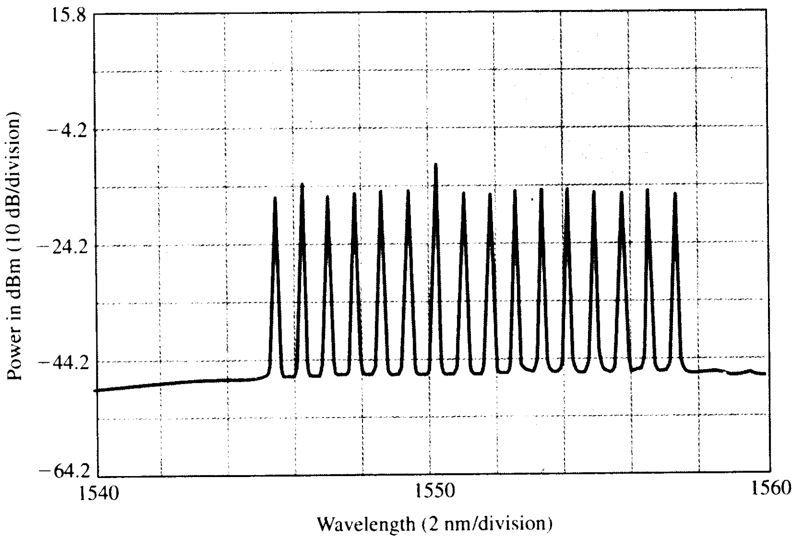


FIGURE 4.8 Optical signal in a WDM system.

bit rate of 320 Gbps and are widely deployed. Systems that can carry 160 wavelengths at 10 Gbps are also available and achieve an amazing bit rate of 1.6 terabits/second. The attraction of WDM is that a huge increase in available bandwidth is obtained without the huge investment associated with deploying additional optical fiber.[3] The additional bandwidth can be used to carry more traffic and can also provide the additional protection bandwidth required by self-healing network topologies.

Early WDM systems differ in substantial ways from electronic FDM systems. In FDM the channel slots are separated by guard bands that are narrow relative to the bandwidth of each channel slot. These narrow guard bands are possible because the devices for carrying out the required modulation, filtering, and demodulation are available. Narrow spacing is not the case for WDM systems. Consequently, the spacing between wavelengths in WDM systems tends to be large compared to the bandwidth of the information carried by each wavelength. For example, "coarse" WDM systems combine wavelengths that are separated by 20 to 30 nm. Much more expensive, ITU-grid WDM systems have wavelength separations of 0.8 nm (which corresponds to 100 GHz) or even 0.4 nm.

4.2 SONET

In 1966 Charles Kao reported the feasibility of optical fibers that could be used for communications. By 1977 a DS3 45 Mbps fiber optic system was demonstrated in Chicago, Illinois. By 1998, 40 Gbps fiber optic transmission systems had become available. The advances in optical transmission technology have occurred at a rapid

[3]Deploying new fiber requires gaining right-of-way in streets and roads as well as digging trenches and other construction.

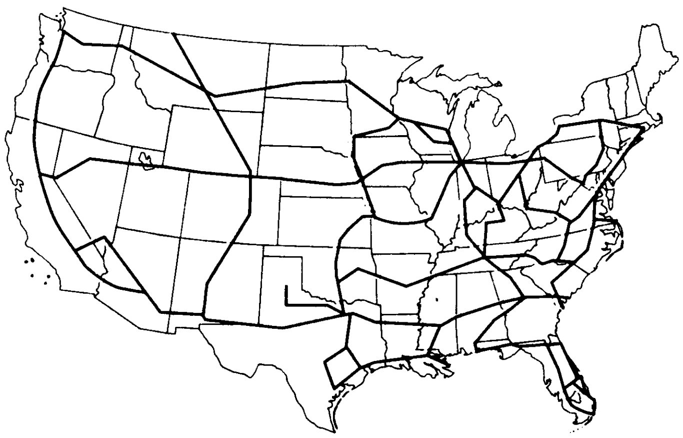


FIGURE 4.9 Optical fiber network for a long-distance telephone carrier in 1998.

rate, and the backbone of telephone networks has become dominated by fiber optic digital transmission systems. As an example Figure 4.9 shows the optical fiber network for a long-distance telephone carrier in 1998. Many large similar networks cover the globe to provide the backbone for today's networks.

The first generation of equipment for optical fiber transmission was proprietary, and no standards were available for the interconnection of equipment from different vendors. The deregulation of telecommunications in the United States led to a situation in which the long-distance carriers were expected to provide the interconnection between local telephone service providers. To meet the urgent need for standards to interconnect optical transmission systems, the **Synchronous Optical Network (SONET)** standard was developed in North America. The CCITT later developed a corresponding set of standards called **Synchronous Digital Hierarchy (SDH)**. Current backbone networks in North America are based on SONET, while in Europe and many other parts of the world they are based on SDH systems.

The SONET/SDH standards introduced several significant concepts for transmission networks. The "S" in SONET refers to a synchronous format that greatly simplifies the handling of lower-level digital signals and reduces the overall cost of multiplexing in the network. An additional feature of the SONET standards is the incorporation of overhead bytes in the frame structure for use in monitoring the signal quality, detecting faults, and signaling among SONET equipment to orchestrate rapid recovery from faults. In the remainder of this section we introduce SONET multiplexing and its frame structure. Section 4.3 considers how SONET equipment is deployed to form today's transport networks.

4.2.1 SONET Multiplexing

The SONET standard uses a 51.84 Mbps *electrical* signal, known as the **synchronous transport signal level-1 (STS-1)**, as a building block to extend the digital transmission

TABLE 4.1 SONET digital hierarchy.

SONET electrical signal	Optical signal	Bit rate (Mbps)	SDH electrical signal
STS-1	OC-1	51.84	
STS-3	OC-3	155.52	STM-1
STS-9	OC-9	466.56	STM-3
STS-12	OC-12	622.08	STM-4
STS-18	OC-18	933.12	STM-6
STS-24	OC-24	1244.16	STM-8
STS-36	OC-36	1866.24	STM-12
STS-48	OC-48	2488.32	STM-16
STS-192	OC-192	9953.28	STM-64

STS-synchronous transport signal; OC-optical channel; STM-synchronous transfer module.

hierarchy into the multigigabit/second range. A higher-level STS-*n* *electrical* signal in the hierarchy is obtained through the interleaving of bytes from the lower-level component signals. Each STS-*n* electrical signal has a corresponding **optical carrier level-*n* (OC-*n*)** signal that is obtained by modulating a laser source. The bit formats of STS-*n* and OC-*n* signals are the same except for the use of scrambling in the optical signal.[4] Table 4.1 shows the SONET and SDH digital hierarchy. Notice that the rate of an STS-*n* signal is simply *n* times the rate of an STS-1 signal.

The SDH standard refers to **synchronous transfer module-*n* (STM-*n*)** signals. The SDH STM-1 has a bit rate of 155.52 Mbps and is equivalent to the SONET STS-3 signal. The STS-1 signal accommodates the DS3 signal from the existing digital transmission hierarchy in North America. The STM-1 signal accommodates the CEPT-4 signal in the CCITT digital hierarchy. The STS-48/STM-16 signal is widely deployed in the backbone of modern communication networks.

SONET uses a frame structure that has the same 8 kHz repetition rate as traditional telephone TDM systems. SONET was designed to be very flexible in the types of traffic that it can handle. SONET uses the term *tributary* to refer to the component streams that are multiplexed together. Figure 4.10 shows how a SONET multiplexer can handle a wide range of tributary types. A slow-speed mapping function allows DS1, DS2, and CEPT-1 signals to be combined into an STS-1 signal. As indicated above a DS3 signal can be mapped into an STS-1 signal, and a CEPT-4 signal can be mapped into an STS-3 signal. Mappings have also been defined for mapping streams of packet information into SONET. Figure 4.10 shows that ATM streams can be mapped into an STS-3*c* signal, and that Packet-over-SONET (POS) allows packet streams also to be mapped into STS-3*c* signals.[5] A SONET multiplexer can then combine STS input signals into a higher-order STS-*n* signal. Details of the SONET frame structure and the mappings into STS signal formats are provided in Section 4.2.2.

---

[4]Scrambling maps long sequences of 1s or 0s into sequences that contain a more even balance of 1s and 0s to facilitate bit-timing recovery.

[5]ATM is introduced in Chapter 7 and discussed in detail in Chapter 9. Packet-over-SONET is discussed with PPP in Chapter 5.

