi decoders . Notice that this method of error correction requires storage of pulse amplitudes. If decision values only are stored, error correction cannot be achieved (only error detection).

An additional application of the unused code space in bipolar coding is to purposely insert bipolar violations to signify special situations such as time division multiplex framing marks, alarm conditions, or special codes to increase the timing content of the line signals. Since bipolar violations are not normally part of the source data, these special situations are easily recognized. Of course, the ability to monitor the quality of the line is compromised when bipolar violations occur for reasons other than channel errors.

4.3.3 Binary Af-Zero Substitution

A major limitation of bipolar (AMI) coding is its dependence on a minimum density of l’s in the source code to maintain timing at the regenerative repeaters. Even when strings of 0’s greater than 14 are precluded by the source, a low density of pulses on the line increases timing jitter and therefore produces higher error rates. Binary Non-zero substitution (BNZS) augments a basic bipolar code by replacing all strings of *N* 0’s with a special JV-length code containing several pulses that purposely produce bipolar violations. Thus the density of pulses is increased while the original Hata are obtained by recognizing the bipolar violation codes and replacing them at the receiving terminal with *N*0’s.

As an example, a three-zero substitution algorithm (B3ZS) is described. This particular substitution algorithm is specified for the standard DS-3 signal interface in North America, It was also used in the LD-4 coaxial transmission system in Canada.

In the B3ZS format, each string of three 0’s in the source data is encoded with either 00V or BOV. A 00V line code consists of 2-bit intervals with no pulse (00) followed by a pulse representing a bipolar violation (V). A BOV line code consists of a single pulse in keeping with the bipolar alternation (B), followed by no pulse (0), and ending with a pulse with a violation (V). With either substitution, the bipolar violation occurs in the last bit position of the three 0’s replaced by the special code. Thus the position of the substitution is easily identified.

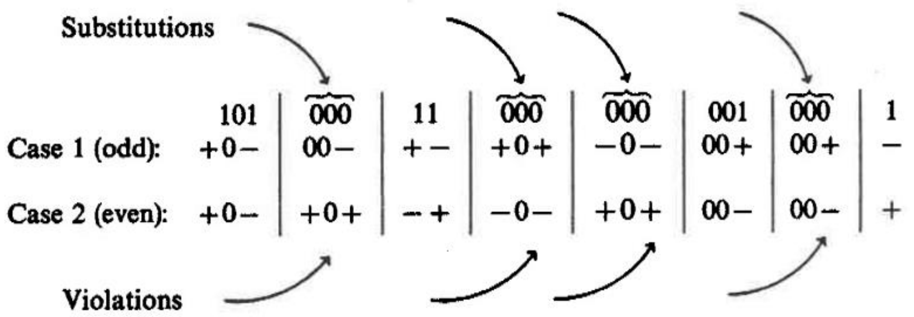
The decision to substitute with 00V or BOV is made so that the number of B pulses (unviolated pulses) between violations (V) is odd. Hence if an odd number of l’s has been transmitted since the last substitution, 00V is chosen to replace three 0’s. If the intervening number of l’s is even, BOV is chosen. In this manner all purposeful violations contain an odd number of intervening bipolar pulses. Also, bipolar violations alternate in polarity so that dc wander is prevented. An even number of bipolar pulses between violations occurs only as result of a channel error. Furthermore, every purposeful violation is immediately preceded by a 0. Hence considerable systematic redundancy remains in the line code to facilitate performance monitoring.

Example. Determine the B3ZS line code for the following data sequence: 101000110000000010001. Use + to indicate a positive pulse, - to indicate a negative pulse, and 0 to indicate no pulse.

B3ZS Substitution Rules

Polarity of Preceding Pulse	Number of Bipolar Pulses (1’s) Since Last Substitution	
	Odd	Even
—	00-	+0+
+	00+	-0-

Solution. There are two possible sequences depending on whether an odd or even number of pulses has been transmitted following the previous violation:

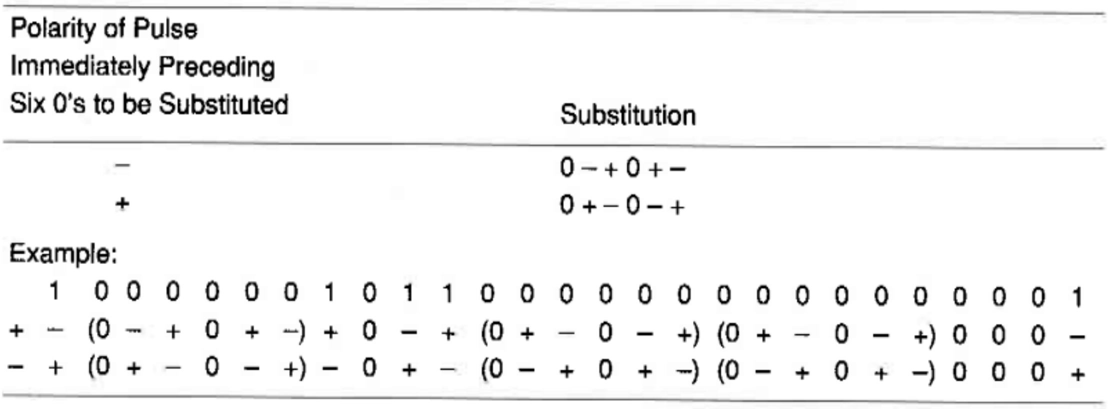


Example indicates that the process of breaking up strings of O’ s by substituting with bipolar violations greatly increases the minimum density of pulses in the line code. In fact, the minimum density is 33% while the average density is just over 60%. Hence the B3ZS format provides a continuously strong timing component. Notice that all BNZS coding algorithms guarantee continuous timing information with no restrictions on source data. Hence BNZS coding supports any application in a completely transparent manner.

Another BNZS coding algorithm is the B6ZS algorithm used on obsolete T2 transmission lines. This algorithm produces bipolar violations in the second and fifth bit positions of the substituted sequence.

ITU recommends another BNZS coding format referred to as high-density bipolar (HDB) coding. As implemented in the El primary digital signal, HDB coding replaces strings of four 0’s with sequences containing a bipolar violation in the last bit position. Since this coding format precludes strings of 0’s greater than three, it is referred to as HDB3 coding. The encoding algorithm is basically the same as the B3ZS algorithm described earlier. Notice that substitutions produce violations only in the fourth bit position, and successive substitutions produce violations with alternating polarities.

TABLE 4.2 B6ZS Substitution Rules



Polarity of Pulse Immediately Preceding Six 0's to be Substituted	Substitution
−	0 − + 0 + −
+	0 + − 0 − +

Example:

	1	0	0	0	0	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
+	−	(0	−	+	0	+	−)	+	0	−	+	(0	+	−	0	−	+)	(0	+	−	0	−	+)	0	0	0	−
−	+	(0	+	−	0	−	+)	−	0	+	−	(0	−	+	0	+	−)	(0	−	+	0	+	−)	0	0	0	+

A fundamental feature of end-to-end digital connectivity as provided by ISDN is 64-kbps transparent channels referred to as clear-channel capability (CCC) [18], Two aspects of a bipolar/AMI line code as used on T1 lines preclude CCC: robbed signaling in the least significant bit of every sixth frame and the need to avoid all-0’s codewords on the channel. Bit robbing for signaling is avoided with common- channel signaling (also an inherent requirement for ISDN deployment). Two means of augmenting T1 lines to allow transparent channels have been developed.

Digital Biphase

Bipolar coding and its extensions BNZS and PST use extra encoding levels for flexibility in achieving desirable features such as timing transitions, no dc wander, and performance monitor ability. These features are obtained by increasing the code space and not by increasing the bandwidth. (The first spectral null of all codes discussed so far, including an NRZ code, is located at the signaling rate 1/7Y)

Many varieties of line codes achieve strong timing and no dc wander by increasing the bandwidth of the signal while using only two levels for binary data. One of the most common of these codes providing both a strong timing component and no dc wander is the digital biphase code, also referred to as “diphase” or a “Manchester” code.

A digital biphase code uses one cycle of a square wave at a particular phase to encode a 1 and one cycle of an opposite phase to encode a 0. Notice that a transition exists at the center of every signaling interval. Hence strong timing components are present in the spectrum. Furthermore, logic 0 signals and logic 1 signals both contain equal amounts of positive and negative polarities. Thus dc wander is nonexistent. A digital biphase code, however, does not contain redundancy for performance monitoring. If in-service performance monitoring is desired, either parity bits must be inserted into the data stream or pulse quality must be monitored.

The frequency spectrum of a digital biphase signal is derived in Appendix C

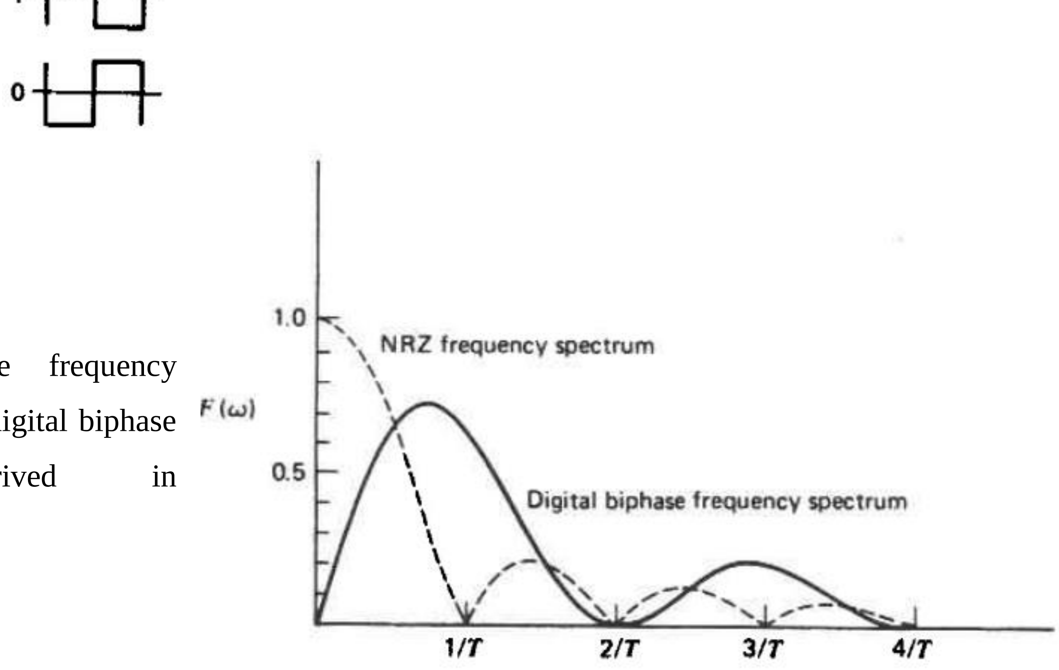


Figure 4.13 Power spectral density of digital biphase.

and plotted in Figure 4.13, where it can be compared to the spectrum of an NRZ signal Notice that a digital biphase signal has its first spectral null at 2*IT*. Hence the extra timing transitions and elimination of dc wander come at the expense of a higher frequency signal. In comparison to three-level bipolar codes, however, the digital biphase code has a lower error rate for equal signal-to-noise ratios. Examination of the frequency spectra in Figure shows that the diphase spectrum is similar to an NRZ spectrum but translated so it is centered about *1/T* instead of direct current. Hence digital biphase actually represents digital modulation of a square wave carrier with one cycle per signal interval. Logic 1 's cause the square wave to be multiplied by +1 while logic 0's produce multiplication by -1. The "Ethernet" IEEE 802.3 local area data network uses digital biphase (Manchester) coding.

4.3.6 Differential Encoding

One limitation of NRZ and digital biphase signals, as presented up to this point, is that the signal for a 1 is exactly the negative of a signal for a 0. On many transmission media, it may be impossible to determine an absolute polarity or an absolute phase reference. Hence the decoder may decode all l's as 0's and vice versa. A common remedy for this ambiguity is to use differential encoding that encodes a 1 as a change of state and encodes a 0 as no change in state. In this manner no absolute reference is necessary to decode the signal. The decoder merely detects the state of each signal interval and compares it to the state of the previous interval. If a change occurred, a 1 is decoded. Otherwise, a 0 is determined.

Differential encoding and decoding do not change the encoded spectrum of purely random data (equally likely and uncorrelated l's and 0's) but do double the error rate. If the detector makes an error in estimating the state of one interval, it also makes an error in the next interval. An example of a differentially encoded NRZ code and a differentially encoded diphase signal is shown in Figure 4.14. All signals of differentially encoded diphase retain a transition at the middle of an interval, but only the 0's have a transition at the beginning of an interval.

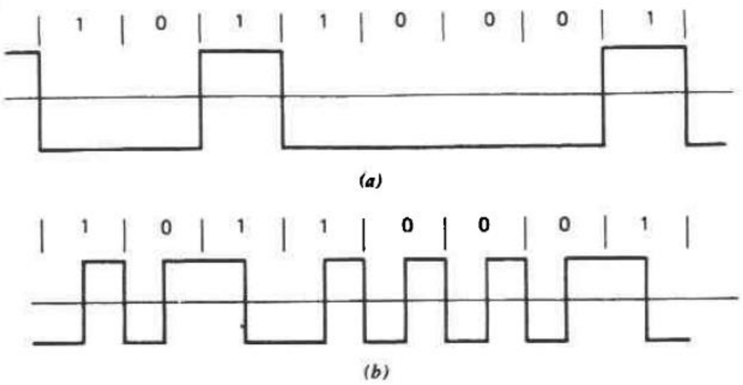


Figure 4.14 Differentially encoded NRZ and diphase signals: (*a*) differentially encoded NRZ; (*b*) differentially encoded diphase.

The 6 CRC bits (CB1 to CB6) of each extended superframe represent a CRC check of all 4608 information bits in the previous superframe. Besides providing end-to-end performance monitoring, the CRC virtually precludes the chances of false framing on a data bit position. Even though static user data can easily simulate the ITS, it is extremely unlikely that user data can spuriously generate valid CRC codes in successive superframes.

The performance parameters measured and reported by the 4-kbps data link (DL) are framing bit errors, CRC errors, out-of-frame (OOF) events, line code (bipolar) violations, and controlled slip events. Individual events are reported as well as event summaries. The four performance summaries reported are:

1. Errored seconds (ESs) (ES = at least one CRC event)
2. Bursty seconds (BSs) (BS = 2-319 ESs)
3. Severely errored seconds (SESs) (SES = >319 ESs or OOFs)
4. Failed seconds (FSs) (10 consecutive SESs)

ESF CSUs typically determine the above parameters on 15-min intervals and store them for up to 24 hr for polling by a controller, The SES report conforms to ITU recommendation G.821. In addition to supporting remote interrogation of performance statistics, the data link carries alarm information, loopback commands, and protection switching commands.

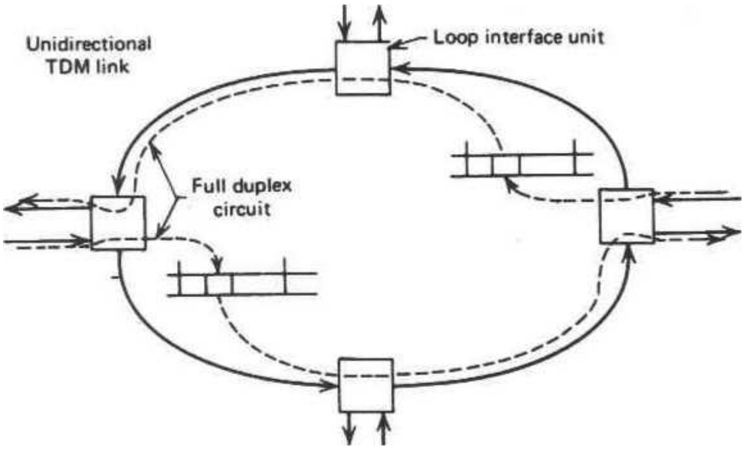
In addition to the previously mentioned features, ESF introduces a new option for per-channel signaling via the robbed signaling bits in every sixth frame. Because an ESF is 24 frames long, there are four signaling bits in every channel in every superframe as opposed to 2 bits in SF format (Figure 4.34). Whereas the two signaling bits in the SF format are designated as A and B bits, the four bits in the ESF case are designated A, B, C, and D. Three signaling modes are defined: 2-state where all bits are A bits, 4-state where the signaling bits are ABAB, and 16-state where the signaling bits are ABCD. The SF format provides the first two signaling modes but not the last.

TIME DIVISION MULTIPLEX LOOPS AND RINGS

In this section a particular form of a TDM network is described that is quite useful in interconnecting distributed nodes. The basic structure of interest is referred to as a TDM loop or TDM ring and is shown in Figure 4.36.

Basically, a TDM ring is configured as a series of unidirectional (two-wire) links arranged to form a closed circuit or loop. Each node of the network is implemented with two fundamental operational features. First, each node acts as a regenerative repeater merely to recover, the incoming bit stream and retransmit it. Second, the net-

Calculation of the CRC actually includes F bits that are set to 1 for purposes of CRC calculation only. Thus, channel errors in the F bits do not create CRC errors (unless they occur in the CRC bits themselves.The nodes recognize the TDM frame structure and communicate on the loop by removing and inserting data into specific time slots assigned to each node. As indicated in Figure, a full-duplex connection can be established between any two nodes by assigning a single time slot or channel for a connection. One node inserts information into the assigned time slot that propagates around the loop to the second node. The destination node removes data as the assigned time slot passes by and inserts return data in the process. The return data propagates around the loop to the original node where it is removed and replaced by new data, and so forth.



Time division multiplex loop.

Since other time slots are not involved with the particular connection shown, they are free to be used for other connections involving arbitrary pairs of nodes. Hence a TDM loop with C time slots per frame can support C simultaneous full-duplex connections.

If, as channels become available, they are reassigned to different pairs of nodes, the transmission facilities can be highly utilized with high concentration factors and provide low blocking probabilities between all nodes. Thus a fundamental attraction of a loop network is that the transmission capacity can be assigned dynamically to meet changing traffic patterns. In contrast, if a star network with a centralized switching node is used to interconnect the nodes with four-wire links, many of the links to particular nodes would be underutilized since they cannot be shared as in a loop configuration.

Another feature of the loop-connected network is the ease with which it can be reconfigured to accommodate new nodes in the network. A new access node is merely inserted into the nearest link of the network and the new node has complete connectivity to all other nodes by way of the TDM channels. In contrast, a star structured network requires transmission to the central node and expansion of the centralized switching facilities.The ability to reassign channels to arbitrary pairs of nodes in a TDM loop implies that the loop is much more than a multiplexer. It is, in fact, a distributed transmission and switching system. The switching capabilities come about almost as a by-product of TDM transmission. TDM loops represent the epitome of integrated transmission and switching.

TDM loops have been used within computer complexes to provide high capacity and high interconnectivity between processors, memories, and peripherals. The loop structure in this application is sometimes more attractive than more conventional bus structures since all transmission is unidirectional and therefore avoids timing problems on bidirectional buses that limit their physical length. Furthermore, as more nodes are added to a bus, the electrical loading increases, causing a limitation on the number of nodes that can be connected to a bus. Loops, on the other hand, have no inherent limits of transmission length or numbers of nodes.

The loop structure of is topologically identical to the token-passing ring developed by IBM and standardized by the IEEE as a 802.5 local area network. However, a token-passing ring operates differently than a TDM loop in that there is only one channel. When a node on a ring becomes active, it uses the entire capacity of the outgoing link until it is through sending its message. In contrast, a node on a loop uses only specific time slots in the TDM structure, allowing other nodes to be simultaneously "connected" using other time slots. In essence, a TDM loop is a distrib- uted-circuit switch and an 802.5 ring is a distributed-packet switch.

A particularly attractive use of a loop with high-bandwidth links is shown in Figure. This figure illustrates the use of add-drop multiplexers (ADMs) that access whatever bandwidth is needed at a local node but pass the rest on to other nodes. In typical applications the amount of bandwidth allocated to each node is quasi-static: It is changed only in response to macroscopic changes in traffic patterns, possibly as a function of the time of day. This basic operation is generally referred to as a cross-connect function as opposed to a switching function, which involves call-by-call recon- figurations. An important point to note about Figure 4.37 is the ability to utilize a general-purpose physical topology but define an arbitrary functional topology on top of it.

Figur **ures in TDM loops.**

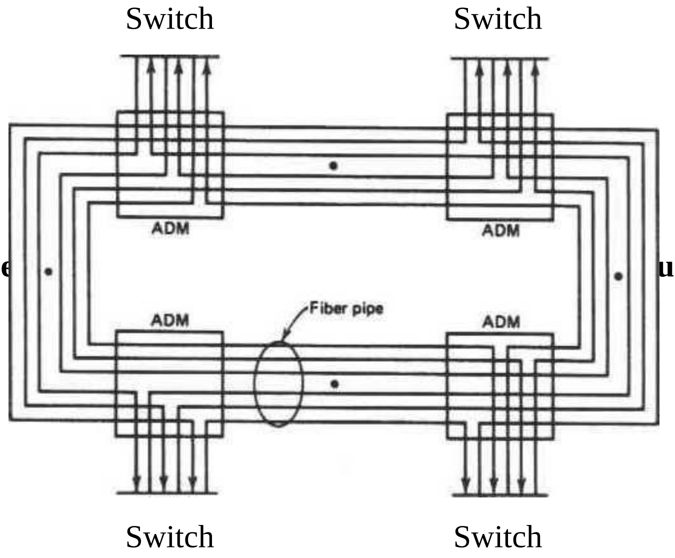


Figure 4.37. Functional mesh, fiber loop and ADMs

One obvious limitation of a loop is its vulnerability to failures of any link or node. The effect of a node failure can be minimized by having bypass capabilities included in each node. When bypassed, a node becomes merely a regenerative repeater, as on T-carrier transmission links. Link failures can be circumvented by providing alternate facilities. Figure 4.38 shows one particular structure using a second, reverse-direction loop to provide backup capabilities in the case of failures. When fully operational, the network can use the reverse loop as a separate, independent network for traffic as needed. Whenever a failure occurs, the nodes adjacent to the break establish a new loop by connecting the forward path to the reverse path at both places. Hence all nodes continue to have full connectivity to any node on the new loop.

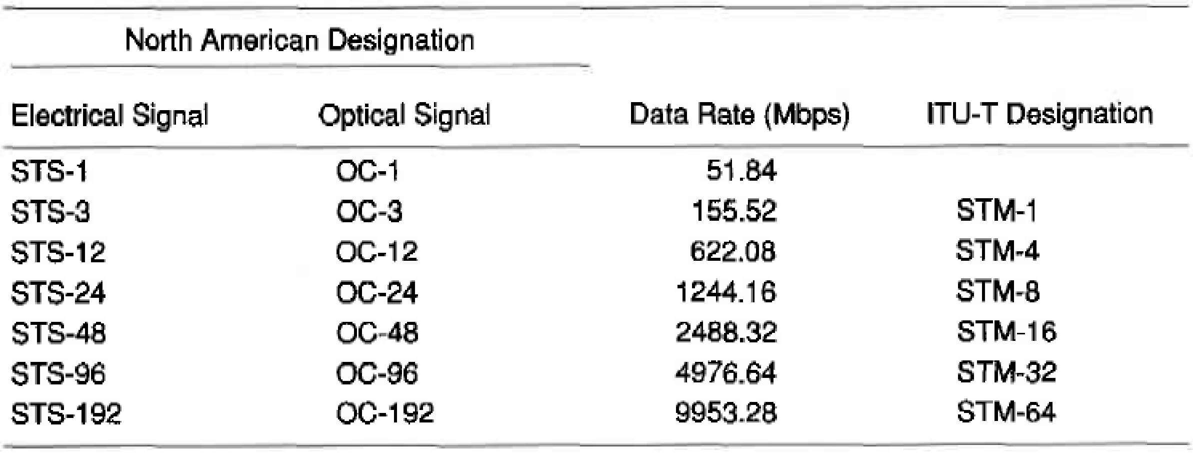
A particular example of the use of the dual reverse loop for both protection and distributed queued access to the channels is the distributed queued dual-bus (DQDB) system developed by QPSX in Australia and standardized by the IEEE as an 802.6 metropolitan area network.

SONET/SDH

The first generations of fiber optic systems in the public telephone network used proprietary architectures, equipment, line codes, multiplexing formats, and maintenance procedures. Some commonality with other systems in the network came from suppliers who also supplied digital radio systems. In these cases, the multiplexing formats and maintenance protocols emulated counterparts in the radio systems, which also had proprietary architectures. The only thing in common with all of the radio and fiber systems from all of the suppliers was that the interface to the network was some number of DS3 cross-connect signals. Proprietary multiplexing formats for multiple DS3 signals evolved because there was no higher level standard compatible with the applications. A DS4 signal, which is composed of six DS3 signals, requires too much bandwidth for radio systems and carries a larger cross section of channels (4032) than needed in many applications.

The Regional Bell Operating Companies and interexchange carriers (IXCs), the users of the equipment, naturally wanted standards so they could mix and match equipment from different suppliers. This became particularly important as a result of competition among the IXCs who desired fiber interfaces to the local exchange carriers (LECs) but did not want to necessarily buy from the same suppliers as the LECs. (It might be necessary for an IXC to interface with a different supplier at each LEC.) To solve these problems, and others, Bellcore initiated an effort that was later taken up by the T1X1 committee of the Exchange Carriers Standards Association (ECS A) to establish a standard for connecting one fiber system to another at the optical level (i.e., “in the glass”). This standard is referred to as the synchronous optical network (SONET). In the late stages of the development of this standard, CCITT became involved so that a single international standard exists for fiber interconnect between telephone networks of different counties. Internationally, the standard is known as the synchronous digital hierarchy (SDH), The SONET standard addresses the following specific issues:

1. Establishes a standard multiplexing format using some number of 51.84-Mbps (STS-1) signals as building blocks.
2. Establishes an optical signal standard for interconnecting equipment from different suppliers.



North American Designation			
Electrical Signal	Optical Signal	Data Rate (Mbps)	ITU-T Designation
STS-1	OC-1	51.84	
STS-3	OC-3	155.52	STM-1
STS-12	OC-12	622.08	STM-4
STS-24	OC-24	1244.16	STM-8
STS-48	OC-48	2488.32	STM-16
STS-96	OC-96	4976.64	STM-32
STS-192	OC-192	9953.28	STM-64

3. Establishes extensive operations, administrations, maintenance, and provisioning (OAM&P) capabilities as part of the standard.
4. Defines multiplexing formats for carrying existing digital signals of the asynchronous multiplexing hierarchy (DS1, DS1C, DS2, DS3).
5. Supports CCITT (ITU-T) digital signal hierarchy (El, E2, E3, E4).
6. Defines a DSO identifiable mapping format for DS1 signals.

7. Establishes a flexible architecture capable of accommodating other applications such as broadband ISDN with a variety of transmission rates. Wide-bandwidth signals (greater than 51.84 Mbps) are accommodated by concatenating multiple STS-1 signals. A STS-3c signal, for example, is an 155.52-Mbps signal that is treated by the network as a single entity.

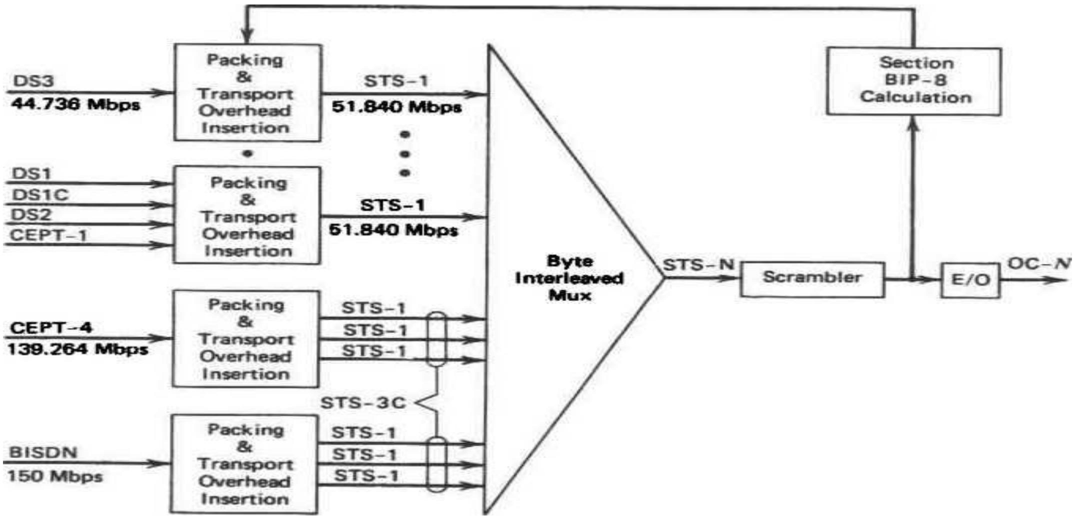
At the lowest level is the basic SONET signal referred to as the synchronous transport signal level 1 (STS-1). Higher level signals are referred to as STS-N signals. An STS-N signal is composed of ~~N~~ byte-interleaved STS-1 signals. The optical counterpart of each STS-N signal is an optical carrier level ~~N~~ signal (OC-N). Table also includes ITU nomenclature for the SDH, which refers to signals as synchronous transport modules ~~N~~ (STM-N). Because common applications of the ITU signal hierarchy cannot efficiently use a 51.84-Mbps signal, the lowest level STM signal is a 155.52Mbps (STS-3c) signal.

Although the SONET specification is primarily concerned with OC-N interconnect standards, STS-1 and STS-3 electrical signals within the SONET hierarchy are useful within a switching office for interconnecting network elements (e.g., multiplexers, switching machines, and cross-connect systems).

SONET Multiplexing Overview

The first step in the SONET multiplexing process involves generation of a 51.840-Mbps STS-1 signal for each tributary. The STS-1 signal contains the tributary (payload) traffic plus transport overhead. As indicated in the figure, a variety of tributary types are accommodated:

1. A single DS3 per STS-1 that can be a standard asynchronous DS3 signal generated by an Ml3 or M23 multiplexer. Asynchronous DS3 inputs are passed transparently through the system to a DS3 output. Because this transparent option exists, any 44.736Mbps signal can be carried within the payload envelope.
2. A group of lower rate tributaries such as DS1, DS 1C, DS2, or El signals can be packed into the STS-1 payload,
3. A higher rate (wideband) signal can be packed into a multiple number of concatenated STS-1 signals. Prevalent examples of higher rate signals are 139.264Mbps fourth-level multiplexes of ITU or a


DS3
44.736 Mbps
Packing & Transport Overhead Insertion
STS-1
51.840 Mbps
Section BIP-8 Calculation
DS1
DS1C
DS2
CEPT-1
Packing & Transport Overhead Insertion
STS-1
51.840 Mbps
Byte Interleaved Mux
STS-N
Scrambler
E/O
OC-N
CEPT-4
139.264 Mbps
Packing & Transport Overhead Insertion
STS-1
STS-1
STS-1
STS-3C
BISDN
150 Mbps
Packing & Transport Overhead Insertion
STS-1
STS-1
STS-1

broadband ISDN signal at 150 Mbps. Each of these applications requires three STS-1 signals concatenated together to form an STS-3C signal. Higher levels of concatenation (to form STS-NC signals) are possible for higher rate tributaries. Concatenated STS-1 signals contain internal control bytes that identify the signal as a component of a higher speed channel so the integrity of the concatenated data can be maintained as it passes through a network.

Functional block diagram of SONET multiplexing.

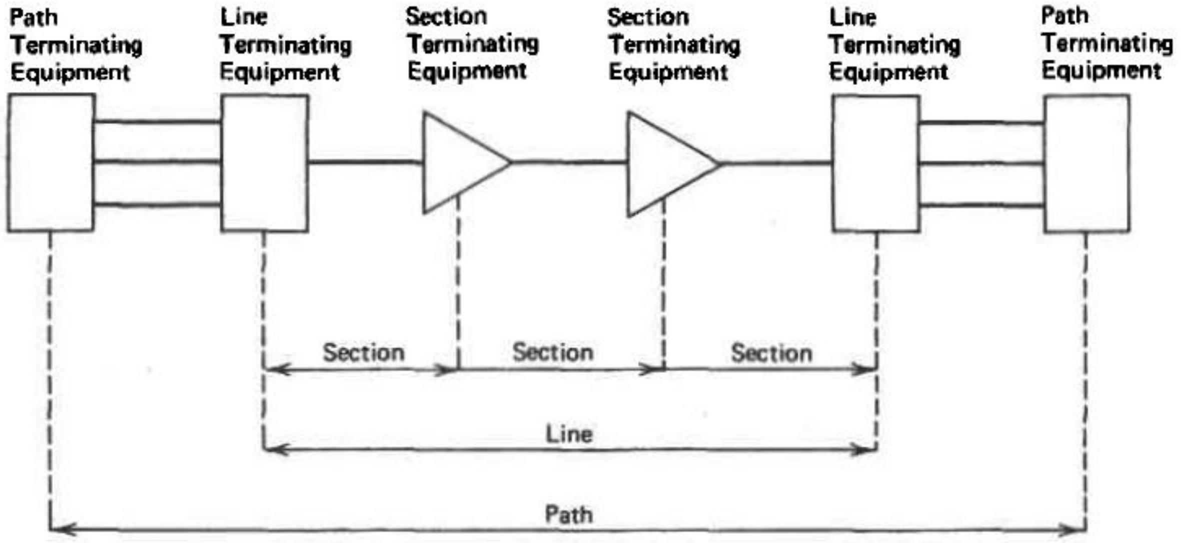
An STS-N signal is created by interleaving bytes from ~~*N*~~ STS-1 signals that are mutually synchronized. All timing (frequency) adjustment is done in generating each of the individual STS-1 signals. STS-1 signals that originate in another SONET node with a possibly different frequency are rate adjusted with the equivalent of byte stuffing (described later) to become synchronized to the clock of the local node. No matter what the nature of the tributary traffic is, all STS-1 tributaries in a STS-N signal have the same high-level format and data rate.

Optical carrier level-IV signals are generated by first scrambling the STS-N signal (except for framing bytes and STS-ID bytes) and then converting the electrical signal to an optical signal. Other than scrambling, the OC-N signal is generated with direct conversion to an optical signal. Thus, the data rates, formats, and framing of the STS- N and OC-N signals are identical.

A SONET system is defined as a hierarchy of three levels—sections, lines, and paths as indicated in Figure. Each ~~of~~ these levels has overhead bandwidth dedicated to administering and maintaining the respective level. As indicated in above Figure, one of the overhead functions provided within an STS-N signal involves calculation and transmission of a parity byte for the entire STS-N signal. Parity is also defined for the other levels of the architecture as described in the following section.

SONET Frame Formats

The frame format of an STS-1 signal is shown in Figure. As indicated, each frame consists of 9 rows of 90 bytes each. The first 3 bytes of each row are allocated to transport overhead with the balance available for path overhead and payload mapping.


Path Terminating Equipment
Line Terminating Equipment
Section Terminating Equipment
Section Terminating Equipment
Line Terminating Equipment
Path Terminating Equipment
Section
Section
Section
Line
Path

SONET SYSTEM HIERARCHY

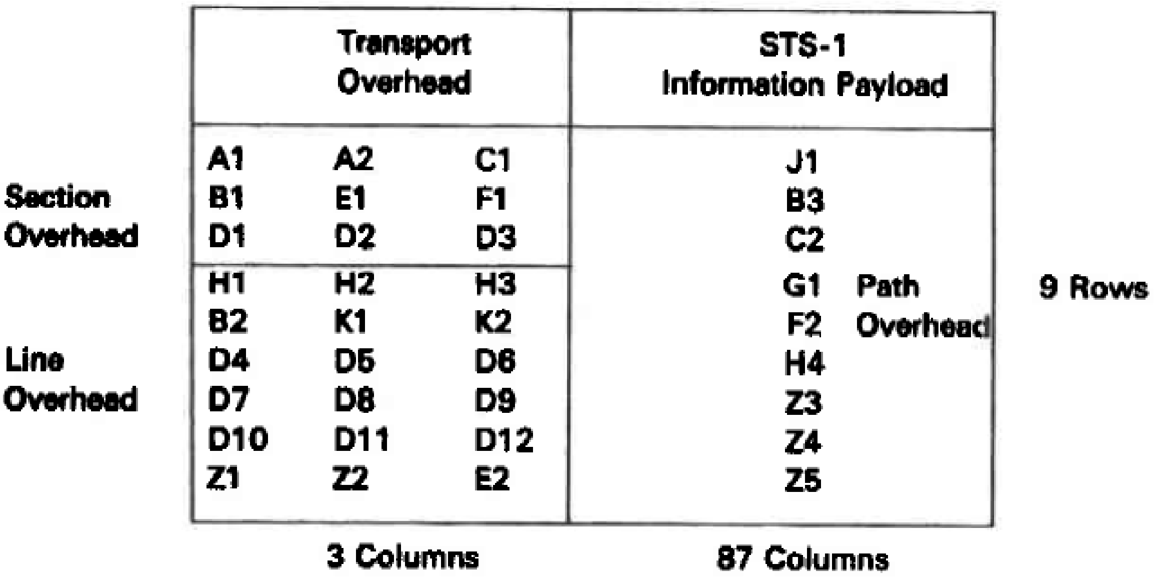
The transport overhead is itself composed of section overhead and line overhead. Path overhead is contained within the information payload as indicated.

The 9 rows of 87 bytes (783 bytes in all) in the information payload block is referred to as the envelope capacity. Because the frame rate is 8 kHz, the composite data rate of each STS-1 signal can be represented as the sum of the transport overhead rate and the information envelope capacity:

STS-1 rate = overhead rate + information envelope rate = 9 x 3 x 8 x 8000
+ 9 x 87 x 8 x 8000 = 1.728 x 10^6 + 50.112 x 10^6
= 51.840 Mbps

The internal format of the envelope capacity is dependent on the type of tributary traffic being carried. One aspect of the envelope format that is common to all types of traffic is the 9 bytes of path overhead indicated in Figure 8.13. The actual location and purpose of this overhead are described in the next two sections.

STS-1 frame format.

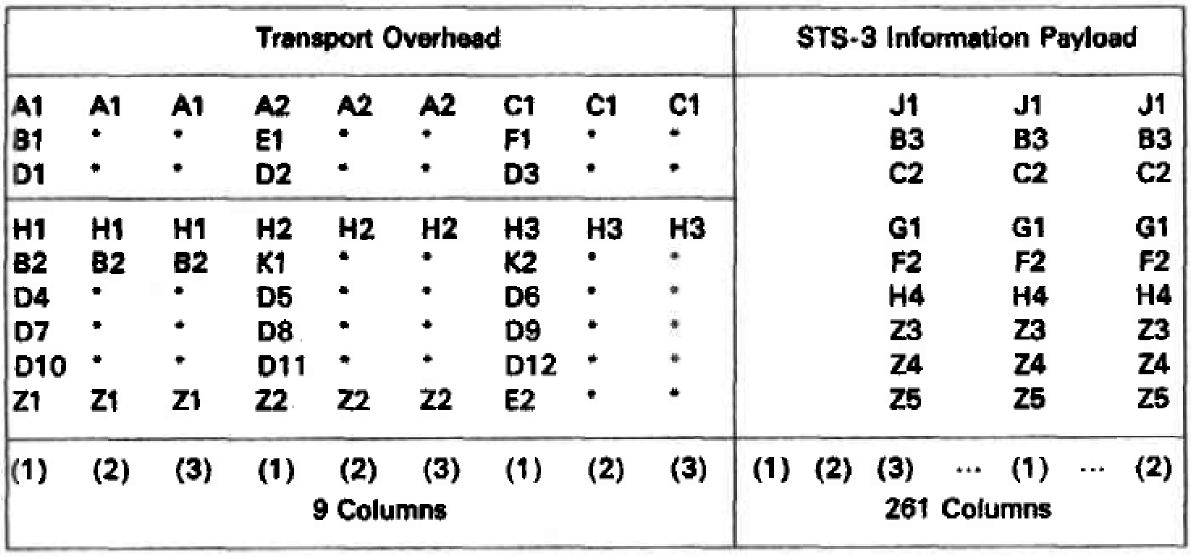


	Transport Overhead			STS-1 Information Payload		
Section Overhead	A1	A2	C1	J1		
	B1	E1	F1	B3		
	D1	D2	D3	C2		
Line Overhead	H1	H2	H3	G1	Path Overhead	9 Rows
	B2	K1	K2	F2		
	D4	D5	D6	H4		
	D7	D8	D9	Z3		
	D10	D11	D12	Z4		
	Z1	Z2	E2	Z5		
	3 Columns			87 Columns		

As a specific example of a higher level (STS-N) signal, Figure 8.14 depicts the details of an STS-3 signal that also represents the STM-1 signal format in ITU terminology. Transmission of the bytes occurs row by row and left to right. Thus, the first 3 bytes of an STS-3 frame are the three framing bytes Al, A1, Al. Most of the section and line overhead functions within an STS-3 signal are carried in the STS-1 number 1 overhead. Thus many of the corresponding bytes of the other STS-1 signals are unused and are so designated with an asterisk. Notice, however, that path overhead is included in the information envelope for each of the STS-1 signals.

After a frame of an STS-N signal is scrambled, a parity byte (BIP-8) is generated that provides even parity over corresponding bits in all bytes of the STS-N frame. This parity byte is inserted into the section overhead of the first STS-1 signal of the next STS-N frame

STS-3 frame format.



Transport Overhead									STS-3 Information Payload		
A1	A1	A1	A2	A2	A2	C1	C1	C1	J1	J1	J1
B1	*	*	E1	*	*	F1	*	*	B3	B3	B3
D1	*	*	D2	*	*	D3	*	*	C2	C2	C2
H1	H1	H1	H2	H2	H2	H3	H3	H3	G1	G1	G1
B2	B2	B2	K1	*	*	K2	*	*	F2	F2	F2
D4	*	*	D5	*	*	D6	*	*	H4	H4	H4
D7	*	*	D8	*	*	D9	*	*	Z3	Z3	Z3
D10	*	*	D11	*	*	D12	*	*	Z4	Z4	Z4
Z1	Z1	Z1	Z2	Z2	Z2	E2	*	*	Z5	Z5	Z5
(1)	(2)	(3)	(1)	(2)	(3)	(1)	(2)	(3)	(1) (2) (3)	··· (1)	··· (2)
9 Columns									261 Columns		

SONET Operations, Administration, and Maintenance

The SONET standard places significant emphasis on the need for operations, administration, and maintenance (OAM) of an end-to-end system. As shown in Figure 8.15, the OAM architecture is based on the section, line, and path layers described previously. OAM standardization is a requirement for mixing equipment from multiple vendors and ease of management of all levels of a system (an individual repeater section or an end-to-end path).

Section Overhead

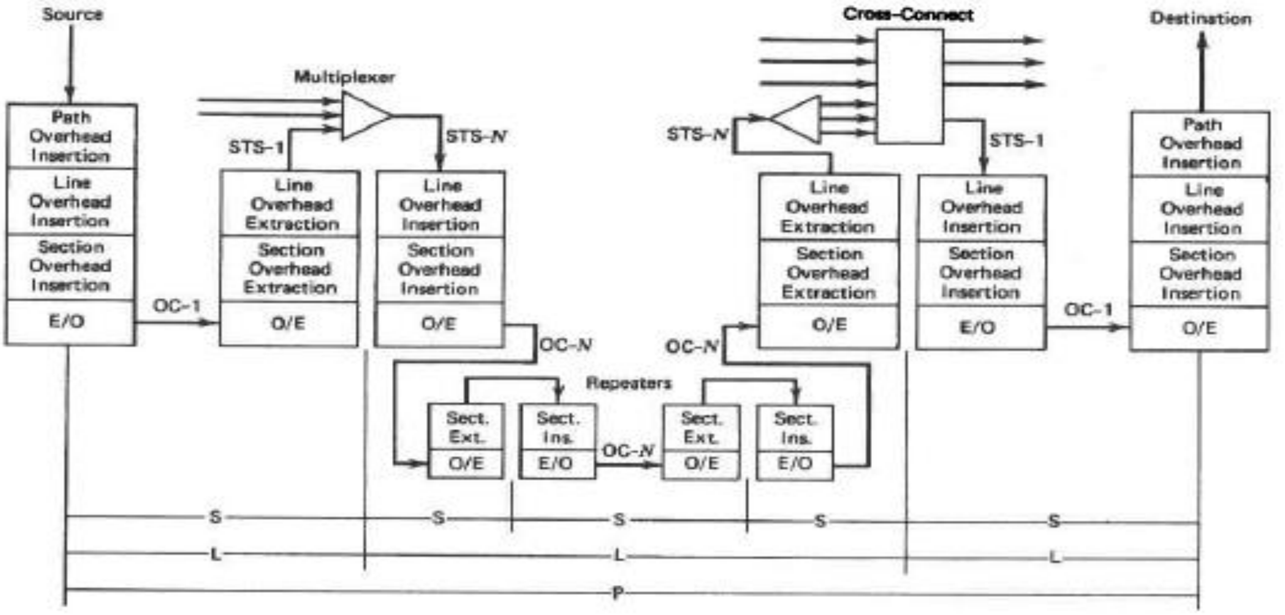
The functional allocation of the 9 bytes of section overhead in each STS-1 frame shown in Figure are:

- A1 Framing byte = F6 hex (11110110)
- A2 Framing byte = 28 hex (00101000)
- C1 STS-1 ID identifies the STS-1 number (1,…, N) for each STS-1 within an STS-N

multiplex

- B1 Bit-interleaved parity byte providing even parity over previous STS-N frame after scrambling
- El Section-level 64-kbps PCM orderwire (local orderwire)
- FI A 64-kbps channel set aside for user purposes
- D1-D3 An 192-kbps data communications channel for alarms, maintenance, control, and administration between sections

The fact that there is such a richness of maintenance support at the section level (from one repeater to another) is indicative of the recognized need for extensive OAM facilities and the availability of economical technology to provide it.



SONET overhead layers.

Line Overhead

The functional allocation of the 18 bytes of line overhead in each STS-1 frame shown in Figure are as follows:

- H1-H3 Pointer bytes used in frame alignment and frequency adjustment of payload data.
- B2 Bit-interleaved parity for line-level error monitoring
- Kl, K2 Two bytes allocated for signaling between line-level automatic protection switching equipment
- D4-D12 A 76-kbps data communications channel for alarms, maintenance, control, monitoring, and administration at the line level
- Z1,Z2 Reserved for future use

- E2 A 64-kbps PCM voice channel for line-level orderwire

Notice that the line-level OAM facilities are similar to those available at the section level with the addition of the protection switching signaling channel and HI, H2, and H3 pointer bytes use for payload framing and frequency adjustment.

Path Overhead

There are 9 bytes of path overhead included in every block (9 x 87 bytes) of information payload. The important aspect of this overhead is that it is inserted when the tributary data are packed into the synchronous payload envelope (SPE) and not removed (processed) until the tributary data are unpacked. Thus, it provides end-to-end OAM support independent of the path through the synchronous network, which may involve numerous intermediate multiplexers, cross-connect switches, or add-drop multiplexers. The exact location of these 9 bytes within the payload envelope is dependent on pointer values defined in the next section. The functions of the path overhead bytes are:

- J1 A 64-kbps channel used to repetitively send a 64-byte fixed-length string so a receiving terminal can continuously verily the integrity of a path; the contents of the message are user programmable
- B3 Bit-interleaved parity at the path level
- C2 STS path signal label to designate equipped versus unequipped STS signals and, for equipped signals, the specific STS payload mapping that might be needed in receiving terminals to interpret the payloads
- G1 Status byte sent from path-terminating equipment back to path-originating equipment to convey status of terminating equipment and path error performance (received BIP error counts)
- F2 A 64-kbps channel for path user
- H4 Multiframe indicator for payloads needing frames that are longer than a single STS frame; multiframe indicators are used when packing lower rate channels (virtual tributaries) into the SPE
- Z3-Z5 Reserved for future use

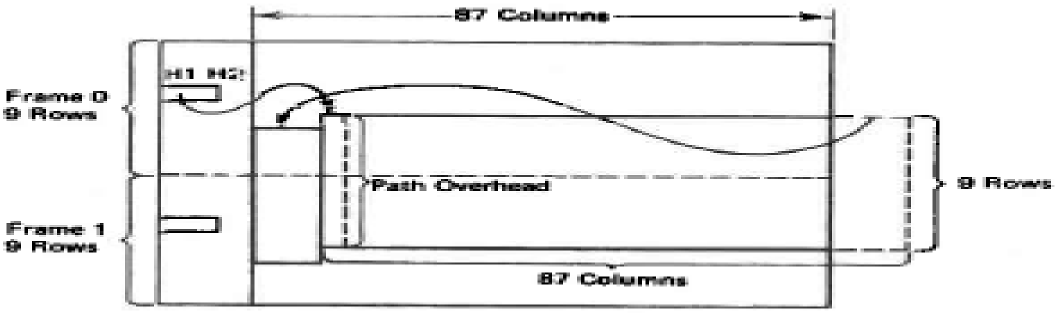
Payload Framing and Frequency Justification

Payload Framing

The location of the 9 bytes of path overhead in the STS-1 envelope is not defined in terms of the STS-1 transport framing. Instead, the path overhead is considered to be the first column of a frame of data referred to as the SPE, which can begin in any byte position within the STS-1 payload envelope (see Figure 8.16). The exact location of the beginning of the SPE (byte J1 of the path overhead) is specified by a pointer in bytes HI and H2 of the STS line overhead. Notice that this means that an SPE typically overlaps two STS-1 frames.

The use of a pointer to define the location of the SPE frame location provides two significant features. First, SPE frames do not have to be aligned with higher level multiplex frames. It may be that when first generated, an SPE is aligned with the line overhead at the originating node (i.e., the pointer value is 0). As the frame is carried through a network, however, it arrives at intermediate nodes (e.g., multiplexers or cross connects) having an arbitrary phase with respect to the outgoing transport framing. If the SPE had to be frame aligned with the outgoing signal, a full SPE frame of storage and delay would be necessary. Thus, the avoidance of frame alignment allows SPEs on incoming links to be immediately relayed to outgoing links without artificial delay. The location of the SPE in the outgoing payload envelope is specified by setting the HI, H2 pointer to the proper value (0-782).

The second advantage of the pointer approach to framing SPE signals is realized when direct access to subchannels such as DSls is desired. Because the pointer provides immediate access to the start of an SPE frame, any other position or time slot within the SPE is also immediately accessible. If the tributary uses a byte-synchronous mapping format, individual channel bytes have fixed positions with respect to the start of the SPE. This capability should be compared to the procedures required to demultiplex a DS3 signal. In a DS3 signal there is no relationship between the higher level framing and the lower level DS2 and DS1 framing positions. In essence, two more frame recovery processes are needed to identify a DSO time slot. The use of pointers in the SONET architecture eliminates the need for more than one frame recovery process when accessing byte-synchronous lower level signals.



Representative location of SPE.

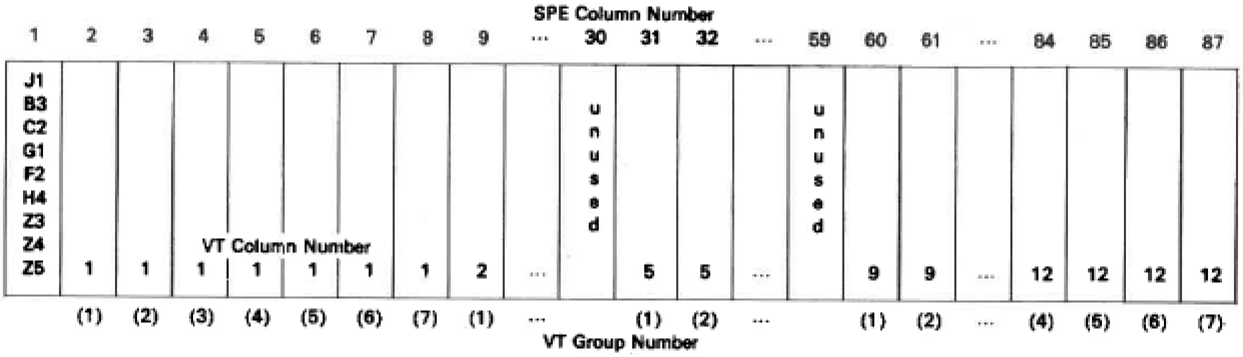
Frequency Justification

Although it is generally intended that SONET equipment be synchronized to each other or to a common clock, allowances must be made for the interworking of SONET equipment that operates with slightly different clocks. Frequency offsets imply that an SPE may be generated with one clock rate but be carried by a SONET transport running at a different rate. The means of accommodating a frequency offset is to accept variable SPE frame rates using dynamic adjustments in the SPE pointers. Pointer adjustments allow SPE frames to float with respect to the transport overhead to maintain a nominal level of storage in interface elastic stores. Figure shows the basic means of accommodating a slow incoming SPE. If the elastic store begins to empty, positive byte stuffing is invoked to skip one information time slot (the slot immediately following the H3 byte) and simultaneously incrementing the pointer to delay the SPE frame by one byte.

Virtual Tributaries

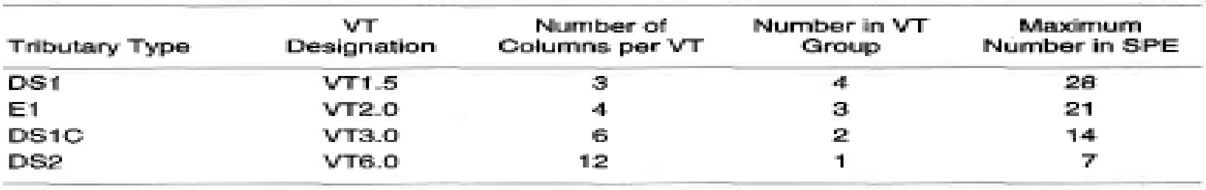
To facilitate the transport of lower rate digital signals, the SONET standard uses sub- STS-1 payload mappings referred to as virtual tributary (VT) structures, as shown in Figure 8.19. This mapping divides the SPE frame into seven equal-sized subframes or VT blocks with 12 columns (108 bytes) in each. Thus, the subframes account for

7 x 12 = 84 columns with the path overhead and two unused columns (reserved bytes R) accounting for the remainder of the 87 columns in an SPE. The rate of each VT structure is determined as 108 x 8 x 8000 = 6.912 Mbps.



SPE mapping for virtual tributaries

SONET Virtual Tributaries



Tributary Type	VT Designation	Number of Columns per VT	Number in VT Group	Maximum Number in SPE
DS1	VT1.5	3	4	28
E1	VT2.0	4	3	21
DS1C	VT3.0	6	2	14
DS2	VT6.0	12	1	7

The VT structures can be individually assigned to carry one of four types of signals. Depending on the data rate of a particular signal, more than one signal may be carried within a VT structure as a VT group. All signals within a VT group must be of the same type, but VT groups within a single SPE can be different types. The particular lower rate signals accommodated as VTs are listed in

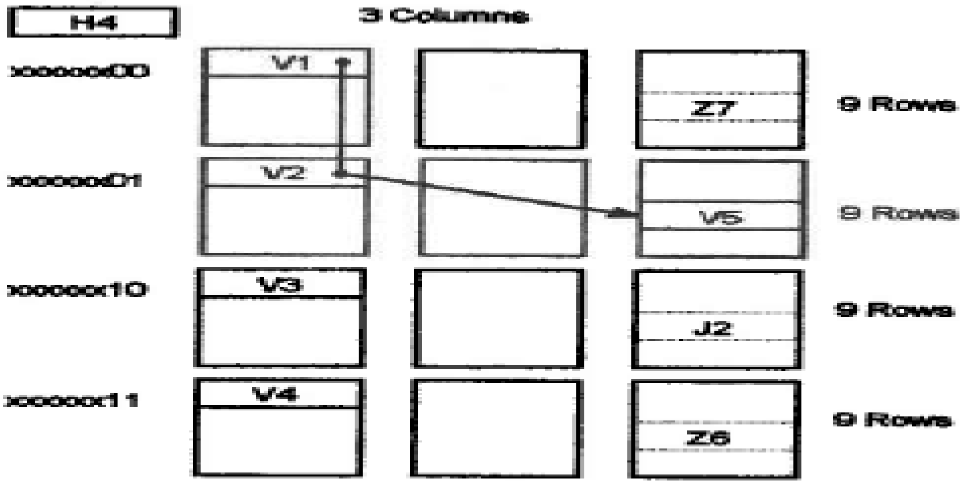
Table. The last column indicates how many of the lower rate signals are carried in a single SPE if all seven VT groups are the same type.

VT-SPE payloads are allowed to float within an STS-1 SPE in the same fashion as pointers to SPE payloads are allowed to float at the STS-1 level. Thus, a second level of pointer logic is defined for VT payloads. Again, a floating VT-SPE allows for minimal framing delays at intermediate nodes and for frequency justification of VT-SPEs undergoing transitions between timing boundaries. High-rate VT-SPEs are accommodated by inserting an information byte into V3 while slow-rate VT-SPEs are accommodated by stuffing into the information byte immediately following V3 when necessary.

Each VT1.5 uses three columns of data to establish 108 bytes in a VT1.5 payload. There are four such payloads in a 12-column VT group. The VI, V2, V3, V4 bytes of the payload have fixed positions within the STS-1 payload. The remaining 104 bytes of the VT 1.5 signal constitute the VT1.5 payload, the start of which is the V5 byte pointed to by VI and V2.

Asynchronous Mapping

The DS1 bit stream is inserted into the information bits (I) with no relationship to the VT-SPE frame or byte boundaries. As indicated, there are two stuffing opportunities (Si and ~~S_2)~~ available in every four-frame superframe. Thus, the VT1.5 superframe carries 771,772, or 773 information bits depending on the value of the stuff control bits C_1 and C_2. The nominal number of information bits in each frame is 193 x 4 = 772. Nominal frames carry information in S_2 while stuffing in Sj.



Super frame structure for VTI.5 tributaries

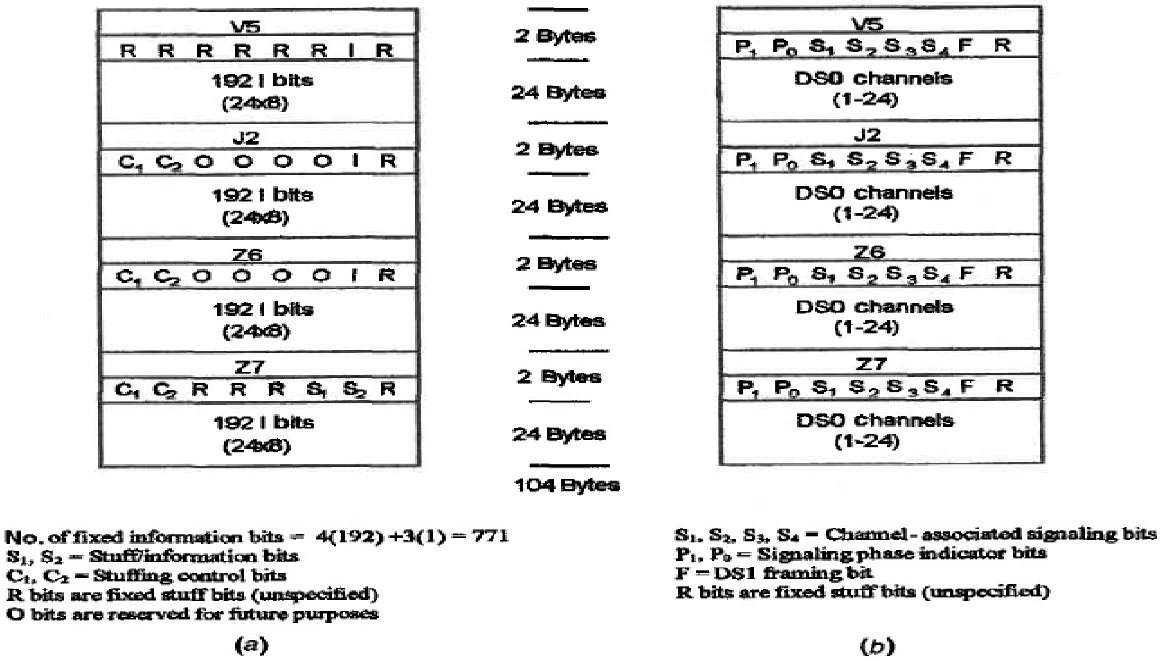
- **A VTl uses three columns of an SPE far 108 bytes in a 500 user, super frame.**
- **VI, V2, V3, V4 bytes have fixed locations in a SEE identified by the but two bits ofH4.**
- **VI and V2 point to V5 which is the first byte of the floating VT1.5 SPE.**
- **SPE overhead bytes V5, J2, Z6, Z7 occur in identical relative positions of an SPE.**

Because the asynchronous operation is compatible with the asynchronous network, it is the format used in most SONET applications. The major advantage of the asynchronous mode of operation is that it provides for totally transparent transmission of the tributary signal in terms of information and in terms of information rate. The major disadvantage of the asynchronous mode is that 64-kbps **DSO** channels and **signaling** bits are not readily extracted.

Byte-Synchronous Multiplexing

In contrast to the asynchronous mapping, the byte-synchronous payload mapping allocates specific bytes of the payload to specific bytes (channels) of the DS 1 tributary signal. Hence, this mode of operation overcomes the main drawback of the asynchronous mode in that 64-kbps DSO channels and signaling bits within the payload are easily identified. In fact, when the DS1 tributary arises from legacy applications, the signaling bits of a DS1 are moved from the least significant bit (LSB) of every sixth frame of respective channels and placed in dedicated signaling bit positions within the VT-SPE. Thus byte-synchronous multiplexing offers an additional feature of converting from in-slot signaling to out-slot signaling for DS1 signals.

An important aspect of the byte-synchronous format is the absence of timing adjustments for the source DS 1 signal. Thus, the DS 1 interface necessarily requires a slip buffer to accommodate a DS1 source that may be unsynchronized to the local SONET clock. Although slips in byte synchronously mapped DS 1

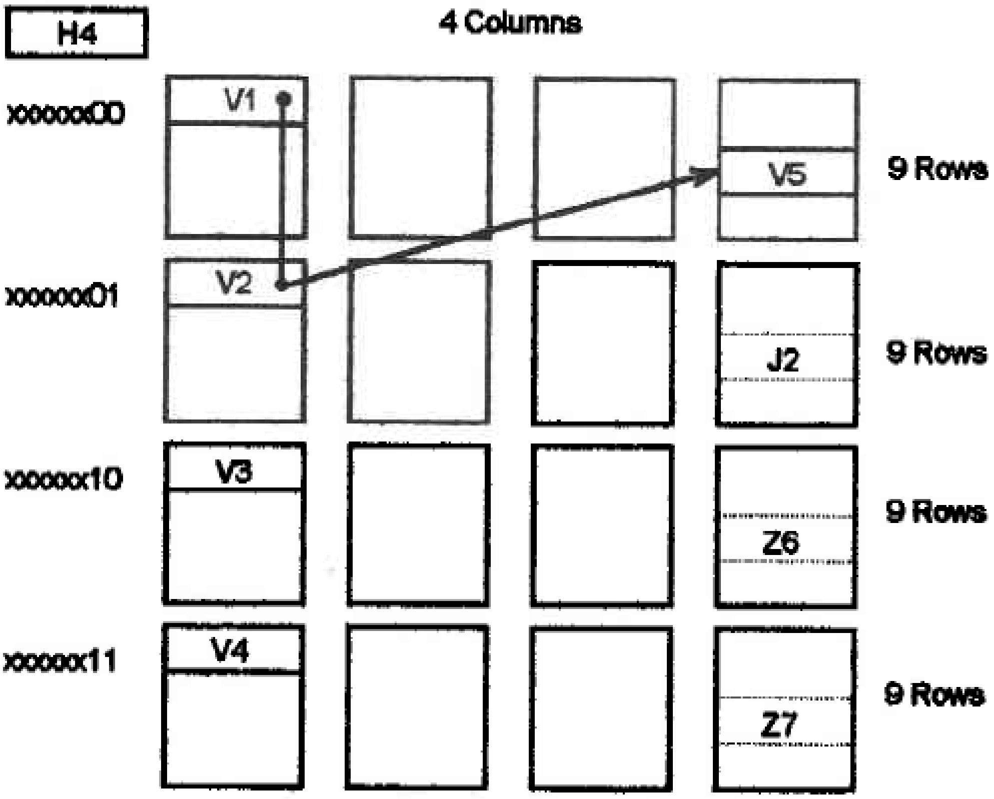


DS1 mappings in VT1.5 SPE: (a) asynchronous; (*b*) byte synchronous.

signals may occur at the SONET network interface (e.g., SONET gateway), slips cannot occur within the SONET network because internal nodes rate adjust the VT1.5 payloads with pointer adjustments.

E1 Mappings

El signals are mapped into VT2 signals with the same basic procedures used for DS1 s. As shown in Figure, the VT2 signal is composed of four columns of bytes in an STS-1 that produce a total of 144 bytes. After removing the VI, V2, V3, and V4 bytes, the VT2 payload has 140 bytes. Formats for asynchronously mapped Els and byte synchronously mapped Els. Notice that the byte-synchronous mapping for a 30-channel El carries channel-associated signaling in slot 16-the form of out-slot signaling designed into El signals at their inception. The same basic format supports common-channel signaling, which is sometimes referred to as a’



Super frame structure for VT2 tributaries

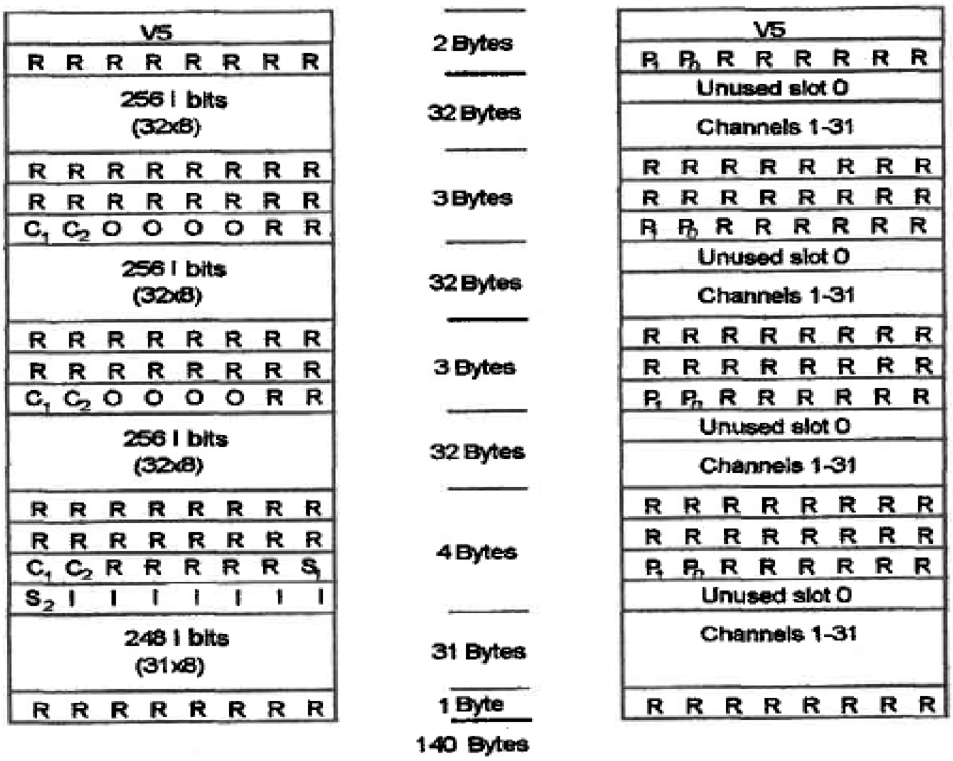
- **A VT2 uses four columns of an SPE for 144 bytes in a 500 μsec, superframe.**
- **VI, V2, V3, V4 bytes have fixed locations in a SPE identical by die last two bits of H4.**
- **VI and V2 point to V5 which is the first byte of the floating VT2 SPE.**
- **SPE overload bytes V5, J2, Z6, Z7 occur in identical relative positions of an SPE.**

.31-channel El format. In this case channel 16 is the CCS channel and channels 1-15 and 17-31 are the bearer channels. Thus, the multiplex mapping is not changed, just the nomenclature of the channels and the SPE type designation in the VT path overhead byte V5.

DS3 Payload Mapping

The previous section describes several alternatives for packing virtual tributaries into an STS-1

envelope. When all seven VTs in an envelope are VT1.5s, a total capacity of 28 DSls is provided—the same as a DS3 signal. Thus one method of carrying a DS3 signal involves demultiplexing it into its constituent DS1 (or DS2 signals) and packing the constituents as virtual tributaries. This approach is attractive in that the virtual tributaries are individually accessible for cross-connect or add-drop multiplexer systems. If the application does not need to access the individual tributaries, it is simpler to pack the DS3 signal directly into an STS-1, as indicated in Figure 8.24. The payload mapping in Figure 8.24 treats the DS3 signal simply as a 44.736-Mbps data stream with no implied internal structure. Thus, this
mapping provides transparent transport of DS3-rate data streams.



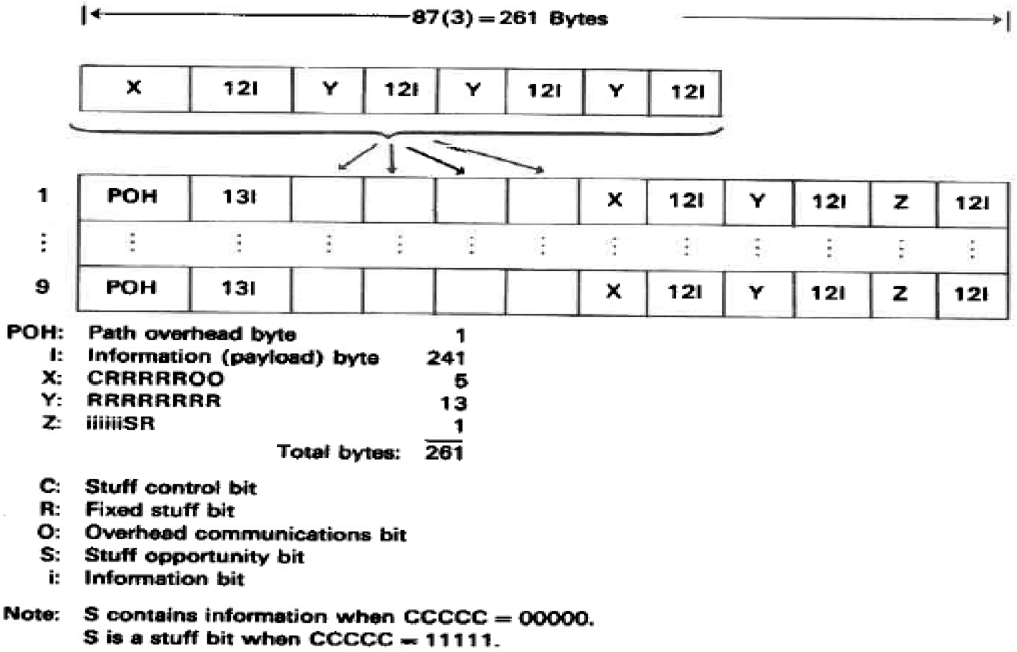
No. of fixed information bits = 4(256) -1 = 1023
S_1, S_2 = Stuff information bits
C_1, C_2 = Stuffing control bits
R bits are fixed stuff bits (unspecified)
O bits are reserved for future purposes

CAS bits in channel 16
P_1, P_0 = Signaling phase indicator bits
R bits are fixed stuff bits (unspecified)

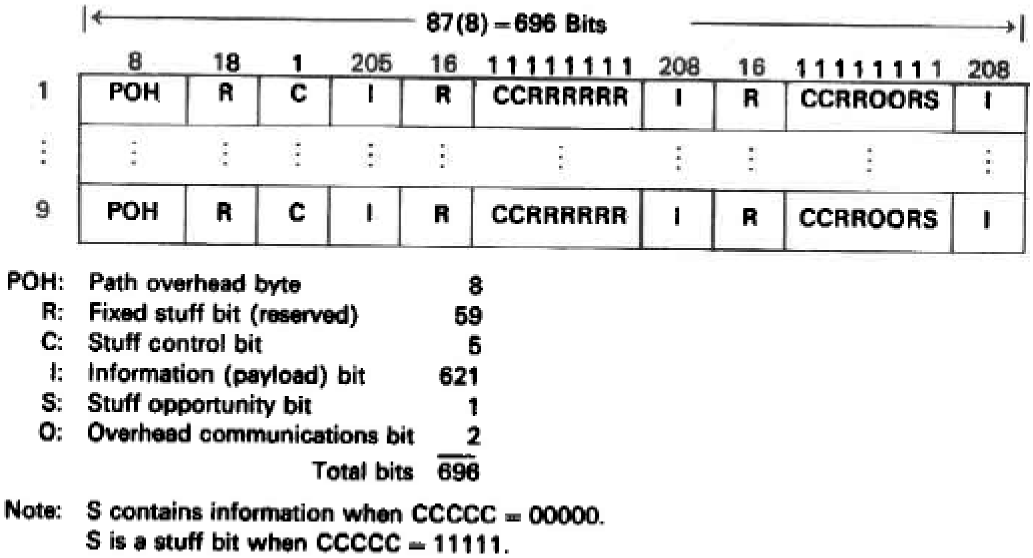
El mappings in VT2 SPE: (a) synchronous; *(b)* byte synchronous.

Each row of a nine-row SPE envelope contains 87 x 8 = 696 bits, which can carry 621 or 622 DS3 data bits depending on the value of the C bits. Notice that this format has five C bits, which allows for single and double bit error correction. The path overhead (POH) bytes carry the 9 bytes of POH

E4 Payload Mapping



Asynchronous 44.736-Mbps (DS3) payload mapping



87(8) = 696 Bits
8
18
1
205
16
1 1 1 1 1 1 1 1
208
16
1 1 1 1 1 1 1 1
208
1
POH
R
C
I
R
CCRRRRRR
I
R
CCRROORS
I
9
POH
R
C
I
R
CCRRRRRR
I
R
CCRROORS
I
POH: Path overhead byte 8
R: Fixed stuff bit (reserved) 59
C: Stuff control bit 5
I: Information (payload) bit 621
S: Stuff opportunity bit 1
O: Overhead communications bit 2
Total bits 696
Note: S contains information when CCCCC = 00000.
S is a stuff bit when CCCCC = 11111.

Asynchronous 139.264-Mbps (E4) payload mapping.

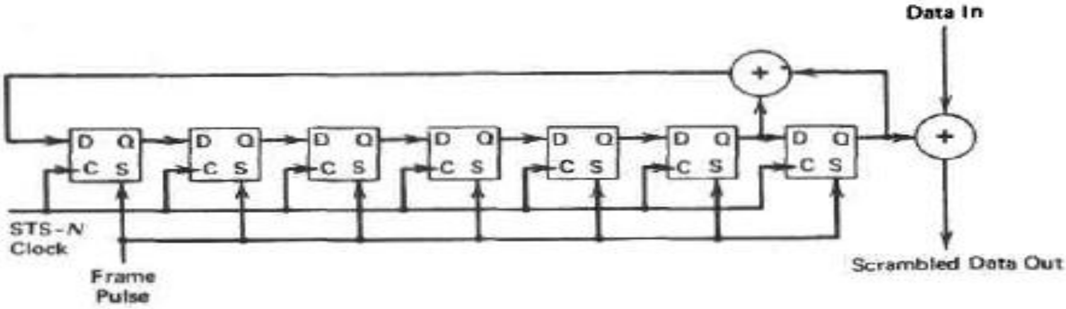
One example of a SONET super rate mapping is shown in Figure for a 139.264Mbps fourth-level ITU-T signal (E4), This signal is packed into a 155.52-Mbps STS-3c (or STM-1) signal. Figure shows only the synchronous payload envelope .(SPE-3c), not the 9 bytes of section and line overhead in each row. Notice that there is only one column of POH within the SPE-3c envelope. The POH bytes carry the 9 bytes of overhead.

The payload mapping in Figure 8.25 treats the 139.264-Mbps signal as a transparent data stream with no implied internal structure. Each row of a nine-row SPE-3c envelope contains 87 x 3 = 261 bytes, which can carry 1934 or 1935 data bits depending on the value of the C bits. Notice that this format also has five C bits, which allows for single and double bit error correction.

SONET Optical Standards

The optical interface standard defined for “mid-span-meet” of SONET equipment allows for either NRZ or RZ line codes on single-mode fibers. Generation of the OC-N signal from the STS-N signal requires a scrambler as shown in Figure 8.26. The scrambler is synchronized to each STS-N frame by presetting the shift register to all l’s immediately after transmitting the last Cl byte of the STS-N section overhead. Thus, the frame codes (A1,A2) and STS-1 ID (Cl) code are not scrambled. A minimum level of timing content is assured by the Al, A2, and Cl bytes along with the static overhead bits of the STS-N frame that are anti-coincident with the scrambler sequence. Because the scrambler is preset at the same point of every frame, every bit position in successive frames experience the same scrambler value. Thus, when static overhead is “exclusive ored” with the scrambler, the same data values arise.

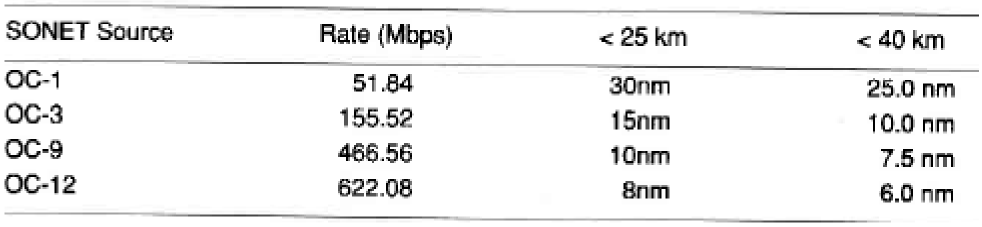
SONET scrambler.



The BER objective is 1 x $10^{“10}$ for optical sections of 40 km or less. Equipment from separate manufacturers can be freely interchanged for applications with distances up to 25 km. Longer distances may require joint engineering.

SONET systems are specified to operate with central wavelengths at 1310 nm with SMF fibers or at 1550 nm with DS-SMF fibers. Operation at 1310 nm with DS-SMF fibers or at 1550 nm with SMF fibers is not disallowed but must be jointly engineered. A range of laser wavelength tolerances and maximum allowable spectral widths is

Representative Maximum Spectra Widths of SONET Sources



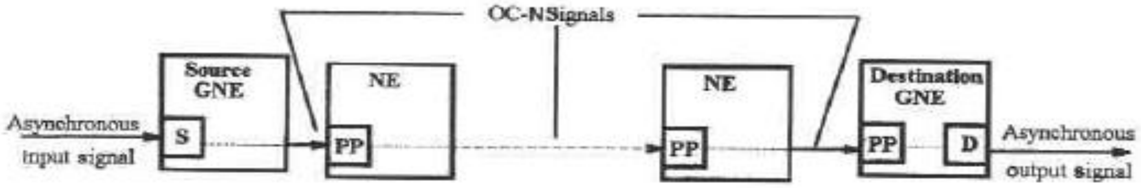
SONET Source	Rate (Mbps)	< 25 km	< 40 km
OC-1	51.84	30nm	25.0 nm
OC-3	155.52	15nm	10.0 nm
OC-9	466.56	10nm	7.5 nm
OC-12	622.08	8nm	6.0 nm

specified for both 1310 and 1550 nm. Table 8.10 provides representative values of the specifications.

SONET Networks

A basic block diagram of a SONET network is shown in Figure. Gateway network elements (GNEs) provide interfaces to external (asynchronous) digital signals. These signals are mapped (synchronized) and unmapped (desynchronized) by the gateway using the appropriate mapping format. At this point only bit stuffing is used to synchronize the asynchronous tributaries to SONET. No pointer adjustments occur in the GNE. As the STS-N signals propagate through the network, pointer adjustments in pointer processing (PP) interfaces may be applied at internal network elements (NEs), but the lower level interface mappings that occur at the GNEs are untouched. If a particular NE accesses VT payloads, VT payloads in the same VT group that pass through the node may experience VT pointer adjustments. Otherwise, V.T pointer adjustments do not occur (only the STS-1 level signals are rate adjusted). The following paragraphs summarize pointer processing aspects of a SONET network:

1. Pointer justification events (PJEs) never occur in an originating GNE.
2. A desynchronizer experiences continuous PJEs only as a result of a synchronization difference between the originating GNE and the terminating



SONET network elements: S, synchronizer; PP, pointer processor; D, desynchronizer

GNE. Synchronization differences/failures at internal nodes of a SONET network produce continuous pointer adjustments, but these get removed when the SPE passes through a node that is synchronized to the source GNE.

3. PJE bursts occur for two possible reasons. The first is a result of a reference switch and a subsequent phase adjustment of a node’s local clock to align it with the phase of the new reference. Bursts can also occur as a result of clock noise in multiple nodes producing near-simultaneous pointer adjustments. In order for all of these adjustments to propagate to a desynchronizing gateway, all of the elastic stores in the path must be at the appropriate threshold. This can only happen if the source GNE has previously produced some abnormal behavior such as a loss of a reference or sustained a rather large amount of wander.

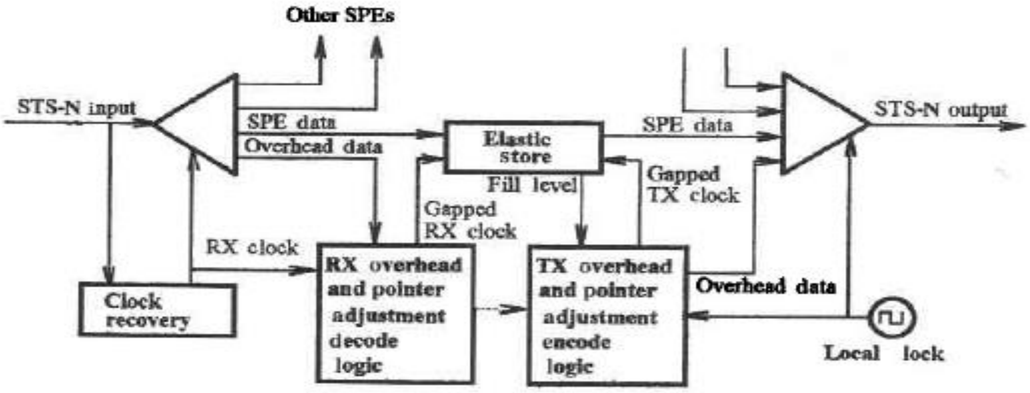
4. A pointer adjustment at the SPE level does not affect a VT signal unless it is passed to a node that accesses the VT and that particular adjustment happens to cause a pointer movement at the VT level. Even when this occurs, the VT pointer adjustment must pass through the network (without absorption) to the desynchronizing gateway to affect the outgoing tributary signal. On average, one of every 30 PJEs at the STS-1 level produces a PJE at the VT1.5 level.

A block diagram of an SPE synchronization circuit (PP) depicting two halves of pointer processing: one half extracts (desynchronizes) the SPE payload from a received signal and the other half synchronizes the SPE to the local STS-1 frame rate. The RX pointer processing block extracts the payload data from the received signal and passes it to the elastic store. The TX pointer processing block monitors the fill level of the elastic store and makes pointer adjustments to maintain a nominal level of storage. The size of the elastic store only needs to be on the order ofbytes in length, not a full frame. The ability to use a relatively small elastic store (as compared to frame-length elastic stores in the asynchronous network) is one of the features of a pointer-based synchronization architecture: The payloads are allowed to float with respect to the STS-1 frame boundaries.

Frequency of Pointer Justification Events

If all NEs of a SONET island use a timing reference that is traceable to a common primary reference source (PRS), PJEs occur only as a result of distribution-induced clock wander that produces no sustained frequency offset. Thus, when all NEs are synchronized to the same reference, PJEs occur at random times and have equal numbers of positive and negative values over the long run.

Continuous PJEs occur only when there is a reference failure at some NEs within a SONET island or the island is intentionally designed to operate in a plesiochronous mode. If the reference failure occurs at some internal node of the SONET island, the resulting PJEs are removed at the next node in the path that is still locked to the same reference as the gateway NE. Thus, a tributary desynchronizer at a GNE must deal.

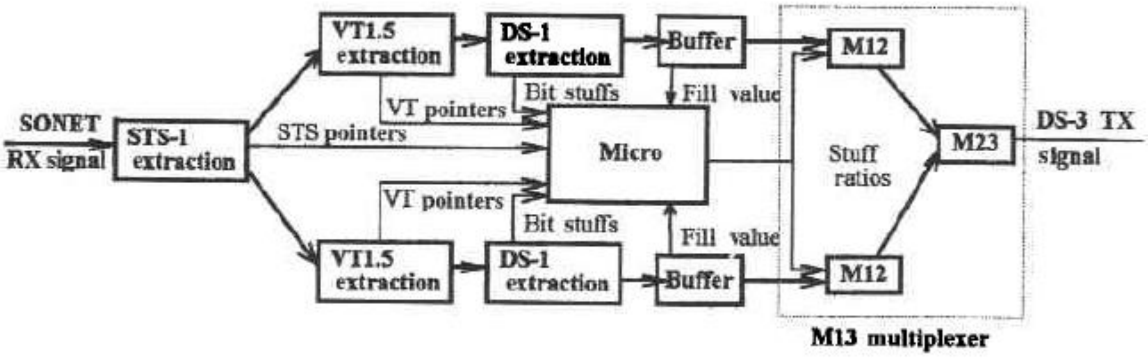


Block diagram of SPE synchronizing equipment RX, receiver, TX, transmitter

SONET Desynchronizers

SONET desynchronizers are necessarily designed with very low clock recovery band- widths to smooth the effects of (1) isolated pointer adjustments, (2) continuous pointer adjustments, (3) pointer adjustment bursts, or (4) combinations of the latter two. A pointer burst is defined as the occurrence of multiple pointer adjustments of one polarity occurring within the decay time of the desynchronizer circuit (i.e., the reciprocal of the desynchronizer closed-loop PLL bandwidth). Thus, it is ironic that as the clock recovery bandwidth is narrowed to smooth the effect of a burst, the probability of a burst occurrence is increased (by definition only). Extremely narrow PLL bandwidths are easiest to implement using digital filtering techniques commonly referred to as bit leaking. Bit leaking is essentially a mechanism for converting byte-sized pointer adjustments into bit- (or fractional-bit-) sized timing adjustments.

VTl.5 desynchronize hardware functional components



The microprocessor is used to perform long-term **averaging** of phase adjustments in lieu of dedicated logic that requires large counters and wide word sizes for low-bandwidth DSP filtering. The first function of the microprocessor is to determine the average DS1 payload frequency offset represented by all frequency adjustment events (bit stuffs, VT pointer adjustments, and STS pointer adjustments). After the average frequency adjustment is determined, a stuff ratio value is calculated that allows insertion into a DS3 signal as shown. (The M12 stage is embedded in the M13 multiplexer.) The elastic store fill level is used for very long term adjustments in the output frequency that arise from finite precision limits of the DSP calculations and for accommodating variations in the DS3 clock, which is typically not synchronized to the SONET line clock.

SONET RINGS

As has been mentioned earlier in this book, the development of large switching machines and transmission systems with extremely large cross sections has impacted telecommunications network architectures with a trend toward fewer hierarchical levels. An undesirable consequence of this trend is increased dependence on the operational status of individual switching machines and transmission paths. A SONET self-healing ring, or more simply a SONET ring, is a network architecture that specifically addresses network survivability. Two basic types of self-healing rings are shown in Figure: a unidirectional ring and a bidirectional ring. The main difference between the two types of rings is how the two directions of a duplex connection are established.

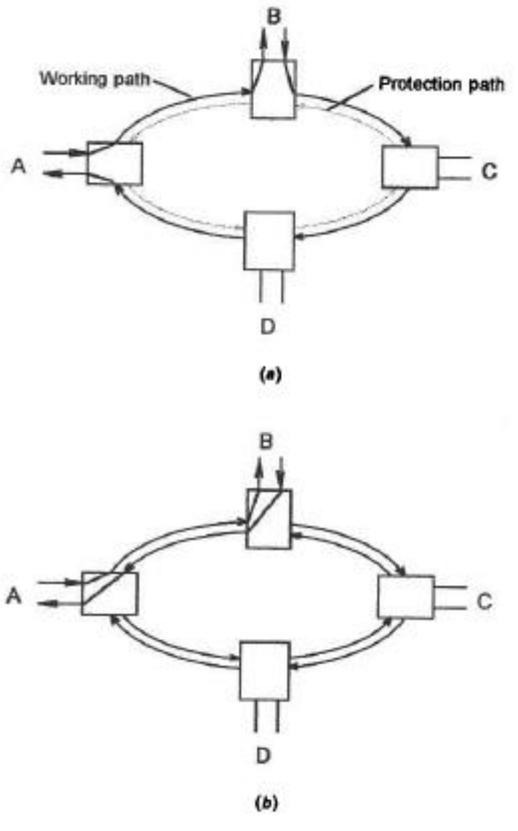
In a unidirectional ring a single time slot of the entire ring is assigned to both halves of a connection. As indicated in Figure, traffic is normally carried only on the (unidirectional) working path with the counterrotating path used for protection. In the example, an STS-1 (out of an OC-48) might be carried directly from A to B, but the returning STS-1 would be carried from B through C and D to A. A bidirectional ring, on the other hand, establishes both halves of the duplex connection over the shortest path in the ring. Thus, no fiber is identified as a pure working fiber and another as a pure protection fiber. Because bidirectional rings provide shorter round trip delays for most connections and allow reuse of time slots on the ring, it is the preferred mode of operation for interoffice networks. Rings for subscriber access applications do not carry much traffic between ADM nodes and therefore are more suited to a unidirectional mode of operation.

Unidirectional Path-Switched Ring

As shown in Figure , a unidirectional path-switched ring (UPSR) transmits the same information from A to B in both directions around the ring. Normally, only the working path is accessed by the

(b)

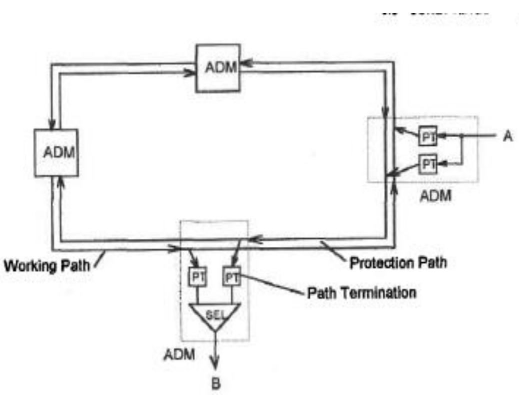
receiving node: If a failure occurs, a node can select the data on the protection channel. Notice that in the example shown selection of the protection path actually leads to a shorter path for the connection from A to B.



(a) Unidirectional and (b) bidirectional rings.

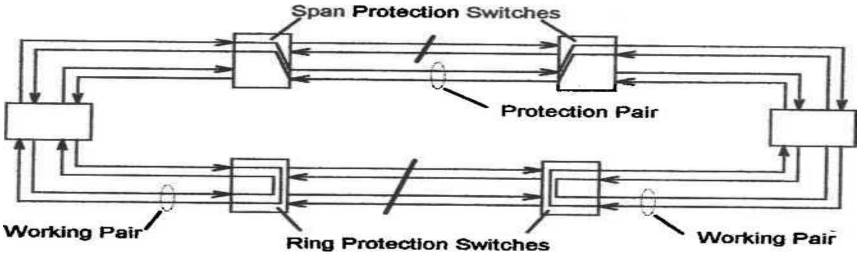
Bidirectional Line-Switched Ring

Bellcore defines two versions of bidirectional line-switched rings (BLSRs): a two-fiber BLSR and a four-fiber BLSR. On a two-fiber BLSR protection is provided by reserving bandwidth in each of two counter rotating fiber paths . If all traffic is to be protected, only 50% of the total system capacity can be used. Under normal conditions connections between two nodes utilize the shortest path between the nodes. If a fault in either direction of transmission occurs, the nodes adjacent to the fault perform ring switches as indicated. A ring switch involves switching traffic from working channels of the failed facility to spare channels of the other facility on the side of the node on which the fault occurs. The protection-switched traffic propagates all the way around the ring, being ignored by intervening nodes, until it is switched back to the working channels by the other node next to the fault. Notice that all nodes (including the nodes adjacent to the fault) communicate on working channels in the same manner as they did before the protection switching. That is, the path terminations are not part of the protection path. The main impact of the rotection switch is an increase in delay for affected traffic (and a momentary insertion of extraneous data when the switch occurs).

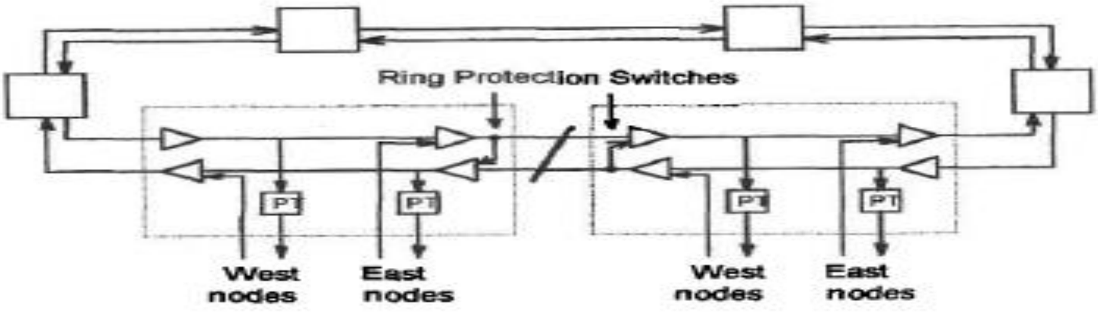


UPSR protection switching.

On a four-fiber BLSR two pairs of fibers are provided for each direction of transmission—one bidirectional working pair and another pair for protection of the first pair. Thus, working and protection channels are carried on different physical facilities. Again, connections are normally set up to use the shortest distance of travel for each side of a connection. If a failure occurs on only a working facility, protection switching occurs similar to “span switching” of a point-to-point system: The traffic is merely



Two-Fiber BLSR protection switches



Four-Fiber BLSR protections switches

switched to and from the protection facility by nodes adjacent to the fault. However, if a fault affects both the working and the protection facilities, a ring switch is needed as shown. Again, protection-switched traffic propagates all the way around the ring without being accessed by intervening nodes. All traffic accesses still occur on the working channels even though the same information is passing through the nodes in the protection path.

A four-fiber BLSR obviously requires more facilities than a two-fiber BLSR but has numerous advantages. First, the protected capacity of the system is twice as large. Second, fiber failures on only the working pair can be accommodated by a span switch with minimal disruption to traffic. Third, multiple separate failures can occur on working pairs and be accommodated by multiple span switches. Fourth, the presence of a spare pair simplifies maintenance testing and possible upgrading of facilities. For these reasons, a four-fiber BLSR is generally favored.

UNIT-II

DIGITAL SWITCHING

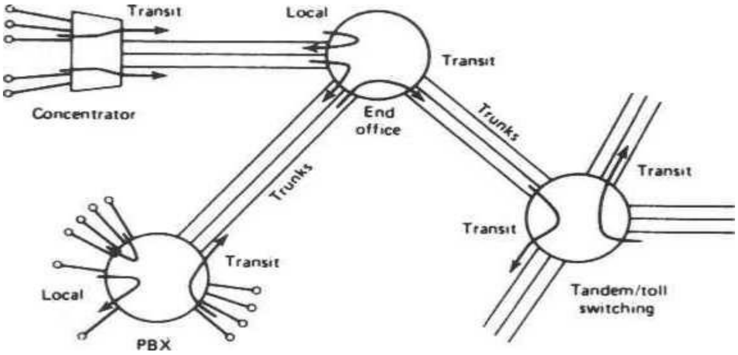
SWITCHING FUNCTIONS

Obviously, the basic function of any switch is to set up and release connections between transmission channels on an “as-needed basis.” The structure and operation of a switch depend on particular applications. Three switching categories for voice circuits are local (line-to-line) switching, transit (tandem) switching, and call distribution.

The most common switching function involves direct connections between subscriber loops at an end office or between station loops at a PBX. These connections inherently require setting up a path through the switch from the originating loop to a specific terminating loop. Each loop must be accessible to every other loop. This level of switching is sometimes referred to as line switching.

Transit connections require setting up a path from a specific incoming (originating) line to an outgoing line or trunk group. Normally, more than one outgoing circuit is acceptable. For example, a connection to an interoffice trunk group can use any one of the channels in the group. Hence transit switching structures can be simplified because alternatives exist as to which outgoing line is selected. Furthermore, it is not even necessary that every outgoing line be accessible from every incoming line. Transit switching functions are required by all switching machines in the telephone network. Some machines such as remote concentrators and toll or tandem switches service only transit traffic (e.g., do not provide local connections).

Call distributors are often implemented with the same basic equipment as PBXs. The mode of operation (software) is significantly different, however, in that incoming calls can be routed to any available attendant. Normally, the software of an automatic call distributor (ACD) is designed to evenly distribute the arriving calls among the attendants.



Local and transit traffic switching examples.

Although it is not an inherent requirement that every incoming line (trunk) be connectable to every attendant, call distributors are normally designed to provide accessibility to all attendants. Furthermore, it is often desirable that nonblocking operations be provided.

SPACE DIVISION SWITCHING

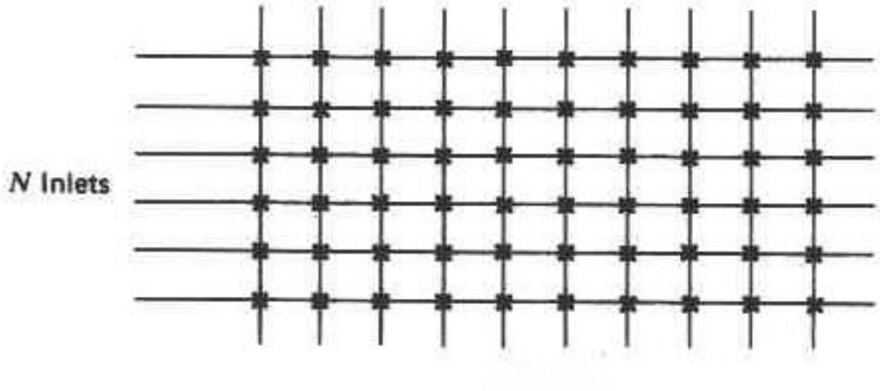
This switching matrix can be used to connect any one of *N*inlets to any one of *M* outlets. If the inlets and outlets are connected to two-wire circuits, only one crosspoint per connection is required.1

5.2 SPACE DIVISION SWITCHING 2

Rectangular crosspoint arrays are designed to provide intergroup (transit) connections only, that is, from an inlet group to an outlet group. Applications for this type of an operation occur in the following:

1. Remote concentrators
2. Call distributors
3. Portion of a PBX or end office switch that provides transit switching
4. Single stages in multiple-stage switches

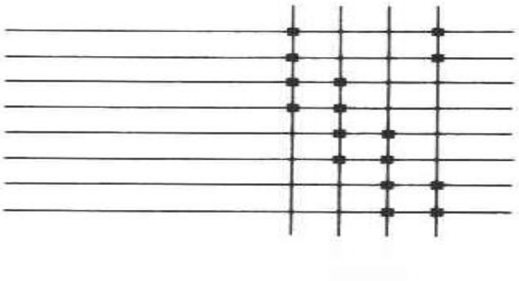
In most of the foregoing applications, it is not necessary that the inlets be connectable to every outlet. In situations involving large groups of outlets, considerable savings in total crosspoints can be achieved if each inlet can access only a limited number of outlets. When such a situation occurs, “limited availability” is said to exist. By overlapping the available outlet groups for various inlet groups, a technique called “grading” is established. Notice that if outlet connections are judiciously chosen, the adverse effect of limited availability is minimized. For example, if inlets 1 and 8 in Figure request a connection to the outlet group, outlets 1 and 3 should be chosen instead of outlets 1 and 4 to avoid future blocking for inlet 2.



Rectangular crosspoint array.

Graded switching structures were often used for access to large trunk groups in electromechanical switches where crosspoints were expensive and individual switching modules were limited in size. Gradings were also used in individual switching stages of large multiple-stage switches where more than one path to any particular outlet exists. Because very large digital matrices can be implemented with full accessibility, graded switch structures are no longer necessary.

Intragroup switching, as in line-to-line switching, requires each line to be connectable to every other line. Thus full availability from all inlets to all outlets of the switching matrix is required. The two matrix structures that can be used to fully interconnect two-wire lines. The dashed lines indicate that corresponding inlets and outlets of two-wire switching matrices are actually connected together to provide bidirectional transmission on two-wire circuits. For purposes of describing switching matrices, however, it is convenient to consider the inlets and outlets of two-wire switching matrices as being distinct.



Graded Rectangular Switching Matrix

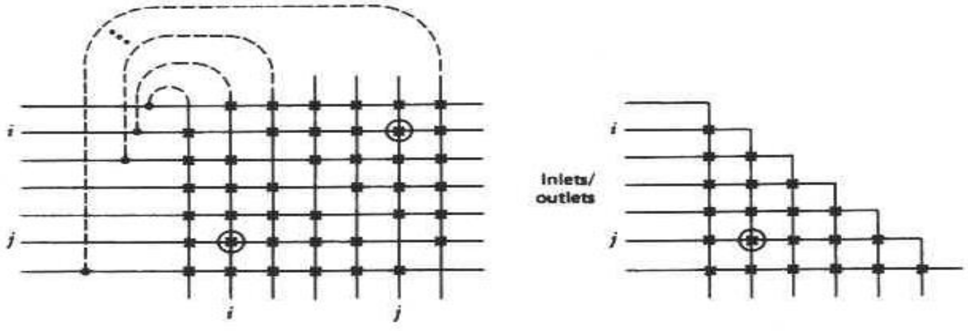
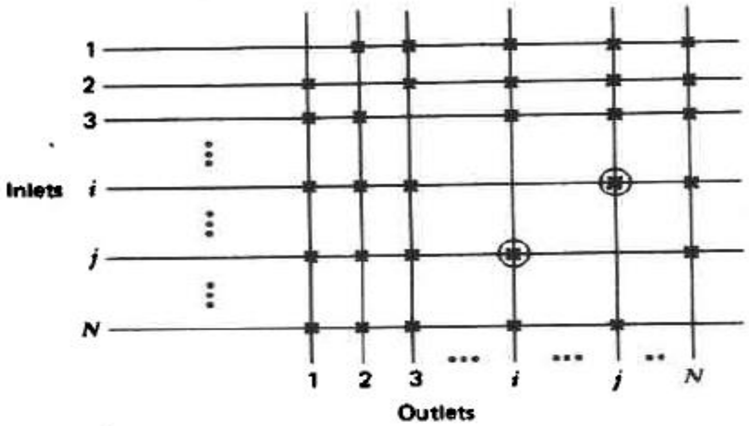


Figure Two-wire switching matrices: *(a)* square; *(b)* triangular (folded).

Both structures in Figure allow any connection to be established by selecting a single crosspoint. However, the square matrix, which is also called a two-sided matrix, allows any particular connection to be established in two ways. For example, if input link *i* is to be connected to input link *i*, the selected crosspoint can be at the intersection of inlet i and outlet *j*or at the intersection of inlet; and outlet *i*. For simplicity these crosspoints are referred to as *(i,j)* and *(j, i)*, respectively. In a typical implementation, crosspoint*(i,f)* is used when input *i* requests service, and crosspoint*(j, i)* is used when input *j* requests service.

In the triangular matrix the redundant crosspoints are eliminated. The crosspoint reduction does not come without complications, however. Before setting up a connection between switch input*i* and switch input i, the switch control element must determine which is larger: *i* or j. If *i* is larger, crosspoint (i,j) is selected. If *i* is smaller, crosspoint*(j, i)* must be selected. With computer-controlled switching, the line number comparison is trivial. In the older, electromechanically controlled switches, however, the added complexity of the switch control is more significant.



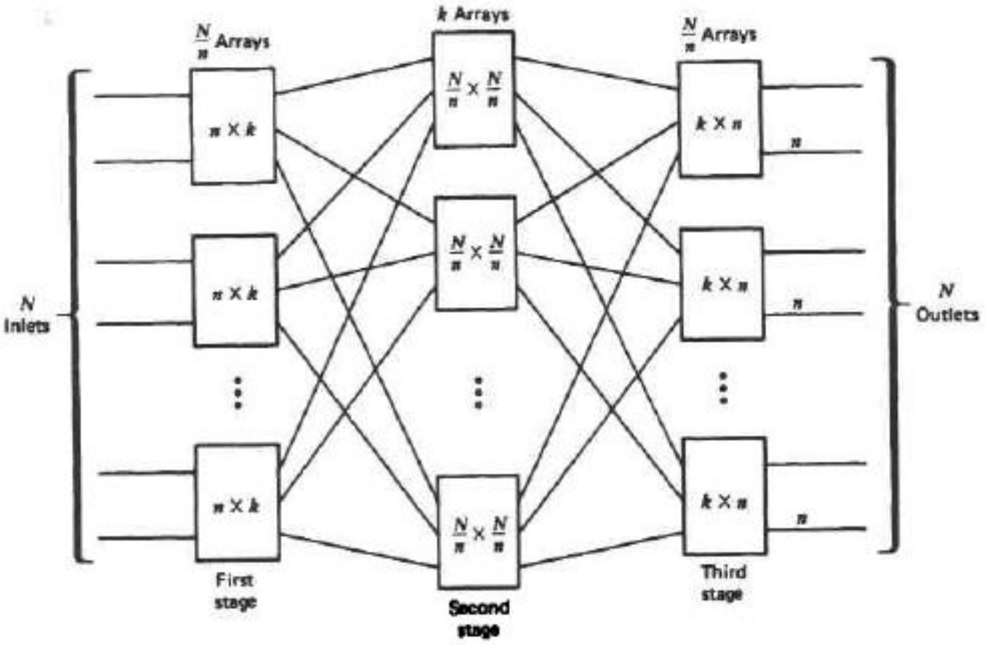
Switching machines for four-wire circuits require separate connections for the go and return branches of a circuit. Thus two separate connections must be established for each service request. Figure depicts a square-matrix structure used to provide both connections. The structure is identical to the square matrix shown in Figure 5.4 for two-wire switching. The difference, however, is that corresponding inlets and outlets are not connected to a common two-wire input. All of the inlets of the four-wire switch are connected to the wire pair carrying the incoming direction of transmission, and all of the outlets are connected to the outgoing pairs. When setting up a connection between four-wire circuits *i* and;, the matrix in Figure 5.5 must select both crosspoints*(i,J)* and *(j, i).* In actual operation these two crosspoints may be selected in unison and implemented as a common module.