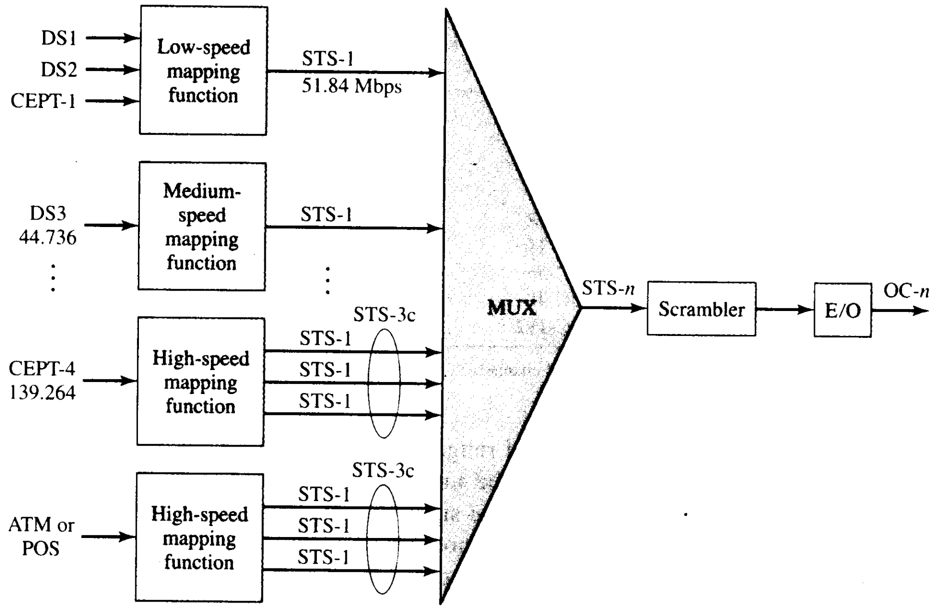


FIGURE 4.10 SONET multiplexing.

4.2.2 SONET Frame Structure

This section examines the SONET system and its frame structure. A SONET system is divided into three layers: sections, lines, and paths as shown in Figure 4.11a. A *section* refers to the span of fiber between two adjacent devices, such as two regenerators. The section layer deals with the transmission of an STS-n signal across the physical medium. A *line* refers to the span between two adjacent multiplexers and therefore in general encompasses several sections. Lines deal with the transport of an aggregate multiplexed stream and the associated overhead. A *path* refers to the span between the two SONET terminals at the endpoints of the system and in general encompasses one or more lines. User equipment, for example, large routers, can act as SONET terminals and be connected by SONET paths.

In general the bit rates of the SONET signals increase as we move from terminal equipment on to multiplexers deeper in the network. The reason is that a typical information flow begins at some bit rate at the edge of the network, which is then combined into higher-level aggregate flows inside the network, and finally delivered back at the original lower bit rate at the outside edge of the network. Thus the multiplexers associated with the path level typically handle signals lower in the hierarchy, for example, STS-1, than the multiplexers in the line level, for example, STS-3 or STS-48, as shown in Figure 4.11a.

Figure 4.11b shows that every section has an associated optical layer. The section layer deals with the signals in their electrical form, and the optical layer deals with the transmission of optical pulses. It can be seen that every regenerator involves converting the optical signal to electrical form to carry out the regeneration function and then back

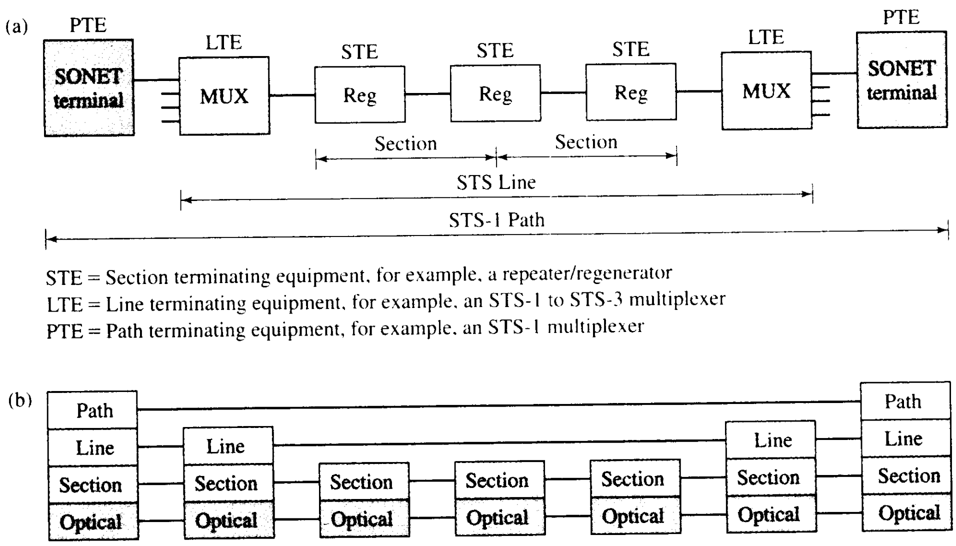


FIGURE 4.11 Section, line, and path layers of SONET.

to optical form. Note also in Figure 4.11b that all SONET equipment implement the optical and section functions. Line functions are found only in the multiplexers and end terminal equipment. The path function occurs only at the end terminal equipment.

Adjacent multiplexers exchange information using frames.[6] Figure 4.12 shows the structure of the SONET STS-1 frame that is defined *at the line level*. A frame consisting of a rectangular array of bytes arranged in 9 rows by 90 columns is repeated 8000 times a second.[7] Thus each byte in the array corresponds to a bit rate of 64 kbps, and the overall bit rate of the STS-1 is

$$8 \times 9 \times 90 \times 8000 = 51.84 \text{ Mbps} \tag{4.2}$$

The first three columns of the array are allocated to *section* and *line overhead*. The section overhead is interpreted and modified at every section termination and is used to provide framing, error monitoring, and other section-related management functions. For example, the first two bytes, A_1 and A_2, of the section overhead are used to indicate the beginning of a frame. The fourth byte B_1 carries parity checks of the transmitted signal and is used to monitor the bit error rate in a section. The last three bytes of the section overhead are used to provide a data communications channel between regenerators that can be used to exchange alarms, control, monitoring and other administrative messages.

The line overhead is interpreted and modified at every line termination and is used to provide synchronization and multiplexing, performance monitoring, line maintenance, as well as protection-switching capability in case of faults. We will see that the first

[6]The student is warned that the term *frame* has different meanings in SONET and in data link layer protocols.
[7]The bits are physically transmitted row by row and from left to right.

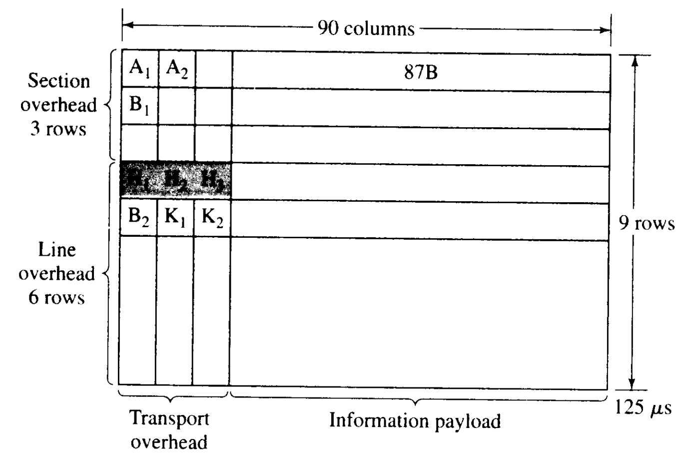


FIGURE 4.12 SONET STS-1 frame format.

three bytes of the line overhead, H_1, H_2, and H_3, play a crucial role in how multiplexing is carried out. The fourth byte B_2 is used to monitor the bit error rate in a line. The K_1 and K_2 bytes are used to trigger recovery procedures in case of faults. The remaining 87 columns of the frame constitute the *information payload* that carries the path layer or "user" information. The bit rate of the information payload is

$$8 \times 9 \times 87 \times 8000 = 50.122 \text{ Mbps} \tag{4.3}$$

The information payload includes one column of *path overhead* information, but the column is not necessarily aligned to the frame for reasons that will soon become apparent. The path overhead includes bytes for monitoring the performance of a path as well as for indicating the content and status of the end-to-end transfer.