Multiple-Stage Switching

In the switching structures described to this point, an inlet is connected directly to an outlet through a single crosspoint. (Four-wire switches use two crosspoints per connection, but only one for an inlet-to-outlet connection.) For this reason, these switching structures are referred to as “single-stage” switches. Single-stage switches have the property that each individual crosspoint can only be used to interconnect one particular inlet-outlet pair. Since the number of inlet-outlet pairs is equal to *N(N* - l)/2 for a triangular array, and *N(N* - 1) for a square array, the number of crosspoints required for a large switch is prohibitive. Furthermore, the large number of crosspoints on each inlet and outlet line imply a large amount of capacitive loading on the message paths. Another fundamental deficiency of single-stage switches is that one specific crosspoint is needed for each specific connection. If that crosspoint fails, the associated connection cannot be established. (An exception is the square, two-wire switch that has a redundant crosspoint for each potential connection. Before the redundant crosspoint could be used as an alternate path, however, the inlet-oriented selection algorithm would have to be modified to admit outlet-oriented selection.)

Analysis of a large single-stage switch reveals that the crosspoints are very inefficiently utilized. Only one crosspoint in each row or column of a square switch is ever in use, even if all lines are active. To increase the utilization efficiency of the crosspoints, and thereby reduce the total number, it is necessary that any particular crosspoint be usable for more than one potential connection. If crosspoints are to be shared, however, it is also necessary that more than one path be available for any potential connection so that blocking does not occur. The alternate paths serve to eliminate or reduce blocking and also to provide protection against failures. The sharing of crosspoints for potential paths through the switch is accomplished by multiple-stage switching. A block diagram of one particular form of a multiple-stage switch is shown in Figure.

The switch of Figure is a three-stage switch in which the inlets and outlets are partitioned into subgroups of *N* inlets and *N* outlets each. The inlets of each subgroup are serviced by a rectangular array of crosspoints. The inlet arrays (first stage) are *n x k* arrays, where each of the *k* outputs is connected to one of the *k* center-stage ar-



Three-stage switching matrix.

rays. The interstage connections are often called junctors. The third stage consists of *k* x *n* rectangular arrays that provide connections from each center-stage array to the groups of *n* outlets-. All center-stage arrays are *N/n* x *N/n* arrays that provide connections from any first-stage array to any third-stage array. Notice that if all arrays provide full availability, there are *k* possible paths through the switch for any particular connection between inlets and outlets. Each of the *k* paths utilizes a separate center- stage array. Thus the multiple-stage structure provides alternate paths through the switch to circumvent failures. Furthermore, since each switching link is connected to a limited number of crosspoints, capacitive loading is minimized.

The total number of crosspointsN_x required by a three-stage switch, as shown in Figure , is

$$N_X = 2Nk + k\left(\frac{N}{n}\right)^2$$

N = number of inlets-outlets

n = size of each inlet-outlet group *k* = number of center-stage arrays

As is demonstrated shortly, the number of crosspoints defined in Equation can be significantly lower than the number of crosspoints required for single-stage switches. First, however, we must determine how many center-stage arrays are needed to provide satisfactory service

Nonblocking Switches

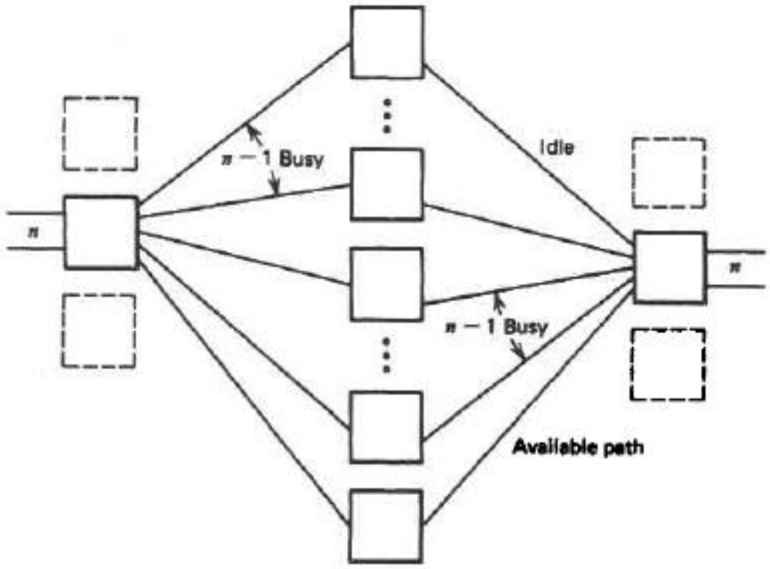
One attractive feature of a single-stage switch is that it is strictly nonblocking. If the called party is idle, the desired connection can always be established by selecting the particular crosspoint dedicated to the particular input-output pair. When crosspoints are shared, however, the possibility of blocking arises. In 1953 Charles Clos of Bell Laboratories published an analysis of three-stage switching networks showing how many center-stage arrays are required to provide a strictly nonblockingoperation. His result demonstrated that if each individual array is nonblocking, and if the number of center stages *k* is equal to *2n-l,* the switch is strictly nonblocking.

The condition for a nonblocking operation can be derived by first observing that a connection through the three-stage switch requires locating a center-stage array with an idle link from the appropriate first stage and an idle link to the appropriate third stage. Since the individual arrays themselves are nonblocking, the desired path can be set up any time a center stage with the appropriate idle linkscan be located. A key point in the derivation is to observe that since each first-stage array has *n* inlets, only *n* - 1 of these inlets can be busy when the inlet corresponding to

Figure 5.3 Graded rectangular switching matrix.

the desired connection is idle. If *k* is greater than *n* - 1, it follows that, at most, *n* - 1 links to center-stage arrays can be busy. Similarly, at most *n*— 1 links to the appropriate third-stage array can be busy if the outlet of the desired connection is idle.

The worst-case situation for blocking occurs (as shown in Figure 5.7) if all *n* - 1 busy links from the first-stage array lead to one set of center-stage arrays and if all *n* - 1 busy links to the desired third-stage array come from a separate set of center-stage arrays. Thus these two sets of center-stage arrays are unavailable for the desired connection. However, if one more center-stage array exists, the appropriate input and output links must be idle, and that center stage can be used to set up the connection. Hence if *k = (n* -1) + *(n* - 1) + 1 = *2n* - 1, the switch is strictly nonblocking. Substituting this value of *k* into Equation 5.1 reveals that for a strictly nonblocking operation of a three-stage switch



$$N_X = 2N(2n-1) + (2n-1)\left(\frac{N}{n}\right)^2$$

As expressed in Equation the number of crosspoints in a nonblocking three- stage switch is dependent on how the inlets and outlets are partitioned into subgroups of size *n.* Differentiating Equation with respect to *n* and setting the resulting expression equal to 0 to determine the minimum reveal that (for large *N)* the optimum value of *n* is *(N/2)*$^{1/2}$. Substituting this value of *n* into Equation then provides an expression for the minimum number of crosspoints of a nonblocking three-stage switch:

Figure 5.8 Probability graph of three-stage network.

$$N_X(\min) = 4N(\sqrt{2N} - 1)$$

where,

N = total number of inlets-outlets.

Table provides a tabulation of N_x(min) for various-sized nonblocking three- stage switches and compares the values to the number of crosspoints in a single-stage square matrix. Both switching structures inherently provide four-wire capabilities, a requirement for digital switches because voice digitization implies four-wire circuits.

As indicated in Table, a three-stage switching matrix provides significant reductions in crosspoints, particularly for large switches. However, the number of crosspoints for large three-stage switches is still quite prohibitive. Large switches typically use more than three stages to provide greater reductions in crosspoints. For example, the No. 1 ESS uses an eight-stage switching matrix that can service up to 65,000 lines. The most significant reductions in crosspoint numbers are achieved not so much from additional stages but by allowing the switch to introduce acceptably low probabilities of blocking.

Crosspoint Requirements of Nonblocking Switches

Number of Lines	Number of Crosspoints for Three-Stage Switch	Number of Crosspoints for Single-Stage Switch
128	7,680	16,256
512	63,488	261,632
2,048	516,096	4.2 million
8,192	4,2 million	67 million
32,768	33 million	1 billion
131,072	268 million	17 billion

Blocking Probabilities: Lee Graphs

Strictly nonblocking switches are rarely needed in most voice telephone networks. Both the switching systems and the number of circuits in interoffice trunk groups are sized to service most requests as they occur, but economics dictates that network implementations have limited capacities that are occasionally exceeded during peak-traffic situations. Equipment for the public telephone network is designed to provide a certain maximum probability of blocking for the busiest hour of the day. The value of this blocking probability is one aspect of the telephone company’s grade of service.

A typical residential telephone is busy 5-10% of the time during the busy hour. Business telephones are often busy for a larger percentage of their busy hour (which may not coincide with a residential busy hour). In either case, network-blocking occurrences on the order of 1%* during the busy hour do not represent a significant reduction in the ability to communicate since the called party is much more likely to have been busy anyway. Under these circumstances, end office

Figure 5.8 Probability graph of three-stage network.

switches and, to a lesser degree, PBXs can be designed with significant reductions in crosspoints by allowing acceptable blocking probabilities.

There are a variety of techniques that can be used to evaluate the blocking probability of a switching matrix. These techniques vary according to complexity, accuracy, and applicability to different network structures. One of the most versatile and conceptually straightforward approaches of calculating blocking probabilities involves the use of probability graphs as proposed by C. Y. Lee. Although this technique requires several simplifying approximations, it can provide reasonably accurate results, particularly when comparisons of alternate structures are more important than absolute numbers. The greatest value of this approach lies in the ease of formulation and the fact that the formulas directly relate to the underlying network structures. Thus the formulations help provide insight into the network structures and how these structures might be modified to change the performance.

In the following analyses we are determining the blocking probabilities of various switching structures using utilization percentages, or “loadings,” of individual links. The notation *p* will be used, in general, to represent the fraction of time that a particular link is in use (i.e., *p* is the probability that a link is busy). In addition to a utilization percentage or loading, *p* is also sometimes referred to as an occupancy. The probability that a link is idle is denoted by $q = 1 - p$.

When any one of *n* parallel links can be used to complete a connection, the composite blocking probability *B* is the probability that all links are busy[+]:

$$B=p^n$$

Transmission and switching equipment in the public network is normally designed for even lower blocking probabilities to provide for growth in the traffic volume.

Equations assume each link is busy or idle independently of other links.

Figure 5.8 Probability graph of three-stage network.

When a series of *n* links are all needed to complete a connection, the blocking probability is most easily determined as 1 minus the probability that they are all available:

$$B = 1\text{-}q^n$$

This graph relates the fact that any particular connection can be established with *k* different paths: one through each center-stage array. The probability that any particular interstage link is busy is denoted by *p'*. The probability of blocking for a three-stage network can be determined as

B = probability that all paths are busy = (probability that an arbitrary path is busy)*

= (probability that at least one link in a path is busy)*

= $(1\text{-}q'^2)k$ (5.6)

where*k* = number of center-stage arrays

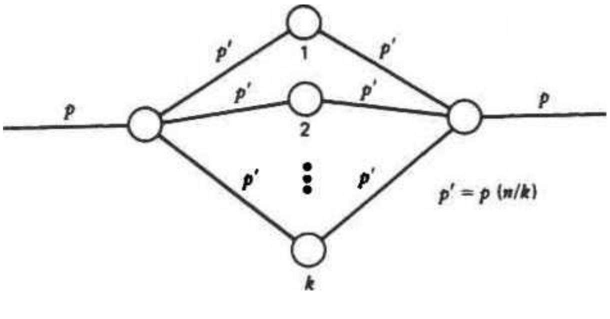
q' = probability that an interstage link is idle, = 1 - *p'*.

If the probability *p* that an inlet is busy is known, the probability *p* that an interstage link is busy can be determined as

$$p' = \frac{p}{\beta} \qquad (p < \beta)$$

where p = *k/n*. Equation presents the fact that when some number of inlets (or outlets) are busy, the same number of first-stage outputs (or third-stage inputs) are also busy. However, there are P = *k/n* times as many interstage links as there are inlets or outlets. Hence the percentage of interstage links that are busy is reduced by the factor β.

The factor P is defined as though *k>n*, which implies that the first stage of the switch is providing space expansion (i.e., switching some number of input links to a larger number of output links). Actuallyβ, (J may be less than 1, implying that the first



stage is concentrating the incoming traffic. First-stage concentration has been used in end office or large PBX switches where the inlets are lightly loaded (5-10%). In tandem or toll offices, however, the incoming trunks are heavily utilized, and expansion is usually needed to provide adequately low-blocking probabilities.

Substituting Equation provides a complete expression for the blocking probability of a three-stage switch in terms of the inlet utilization^:

$$B = \left[1 - \left(1 - \frac{p}{\beta} \right)^2 \right]^k$$

VISION SWITCHING 11

The number of center arrays was chosen in each case to provide a blocking probability on the order of 0.002. The inlet utilization in each example was assumed to be 10%. Notice that the designs with small but finite blocking probabilities are significantly more cost effective than nonblocking designs.

The switch designs in Table assume that the inlets are only 10% busy, as might be the case for an end office switch or a PBX. The dramatic savings in crosspoints for large switches is achieved by introducing significant concentration factors (1/p) into the middle stage. When the inlet utilization is higher (as typically occurs in tandem switches), high concentration factors are not acceptable, and the crosspoint requirements therefore increase. Table 5.3 lists corresponding crosspoint requirements and implementation parameters for inlet loadings of 70%.

The results presented in Tables 5.2 and 5.3 indicate that very large switches still require prohibitively large numbers of crosspoints, even when blocking is allowed. As mentioned previously, very large switches use more than three stages to provide further reductions in crosspoints. Figure 5.9 shows a block diagram of a five-stage switch obtained by replacing every center-stage array in Figure 5.6 with a three-stage array. This particular structure is not optimum in terms of providing a given level of performance with the fewest crosspoints, but it is a useful design because of its modularity. (Furthermore, it is a lot easier to analyze than some other five-stage structures.)

If the middle three stages of a five-stage switch as shown in Figure 5.9 are strictly nonblocking ($k_2 = 2n_2 - 1$), the design provides a savings of 8704 crosspoints in each

Three-Stage Switch Designs for Blocking Probabilities of 0.002 and Inlet Utilization of 0.1

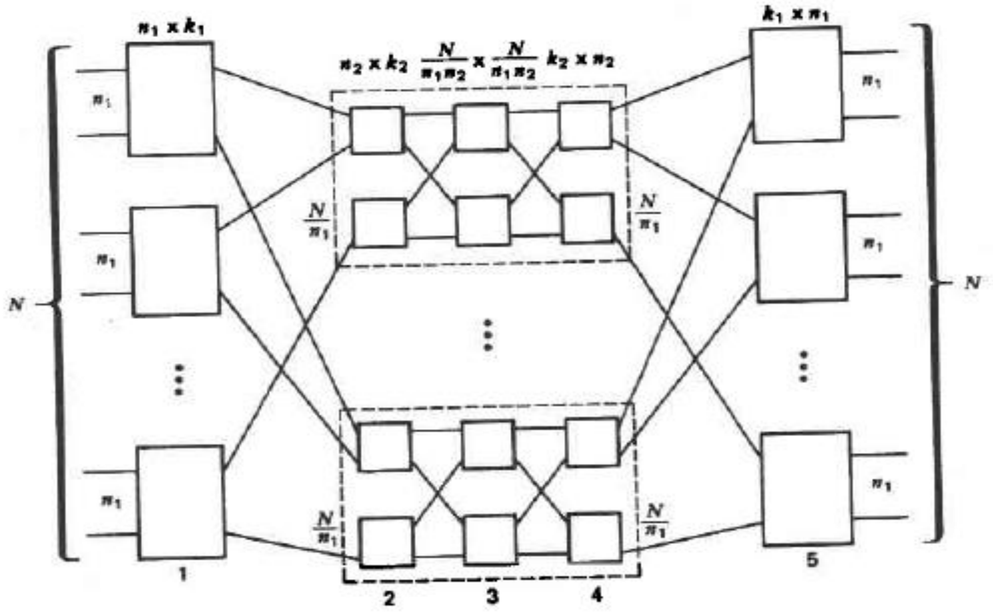
Switch Size, *N*	*n*	*k*	*β*	Number of Crosspoints	Number of Crosspoints in Nonblocking Design	
128	8	5	0.625	2,560	7,680	(*k*= 15)
512	16	7	0.438	14,336	63,488	(*k*=-31)
2,048	32	10	0.313	81,920	516,096	(*k* = 63)
8,192	64	15	0.234	491,520	4.2 million	(*k*= 127)
32,768	128	24	0.188	3.1 million	33 million	(*k*-=255)
131,072	256	41	0.160	21.5 million	268 million	(*k*= 511)

Three-Stage Switch Designs for Blocking Probabilities of 0.002 and Inlet Utilizations of 0.7

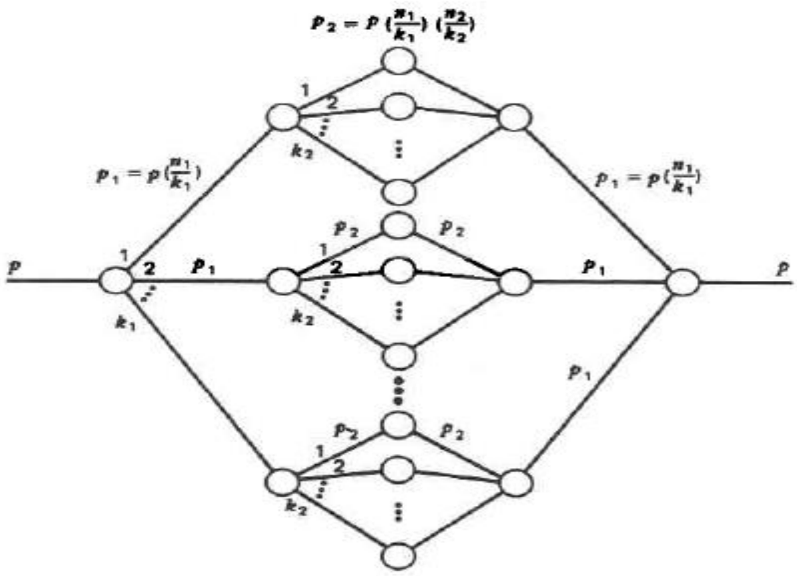
Switch Size N	n	k	β	Number of Crosspoints	Number of Crosspoints in Nonblocking Design	
128	8	14	1.75	7,168	7,680	(k = 15)
512	16	22	1.38	45,056	63,488	(k=-31)
2,048	32	37	1.16	303,104	516,096	(k = 63)
8,192	64	64	1.0	2.1 million	4.2 million	(k = 127)
32,768	128	116	0.91	15.2 million	33 million	(k -=255)
131,072	256	215	0.84	113 million	268 million	(k= 511)

center-stage array of the 32,768-line, three-stage switch designs presented earlier. Hence a little over 1 million crosspoints are saved in the 32,768-line tandem switch design of Table 5.3. Since the middle stages do not introduce blocking, the performance of this five-stage switch is identical to the performance of the three-stage design. Naturally, a more cost-effective design could be obtained by allowing small amounts of blocking in the middle stages. The probability graph of the five-stage switch is shown in Figure 5.10. From this graph, the blocking probability is determined as follows:

$$B = \{1 - (q_1)^2 [1 - (1 - q_2^2)^{k_2}]\}^{k_1}$$



Five Stage Switching


$p_2 = p(\frac{n_1}{k_1})(\frac{n_2}{k_2})$
$p_1 = p(\frac{n_1}{k_1})$
p
p_1
p_2
1
2
k_1
k_2

probabilitygraphof five stage network

where$q_1 = 1 – p_1$ and $q_2 = 1 – p_2$

Even greater crosspoint reductions can, of course, be achieved by using more stages to replace the rather large first- and third-stage arrays. For example, the total number of crosspoints in the 32,000-line switch can be reduced to less than 3 million. The 130,000-line switch is not practical with electromechanical switching matrices but is well within the capabilities of a digital time division switch.

Blocking Probabilities: Jacobaeus

The formulations of blocking probability obtained from probability graphs entail several simplifying assumptions. One of these assumptions involves expressing the composite blocking probability of the alternate paths as the product of the blocking probabilities of each individual path. This step assumes that the individual probabilities are independent. In fact, the probabilities are not independent, particularly when significant amounts of expansion are present. Consider a switching matrix with k = 2*n* - 1. Equation produces a finite blocking probability even though the switch is known to be strictly nonblocking. The inaccuracy results because when $2n$ - 2 paths are busy, the remaining path is assumed to be busy with a probability of 1 - $(q')^2$. In fact, the remaining path is necessarily idle.

In general, when space expansion exists, the assumption of independent individual probabilities leads to an erroneously high value of blocking. The inaccuracy results because as more and more paths in a switch are found to be busy, the remaining paths are less and less likely to be in use (only a subset of *n* of the interstage links can ever be busy at any one time).

A more accurate but not exact analysis of multistage switching matrices was presented in 1950 by C. Jacobaeus. Although the analysis is conceptually straightforward, it does involve a considerable amount of manipulation that is not presented here. The resulting equation for a three-stage switch is obtained from reference as

$$B = \frac{(n!)^2}{k!(2n-k)!} p^k (2-p)^{2n-k}$$

Where, *n* = number of inlets (outlets) per first- (third-) stage array

k = number of second-stage arrays

p = inlet utilization

In the interest of comparing the two methods, Equations 5.6 and 5.10 have been evaluated for three-stage switches with varying amounts of space expansion.

Table 5.4 reveals that the two analyses are in close agreement for near-unity expansion factors. In fact, if p = 1, the two formulations produce identical results. As expected, the Lee graph analysis (Equation 5.8) produces overly pessimistic values for the blocking probability when $\beta > 1$.

Table reveals that a Lee graph analysis (Equation 5.8) consistently underestimates the blocking probability when concentration exists. Actually, the Jacobaeus analysis presented in Equation 5.10 also underestimates the blocking probability if large concentration factors and high blocking probabilities are used. When necessary, more accurate techniques can be used for systems with high concentration and high blocking. However, switches with high blocking probabilities normally have no practical interest so they are not considered here.

TABLE 5.4 Comparison of Blocking Probability Analyses (p = **0.7)**[a]

Number of Center Stages,	Space Expansion, β	Lee Equation	Jacobaeus Equation
14	0.875	0.548	0.598
16	1.0	0.221	0.221
20	1.25	0.014	0.007
24	1.50	3.2 x 10^{-4}	2.7 x10^{-5}
28	1.75	3.7 x10^{-6}	7.7 x10^{-9}
31[b]	1.94	8.5 x 10^{-8}	0.1 x 10^{-12}

[a] Switch size *N* = 512; inlet group size n = 16; inlet utilization *p* = 0.7.

[b] Nonblocking.

TABLE 5.5 Comparison of Blocking Probability Analyses {*p*=0.1)[a]

Number of Center Stages,	Space Expansion, β	Lee Equation 5.8	Jacobaeus Equation
6	0.375	0,0097	0.027
8	0.5	2.8 x 10-^{4}	8.6 x 10^{-4}
10	0.625	4.9 x 10^{-8}	1.5 x 10^{-5}
12	0.75	5.7 x 10^{-8}	1.4 x 10^{-7}
14	0.875	4.0 x 10^{-10}	7.8 x 10^{-10}
16	1.0	2.9 x 10^{-12}	2.9 x 10^{-12}

[a] Switch size /V = 512; inlet group size 16; inlet utilization p = 0.1.

Users of PBXs sometimes experienced high blocking probabilities, but blocking in these cases usually arises from too few tie lines to other corporate locations or too few trunk circuits to the public network.

Up to this point, the blocking probability analyses have assumed that a specific inlet is to be connected to a specific outlet. Also, it has been assumed that the requests for service on the individual lines are independent. These assumptions are generally valid for switching one subscriber line to another in an end office switch or for connecting one station to another in a PBX. Neither of these assumptions apply to connections to or from a trunk group.

When connecting to a trunk circuit, any circuit in a trunk group is acceptable. Thus the blocking probability to a specific circuit is only as important as its significance in the overall blocking to the trunk group. The blocking probability to any particular circuit in a trunk group can be relatively large and still achieve a low composite blocking probability to the trunk group as a whole. If the blocking probabilities to the individual trunks are independent, the composite blocking probability is the product of the individual probabilities. However, the paths to the individual trunk circuits normally involve some common links (e.g., the junctors from a first-stage array to all second-stage arrays). For this reason the individual blocking probabilities are nor* mally dependent, which must be considered in an accurate blocking probability analysis.

As an extreme example, consider a case where all trunks in a trunk group are assigned to a single outlet array in a three-stage switch. Since the paths from any particular inlet to all trunks in the group are identical, the ability to select any idle trunk is useless. In practice, the individual circuits of a trunk group should be assigned to separate outlet arrays.

Another aspect of trunk groups that must be considered when designing a switch or analyzing the blocking probabilities involves the interdependence of activity on the individual circuits within a trunk group. In contrast to individual subscriber lines or PBX stations, individual circuits in a trunk group are not independent in terms of their probabilities of being busy or idle. If some number of circuits in a trunk group are tested and found to be busy, the probability that the remaining circuits are busy is increased. The nature of these dependencies is discussed more fully in Chapter 12. At this point it is only necessary to point out that these dependencies cause increased blocking probabilities if the individual trunks are competing for common paths in the switch. Again, the effect of these dependencies is minimized by assigning the individual trunks to separate inlet-outlet arrays so that independent paths are involved in connections to and from the trunk group. This process is sometimes referred to as decorrelating the trunk circuits.

One last aspect of the blocking probability as a grade of service parameter that must be mentioned involves variations in the loading of the network by individual users. In the design examples for an end office presented earlier, it was tacitly assumed that all subscribers are busy 10% of the time during a busy hour. In fact, some subscribers are active much more than 10% of the time, and other subscribers are active less than 10% of the time. In terms of traffic theory, some subscribers present more than 0.1 erlangs of traffic to the network, whereas others present less.

When a switch is partitioned into subgroups (as all large switches must be) and the traffic is concentrated by first-stage switching arrays, a few overactive subscribers in one subgroup can significantly degrade service for the other subscribers in the subgroup. It does not matter that the subscribers in some other subgroup may be experiencing lower than average blocking. Their essentially nonblocking service is no compensation for those subscribers experiencing a relatively poor grade of service.

Operating companies have traditionally solved the problem of overactive subscribers by specifically assigning the most active lines (businesses) to separate inlet groups of the switch. Sometimes this procedure requires making traffic measurements to determine which lines are most active and reassigning these lines to different parts of the switch. These procedures fall into the general category of line administration. If the subgroups are large enough, or if the designs provide adequate margin for overactive users, this aspect of line administration can be minimized. One feature of a digital switch that can be utilized in this regard is the ability to design economical switches with very low nominal blocking probabilities so that wide variations in traffic intensities can be accommodated.

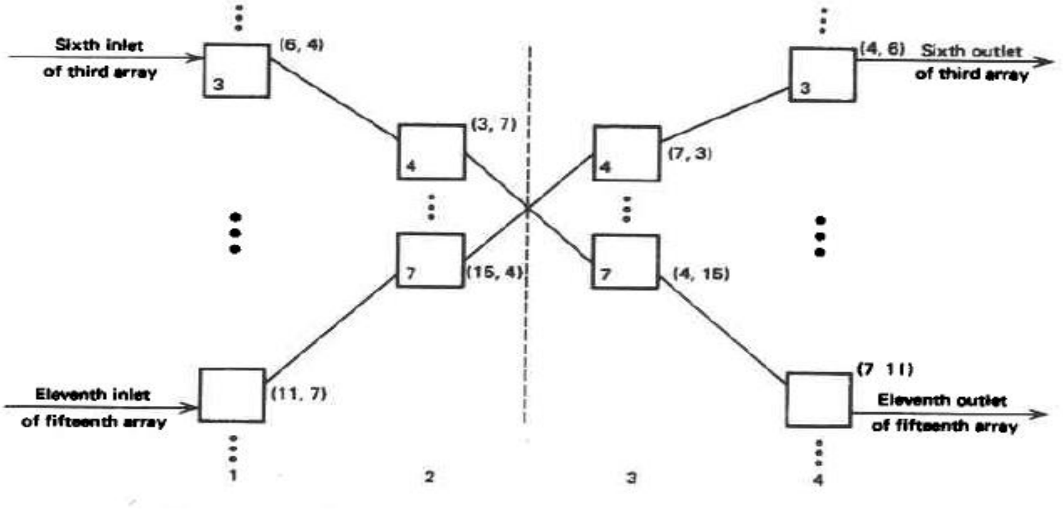
Even a modem digital switch can experience loading problems when confronted with extreme traffic conditions. An Internet service provider (ISP) in a metropolitan area may attract an extremely large amount of traffic that all passes through a single class 5 switch. Although the connections to the ISP are lines, as far as the switch is concerned, they actually represent a trunk group with very heavy traffic, so much so that special line administration is required.

Folded Four-Wire Switches

Multiple-stage switches can be used for either two- or four-wire switching operations. Figure depicts a four-wire connection through a four-stage switch. Notice that two paths must be established for the complete connection. A two-wire connection requires only one path since each outlet is externally connected to its corresponding inlet.

The two paths shown in Figure demonstrate a particularly useful relationship: One path is a mirror image of the other path. If the switch diagram is folded about the vertical center line, the paths coincide. Hence this method of setting up connections is sometimes referred to as a folded operation. When all connections in the switch are set up with a folded relationship, several benefits result.

First of all, only one pathfinding operation is needed since the reverse path is automatically available as a mirror image of the forward path. In essence, every crosspoint on one side is paired with another crosspoint in a corresponding array on the opposite side of the switch. Whenever one crosspoint of a pair is used in a connection, the other crosspoint in the pair is also used. For example, the third inlet array in the first stage uses crosspoint (6,4) to connect its sixth inlet to its fourth outlet (leading to the fourth array of the second stage). The corresponding crosspoint in the third outlet array of the last stage connects its fourth inlet (coming from the fourth array in the fourth stage) to its sixth outlet. In general, crosspoint*ij* in one array is paired with crosspoint*j, i* in the corresponding array on the opposite side of a switch. Since the availability of one crosspoint in a pair ensures the availability of the other, the reverse path is automatically specified and available.



Four-wire connection through four-stage switch

A second advantage of the folded four-wire operation results because the amount of information specifying the status of the switch can be cut in half. Only the status ofeach pair of crosspoints or associated junctor is needed to find an available path through the switch.

A third benefit of the folded structure occurs because the blocking probability is one-half of the probability of finding two paths independently. It might seem that pairing the crosspoints in the described manner would restrict the paths available for a particular connection. On the contrary, the crosspoint pairings merely guarantee that a reverse path is automatically available for any selected path in the forward direction.

The folded operation in the preceding paragraphs referred to a switch with an even number of switching stages. An even number was chosen because the concept is easiest to demonstrate when no center stage is present. The basic approach can be extended to odd numbers of switching stages if the center stage contains an even number of arrays and is folded about a horizontal line at the center of the stage. In this manner, crosspoint*i,j*in the top center-stage array is paired with crosspoint*j,i*in the bottom center-stage array, and so on.

Pathfinding

Determining a path through a single-stage switch is virtually automatic since the necessary crosspoint is uniquely specified by the inlet-outlet pair to be connected. In contrast, availability of more than one path in a multiple-stage switch complicates the path selection process. The call processor of the switch must keep track of which potential paths for a particular connection are available in a state store. A pathfinding routine processes the state store information to select an available path. Whenever a new connection is established or an old one released, the state store is updated with the appropriate information.

Pathfinding Times

Pathfinding operations require the use of common equipment and must therefore be analyzed to determine the rate at which connect requests can be processed. The time required to find an available path is directly dependent on how many potential paths are tested before an idle one is found. Some systems can test a number of paths in parallel and thereby shorten the processing time. Since the expected number of potential paths that must be tested to find an idle path is a function of link utilization, pathfinding times unfortunately increase when the common control equipment is busiest.

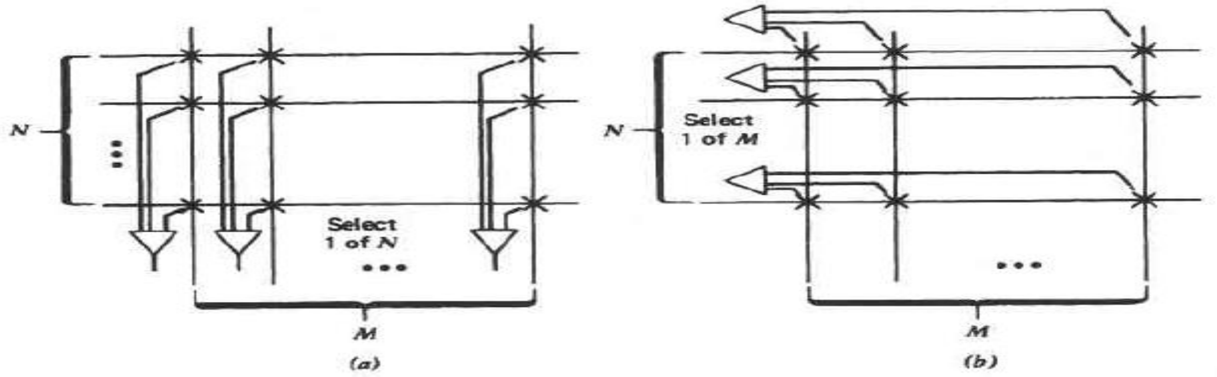
Assume that the probability of a complete path through the switch being busy is denoted by *p*. If each of *k* possible paths through the switch has an equal and independent probability of being busy, the expected number of paths N_p that must be tested before an idle path is found is determined.

Switch Matrix Control

When an available path through a common control switching network is determined, the control element of the switch transfers the necessary information to the network to select the appropriate crosspoints. Crosspoint selection within a matrix is accomplished in one of two ways. The control may be associated with the output lines and therefore specify which inputs are to be connected to the associated outputs or the control information may be associated with each input and subsequently specify to which outputs the respective inputs are to be connected. The first approach is referred to as *output-associated control* while the second is called *input-*

associated control. These two control implementations are presented in Figure

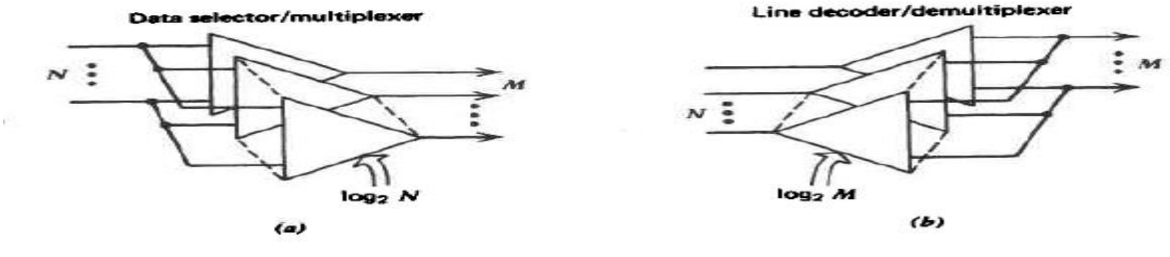
Input-associated control was inherently required in step-by-step switches where the information (dial pulses) arrived on the input link and was used to directly select the output links to each successive stage. In common control systems, however, the address information of both the originating line and the terminating line is simultaneously available. Hence the connection can be established by beginning at the desired outlet and proceeding backward through the switch while selecting inputs to each stage. .



Switch matrix control: *(a)* output associated; (*b*) input associated.

The implementation of both types of digital crosspoint arrays using standard components is shown in Figure. Output-associated control uses a conventional data selector/multiplexer for each matrix output. The number of bits required to control each data selector is $\log_2 N$, where N is the number of inlets. Thus the total number of bits required to completely specify a connection configuration is $M \log_2 N$.

Input-associated control can be implemented using conventional line decoders/demultiplexers. The outputs are commoned using a “wired-or” logic function. Thus the output gates of each decoder circuit must be open-collector or tristate devices if transistor-transistor-logic (TTL) is used. The total number of bits required to specify a connection configuration in this case is $N\log_2 M$.A significant drawback of input-associated control arises from the need to disable unused inputs to prevent cross connects when another input selects the same output.



Standard component implementation of digital crosspoint array: (a) output- associated control; (*b*) input-associated control.

With output-associated control, unused outputs can remain connected to an input without preventing that input from being selected by another output. For this reason and for generally greater speeds of operation, digital switching networks typically use output-associated control. Notice, however, that the total amount of information needed to specify a connection configuration with input-associated control is less than that with output control if the number of inputs N is much smaller than the number of outputs M $(N \log_2 M < M \log_2 N)$. Furthermore, input-associated control is more flexible in terms of wired-or (e.g., bus) expansion.

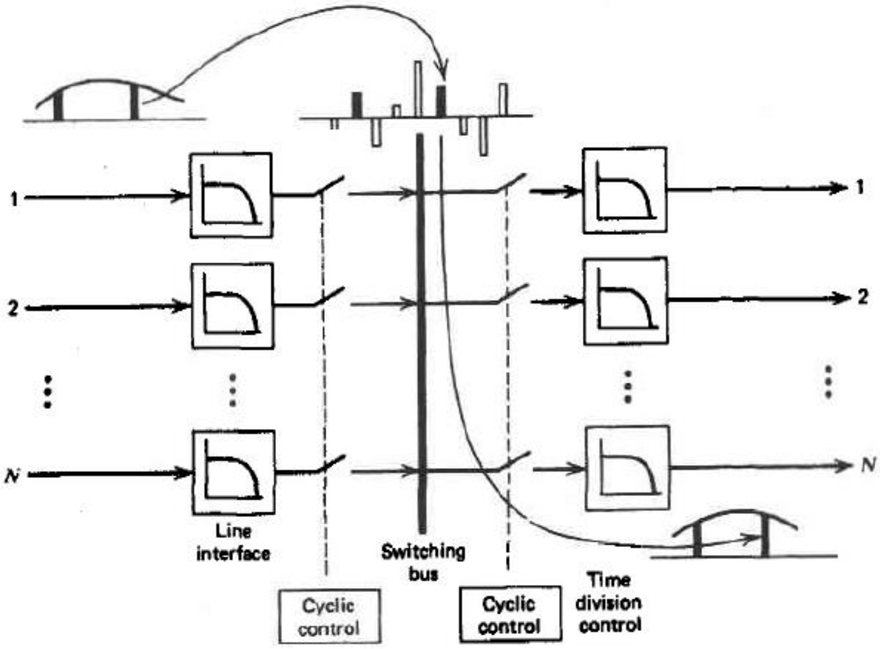
TIME DIVISION SWITCHING

As evidenced by multiple-stage switching, sharing of individual crosspoints for more than one potential connection provides significant savings in implementation costs of space division switches. In the cases demonstrated, the crosspoints of multistage space switches are shared from one connection to the next, but a crosspoint assigned to a particular connection is dedicated to that connection for its duration. Time division switching involves the sharing of crosspoints for shorter periods of time so that individual crosspoints and their associated interstage links are continually reassigned to existing connections. When the crosspoints are shared in this manner, much greater savings in crosspoints can be achieved. In essence, the savings are accomplished by time division multiplexing the crosspoints and interstage links in the same manner that transmission links are time division multiplexed to share interoffice wire pairs.

Time division switching is equally applicable to either analog or digital signals. At one time, analog time division switching was attractive when interfacing to analog transmission facilities, since the signals are only sampled and not digitally encoded. However, large analog time division switches had the same limitations as do analog time division transmission links: the PAM samples are particularly vulnerable to noise, distortion, and crosstalk. Thus, large electronic switching matrices have always incorporated the cost of digitizing PAM samples to maintain end-to-end signal quality. The low cost of codecs and the prevalence of digital trunk interconnections imply that analog switching is now used in only the smallest of switching systems (e.g., electronic key systems).

Analog Time Division Switching

Although analog time division switching has become obsolete, it is a good starting point to establish the basic principle of time division switching. Figure depicts a particularly simple analog time division switching structure. A single switching bus supports a multiple number of connections by interleaving PAM samples from receive line interfaces to transmit line interfaces. The operation is depicted as though the receive interfaces are separate from the respective transmit interfaces. When connecting two-wire analog lines, the two interfaces are necessarily implemented in a common module. Furthermore, in some PAM-PBX systems, analog samples were simultaneously transferred in both directions between the interfaces.



Analog time division switching.

Included in Figure are two cyclic control stores. The first control store controls gating of inputs onto the bus one sample at a time. The second control store operates in synchronism with the first and selects the appropriate output line for each input sample. A complete set of pulses, one from each active input line, is referred to as a frame. The frame rate is equal to the sample rate of each line. For voice systems the sampling rate ranges from 8 to 12 kHz. The higher sampling rates were sometimes used to simplify the bandlimiting filter and reconstructive filters in the line interfaces.

Digital Time Division Switching

The analog switching matrix described in the preceding section is essentially a space division switching matrix. By continually changing the connections for short periods of time in a cyclic manner, the configuration of the space division switch is replicated once for each time slot. This mode of operation is referred to as time multiplexed switching. While this mode of operation can be quite useful for both analog and digital signals, digital time division multiplexed signals usually require switching between time slots as well as between physical lines. Switching between time slots represents a second dimension of switching and is referred to as time switching.

In the following discussion of digital time division switching it is assumed, unless otherwise stated, that the switching network is interfaced directly to digital time division multiplex links. This assumption is generally justified since, even when operating in an analog environment, the most cost-effective switch designs multiplex groups of digital signals into TDM formats before any switching operations take place. Thus most of the following discussion is concerned with the internal structures of time division switching networks and possibly not with the structure of an entire switching complex.

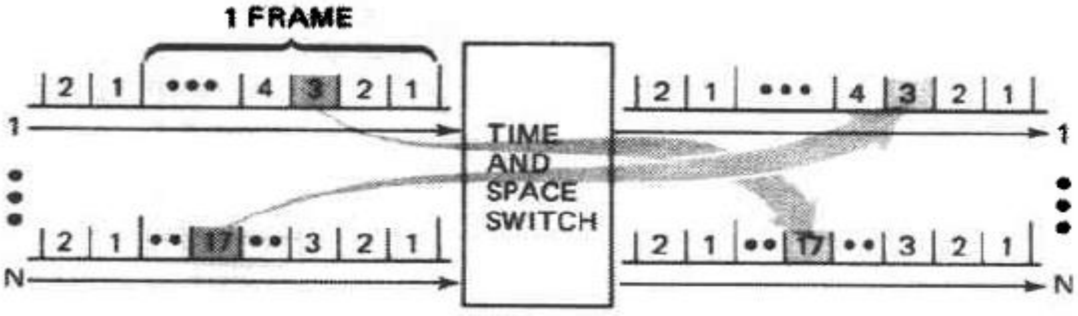
The basic requirement of a time division switching network is shown in Figure. As an example connection, channel 3 of the first TDM link is connected to channel 17 of the last TDM link. The indicated connection implies that information arriving in time slot 3 of the first input link is transferred to time slot 17 of the last output link. Since the voice digitization process inherently implies a four-wire operation, the return connection is required and realized by transferring information from time slot 17 of the last input link to time slot 3 of the first output link. Thus each connection requires two transfers of information, each involving translations in both time and space.

A variety of switching structures are possible to accomplish the transfers indicated in Figure . All of these structures inherently require at least two stages: a space division switching stage and a time division switching stage. As discussed later, larger switches use multiple stages of both types. Before discussing switching in both dimensions, however, we discuss the characteristics and capabilities of time switching alone.

A Digital Memory Switch

Primarily owing to the low cost of digital memory, time switching implementations provide digital switching functions more economically than space division implementations. Basically, a time switch operates by writing data into and reading data out of a single memory. In the process, the information in selected time slots is interchanged, as shown in Figure. When digital signals can be multiplexed into a single TDM format, very economical switches can be implemented with time switching alone. However, practical limitations of memory speed limit the size of a time switch so that some amount of space division switching is necessary in large switches. As demonstrated in later sections, the most economical multistage designs usually perform as much switching as possible in the time stages.

The basic functional operation of a memory switch is shown in Figure. Individual digital message circuits are multiplexed and demultiplexed in a fixed manner



Time and space division switching

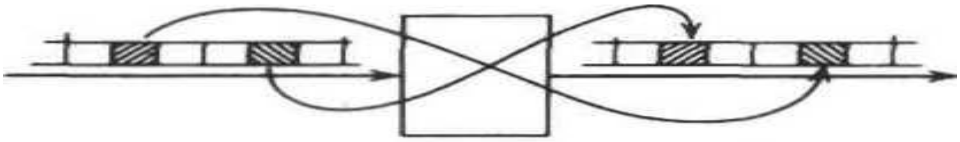
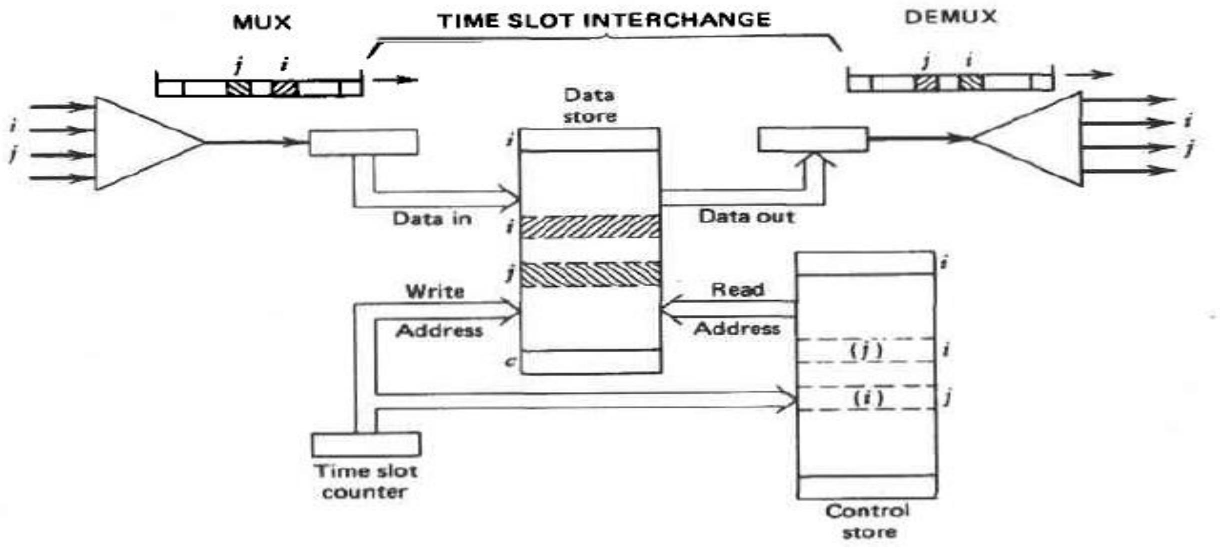


Figure 5.16 Time slot interchange operation.

to establish a single TDM link for each direction of travel. The multiplexing and demultiplexing functions can be considered as part of the switch itself, or they may be implemented in remote transmission terminals. In either case, a byte-interleaved form of multiplexing is required. The asynchronous transmission hierarchy (DS2, DS3, DS4) uses bit interleaving and therefore requires back-to-back demultiplexing and multiplexing operations before switching can be accomplished. In contrast, the SONET transmission format described in Chapter 8 can provide byte interleaving specifically so it can be more directly interfaced to a digital switching system.

The exchange of information between two different time slots is accomplished by a time slot interchange (TSI) circuit. In the TSI of Figure data words in incoming time slots are written into sequential locations of the data store memory. Data words for outgoing time slots, however, are read from addresses obtained from a control store. As indicated in the associated control store, a full-duplex connection between TDM channel *i* and TDM channel *j* implies that data store address *i* is read during outgoing time slot *j* and vice versa. The data store memory is accessed twice during each link time slot. First, some control circuitry (not shown) selects the time slot number



MUX/TSI/DEMUX memory switch.

as a write address. Second, the content of the control store for that particular time slot is selected as a read address.

Since a write and a read are required for each channel entering (and leaving) the TSI memory, the maximum number of channels *c* that can be supported by the simple memory switch is

Figure 5.17 MUX/TSI/DEMUX memory switch.

$$c = \frac{125}{2t_c}$$

where 125 is the frame time in microseconds for 8 kHz sampled voice and t_c is the memory cycle time in microseconds.

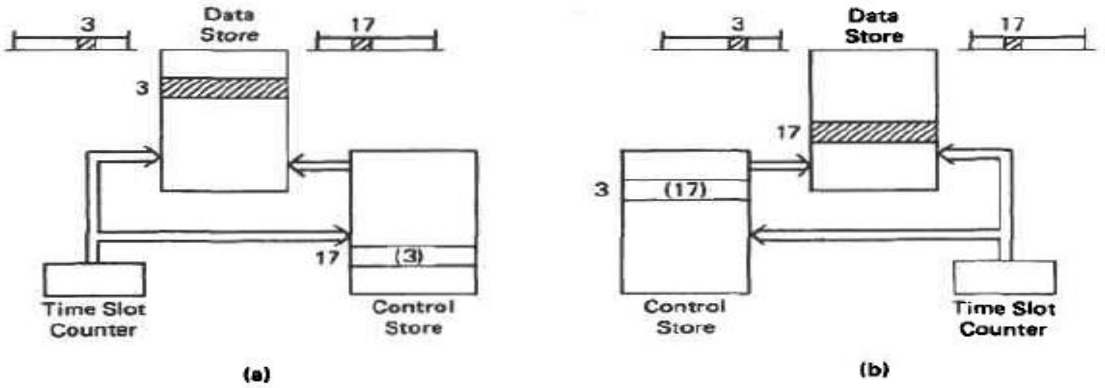
As a specific example, consider the use of a 15.2 nsec of memory. Equation indicates that the memory switch can support 4096 channels (2048 full duplex connections) in a strictly nonblocking mode of operation. The complexity of the switch (assuming digitization occurs elsewhere) is quite modest: The TSI memory stores one frame of data organized as 4096 words by 8 bits each. The control store also requires 4096 words, but each word has a length equal to $\log_2(c)$ (which is 12 in the example). Thus the memory functions can be supplied by 4096 x 8 and 4096 x 12 bit random-access memories (RAMs). The addition of a time slot counter and some gating logic to select addresses and enable new information to be written into the control store can be accomplished with a handful of conventional integrated circuits (ICs).

This switch should be contrasted to a space division design that requires more than 1.5 million crosspoints for a nonblocking three-stage switch. Although modem IC technology might be capable of placing that many digital crosspoints in a few ICs, they could never be reached because of pin limitations. As mentioned, one of the main advantages of digital signals is the ease with which they can be time division multiplexed. This advantage arises for communication between integrated circuits as well as for communication between switching offices.

Time Stages in General

Time switching stages inherently require some form of delay element to provide the desired time slot interchanges. Delays are most easily implemented using RAMs that are written into as data arrive and read from when data are to be transferred out. If one memory location is allocated for each time slot in the TDM frame format, the information from each TDM channel can be stored for up to one full frame time without being overwritten.

There are two basic ways in which the time stage memories can be controlled: written sequentially and read randomly or written randomly and read sequentially. Figure 5.18 depicts both modes of operation and indicates how the memories are accessed to translate information from time slot 3 to time slot 17. Notice that both modes of operation use a cyclic control store that is accessed in synchronism with the time slot counter.



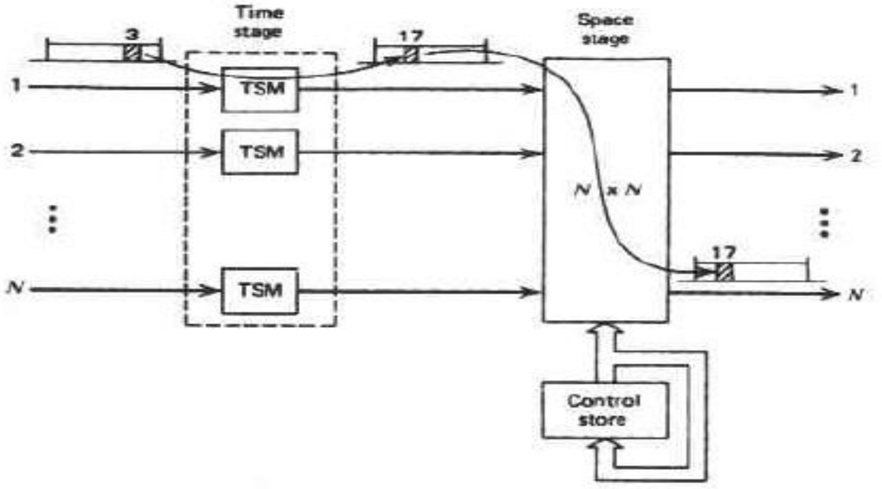
Time switch modules: *(a)* sequential writes/random reads; *(b)* random writes/ sequential reads.

The first mode of operation in Figure implies that specific memory locations are dedicated to respective channels of the incoming TDM link. Data for each incoming time slot are stored in sequential locations within the memory by incrementing a modulo-c counter with every time slot. As indicated, the data received during time slot 3 are automatically stored in the third location within the memory. On output, information retrieved from the control store specifies which address is to be accessed for that particular time slot. As indicated, the seventeenth word of the control store contains the number 3, implying that the contents of data store address 3 is transferred to the output link during outgoing time slot 17.

The second mode of operation depicted in Figure is exactly the opposite of the first one. Incoming data are written into the memory locations as specified by the control store, but outgoing data are retrieved sequentially under control of an outgoing time slot counter. As indicated in the example, information received during time slot 3 is written directly into data store address 17, where it is automatically retrieved during outgoing TDM channel number 17. Notice that the two modes of time stage operation depicted in Figure are forms of output-associated control and input-associated control, respectively. In a multiple-stage design example presented later, it is convenient to use one mode of operation in one time stage and the other mode of operation in another time stage.

TWO-DIMENSIONAL SWITCHING

Larger digital switches require switching operations in both a space dimension and a time dimension. There are a large variety of network configurations that can be used to accomplish these requirements. To begin with, consider the simple switching structure shown in Figure. This switch consists of only two stages: a time stage T followed by a space stage S. Thus this structure is referred to a time-space (TS) switch.



Time-space (TS) switching matrix .

The basic function of the time stage is to delay information in arriving time slots until the desired output time slot occurs. At that time the delayed information is transferred through the space stage to the appropriate output link. In the example shown the information in incoming time slot 3 of link 1 is delayed until outgoing time slot 17 occurs. The return path requires that information arriving in time slot 17 of link *N* be delayed for time slot 3 of the next outgoing frame. Notice that a time stage may have to provide delays ranging from one time slot to a full frame.

Associated with the space stage is a control store that contains the information needed to specify the space stage configuration for each individual time slot of a frame. This control information is accessed cyclically in the same manner as the control information in the analog time division switch. For example, during each outgoing time slot 3, control information is accessed that specifies interstage link number 1 is connected to output link JV. During other time slots, the space switch is completely reconfigured to support other connections.

As indicated, a convenient means of representing a control store is a parallel end- around-shift register. The width of the shift register is equal to the number of bits required to specify the entire space switch configuration during a single time slot. The length of the shift register conforms to the number of time slots in a frame. Naturally, some means of changing the information in the control store is needed so that new connections can be established. In actual practice, the control stores may be implemented as RAMs with counters used to generate addresses in a cyclic fashion (as in the time stage control stores shown previously).

Implementation Complexity of Time Division Switches

In previous sections, alternative space division switching structures were compared in terms of the total number of crosspoints required to provide a given grade of service. In the case of solid-state electronic switching matrices, in general, and time division switching, in particular, the number of crosspoints alone is a less meaningful measure of implementation cost. Switching structures that utilize ICs with relatively large numbers of internal crosspoints are generally more cost effective than other structures that may have fewer crosspoints but more packages. Hence a more relevant design parameter for solid-state switches would be the total number of IC packages. If alternate designs are implemented from a common set of ICs, the number of packages may closely reflect the number of crosspoints.

In addition to the number of crosspoints in space division stages, a digital time division switch uses significant amounts of memory that must be included in an estimate of the overall cost. The memory count includes the time stage memory arrays and the control stores for both the time stages and the space stages. In large switches, the number of space stage crosspoints can be reduced at the expense of increasing the amount of memory required. Thus, a thorough analysis of implementation complexity requires knowing the relative cost of a crosspoint relative to the cost of a bit of memory. Because a crosspoint is closely associated with an external connection, it costs significantly more than a bit of memory. The use of standard medium-scale integration ICs leads to one crosspoint costing about the same as 100 bits of memory.[1]' Use of custom- or application-specific ICs with integrated memory can change this factor, particularly because large-scale IC packages provide access to many more crosspoints per pin. For purposes of illustrating various design trade-offs, the following analyses of implementation complexity continue to consider crosspoint costs to be 100 times the cost of a memory bit. Depending on the implementation approach, this factor may not be accurate, but minimizing the cost of a matrix is no longer much of a concern, except in extremely large switching systems (e.g., in switches exceeding 100,000 voice channels).

The implementation complexity is expressed as follows:

$$\text{Complexity} = N_{\text{X}} + \frac{N_{\text{B}}}{100}$$

where, N_x = number of space stage crosspoints

N_B = number of bits of memory

The implementation complexity determined in obviously dominated by the number of crosspoints in the space stage. A significantly lower complexity (and generally lower cost) can be achieved if groups of input links are combined into higher level multiplex signals before being switched. The cost of the front-end multiplexers is relatively small if the individual DS1 signals have already been synchronized for switching. In this manner, the complexity of the space stage is decreased appreciably while the overall complexity of the time stage increases only slightly. (See the problems at the end of this chapter.) The implementation costs are reduced proportionately, up to the point that higher speeds dictate the use of a more expensive technology.

A significant limitation of the TS structure of Figure occurs when a connection has to be made to a specific channel of an outlet as opposed to any channel of an outlet. A connection to a specific channel is blocked whenever the desired time slot of the inlet TSM is already in use. For example, if time slot 17 of the first inlet is in use for some connection from the first inlet to some link other than link *N*, the connection from channel 3 of inlet 1 to channel 17 of inlet *N* cannot be made. Because of this limitation, the TS structure is useful only if the outlets represent trunk groups, which implies any channel of an outlet is suitable.* Applications that require connections to specific channels require additional stages for adequate blocking probability performance.

Multiple-Stage Time and Space Switching

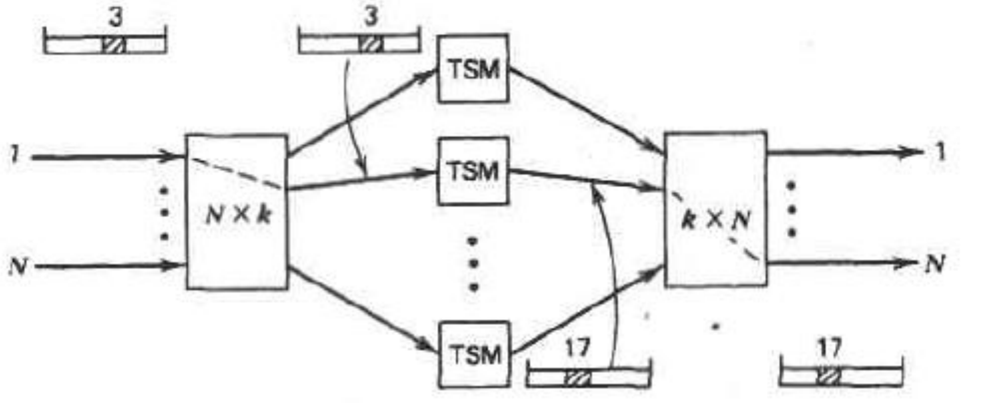
As discussed in the preceding section, an effective means of reducing the cost of a time division switch is to multiplex as many channels together as practical and perform as much switching in the time stages as possible. Time stage switching is generally less expensive than space stage switching primarily because digital memory is much cheaper than digital crosspoints (AND gates). To repeat, the crosspoints themselves are not so expensive, it is the cost of accessing and selecting them from external pins that makes their use relatively costly.

Naturally, there are practical limits as to how many channels can be multiplexed into a common TDM link for time stage switching. When these practical limits are reached, further reductions in the implementation complexity can be achieved only by using multiple stages. Obviously, some cost savings result when a single space stage matrix of a TS or ST switch can be replaced by multiple stages.

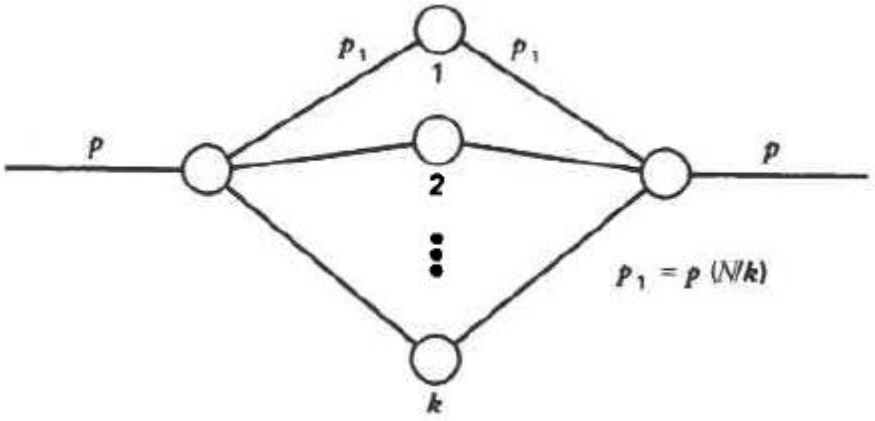
A generally more effective approach involves separating the space stages by a time stage, or, conversely, separating two time stages by a space stage. The next two sections describe these two basic structures. The first structure, consisting of a time stage between two space stages, is referred to as a space-time-space (STS) switch. The second structure is referred to as a time-space-time (TST) switch.

STS Switching

A functional block diagram of an STS switch is shown in Figure. Each space switch is assumed to be a single-stage (nonblocking) switch. For very large switches, it may be desirable to implement the space switches with multiple stages. Establishing a path through an STS switch requires finding a time switch array with an available write access during the incoming time slot and an available read access during the desired outgoing time slot. When each individual stage (S, T, S) is nonblocking, the operation is functionally equivalent to the operation of a three-stage space switch. Hence a probability graph in Figure of an STS switch is identical to the probability graph of Figure for three-stage space switches. Correspondingly, the blocking probability of an STS switch is



Space-time-space (STS) switching structure.



Probability graph of STS switch with nonblocking stages

$$B = (1 - q'^2)^k$$

where $q' = 1 - p' = 1 - p/\beta \qquad (\beta = k/N)$

k = number of center-stage time switch arrays

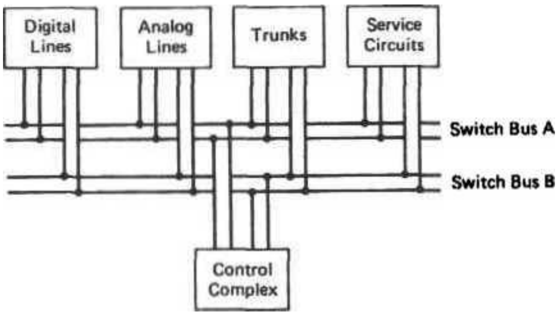
Assuming the spaceswitches are single-stage arrays and that each TDM link has *c*message channels, wemay determine the implementation complexityofan STS switch as

Complexity = number of space stage crosspoints + (number of space stage control bits + number of time stage memory bits + number of time stage control bits)/100

$$= 2kN + \frac{2kc \log_2 N + kc(8) + kc \log_2 c}{100}$$

The value of implementation complexity obtained in Example 5.3 should be compared to the number of crosspoints obtained for an equivalent-sized three-stage switch listed in Table 5.2. The space switch design requires 81,920 crosspoints while the STS design requires only 430 equivalent crosspoints. The dramatic savings comes about as a result of the voice signals having already been digitized and multiplexed (for transmission purposes). When digital switches were first inserted into an analog environment, the dominant cost of the switch occurred in the line interface. Digital interface costs are much lower than analog interface costs, particularly on a per-channel basis.

Probability graph of No.4 ESS matrix



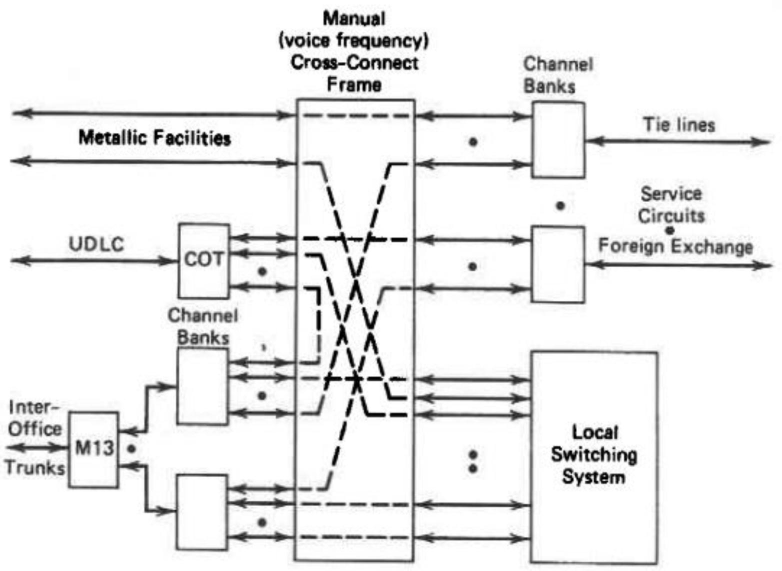
System 75 matrix architecture.

As an example, congestion analyses show that if 800 stations are 37.5% busy, on average, the probability that a station requests service when 400 stations are already busy is 10"^{5}. Thus the system is virtually nonblocking for voice applications. Because data connections are often full-period connections, the blocking probability for voice connections may become more significant in applications involving intensive voiceband data switching

DIGITAL CROSS-CONNECT SYSTEMS

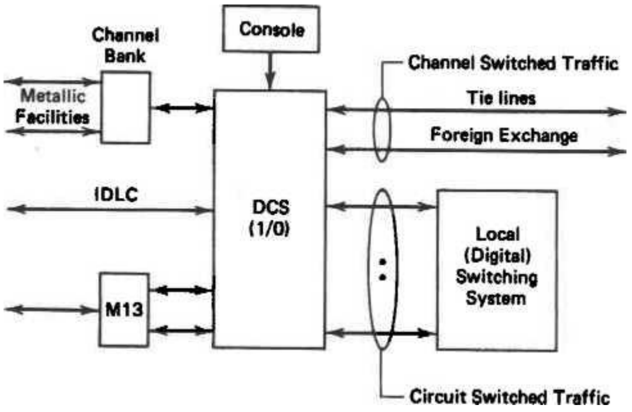
A DCS is basically a digital switching matrix with an operations interface for setting up relatively static connections between input and output signals or channels. Instead of establishing connections in response to signaling information pertaining to call-by- call connection requests, DCS connections are established in response to network configuration needs. The most basic function of a DCS is to act as an electronic patch panel in lieu of a manual cross-connect facility. Manual cross-connect frames were typically installed in switching offices as demarcation points between transmission facilities and switching machines and in wire centers as demarcation points between feeder cables and distribution facilities. In both cases, the major purpose of the crossconnect frame was to allow rearrangement of transmission circuits and access for testing the circuits in both directions.

Figure shows a typical manual cross-connect environment while Figure shows the same function implemented with a DCS, Manual cross connects are implemented with bridging clips and punch-down wires. Cross connections within the DCS are established by entering commands at the As indicated in the two figures, a major feature of a DCS is its elimination of back-to-back multiplexing functions when cross connecting individual channels within TDM transmission links. Additional advantages of a DCS system with respect to manual cross-connect systems are:



Fi lanual cross-connect system.

Automatic Record Keeping. Because the cross connects are under processor control, circuit connection reports are readily available through the management interface. In contrast, records in manual systems were inherent by error prone and often out of date.



Remote and Rapid Provisioning. Provisioning is the basic process of providing (or discontinuing) service to a particular customer. The basic operations involved are outside-plant cross connections, inside-plant cross connections, configuration changes in switching system data base, and customer record updates in business (billing) systems. Obviously, the more these processes were automated, the faster and more accurately they could be performed.

Automated Test Access. Testing analog circuits at a manual cross-connect frame involves physically breaking the connection (by removing bridging clips) and attaching the test equipment to the side of the circuit to be tested. All manual operations are eliminated with an electronic patch panel by entering commands at the management console to connect the desired test tones and (DSP) measurement channels to the circuit under test.

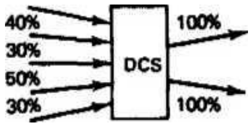
Figure depicts two types of network traffic: circuit-switched traffic and channel-switched traffic. Circuit-switched traffic represents locally switched traffic (DSO circuits typically) and channel switched-traffic refers to leased line equivalents of digital channels. Channel-switched traffic might terminate at another public network switching office as in a foreign exchange (FX) circuit or at a distant customer premise as a tie line of a private network. In the latter case, more than one DSO channel might be concatenated together to form a single higher rate channel referred to as a fractional T1 circuit or *M* x 64-kbps channel. Channel-switched services typically account for over one-half of the transmission bandwidth between U.S. metropolitan offices. The process of separating channel-switched services from circuit-switched services is often referred to as "grooming." Figure also shows that universal digital loop carrier (UDLC) becomes integrated digital loop carrier (IDLC) in a digital environment

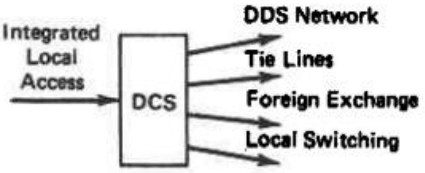
Consolidation and Segregation

Two basically different DCS grooming functions are depicted in Figures 5.32 and 5.33: consolidation and segregation. When multiple-access lines carrying traffic destined to a common distant node are partially filled, the per-channel costs of transport to the remote node can be reduced by consolidating the traffic. Conversely, when different types of traffic originate at a single location, it is desirable to allow a single facility (e.g., a T1 line) to carry all types of traffic and segregate it at the DCS. Examples of such traffic are circuit-switched channels, DDS channels, tie lines, multidrop data circuits, FX circuits, or other special services that might be available only at another office.

Partially Filled Totally Filled Local Access Network

SOX



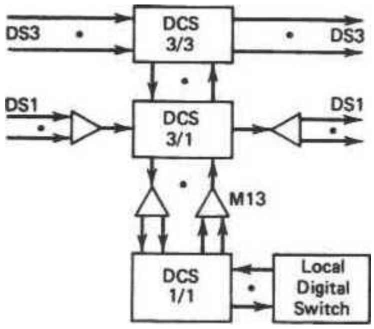


REFERENCES 35

DCS Hierarchy

DCS Hierarchy

The cross-connect system depicted in Figure terminates DS1 signals and interchanges DSO signals, which leads to the designation DCS 1/0. Similarly, a digital cross-connect system that terminates DS3 signals and rearranges DS1 signals within the DS3s is referred to as a DCS 3/1. If a DCS, such as a DCS 3/0, provides rearrangement of lower level signals, such as DSOs, it does not necessarily mean that it also provides cross-connect services for intermediate-level signals, such as DSls. Cross connection of DSls requires transparent mapping of the entire 1.544 Mbps, which includes the framing channela capability that may not be provided when 64-kbps DSO channels are the primary focus. When a DCS provides multiple levels of cross connects, the intermediate levels are sometimes designated as in DCS 3/1/0 for terminating DS3s and rearranging DSls and DSOs. In most cases, it is not necessary to provide cross-connect services at all levels of the digital signal hierarchy for all of the terminations. Figure 5.34 depicts a DCS hierarchy that provides rearrangement of lower level signals on only subsets of higher level signals. The higher level (e.g., DS3) signals that are not cross connected to a lower level DCS may be unchannelizedhighspeed (44.736-Mbps) signals or transit DS3s that terminate on



a DCS 3/1 or 3/0 at a distant location. The primary purpose of the DCS 3/3 is to provide network restoration or protection switching of the DS3 signals and to possibly provide time-of-day rearrangement of trunk group assignments.

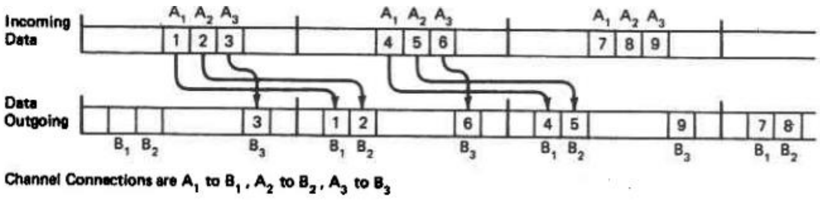
Integrated Cross-Connect Equipment

Figure depicts distinct multiplexing equipment and two distinct and colocated switching systems: the DCS and the local digital switch. Although DCS functions within the public network are traditionally implemented with separate equipment, private networks often utilize equipment that provides integrated multiplexing and crossconnect functions. This equipment has evolved from CPE-based T1 multiplexers because TSI circuits are insignificant hardware additions to T1 multiplexing hardware. Such equipment are variously referred to as intelligent multiplexers, nodal processors, or networking T1 multiplexers. Cross-connect functions are also incorporated into newer DLC systems because, again, the cost of the TSI function is virtually nil.

Integration of cross-connect functions with higher level digital signals (such as a DS3) has not occurred because the manner in which the higher level signals are multiplexed (described in is not amenable for direct termination on a switching system. The newer form of (synchronous) multiplexing, as specified by the SONET standard, allows integration of higher level digital multiplexers with cross-connect systems for both public and private equipment.

Although cross-connect functions are basically nothing more than “pegged” or “nailed-up” circuit-switched connections, the two functions are traditionally implemented separately because of the following differences in the application requirements:

1. A DCS needs to be strictly nonblocking at the DSO level, which is generally uneconomical in large public network switches.
2. Transparent cross-connection of a 1.544-Mbps signal requires transport of the framing bit that is not possible in typical digital circuit switches because DS1 interface equipment assumes the DS1 signals are channelized with framing in the 193rd bit.
3. DCS functionality does not involve processing of signaling bits so fully functional circuit switch interfaces have excess costs when used as interfaces to cross-connect systems.
4. Cross connecting multiple, concatenated DSO channels (fractional T1 channels) requires maintaining order in the concatenated data stream. Although the order can always be maintained by careful mapping of the individual 64-kbps connections, it is a function that is typically not provided in DSO circuit switching software. Figure shows how the order of bytes in the concatenated data stream can be transposed by ill-chosen 64-kbps connections.



Transposition of data in concatenated time slots.

DIGITAL SWITCHING IN AN ANALOG ENVIRONMENT

When digital end office switches (or PBXs) are installed in an analog environment, the analog interfaces are necessarily unchanged. Although the digital switch may interface with digital subscriber carrier or digital fiber feeder systems, these systems merely extend the analog interface point closer to the subscriber.

Zero-Loss Switching

As already mentioned, a well-designed digital transmission and switching system adds no appreciable degradation to the end-to-end quality of digitized voice. In particular, the derived analog signal coming out of a decoder can be adjusted to the same level as that presented to the encoder at the far end. Curiously, zero-loss transmission presents some significant problems when digital switching is used in the analog transmission environment of a class 5 central office.

Analog end office switches are two-wire switches designed to interconnect bidirectional two-wire customer loops. The voice digitization process, however, inherently requires separation of the go and return signal paths involved in a connection. Thus, a class 5 digital end office or digital PBX must be a four-wire switch. When inserted into a two-wire analog environment, hybrids are required to separate the two directions of transmission. As shown in Figure, hybrids at each end of the internal digital subnetwork produce a four-wire circuit with the potential for echoes and singing. (Amplifiers are shown in conjunction with the encoders to offset forward path loss inherent in the hybrids.) Instability arises as a result of impedance mismatches at the hybrids causing unwanted coupling from the receive portion to the transmit portion of the four-wire connection. Impedance mismatches occur because of the variability in the lengths and wire sizes of the subscriber loops. In particular, loaded and unloaded wire pairs have markedly different impedance characteristics.

The instability problems are compounded by certain amounts of artificial delay that are required in a digital time division switch. Although the delay through a digital switch (several hundred microseconds, typically) is basically unnoticeable to a user,

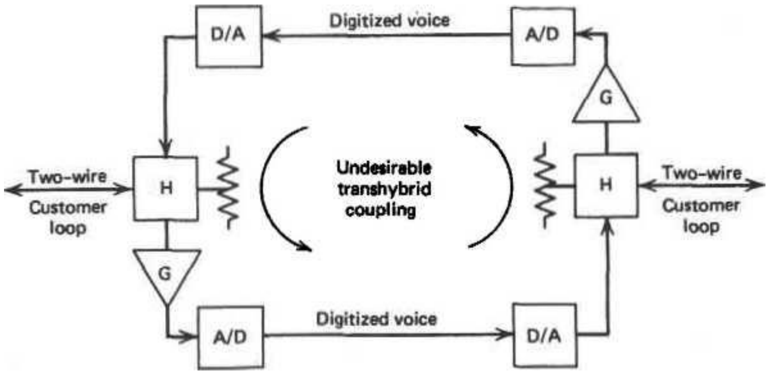


Figure Four wire digital switch with analog interfaces

it represents the equivalent of as much as 30 to 40 miles of wire. This increased delay has the effect of lowering the oscillation frequencies that might otherwise be outside the voiceband and effectively be removed by encoder/decoder filters.

As hybrids were traditionally used at the interface to the toll network where the transmission medium is basically four-wire. In these instances, the instability of the four-wire circuit was controlled by designing a prescribed amount of net attenuation (by way of net loss) into the transmission path of shorter toll network circuits. On the longer circuits, the echoes and singing are eliminated by echo suppressors or echo cancelers. Class 5 digital switches could prevent instability in the same manner: by designing a certain amount of loss into the encoding/decoding process. While the necessary amount of signal attenuation (approximately 2-3 dB in each path) could be tolerated on local connections, the added loss to some toll connections would be unacceptably large.

From the foregoing discussion it can be seen that the use of selectable attenuation is one solution to the instability problems. The necessary loss is inserted into the talking path for local connections but not used on long-distance calls, which already have loss designed into them. A second solution involves matching the impedance at the hybrids more closely. Before the advent of DSP nearly complete elimination of the unwanted coupling was prohibitively expensive. However, adequate isolation of the transmission paths could be accomplished with just two different matching networks: one for loaded loops and one for unloaded loops. Impedance matching has been simplified by the use of advanced interface electronics (DSP) that contain trainable and automatic impedance matching circuitry. The impedance matching circuits essentially represent short-delay echo cancelers.

Notice that an all-digital network (with four-wire telephones) avoids instability problems because there are no two-wire analog lines. Voice is digitized at the telephone and uses a separate path from the receive signal all the way to the destination telephone. Thus an end to-end four-wire circuit completely eliminates echoes and singing, allowing all connections to operate on a zero-loss basis.

BORSCHT

The basic functional requirements of the subscriber loop interface are described. These requirements are repeated here with two additional requirements for a digital switch interface: coding and hybrid. The complete list of interface requirements is unaffectionately known as BORSCHT

B: Battery feed

O: Overvoltage protection

R: Ringing

S: Supervision

C: Coding

H: Hybrid

T:Test

the high-voltage, low-resistance, and current requirements of many of these functions are particularly burdensome to integrated circuit implementations. First-generation digital end office switches reduced the termination costs by using analog switching (concentrators) to common codecs. The DMS-100 of Northern Telecom and the No. 5 ESS of AT&T use analog concentration at the periphery. Integrated circuit manufacturers have worked diligently to implement the BORSCHT functions in what is called a subscriber loop interface circuit (SLIC). Per-line SLICs allow implementation of per-line BORSCHT functions. SLICs can be used in PBX applications with a minimum of other external components. In central office applications, where lightning protection and test access are more demanding, SLICs typically need other components for a complete interface.

Conferencing

In an analog network conference calls are established by merely adding individual signals together using a conference bridge. If two people talk at once, their speech is superposed. Furthermore, an active talker can hear if another conferee begins talking. Naturally, the same technique can be used in a digital switch if the signals are first converted to analog, added, and then converted back to digital.

As described, (i.255 and the i4-law (ITU) code were designed with the specific property of being easily digitally linearizable (EDL). With this property, the addition function can be performed digitally by first converting all codes to linear formats, adding them, and then converting back to compressed formats. To the user, the operation is identical to the customary analog summation. For a conference involving N conferees, N separate summations must be performed, one for each conferee and containing all signals but his own. For a description of the conferencing algorithm in the System 75 PBX of AT&T, see reference. For more general descriptions of conferencing implementations in digital switches, see reference.

Another conferencing technique involves monitoring the activity of all conferees and switching the digital signal of the loudest talker to all others. Although this technique is functionally different from a customary analog conference bridge, it is advantageous for large conferences because the idle channel noise of the inactive talkers does not get added into the output of the conference bridge. High-quality conference circuits also include echo cancelers so higher signal powers can be provided.

7

NETWORK SYNCHRONIZATION CONTROL AND MANAGEMENT

In Chapters 4 and 6 some synchronization requirements of transmission systems are discussed. These requirements involve carrier recovery for coherent detection of modulated signals, clock recovery for sampling incoming data, and framing procedures for identifying individual channels in a TDM signal format. All of these considerations are inherent in digital transmission systems and, for the most part, operate independently of other equipment in a network. An instance of one subsystem's dependency on another was noted for T1 lines. T1 source data must include minimum densities of 1's to maintain timing on the original transmission link. In contrast, other line codes are described that maintain clock synchronization independently of the source data.

This chapter discusses network-related synchronization considerations for interconnecting various digital transmission and switching equipment. Foremost among these considerations is synchronization of switching equipment. When digital switching equipment was first installed in the public telephone network, the interfaces were analog transmission systems. Hence, each switching machine could operate with an autonomous frequency source (clock) that converted all voice signals into digital signals with precisely the same data rate (nominally 64 kbps). These switching matrices were designed to carry one channel rate and one channel rate only. The advent of subsequent digital switch interconnection required switches to carry digital channels originating someplace else in the network—from a different frequency source. Thus, network synchronization requirements arose.

When individual synchronous transmission and switching equipments are interconnected to form a network, certain procedures need to be established that either synchronize the clocks to each other or provide for their interoperability when each subsystem uses independent clocks. Following the discussion of network clock synchronization, the concept of synchronization is extended to other aspects of network control.

7.1 TIMING

All digital systems inherently require a frequency source, or "clock," as a means of timing internal and external operations. Operations timed from a single frequency source do not require particularly stable sources since all commonly clocked elements experience timing variations in common. A different situation occurs when transfers are made from one synchronous equipment to another (as from a transmitter to a receiver). Even if the clock of the receiving terminal is "synchronized" to the transmitting terminal on a long-term or average basis, short-term variations in either clock may jeopardize the integrity of the data transfer. Thus it is generally necessary to use frequency sources (oscillators) in both the transmitter and the receiver that are as stable as is economically feasible.

7.1.1 Timing Recovery: Phase-Locked Loop

A common means of synchronizing a receiver clock to a transmitter clock uses a phase-locked loop (PLL) as shown in Figure 7.1. A phase detector continuously measures the phase difference between the incoming clock and a locally generated clock. The phase detector in Figure 7.1 merely measures the difference in the zero crossings between the two signals. When the zero crossing of the line clock precedes the zero crossing of the local clock, a positive voltage is generated; otherwise, a negative voltage is produced. The output of the phase detector is filtered to eliminate as much receive noise as possible, and then the phase measurement adjusts the frequency of the voltage-controlled oscillator (VCO) to reduce the phase difference. Some amount of noise or interference inevitably passes through the phase detector and the filter, causing erroneous adjustments in the VCO frequency. As time passes, however, a frequency offset produces ever-increasing phase shifts. When the phase difference builds up, it is easier to detect, and the appropriate changes in the VCO occur. Hence the local

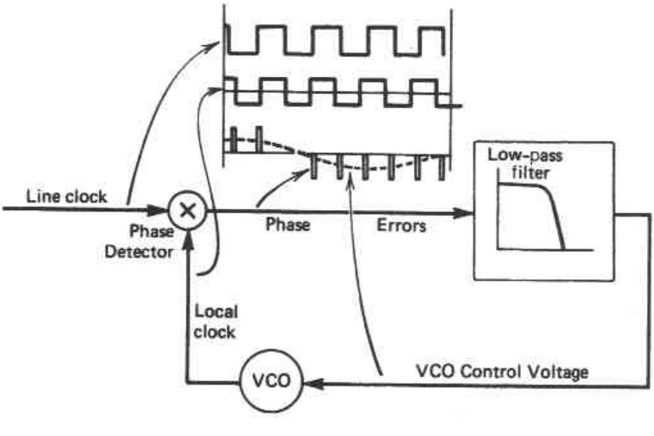


Figure 7.1 Phase-locked loop clock recovery circuit.

clock maintains the desired average frequency but inherently produces short-term variations of phase and frequency as it "hunts" the underlying frequency of the line clock.

The line clock in Figure 7.1 is shown to have a transition in every clock interval, a situation that does occur with Manchester or diphase-type line codes. With some line codes (bipolar or AMI in particular) there are no clock transitions during intervals of binary 0's. In these cases either the clock extraction circuitry inserts artificial transitions extrapolated from previous intervals or the phase detector is disabled during intervals when no pulse is detected.

7.1.2 Clock Instability

The variation in the output frequency of the VCO described above is an example of clock instability. All clocks have a certain amount of instability—even free-running oscillators. An important aspect of clock instability is its frequency: the rate at which the clock frequency changes from being too high to being too low. The frequency of the instability can be directly observed as the frequency spectrum of the VCO control voltage in a PLL clock recovery circuit. When the VCO control voltage varies slowly, the variations are referred to as clock wander. When the variations are more rapid, the instability of the clock is referred to as jitter. The most generally accepted dividing point between wander and jitter is 10 Hz. Pure wander on a T1 line therefore produces positive phase errors for more than 77,200 bit intervals (0.05 sec) followed by more than 77,200 bit intervals of only negative phase errors.* The main sources of clock instability (both wander and jitter) in a network are:

1. Noise and interference
2. Changes in the length of transmission media
3. Changes in velocity of propagation
4. Doppler shifts from mobile terminals
5. Irregular timing information

Noise and Interference

If the low-pass filter of a PLL is designed with a very low cutoff frequency, it can filter out almost all of the noise and interference on a transmission link that would otherwise corrupt the timing recovery. There are three main reasons why arbitrarily low pass filters cannot be used. First, the ability of the PLL to acquire synchronization (e.g., lock on to the underlying clock) is inversely related to the PLL bandwidth. If the VCO begins oscillating at the wrong frequency and the bandwidth is too narrow, the PLL may never pull the oscillator to the frequency of the line clock. Sometimes, this problem is accommodated by using two bandwidths: a wide one to acquire synchronization and a narrow one that is selected after lock is achieved.

*As discussed later, observing wander in this way requires extremely stable clocks as a reference. The normal PLL clock recovery circuitry cannot be used to observe wander because the VCO tracks the relatively long-term phase offsets. Thus the clock recovery circuit does not filter out the wander but passes it on.

A second consideration that generally precludes the use of very narrow filters, even after acquisition, is that the source may vary in frequency that cannot be tracked by a slowly responding PLL. In this case, the recovered clock does not track the ideal sample times leading to high error rates or, worse yet, the PLL loses synchronization altogether and has to reacquire lock.

The third limiting factor for low-bandwidth PLLs is the instability of the VCO itself. If the VCO begins to drift in frequency, very low bandwidth filters preclude adjusting the VCO input voltage soon enough to prevent bit errors or possible loss of synchronization.

Because operational considerations dictate certain minimum PLL bandwidths, noise and interference on the transmission link always cause the recovered clock to be more impaired than the source clock. The PLL does, however, eliminate that portion of a disturbance with frequency content above the bandwidth of the PLL. Thus disturbances with low-frequency content are the most difficult to deal with. Systematic jitter, as produced by particular patterns of intersymbol interference, can have an arbitrarily low frequency content and is discussed more fully in a later section.

An important consideration in the design of a digital transmission link is the accumulation of jitter in tandem clock recovery circuits. If a recovered clock is used to time the transmission of outgoing data, as in a regenerative repeater, some amount of incoming jitter is imbedded into the outgoing clock. The clock recovery circuit in the next receiver tracks its incoming clock but introduces even more jitter due to noise and interference on the second section. Thus jitter accumulates at every regenerative repeater using its received line clock as its transmit clock. If there is a large number of regenerative repeaters, the jitter can accumulate to a point where subsequent clock recovery circuits have difficulty tracking the receive clock, produce sampling errors, and possibly lose lock.

Changes in Length of Transmission Media

Path length changes occur as a result of thermal expansion or contraction of guided transmission media or of atmospheric bending of a radio path. While a path is increasing in length, the effective bit rate at the receiver is reduced because more and more bits are being "stored" in the medium. Similarly, as the path shortens, the bit rate at the receiver increases because the number of bits stored in the transmission link is decreasing. After the path has stabilized, the receive signal returns to the nominal data rate. The most significant changes in path length occur with communications satellites. Geostationary satellites produce path length variations of approximately 200 miles, which cause propagation time variations of approximately 1 msec [1]. Path length changes also occur in guided transmission media such as copper wire and optical fibers. These changes are referred to as diurnal changes because they occur once a day.

Changes in the Velocity of Propagation

Temperature changes not only cause expansion and contraction of wireline transmission media but also can change those propagation constants of the media that determine the velocity of propagation. The resulting change in received clocked stability, however, is much less than that produced by the change in path length [2].

The propagation velocity of radio waves in the atmosphere also changes with temperature and humidity. Although these velocity changes are more significant than those occurring in wirelines, they are still smaller than the path-length-induced variations. Notice that a change in propagation velocity is effectively equivalent to a change in path length since the number of bits "stored" in the transmission path is changed.

Doppler Shifts

The most significant source of potential timing instability in a received clock occurs as a result of Doppler shifts from airplanes or satellites. For example, a Doppler shift induced by a 350-mph airplane amounts to an equivalent clock instability of 5×10^{-7}. Digital mobile telephone receivers must accommodate Doppler shifts equivalent to clock instabilities of about one part in 10^7.* Again, Doppler shifts occur, in essence, as a result of path changes.

Irregular Timing Information

As discussed in Chapter 4, a fundamental requirement of a digital line code is that it provide sufficient timing information to establish and maintain a receiver line clock. If the timing information is data dependent, jitter in the recovered clock increases during periods of relatively low density timing marks. The magnitude of the jitter is dependent not solely on the density of timing marks but also on the timing (data) patterns. In an ideal repeater, only the density would matter. In practice, however, various imperfections lead to pattern-dependent jitter [3].

As discussed later in this chapter, higher level digital multiplexers insert overhead bits into a composite data stream for various purposes. When the higher rate data stream is demultiplexed, the arrival rate of data within individual channels is irregular. This irregularity produces timing jitter when generating new line clocks for the lower rate signals. This source of jitter (waiting time jitter) is often the most troublesome and is discussed in more detail later.

7.1.3 Elastic Stores

The timing instabilities described in the preceding paragraphs essentially represent changes in the number of bits stored in a transmission link. In the case of noise- and interference-induced jitter, the change in "bits stored" occurs because data are sampled a little earlier or a little later than nominal. Since the outgoing data of a regenerative repeater are transmitted according to the recovered clock, a phase offset in the clock means the delay through the repeater is different from when there is no misalignment in timing.

*If you are paranoid about cooperation between cellular operators and law enforcement authorities, you may not want to use your cell phone while speeding.

If phase offsets in successive regenerative repeaters coincide, a net change of several bits of storage in a long repeated transmission link occurs. Since these extra bits enter or leave the transmission link over relatively short periods of time, the accumulated jitter may represent a relatively large, but short lived, instability in the receive clock.

Because regenerative repeaters use incoming sample clocks as output clocks, sustained timing differences between inputs and outputs do not exist. The endpoints of a transmission link, however, may interface to a local clock. In this case a difference between a received and a relatively fixed local clock must be reconciled with an elastic store. An elastic store is a data buffer that is written into by one clock and read from by another. If short-term instabilities exist in either clock, the elastic store absorbs the differences in the amount of data transmitted and the amount of data received. An elastic store can compensate only for short-term instabilities that produce a limited difference in the amounts of data transmitted and received. If sustained clock offsets exist, as with highly accurate but unsynchronized clocks, an elastic store will eventually underflow or overflow.

TDM–Switch Interface

A typical need for an elastic store occurs when a digital transmission link is interfaced to a digital time division switch. As shown in Figure 7.2, the elastic store is placed between the incoming digital transmission link and the inlet side of the switch. In most instances the digital switch provides timing for all outgoing TDM links so that no timing discrepancies exist between these links and the switch. For the time being, assume that the far end of the digital link derives its clock from the receive signal and uses that clock to time digital transmissions returning to the switch. This is the situation that arises when a remote channel bank is connected to a digital switch through T1 lines and is commonly referred to as channel bank "loop timing." When loop timing is used, the line clock on the incoming link of the switch is nominally synchronized to the switch clock. However, for reasons discussed previously, a certain amount of instability in the incoming clock necessarily exists. The elastic store absorbs these instabilities so that purely synchronized data are available for the switch.

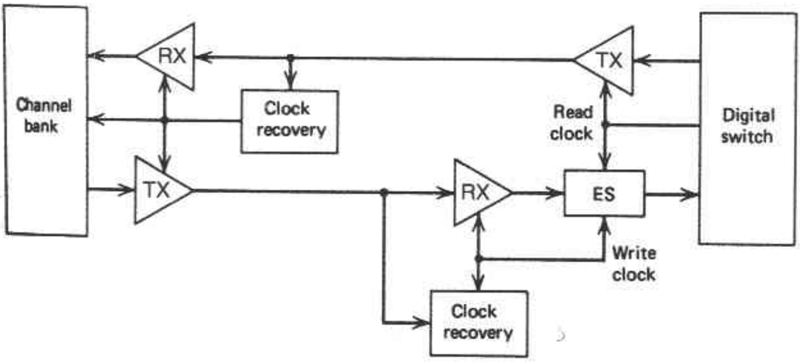


Figure 7.2 Interface between TDM transmission link and a digital switch using an elastic store.

In essence, the loop formed by the transmission links and the elastic store maintains a constant and integral number of clock intervals between the inlet and outlet of the switch. Thus, from a timing point of view, the inlets and outlets operate as though directly connected to each other using a common source of timing.

Removal of Accumulated Jitter

Another application for an elastic store is shown in Figure 7.3, where it is used to remove transmission-induced timing jitter in a regenerative repeater. Normally, a regenerative repeater establishes the transmit timing directly from the locally derived sample clock. In Figure 7.3, however, the transmit timing is defined by a separate local clock. The elastic store absorbs the short-term instabilities in the receive clock, but the long-term frequency of the transmit clock is controlled by maintaining a certain "average level of storage" in the elastic store. Thus the transmit clock is synchronized to the line clock on a long-term basis, but not on a short-term basis. If the elastic store is large enough to accommodate all transient variations in the data rate, high-frequency instability of the input clock is removed.

All regenerative repeaters, regardless of the mechanism used to recover timing, derive their output clocks by averaging the incoming timing information over a period of time. Tuned circuits average the incoming clock for relatively few signal intervals; phase-locked loops do so for many intervals. In all cases a certain amount of storage or delay is implied. An elastic store is merely a mechanism to increase the available delay so that output timing adjustments can be made more gradually. Notice that an elastic store always inserts significant amounts of artificial delay into the data path. To remove arbitrarily low frequency instabilities (wander), arbitrarily large elastic stores (and arbitrarily stable VCOs) are required. Thus a jitter-removing elastic store should be inserted into a chain of repeaters only when jitter accumulation threatens the ability of a regular regenerative repeater to maintain synchronization.

Jitter is not just a phenomenon of a digital transmission link but also occurs in digital storage systems. For example, jitter removal like that shown in Figure 7.3 is used in laser disc players to eliminate blurring of the visual images [4].

Elastic Store Implementations

The required size of an elastic store varies from a few bits to several hundred bits for high-speed, long-distance communications links. Figure 7.4 shows one means of implementing a small elastic store utilizing a series-to-parallel converter, a register, and

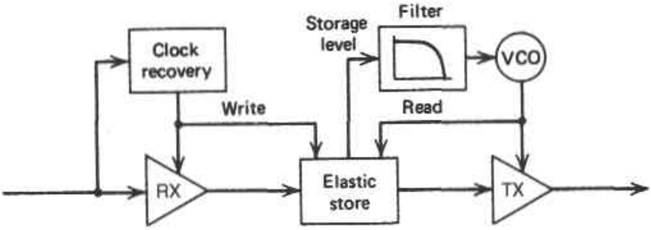


Figure 7.3 Jitter-removing regenerative repeater.

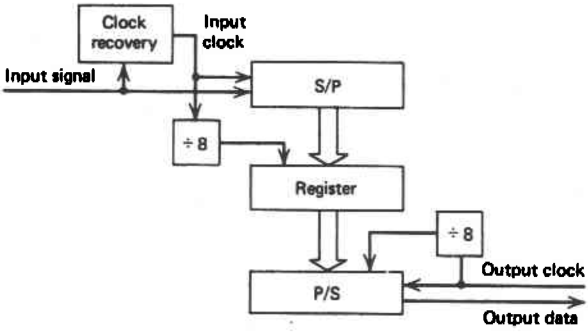


Figure 7.4 Basic implementation of an elastic store.

a parallel-to-series converter. As indicated, incoming data are transferred into the register as soon as each word is accumulated in the series-to-parallel converter. Some time later, data in the register are transferred to the output parallel-to-series converter as a complete word is shifted out. Notice that transfers to the parallel-to-series converter are independent of the incoming clock. As long as output transfers occur between input transfers, no data are lost, and short-term jitter is absorbed by varying delays through the elastic store.

Normally, some control circuitry (not shown) is needed to initialize the elastic store so that the first transfer into the register occurs midway between output transfers. This process means some incoming data are initially discarded by the series-to-parallel register until the desired transfer time occurs.

The relative times of the parallel transfers into and out of the holding register provide a direct indication of the relative phase of the input and output clocks. Thus the parallel transfer clocks contain the information needed to generate VCO control voltages if the elastic store is being used to remove accumulated transmission jitter.

The basic structure shown in Figure 7.4 can be extended to implement larger elastic stores, as shown in Figure 7.5. The only change involves the substitution of a first-in, first-out (FIFO) buffer for the holding register of Figure 7.4. This data buffer is designed specifically to allow input transfers under the control of one clock while outputs are controlled by a different clock. Normally, the FIFO buffer is initialized by inhibiting output transfers until it is half full. In fact, some commercially available FIFO buffers have an output signal specifically indicating when it is half full or greater.

7.1.4 Jitter Measurements

A simple circuit for measuring timing jitter is shown in Figure 7.6. As indicated, it is nothing more than a phase-locked loop (PLL) with the output of the phase comparator providing the measurement of the timing jitter. Normally, the bandwidth of the lowpass filter (LPF) is very small so the VCO ignores short-term jitter in the timing signal. If there is no jitter at all, the output of the phase comparator is constant and no signal is passed by the high-pass selection filter (HPF).

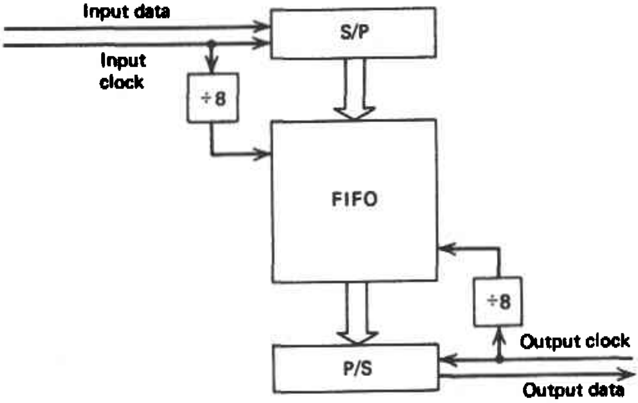


Figure 7.5 FIFO implementation of an elastic store.

Very low frequency jitter cannot be measured by the circuit in Figure 7.6 because the VCO tracks slowly changing phase shifts. Low-frequency jitter may be of no concern, however, because it can be tracked by a PLL. Higher frequency jitter, on the other hand, is more apt to cause sampling errors or a loss of lock in the clock recovery of a repeater. Thus the spectral content of the jitter as well as its magnitude is of interest. Besides not being able to measure low-frequency jitter, the circuit of Figure 7.6 cannot operate if it cannot lock onto the fundamental clock signal. Section 7.4.7 describes another way to measure jitter and other timing impairments within a network.

Phase jitter is commonly specified by communications theorists as a power measurement in units of radians squared or cycles squared. (One cycle = 2π radians.) As indicated in Figure 7.7, phase jitter power is then a measure of the variance of the number of clock cycles or unit intervals (UIs) stored in the transmission link. In a physical sense jitter "power" has little meaning because it represents timing variations, not power. Some physical justification for expressing jitter as a power can be obtained by observing that the rms power σ_v^2 of the phase detector output signal is proportional to the rms phase jitter σ_ϕ^2:

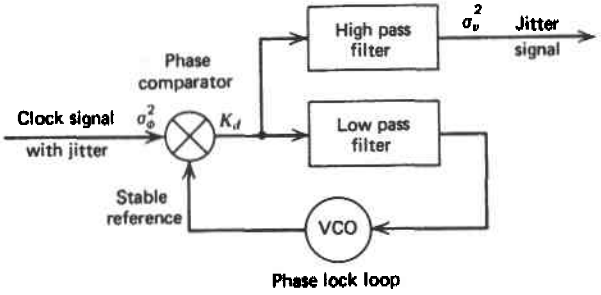


Figure 7.6 Circuit for measuring timing jitter.

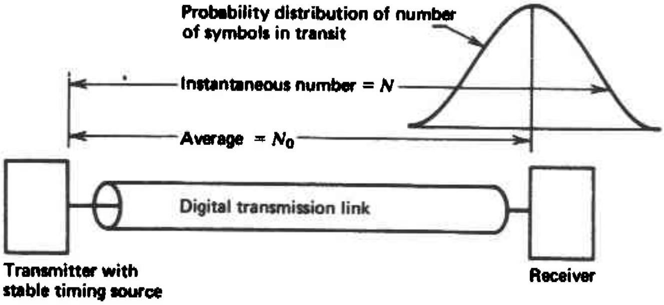

Probability distribution of number of symbols in transit
Instantaneous number = N
Average = N_0
Digital transmission link
Transmitter with stable timing source
Receiver

Figure 7.7 Phase jitter modeled as the variance in the number of symbols "stored" in the transmission link.

$$\sigma_{\nu}^2 = K_d^2 \sigma_{\phi}^2 \quad (\mathrm{V}^2) \tag{7.1}$$

where K_d is the phase detector gain factor in volts per radian.

Example 7.1. Given an rms phase jitter of 10.7 dB relative to one UI, what is the standard deviation of the phase offset?

Solution. The variance of the signal phase is determined as $\sigma_{\phi}^2 = 10 \exp(10.7/10) =$ 11.76 UI squared. Hence, the standard deviation is $(11.76)^{1/2} = 3.43$ UI (symbol intervals). Since 68% of a normal probability distribution lies within one standard deviation, the phase of this signal is within ±3.43 symbol intervals for 68% of the time. One percent of the time the signal phase will be outside 2.6 standard deviations, or ±8.9 symbol intervals.

If phase jitter arises as a result of additive Gaussian noise on a stable signal, the phase noise can be approximated as

$$\sigma_{\phi}^2 = \frac{\sigma_n^2}{2P_s} \quad (\mathrm{rad}^2) \tag{7.2}$$

where σ_n^2 = additive noise power
P_s = signal power

Equation 7.2 is the basic equation of phase jitter produced by additive noise on a continuous sinusoid [5]. When timing is extracted from a data signal, the timing information is usually not continuous. The distinction is not important because jitter produced by additive noise is normally insignificant compared to other sources [6]. For an analysis of phase jitter produced by regenerative repeaters operating on randomly occurring timing transitions, see reference [7].

TABLE 7.1 Maximum Instability of DS1 Customer Interface

Frequency Band	Maximum Peak-to-Peak Variation
< 10 Hz	28 UIs in 24 hr
< 10 Hz	5 UIs in 15 min
10 Hz–40 kHz	5 UIs
8 kHz–40 kHz	0.1 UIs

As an example of a particular instability specification, Table 7.1 lists the wander and jitter specifications of a DS1 digital carrier (T1) interface to the public network [8]. Notice that the higher frequency jitter specification allows deviations in pulse centers of only 5%, which effectively specifies the accuracy of the data sample clock. Larger variations are allowed for lower frequency instabilities because the clock recovery circuits can track the changes to maintain a good sample clock.

7.1.5 Systematic Jitter

An original analysis of jitter in a chain of digital regenerators was reported by Byrne, Karafin, and Robinson [3]. Figure 7.8 shows the basic model of their analysis. Each of the regenerative repeaters in a T-carrier line extracts timing from the received waveform and passes that timing on to the next regenerator as a transmit clock. Because of implementation imperfections (primarily intersymbol interference) in the timing recovery circuits, jitter produced by repeaters is dependent on the data patterns. One worst-case pattern produces an extreme phase lag. Another pattern produces an extreme phase lead. When the data pattern shifts from one worst case to the other, a phase ramp occurs. Because every repeater has the same basic implementation, the jitter produced by individual repeaters tends to be coherent. The systematic nature of this jitter makes it the most significant source of accumulated line clock jitter at the end of a chain of repeaters.