Consider next how the end-to-end user information is organized *at the path level.* The SONET terminal equipment takes the user data and the path overhead and maps it into a **synchronous payload envelope (SPE)**, which consists of a byte array of nine rows by 87 columns, as shown in Figure 4.13. The path overhead uses the first column of this array. This SPE is then inserted into the STS-1 frame. The SPE is not necessarily aligned to the information payload of an STS-1 frame. Instead, the first two bytes of the line overhead are used as a **pointer** that indicates the byte within the information payload where the SPE begins. Consequently, the SPE can be spread over two consecutive frames as shown in Figure 4.13.[8] The use of the pointer makes it possible to extract a tributary signal from the multiplexed signal. This feature gives SONET multiplexers an *add-drop capability*, which means that individual tributaries can be dropped and individual tributaries can be added without having to demultiplex the entire signal.

---

[8]Imagine an STS-1 signal as a conveyor belt with 90 × 9-byte frames drawn on the belt. The SPEs are boxes of size 90 × 9 that are placed on the conveyor belt but that are not necessarily aligned to the frame boundary. This situation occurs because boxes are transferred between conveyor belts "on the fly."

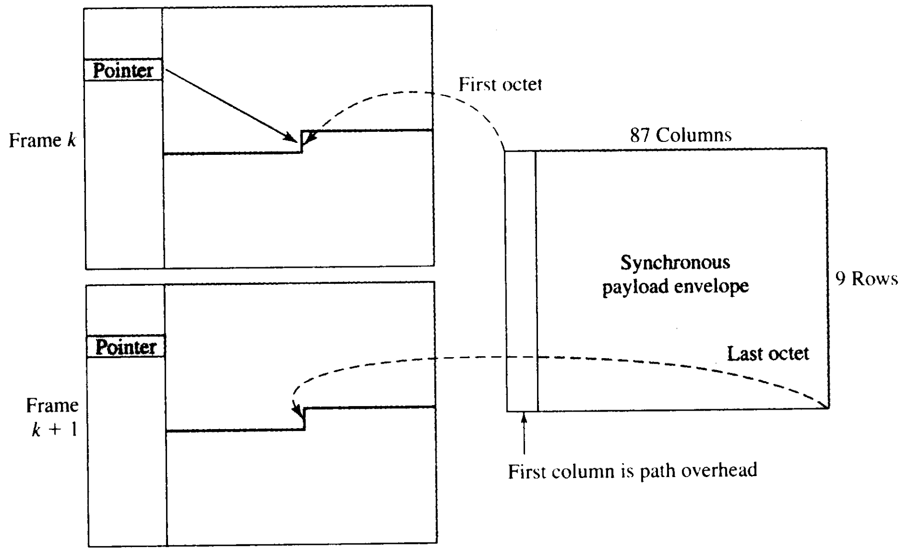


FIGURE 4.13 The synchronous payload envelope can span two consecutive frames.

The pointer structure consisting of the H_1, H_2, and H_3 bytes, shown in Figure 4.12 and Figure 4.13, maintains synchronization of frames and SPEs in situations where their clock frequencies differ slightly. If the payload stream is faster than the frame rate, then a buffer is required to hold payload bits as the frame stream falls behind the payload stream. To allow the frame to catch up, an extra SPE byte is transmitted in a frame from time to time. This extra byte, which is carried by H_3 within the line overhead as shown in Figure 4.14a, clears the backlog that has built up. Whenever this byte is inserted, the pointer is moved forward by one byte to indicate that the SPE starting point has been moved one byte forward. When the payload stream is slower than the frame stream, the number of SPE bytes transmitted in a frame needs to be reduced by one byte from time to time. This correction is done by stuffing an SPE byte with dummy information and then adjusting the pointer to indicate that the SPE now starts one byte later, as shown in Figure 4.14b.

Now consider how n STS-1 signals are multiplexed into an STS-n signal. Each incoming STS-1 signal is first synchronized to the local STS-1 clock of the multiplexer as follows. The section and line overhead of the incoming STS-1 signal are terminated, and its payload (SPE) is mapped into a *new* STS-1 frame that is synchronized to the local clock as shown in Figure 4.15. The pointer in the new STS-1 frame is adjusted as necessary, and the mapping is done on the fly. This procedure ensures that all the incoming STS-1 frames are mapped into STS-1 frames that are synchronized with respect to each other. The STS-n frame is produced by interleaving the bytes of the n synchronized STS-1 frames, in effect producing a frame that has nine rows, $3n$ section and line overhead columns, and $87n$ payload columns. To multiplex k STS-n signals into an STS-kn signal, the incoming signals are first de-interleaved into STS-1 signals and then the above procedure is applied to all kn STS-1 signals.

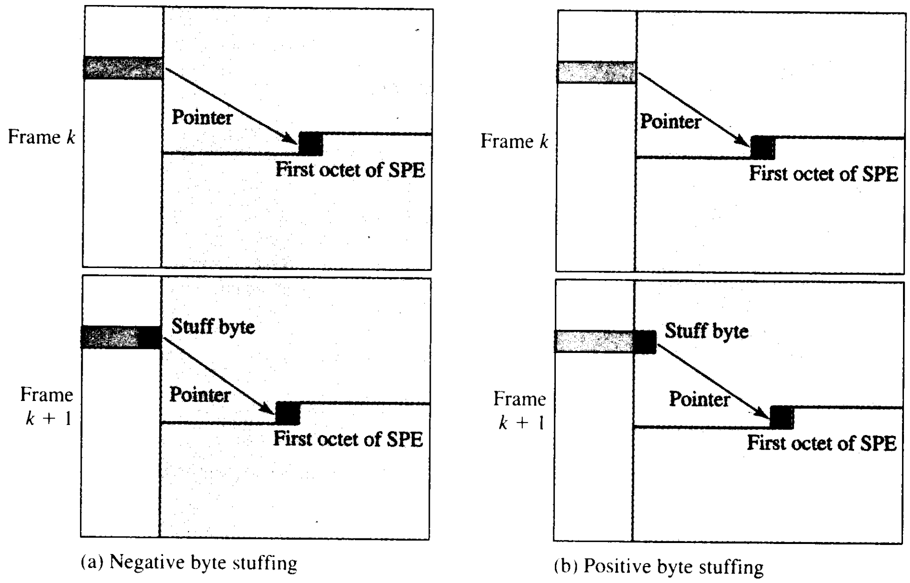


FIGURE 4.14 SPE byte stuffing into line overhead.

Various mappings have also been defined to combine lower-speed tributaries of various formats into standard SONET streams as shown in Figure 4.10. For example, a SONET STS-1 signal can be divided into *virtual tributary* signals that accommodate lower-bit-rate streams. In each SPE, 84 columns are set aside and divided into seven groups of 12 columns. Each group constitutes a virtual tributary and has a bit rate of $12 \times 9 \times 8 \times 8000 = 6.912$ Mbps. Alternatively, each virtual tributary can be viewed as $12 \times 9 = 108$ voice channels. Thus mappings have been developed so that a virtual tributary can accommodate four T-1 carrier signals ($4 \times 24 = 96 < 108$), or three CEPT-1 signals ($3 \times 32 = 96 < 108$). The SPE can then handle any mix of T-1 and CEPT-1 signals that can be accommodated in its virtual tributaries. In particular the SPE can handle a maximum of $7 \times 4 = 28$ T-1 carrier signals or $3 \times 7 = 21$ CEPT-1 signals. A mapping has also been developed so that a single SPE signal can handle one DS3 signal.

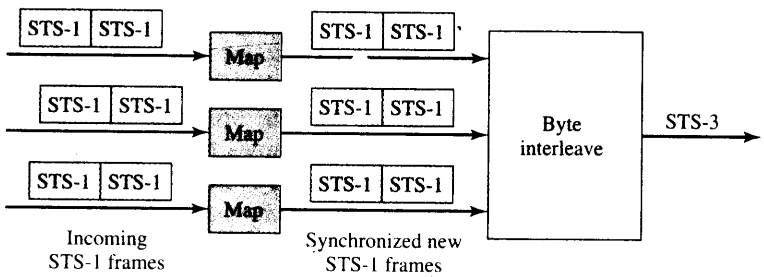


FIGURE 4.15 Synchronous multiplexing in SONET.

Several STS-1 frames can be concatenated to accommodate signals with bit rates that cannot be handled by a single STS-1. The suffix *c* is appended to the signal designation when *concatenation* is used to accommodate a signal that has a bit rate higher than STS-1. Thus an STS-3c signal is used to accommodate a CEPT-4 139.264 Mbps signal. Concatenated STS frames carry only one column of path overhead because they cannot be divided into finer granularity signals. For example, the SPE in an STS-3 frame has $86 \times 3 = 258$ columns of user data, whereas the SPE in an STS-3c frame carries $87 \times 3 - 1 = 260$ columns of user data. Mappings have also been developed so that an STS-3c frame can carry streams of ATM cells or streams of IP packets.