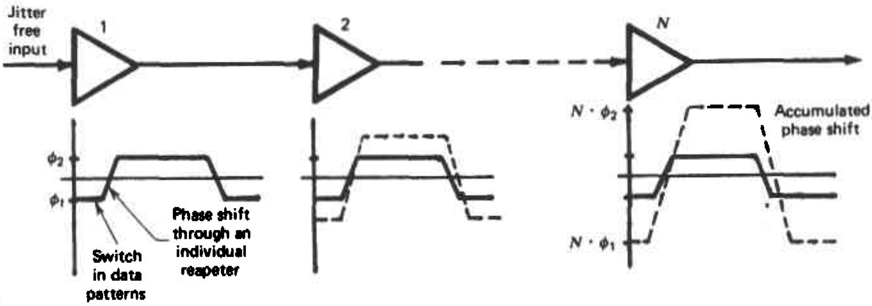


Figure 7.8 Model of systematic jitter in a string of regenerative repeaters: Φ_2 = phase shift produced by worst-case data pattern for phase lead; Φ_1 = phase shift produced by worst-case data pattern for phase lag.

As indicated in Figure 7.8, the last repeater in the chain experiences a large phase ramp equal to the number of repeaters times the phase shift of each individual repeater. This phase ramp represents an abrupt change in the clock frequency that may cause bit errors or a complete loss of synchronization. Thus there is a limit to the number of repeaters that can be used without jitter removal. For more analyses of jitter accumulation including the combined effects of systematic jitter and random perturbations, see references [9], [10], and [11].

7.2 TIMING INACCURACIES

In the preceding section the nature of certain instabilities or transient variations in timing was discussed. Although these variations represent shifts in the frequency of a line clock, the shifts are only temporary and can be absorbed by elastic stores. In some instances digital communications equipment using autonomous frequency sources must be interconnected. When this happens, the clock rates of the two systems are never exactly the same, no matter how much accuracy is designed into the frequency sources. An offset in the two clocks, no matter how small, cannot be reconciled by elastic stores alone.

In the preceding section, channel bank loop timing was mentioned as an example of how remote terminals are synchronized to a digital switch. When the remote terminal is another digital switch using its own frequency source as a reference, a different situation results. As shown in Figure 7.9, the outgoing clock for each direction of transmission is defined by the local switch clock. Thus the incoming clock at each switch interface contains not only transmission-line-induced jitter but also a small and unavoidable frequency offset.

7.2.1 Slips

As indicated in Figure 7.9, the interface of each incoming digital link necessarily contains an elastic store to remove transmission link timing jitter. The elastic store at the first digital switch is written into by the recovered line clock but *read from* the local rate R_1. If the average rate of the recovered line clock R_2 is different from R_1, the elastic

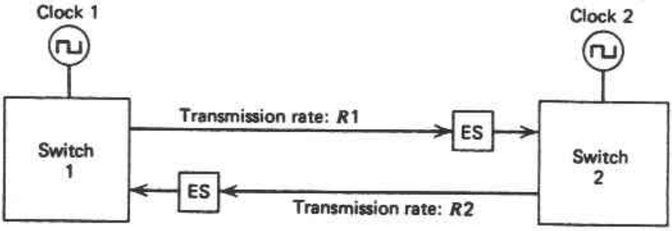


Figure 7.9 Connections between autonomously timed digital switches.

store will eventually underflow or overflow, depending on which rate is larger. When R_2 is greater than R_1, the elastic store at the first digital switch overflows, causing a loss of data. If R_2 is less than R_1, the same elastic store underflows, causing extraneous data to be inserted into the bit stream entering the switch. Normally, the extraneous data are a repetition of data bits already transferred into the switch. Disruptions in the data stream caused by underflows or overflows of an elastic store are referred to as "slips."

Uncontrolled slips represent very significant impairments to a digital network because they generally cause a loss of frame synchronization. Therefore, slips are allowed to occur only in prescribed manners that do not upset framing. One general approach to controlling the slips is to ensure that they occur only in the form of a repetition or deletion of an entire frame. Thus the time slot counters and framing logic associated with the multiplex group remain synchronized. Controlled slips comprising entire frames can be assured by using elastic stores with at least one frame of storage. As a slip occurs, the storage level in the elastic store is effectively increased or decreased by a full frame. Rather than actually inserting or deleting a frame of information, the desired effect is achieved more easily by indexing address pointers in a random-access memory. Such a system is shown in Figure 7.10.

The elastic store in Figure 7.10 operates by sequentially writing input information into memory addresses corresponding to individual TDM channels. Data for individual output channels are obtained by reading the same addresses in the same sequential manner. Ideally, if there is no offset between the clock rates, the read times of each channel occur midway between write times for corresponding channels. The elastic store then has the capability of absorbing transmission delay variations up to one-half of a frame time.

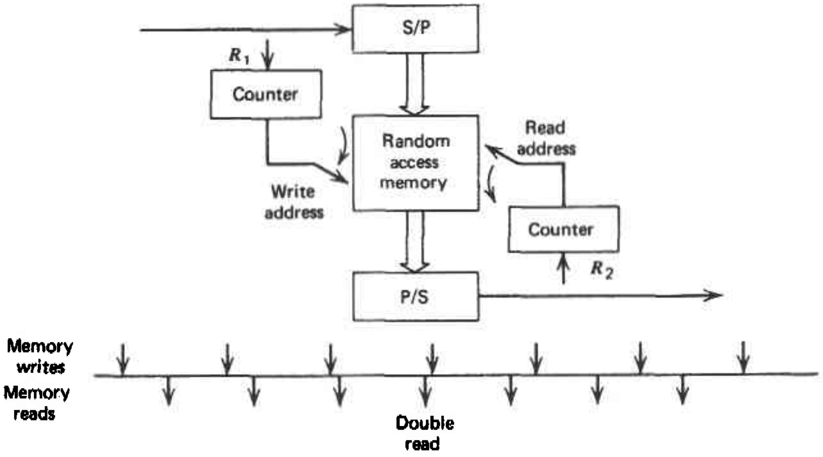


Figure 7.10 Elastic store operation with a one-frame memory.

The timing diagram in Figure 7.10 depicts an exaggerated timing offset in which the switch clock R_2 is greater than the incoming clock R_1. As indicated, the read times catch up gradually with the write times until a "double read" occurs. At that time the information retrieved for each channel is a repetition of the information retrieved for the previous outgoing frame. Although write and read times for only one channel are shown, the corresponding times for all other channels have the same relationship. Thus all channels slip together. Notice that R_1 is greater than R_2, a slip occurs when a "double write" on all channels causes the information in the previous incoming frame to be overwritten.

The elastic store operation depicted in Figure 7.10 is very similar to the operation of a time slot interchange memory described in Chapter 5. This relationship is exploitable in a TST switch where the inlet memory can provide both the elastic store function and the time switching function. When the two functions are combined, slips generally occur at different times for different channels. Nevertheless, individual channels maintain proper frame alignment since each channel is transferred through the inlet memory using dedicated memory addresses.

One attractive feature of using the inlet memory as an elastic store is that, when setting up a new connection, an internal switching time slot can be chosen so that the inlet memory read is halfway between inlet memory writes for the particular channel. Thus a slip in that connection will not occur for a long time, probably not until long after the connection is released. (With a clock inaccuracy of one part in 10^8, the time between slips in any one channel is 3.5 hr.)

One potential problem with the elastic store in Figure 7.10 occurs when write and read times nearly coincide. When both accesses to a single channel occur one after the other, transient timing instabilities can cause the two accesses to cross back and forth with respect to each other. Thus slips caused by double reads may follow slips caused by double writes and vice versa. To remedy this situation, some amount of hysteresis is needed in the counter adjustment process. The hysteresis, in turn, implies that additional storage is needed to defer the occurrence of one type of slip after a slip of the other type has recently occurred.

One means of implementing an elastic store with the desired hysteresis is to use two frames of storage as shown in Figure 7.11. For convenience, the elastic store is divided into an A-frame memory and a B-frame memory. The counter logic again accesses the memories in sequential fashion except that frames are written alternately into the A and B memories. Under normal operation, the memories are accessed in the same way for output data. When a slip is imminent, however, control logic causes the output channel counter to be reset so that the A memory is read twice in a row. This situation is depicted in the timing diagram of Figure 7.11, which again assumes that R_2 is greater than R_1. The important point to be noticed in the timing diagram is that after the counter adjustment produces a double read of memory A, the write and read times of each individual memory are approximately one frame time apart. Thus another adjustment can be deferred until the write and read accesses again drift one full frame time with respect to each other. The structure and mode of operation shown in Figure 7.11 describe the elastic store used for DS1 signal interfaces to AT&T's (now Lucent's) No. 4 ESS [12].

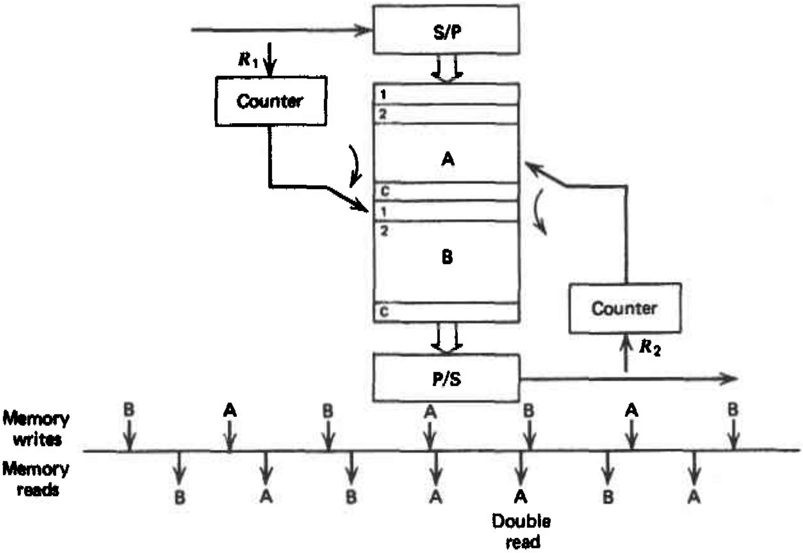


Figure 7.11 Elastic store with a two-frame memory.

Slip Rate Objectives

If the difference between an elastic store's input data rate and its output data rate is ΔR,* the time between slips is

$$\Delta T = \frac{N}{\Delta R} \tag{7.3}$$

where N is the number of bits that get dropped or repeated whenever a slip occurs. Normally, a slip involves a full frame of data, in which case the time between slips is determined as

$$\Delta T = \frac{1}{\Delta F} \tag{7.4}$$

where ΔF is the difference in frame rates.

As long as slips are controlled so they do not disrupt framing,† their only effect is an infrequent repetition or deletion of the information within affected TDM channels. The audible effect of slips on a digitized voice signal is an occasional "click." Only one slip in 25 produces an audible click in PCM voice [1]. Voice signals can therefore tolerate several slips per minute [13].

*Typically, clock offsets are specified in relative terms (e.g., one part in 10^6). A clock that is accurate to P parts per million (ppm) has a maximum offset of $\Delta R = R \cdot P/10^6$.

†With single-frame slips, framing is disrupted in the sense that framing pattern is shifted. Thus, the frame sequence has to be reacquired, but this process is simplified by the fact that the location of the framing bits and the integrity of the message channels are maintained.

A more subtle and troublesome aspect of slips occurs when a digitized channel carries voiceband data. High-speed data modems for the analog telephone network use QAM modulation with coherent detection in the receiver. Since these modems are particularly sensitive to phase shifts, they are particularly vulnerable to slips. An 8-bit slip in a digitized modem signal using a carrier of 1800 Hz causes an instantaneous phase shift of 81°. Obviously, a phase shift of this size causes a data error, but more importantly, it upsets the carrier recovery circuitry in the receiver and causes multiple errors. A single slip can upset the operation of some voiceband modems for several seconds [14, 15].

Characterizations of the effects of slips on Group 3 facsimile equipment [16] reveal that a single slip can cause the loss of four to eight scan lines without an error report. Sometimes the loss of the lines is not immediately evident to the reconstructed image. Diagonal lines, however, readily reveal missing vertical space.

Encrypted traffic (voice or data) is more susceptible to slips since the encryption–decryption process usually relies on bit-synchronous scramblers and unscramblers. When the bit count is altered by insertion or deletion of bits in a time slot, counters in the source and destination become unsynchronized. At best, the decryption process causes every slip to be audible. At worst, unintelligible speech or data result until the unscrambler is resynchronized. Another important aspect of encrypted communication is transmission of encryption keys or indexes to encryption keys. If synchronization is lost for some reason, it may be necessary to resynchronize the encryption keys, thereby compromising security.

When a digital transmission link is being used to transmit data directly, the effect of a slip may not be any more significant than a single channel error. Most data communications receiving equipment requests a complete retransmission of any block of data not satisfying certain redundancy checks. Thus one error is as bad as many errors or a complete loss of data. The effect of the slip will be more significant, however, if the communications protocol [e.g., the DDCMP of Digital Equipment Corporation (Compaq)] relies on byte count procedures to delimit message blocks. Insertion or deletion of data by the network causes the receive counter to become unsynchronized, and the normal exchange of information is disrupted until the loss of synchronization is recognized.

From the foregoing considerations for data transmission, the slip rate objective for the AT&T network and adopted by Bellcore and ANSI for North America was set at one slip in 5 hr for an end-to-end connection [17–19]. Since slips can occur at multiple points within a network, the objective for slips at individual trunk and switching interfaces was set at one slip every 20 hr.

Example 7.2. Determine the relative accuracy requirements of two independent clocks to maintain a mutual slip rate objective of one slip in 20 hr. Assume a frame rate of 8 kHz as in PCM voice signals.

Solution. The slip rate objective implies that the frame rate produced by one clock can be different than the frame rate produced by the other clock by no more than

$$\Delta F = \frac{1}{20 \times 60 \times 60} = 1.39 \times 10^{-5} \text{ slips per second}$$

Since there are 8000 frames per second, the relative accuracy is determined as

$$\frac{1.39 \times 10^{-5}}{8000} = 1.7 \times 10^{-9} \text{ slips/frame}$$

Hence the clocks must be accurate to 1.7 parts in 10^9.

Because Example 7.2 determines only a maximum relative inaccuracy, the absolute inaccuracy of each individual clock must be less than $(1.7 \times 10^{-9})/2$, or 0.85 parts in 10^9.

7.2.2 Asynchronous Multiplexing

In the preceding section certain aspects of network synchronization were discussed that implied the need for clock synchronization to prevent a loss of data by way of slips. In this section a procedure referred to as "pulse stuffing" is discussed that avoids both slips and clock synchronization. The term pulse stuffing can be somewhat misleading since it implies that pulses are inserted into the line code to make timing adjustments. Actually, pulse stuffing involves only the data stream and is independent of the line code or modulation system in use. Pulse stuffing is a term commonly used in North America while the same concept is referred to as "justification" in Europe.*

The basic concept of pulse stuffing involves the use of an output channel whose rate is purposely higher than the input rate. Thus the output channel can carry all input data plus some variable number of "null," or "stuff," bits. The null bits are not part of the incoming data. They are inserted, in a prescribed manner, to pad the input data stream to the higher output rate. Naturally, the extraneous null, or stuff, bits must be identifiable so that "destuffing" can recover the original data stream.

The practice of pulse stuffing arose when the initial digital TDM hierarchy (Table 1.10) was defined. At this time there were only isolated digital transmission links within the network that precluded synchronizing them to a common clock. When it came time to combine lower rate tributaries (e.g., DS1s) into higher level signals (e.g., DS2s or DS3s), the multiplexing procedure necessarily had to accommodate tributaries operating at slightly different rates. The generic term for combining these unsynchronized signals is asynchronous multiplexing. In this context, "asynchronous" refers to multiplexing of unsynchronized tributaries into a higher level signal (using pulse stuffing). It does not refer, in any way, to a means of transmission. The higher level signal is always carried on a synchronous transmission link.

As the amount of digital equipment in the network grew and more and more of it became interconnected, both the means and the necessity for a different form of mul-

*Justification is the printing industry practice that aligns the right side of lines of text by inserting a variable amount of space within the line. As you will see in the following discussion, aligning individual tributaries to the rate of a higher level multiplexer is conceptually the same process.

tiplexing arose. The chosen approach is referred to as synchronous multiplexing (SONET) in North America and Synchronous Digital Hierarchy (SDH) in the rest of the world. The principles of SONET and SDH are described in conjunction with fiber systems in the next chapter.

Pulse-Stuffing Concepts

As a starting point for understanding the need for pulse stuffing, consider the simple two-channel, bit-interleaved multiplexer in Figure 7.12. As indicated, within any string of even-numbered bits in the multiplexer output, the number of bits carried in each subchannel is necessarily identical. Thus the rates of the subchannels are also identical. If the two input data streams are running at different rates, the output can be synchronized to one of the channels but not both. Thus slips would necessarily occur in at least one of the tributaries.

As a simplified example of pulse stuffing, Figure 7.13 shows a two-channel, bit-interleaved format as before but with the additional detail needed to allow adjustments of the information flow within each subchannel. As indicated, the multiplexed output is formatted into 10-bit master frames with 5 bits assigned to each subchannel. The first 3 bits in each subchannel of each master frame always carry data from the respective tributaries. The fourth bit in each subchannel (C_1 and C_2) specify whether the last bits (S_1 and S_2) carry data or are stuff bits. When C_1 is a "1," a bit is stuffed; otherwise S_1 carries tributary data. Hence each master frame can carry 3 or 4 bits from each tributary. If, on average, each tributary sends 3.5 bits during a master frame, variations of ±14% in the tributary clock rates can be accommodated.

An important point to notice about an asynchronous multiplexer is that the output frame structure is unrelated to the frame structure of the lower level inputs. As far as the higher level multiplexer is concerned, each input signal is merely a serial bit stream with no particular structure assumed. Framing bits in the lower level multiplex signals are transmitted right along with the information bits. After the higher level signal is demultiplexed and the tributaries are unstuffed, framing of the lower level signals must be established for further demultiplexing.

Although pulse stuffing can be implemented with a variety of higher level framing formats, the generally most desirable features of a pulse-stuffing format are identified as follows:

1. The use of fixed-length master frames with each channel allowed to stuff or not to stuff a single bit in the master frame

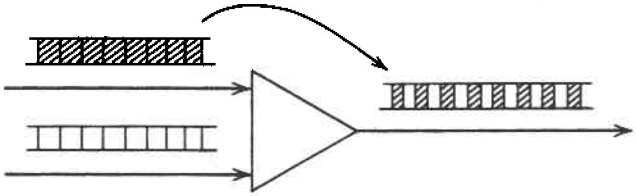


Figure 7.12 Two-channel multiplexer showing equal output data rates for each input.

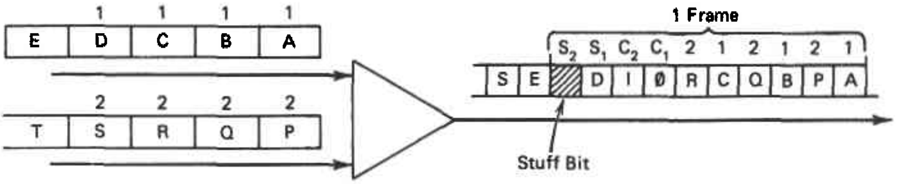


Figure 7.13 Simplified pulse-stuffing example.

2. Redundant stuffing specifications
3. Noninformation bits distributed across a master frame

Timing offsets are generally quite small, so only small adjustments from a single occasional stuff bit are required. Thus large numbers of tributary bits can be combined into a master frame with one specific bit position identified as the S bit. Nominally, approximately one-half of the master frames contain the maximum number of information bits N_M, and the other half contain $N_M - 1$ information bits.

The purpose of a pulse-stuffing operation is to prevent a loss of data when two interconnected digital transmission links are unsynchronized with respect to each other. If single bit errors can cause a stuff bit to be interpreted as information (or vice versa), the basic objective is lost. Furthermore, notice that an erroneous interpretation of a stuff code causes embedded, lower level multiplex signals to lose framing. For these reasons, the interpretation of C bits must be encoded redundantly. Under the assumption that channel errors are random, the probability of misinterpreting a stuff code is

$$P_F = \sum \binom{2n+1}{n+1} p^{n+1}(1-p)^n \tag{7.5}$$

where p = probability of a channel error
n = number of correctable stuff code errors ($2n + 1$ bits in a stuff code)

Information bits should be distributed across a master frame for several reasons. First, by separating these bits as much as possible, errors in redundant stuffing specification bits (C bits) are more likely to be independent. If the specification bits are too close together and burst errors are prevalent, the redundant encoding is of little use. Second, by distributing noninformation bits, the irregularity of information flow is minimized. When higher level multiplex signals are demultiplexed, a clock for each individual lower level signal needs to be derived from the irregular information rate in each channel. Generation of a suitably stable clock synchronized to the information rate is simplified if information bursts or gaps are minimized. Furthermore, elastic stores needed to smooth out the information rates are smaller when the length of information gaps is minimized.

M12 Multiplexer

An example of a higher level multiplexing format is provided in Figure 7.14. This is the format used for 6.312-Mbps DS2 signals in the North American digital hierarchy.* A DS2 signal is derived by bit interleaving four DS1 signals and adding the appropriate overhead bits.

A DS2 master frame is 1176 bits long. Of these there are 1148 information bits (287 per channel), 11 framing bits (M_0, M_1, F_0, F_1), 12 stuffing control bits (C_1, C_2, C_3, C_4), 4 S bits (S_1, S_2, S_3, S_4), and an alarm bit X. Since an S bit can be a null bit or an information bit, each channel can send 287 or 288 bits in a master frame. An S bit is designated as an information bit if all three of the corresponding C bits are 0. The S bit is a null (stuff) bit if all three corresponding C bits are 1. Obviously, this encoding procedure allows for single error correction in the stuffing control bits.

The first level of framing is established by the alternating F_0, F_1, F_0, . . . pattern. Notice that exactly 146 bits separate the F_0 and F_1 bits. Another level of framing for identifying the C and S bits is established by the M_0 and M_1 bits. A fourth M bit (X) is not used for framing and therefore can be used as an alarm service digit. Similar frame structures exist for other higher level digital signals. Figures 7.15–7.18 show the structures for DS3, DS4, DS1C, and the E2 second-level digital signal of the ITU, respectively.

Example 7.3. Determine the minimum and maximum input channel rates accommodated by an M12 multiplexer. Also determine the rate of DS1 misframes caused by an erroneous interpretation of a stuffed bit. Assume the bit error rate is 10^{-6}.

Solution. The maximum information rate per channel is determined as

$$\frac{6.312 \times 288}{1176} = 1.5458 \text{ Mbps}$$

The minimum information rate per channel is determined as

$$\frac{6.312 \times 287}{1176} = 1.5404 \text{ Mbps}$$

Since there are three possible combinations of two errors in the C bits, the probability of misinterpreting an S bit is closely approximated by $3 \times (10^{-6})^2 = 3 \times 10^{-12}$. The duration of each master frame is 1176/6.312 = 186 μsec. Thus the rate of misframes per DS1 signal is

$$\frac{3 \times 10^{-12}}{186 \times 10^{-6}} = 0.016 \times 10^{-6} \text{ misframes per second}$$

which is equivalent to one misframe every 2 years.

*DS2 signals are no longer transmitted as individual signals. They only exist as an intermediate level between 28 DS1s and a DS3 signal.

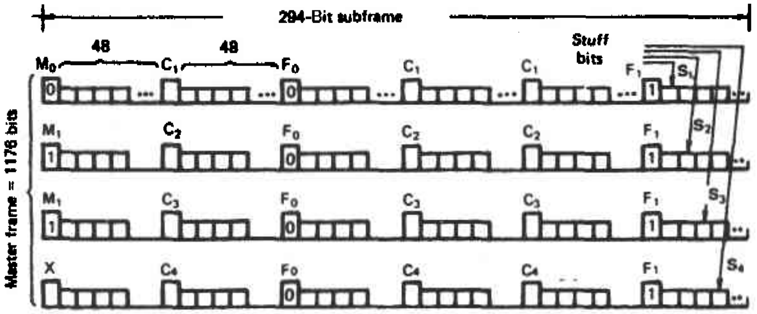


Figure 7.14 Frame format of DS2 digital signal. Stuffing occurs in channel *i* when the previous C_i bits = 111; X is an alarm bit that equals 1 for no-alarm condition. Framing is established by the $F_0F_1F_0$ . . . sequence with 146 intervening bits.

Example 7.3 demonstrates that the tolerance of a 1.544-MHz DS1 clock is −3.572 to +1.796 kHz. Thus the relative accuracy between the DS1 and DS2 clocks must be 1.796/1544, or only 1 part in 860. This relatively large timing tolerance is much greater than what is required for reasonable clock and line instabilities. The timing adjustment capabilities and unsymmetric tolerance range were chosen out of a desire to (1) minimize DS2 reframe times, (2) provide a line clock that is a multiple of 8 kHz, and (3) minimize waiting time jitter [20].

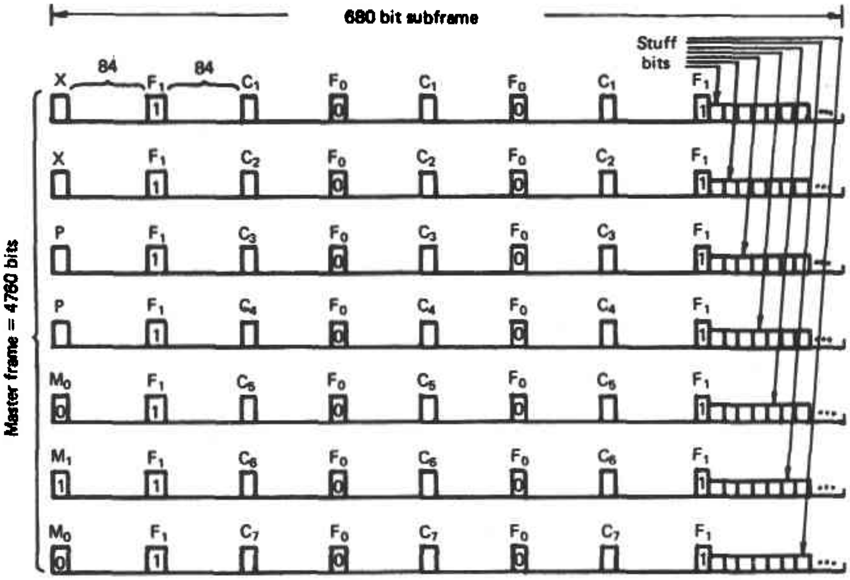


Figure 7.15 Frame format of DS3 digital signal; P is even parity over all message bits in the previous master frame. Stuffing occurs in channel *i* when the previous C_i bits = 111. The X (alarm) bits and the P bits must be 11 or 00 so the $M_0M_1M_0$ sequence can identify the end of the master frame.

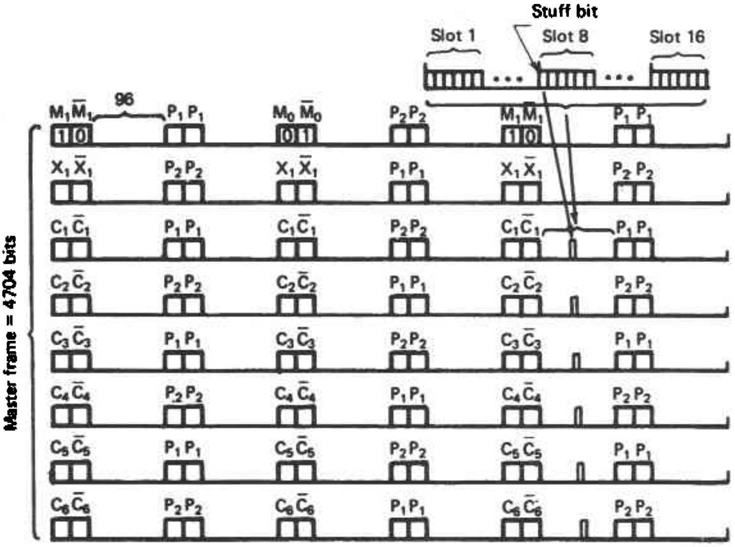


Figure 7.16 Frame format of DS4: C_i bits = 111 implies stuff the eighth message bit position for channel i following the last C_i; P_1 is even parity over the 192 previous odd-numbered message bits; P_0 is even parity over the 192 previous even-numbered message bits.

Elastic Store Size Requirements

A functional model of an M12 multiplexer is shown in Figure 7.19. Associated with each lower level (DS1) input is an elastic store to hold incoming data until it is transferred to the higher level (DS2) output. The elastic stores serve two purposes: to remove the arrival jitter of the incoming data and to hold data for the proper time slots. In addition to generating framing, the control logic of the multiplexer monitors the

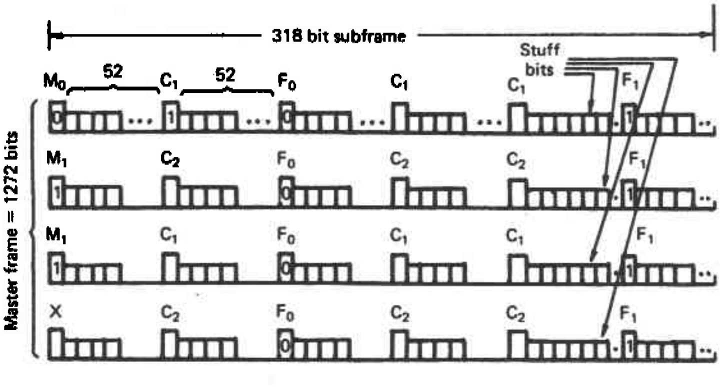


Figure 7.17 Frame format of DS1C digital signal. Stuffing occurs in channel i when the previous C_i bits = 111; X is an alarm bit that equals 1 for no-alarm condition. Framing is established by the $F_0F_1F_0$ . . . sequence with 158 intervening bits.

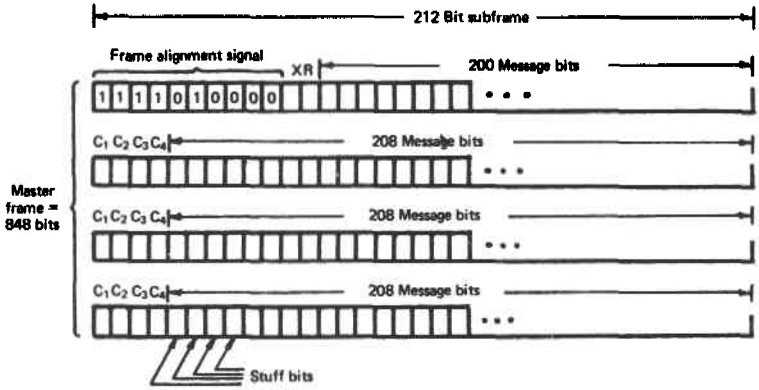


Figure 7.18 Frame format of second-level digital signal of ITU-T (E2). Stuffing occurs in channel *i* when the previous C_i bits are 111.

storage level (which serves as a phase comparator) of each elastic store and initiates a stuffing operation whenever the elastic store is less than half full. Conversely, no stuffing occurs when the elastic store is more than half full.

Since stuffing can occur at only certain times and only at a certain maximum rate, the elastic store must be at least as large as the peak jitter (peak phase offset) of the incoming signal. As discussed at the beginning of this chapter, jitter accumulates along the entire length of a repeatered transmission link. Thus longer links require larger elastic stores if slips are to be prevented.

As an example of the relationship between line length (number of repeaters) and number of bits of elastic storage needed by an M12 multiplexer, refer to Figure 7.20. This figure was submitted by AT&T to the CCITT special study group on jitter [21]. The analysis is an extension of the systematic jitter analysis of reference [3].

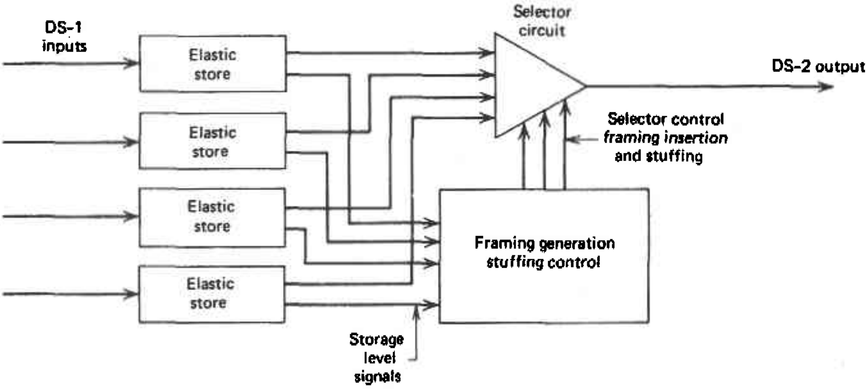


Figure 7.19 Functional diagram of a M12 multiplexer.

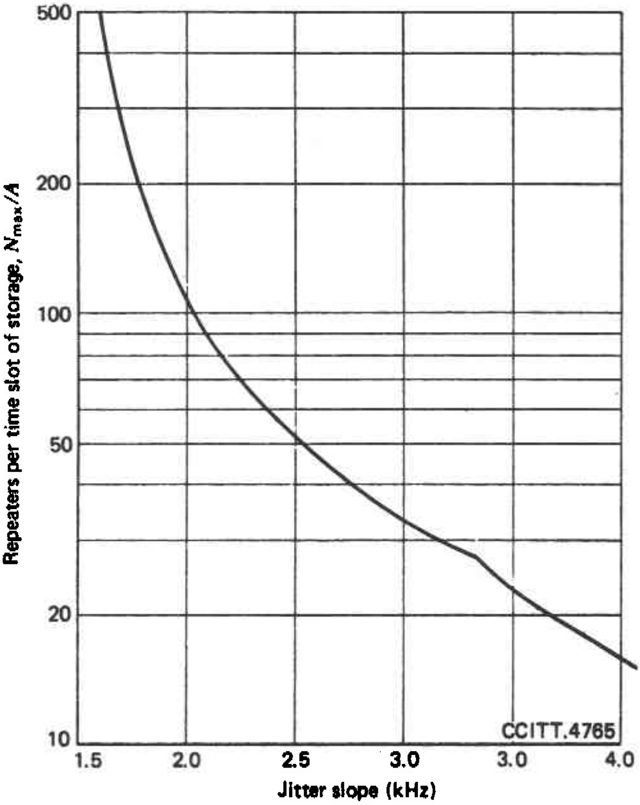


Figure 7.20 Maximum number of regenerative repeaters as a function of elastic store size and jitter slope; A = number of cells in elastic store.

The abscissa of Figure 7.20 is the maximum-phase slope produced by a clock recovery circuit in the presence of a worst-case shift in data patterns (worst-case systematic jitter). Since phase slope is nothing more than frequency offset, the required elastic store size can be determined as the maximum phase slope times its maximum duration. Since the total phase slope is proportional to the number of repeaters, Figure 7.20 displays the maximum number of repeaters per storage cell in an elastic store versus the jitter slope of an individual repeater when making worst-case timing transition.

As an example, first-generation T1 repeaters produce a worst-case slope of 2.4 kHz. M12 multiplexers allocate 5 bits of storage to input phase jitter (3 more bits are included for implementation ease and waiting time jitter). From Figure 7.20 it can be seen that the ratio of N_{max} to A is 56, which implies that $N_{max} = 56 \times 5 = 280$ repeaters.*

*The performance of long T1 lines is not as much of a concern as it once was because most long-distance DS1 circuits are now embedded in fiber links, which have much greater repeater spacing.

7.2.3 Waiting Time Jitter

When demultiplexing a higher level TDM data stream, it is necessary to generate a clock for each derived subchannel. Because the subchannels are transferred (or transmitted) as a synchronous data stream, the derived clock must be continuous. Derivation of subchannel clocks is complicated by the insertion into TDM data streams of overhead bits that create gaps in the bit arrival times. Irregularity in the data arrival rate caused by these gaps is referred to as *waiting time jitter*.

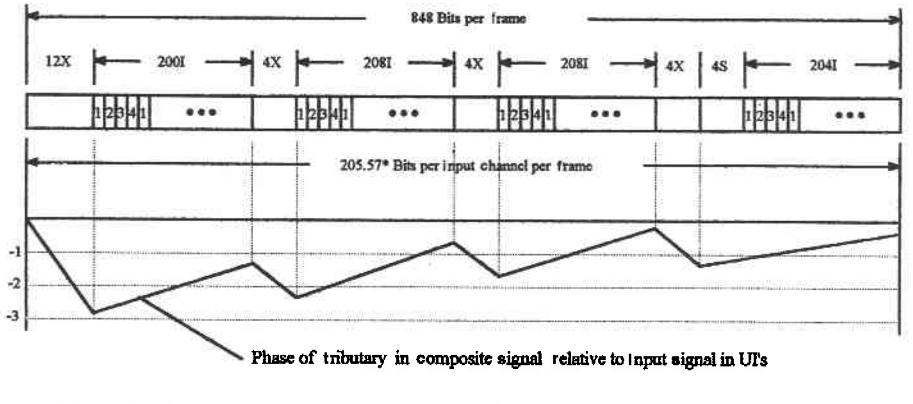
Most of the overhead bits (e.g., framing bits, parity bits, stuffing control C bits) occur on a regular and predictable basis. The waiting time jitter caused by these gaps (sometimes referred to as *mapping jitter**) can be eliminated easily with an elastic store and an output clock derived from the incoming line clock. For example, a single PCM channel clock at 64 kbps can be derived from a 1.544-Mbps T1 line clock by multiplying by 8 and dividing by 193. Mapping jitter as it occurs in a E1 signal mapped into an E2 signal is depicted in Figure 7.21. In this figure the phase of the tributary data mapped into the higher level signal is shown relative to an unmapped (continuous) data clock at the same average rate. Notice that the phase of the tributary falls behind the reference during periods of data gaps but catches up during data fields because the average tributary data rate during these fields is slightly higher than the reference. Notice further that the fill level of an elastic store does not get perfectly reconciled at the end of a single E2 master frame. This result occurs because, on average, a nonintegral number of bits of a tributary are carried in a master frame.

In contrast to mapping data rate variations, waiting time jitter produced by pulse stuffing is significantly more difficult to deal with. The difficulty arises because waiting times produced by stuffed pulses are irregular and unpredictable. For this reason the subchannel output clocks derived from a pulse-stuffed TDM line must be derived independently and only from the average arrival rate of each channel's data—not from the higher level TDM rate!

Output clocks from M12 demultiplexers are generated using jitter-removing elastic stores, as shown in Figure 7.22. If large elastic stores and very slowly adjusted output clocks are used, most of the jitter can be removed. Unfortunately, waiting time jitter has frequency components down to zero frequency so the jitter (e.g., wander) can never be eliminated entirely. However, the jitter can be confined to as low a band of frequencies as desired by using a large enough elastic store. Figure 7.23 depicts time interval errors resulting from the pulse stuffing–destuffing process. The key point to notice is a full bit of offset between the source data clock and the mapped data can occur at any time but an adjustment can only be made when the next stuff opportunity occurs—hence the term waiting time jitter.

Waiting time jitter is basically a function of how often pulses are stuffed, but it is also dependent on the ratio of actual stuffs to stuffing opportunities. If the input clock

*Some references include mapping jitter to include waiting time jitter. Here, mapping jitter is used to represent repetitive data rate variations that occur when both the tributary and the higher level transport signals are respectively at their precise rates. Thus, waiting time jitter only occurs as a result of frequency adjustments.


848 Bits per frame
12X
200I
4X
208I
4X
208I
4X
4S
204I
1 2 3 4 1 •••
205.57* Bits per input channel per frame
-1
-2
-3
Phase of tributary in composite signal relative to Input signal in UI's

X: Overhead bit position I: Information bit position for a tributary S: Timing adjustment (stuff) bit position

* Every composite frame carries 205 information bits per tributary plus one possibly additional information bit per tributary in an S bit position depending on need.

Figure 7.21 Mapping Jitter of a CCITT E1 in a CCITT E2 signal.

is jitter free, the output jitter peaks when one-half of the opportunities are used. From the point of view of maximum tolerance for clock offsets, a stuffing ratio of $\frac{1}{2}$ is ideal. To reduce the waiting time jitter, however, stuffing ratios of approximately $\frac{1}{3}$ are often used. For a thorough analysis of waiting time jitter, see reference [22]. As an example of waiting time jitter dependence on stuffing (justification) ratios, see Figure 7.24 obtained from reference [21]. The abscissa of Figure 7.24 represents the ratio of stuffs to opportunities while the ordinate is jitter power produced by a single pulse-stuffing process. The jitter power is expressed in decibels relative to one slot squared (a slot is an older term for a unit interval, U1). Curve A shows the output jitter pro-

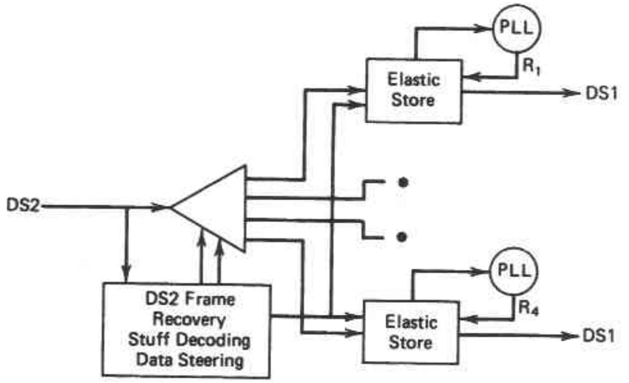


Figure 7.22 Functional diagram of M12 demultiplexer/desynchronizer.

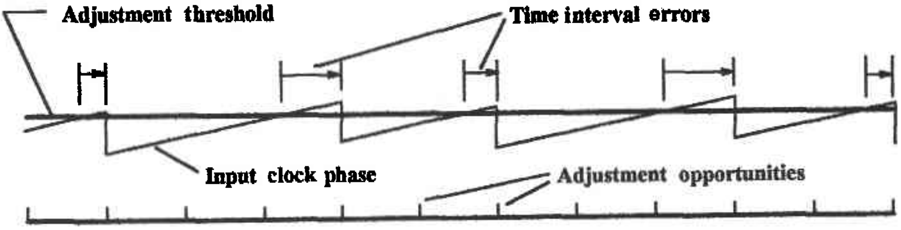


Figure 7.23 Time interval errors produced by adjustment waiting times.

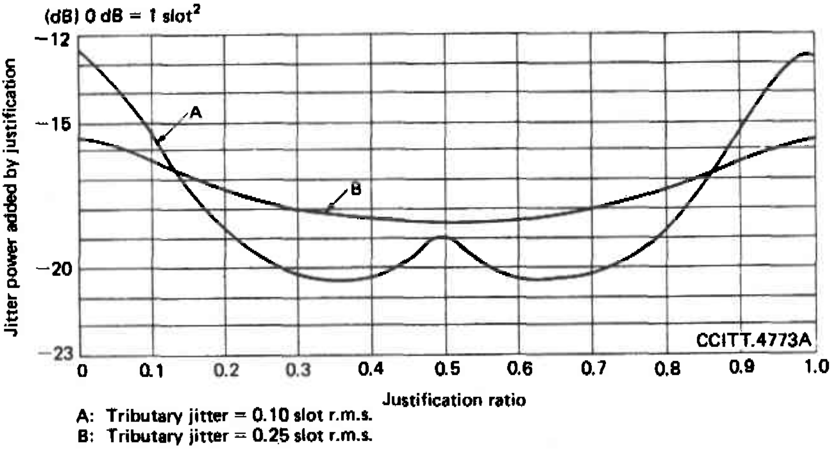


Figure 7.24 Waiting time jitter dependence on justification ratio.

duced when the input jitter is –20 dB (0.1 UI rms). Curve *B* shows the output jitter when the input jitter is –12 dB (0.25 UI rms). Figure 7.24 shows jitter produced by a single multiplexer. From measured data [22], a good order-of-magnitude estimate of the waiting time jitter accumulated by N_m tandem M_{12} stuffing–destuffing operations is

$$\sigma_w^2 = \frac{N_m}{100}\,(\mathrm{UI})^2 \tag{7.6}$$

7.3 NETWORK SYNCHRONIZATION

As discussed in the preceding section, whenever a digital transmission link is connected to a digital switch, it is desirable to synchronize the two systems by having the transmission link obtain its timing from the switch. An obvious exception to this mode of operation occurs when a digital transmission link is connected to a digital switch on both ends. Generally, a transmission link in an all-digital network derives its timing from just one of the switches to which it is connected. If the other switch is not synchronized to the first in some manner, an unsynchronized interface necessarily results. This section is concerned with network synchronization as a whole, not simply the synchronization of a single interface. Basically, network synchronization involves synchronizing the switches of the network. The transmission links can then be synchronized automatically by deriving timing directly from a switching node.

There are two basic reasons for paying assiduous attention to the timing requirements of a digital network. First, the network must prevent uncontrolled slips that could produce misframes, inadvertent disconnects, and cross connects. It is generally very difficult or very expensive to prevent slips altogether. Thus a second aspect of a network timing plan requires establishing a maximum rate of controlled slips as part of the end-to-end circuit quality objectives.

Synchronizing private networks is sometimes difficult because the network topologies are not designed with network synchronization in mind, and the switching equipment (PBXs) are not designed to provide synchronization to other nodes. Furthermore, the private networks often interface to multiple carriers in multiple locations. Determining which signals to synchronize to, particularly on a dynamic basis when a reference signal becomes unavailable, is exceptionally difficult.

There are six basic approaches used, or considered for use, in synchronizing a digital network:

1. Plesiochronous
2. Networkwide pulse stuffing
3. Mutual synchronization
4. Network master
5. Master–slave clocking
6. Packetization

7.3.1 Plesiochronous

A plesiochronous network does not synchronize the switches but merely uses highly accurate clocks at all switching nodes so the slip rate between the nodes is acceptably low. This mode of operation is the simplest to implement since it avoids distributing timing throughout the network. A plesiochronous network, however, implies that the smaller switching nodes carry the cost burden of highly accurate and redundant timing sources. As a compromise, large networks can be divided into subnetworks for timing purposes and use plesiochronous operations for inter-subnetwork synchronization and some other, more cost-effective, means of providing intra-subnetwork synchronization. As described in Section 7.5, the public telephone network in the United States uses plesiochronous synchronization at the upper levels.

Plesiochronous timing is also used to synchronize international digital network interconnections. In recommendation G.811 [23], the ITU has established the stability objectives for clocks of all international gateway digital switches. The stability objective of one part in 10^{11} implies that slips between international gateway switches will occur at a rate of one per 70 days. (This assumes one clock is positive one part in 10^{11} and another clock is negative one part in 10^{11}.)

7.3.2 Networkwide Pulse Stuffing

If all internal links and switches of a network were designed to run at nominal rates slightly higher than the nominal rates of the voice digitization processes, all voice signals could propagate through the network without slips by stuffing the information rate up to the local channel rate. None of the clocks would have to be synchronized to each other, and relatively coarse clock accuracies could be tolerated. At every interface between systems running under different clocks, however, the individual channels would have to be unstuffed from the incoming rate and stuffed up to the local or outgoing rate. In essence, the TDM links of the network would provide TDM channels through which user data flows at lower and variable rates, the differences being absorbed by internal pulse stuffing.

In contrast to pulse-stuffing operations of higher level multiplexers where all channels in a lower level digital signal are stuffed as a group, switching operations imply that each channel must be stuffed independently. The need for separate pulse-stuffing operations is illustrated in Figure 7.25, which depicts two voice signals being switched into a common TDM outlet link. Obviously, the bit rate R_3 of both output channels is identical. If the two channels originate in portions of the network running under different clocks R_1 and R_2, the pulse-stuffing adjustments must be made separately for each channel. The complexity of stuffing and unstuffing every 64-kbps channel at every network element would have been extremely expensive when the digital network began to take shape.

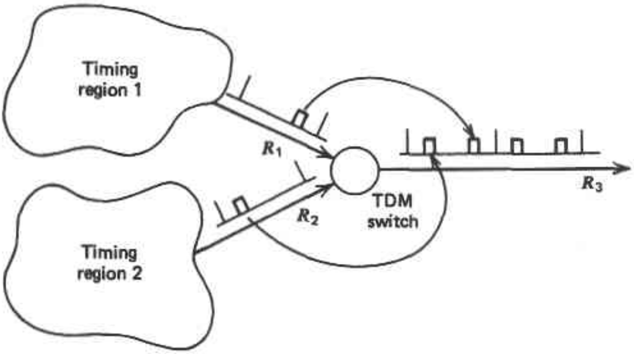


Figure 7.25 Switching two channels with different rates onto a common TDM output.

Although the availability of low-cost logic could minimize the cost aspect of networkwide pulse stuffing, other problems would occur. First, the 64-kbps clocks for speech reconstruction would be required for each channel and would contain relatively large amounts of waiting time jitter. Second, the network would no longer provide byte framing so the channel recovery process would have to also include byte framing logic. Byte boundaries of PCM data can be readily determined from statistical data patterns in the bit positions (e.g., the polarity bit), but other applications for the channels may require explicit byte boundary identifiers.

7.3.3 Mutual Synchronization

The two preceding sections discuss modes of operation for the network that do not involve synchronization of individual clocks. This section and the next two describe network timing plans that synchronize each individual clock to a common frequency. The first method, mutual synchronization, establishes a common network clock frequency by having all nodes in the network exchange frequency references as shown in Figure 7.26. Each node averages the incoming references and uses this for its local and transmitted clock. After an initialization period, the network clock normally converges to a single stable frequency. Under certain conditions, however, the averaging process can become unstable [24].

The main attractiveness of a mutually synchronized network is its ability to remain operational in spite of a clock failure in any node. The main disadvantages are the uncertainties of the exact average frequency and unknown transient behavior. Mutual synchronization has not been considered for the North American telephone network. In Great Britain, however, a hierarchical timing structure was once considered that utilized mutual synchronization within some portions of the network [25].

7.3.4 Network Master

Another method of synchronizing the network is shown in Figure 7.27. With this method a single master clock is transmitted to all nodes enabling them to lock onto a common frequency. As indicated, all network nodes are directly connected to the network master, implying the need for a separate transmission network dedicated to the

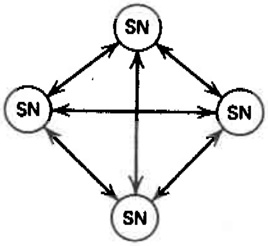


Figure 7.26 Mutual synchronization: SN, switching node.

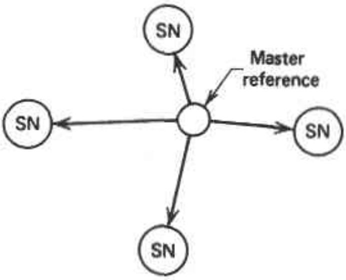


Figure 7.27 Network master synchronization: SN, switching node.

distribution of the reference. Reliability considerations also imply that alternate paths be provided to each node. Because of cost considerations for the separate timing network and reliability problems with reference distribution, a network master with direct transmission to each node is undesirable.

Something similar to a master synchronized network is evolving through the use of Global Positioning System (GPS) satellites for timing distribution to network nodes. Switching systems of telecommunications networks around the world are using GPS and other satellite systems to synchronize their switching office clocks. As the costs of the GPS receivers and suitably stable oscillators drop in cost, more and more network nodes are being timed from this master (highly accurate) source. Because CDMA digital mobile systems also lock to GPS, lower cost synchronization systems have also been developed that obtain GPS timing from the CDMA base stations in lieu of directly receiving GPS signals that often require outside antennas to receive from multiple GPS satellites simultaneously.

7.3.5 Master–Slave Synchronization

The main drawback to network master synchronization as described in the preceding section is its need for separate and reliable transmission facilities to every node. Figure 7.28 shows a network configuration that disseminates a master reference by way of the message links themselves. A network reference frequency is transmitted to a few selected higher level switching nodes. After these nodes synchronize their clocks to the reference and remove transmission link-induced timing jitter, the reference is passed on to lower level switches by way of existing digital links. The next lower level switches, in turn, synchronize to an incoming link from the higher level and pass timing on to another level of switches by way of their outgoing digital links. The process of passing the reference downward from one level to the next is referred to as "master–slave synchronization."

Since all switching nodes in the network are synchronized either directly or indirectly to the same reference, they all run at the same nominal clock rate. Thus slips should not occur. However, because of the different paths through which timing is disseminated, short-term frequency differences can occur between some nodes. If these

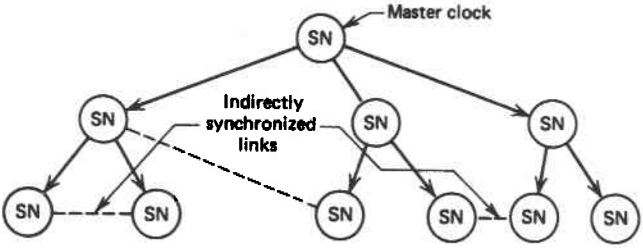


Figure 7.28 Master–slave synchronization.

nodes are synchronized indirectly, as shown in Figure 7.28, infrequent slips might occur. Furthermore, reliability considerations imply that backup clocks must be provided in all switches should the clock distribution system fail. When this happens, slips become more likely, but only after relatively stable backup clocks have had enough time to drift from the common reference frequency.

AT&T and the United States Independent Telephone Association (USITA) originally selected master–slave synchronization for the switched digital network in the United States [26]. The reference frequency was located in Hillsboro, Missouri, from which selected No. 4 ESS switching centers received their timing by way of dedicated transmission facilities. Synchronization of all other switches occurred by way of existing digital transmission links. As discussed in Section 7.4, the original plan has been changed to use plesiochronous synchronization at the highest level. Furthermore, as mentioned in the previous section, the availability of GPS timing sources is leading to more and more nodes at the higher level and, consequently, fewer and fewer slave nodes.