4.3 TRANSPORT NETWORKS

A **transport network** provides high bit rate connections to clients at different locations much like a telephone network provides voice circuits to users. The clients of a transport network can be large routers, large telephone switches, or even other networks as shown in Figure 4.16. The connections provided by a transport network can form the backbone of multiple, independent networks. For example, in Figure 4.16 the routers are interconnected to form an ISP backbone network, and the telephone switches are interconnected to form a telephone network.

A failure of a single connection in a transport network can be disastrous because each connection can carry so much traffic. For example, an OC-48 signal can carry up to 32,000 voice calls and can easily accommodate all the traffic from a small city or a large section of a large city. Another example involves large corporate and financial networks that use DS3 signals contained in STS-1s to carry all their information. The severity of a failure is much worse when an entire optical fiber is cut. Current WDM systems can handle nearly 200 OC-192 signals, which translate into approximately

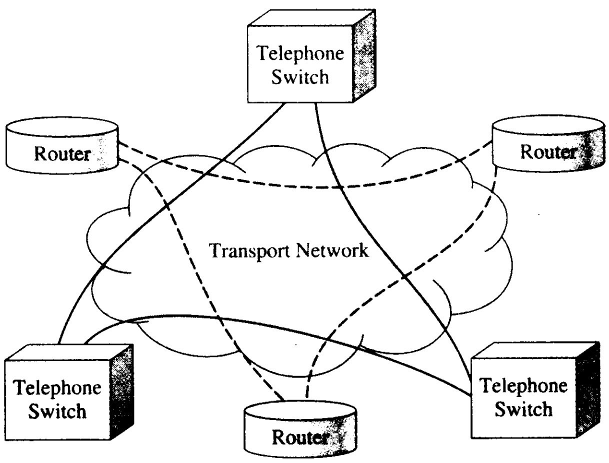


FIGURE 4.16 Transport network.

800 OC-48 signals and in turn potentially 25 million voice calls! Clearly, transport networks need to be designed to be very resilient with respect to faults.

In this section we consider first how SONET multiplexers are organized to form transport networks that provide services in the form of STS-n or virtual tributary connections. We introduce SONET linear, ring, and mesh topologies and discuss associated protection and restoration schemes. We then discuss how all-optical networks can be organized similarly to provide wavelength connections between clients.

4.3.1 SONET Networks

SONET systems produced a major advance in how multiplexing is carried out in transport networks. Prior to SONET, "asynchronous" multiplexing systems such as those used with DS1 and DS3 signals used bit stuffing to deal with bit slips and so required the entire multiplexed stream to be demultiplexed to access a single tributary, as shown in Figure 4.17a. Transit tributaries would then have to be remultiplexed onto the next hop. Thus every point of tributary removal or insertion required a back-to-back demultiplexer-multiplexer pair. Back-to-back multiplexers are an unnecessary expense in situations where most of the traffic is transit and only a few tributaries need to be dropped.

SONET produced significant reduction in cost by enabling **add-drop multiplexers (ADM)** that can insert and extract tributary streams without disturbing tributary streams that are in transit as shown in Figure 4.17b. The overall cost of the network can then be reduced by replacing back-to-back multiplexers with SONET ADMs.

Transport networks are built using SONET ADMs arranged in linear and ring topologies. In the next section, we discuss the operation of these topologies and their fault protection and recovery mechanisms. We then discuss briefly how SONET cross-connects can be used to build mesh topology networks.

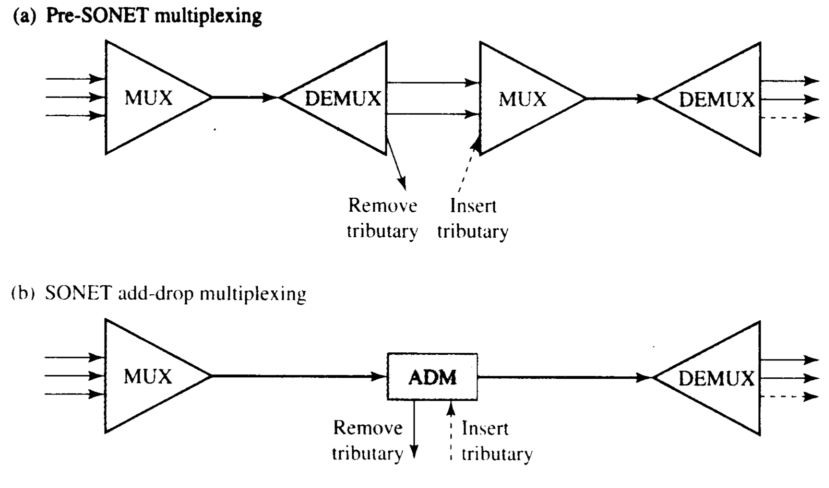


FIGURE 4.17 SONET add-drop multiplexing.

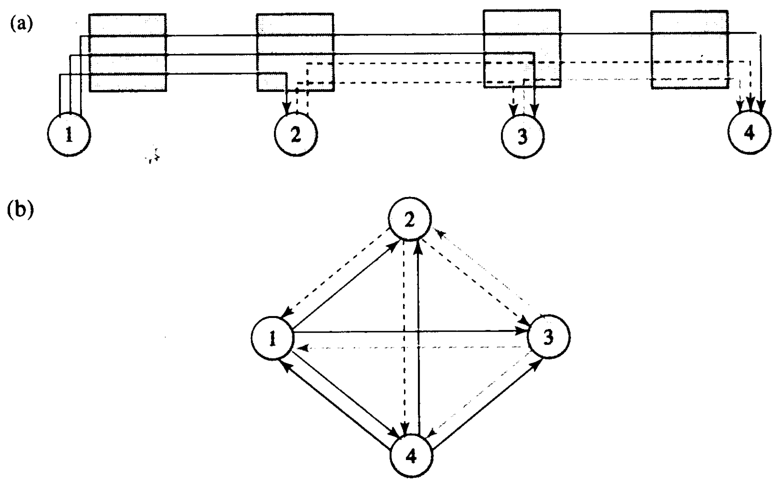


FIGURE 4.18 (a) ADMs arranged in linear topology to four terminals; (b) from the terminals' viewpoint: a fully connected mesh. A similar linear arrangement in the opposite direction provides paths in the reverse direction.

LINEAR SONET NETWORKS AND AUTOMATIC PROTECTION SWITCHING

Figure 4.18a shows how ADMs can be arranged in linear fashion to interconnect various SONET terminals. These terminals can be part of other equipment, for example, they could be interfaces to a large router. In this example each terminal has a SONET path to each of its downstream nodes. A similar linear arrangement in the opposite direction provides paths in the reverse direction. At each ADM, only the signals that are destined for that site are accessed and dropped, and transit signals are passed through. Signals generated locally and destined for other sites are inserted or "added" at the ADM. From the point of view of the terminals, they are arranged in the fully connected mesh shown in Figure 4.18b. In general, SONET ADMs can be used to create different "virtual" topologies by selecting which terminals are connected directly by paths.

The SONET standards define **automatic protection switching (APS)** schemes that provide linear protection against failures *at the line layer.* Recall that in SONET terminology, a line connects two multiplexers with a single OC-N optical signal. Therefore *protection at the line level* applies to a multiplexed signal while it traverses the line between two multiplexers. Figure 4.19 shows two such SONET multiplexers connected

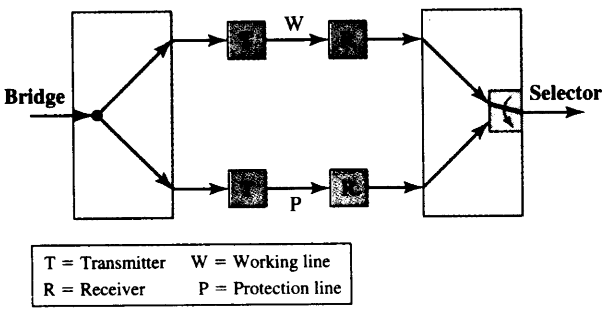


FIGURE 4.19 A working line and a protection line operate in parallel to provide 1+1 (one plus one) linear APS protection.

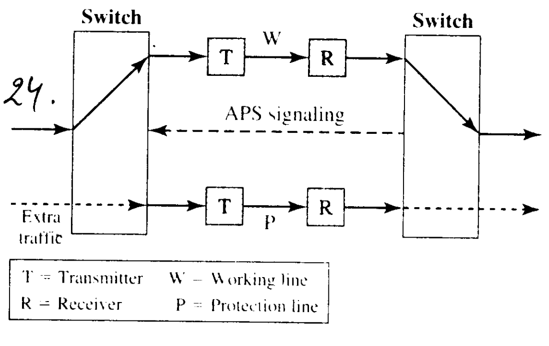


FIGURE 4.20 A protection line provides an alternate path for a single working line in 1:1 (one for one) linear APS protection.

in a 1+1 ("one plus one") configuration using *two* SONET lines, a working line and a protection line. At the upstream node, the signal is "bridged" electrically into working and protection lines that carry the same payload across the two lines. At the downstream node, the two received signals are monitored for failures. The monitoring looks for loss of signal, loss of framing, bit-error-rate levels, as well as alarm signals in the overhead. A selector in the downstream node picks the better signal based on the information provided by the two monitors and does not need to coordinate with the upstream node. Recovery from failures can be done very quickly because the monitoring and selection functions are typically implemented in hardware. However, the 1+1 APS scheme is inefficient because it uses twice the bandwidth required by an unprotected signal.

Figure 4.20 shows a 1:1 ("one for one") APS arrangement. In this approach the signal is only transmitted in the working line during normal operation. The optical signal that is received in the working line is monitored for degradation and a request to switch to the protection line is sent on a reverse signaling channel when a failure is detected. Upon receipt of the request, the upstream node switches the signal to the protection line. The 1:1 APS scheme takes more time to recover from failure than the 1+1 APS scheme because of the need to signal for a switchover. However, the 1:1 scheme can be more efficient in bandwidth usage because the protection line can be used to carry extra traffic when there are no failures. The extra traffic is preempted when the protection line is needed to protect working traffic. The line overhead (in K_1 and K_2 bytes) in the SONET frame contains fields that are used to exchange requests and acknowledgments for protection switch actions.

Figure 4.21 shows that the 1:1 APS scheme can be generalized to a 1:*n* APS scheme where one protection line protects *n* working lines. The scheme assumes that the working lines are unlikely to fail at the same time. This assumption is reasonable if the lines use diverse transmission routes. Both 1:1 and 1:*n* APS schemes are *revertive* in that the working signal must be switched back to its original line once the fault has been repaired.

The SONET linear APS specifications require that fault recovery be completed within 50 milliseconds in all of the above schemes.

RING NETWORKS

SONET ADMs can also be arranged in ring topology networks. Figure 4.22 shows three sites, a, b, and c, that are connected by three add-drop multiplexers. The ADMs

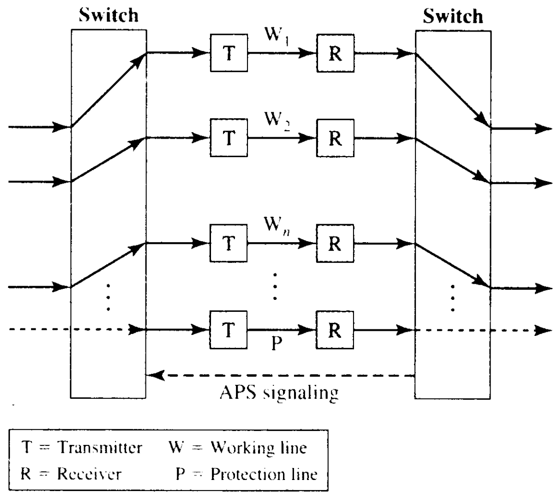


FIGURE 4.21 A protection line provides an alternate path for *n* working lines in 1:*n* linear APS protection.

are all connected in a unidirectional ring by an OC-3*n* optical transmission system that carries three STS-*n* signals. Figure 4.22 shows how, at node b, two STS-*n* tributaries are inserted destined for node c and for node a, respectively. The first tributary terminates at node c, and the second tributary flows across node c and terminates at node a. The ADM at each site also removes two STS-*n* tributaries and inserts two STS-*n* tributaries, and it passes one STS-*n* tributary unchanged as shown in Figure 4.23a. The first inserted tributary is destined to the next node, and the other inserted tributary is destined to the remaining node. For example, the ADM at site c removes the tributaries indicated by the solid and dashed lines that originated at nodes b and a, respectively. The ADM at site c also inserts tributaries destined from nodes a and b that are indicated by solid lines. The network in Figure 4.23a has a physical ring topology, but in fact, each pair of nodes is connected *directly* by an STS-*n* tributary, and so the three nodes are *logically* configured in a fully connected topology, as shown in Figure 4.23b. If switches at each of the three sites are interconnected by these tributaries, then the switches would see a fully connected topology.

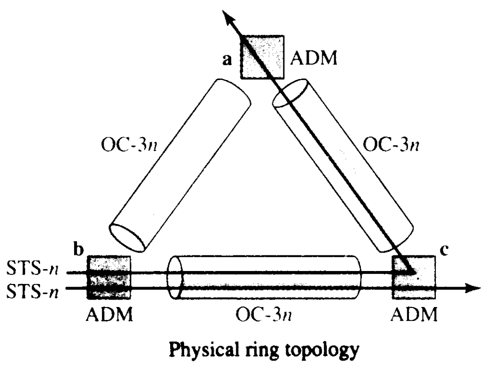


FIGURE 4.22 SONET ring network.

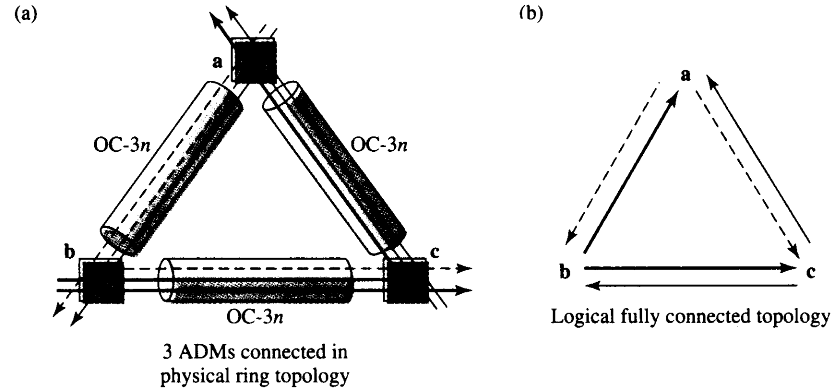


FIGURE 4.23 Configuration of logical networks using ADMs.

SONET rings can be deployed in a *self-healing* manner so that they recover from failures. Self-healing rings can provide protection at the *line level* as in linear APS systems, and they can also provide protection at the *path level.* Note that path level protection is end-to-end in that it covers the path level signal from its source terminal to its destination terminal.

We first consider the **unidirectional path switched ring (UPSR)**, which provides protection at the path level. Figure 4.24 shows a two-fiber ring in which data travels in one direction in one ring and in the opposite direction in the opposite ring. By convention, working traffic flows clockwise and protection traffic works counterclockwise.

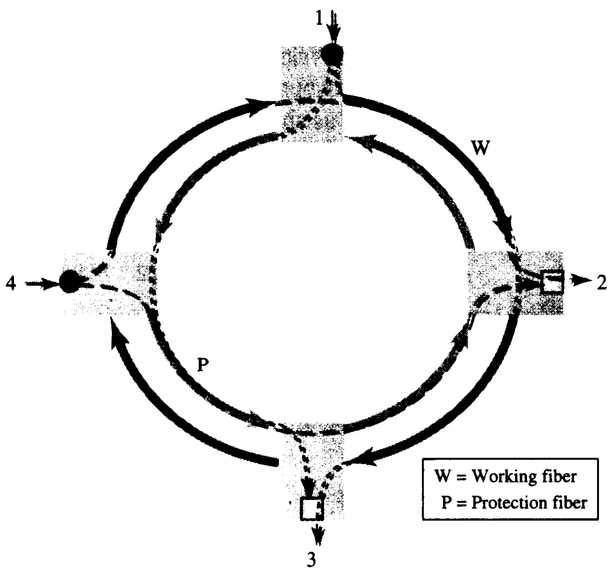


FIGURE 4.24 Unidirectional path switched ring.

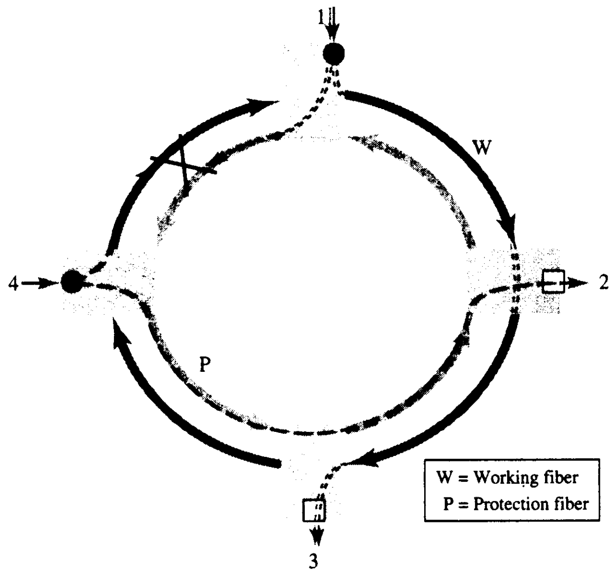


FIGURE 4.25 Protection switching in UPSR.