7.3.6 Packetization

The synchronization discussions of the five preceding sections have assumed implicitly that a synchronous, circuit-switched network was being considered, since prevailing digital voice networks operate in that manner. For completeness, however, another form of network must be mentioned—a packet-switched network.

As discussed in Chapter 10, packet-switched networks break up messages into identifiable blocks (packets or cells) of data. In between the blocks, the transmission links carry either idle codes or control messages. If all messages (control and data) are separated by a nominal interval of idle transmission, elastic stores can be reset in preparation for the next block. As long as each block is limited in length, the elastic stores can absorb clock differences and avoid losses of data. (In essence, slips occur in the idle codes.)

7.3.7 Network Timing Performance Measurements

After choosing a synchronization architecture for a network, it is necessary to be able to measure the quality of the timing signals within the network to qualify equipment, determine performance margins, isolate faulty equipment or transmission links, and possibly evaluate design alternatives. ANSI and ITU standards use the concept of time interval error (TIE) and maximum time interval error (MTIE) for these purposes. An additional measure of clock performance is a time variance (TVAR).

Maximum Time Interval Error

A TIE is the difference in delay between a timing signal and an ideal timing signal measured at the end of a particular time period (i.e., the error at the end of a time interval). An MTIE is the maximum peak-to-peak variation in TIE values that occur within a specified time interval. These concepts are illustrated in Figure 7.29. Figure 7.29*a* displays the TIE that occurs when measuring a perfectly stable but inaccurate timing signal. Because the frequency of the signal under test differs from the reference frequency by a constant value (Δf), the TIE is directly proportional to the measurement interval S. The error is determined by counting clock cycles in both signals and expressing the difference as the time required by the test signal to catch up (or fall back) to the current reference count. Thus the TIE produced by a constant-frequency offset is

$$\begin{aligned} \text{TIE} &= \frac{\Delta n}{f} \\ &= \frac{S(f+\Delta f) - S(f)}{f} \\ &= S\left(\frac{\Delta f}{f}\right) \qquad (7.7) \end{aligned}$$

where Δn is the accumulated difference in clock cycles, Δf is the frequency offset, and S is the measurement interval in seconds.

Because the TIE in Figure 7.29*a* increases monotonically with S, the MTIE and the TIE are identical. In contrast, Figure 7.29*b* depicts the TIE produced by a timing signal with no long-term frequency offset but some short-term instabilities (jitter and wander) as would occur if the reference was relayed through a network. As indicated, the TIE varies as a function of S but is bounded. The MTIE is the largest peak-to-peak difference in TIE values in a measurement interval S. The determination of MTIE with no long-term frequency offsets is also shown in Figure 7.30, where periodic samples of time interval errors are depicted.

Figure 7.29*c* displays the more general case where the signal under test contains both instability and a frequency offset. In this case the choice of the value for S is critical. If S is too small, jitter will mask the presence of the offset. If S is too large, the frequency offset dominates the TIE and MTIE measurements. Thus periodic measurements need to be recorded for a time history to fully characterize the timing imperfections.

The first three examples in Figure 7.29 assume the use of a "perfect" reference with which to make the measurements. If an "imperfect" reference is used, meaningful measurements are obtained as long as the TIE variations are significantly greater than

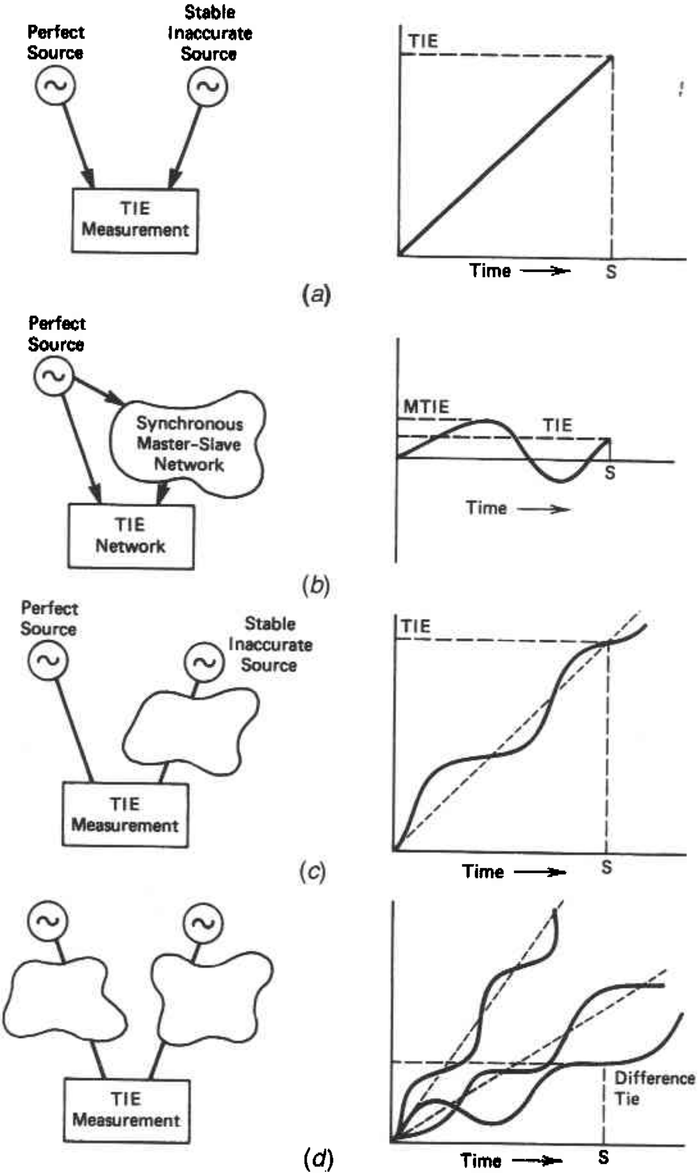


Figure 7.29 TIE and MTIE for various clock imperfections: (*a*) pure offset; (*b*) pure jitter; (*c*) offset and jitter; (*d*) TIE difference measurement.

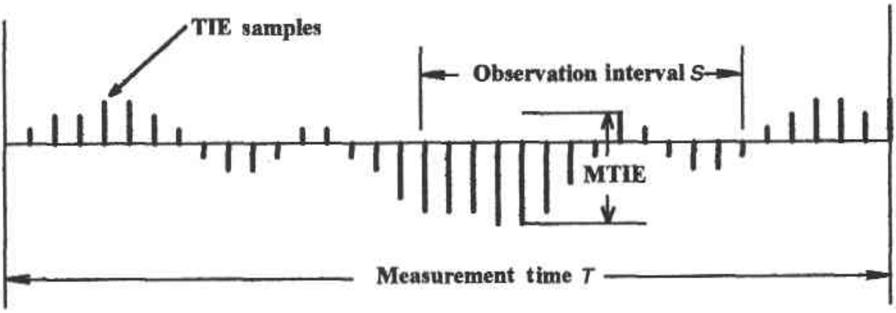


Figure 7.30 Maximum time interval error example.

the imperfections of the reference. Sometimes the relative performance between two timing signals is of interest. For example, a PBX with T1 connections to two different places (or carriers) is more dependent on the timing differences between the two transmission links than on the absolute performances. Figure 7.29*d* depicts measuring the relative performance of two such links. In this case, all measurements are meaningful, even arbitrarily low frequency wander, which can cause slips but is difficult or impossible to measure in an absolute sense. If relative measurements indicate timing problems exist, absolute measurements may be needed to isolate the problem source.

Time Variance

Neither TIE nor MTIE measurements convey any information regarding the frequency content of the jitter (other than that conveyed by the measurement interval S). A more general statistical characterization of the jitter requires representing the jitter magnitude as a function of frequency, or equivalently, as a function of time between TIE samples. The time variance (TVAR) is such a measure. TVAR values are determined as the expected variance of second-order differences between TIE samples that are separated by a time τ, where τ varies from zero (or some fraction of a second) to some maximum observation period. TVAR values are customarily measured in units of time squared (e.g., nanoseconds squared). The formula for calculating TVAR values from TIE samples $x_i (i = 1, \ldots, N)$ is

$$\begin{aligned}\mathrm{TVAR}(\tau) &= \sigma_x^2(\tau)\\ &= \tfrac{1}{6}E[(\Delta^2 x)^2]\\ &= \frac{1}{6n^2(N-3n+1)}\sum_{j=1}^{N-3n+1}\left[\sum_{k=0}^{n-1}(x_{j+2n+k} - 2x_{j+n+k} + x_{j+k})\right]^2 \qquad (7.8)\end{aligned}$$

where $\tau = n\tau_0$ (τ_0 is the sampling interval) and the observation period is $N\tau_0$.

The use of second-order differences removes the effect of a dc offset or of a linear phase ramp in the TVAR samples. Thus, there is no need to use a synchronized reference to determine the TIE samples before determining the TVAR values. However, if TVAR values for long observation periods τ are to be determined, a stable reference is needed to preclude reference wander from influencing the measurement values.

Time Devlation

Time deviation (TDEV) measurements are merely the square root of the TVAR measurements. Thus, TDEV and TVAR values have the same relationship as the classical standard deviation and variance of a probability distribution. TDEV is customarily measured in nanoseconds.

7.4 U.S. NETWORK SYNCHRONIZATION

The original plan for synchronization of the U.S. network formulated by the Bell System and the U.S. Independent Telephone Association was to use master–slave synchronization with a single master clock [26]. Due to the breakup of the network into multiple independent companies and to difficulties in reliably distributing a highly accurate master, the synchronization architecture for the United States was changed to a plesiochronous/hierarchical design in the late 1980s [13, 18, 19]. Although the hierarchical design is still in use, it is gradually changing to a "flatter" design by incorporating top-level functionality in more and more nodes.

7.4.1 Synchronization Regions

As shown in Figure 7.31, the public network is partitioned into synchronization regions that are internally synchronized with a master–slave timing hierarchy that establishes different levels of timing quality: stratum 1 to stratum 4. Stratum 1 clocks have the highest quality while stratum 4 clocks have the lowest. Timing for each region is established by a primary reference source (PRS) at stratum 1. Stratum 1 clocks are free-running clocks with inaccuracies no greater than one part in 10^{11}. Some regions may have their own PRS while others may use a synchronization signal from another party (e.g., an interexchange carrier such as AT&T).

For the most part, the synchronization regions correspond to LATAs. Every region must have at least one stratum 2 clock, which is typically associated with an access tandem switch. Toll offices within LATAs may also have stratum 2 clocks. All toll switches within AT&T network contain stratum 2 clocks [13]. Except when they use a common PRS, the synchronization regions are independently synchronized. Thus connections between the regions (using interexchange carriers like AT&T) are typically plesiochronously timed. Within a single region nodes are synchronized in a master–slave hierarchy as indicated.

The accuracy requirements of the four levels of stratum clocks are provided in Table 7.2. These accuracies pertain only to situations in which the nodes are operating in a free-running mode. Normally the nodes are synchronized to higher level clocks so the long-term accuracy is traceable back to the respective PRS. In addition to listing free-running accuracies, Table 7.2 lists accuracies that must be met during holdover

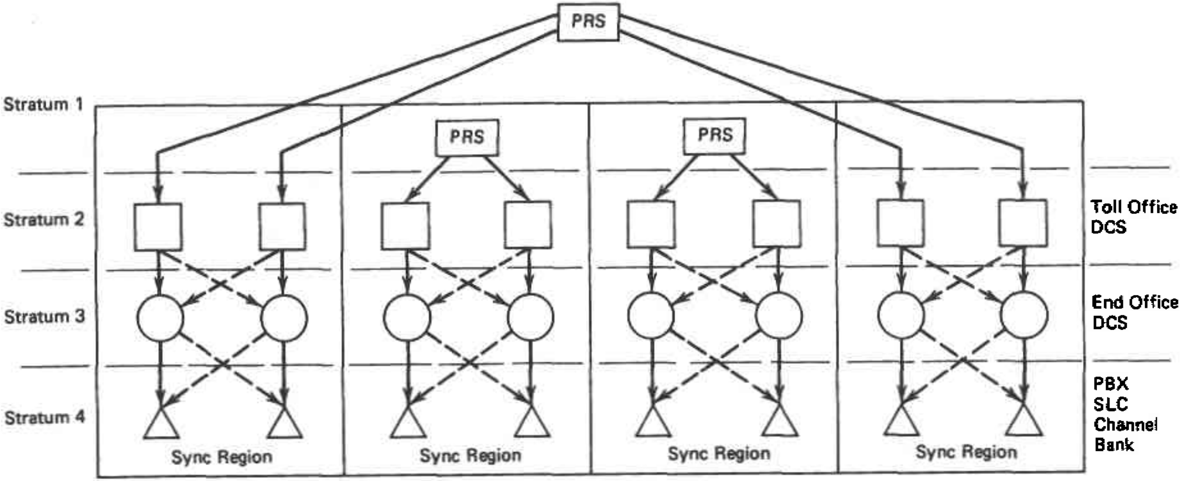


Figure 7.31 North American network synchronization.

TABLE 7.2 Stratum Clock Accuracy Requirements

Level	Free-Run Accuracy	24-Hr Holdover Accuracy
Stratum 1	$\pm 1 \times 10^{-11}$	Not applicable
Stratum 2	$\pm 1.6 \times 10^{-8}$	$\pm 1 \times 10^{-10}$ per day
Stratum 3E	$\pm 4.6 \times 10^{-6}$	$\pm 1 \times 10^{-8}$ (day 1)
Stratum 3	$\pm 4.6 \times 10^{-6}$	< 255 slips in day 1
Stratum 4	$\pm 32 \times 10^{-6}$	Not required

states. The holdover mode for stratum 2 and 3 clocks requires these nodes to gradually transition to the free-running state when they lose their reference signals.* A stratum 2 clock, for example, loses the previously established frequency at the rate of 1×10^{-10} per day, implying it would take 160 days to reach the worst-case free-running accuracy. Stratum 4 clocks, however, have no holdover requirements so they enter the free-running state immediately upon losing their reference(s).

Network synchronization requirements also specify how the various nodes respond to degradation or complete failure of their references. Stratum 2 and 3 clocks must bridge short interruptions in the reference with minimum specified time-keeping errors [19]. Stratum 3 clocks, which typically have a primary reference and a secondary reference, must perform a very gradual switchover to the secondary reference when the primary fails. Abrupt switchovers, as often occur in stratum 4 nodes (PBXs), can cause phase transients in the output clock, which can in turn disrupt synchronization in all downstream devices (e.g., within a private network).

7.4.2 Primary Reference Sources

The ANSI MTIE specification for the accuracy of a primary reference source is shown in Figure 7.32. Notice that jitter (instability that can be observed in less than 0.05 sec) is not specified. Wander, as measured over 500-sec time intervals, is limited to 3000 nsec. The asymptote for long-term timing errors corresponds to inaccuracies of 1×10^{-11}.

In addition to being designed with highly accurate (cesium beam or rubidium) clocks, all PRSs must be continuously verified with universal coordinated time (UTC). Such verification can be achieved by using a UTC-based navigation system such as Loran-C or GPS. MCI uses Loran-C and GPS to directly synchronize its PRS. AT&T uses GPS to monitor (verify) the long-term accuracy of each PRS node established in North America.† The typical accuracy of these nodes is much better than the ANSI or ITU requirement [13].

*Holdover operations are typically implemented by storing the last valid correction value (e.g., filtered phase detector output) and maintaining that value as an offset correction to a highly stable (temperature-compensated) oscillator.

†Originally, AT&T had 16 PRS nodes. Subsequent large-scale usage of GPS has led to the establishment of hundreds of PRS in the AT&T network.

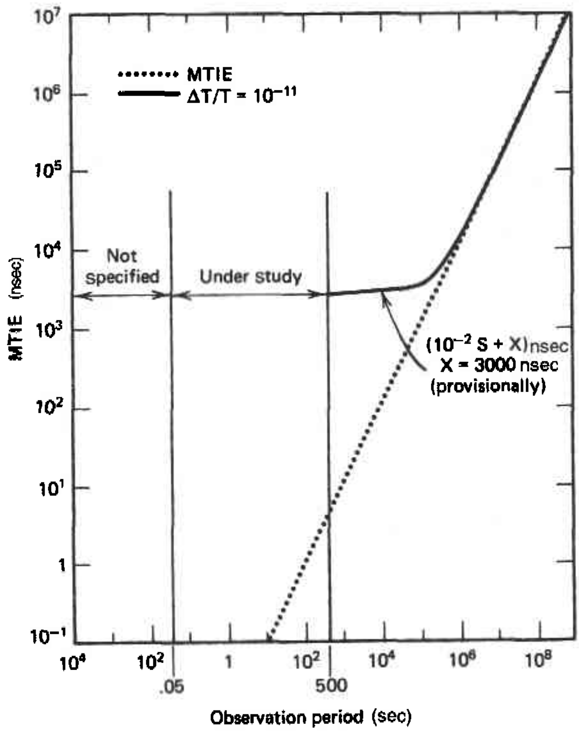


Figure 7.32 Permissible MTIE versus observation period at the output of a primary reference source.

7.4.3 1996 AT&T Synchronization Architecture

Beginning in 1996 AT&T began a move away from a two-tiered synchronization architecture to a single-level architecture with a stratum 2 clock in every office synchronized to GPS [27]. Significant features of this architecture are:

1. The synchronization distribution network has been eliminated.
2. Synchronization is independent of the network topology so the traffic-carrying network can be changed without affecting synchronization.
3. Each node is monitored by two adjacent nodes and each node monitors two adjacent nodes for performance verification.
4. Performance verification involves the use of both MTIE and TDEV.
5. DS1 timing signals are derived from SONET optical signals.

7.5 NETWORK CONTROL

The synchronization procedures described in the preceding section represent methods for controlling the timing between transmission and switching systems. In this section synchronization is discussed in a more general sense. Instead of just time or frequency control, the synchronization concept is extended to higher level functions of connection control and network control as a whole. Fundamental to the control concept is the interaction between two processes (e.g., the exchange of information from one switching machine to another to set up or monitor a connection).

A particularly useful means of defining the interaction of two processes is a state transition diagram. The main purpose of the state transition diagram is to abstract the operational states of a process and indicate what events cause transitions from one state to another. When these events are messages (signaling tones or CCS messages) from another process, the state transition diagram effectively defines how two communicating processes interact.

7.5.1 Hierarchical Synchronization Processes

As an example of a more generalized concept of synchronization, Figure 7.33 has been included to demonstrate three distinct levels of control for a conventional telephone connection using digital transmission and multiplexing. The lowest level process shown in Figure 7.33*a* depicts nothing more than the clock synchronization process required to transmit and receive digital information. There are only two states to the process in both the transmitter and the receiver. The purpose of clock synchronization is to cause transitions between the two states in the receiver to coincide with transitions in the clocking process of the transmitter. To accomplish this, a certain amount of transmission capacity is required in the form of line code transitions.

Figure 7.33*b* depicts a higher level synchronization process involving the framing of a time division multiplexer. Both processes represent a modulo-N counter, where N is the number of channels in the TDM frame. The two processes are synchronized (framed) by utilizing some of the transmission capacity to send framing patterns. Once the receiver acquires framing, the counter in the receiver counts in synchronism with the counter in the transmitter so that individual TDM channels are properly identified.

Figure 7.33*c* provides state transition diagrams of a somewhat more complicated but easily understood process. The figure depicts the connection control of a conventional telephone call. The state transition diagram of the first process represents a subscriber placing a call (going off-hook). The second state transition diagram represents the sequence of states the control element in the local switch goes through to set up the connection.

As indicated, the process begins by the originating subscriber going off-hook and waiting for the dial tone. When the switch recognizes the off-hook signal (current flow in the line), it connects the subscriber line to a digit receiver that returns a dial tone. The subscriber then dials the address of the desired telephone and enters another wait state. Upon receiving the last digit of the address, the switch control processes the request. Once the status of the called party is determined, a busy tone or a ringback tone is returned to the originating subscriber. A busy tone prompts the subscriber to hang up (go on-hook) while a ringback signal causes the subscriber to stay in the wait state until the called party answers or until the caller "times out" and abandons the call.

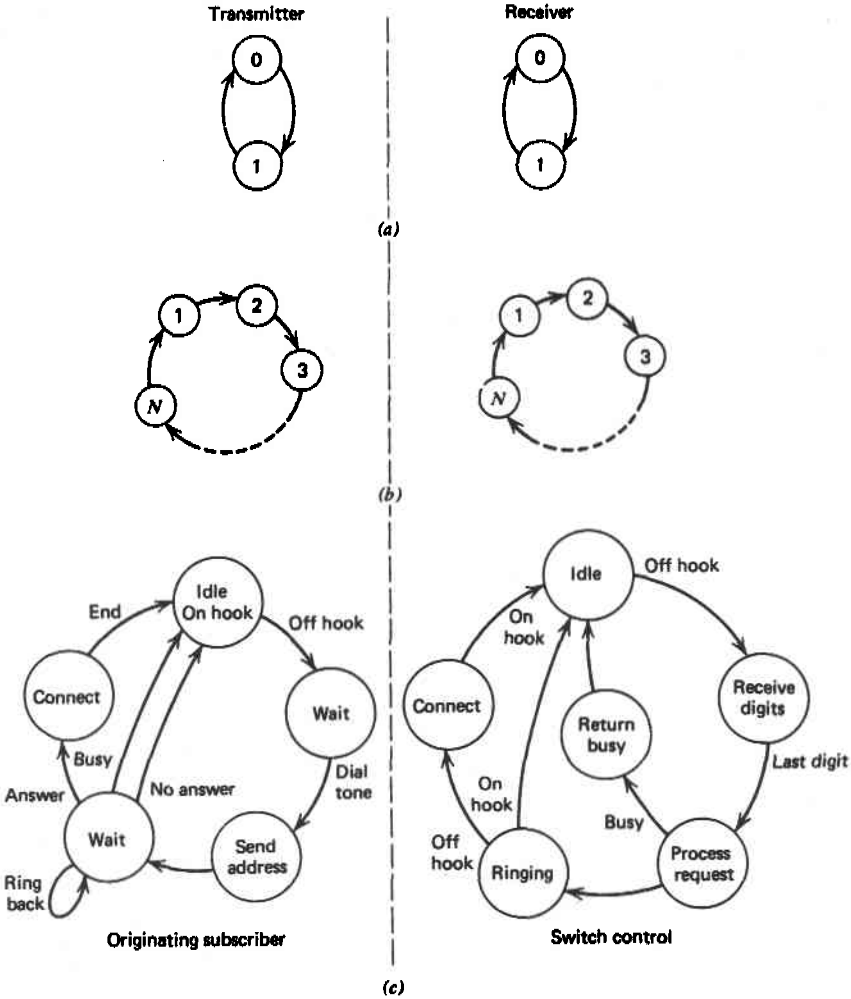


Figure 7.33 State transition diagram of synchronization processes: (*a*) timing process; (*b*) framing process; (*c*) telephone connection process.

When the called party answers, both processes enter the connected state and communication between the end users begins. The end users then get involved in yet another level of "synchronization." Voice telephone users begin by exchanging greetings and identities to establish a "connection" between their thought processes to communicate on a mutually understood subject. The message exchange process also requires synchronization so that only one person talks at a time. Hence various forms of control signals are needed to "turn the line around." Although being somewhat subtle in nature, these control signals represent transmission overhead in the same sense as control signals within the network. A talker may indicate his end of transmission by asking a question, by inflections in the voice, by the message itself, or more commonly by a pause.

Data communications equipment goes through the same basic procedures in order to establish connections and exchange information. In this case, the procedures are defined more formally and, consequently, are more restrictive. The formal rules of communication between data communications equipment are usually referred to as a "protocol." Data communications protocols typically include a definition of certain control codes, code interpretations, message framing, turn-around procedures for half-duplex lines, error control, message sequencing, fault control, and recovery.

Automated fault control and recovery procedures for communications networks can become quite involved and difficult to implement reliably. When individual voice circuits malfunction (e.g., become noisy or disconnected), the recovery procedures are left to the users. They merely redial the connection and take up where they left off. However, large trunk groups or switching systems must be designed for higher levels of dependability and maintainability. The dependability criterion ensures that failures or malfunctions rarely occur or that they are circumvented automatically by protection switching. High levels of maintainability ensure that failures are repaired quickly when they occur. Within switching systems, most of the instructions and memory words of the processor are dedicated to hardware and software performance monitoring, recovery procedures, and maintenance diagnostics.

7.6 NETWORK MANAGEMENT

In addition to controlling individual connections and equipment, a communications network must also manage its facilities on more macroscopic levels. The basic goal of network management is to maintain efficient operations during equipment failures and traffic overloads. The main considerations are routing control and flow control.

7.6.1 Routing Control

Routing control refers to procedures that determine which paths in a network are assigned to particular connections. If possible, connections should use the most direct routes at the lowest levels of the network. The direct routes are obviously desirable because they use fewer network facilities and generally provide better transmission quality. However, economic considerations often limit the capacities of the direct routes so that alternate routes are needed to maintain suitably low blocking probabilities between one switching machine and another.

If a trunk group between two switching machines contains enough circuits to provide an acceptably low blocking probability, a significant number of the circuits in the group are idle during average traffic loads. A more economical design allocates a limited number of heavily utilized trunks in the direct route and provides alternate routes for overflow (alternately routed) traffic. In this manner the users are able to share larger portions of the network. Chapter 12 presents basic examples of how a network can be engineered to minimize the transmission facilities while providing a given grade of service (blocking probability). As discussed in Chapter 1, the use of centralized control for the network, with common-channel signaling, provides significant efficiencies of operation in congested networks.

7.6.2 Flow Control

In the preceding section, alternate routing is discussed as one aspect of managing traffic in a communications network. Routing algorithms are concerned only with the utilization of paths or directions of travel within a network. Another requirement of network management is to control the amount of traffic in a network. Managing the rate at which traffic enters a network is referred to as flow control. A network without effective flow control procedures becomes very inefficient or ceases to function entirely when presented with excessively heavy traffic conditions.

The generalized performance of a large, uncontrolled network is shown in Figure 7.34 as a function of the offered traffic. As indicated, when light traffic conditions exist, the network carries all traffic requests presented. As the load increases, however, some of the offered traffic is rejected because no appropriate circuits are available for particular connections; that is, blocking exists. As the input load increases even further, a network with no flow control eventually begins to carry less traffic than it does when lighter loads are presented. If the offered load increases even more, the network may even cease to carry any traffic at all.

The reason that the volume of carried traffic decreases when the offered traffic exceeds some critical value is that partially completed requests tie up network resources while trying to acquire other resources tied up by other partially completed requests. Thus a form of dynamic deadlock occurs. Prior to the development of centralized network control, this situation would often arise on busy calling days (e.g., Mother's Day). In a network with distributed control all sources of traffic are serviced by successively seizing trunks to intermediate switching nodes until the destination is reached. If heavy traffic exists, requests emanating from all sides of the network en-

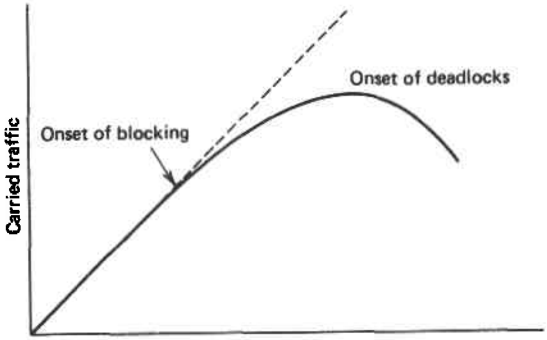


Figure 7.34 Traffic carried versus traffic offered for a network with no flow control.

counter congestion somewhere near the middle. At that time, the partially completed connections are tying up facilities needed by other requests. In the limit, when extremely heavy traffic exists, all network resources are held by partially completed requests and no complete connections can be established!

Another example of the need for flow control is automobile traffic at a metropolitan intersection. Have you ever encountered an intersection in which your lane of traffic was blocked by cross traffic backed up into the intersection? In essence, the driver blocking your direction of travel seized a common resource (the middle of the intersection) without being able to obtain the next resource required (the other side of the intersection). Bumper-to-bumper traffic in one direction can significantly degrade the throughput in other directions. With heavy traffic in all directions, total throughput can grind to a halt until the congestion is relieved from the periphery inward.

The fundamental principle demonstrated by these examples is that efficient use of the common resources of a heavily loaded network requires some form of flow control. In the automobile example, smooth operation of an intersection depends on each driver looking ahead and not entering an intersection unless he can get all the way across. A telecommunications network must use the same basic principle (it is hoped with more discipline). The control elements at the periphery of the network must be aware of the internal status of the network and control the flow of traffic from its sources.

More than one level of flow control is sometimes implemented within a network. In a data communications link, some form of flow control is required to keep a source terminal from overloading the terminal at the other end of the link. The receiving terminal uses a reverse channel (sometimes with a lower bit rate) to inform the source when to cease and when to begin transmissions. This level of flow control in a circuit-switched network involves the terminals themselves and is of no concern to the network since the traffic flows within an established connection. Of more concern to a circuit-switched network is how to control the flow of connection requests into the interior of the network. In setting up a long-distance connection, the first few circuits required should not be seized unless there is a reasonable chance that all of the circuits necessary to complete the connection can be obtained. Partially completed circuits only degrade the network capacity by increasing congestion without satisfying a service request. Network flow control is greatly simplified with common-channel signaling support for centralized network control. The following paragraphs describe basic, uncentralized flow control techniques and how centralized control simplifies their implementation.

Trunk Directionalization

The operation of trunk circuits can be classified according to two different ways of controlling seizures for particular calls. Two-way trunks can be seized at either end. One-way trunks, on the other hand, can be seized only at one end. (Notice that this has nothing to do with the direction of message transfer on established connections, which is always in both directions.) When one-way trunking is used, the trunk group is usually partitioned into one group that can be seized at one end and one group that can be seized at the other end. Two-way trunk groups are obviously more flexible in servicing fluctuating traffic patterns, but they are more difficult to control since the possibility of simultaneous seizures (called glare) at both ends must be resolved.

A useful feature to incorporate into two-way trunks is the ability to directionalize them by marking them busy at one end and effectively creating one-way trunks. With this mechanism, a distant, overloaded switching node can be relieved of additional incoming traffic while providing it sole access to the trunk group for outgoing traffic. Thus the overloaded node relieves its congestion while inhibiting new arrivals.

When the network as a whole experiences heavy traffic loads, trunk directionalization can be used to reduce the flow of connect requests into the interior of the network while establishing one-way trunks from the interior to the periphery. Thus connect requests that manage to get to the interior have a much better chance of obtaining all facilities required for the complete connection.

Cancellation of Alternate Routing

Alternate routing of traffic accommodates localized overloads by transferring traffic to underutilized routes. During networkwide overloads, however, alternate routing is undesirable for two reasons. First, alternate routes imply that a greater number of transmission and switching facilities are needed for a connection. If these same facilities could be assigned to two or more direct connections, the total number of links per call could be reduced and the network could carry more traffic.

Second, the probability that an alternately routed call can acquire all the necessary resources is relatively low. Trying to set up a connection with a large number of facilities is undesirable if the probability of getting all of the facilities is low (particularly so, if some facilities are fruitlessly tied up while less demanding requests are pending).

Code Blocking

Code blocking refers to artificially blocking calls intended for specific destination codes. If the calls are blocked at originating end offices before they acquire internal network facilities, the destinations are relieved of incoming traffic without tying up facilities that may be needed for outgoing requests from the specified areas.

The method of flow control is particularly useful in times of natural disasters, which typically stimulate large numbers of calls both into and out of the area of the disaster. In these events a network control center can initiate code blocking for all, or a large percentage, of the calls into the area. The principle of giving preference to the outgoing calls serves two purposes. First, no network facilities are seized unless there is a reasonable chance of obtaining all facilities necessary. It is the trunks into or out of the disaster area that are the focal point of network congestion. Once one of these trunks is seized, the rest of the connection can probably be established. Second, code blocking is useful because outgoing calls are probably more important than incoming calls.

Centralized Connection Control

All of the flow control procedures described previously are designed to eliminate seizures of common resources if the desired connection has a low completion probability.

Because of the distributed nature of network control implied by these operations, these control procedures are necessarily probabilistic. To maintain a certain amount of network efficiency, the network is purposely operated at less than maximum capacity.

A more desirable mode of operation, from a throughput point of view, is to allocate network facilities from a single centralized control node. Since this central node has access to the status of all network resources, no facilities are assigned to a particular request unless all facilities needed for the complete connection are available. Network transmission links are assigned in a manner that is analogous to the assignment of internal links of common control switches.

Complete centralized control of a network as large as the public telephone network is obviously infeasible from the point of view of maintaining status of all interstage links within end office switches and from the point of view of survivability of the network when the control node fails. However, many aspects of centralized control have been implemented in North America and around the world with common-channel signaling (CCS). For example, INWATS call requests are routed to a central node that determines if the destination is busy or not. If the destination is busy, the originating end office is instructed to return the busy tone without any of the internal transmission links ever being seized. This mode of operation is particularly useful for 800 numbers (INWATS) that occasionally experience very heavy traffic flow because of national television announcements.

Without CCS, the previous mode of operation was to return a busy tone all the way through the network from the place at which the busy circuit or subscriber is located. Thus the path through the network was tied up during the time the busy tone was being returned. CCS allows the originating office to return the busy tone so internal network facilities can be released and reassigned immediately upon detecting the busy condition.

REFERENCES

1 M. Decina and U. deJulio, "International Activities on Network Synchronization for Digital Communication," *IEEE International Communications Conference*, 1979.

2 J. R. Pierce, "Synchronizing Digital Networks," *Bell System Technical Journal*, Mar. 1969, pp. 615–636.

3 C. J. Byrne, B. J. Karafin, and D. B. Robinson, "Systematic Jitter in a Chain of Digital Repeaters," *Bell System Technical Journal*, Nov. 1963, pp. 2679–2714.

4 "Digitally Dejittering Laser Disc Players," *IEEE Spectrum*, Feb. 1990, p. 14.

5 F. M. Gardner, *Phaselock Techniques*, 2nd ed., Wiley, New York, 1979.

6 E. D. Sunde, "Self-Timing Regenerative Repeaters," *Bell System Technical Journal*, July 1957, pp. 891–938.

7 D. L. Duttweiler, "The Jitter Performance of Phase-Locked Loops Extracting Timing from Baseband Data Waveforms," *Bell System Technical Journal*, Jan. 1976, pp. 37–58.

8 "Carrier-to-Customer Installation-DSI Metallic Interface," ANSI T1. 403–1989, American National Standards Institute, New York, 1989.

UNIT IV

DIGITAL SUBSCRIBER ACCESS

INTEGRATED SERVICES DIGITAL NETWORK

In addition to the digitization of the internal portions of public telephone networks, a lesser known but also significant change involved the development of common-channel signaling (CCS) for network control. Both the digitization and the use of CCS started at internal portions of the network and migrated toward the periphery. Except for some special data service offerings and a few network-based features derived from the signaling network, these facilities provided no direct benefit to the end users. As shown in Figure, ISDN is a service offering that extends access to both of these facilities to the end user. Access to the digital transport facilities occurs on 64-kbps bearer (B) channels while access to the signaling network occurs on 16- or 64-kbps signaling (D) channels. Major features or benefits made available by these channels are listed in Tables.

Two levels of digital access to the ISDN network have been standardized: basic rate access and primary rate access. As shown in Figure, the (worldwide) basic rate interface (BRI) standard is also referred to as a 2B + D interface. In North America, the primary rate interface (PRI) standard is sometimes referred to as 23B + D while the ITU-T counterpart is 30B + D. The North American PRI is fundamentally a 1.544Mbps DS1 signal with the D channel replacing one of the 24 message channels (usually the last one). To achieve a 64-kbps clear channel capability, a B8ZS line code is used to eliminate one’s density requirements and common-channel signaling frees up the signaling bits so the full 64-kbps bandwidth is available for user data. The ITU-T

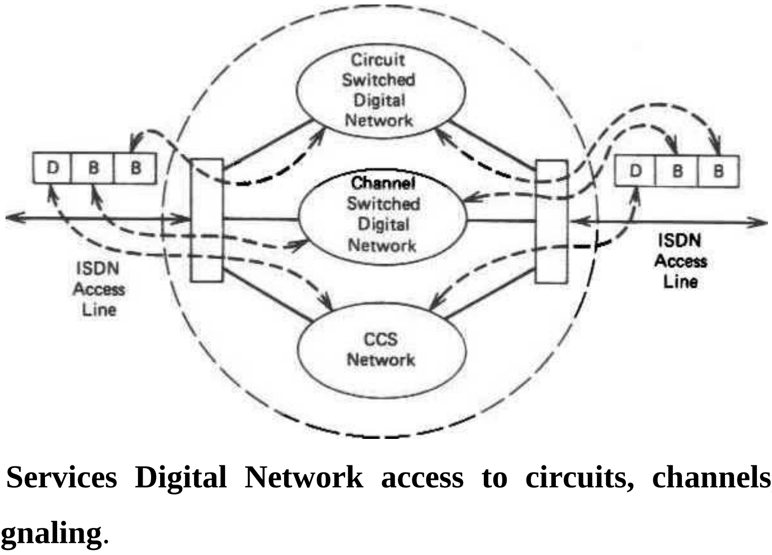

Circuit Switched Digital Network
D
B
B
Channel Switched Digital Network
D
B
B
ISDN Access Line
ISDN Access Line
CCS Network

Figure Integrated Services Digital Network access to circuits, channels, leased lines, and common-channel signaling.

Features of ISDN B Channels

- End-to-end four-wire digital circuits: no loss or echoes for voice circuits using digital instruments
- Shared network access for voice, data, and leased lines
- Relatively high bandwidth data channels (64 kbps)
- Lower error rates than typical voiceband modems
- In-service performance monitoring
- Possible expansion of speech bandwidth because elimination of tandem encodings allows greater one-time quantization errors
- PRI is a 2.048-Mbps El digital signal with the D channel occupying the signaling channel (time slot 16). Because a single D channel can support more than one PRI, 24B and 3IB interfaces are allowed for additional PRIs in a group of PRIs.

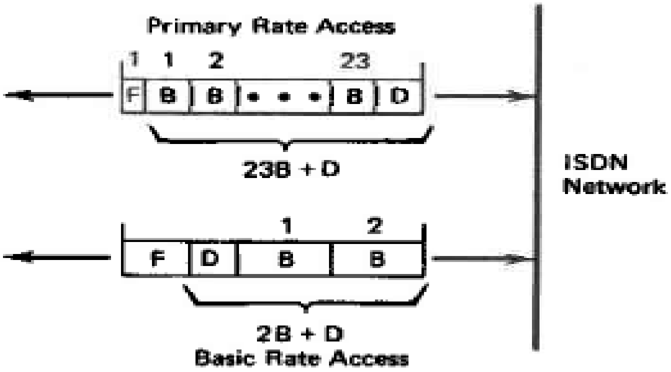
ISDN Basic Rate Access Architecture

An ISDN basic rate access line is a standard copper pair that has been specially conditioned to support a bidirectional 160-kbps aggregate data rate. Transmission technology required for basic rate access is generally referred to as the digital subscriber loop (DSL). Complications arise when using existing analog pairs. The principal considerations are bridged taps and wire gauge changes, both of which cause reflections that impact higher speed digital signals. To allow flexibility in the selection and deployment of the DSL, the ITU-T basic rate specification does not define a two-wire transmission standard. Instead, it establishes an interface standard that assumes the presence of a network termination module that converts any chosen transmission system to the standard interface. In the interest of supporting deregulated customer premises equipment, the Exchange Carriers Standard Association in the United States established a basic rate transmission standard so CPE equipment could connect directly to the transmission link or select network termination modules from alternate vendors. Figure depicts the architecture and associated terminology of a North American BRI.

Features of ISDN D Channels

- Signaling simultaneous with active connections
- Calling number identification
- Far-end supervision
- User-to-user message transfer
- Telemetry for fire alarms, security, meter reading, etc,

- Access to packet-switching network
  - Support for network services such as multiple directory numbers sharing one or more **B** channels, trunk group blocking statistics, and identification of calling number for abandoned or blocked calls



Basic rate and primary rate access to ISDN.

Module Definitions

NT1: A network termination module for layer 1 functions that provides physical and electrical termination of the transmission link only. In essence, the NT1 isolates the user from the transmission technology but does not demultiplex or process D channel messages.

NT2: A second level of network termination that implements functions associated with layers 2 and 3 of the OSI protocol stack. Thus, NT2 equipment extracts and processes D channel messages. Representative NT2 equipment includes PBXs, multiplexers, or LAN gateways.

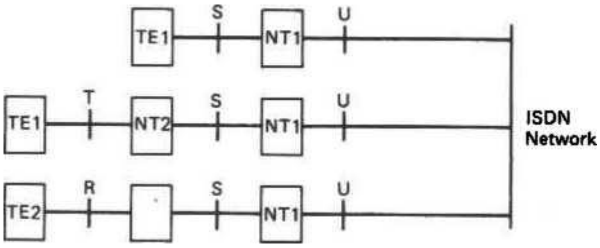
TE1: Type 1 terminal equipment such as a digital telephone that complies with the ISDN S interface recommendation.

TA: Terminal adapter used to convert from an arbitrary (R) interface to the ISDN S interface.

TE2: A non-ISDN terminal that requires a terminal adapter to interface to the ISDN S interface. Prevalent examples of a TE2 equipment are analog telephones or asynchronous (RS-232) data terminals.

Reference Points

U: Interface to the two-wire transmission line.



T: CCITT ISDN interface defined in Recommendation 1.430.

S: Interface to NT2 equipment identical to a T interface.

R: A non-ISDN interface such as an analog tip and ring.

S/T Interface

The S/T interface is defined in ITU-T recommendation 1.430 to be supplied by network termination equipment (NT2/NT1). It is intended for customer premises installations only. (No overvoltage protection is prescribed.) The most significant aspects of the S/T interface are:

- Four-wire facility (one pair for each direction)
- One kilometer maximum required distance
- Alternate space inversion line code (which is the inverse of an AMI line code: see Figure 11.4)
- Point-to-point or point-to-multipoint configurations
- Data rate of 192 kbps with 48 kbps of framing, control, and synchronization

The frame structure at reference points S and T is shown in Figure 11.5. As indicated, each 250-msec frame contains 48 bits. Thirty-eight of these bits are common to both directions of transmission and are defined as follows:

16 bits in first B channel (Bl)

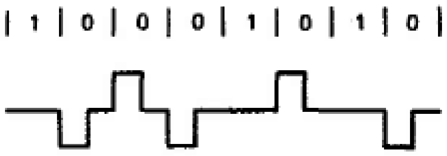
16 bits in second B channel (B2)

4 bits in the D channel 1 bit in the framing channel F

1 bit in the auxiliary framing channel F_A

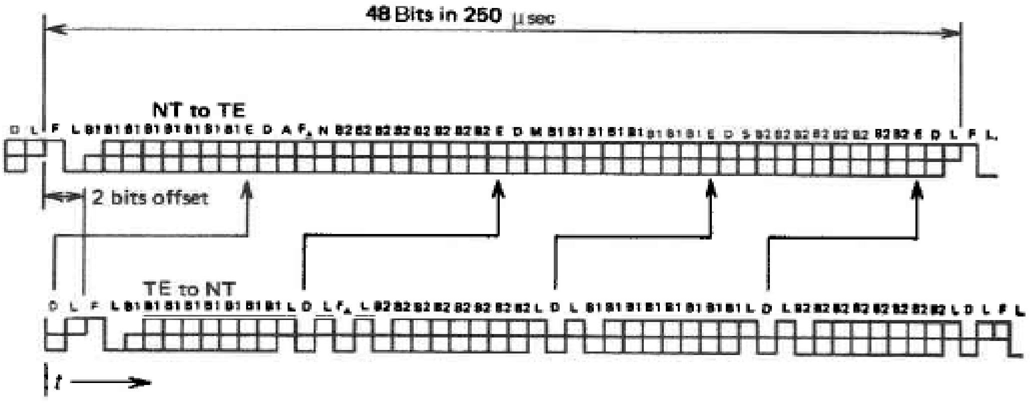
The remaining 10 bits are assigned different functions depending on the direction of transmission. From the TE to the NT all remaining 10 bits are defined as L bits, which are used to maintain dc balance. The definition of the 10 bits from the NT to the TE

2 L bits for maintaining dc balance



Alternate space inversion code

S/T frame structure.


48 Bits in 250 μsec
NT to TE
D L F L B1 B1 B1 B1 B1 B1 B1 B1 E D A F_A N B2 B2 B2 B2 B2 B2 B2 B2 E D M B1 B1 B1 B1 B1 B1 B1 B1 E D S B2 B2 B2 B2 B2 B2 B2 B2 E D L F L
2 bits offset
TE to NT
D L F L B1 B1 B1 B1 B1 B1 B1 B1 L D L F_A L B2 B2 B2 B2 B2 B2 B2 B2 L D L B1 B1 B1 B1 B1 B1 B1 B1 L D L B2 B2 B2 B2 B2 B2 B2 B2 L D L F L
t

4 E bits that echo D bits received from the TE(s)

1 A bit for activation

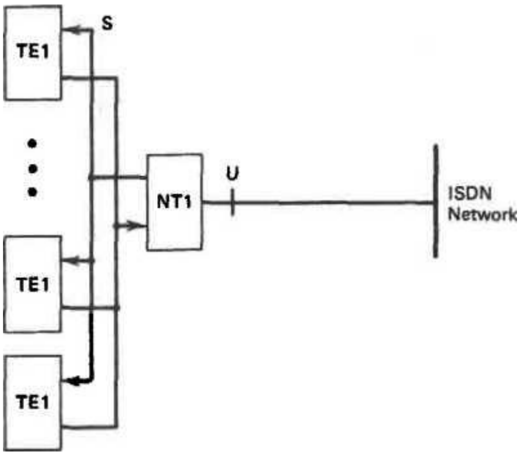
1 N bit, which is the complement of the F_A bit 1 M bit for multiframe identification

1 S bit for S channel

Figure indicates that the framing bit F is always a binary 0. Even though a positive voltage level is indicated, either a positive or negative voltage is allowed so the receivers are not sensitive to wiring polarity. A transmitter always produces the same level, however, so the receiver always receives the same polarity in every framing bit. As an aid in rapid acquisition of the framing pattern, the framing bit always represents a line code violation (it is the same polarity as the previous 0). To maintain dc balance, an L bit with the opposite polarity of the F bit always follows the F bit. The first 0 in a data block following the L bit is encoded with the same polarity as the L bit, which implies another line code violation. Direct-current balancing of this violation is the purpose of the L bit at the end of each data block, which also assures that the next (fixed-polarity) F bit produces a line code violation.

The reason for the additional L bits in the frame from the TE to the NT arises because more than one TE can be connected to the S interface as a passive bus. Because the TEs transmit independently of each other, each individual transmission (D channel bits and B channel bytes) is individually dc balanced.

Passive bus operations are also the reason for the existence of the NT-to-TE E bits. Multiple station access to the D channel is controlled by having a terminal wait for an idle code on the NT-to-TE D channel before transmitting on the TE-to-NT D channel. When a terminal begins D channel transmission, it monitors the incoming E bits. If an incoming E bit does not match the previously transmitted D bit, that terminal stops transmitting and waits to seize the channel at a later time. Two levels of priority are defined for accessing the D channel. Signaling information is the highest priority while user packet messages are the lower priority. All terminals on a passive bus have equal priorities within each level.



S-bus connections

ISDN U Interface

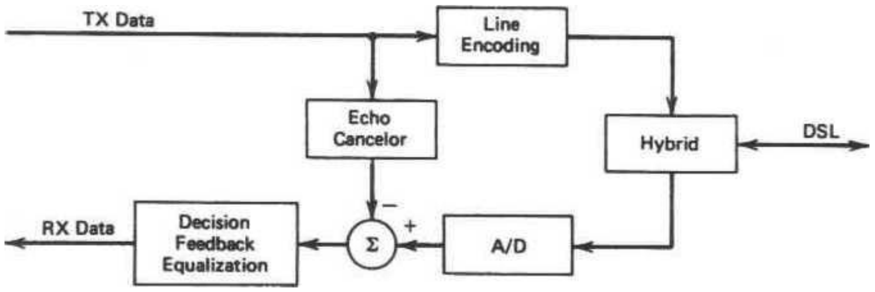
Prior to the establishment of a standard U interface by the Exchange Carriers Standards Association in North America a number of basic rate transmission systems were developed by telephone equipment suppliers around the world and put into service.

One of these involves the use of time compression multiplex (TCM). TCM, developed by NEC in Japan, provides full-duplex transmission on a single pair of wires by alternately transmitting bursts of data in each direction. For this reason it is sometimes referred to as “ping-pong” transmission. One big advantage of TCM transmission is that near-end crosstalk is avoided because a station is never receiving while transmitting. The big disadvantage is that the burst data rate must be more than twice the desired data throughput.

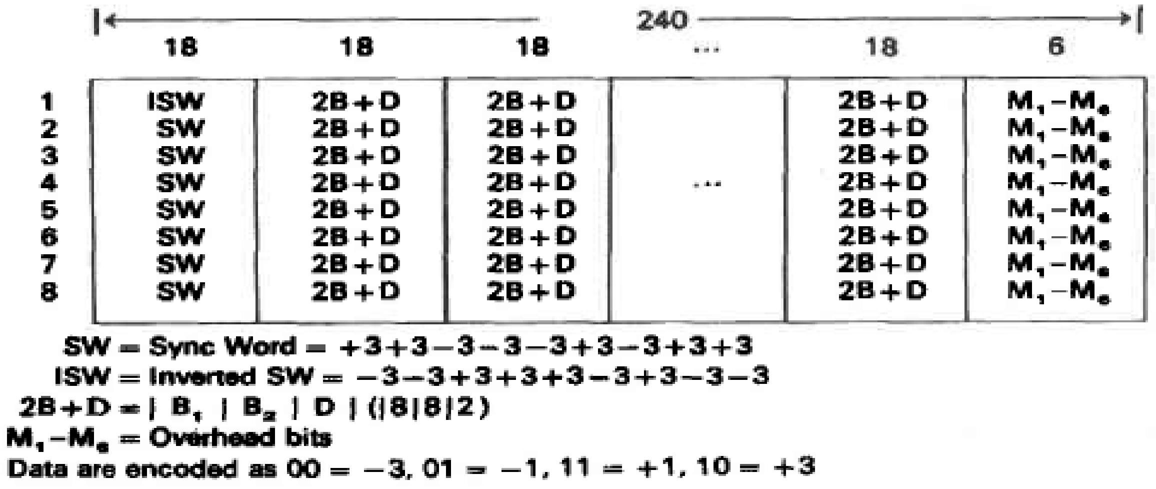
AT&T in the United States also developed a BRI transmission system for the No. 5 ESS end office switching system available with generic releases 5E4 and 5E5, this system has a 160-kbps data rate utilizing a 50% duty cycle AMI line code. Full-duplex transmission is achieved by simultaneous transmission in both directions using hybrids and echo cancelers (ECs) to separate the two signals. Beginning with generic 5E6 the No. 5 ESS supports both the AT&T (Lucent) U interface (referred to as a 5E4/5E5 U interface) and the ANSI U interface.

Like the 5E4/5E5 U interface, the ANSI U interface uses simultaneous transmission in both directions with echo cancelers and a data rate of 160 kbps. The major difference is the use of a four-level line code referred to as 2B1Q (two binary digits in one quaternary digit). Thus, the symbol (baud) rate on the line is 80,000 symbols/sec. Because the line code itself does not prevent dc wander, dc restoration is necessary. A 2B1Q line code was chosen primarily because the lower symbol rate minimizes the two dominant transmission limitations in this application: intersymbol interference and near-end crosstalk [6].

The frame format and superframe structure of the ANSI U interface are shown in Figure 11.8. Each frame consists of 240 bits containing 18 framing bits, 216 payload bits (12 fields of 18 2B + D data bits), and 6 overhead bits. Because the frame rate is 667 frames/sec the data rate is 160 kbps. The 6 overhead bits are organized as a block of 48 bits in an eight-frame superframe. Functions included in the overhead bits are 24 bits of embedded operations channel, 1 activation bit, 1 deactivation bit, 1 far-end block error bit, 12 CRC bits, and 9 fixed 1 bits. All bits except the framing bits are scrambled for transmission.



ISDN DSL TX/RX block diagram



	240					
	18	18	18	...	18	6
1	ISW	2B+D	2B+D		2B+D	M_1–M_6
2	SW	2B+D	2B+D		2B+D	M_1–M_6
3	SW	2B+D	2B+D		2B+D	M_1–M_6
4	SW	2B+D	2B+D	...	2B+D	M_1–M_6
5	SW	2B+D	2B+D		2B+D	M_1–M_6
6	SW	2B+D	2B+D		2B+D	M_1–M_6
7	SW	2B+D	2B+D		2B+D	M_1–M_6
8	SW	2B+D	2B+D		2B+D	M_1–M_6

SW = Sync Word = +3+3−3−3−3+3−3+3+3

ISW = Inverted SW = −3−3+3+3+3−3+3−3−3

2B+D = | B_1 | B_2 | D | (|8|8|2)

M_1–M_6 = Overhead bits

Data are encoded as 00 = −3, 01 = −1, 11 = +1, 10 = +3

ANSI U interface frame and superframe **structure.**

ISDN D Channel Protocol

The D channel protocol is defined in two separate series of ITU-T recommendations: the I series and the Q series. The data link layer (LAPD) is defined in 1.441 or Q.921. This protocol is similar to LAPB of the X.25 standard except Q.921 allows more than one logical link. (Therefore separate “connections” can exist for signaling, packet network, or far-end terminals.) The main functions of the data link layer are message sequencing, error checking and retransmission, and data layer link recovery. The network layer of the D channel protocol is defined in 1.451 or Q.931. This layer provides connection setup, alerting, routing, and release of ISDN calls. When a B channel accesses a packet network, the X.25 protocol is used while in the connected state.