At each ADM, say, node 1 in Figure 4.24, the path signal to another node, say, 2, is bridged into both the working fiber and the protection fiber. Thus information can flow between any pair of nodes in the ring along two paths, providing $1 + 1$ protection at the path level. For example, Figure 4.24 shows a path signal inserted at node 4 that travels around the ring in both directions and exits at node 2.

In UPSR each exit node monitors the two received *path* signals and selects the better one. For example, suppose that a fiber is cut between two nodes, say, nodes 4 and 1 in Figure 4.25. Node 1 can no longer receive the working signal in the clockwise direction, so it inserts a path alarm indication in the overhead of every affected path signal. A selector at the receiver then selects the path that arrives in the protection fiber as shown in Figure 4.25. In this example we see that the selector in node 2 switches to the counterclockwise signal to receive the path signal from node 4.

UPSR rings can provide fast path protection, but are inefficient in terms of bandwidth usage since two paths are used to carry every signal. If a path uses an STS-n signal, then the path will use STS-n in every hop in both rings. UPSR rings are used widely in the lower speed rings in the access portion of networks where traffic is gathered from various remote sites and carried back to a central hub site prior to being directed to other parts of the network.

SONET rings can also be configured to provide protection at the *line* level. Figure 4.26 shows a four-fiber **bidirectional line switched ring (BLSR).** Adjacent ADMs in the ring are connected by a working fiber pair and a protection fiber pair. Suppose that failure disrupts the working pair between nodes 2 and 3 as shown in Figure 4.27. The BLSR operation recovers from this failure by switching both working channels to the protection channels between nodes 2 and 3. In effect, line APS protection is used in identical fashion to linear ADM networks. This type of recovery is called *span switching*.

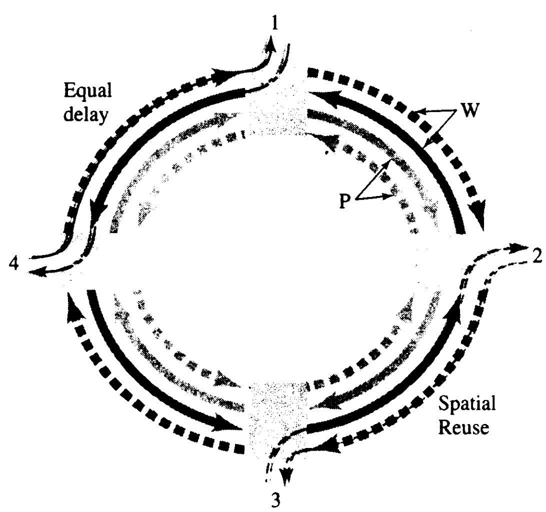


FIGURE 4.26 Bidirectional line switched ring: Standby bandwidth is shared.

Now suppose that the working pair and the protection pair between adjacent nodes fail together as shown in Figure 4.28. In this case BLSR uses the protection pair in the directions away from the failure to restore the two working lines that spanned the failure. The nodes on either side of the failure bridge their working line onto the protection line that flows in the direction away from the failure. In effect the working lines that traversed the failed path are now routed to their adjacent node the long way around the ring as shown in Figure 4.28. This type of recovery is called *ring switching*.

BLSR is more efficient than UPSR in that traffic can be routed along the shortest path, so that the bandwidth around the ring can be used to support more traffic. In addition, the protection fibers in BLSR can be used to carry extra traffic when there are

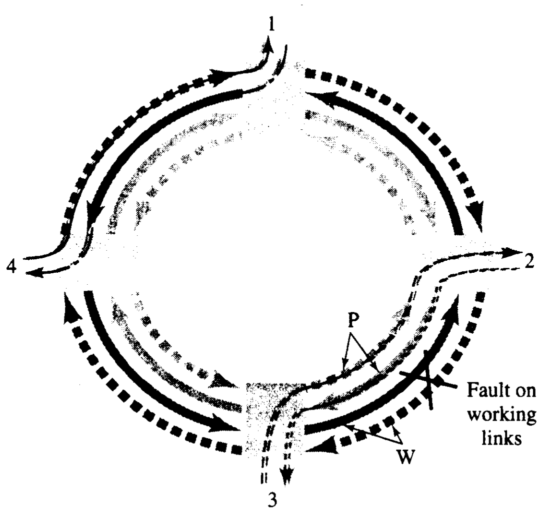


FIGURE 4.27 Span switching BLSR: Failed *line* is restored locally.

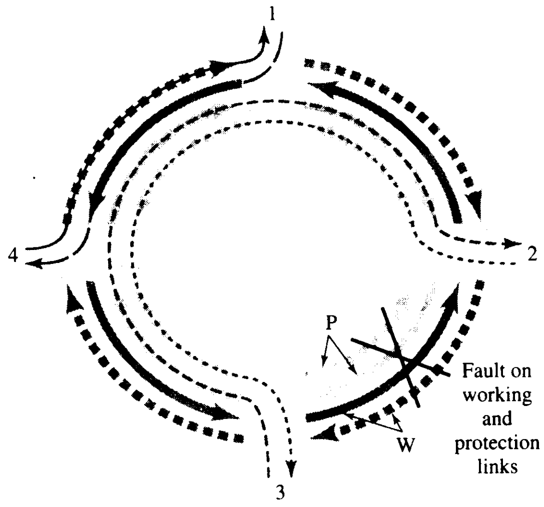


FIGURE 4.28 Ring switching in BLSR: Failed line is restored locally by switching around the ring.

no faults in the ring. For this reason, BLSR is preferred in high-speed backbone networks that involve very expensive transmission lines that cover thousands of kilometers. On the other hand, BLSR requires fairly complex signaling to enable the nodes to determine when span switching and when ring switching is being used, and to clear extra traffic from the ring when a recovery from faults is underway.

The capability to manage bandwidth flexibly and to respond quickly to faults has altered the topology of long-distance and metropolitan area networks from a mesh of point-to-point links to interconnected ring networks. SONET ring networks can be deployed in a metropolitan area as shown in Figure 4.29. User traffic is collected by

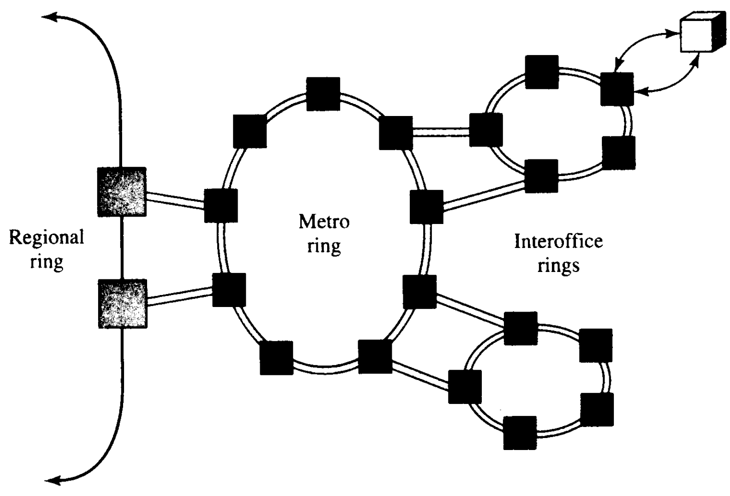


FIGURE 4.29 SONET ring structure in local, metropolitan, and regional networks.

access networks and directed to access nodes such as a telephone office. A number of such nodes are interconnected in a first-tier ring network. Large users that cannot afford to lose service may be connected to an access node with dual paths as shown. A metropolitan area ring operating at a higher rate may in turn interconnect the first tier ring networks. To provide protection against faults, rings may be interconnected using matched inter-ring gateways as shown between the interoffice ring and the metro ring and between the metro ring and the regional ring. The traffic flow between the rings is sent simultaneously along the primary and secondary gateway. Automated protection procedures determine whether the primary or secondary incoming traffic is directed into the ring. The metropolitan area ring, in turn, may connect to the ring of an interexchange or regional carrier as shown in the figure.

SONET CROSS-CONNECTS AND MESH NETWORKS

SONET ring networks have been widely deployed because of their ability to provide continuous service even in the presence of faults. However, SONET ring networks have proved to be quite difficult to manage in an environment of rapid growth. As networks are required to carry more traffic, certain spans in a ring will become congested. Unfortunately, to increase the capacity of a single span in a ring network, *all* the ADMs need to be upgraded at the same time. This upgrade involves tremendous expense and effort.

An alternative approach to meeting increased traffic demand is to build multiple parallel ring networks. This approach is compatible with WDM transmission systems, which can provide the parallel optical transmission channels required by the parallel rings using a single optical fiber. However, additional cost must again be incurred to interconnect the parallel rings so that they can exchange traffic. At each node, the signal carried in each wavelength must be connected to a separate ADM. To exchange traffic between ADMs at a node, the traffic must first be dropped from a source ADM, connected to a switch, and then added into the destination ADM.

For the above reasons, *SONET cross-connect systems* have received increased attention. A cross-connect system can take SONET OC-n optical signals at its inputs, decompose these signals into its component STS-1 or other tributaries, switch these component signals to output ports, and combine the component signals into outgoing SONET OC-n signals.

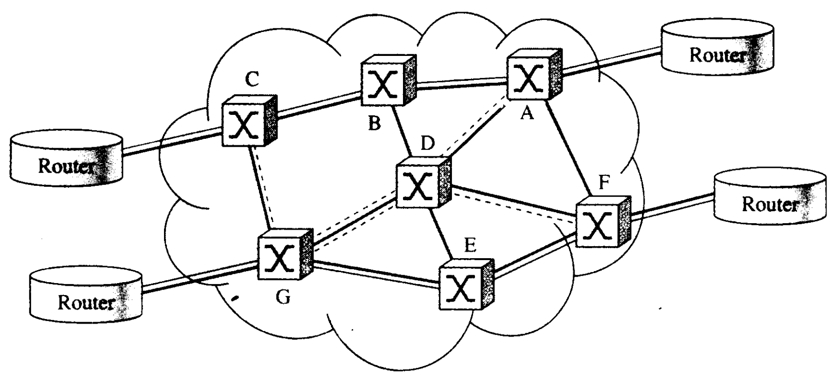


FIGURE 4.30 Mesh topology networks using SONET cross-connects.

Mesh networks are constructed by interconnecting cross-connect systems with SONET optical lines as shown in Figure 4.30. When certain segments become congested, mesh networks have the advantage that only the capacity of the affected segments needs to be upgraded. Mesh networks require less protection bandwidth than ring networks, but they also require more complex protection schemes to recover from failures. For example, the dashed lines in Figure 4.30 represent shared protection paths for the two connections denoted by the solid lines.

THE VIRTUALIZATION OF CIRCUITS

The initial notion of a connection corresponding to a physical circuit has been virtualized with the introduction of multiplexers. An individual digital voice connection consists of a sequence of PCM bytes that flows across the network as part of aggregated TDM flows. The notion of a circuit no longer corresponds to a physical connection but rather to an allocation of time slots and crosspoints at regularly scheduled times. When we examine packet switching we will see that the notion of a connection can be virtualized even further where packet flows follow a fixed path but are not allocated specific transmission or switching times. In the next section we see that the notion of an optical wavelength connection is also virtualized in an analogous fashion where multiple optical signals are combined into a composite multiwavelength optical signal that flows across a given optical fiber.

4.3.2 Optical Transport Networks

Optical transport networks provide *optical wavelength* connections between attached clients. There are many similarities between the way a SONET OC-n signal carries STS-n signals and the way a WDM optical signal carries multiple wavelength signals. Thus it is not surprising that optical transport networks can use many of the same methods as SONET networks.

Optical add-drop multiplexers (OADM) have been designed for WDM systems. An OADM takes a multiwavelength signal arriving in an input fiber, drops one or more pre-selected wavelengths at a site, and adds one or more pre-selected wavelengths into the multiwavelength signal that exits in an output fiber. Wavelengths carrying transit traffic can "bypass" the given site. Ideally, all processing in an OADM is performed in the optical domain, so expensive optical-to-electrical conversion is avoided.

OADMs can be arranged in linear and ring topologies. The assignment of wavelength paths can then be used to create networks with various virtual topologies. In these topologies a *light path* between two terminals is created by inserting information at an assigned wavelength at the source terminal, bypassing intermediate OADM nodes, and removing the information at the destination terminal. Figure 4.31a shows a chain of optical add-drop multiplexers in which a single fiber connects adjacent multiplexers. Each fiber contains a set of four wavelengths that are removed and inserted to provide a one-directional communication link from upstream to downstream nodes. Thus a has a link to each of b, c, and d; b has a link to each of c and d; and c has a link to d.

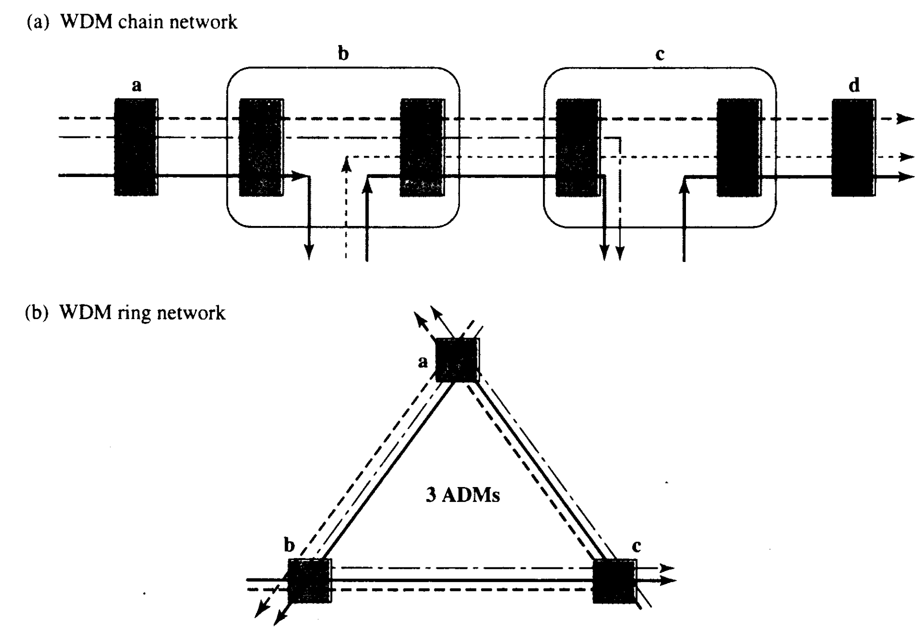


FIGURE 4.31 Network configurations using WDM multiplexers.

Figure 4.31b shows a WDM ring network in which three nodes are connected by an optical fiber that carries three wavelengths. Each node removes two wavelengths and inserts two wavelengths so that each pair of nodes is connected by an information stream flowing in one wavelength. In effect a fully connected logical network is produced. We again see that through the assignment of wavelengths, it is possible to obtain virtual topologies that differ from the physical topology.

The introduction of WDM and optical add-drop multiplexers into a network adds a layer of logical abstraction between the optical fiber physical topology and the logical topology that is seen by the systems that send traffic flows through the network. The physical topology consists of the optical add-drop multiplexers interconnected with a number of optical fibers. The manner in which light paths are defined by the OADMs in the WDM system determines the topology that is seen by SONET ADMs that are interconnected by these light paths. The systems that input tributaries into the SONET network in turn may see another different topology that is defined by the SONET system. For example, in Figure 4.31b each node could correspond to a different metropolitan area. Each metropolitan area might have a network of interconnected SONET rings. The light paths between the areas provide a direct interconnection between these metropolitan networks.

An advantage of all optical networks is that they can carry signals transparently. In WDM each wavelength is modulated separately, so each wavelength need not carry information in the same transmission format. Thus some wavelengths might carry SONET-formatted information streams, while others might carry Gigabit-Ethernet-formatted information or other transmission formats.

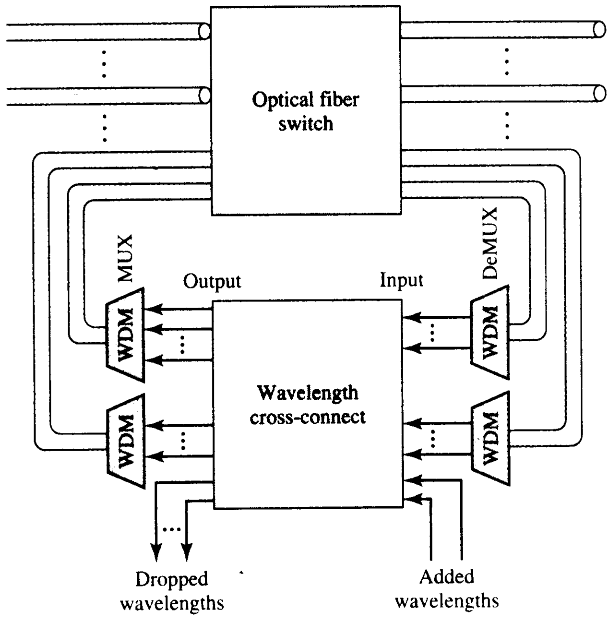


FIGURE 4.32 All optical fiber switch and cross-connect.