HIGH-DATA-RATE DIGITAL SUBSCRIBER LOOPS

A basic rate ISDN digital subscriber loop provides an aggregate, bidirectional data rate of 160 kbps on a single pair of wires. This section describes several transmission techniques that allow much larger bandwidths on copper wire pairs. These new techniques are enabled by the availability of low-cost, high-performance digital signal processing. Several versions of high-speed digital subscriber lines have been developed. The various versions are collectively referred to as xDSL

Asymmetric Digital Subscriber Line

ADSL allows for high data rates to the subscriber and moderate to low data rates from the subscriber to the network. ADSL technology was originally conceived as a means of delivering switched digital video services over a copper loop, which obviously do not require high data rates from the subscriber. Although video applications did not

Versions of Digital Subscriber Lines

DSL	Digital Subscriber Line (ISDN basic rate)
ADSL	Asymmetric DSL (9 Mbps downstream, 640 kbps upstream)[8]
HDSL	High-bit-rate DSL (T1/E1 service on two pairs)
SDSL	Single-line DSL (T1/E1 service on one pair)
VDSL	Very high bit rate DSL (52 Mbps downstream, 2.3 Mbps upstream)[3]

materialize, the asymmetric data rate is also suited for Internet access. The philosophy behind the asymmetric data rates is that subscribers typically need to receive high- bandwidth data (for Internet file downloads) but normally need to transmit (query) at a relatively low data rate. The subscriber's receive data rate on ADSL varies between 1.5 and 9 Mbps while the subscriber transmit rate varies between 16 and 640 kbps/ The specific data rate utilized depends on the transmission quality of the particular wire pair.

In contrast to ISDN channels, which represent extensions of the digital facilities of the public switched telephone network (PSTN), ADSL channels are separated from the public network at the line interface of the PSTN. As shown in Figure 11.9, ADSL lines terminate at a telephone company central office (or remote terminal) where the data streams are forwarded to and received from a facility that is separate from the telephone network. Transmission between the line interface and an Internet service provider, for example, is typically provided with an ATM connection.

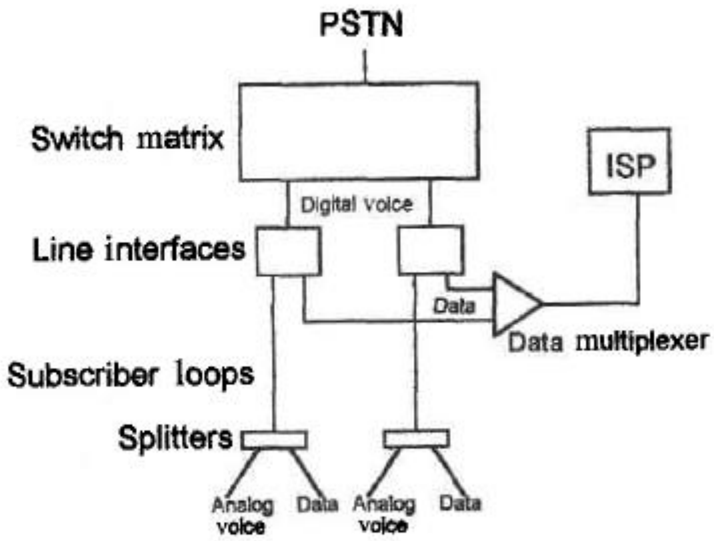
ADSL has two major advantages over ISDN access. First, ADSL data rates provided to the subscriber are significantly higher than the 128-kbps ISDN basic rate. Second, ADSL piggy backs digital transmission on a standard analog telephone wire pair. Thus, existing analog telephones are retained on ADSL but are either replaced by digital phones or are connected through conversion devices when ISDN is utilized.

Two versions of ADSL transmission links have been developed: carrierless amplitude and phase (CAP) modulation and discrete multitone (DMT) modulation. CAP is the first version deployed but DMT has been selected as the standard [8], Because DMT makes more intensive use of DSP, a DMT implementation typically requires more power—a significant consideration for remote terminal deployment. CAP, on the other hand, its not generally considered to be as flexible as DMT in achieving maximum data rates on some wire pairs or in some interference environments. For a thorough comparison of the two alternatives

DMT Implementation

Basic parameters of the standard ADSL DMT implementation are provided

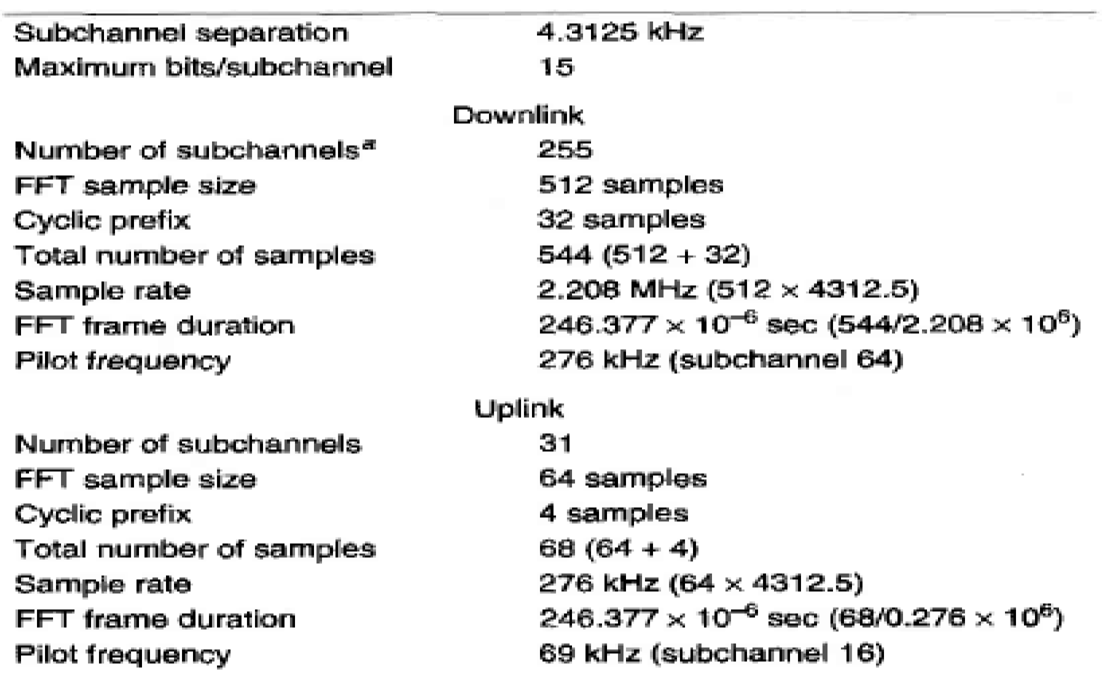
A less ambitious version of ADSL referred to as G.Lite only attempts to achieve 1.544 Mbps downstream and 384 kbps upstream.



/OICEBAND MODEMS 9

ADSL network configuration.

ADSL DMT Implementation Parameter



Subchannel separation	4.3125 kHz
Maximum bits/subchannel	15
Downlink	
Number of subchannels[a]	255
FFT sample size	512 samples
Cyclic prefix	32 samples
Total number of samples	544 (512 + 32)
Sample rate	2.208 MHz (512 × 4312.5)
FFT frame duration	246.377×10^{-6} sec ($544/2.208 \times 10^{6}$)
Pilot frequency	276 kHz (subchannel 64)
Uplink	
Number of subchannels	31
FFT sample size	64 samples
Cyclic prefix	4 samples
Total number of samples	68 (64 + 4)
Sample rate	276 kHz (64 × 4312.5)
FFT frame duration	246.377×10^{-6} sec ($68/0.276 \times 10^{6}$)
Pilot frequency	69 kHz (subchannel 16)

a modulator and an FFT as the corresponding demodulator. The following paragraphs identify the basic function of each block.

PRS Source: A pseudorandom sequence generator provides a prescribed sequence for characterizing the channel during a training period. Characteristics of the channel that are determined during the training period are attenuation and phase distortion across the band, noise/interference levels across the band, and the information capacity of individual subchannels.

Channel Allocation: Provides assignment of data bits to individual subchannels according to the subchannel capacity determinations obtained during training.

FEC: A combination of Reed-Solomon and convolutional coding.

IFFT QAM Modulation: Conversion of data values to quadrature channel signal amplitudes and conversion to a time-domain waveform using an inverse FFT.

D/A: Digital-to-analog conversion.

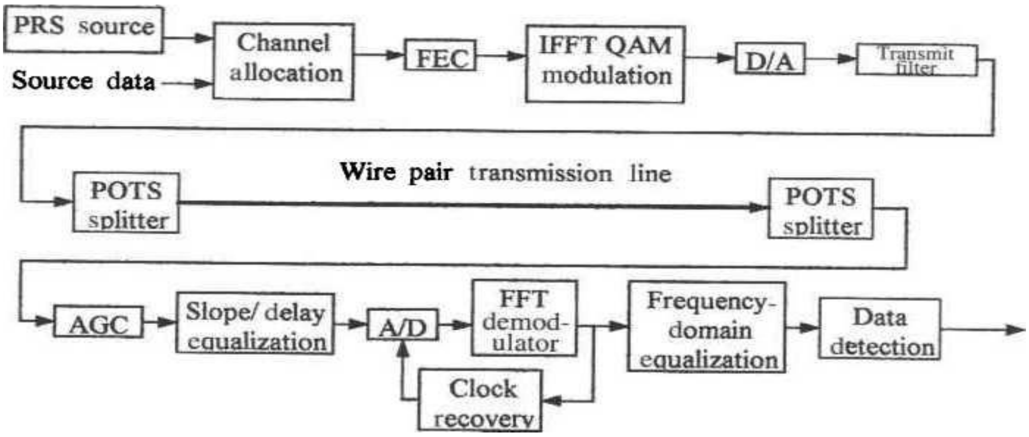
TX Filter: Bandpass filter to prevent interference into the voiceband and to smooth the discrete DSP samples.

POTS Splitter*:* Used at both ends of the line to separate the analog voice from the data. (Simpler versions of ADSL incorporate the splitter function in the modems to simplify installation.)

AGC*:* Automatic gain control to adjust overall receive level.

Slope/Delay Equalization*: A* front-end equalizer to partially flatten the frequency response and equalize extreme delay variations in the channel,

A/D: Analog-to-digital conversion.



ADSL DMT block diagram

FFT QAM Demodulation: FFT conversion of time-sampled waveform to frequency domain where data values are related to amplitudes of quadrature carriers.

Clock Recovery: A/D sample timing obtained by locking to pilot frequency. The desired sampling rate is eight times the pilot frequency so l-of-8 phase ambiguity has to be determined by monitoring framing/synchronization bit integrity.

Frequency-Domain Equalizer: Multiplication of complex (quadrature) frequency spectrum by amplitude and phase equalization parameters obtained during training period.

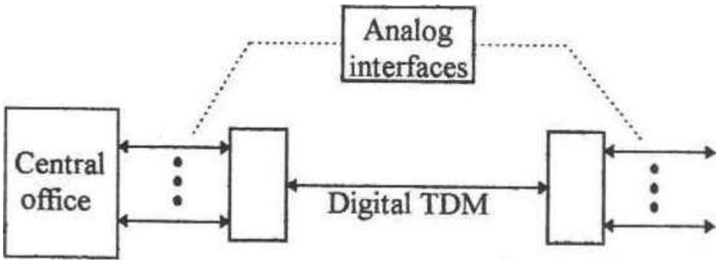
Data Detection and Interleaving: Slicing of quadrature amplitude values to decode data and subsequent generation of composite stream identical to original source data.

VDSL

VDSL is an expanded version of ADSL to achieve even higher bandwidths on particularly short lines as would be available from remote terminals of fiber to the curb systems. A primary motivation for VDSL is potential distribution of HDTV signals. Although several different modulation techniques have been proposed for VDSL, a DMT version seems to be favored.

DIGITAL LOOP CARRIER SYSTEMS

The primary purpose of a digital loop carrier (DLC) system is to reduce or eliminate copper pairs from a central office to the vicinity of a group of subscribers. Even though the transmission link from the central office to the DLC remote terminal is digital, the transmission links from the remote terminal to the subscribers are typically conventional analog loops. Thus, the main purpose of a DLC is not to provide digital subscriber access. Nevertheless, some DLC systems (particularly fiber-based DLC systems) provide options for Tl, ISDN, or xDSL digital interfaces. Moreover, the fact that the DLC remote terminal is relatively close to the subscriber locations facilitates the use of these interfaces. (A short copper drop from the remote terminal has low attenuation and crosstalk and is less likely to have loading coils, bridged taps, or multiple sections of wire with varying gauges.) Although a DLC is a natural mechanism to provide enhanced services through ISDN or xDSL digital interfaces, mechanical packaging and power considerations of these interfaces represent unique requirements with respect to strictly POTS applications.



Universal digital loop carrier system.

Universal Digital Loop Carrier Systems

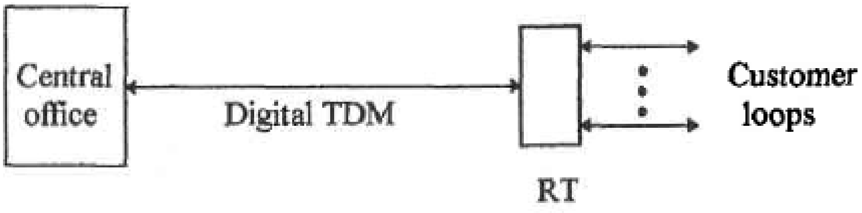
A universal digital loop carrier (UDLC) system can be interfaced to any switching system: analog or digital. The interface between the local switching system (end office) of the public network and the central office terminal (COT) involves individual circuits (e.g., individual analog tip and ring connections). The multiplexed digital transmission links between the COT and the remote terminal (RT) can be wire pairs or fiber. Each interface of a COT is paired with a corresponding subscriber interface at the RT so the use of a UDLC is transparent to both the switch and the subscriber. In its simplest mode of operation, the UDLC uses pure multiplexing between the COT and the RT so that there is a one-to-one correspondence between a particular TDM channel and the COT/RT interface pair. Some systems can also be configured with concentration wherein the

COT/RT pairs are dynamically assigned transmission channels. If the number of requested conversations exceeds the number of channels, blocking occurs. The possibility of blocking introduces non transparency and implies that some means of returning a reorder tone (fast busy) is needed in the RT.

UDLC installations are configured to match each particular interface of the central office switch with a complimentary interface in the RT. A fully capable system must provide a wide variety of interfaces such as loop-start line, ground-start PBX trunk, foreign exchange lines, and coin telephone interfaces. In some early systems the configuration process involved nothing more than physically installing matching interfaces in respective equipment slots of the COT and RT. More recent systems typically utilize line units with multiple-service capabilities. These systems can be configured electronically (i.e., no straps) with either a local or a remote management interface.

Integrated Digital Loop Carrier Systems

Whenever a universal digital loop carrier system is interfaced with a digital switch, obvious inefficiencies occur in terms of back-to-back demultiplexing-multiplexing and D/A-A/D conversion. As shown in Figure, an integrated digital loop carrier (IDLC) system eliminates the inefficiency by directly connecting the DLC TDM link to the digital matrix. Typically the direct digital connections are either DS1 or El cross-connect signals. Thus, a fiber-based DLC system will typically interface with the switch through multiplexing/demultiplexing equipment as some number of digital cross-connect signals.



Integrated digital loop carrier system.

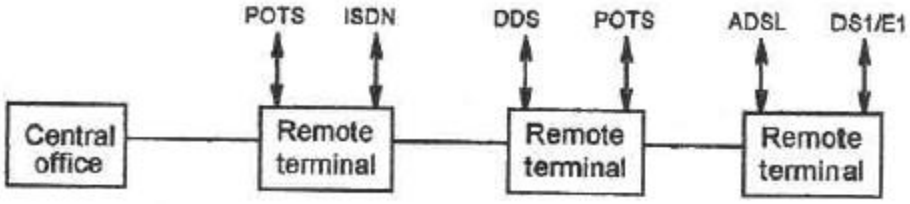
From a functional point of view an early IDLC systems was nothing more than a distributed switching system wherein some line interfaces of the switch are moved to remote locations. Most central office switch manufacturers provide such a capability— sometimes as simply as using channel banks for analog line interfaces that can be colocated with the matrix or remoted. Central office switch vendors also offer remote switching modules wherein some portion of the matrix itself is remotely located. These configurations provide remote concentration for efficient use of the connecting transmission link and, in some cases, provide local switching in the remote module so connections between two ports of a remote module do not use the transmission link. (If remote local switching is not provided, a connection between two ports of a remote module requires two channels of the transmission link.)

Remote modules of a particular switching system vendor are often implemented with proprietary signaling and management processes that preclude use of DLC equipment from other vendors (except when a DLC vendor develops the ability to emulate a particular switch vendor's remote modules). In the interest of opening IDLC applications to competing vendors, Bellcore established an IDLC standard referred to as GR-303 that compliant switching system vendors must support (possibly in addition to a proprietary IDLC capability). Included in the GR-303 specification are definitions for signaling, provisioning, testing, alarm surveillance, and performance monitoring.

Due, in part, to a move to unbundle LEC local loop services GR-303 has assumed a much broader scope than just an IDLC application. Because a GR-303 capability includes being able to define and administer a myriad of switch interface types, the GR- 303 standard can be used for interfacing other types of equipment such as xDSL equipment. ETSI has established a similar IDLC standard for international (ITU) switching equipment referred to as a V5 interface.

Next-Generation Digital Loop Carrier Systems

The term next-generation digital loop carrier (NGDLC) has been adopted within the industry to refer to DLC systems that adhere to GR-303 and provide additional configuration options and interfaces. There is no precise definition of what constitutes an NGDLC system. Four basic attributes are adherence to GR-303, optical fiber transmission capabilities (e.g., SONET/SDH), generally larger line sizes, and an ability to interface with a central office operational support system for diagnostics, alarms, and remote provisioning [14]. Other aspects commonly available in NGDLC systems are depicted .



Next-generation digital loop carrier.

The most important aspect of an NGDLC system from a services point of view is the availability of new services such as copper or fiber digital subscriber interfaces in addition to the conventional POTS interfaces. The most desired digital interfaces are Tl/El, primary rate and basic rate ISDN, ADSL, and VDSL. Other digital interfaces may also be provided for services like broadband data and digital video. Although the diversity of interfaces belies use of a single multiple service line interface for total electronic provisioning, the system must report inconsistencies between installed hardware and the electronic database. A particularly desirable feature of the system shown in Figure is drop-and-insert capability, which allows distributed access to a single backbone digital route. GR-303 identifies star configurations, linear ADM distribution, and ADM rings as desirable topologies. NGDLC systems that support advanced interfaces or topologies necessarily use a COT to separate out the special services from the POTS. In essence, the COT performs cross-connect operations to groom and distribute various types of traffic.

FIBER IN THE LOOP

Fiber in the loop (MIL) is a generic term that refers to one of three more specific descriptions of the use of fiber for local distribution. The first category is fiber to the cabinet or fiber to the neighborhood. These systems are often installed by local telephone companies as part of the feeder portion of their local distribution facilities. Traditional copper pairs for voice service extend from the cabinet to subscriber residences. Because a relatively long copper pair is used for “the last mile,” they do not provide much opportunity for enhanced service offerings. Their use and justification are strictly based on reducing the cost of POTS distribution.

The second category of FTTL is a fiber-to-the-curb (FTTC) system. As the name implies, these systems are designed to reach within 1000 feet of a subscriber residence. An FTTC system is generally intended to provide enhanced services such as video or high-speed data using ADSL or VDSL. Distribution of the enhanced services from the “curb” location is carried over wire pairs or coaxial cable. These systems are essentially identical to advanced DLC systems with optical transport.

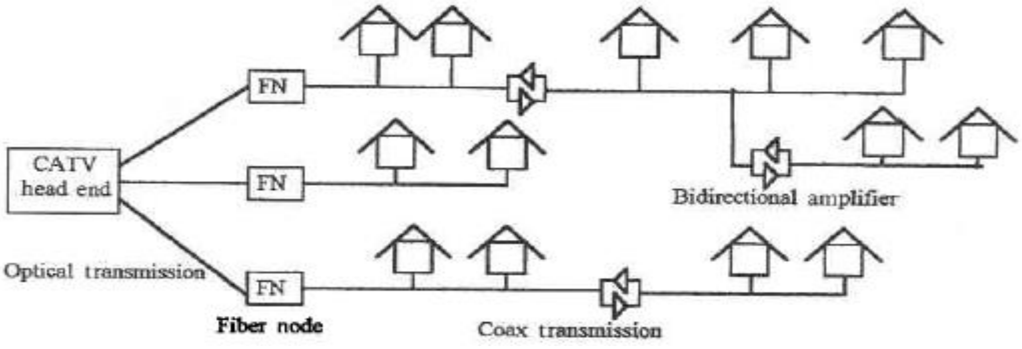
The third category of FITL is fiber to the home (FTTH). These systems obviously offer opportunities for extremely large bandwidths to the home but have significant deployment obstacles. First, installation in established neighborhoods is expensive because underground installations (under streets and driveways) are normally required. Second, providing network power to subscriber equipment is a major problem. The lack of power for enhanced service applications is not much of a consideration but power to telephones is. Local exchange carriers go to great efforts to ensure independence from commercial power systems for both their office switching systems and the connected subscribers. (A central office typically maintains enough batteries to keep a system up for 24 hours if commercial power is lost. If power is out for longer than this, diesel generators are available.) A further complication of providing telephone service over FTTH is the need to convert a digital voice channel to analog for interfacing to conventional telephones—an additional expense and power problem.

HYBRID FIBER COAX SYSTEMS

The cable TV systems installed around the country can be augmented with downstream data transmission to subscribers by merely adding “cable modems” utilizing unused or displaced TV channel bandwidth. Upstream transmission from the subscriber to a cable TV head end is much more difficult. Although many cable TV systems were designed and installed with upstream transmission as an option, the bandwidth available to the upstream channels is generally limited and often subject to very high noise and interference levels. Upstream transmission limitations can be somewhat alleviated by utilizing a conventional telephone connection with voiceband modems with data rates up to 28.8 kbps. The telephone modem connection is used in the same basic manner as the upstream channel of an ADSL while relative high- bandwidth downstream data are carried on the cable. The disadvantages of this solution include the need for a subscriber telephone line, the cost of large numbers of telephone channel connections into a service provider, and the need to coordinate dialup telephone connections with particular cable channel users.

The basic configuration of a hybrid fiber coax (HFC) system is depicted in Figure. The coaxial cable portions of CATV systems are configured as tree-and-branch topologies with all customers receiving the same, multichannel broadcast signal. Amplifiers are inserted wherever the signal level gets unacceptably low from attenuation and branching loses. Bidirectional amplifiers are shown in Figure under the assumption that this system is an application with a return path from the residences. In TV-only applications the return path (if there is one) is used for premium channel selection. In expanded service applications the return path carries voice or data with frequency division multiplexed cable modems in a band from 5 to 42 MHz.

The optical fiber transmission portion of an HFC system represents a replacement of relatively long haul coaxial cable sections with numerous amplifiers. For this reason, the optical transmitters and receivers are designed to carry a wideband analog signal. Notice that the optical links are shared by a large number of customers (anywhere



Hybrid fiber coax system configuration.

from 100 to 1500). Upgrading an HFC system for new services typically requires greater penetration of the optical fiber portions so that fewer households are connected to a common coaxial cable segment. In the limiting situation, wherein each household is connected through a dedicated coaxial cable, an HFC system becomes an FTTC system.

Downstream digital services can utilize cable modems that typically pack 30-40 Mbps into a 6-MHz analog TV channel. 64-QAM modulation is commonly used. In newer HFC systems, new downstream digital services can be carried at frequencies above 450 MHz while the band from 54 to 450 MHz is reserved for traditional analog TV.

A major impediment to upgrading an HFC system for return channel services is the shared use of a coaxial cable segment common to some number of households. The network termination within each home is passive and bidirectional, which means that all noise and interference within a home is passed onto the common cable to all other homes. Thus, a single source of interference can disrupt the signal to all other homes served by the common coaxial cable. Furthermore, the noise and interference of all households are additive, indicating the need to limit the number of households served by a single coax segment. An additional drawback of the shared cable is the need for some form of encryption for content security. TDMA return channels help minimize the interference problem by blocking all output energy from a residence except when an allotted time slot occurs.

Using an HFC system for POTS has the same basic drawbacks as an FTTH system in that there is no inherent facility for line powering the telephones. Thus, HFC might not be used for primary (“lifeline”) POTS distribution but could be quite effective in providing secondary telephone applications. The main attribute of HFC for enhanced services is the ability to provide dynamic assignment of high-rate digital downstream channels and relatively low rate full-period upstream channels.

VOICEBAND MODEMS

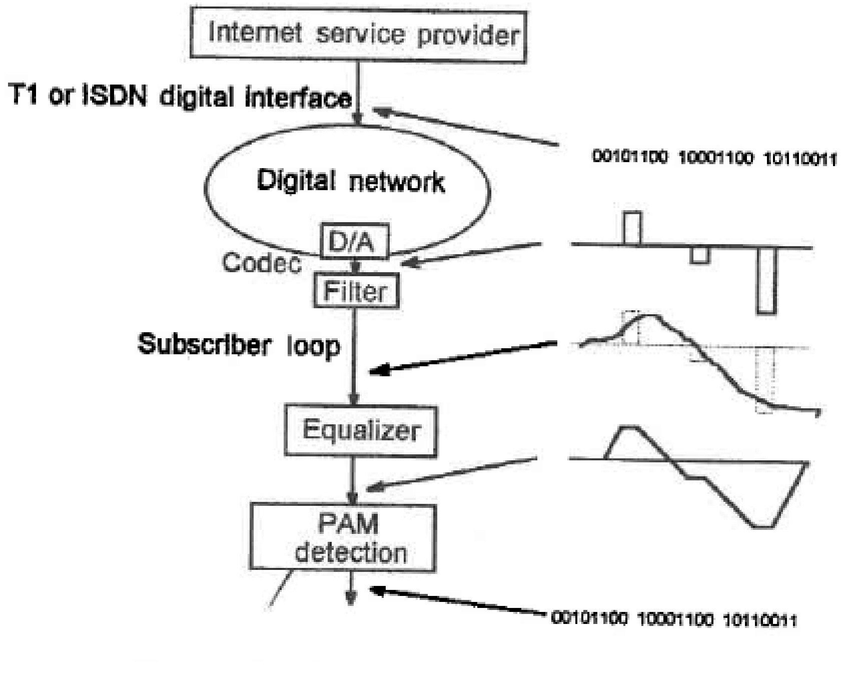
Voiceband modem technology improved dramatically in the early 1990s with the culmination of 33.6 kbps becoming standard with ITU recommendation V.34, The rapid advance of voiceband modem performance was due to two primary factors: the availability of economical DSP technology for equalization/echo canceling and the improved quality of the network in terms of lower noise and distortion resulting from the near-all-digital implementation. In an all-digital network the only significant source of noise is the quantization noise of the analog-to-digital conversion.

PCM Modems

A V.34 voiceband modem provides data rates that are near the theoretical limit imposed by quantization noise alone. Recognition that the principal source of noise in the end-to-end connection is the quantization noise of the A/D converters leads to alternative modem implementations that directly utilize the digital 64-kbps channel and eliminate the quantization noise. These modems are commonly referred to as PCM modems.

As shown in Figure, a V.90 PCM modem relies on the digital network to carry an unaltered digital signal from a digital source to a digital-to-analog conversion device (codec) at an analog subscriber interface. The codec converts the PCM codewords to PAM samples that are detected by the receiving customer premises modem and converted back to the original PCM data. Successful data detection requires the receiving circuitry to adequately equalize the combined distortion of the D/A smoothing filter and the transmission link, to know the quantization levels of the codec, and to become synchronized to the D/A conversion clock. The equalization and quantization requirements are determined during an initialization process while clock synchronization requires processing of data transitions in the multilevel received waveform.

It might seem that a PCM modem could provide a data rate of 64 kbps. However, several factors restrict the data rate to something less than 64 kbps. The first of these is the bandpass filter in the D/A codec (for 60 Hz elimination and sample smoothing). A second constraint is the possibility that the digital path through the network might



Downstream V.90 modem concept

include a digital pad for changing the signal level of the assumed analog signal/ A third constraint is the possibility (in North America) that robbed bit signaling might be in use on one or more of the digital links. 18 DIGITAL SUBSCRIBER ACCESS

The fact that the overall bandwidth of the channel is slightly over 3kHz implies that the maximum, intersymbol, interference-free pulse rate is just over 6 kHz. Thus, the actual sample rate of 8 kHz implies that some amount of intersymbol interference is inevitable (assuming the PCM samples are independent).

The lack of a low-frequency transmission response is accommodated by V.90 modems by utilizing every eighth PCM sample solely for dc restoration. Thus, the maximum data rate is 56 kbps. If robbed bit signaling is present, its effects are minimized by determining, during initialization, which frames in the six frame sequences are signaling frames and then using only 7 bits per sample in those frames and 8 bits per sample in the nonsignaling frames. Digital pads can be accommodated by detecting their presence and modifying the digital codewords according to the particular amount of attenuation being inserted by the pad.

The V.90 uplink direction is implemented as a conventional (V.34) modem, thereby implying asymmetry in the data rates. It is conceivable that PAM signaling could be used in both directions, but the uplink is more complicated to implement and is often unnecessary because most applications (e.g., Internet access) are inherently asymmetric in the data rates required. It is also possible to utilize analog PAM on both ends of the connection (as opposed to just one end), but these implementations are considerably more complicated.

The main attraction of PCM modems is that they provide almost the same data rate as an ISDN B channel but do not require changes in the line interfaces or special treatment of the customer loop (e.g., the elimination of bridged taps). When the customer loop is a short drop from a remote terminal of a digital loop carrier, the maximum data rate of 53-56 kbps is assured.

LOCAL MICROWAVE DISTRIBUTION SERVICE

The FCC has allocated two separate microwave frequency bands for digital radio access communications services. The first of these is Multichannel Multipoint Distribution Service (MMDS), which operates at 2 GHz. MMDS is essentially a wireless cable system and, as such, provides only one-way transmission. Reverse channel communications requires dial-up modem connections through the telephone network.

Local Microwave Distribution Service (LMDS) operates at 28 GHz and provides cell-based, two-way communications. Bandwidths available with LMDS are dependent on a particular service provider’s frequency plan. Typically, these plans provide bidirectional data rates on the order of 1.5-

50 Mbps, although the total allocated bandwidth of almost 1 GHz can be partitioned to provide higher rates or asymmetric rates if desired. International LMDS allocations have been made in a range of 25-31 GHz. These systems are also referred to as Local Multipoint Communications Systems (LMCS) in Canada and Brazil.

The principal applications supported by LMDS are voice, video, and high-bandwidth data. The immense amount of bandwidth available is the main attraction. The major cost of an LMDS system is associated with the subscriber radio equipment. Thus, after the cell sites are established, additional costs are incurred only as subscribers begin paying for service. This situation is in contrast with new fiber-based facilities that require major, up-front investment before revenues are realized. The most immediate opportunity for LMDS is to offer an alternative to high-speed digital leased lines for businesses. High-speed Internet access and HDTV to individual residences are secondary opportunities.

The major disadvantages of LMDS are the need for FCC licensing, line-of-sight transmission, distance limits of 2-3 miles, extreme attenuation in heavy rainfalls, potential interference from other services such as satellites in the same bands, and the need for local power in all subscriber (telephone) sites. Fade margins or dynamic power control on the order of 40 dB overcome all but the most severe rainfall rates.

The requirement for line-of-sight transmission means that a single base station cannot communicate with all locations in a cell (except maybe in West Texas or the Australian outback). Overlapping cells provide greater coverage, but some locations are sure to be in the shadow of all base stations unless reflectors can be installed. Line-of- sight transmission can be further compromised by the appearance of buildings or trees after initial deployment.

DIGITAL SATELLITE SERVICES

Direct Satellite Service (DSS) has recently become a viable alternative to cable TV services. Because the DSS system utilizes digital transmission for its TV channels, it is straightforward for a DSS provider to offer data communications services to the TV subscribers. However, because DSS is a one-way communications service, dial-up modem connections through the telephone network are required for two-way data communications. DSS is also limited to the total bandwidth (400 MHz) of the satellite transponders, which must be shared by the TV transmissions and any allocated data channels.

In contrast to DSS, which uses geostationary satellite orbits, Low Earth Orbit Satellites of Iridium and Teledesic provide opportunities for lower cost, bidirectional user terminals

UNIT-V

TRAFFIC ANALYSIS

Except for station sets and their associated loops, a telephone network is composed of a variety of common equipment such as digit receivers, call processors, interstage switching links, and interoffice trunks. The amount of common equipment designed into a network is determined under an assumption that not all users of the network need service at one time. The exact amount of common equipment required is unpredictable because of the random nature of the service requests. Networks conceivably could be designed with enough common equipment to instantly service all requests except for occurrences of very rare or unanticipated peaks. However, this solution is uneconomical because much of the common equipment is unused during normal network loads. The basic goal of traffic analysis is to provide a method for determining the cost-effectiveness of various sizes and configurations of networks.

Traffic in a communications network refers to the aggregate of all user requests being serviced by the network. As far as the network is concerned, the service requests arrive randomly and usually require unpredictable service times. The first step of traffic analysis is the characterization of traffic arrivals and service times in a probabilistic framework. Then the effectiveness of a network can be evaluated in terms of how much traffic it carries under normal or average loads and how often the traffic volume exceeds the capacity of the network.

The techniques of traffic analysis can be divided into two general categories: loss systems and delay systems. The appropriate analysis category for a particular system depends on the system's treatment of overload traffic. In a loss system overload traffic is rejected without being serviced. In a delay system overload traffic is held in a queue until the facilities become available to service it. Conventional circuit switching operates as a loss system since excess traffic is blocked and not serviced without a retry on the part of the user. In some instances "lost" calls actually represent a loss of revenue to the carriers by virtue of their not being completed.

(12.14)

Store-and-forward message or packet switching obviously possesses the basic characteristics of a delay system. Sometimes, however, a packet-switching operation can also contain certain aspects of a loss system. Limited queue sizes and virtual circuits both imply loss operations during traffic overloads. Circuit-switching networks also incorporate certain operations of a delay nature in addition to the loss operation of the circuits themselves. For example, access to a digit receiver, an operator, or a call processor is normally controlled by a queuing process.

The basic measure of performance for a loss system is the probability of rejection (blocking probability). A delay system, on the other hand, is measured in terms of service delays. Sometimes the average delay is desired, while at other times the probability of the delay exceeding some specified value is of more interest.

TRAFFIC CHARACTERIZATION

Because of the random nature of network traffic, the following analyses involve certain fundamentals of probability theory and stochastic processes. In this treatment only the most basic assumptions and results of traffic analysis are presented. The intent is to provide an indication of how to apply results of traffic analysis, not to delve deeply into analytical formulations. However, a few basic derivations are presented to acquaint the user with assumptions in the models so they can be appropriately applied.

In the realm of applied mathematics, where these subjects are treated more formally, blocking probability analyses are referred to as congestion theory and delay analyses are referred to as queuing theoiy. These topics are also commonly referred to as traffic flow analysis. In a circuit-switched network, the “flow” of messages is not so much of a concern as are the holding times of common equipment. A circuit- switched network establishes an end-to-end circuit involving various network facilities (transmission links and switching stages) that are held for the duration of a call. From a network point of view, it is the holding of these resources that is important, not the flow of information within individual circuits. '

On the other hand, message-switching and packet-switching networks are directly concerned with the actual flow of information, since in these systems traffic on the transmission links is directly related to the activity of the sources.

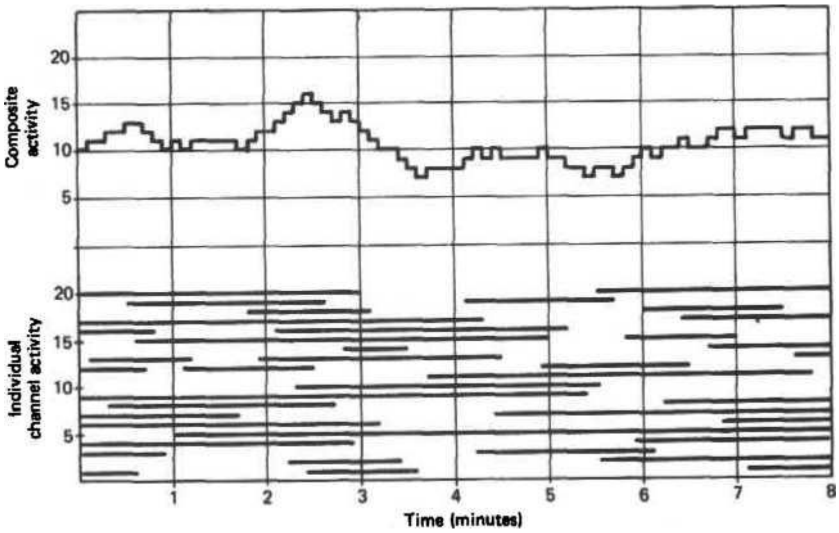
Circuit switching does involve certain aspects of traffic flow in the process of setting up a connection. Connect requests flow from the sources to the destinations acquiring, holding, and releasing certain resources in the process. As was discussed, controlling the flow of connect requests during network overloads is a vital function of network management.

The unpredictable nature of communications traffic arises as a result of two underlying random processes: call arrivals and holding times. An arrival from any particular user is generally assumed to occur purely by chance and be totally independent of arrivals from other users. Thus the number of arrivals during any particular time interval is indeterminate. In most cases holding times are also distributed randomly. In some applications this element of randomness can be removed by assuming constant holding times (e.g., fixed-length packets). In either case the traffic load presented to a network is fundamentally dependent on both the frequency of arrivals and the average holding time for each arrival. Figure 12.1 depicts a representative situation in which both the arrivals and the holding times of 20 different sources are unpredictable. The bottom of the figure depicts activity of each individual source while the top displays the instantaneous total of all activity. If we assume that the 20 sources are to be connected to a trunk group, the activity curve displays the number of circuits in use at any particular time. Notice that the maximum number of circuits in use at any one time is 16 and the average utilization is a little under 11 circuits. In general terms, the trunks are referred to as servers, and a trunk group is a server group.

Traffic Measurements

One measure of network capacity is the volume of traffic carried over a period of time. Traffic volume is essentially the sum of all holding times carried during the interval. The traffic volume represented in Figure 12.1 is the area under the activity curve (approximately 84 call minutes).

A more useful measure of traffic is the traffic intensity (also called traffic flow). Traffic intensity is obtained by dividing the traffic volume by the length of time during which it is measured. Thus traffic intensity represents the average activity during a period of time (10.5 in Figure 12.1). Although traffic intensity is fundamentally dimen- sionless (time divided by time), it is usually expressed in units of erlangs, after the Danish pioneer traffic theorist A. K. Erlang, or in terms of hundred (century) call seconds per hour (CCS). The relationship between erlangs and CCS units can be derived by observing that there are 3600 sec in an hour:



Activity profile of network traffic (all calls carried).

The maximum capacity of a single server (channel) is 1 erlang, which is to say that die server is always busy. Thus the maximum capacity in erlangs of a group of servers is merely equal to the number of servers. Because traffic in a loss system experiences infinite blocking probabilities when the traffic intensity is equal to the number of servers, the average activity is necessarily less than the number of servers. Similarly, delay systems operate at less than full capacity, on average, because infinite delays occur when the average load approaches the number of servers.

$$1 \text{ erlang} = 36 \text{ CCS}$$

Two important parameters used to characterize traffic are the average arrival rate A, and the average holding time f_m. If the traffic intensity X is expressed in erlangs, then

$$A = \lambda t_{\mathrm{m}}$$

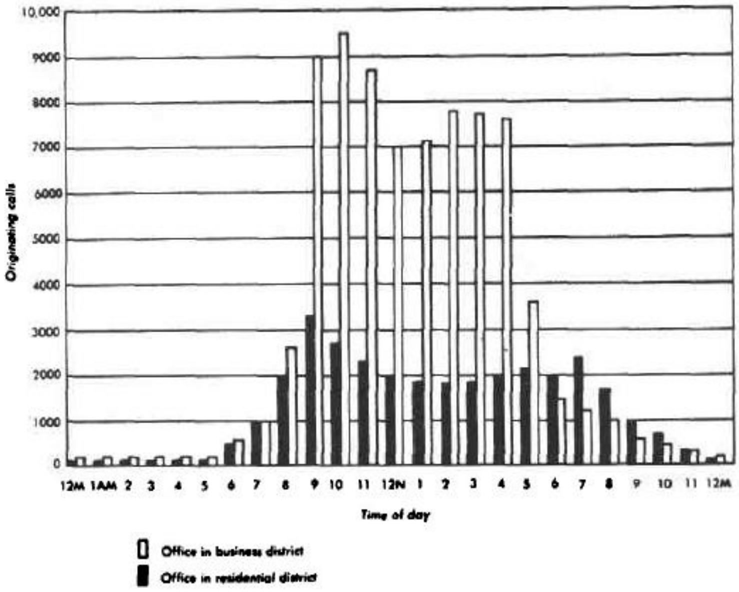
where *k* and t_m are expressed in like units of time (e.g., calls per second and seconds per call, respectively).

Notice that traffic intensity is only a measure of average utilization during a time period and does not reflect the relationship between arrivals and holding times. That is, many short calls can produce the same traffic intensity as a few long ones. In many of the analyses that follow the results are dependent only on the traffic intensity. In some cases, however, the results are also dependent on the individual arrival patterns and holding time distributions.

Public telephone networks are typically analyzed in terms of the average activity during the busiest hour of a day. The use of busy-hour traffic measurements to design and analyze telephone networks represents a compromise between designing for the overall average utilization (which includes virtually unused nighttime hours) and designing for short-duration peaks that may occur by chance or as a result of TV commercial breaks, radio call-in contests, and so on.

Busy-hour traffic measurements indicate that an individual residential telephone is typically in use between 5 and 10% of the busy hour. Thus each telephone represents a traffic load of between 0.05 and 0.10 erlangs.

Business telephones usually produce loading patterns different from residential phones. First, a business phone is generally utilized more heavily. Second, the busy hour of business traffic is often different from the busy hour of residential traffic. Figure 12.2 shows a typical hourly variation for both sources of traffic. The trunks of a telephone network are sometimes designed to take advantage of variations in calling patterns from different offices. Toll connecting trunks from residential areas are often busiest during evening hours, and trunks from business areas are obviously busiest during midmoming or midafternoon. Traffic engineering depends not only on overall traffic volume but also on time-volume traffic patterns within the network.



Traffic volume dependence on time of day.

A certain amount of care must be exercised when determining the total traffic load of a system from the loading of individual lines or trunks. For example, since two phones are involved in each connection, the total load on a switching system is exactly one-half the total of all traffic on the lines connected to the switch. In addition, it may be important to include certain setup and release times into the average holding times of some common equipment. A 10-sec setup time is not particularly significant for a 4-min voice call but can actually dominate the holding time of equipment used for short data messages. Common equipment setup times also become more significant in the presence of voice traffic overloads. A greater percentage of the overall load is represented by call attempts since they increase at a faster rate than completions.

An important distinction to be made when discussing traffic in a communications network is the difference between the offered traffic and the carried traffic. The offered traffic is the total traffic that would be carried by a network capable of servicing all requests as they arise. Since economics generally precludes designing a network to immediately cany the maximum offered traffic, a small percentage of offered traffic typically experiences network blocking or delay. When the blocked calls are rejected by the network, the mode of operation is referred to as blocked calls cleared or lost calls cleared. In essence, blocked calls are assumed to disappear and never return. This assumption is most appropriate for trunk groups with alternate routes. In this case a blocked call is normally serviced by another trunk group and does not, in fact, return.

The carried traffic of a loss system is always less than the offered traffic. A delay system, on the other hand, does not reject blocked calls but holds them until the necessary facilities are available. With the assumption that the long-term average of offered traffic is less than the capacity of the network, a delay system carries all offered traffic. If the number of requests that can be waiting for service is limited, however, a delay system also takes on properties of a loss system. For example, if the queue for holding blocked arrivals is finite, requests arriving when the queue is full are cleared.

Arrival Distributions

The most fundamental assumption of classical traffic analysis is that call arrivals are independent. That is, an arrival from one source is unrelated to an arrival from any other source. Even though this assumption may be invalid in some instances, it has general usefulness for most applications. In those cases where call arrivals tend to be correlated, useful results can still be obtained by modifying a random arrival analysis. In this manner the random arrival assumption provides a mathematical formulation that can be adjusted to produce approximate solutions to problems that are otherwise mathematically intractable.

Negative Exponential Interarrival Times

Designate the average call arrival rate from a large group of independent sources (subscriber lines) as *X*. Use the following assumptions:

1. Only one arrival can occur in any sufficiently small interval.
2. The probability of an arrival in any sufficiently small interval is directly proportional to the length of the interval. (The probability of an arrival is *X At*, where *At* is the interval length.) ’
3. The probability of an arrival in any particular interval is independent of what has occurred in other intervals.

It is straightforward [1] to show that the probability distribution of interamval times is

$$P_0(\lambda t) = e^{-\lambda t}$$

Equation defines the probability that no arrivals occur in a randomly selected interval *t.* This is identical to the probability that *t* seconds elapse from one arrival to the next.

A more subtle implication of the independent arrival assumption involves the number of sources, not just their calling patterns. When the probability of an arrival in any small time interval is independent of other arrivals, it implies that the number of sources available to generate requests is constant. If a number of arrivals occur immediately before any subinterval in question, some of the sources become busy and cannot generate requests. The effect of busy sources is to reduce the average arrival rate. Thus the interarrival times are always somewhat larger than what Equation predicts them to be. The only time the arrival rate is truly independent of source activity is when an infinite number of sources exist.

If the number of sources is large and their average activity is relatively low, busy sources do not appreciably reduce the arrival rate. For example, consider an end office that services 10,000 subscribers with 0.1 erlang of activity each. Normally, there are 1000 active links and 9000 subscribers available to generate new arrivals. If the number of active subscribers increases by an unlikely 50% to 1500 active lines, the number of idle subscribers reduces to 8500, a change of only 5.6%. Thus the arrival rate is relatively constant over a wide range of source activity. Whenever the arrival rate is fairly constant for the entire range of normal source activity, an infinite source assumption is justified.

It is pointed out that Lee graph analyses overestimate the blocking probability because, if some number of interstage links in a group are known to be busy, the remaining links in the group are less likely to be busy. A Jacobaeus analysis produces a more rigorous and accurate solution to the blocking probability, particularly when space expansion is used. Accurate analyses of interarrival times for finite sources are also possible. These are included in the blocking analyses to follow.

Poisson Arrival Distribution

Equation merely provides a means of determining the distribution of interarrival times. It does not, by itself, provide the generally more desirable information of how many arrivals can be expected to occur in some arbitrary time interval. Using the same assumptions presented, however, the probability of *j* arrivals in an interval *t* can be determined as

Equation 12.3 is the well-known Poisson probability law. Notice that when *j* = 0, the probability of no arrivals in an interval *t* is $P_0(t)$, as obtained in Equation 12.2. ' Again, Equation 12.3 assumes arrivals are independent and occur at a given average rate irrespective of the number of arrivals occurring just prior to an interval in question. Thus the Poisson probability distribution should only be used for arrivals from a large number of independent sources.

$$P_j(\lambda t) = \frac{(\lambda t)^j}{j!} e^{-\lambda t}$$

Equation defines the probability of experiencing exactly *j* arrivals in *t* seconds. Usually there is more interest in determining the probability of *j* or more arrivals in *t* seconds:

is

$$P_{\geq j}(\lambda t) = \sum_{i=j}^{\infty} P_i(\lambda t)$$

$$= 1 - \sum_{i=0}^{j-1} P_i(\lambda t)$$

$$= 1 - P_{<j}(\lambda t)$$

Holding Time Distributions

The second factor of traffic intensity as specified in Equation 12.1 is the average holding time t_m. In some cases the average of the holding times is all that needs to be known about holding times to determine blocking probabilities in a loss system or delays in a delay system. In other cases it is necessary to know the probability distribution of the holding times to obtain the desired results. This section describes the two most commonly assumed holding time distributions: constant holding times and exponential holding times.

Constant Holding Times

Although constant holding times cannot be assumed for conventional voice conversations, it is a reasonable assumption for such activities as per-call call processing requirements, interoffice address signaling, operator assistance, and recorded message playback. Furthermore, constant holding times are obviously valid for transmission times in fixed-length packet networks.

When constant holding time messages are in effect, it is straightforward to use Equation 12.3 to determine the probability distribution of active channels. Assume, for the time being, that all requests are serviced. Then the probability of *j* channels being busy at any particular time is merely the probability that *j* arrivals occurred in the time interval of length t_m immediately preceding the instant in question. Since the average number of active circuits over all time is the traffic intensity A = Xt_m, the probability ofy circuits being busy is dependent only on the traffic intensity:

$$P_j(\lambda t_{\mathrm{m}}) = P_j(A)$$

$$= \frac{A^j}{j!} e^{-A}$$

where $\lambda =$ arrival rate
$t_{\mathrm{m}} =$ constant holding time
$A =$ traffic intensity (erlangs)

Exponential Holding Times

The most commonly assumed holding time distribution for conventional telephone conversations is the exponential holding time distribution:

$$P(>t) = e^{-t/t_{\mathrm{m}}}$$

where r_m is the average holding time. Equation 12.6 specifies the probability that a holding time exceeds the value *t*. This relationship can be derived from a few simple assumptions concerning the nature of the call termination process. Its basic justification, however, lies in the fact that observations of actual voice conversations exhibit a remarkably close correspondence to an exponential distribution.

The exponential distribution possesses the curious property that the probability of a termination is independent of how long a call has been in progress. That is, no matter how long a call has been in existence, the probability of it lasting another *t* seconds is defined by Equation 12.6. In this sense exponential holding times represent the most random process possible. Not even knowledge of how long a call has been in progress provides any information as to when the call will terminate.

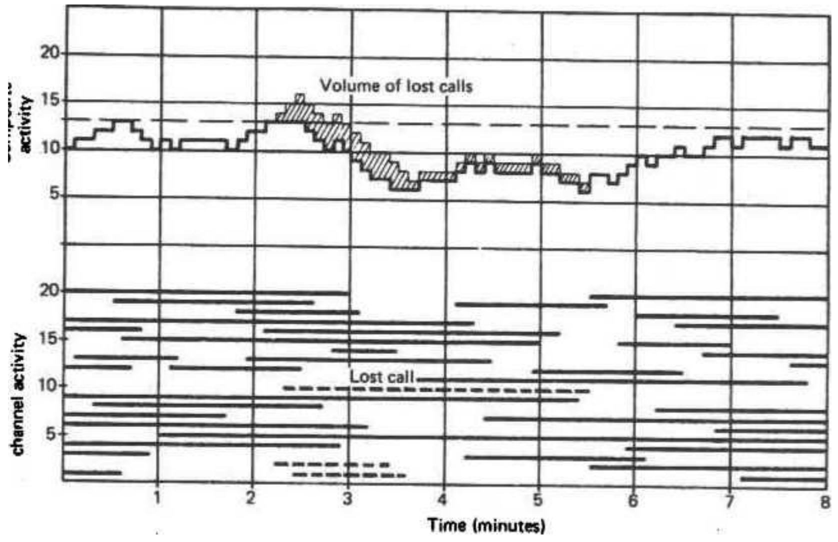
Combining a Poisson arrival process with an exponential holding time process to obtain the probability distribution of active circuits is more complicated than it was for constant holding times because calls can last indefinitely. The final result, however, proves to be dependent on only the average holding time. Thus Equation 12.5 is valid for exponential holding times as well as for constant holding times (or any holding time distribution). Equation 12.5 is therefore repeated for emphasis: The probability of *j* circuits being busy at any particular instant, assuming a Poisson arrival process and that all requests are serviced immediately, is

$$P_j(A) = \frac{A^j}{j!} e^{-A}$$

where *A* is the traffic intensity in erlangs. This result is true for any distribution of holding times.

LOSS SYSTEMS

The basic reason for the discrepancy is indicated in Figure 12.3, which depicts the same traffic pattern arising from 20 sources as is shown previously in Figure 12.1. Figure 12.3, however, assumes that only 13 circuits are available to carry the traffic. Thus the three arrivals at *t* — 2.2, 2.3, and 2.4 min are blocked and assumed to have left the system. The total amount of traffic volume lost is indicated by the shaded area, which is the difference between all traffic being serviced as it arrives and traffic being carried by a blocked calls cleared system with 13 circuits. The most important feature to notice in Figure 12.3 is that the call arriving at *t* - 2.8 is not blocked, even though the original profile indicates that it arrives when all 13 circuits are busy. The reason it is not blocked is that the previously blocked calls left the system and therefore reduced the congestion for subsequent arrivals. Hence the percentage of time that the original traffic profile is at or above 13 is not the same as the blocking probability when only 13 circuits are available.



12.2.1 Lost Calls Cleared

The first person to account fully and accurately for the effect of cleared calls in the calculation of blocking probabilities was A. K. Erlang in 1917. In this section we discuss Erlang’s most often used result: his formulation of the blocking probability for a lost calls cleared system with Poisson arrivals. Recall that the Poisson arrival assumption implies infinite sources. This result is variously referred to as Erlang’s formula of the first kind, #ijv(A); the Erlang-fl formula; or Erlang’s loss formula.

A fundamental aspect of Erlang’s formulation, and a key contribution to modem stochastic process theory, is the concept of statistical equilibrium. Basically, statistical equilibrium implies that the probability of a system’s being in a particular state (number of busy circuits in a trunk group) is independent of the time at which the system is examined. For a system to be in statistical equilibrium, a long time must pass (several average holding times) from when the system is in a known state until it is again examined. For example, when a trunk group first begins to accept traffic, it has no busy circuits. For a short time thereafter, the system is most likely to have only a few busy circuits. As time passes, however, the system reaches equilibrium. At this point the most likely state of the system is to have $A = Xt_m$ busy circuits.

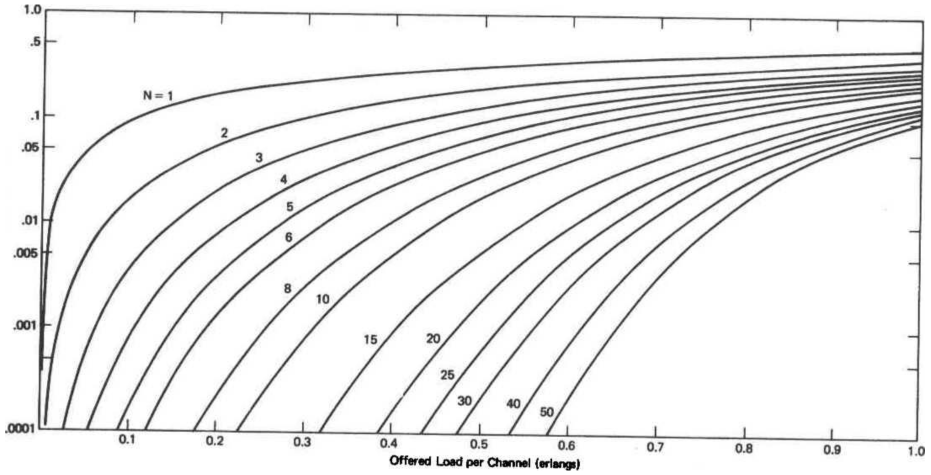
When in equilibrium, a system is as likely to have an arrival as it is to have a termination. If the number of active circuits happens to increase above the average *A*, departures become more likely than arrivals. Similarly, if the number of active circuits happens to drop below *A*, an arrival is more likely than a departure. Thus if a system is perturbed by chance from its average state, it tends to return.

Although Erlang's elegant formulation is not particularly complicated, it is not presented here because we are mostly interested in application of the results. The interested reader is invited to see reference [2] or [3] for a derivation of the result:

$$B = E_{1,N}(A) = \frac{A^N}{N! \sum_{i=0}^{N} (A^i/i!)}$$

where N = number of servers (channels)
A = offered traffic intensity, λt_m (erlangs)

Equation specifies the probability of blocking for a system with random arrivals from an infinite source and arbitrary holding time distributions. The blocking probability of Equation 12.8 is plotted in Figure 12.4 as a function of offered traffic intensity for various numbers of channels. An often more useful presentation of Erlang' s results is provided in Figure 12.5, which presents the output channel utilization for various blocking probabilities and numbers of servers. The output utilization p represents the traffic carried by each circuit:



Blocking probability of lost calls cleared system.

*(1*The greater circuit efficiency obtained by combining traffic into large groups is often referred to as the advantage of large group sizes. This efficiency of circuit utilization is the basic motivation for hierarchical switching structures. Instead of interconnecting a large number of nodes with rather small trunk groups between each pair, it is more economical to combine all traffic from individual nodes into one large trunk group and route the traffic through a tandem switching node. Figure 12.7 contrasts a mesh versus a star network with a centralized switching node at the center. Obviously, the cost of the tandem switch becomes justified when the savings in total circuit miles is large enough

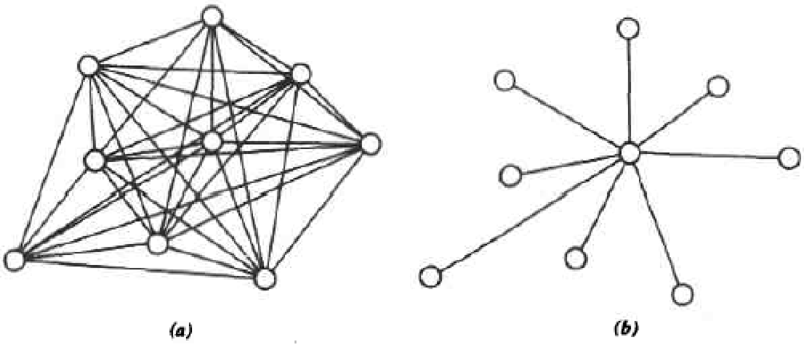


Figure 12.7 Use of tandem switching to concentrate traffic: (*a*) mesh; (*b*) star.

Accurate values of traffic intensities are not always available. Furthermore, even when accurate traffic measurements are obtainable, they do not provide an absolute indication of how much growth to expect. Thus only limited confidence can be attached to calculations of blocking probabilities in an absolute sense. The main value of these analyses is that they provide an objective means of comparing various network sizes and configurations. The most cost-effective design for a given grade of service is the one that should be chosen, even if the traffic statistics are hypothetical. If a network is liable to experience wildly varying traffic patterns or rapid growth, these factors must be considered when comparing design alternatives. A network with a somewhat larger initial cost may be more desirable if it can absorb or grow to accommodate unanticipated traffic volumes more easily.

LOST CALLS RETURNING In the lost calls cleared analyses just presented, it is assumed that unserviceable requests leave the system and never return. As mentioned, this assumption is most appropriate for trunk groups whose blocked requests overflow to another route and are usually serviced elsewhere. However, lost calls cleared analyses are also used in instances where blocked calls do not get serviced elsewhere. In many of these cases, blocked calls tend to return to the system in the form of retries. Some examples are subscriber concentrator systems, corporate tie lines and PBX trunks, calls to busy telephone numbers, and access to WATS lines (if DDD alternatives are not used). This section derives blocking probability relationships for lost calls cleared systems with random retries.The following analysis involves three fundamental assumptions regarding the nature of the returning calls:All blocked calls return to the system and eventually get serviced, even if multiple retries are required.The elapsed times between call blocking occurrences and the generation of retries are random and statistically independent of each other. (This assumption allows the analysis to avoid complications arising when retries are correlated to each other and tend to cause recurring traffic peaks at a particular waiting time interval.) The typical waiting time before retries occur is somewhat longer than the average holding time of a connection. This assumption essentially states that the system is allowed to reach statistical equilibrium before a retry occurs. Obviously, if retries occur too soon, they are very likely to encounter congestion since the system has not had a chance to “relax.” In the limit, if all retries are immediate and continuous, the network operation becomes similar to a delay system. In this case, however, the system does not queue requests—the sources do so by continually “redialing.” When considered in their entirety, these assumptions characterize retries as being statistically indistinguishable from first-attempt traffic.* Hence blocked calls merely add to the first-attempt call arrival rate. Consider a system with a first-attempt call arrival rate of *X*. If a percentage *B* of the calls is blocked, *B* times *X* retries will occur in the future. Of these retries, however, a percentage *B* will be blocked again. Continuing in this manner, the total arrival rate *X′* after the system has reached statistical equilibrium can be determined as the infinite series

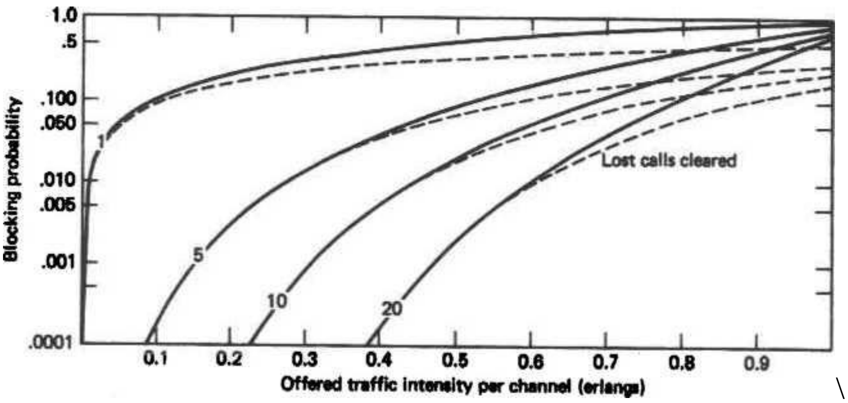
$$\lambda' = \lambda + B\lambda + B^2\lambda + B^3\lambda + \ldots$$

$$= \frac{\lambda}{1 - B}$$

where *B* is the blocking probability from a lost calls cleared analysis with traffic

cleared analysis of Equation 12.8. First, determine an estimate of *B* using *X* and then calculate *X'*. Next, use *V* to obtain a new value of *B* and an updated value of *V*. Continue in this manner until values of *X* and *B* are obtained.

When measurements are made to determine the blocking probability of an outgoing trunk group, the measurements cannot distinguish between first-attempt calls (demand traffic) and retries. Thus if a significant number of retries are contained in the measurements, this fact should be incorporated into an analysis of how many circuits must be added to reduce the blocking of an overloaded trunk group. The apparent offered load will decrease as the number of servers increases because the number of re- tries decreases. Thus fewer additional circuits are needed than if no retries are contained in the measurements.

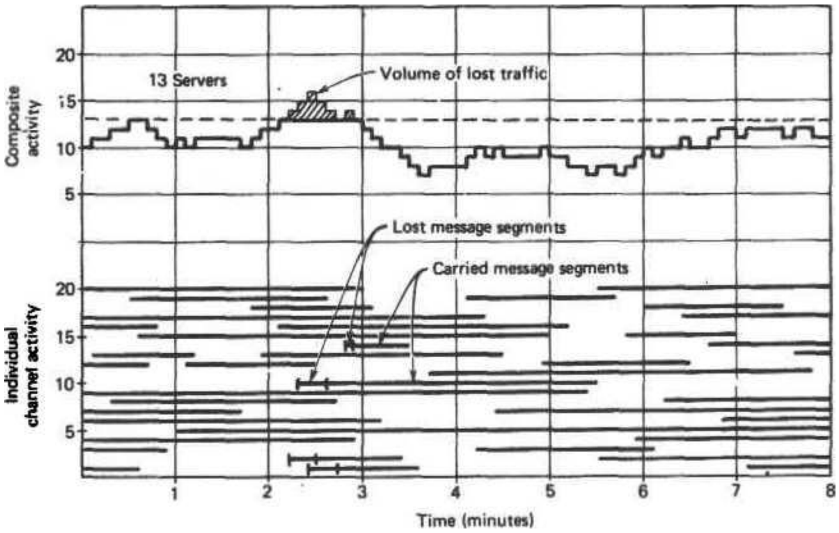


Lost Calls Held

In a lost calls held system, blocked calls are held by the system and serviced when the necessary facilities become available. Lost calls held systems are distinctly different from the delay systems discussed later in one important respect: The total elapsed time of a call in the system, including waiting time and service time, is independent of the waiting time. In essence, each arrival requires service for a continuous period of time and terminates its request independently of its being serviced or not. Figure 12.9 demonstrates the basic operation of a lost calls held system. Notice that most blocked calls eventually get some service, but only for a portion of the time that the respective sources are busy.

Although a switched telephone network does not operate in a lost calls held manner, some systems do. Lost calls held systems generally arise in real-time applications in which the sources are continuously in need of service, whether or not the facilities are available. When operating under conditions of heavy traffic, a lost calls held system typically provides service for only a portion of the time a particular source is active.

Even though conventional circuit switching does not operate according to the theoretical model of lost calls held, Bell System traffic engineers have used it to calculate blocking probabilities for trunk groups [4]. A lost calls held analysis always produces a larger value for blocking than does Erlang’s loss formula. Thus the lost calls held analysis produces a conservative design that helps account for retries and day-to-day variations in the busy-hour calling intensities. In contrast, CCITT recommendations [5] stipulate Erlang-B formulas should be used in determining blocking probabilities.



One example of a system that closely fits the lost calls held model is time assignment speech interpolation (TASI). A TASI system concentrates some number of voice sources onto a smaller number of transmission channels. A source receives service (is connected to a channel) only when it is active. If a source becomes active when all channels are busy, it is blocked and speech clipping occurs. Each speech segment starts and stops independently of whether it is serviced or not. TASI systems were originally used on analog long-distance transmission links such as undersea cables. More modem counterparts of TASI are referred to as digital circuit multiplication (DCM) systems. In contrast to the original TASI systems, DCM systems can delay speech for a small amount of time, when necessary, to minimize the clipping. In this case, a lost calls held analysis is not rigorously justified because the total time a speech segment is “in the system” increases as the delay for service increases. However, if the average delay is a small percentage of the holding time, or if the coding rate of delayed speech is reduced to allow the transmission channel time to “catch up,” a lost calls held analysis is still justified. Recall that controlling the coding rate is one technique of traffic shaping used for transporting voice in an ATM network.

Lost calls held systems are easily analyzed to determine the probability of the total number of calls in the system at any one time. Since the duration of a source’s activity is independent of whether it is being serviced, the number in the system at any timp i$_s$ identical to the number of active sources in a system capable of carrying all traffic as it arises. Thus the distribution of the number in the system is the Poisson distribution provided earlier in Equation 12.3. The probability that *i* sources requesting service are being blocked is simply the probability that $i + N$ sources are active when N i s the number of servers. Recall that the Poisson distribution essentially determines the desired probability as the probability that $i + N$ arrivals occurred in the preceding f$_{ra}$ seconds. The distribution is dependent only on the product of the average arrival rate X and the average holding time t_m.

$$\text{prob(clipping)} = \sum_{j=50}^{99} P_j(40) = 0.04$$

Example 12.9 demonstrates that TASI systems are much more effective for large group sizes than for small ones. The 36% clipping factor occurring with 5 channels produces unacceptable voice quality. On the other hand, the 4% clipping probability for 50 channels can be tolerated when the line costs are high enough.

In reality, the values for blocking probabilities obtained in Example 12.9 are overly pessimistic because an infinite source assumption was used. The summations did not include the case of all sources being active because there needs to be at least one idle source to create an arrival during the time congestion. A more accurate solution to this problem is obtained in a later section using a finite source analysis.

Lost Calls Cleared—Finite Sources

As mentioned previously, a fundamental assumption in the derivation of the Poisson arrival distribution, and consequently Erlang's loss formula, is that call arrivals occur independently of the number of active callers. Obviously, this assumption can be justified only when the number of sources is much larger than the number of servers. This section presents some fundamental relationships for determining blocking probabilities of lost calls cleared systems when the number of sources is not much larger than the number of servers. The blocking probabilities in these cases are always less than those for infinite source systems since the arrival rate decreases as the number of busy sources increases.

When considering finite source systems, traffic theorists introduce another parameter of interest called time congestion. Time congestion is the percentage of time that all servers in a group are busy. It is identical to the probability that all servers are busy at randomly selected times. However, time congestion is not necessarily identical to blocking probability (which is sometimes referred to as call congestion). Time congestion merely specifies the probability that all servers are busy. Before blocking can occur, there must be an arrival.

In an infinite source system, time congestion and call congestion are identical because the percentage of arrivals encountering all servers busy is exactly equal to the time congestion. (The fact that all servers are busy has no bearing on whether or not an arrival occurs.) In a finite source system, however, the percentage of arrivals encountering congestion is smaller because fewer arrivals occur during periods when all servers are busy. Thus in a finite source system, call congestion (blocking probability) is always less than the time congestion. As an extreme example, consider equal numbers of sources and servers. The time congestion is the probability that all servers are busy. The blocking probability is obviously zero.

The same basic techniques introduced by Erlang when he determined the loss formula for infinite sources can be used to derive loss formulas for finite sources [3], Using these techniques, we find the probability of *n* servers being busy in a system with *M* sources and *N* servers is

$$P_N = \frac{\binom{M}{N}(\lambda' t_{\mathrm{m}})^N}{\Sigma_{i=0}^{N}\binom{M}{i}(\lambda' t_{\mathrm{m}})^i}$$

where *V* is the calling rate per *idle* source and t_m is the average holding time. Equation is known as the truncated Bemoullian distribution and also as the Engset distribution.

$$B = \frac{\binom{M-1}{N}(\lambda' t_{\mathrm{m}})^N}{\Sigma_{i=0}^{N}\binom{M-1}{i}(\lambda' t_{\mathrm{m}})^i}$$

Setting *n* = *Nin* Equation 12.11 produces an expression for the time congestion:

$$\lambda' t_{\mathrm{m}} = \frac{\rho}{1-\rho(1-B)}$$

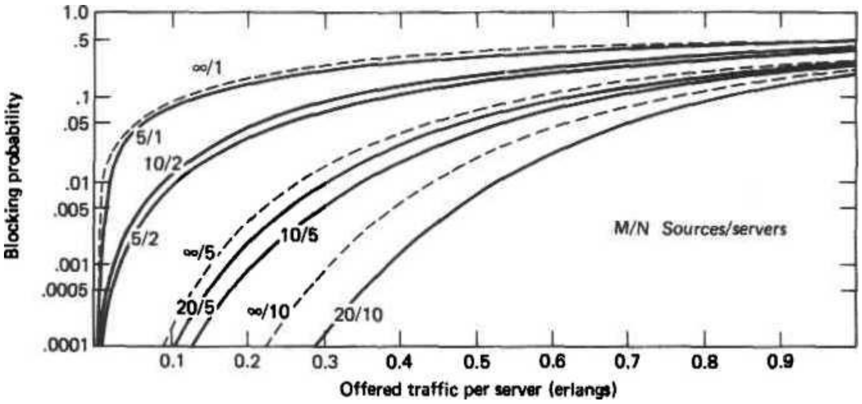
Using the fact that the arrival rate when *N* servers are busy is (M - *N)/M* times the arrival rate when no servers are busy, we can determine the blocking probability for lost calls cleared with a finite source as follows:

which is identical to P_N (the time congestion) for M - 1 sources.

Equations are easily evaluated in terms of the parameters A,' and t_m. However, X' and t_m do not, by themselves, specify the average activity of a source. In a lost calls cleared system with finite sources the effective offered load decreases as the blocking probability increases because blocked calls leave and do not return. When a call is blocked, the average activity of the offering source decreases, which increases the average amount of idle time for that source. The net result is that X' decreases because the amount of idle time increases. If the average activity of a source assuming no traffic is cleared is designated as p = fa_m, the value of $X't_m$ can be determined as where B is the blocking probability defined by Equation 12.13.

The difficulty with using the unblocked source activity factor p to characterize a source's offered load is now apparent. The value of X' $\mathrm{f_m}$ depends on B, which in turn depends on X' t_m. Thus some form of iteration is needed to determine B when the sources are characterized by p (an easily measured parameter) instead of X'. If the total offered load is considered to be Mp, the carried traffic is

$$A_{\text{carried}} = M\rho(1 - B)$$



Blocking probability of lost calls cleared with finite sources.

A table of traffic capacities for finite sources is provided in Appendix D.2, where the offered load $A = Mp$ is listed for various combinations of M, N, and B. Some of the results are plotted in Figure 12.10, where they can be compared to blocking probabilities of infinite source systems. As expected, infinite source analyses (Erlang-B) are acceptable when the number of sources *M* is large.

sources are used, the offered load of 7.26 erlangs is higher than the 7.04 erlangs obtainable from interpolation in Table D.2 as the maximum offered load for *B* = 1%.

It is worthwhile comparing the result of Example 12.10 to a result obtained from an infinite source analysis (Erlang-fl). For a blocking probability of 1 %, Table D. 1 reveals that the maximum offered load for 12 servers is 5.88 erlangs. Thus the maximum number of sources can be determined as 5.88/0.333 = 17.64. Hence in this case an infinite source analysis produces a result that is conservative by 15%.

12.2.2 Lost Calls Held—Finite Sources

A lost calls held system with finite sources is analyzed in the same basic manner as a lost calls held systems with infinite sources. At all times the number of calls "in the system" is defined to be identical to the number of calls that would be serviced by a strictly nonblocking server group. Thus Equation 12.11 is used to determine the probability that exactly *n* calls are in the system:

$$P_n = \frac{\binom{M}{n}(\lambda' t_{\mathrm{m}})^n}{\sum_{i=0}^{M} \binom{M}{i}(\lambda' t_{\mathrm{m}})^i}$$

$$= \frac{\binom{M}{n}(\lambda' t_{\mathrm{m}})^n}{(1 + \lambda' t_{\mathrm{m}})^M}$$

Because no calls are cleared, the offered load per idle source is not dependent

$$\lambda' t_{\mathrm{m}} = \frac{\lambda t_{\mathrm{m}}}{1 - \lambda t_{\mathrm{m}}} = \frac{\rho}{1 - \rho}$$

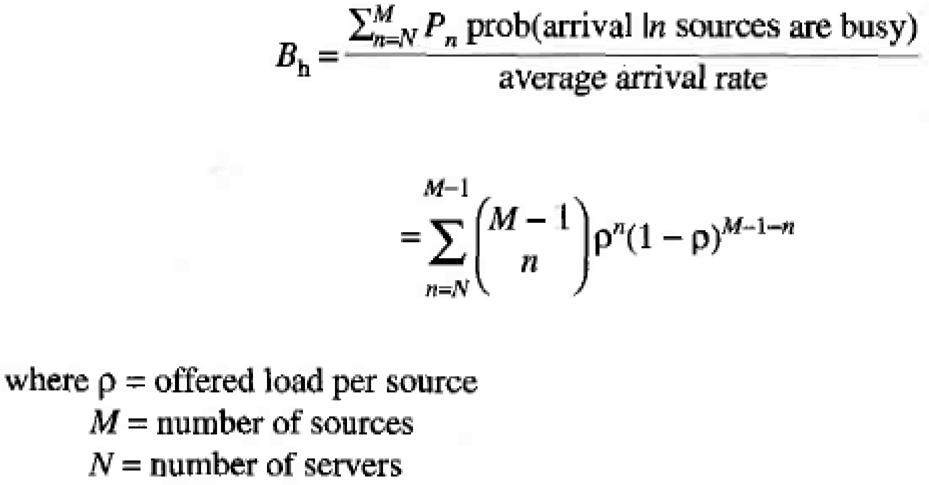
$$P_n = \binom{M}{n} \rho^n (1 - \rho)^{M-n}$$

Combining Equations 12.16 and 12.17 produces a more useful expression for the probability that n calls are in the system:

$$P_N = \sum_{n=N}^{M} P_n$$

If there are N servers, the time congestion is merely the probability that N or more servers are busy:

The blocking probability, in a lost calls held sense, is the probability of an arrival encountering AT or more calls in the system:



$$B_{\mathrm{h}} = \frac{\sum_{n=N}^{M} P_n \text{ prob(arrival } |n \text{ sources are busy)}}{\text{average arrival rate}}$$

$$= \sum_{n=N}^{M-1} \binom{M-1}{n} \rho^n (1-\rho)^{M-1-n}$$

where ρ = offered load per source
M = number of sources
N = number of servers

where p = offered load per source M - number of sources N = number of servers

NETWORK BLOCKING PROBABILITIES

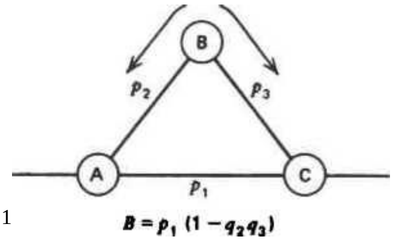
In the preceding sections basic techniques of congestion theory are presented to determine blocking probabilities of individual trunk groups. In this section techniques of calculating end-to-end blocking probabilities of a network with more than one routebetween endpoints is considered. In conjunction with calculating the end-to-end blocking probabilities, it is necessary to consider the interaction of traffic on various routes of a network. Foremost among these considerations is the effect of overflow traffic from one route onto another. The following sections discuss simplified analyses only. More sophisticated techniques for more complex networks can be obtained in references [9],

End-to-End Blocking Probabilities

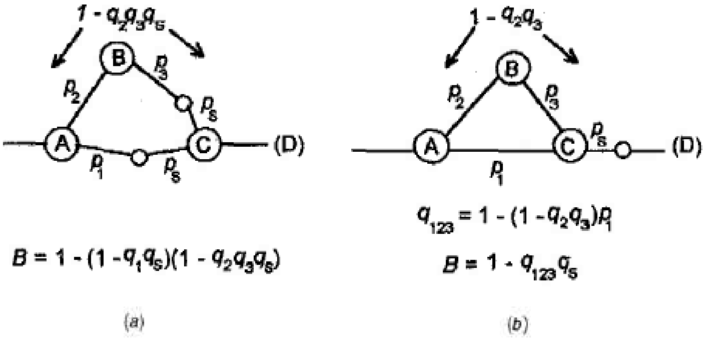
Generally, a connection through a large network involves a series of transmission links, each one of which is selected from a set of alternatives. Thus an end-to-end blocking probability analysis usually involves a composite of series and parallel probabilities. The simplest procedure is identical to the blocking probability (matching loss) analyses for switching networks. For example, Figure depicts a representative set of alternative connections through a network and the resulting composite blocking probability.

The blocking probability equation in Figure 12.12 contains several simplifying assumptions. First, the blocking probability (matching loss) of the switches is not included. In a digital time division switch, matching loss can be low enough that it is easily eliminated from the analysis. In other switches, however, the matching loss may not be insignificant. When necessary, switch blocking is included in the analysis by considering it a source of blocking in series with the associated trunk groups.

When more than one route passes through the same switch, as in node C of Figure 12.12, proper treatment of correlation between matching losses is an additional complication. A conservative approach considers the matching loss to be completely correlated. In this case the



1



Incorporating switch-matching loss into end-to-end blocking analysis: ***(a)*** independent switch blocking; (***b***) correlated switch blocking.

matching loss is in series with the common link. On the other hand, an optimistic analysis assumes that the matching losses are independent, which implies that they are in series with the individual links. Figure 12.13 depicts these two approaches for including the matching loss of switch C into the end-to-end blocking probability equation of Figure 12.12. In this case, the link from C to D is the common link.

A second simplifying assumption used in deriving the blocking probability equation in Figure 12.12 involves assuming independence for the blocking probabilities of the trunk groups. Thus the composite blocking of two parallel routes is merely the product of the respective probabilities (Equation 5.4). Similarly, independence implies that the blocking probability of two paths—in series—is 1 minus the product of the respective availabilities (Equation 5.5). In actual practice individual blocking probabilities are never completely independent. This is particularly true when a large amount of traffic on one route results as overflow from another route. Whenever the first route is busy, it is likely that more than the average amount of overflow is being diverted to the second route. Thus an alternate route is more likely to be busy when a primary route is busy.

In a large public network, trunks to tandem or toll switches normally carry traffic to many destinations. Thus no one direct route contributes an overwhelming amount of overflow traffic to a particular trunk group. In this case independent blocking probabilities on alternate routes are justified. In some instances of the public network, and often in private networks, overflow traffic from one route dominates the traffic on tandem routes. In these cases failure to account for the correlation in blocking probabilities can lead to overly optimistic results.

The correlations between the blocking probabilities of individual routes arise because congestion on one route produces overflows that tend to cause congestion on other routes. External events stimulating networkwide overloads also cause the blocking probabilities to be correlated. Thus a third assumption in the end-to-end blocking probability equation of Figure 12.12 is that traffic throughout the network is independent. If fluctuations in the traffic volume on individual links tend to be correlated (presumably because of external events such as television commercials, etc.), significant degradation in overall performance results.

12.3.2 Overflow Traffic

The second source of error in Example 12.13 occurred because an Erlang-/} analysis used the average volume of overflow traffic from the first group to determine the blocking probability of the second trunk group. An Erlang-B analysis assumes traffic arrivals are purely random, that is, they are modeled by a Poisson distribution. However, a Poisson arrival distribution is an erroneous assumption for the traffic offered to the second trunk group. Even though arrivals to the first group may be random, the overflow process tends to select groups of these arrivals and pass them on to the second trunk group. Thus instead of being random the arrivals to the second group occur in bursts. This overflow effect is illustrated in Figure 12.14, which portrays a typical random arrival pattern to one trunk group and the overflow pattern to a second group. If a significant amount of the traffic flowing onto a trunk group results as overflow from other trunk groups, overly optimistic values of blocking probability arise when all of the traffic is assumed to be purely random.

The most common technique of dealing with overflow traffic is to relate the overflow traffic volume to an equivalent amount of random traffic in a blocking probability sense. For example, if the 1.62 erlangs of overflow traffic in Example 12.12 is equated to 2.04 erlangs of random traffic, a blocking probability of 1.3% is obtained for the second trunk group. (This is the correct probability of blocking for the second group since both groups are busy if and only if the second group is busy.)

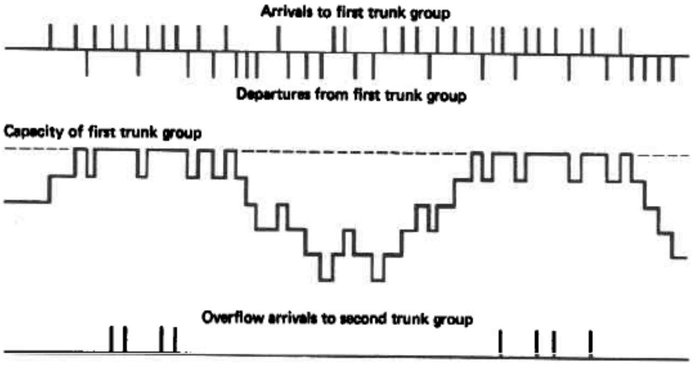

Arrivals to first trunk group
Departures from first trunk group
Capacity of first trunk group
Overflow arrivals to second trunk group

Figure 12.14 Bursty characteristic of overflow traffic.

This method of treating overflow traffic is referred to as the equivalent random theory [12]. Tables of traffic capacity are available [13] that incorporate the overflow effects directly into the maximum offered loads. The Neal-Wilkinson tables used by Bell System traffic engineers comprise one such set of tables. The Neal-Wilkinson tables, however, also incorporate the effects of day-to-day variations in the traffic load. Arrival* to first trunk group

(Forty erlangs on one day and 30 erlangs on another is not the same as 35 erlangs on both days.) These tables are also used for trunk groups that neither generate nor receive overflow traffic. The fact that cleared traffic does not get serviced by an alternate route implies that retries are likely. The effect of the retries, however, is effectively incorporated into the value of *B* by equivalent randomness.

DELAY SYSTEMS

The second category of teletraffic analysis concerns systems that delay nonserviceable requests until the necessary facilities become available. These systems are variously referred to as delay systems, waiting-call systems, and queuing systems. Call arrivals occurring when all servers are busy are placed in a queue and held until service commences. The queue might consist of storage facilities in a physical sense, such as blocks of memory in a message-switching node, or the queue might consist only of a list of sources waiting for service. In the latter case, storage of the messages is the responsibility of the sources themselves.

Using the more general term *queueing theory*, we can apply the following analyses to a wide variety of applications outside of telecommunications. Some of the more common applications are data processing, supermarket check-out counters, aircraft landings, inventory control, and various forms of service bureaus. These and many other applications are considered in the field of operations research. The foundations of queuing theory, however, rest on fundamental techniques developed by early telecommunications traffic researchers. In fact, Erlang is credited with the first solution to the most basic type of delay system. Examples of delay system analysis applications in telecommunications are message switching, packet switching, statistical time division multiplexing, multipoint data communications, automatic call distribution, digit receiver access, signaling equipment usage, and call processing. Furthermore, many PBXs have features allowing queued access to corporate tie lines or WATS lines. Thus some systems formerly operating as loss systems now operate as delay systems.

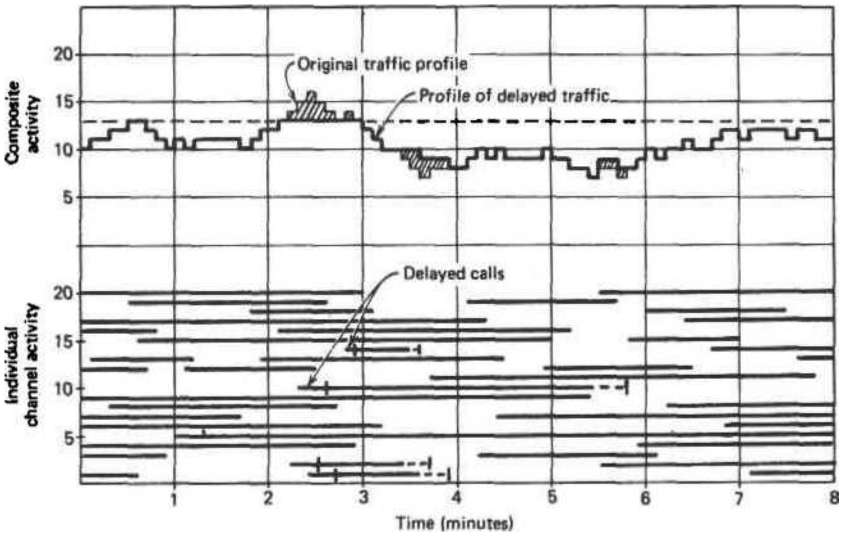
In general, a delay operation allows for greater utilization of servers (transmission facilities) than does a loss system. Basically, the improved utilization is achieved because peaks in the arrival process are “smoothed” by the queue. Even though arrivals to the system are random, the servers see a somewhat regular arrival pattern. The effect of the queuing process on overload traffic is illustrated in Figure 12.15. This figure displays the same traffic patterns presented earlier in Figures 12.1,12.3, and 12.9. In this case, however, overload traffic is delayed until call terminations produce available channels.

In most of the following analyses it is assumed that all traffic offered to the system eventually gets serviced. One implication of this assumption is that the offered traffic intensity *A* is less than the number of servers *N*. Even when *A* is less than *N*, there are two cases in which the carried traffic might be less than the offered traffic. First, some sources might tire of waiting in a long queue and abandon the request. Second, the capacity for storing requests may be finite. Hence requests may occasionally be rejected by the system.

A second assumption in the following analyses is that infinite sources exist. In a delay system, there may be a finite number of sources in a physical sense but an infinite number of sources in an operational sense because each source may have an arbitrary number of requests outstanding (e.g., a packet-switching node). There are instances in which a finite source analysis is necessary, but not in the applications considered here.

An additional implication of servicing all offered traffic arises when infinite sources exist. This implication is the need for infinite queuing capabilities. Even though the offered traffic intensity is less than the number of servers, no statistical limit exists on the number of arrivals occurring in a short period of time. Thus the queue of a purely lossless system must be arbitrarily long. In a practical sense, only finite queues can be realized, so either a statistical chance of blocking is always present or all sources can be busy and not offer additional traffic.

When analyzing delay systems, it is convenient to separate the total time that a request is in the system into the waiting time and the holding time. In delay systems analysis the holding time is more commonly referred to as the service time. In contrast to loss systems, delay system performance is generally dependent on the distribution of service times and not just the mean value t_m. Two service time distributions are considered here: constant service times and exponential service times. Respectively, these distributions represent the most deterministic and the most random service times possible. Thus a system that operates with some other distribution of service times performs somewhere between the performance produced by these two distributions.



The basic purpose of the following analyses is to determine the probability distribution of waiting times. From the distribution, the average waiting time is easily determined. Sometimes only the average waiting time is of interest. More generally, however, the probability that the waiting time exceeds some specified value is of interest. In either case, the waiting times are dependent on the following factors:

1. Intensity and probabilistic nature of the offered traffic
2. Distribution of service times
3. Number of servers
4. Number of sources
5. Service discipline of the queue

The service discipline of the queue can involve a number of factors. The first of these concerns the manner in which waiting calls are selected. Commonly, waiting calls are selected on a first-come, first-served (FCFS) basis, which is also referred to as first-in, first-out (FIFO) service. Sometimes, however, the server system itself does not maintain a queue but merely polls its sources in a round-robin fashion to determine which ones are waiting for service. Thus the queue may be serviced in sequential order of the waiting sources. In some applications waiting requests may even be selected at random. Furthermore, additional service variations arise if any of these schemes are augmented with a priority discipline that allows some calls to move ahead of others in the queue.

A second aspect of the service discipline that must be considered is the length of the queue. If the maximum queue size is smaller than the effective number of sources, blocking can occur in a lost calls sense. The result is that two characteristics of the grade of service must be considered: the delay probability and the blocking probability. A common example of a system with both delay and loss characteristics is an automatic call distributor with more access circuits than attendants (operators or re- servationists). Normally, incoming calls are queued for service. Under heavy loads, however, blocking occurs before the ACD is even reached. Reference [14] contains an analysis of a delay system with finite queues and finite servers.

To simplify the characterization of particular systems, queuing theorists have adopted a concise notation for classifying various types of delay systems. This notation, which was introduced by D. G. Kendall, uses letter abbreviations to identify alternatives in each of the categories listed. Although the discussions in this book do not rely on this notation, it is introduced and used occasionally so the reader can relate the following discussions to classical queuing theory models. The interpretation of each letter is specified in Figure 12.16.

The specification format presented in Figure 12.16 actually represents an extension of the format commonly used by most queuing theorists. Thus this format is sometimes abbreviated by eliminating the last one or two entries. When these entries are eliminated, infinite case specifications are assumed. For example, a single-server system with random input and negative exponential service times is usually specified as M/M/l. Both the number of sources and the permissible queue length are assumed infinite.

Exponential Service Times

The simplest delay system to analyze is a system with random arrivals and negative exponential service times: M/M/N. Recall that a random arrival distribution is one with negative exponential interarrival times. Thus in the shorthand notation of queuing theorists, the letter M always refers to negative exponential distributions (an M is used because a purely random distribution is memoryless)

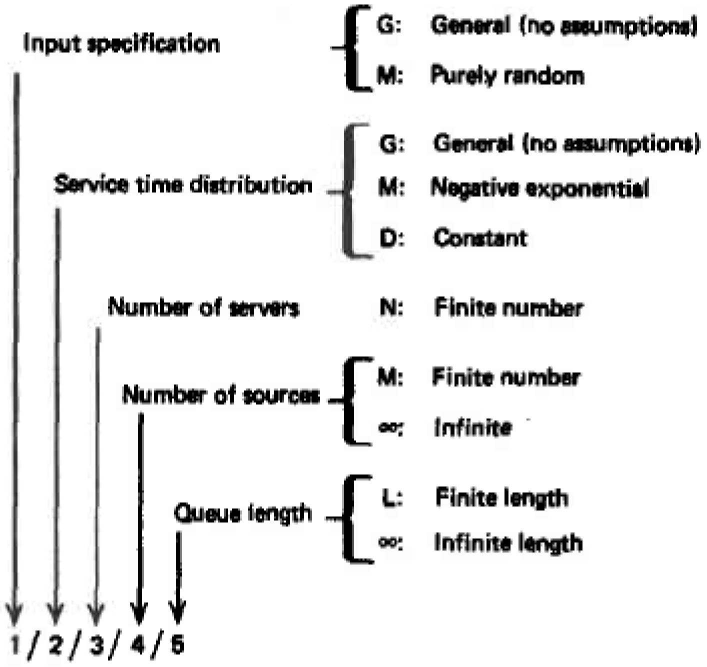


Figure 12.16 Queueing system notation.

In the M/M/1 system and all other systems considered here, it is assumed that calls are serviced in the order of their arrival. The following analyses also assume that the probability of an arrival is independent of the number of requests already in the queue (infinite sources). From these assumptions, the probability that a call experiences congestion and is therefore delayed was derived by Erlang:

The probability of delay *p(>0)* is variously referred to as Erlang's second formula, ^2jv0^)> Erlang s delay formula; or the Erlang-C formula. For single-server systems *(N* = 1) the probability of delay reduces to p, which is simply the output utilization or traffic earned by the server. Thus the probability of delay for a single-server system is also equal to the offered load Xt_m (assuming Xt_m< 1).

The distribution of waiting times for random arrivals, random service times, and a FIFO service discipline is

$P(>t) = p(> 0)e^{HN\sim A)t/,}m$ (12.25)

where $p(>0)$ = probability of delay given in Equation 12.24

t_m = average service time of negative exponential service time distribution

Equation 12.25 defines the probability that a call arriving at a randomly chosen instant is delayed for more than t/t_m service times. Figure 12.17 presents the relationship of Equation 12.25 by displaying the traffic capacities of various numbers of servers as a function

$$\text{prob(delay)} = p(>0) = \frac{NB}{N - A(1 - B)}$$

where N = number of servers

A = offered load (erlangs)

B = blocking probability for a lost calls cleared system

Constant Service Times

This section considers delay systems with random arrivals, constant service times, and a single server (M/D/1). Again, FIFO service disciplines and infinite sources are as-sumed. The case for multiple servers has been solved [3] but is too involved to include here. Graphs of multiple-server systems with constant service times are available in reference [15].

The average waiting time for a single server with constant service times is determined as

$$\bar{t} = \frac{\rho t_m}{2(1 - \rho)}$$

where p =*A* is theserver utilization. Notice thatEquation 12.28producesanaverage waitingtimethatisexactlyone-halfofthatforasingle-server system withexponential servicetimes.Exponentialservicetimescausegreater averagedelaysbecause there aretworandom processes involved in creatingthedelay.Inbothtypesof systems,delays occur when a large burst of arrivals exceeds the capacity of the servers. With exponential service times, however, long delays also arise because of excessive service times of just a few arrivals. (Recall that this aspect of conventional message-switching systems is one of the motivations for breaking messages up into packets in a packet- switching network.)

If the activity profile of a constant service time system (M/D/1) is compared with the activity profile of an exponential service time system (M/M/1), the M/D/1 system is seen to be active for shorter and more frequent periods of time. That is, the M/M/1 system has a higher variance in the duration of its busy periods. The average activity of both systems is, of course, equal to the server utilization p. Hence the probability of delay for a single-server system with constant service times is identical to that for exponential service times: $p(>0) = Xt_m$.

$$
\begin{aligned}
p(>t) &= p[>(k+r)t_m] \\
&= 1-(1-\rho)\sum_{i=0}^{k} \frac{\rho^i (i - t/t_m)^i e^{-\rho(i-t/t_m)}}{i!} \\
&= 1-(1-\rho)e^{\lambda t}\sum_{i=0}^{k} \frac{(i\rho - \lambda t)^i e^{-i\rho}}{i!}
\end{aligned}
$$

where k = largest integral quotient of t/t_m
r = remainder of t/t_m
ρ = server utilization, $= \lambda t_m$

The probability of congestion for larger *N* is relatively close to that for exponential service times. Thus Equation 12,25 can be used as a close approximation for*p(*>0) for multiple-server systems with arbitrary service time distributions.

For single-server systems with constant holding times, the probability of delay greater than an arbitrary value *t* is

Finite Queues

All of the delay system analyses presented so far have assumed that an arbitrarily large number of delayed requests could be placed in a queue. In many applications this assumption is invalid. Examples of systems that sometimes have significantly limited queue sizes are store-and-forward switching nodes (e.g., packet switches and ATM switches), automatic call distributors, and various types of computer input/output devices. These systems treat arrivals in three different ways, depending on the number "in the system" at the time of an arrival:

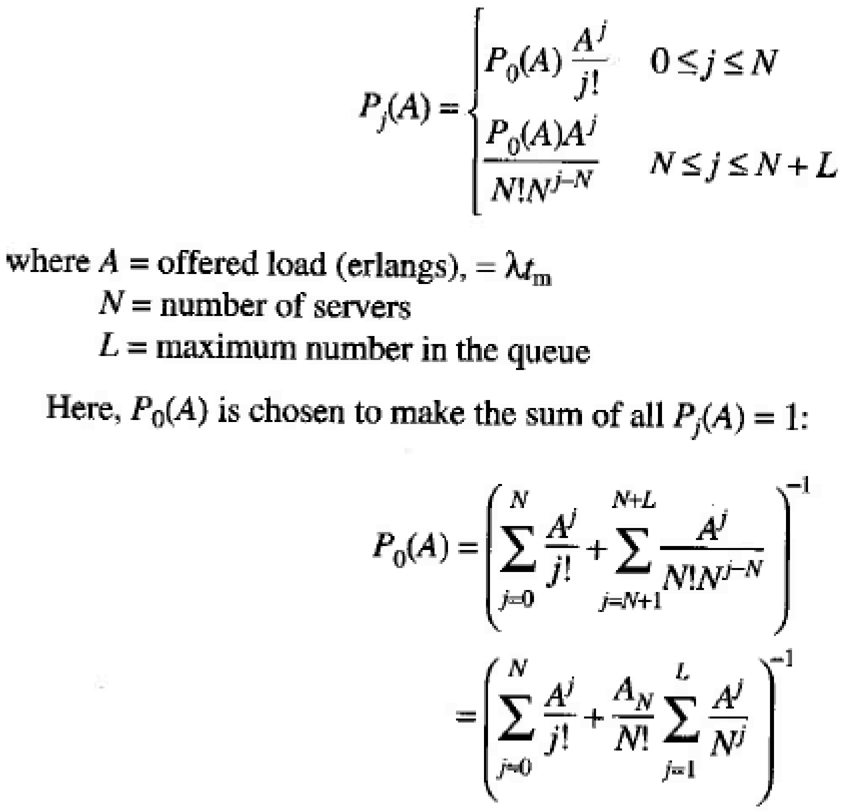
Immediate service if one or more of *N* servers are idle

0. Delayed service if all servers are busy and less than *L* requests are waiting
1. Blocked or no service if the queue of length *L* is full

In finite-queue systems the arrivals getting blocked are those that would otherwise experience long delays in a pure delay system. Thus an indication of the blocking probability of a combined delay and loss system can be determined from the probability that arrivals in pure delay systems experience delays in excess of some specified value. However, there are two basic inaccuracies in such an analysis. First, the effect of blocked or lost calls cleared is to reduce congestion for a period of time and thereby to reduce the delay probabilities for subsequent arrivals. Second, delay times do not necessarily indicate how many calls are "in the system." Normally, queue lengths and blocking probabilities are determined in terms of the number of waiting requests, not the amount of work or total service time represented by the requests. With constant service times, there is no ambiguity between the size of a queue and its implied delay. With exponential service times, however, a given size can represent a wide range of delay times.

A packet-switching node is an example of a system in which the queue length is most appropriately determined by implied service time and not by the number of pending requests. That is, the maximum queue length may be determined by the amount of store-and-forward memory available for variable-length messages and not by some fixed number of messages.

For a system with random input, exponential service times, *N* servers, an infinite source, and a maximum queue length of *L* (M/M/N/«/£), the probability of *j* calls in the system is



$$P_j(A) = \begin{cases} P_0(A)\dfrac{A^j}{j!} & 0 \le j \le N \\ \dfrac{P_0(A)A^j}{N!N^{j-N}} & N \le j \le N+L \end{cases}$$

where A = offered load (erlangs), = λt_{m}
N = number of servers
L = maximum number in the queue

Here, $P_0(A)$ is chosen to make the sum of all $P_j(A) = 1$:

$$P_0(A) = \left(\sum_{j=0}^{N} \frac{A^j}{j!} + \sum_{j=N+1}^{N+L} \frac{A^j}{N!N^{j-N}} \right)^{-1}$$

$$= \left(\sum_{j=0}^{N} \frac{A^j}{j!} + \frac{A_N}{N!} \sum_{j=1}^{L} \frac{A^j}{N^j} \right)^{-1}$$

It is worth noting that if there is no queue (L = 0), these equations reduce to those of the Erlang loss equation (12.8). If *L* is infinite, Equation 12.31 reduces to Erlang's delay formula, Equation 12.24. Thus these equations represent a general formulation that produces the pure loss and pure delay formulas as special cases.

The waiting time distribution [3] is

$$P(>0) = \sum_{j=N}^{N+L} P_j(A) = P_N(A)\,\frac{1-\rho^{L+1}}{1-\rho}$$

where $\rho = A/N$ is the offered load per server.

The loss, or blocking probability, is determined as

$$B = P_{N+L}(A) = \frac{P_0(A)A^{N+L}}{N!N^L}$$

$$p(>t) = P_N(A)\sum_{j=0}^{L-1}\frac{\rho^j}{j!}\int_{Nt/t_m}^{\infty} x^j e^{-x}dx$$

from which the average delay can be determined as

$$\bar{t} = \frac{[p(>0) - P_{N+L}(A)]t_m}{N-A}$$

»). ***Single-Server Equations***

Because most queuing applications involve single-server configurations, the previous equations are listed explicitly for N = 1:

, Prob(y calls in system) (12.30):

$$P_j(\rho) = P_0(\rho)\rho^j$$

$$P_0(\rho) = \left(1 + \rho + \rho \sum_{j=1}^{L} \rho^j\right)^{-1} = \left(\sum_{j=0}^{L+1} \rho^j\right)^{-1} = \frac{1-\rho}{1-\rho^{L+2}}$$

Probability of delay (12.31):

$$p(>0) = \frac{P_0(\rho)\rho(1-\rho^{L+1})}{1-\rho} = \frac{\rho(1-\rho^{L+1})}{1-\rho^{L+2}}$$

Probability of loss (12.32):

$$B = \frac{(1-\rho)\rho^{L+1}}{1-\rho^{L+2}}$$

Average delay (12.34):

$$\bar{t} = \frac{[p(>0) - P_{L+1}(\rho)]t_{\mathrm{m}}}{1-\rho} = \frac{\rho(1-\rho^L)t_{\mathrm{m}}}{(1-\rho)(1-\rho^{L+2})}$$

The blocking probability of a single-server system *(N* = 1) is plotted in Figure When using Figure 12.19, keep in mind that the blocking probability is determined by the number of waiting calls and not by the associated service time. Furthermore, since the curves of Figure 12.19 are based on exponential service times, they overestimate the blocking probabilities of constant holding time systems (e.g., fixed-length packet networks). However, if fixed-length packets arise primarily from longer, exponentially distributed messages, the arrivals are no longer independent, and the use of Figure 12.19 (or Equation 12.38) as a conservative analysis is more appropriate.

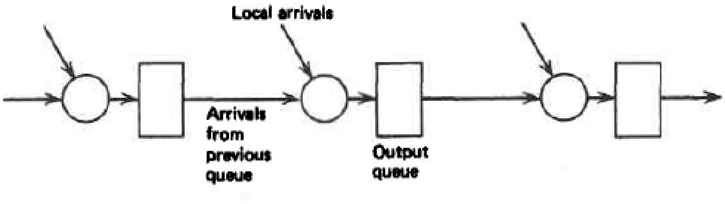
ATM Cell Queues

Analysis of queuing delays and cell loss in an ATM switching node is complicated. The cells have a fixed length of 53 bytes so it would seem that a constant service time analysis would be appropriate. This assumption is valid for voice traffic inserted onto wide-bandwidth signals such as 155-Mbps STS-ls. In this case the service time is much shorter that the duration of a speech burst (e.g., 2.7 (isec versus several tens of milliseconds). Even though correlated arrivals occur from individual sources, the arrival times are separated by many thousands of service times so they appear independent.

When ATM voice is carried in CBR trunk groups, a different situation results. In this case the service times of the voice cells may be only slightly smaller than the interval between voice cell generation, and the average delay would indicate that two or more cells from the same source could be present in the queue at one time. Thus, a queuing analysis that assumes exponentially distributed service times is more appropriate even though the variable-length talk spurts are broken up into fixed-length cells.

Tandem Queues

All of the equations provided in previous sections for delay system analysis have dealt with the performance of a single queue. In many applications a service



Tandem queues

request undergoes several stages of processing, each one of which involves queuing. Thus it is often desirable to analyze the performance of a system with a number of queues in series. Figure 12.20 depicts a series of queues that receive, as inputs, locally generated requests and outputs from other queues. Two principal examples of applications with tandem queues are data processing systems and store-and-forward switching networks.

Researchers in queuing theory have not been generally successful in deriving formulas for the performance of tandem queues. Often, simulation is used to analyze a complex arrangement of interdependent queues arising in systems like store-and-for- ward networks. Simulation has the advantage that special aspects of a network's operation—like routing and flow control—can be included in the simulation model. The main disadvantages of simulation are expense and, often, less visibility into the dependence of system performance on various design parameters.

One tandem queuing problem that has been solved [19] is for random inputs and random (negative exponential) holding times for all queues. The solution of this system is based on the following theorem: In a delay system with purely random arrivals and negative exponential holding times, the instants at which calls terminate is also a negative exponential distribution.

The significance of this theorem is that outputs from an M/M/N system have statistical properties that are identical to its inputs. Thus a queuing process in one stage does not affect the arrival process in a subsequent stage, and all queues can be analyzed independently. Specifically, if a delay system with *N* servers has exponentially distributed.