Optical mesh networks based on *optical cross-connect* and *optical fiber* switching systems are currently being considered for deployment in the backbone of transport networks. The purpose of a fiber switch is to transfer entire multiwavelength signals from input ports to output ports without WDM demultiplexing. Figure 4.32 shows the relationship between a fiber switch and an optical cross-connect. Optical signals that carry WDM signals arrive in individual fibers to a node. Optical signals that carry signals destined for this node are switched to the local wavelength cross-connect; optical signals that bear only transit signals are switched across the node. The optical signals that are switched to the local cross-connect are demultiplexed into its component wavelengths. The wavelength signals that are destined for this node are then dropped. Locally generated wavelength signals are added through the cross-connect. The outputs from the cross-connect are fed to WDM multiplexers that produce composite optical signals that are then inserted into the fiber switch and transferred out in an outgoing fiber. Note that the cross-connect system must ensure that the signals fed into the WDM multiplexer have distinct wavelengths. Tunable lasers or some other means of wavelength translation may be needed to accomplish this.

Optical networks are seen as a means to keep network operation simple as the volume of traffic carried by networks continues to increase. The cost of demodulating a single WDM signal (that can carry hundreds of Gbps) and processing its components in the electronic domain is extremely high (see Figure 3.58). System costs can be kept down by keeping WDM signals in the optical domain as they traverse the network. The combination of "long-haul" optical transmission systems with optical switches that allow optical signals to bypass transit nodes can keep the number of optical-to-electronic conversions to a minimum.

HISTORY REPEATS ITSELF . . . AGAIN

Optical transmission is still in the early stages of development relative to its potential, so it is interesting to examine its possible evolution given the history of networks in the nineteenth and twentieth centuries. During this period networks went through a cycle from digital techniques in telegraphy to analog techniques in the early phase of telephony and back to digital techniques in modern networks. WDM technology is clearly in the analog phase of this development cycle, and necessarily so because of limitations in electronic and optical devices. By looking to the past, we can see clearly that optical time-division multiplexing must be on the horizon. With the development of optical implementation of simple logical operations, we can also expect some forms of optical packet switching and optical code division systems (introduced in Chapter 6). Looking further into the future, optical computing, should it become available, would affect networking as much as computer control changed signaling in telephone networks and inexpensive processing made the Internet possible. These insights are gained from looking to the past. Of course, then we have the radical ideas that (appear to) come out of nowhere to change the course of history. Stay tuned!

4.4 CIRCUIT SWITCHES

A network is frequently represented as a cloud that connects multiple users as shown in Figure 4.33a. A circuit-switched network is a generalization of a physical cable in the sense that it provides connectivity that allows information to flow between inputs and outputs to the network. Unlike a cable, however, a network is geographically distributed and consists of a graph of transmission lines (that is, links) interconnected by switches (nodes). As shown in Figure 4.33b, the function of a **circuit switch** is to transfer the signal that arrives at a given input to an appropriate output. The interconnection of a sequence of transmission links and circuit switches enables the flow of information between inputs and outputs in the network.

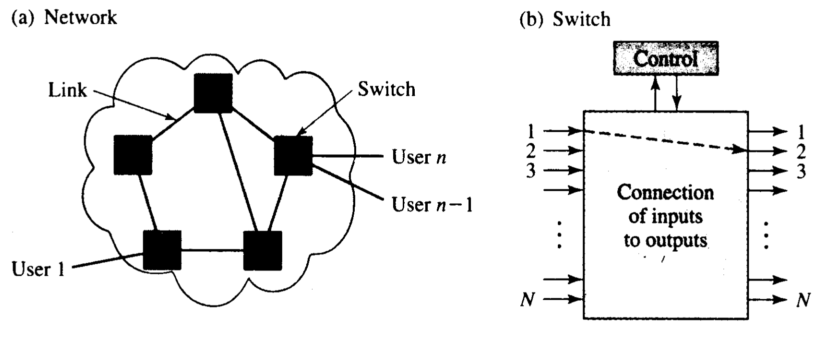


FIGURE 4.33 A network consists of links and switches.

In the first part of this section we consider the design of circuit switches that transfer the information from one incoming line to one outgoing line. The first telephone switches were of this type and involved the establishment of a physical path across the switch that enabled the flow of electric current from an input line to an output line. The principle of circuit switches is general, however, and one could consider the design of optical circuit switches that enable the transfer of optical signals from an input line to an output line.

In many cases the input lines to a switch contain multiplexed information flows, and the purpose of the switch is to transfer each specific subflow from an input line to a specific subflow in a given output line. In principle the incoming flows must first be demultiplexed to extract the subflows that can then be transferred by the switch to the desired output links. In Section 4.4.2 we consider the case where the incoming and outgoing flows are time-division multiplexed streams. The associated digital circuit switches form the basis for modern telephone switches.

4.4.1 Space-Division Switches

The first switches we consider are called **space-division switches** because they provide a separate physical connection between inputs and outputs so the different signals are separated in space. Figure 4.34 shows the crossbar switch, which is an example of this type of switch. The **crossbar switch** consists of an $N \times N$ array of *crosspoints* that can connect any input to any available output. When a request comes in from an incoming line for an outgoing line, the corresponding crosspoint is closed to enable information to flow from the input to the output. The crossbar switch is said to be a **nonblocking switch;** in other words, connection requests are never denied because of lack of connectivity resources, that is, crosspoints. Connection requests are denied only when the requested outgoing line is already engaged in another connection.

The complexity of the crossbar switch as measured by the number of crosspoints is N^2. This number grows quickly with the number of input and output ports. Thus a 1000-input-by-1000-output switch requires 10^6 crosspoints, and a 100,000 by 100,000 switch requires 10^{10} crosspoints. These numbers of crosspoints cannot be implemented. In the next section we show how the number of crosspoints can be reduced by using multistage switches.

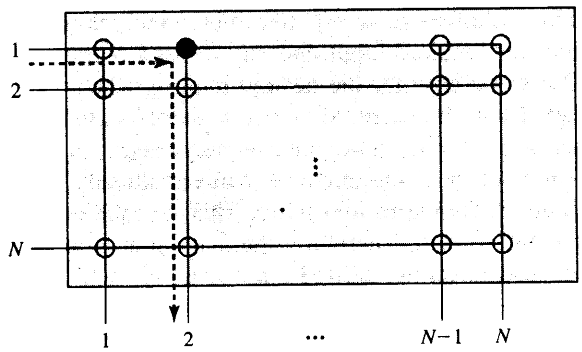


FIGURE 4.34 Crossbar switch.

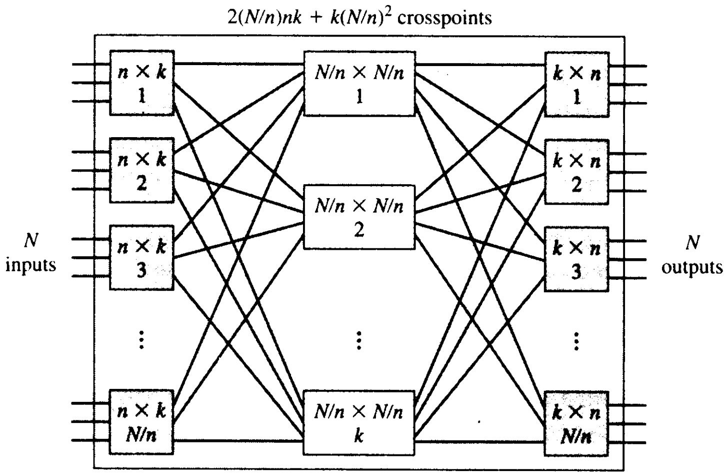


FIGURE 4.35 Multistage switch (with three smaller space-division switches).

◆ MULTISTAGE SWITCHES

Figure 4.35 shows a **multistage switch** that consists of three stages of smaller space-division switches. The N inputs are grouped into N/n groups of n input lines. Each group of n input lines enters a small switch in the first stage that consists of an $n \times k$ array of crosspoints. Each input switch has one line connecting it to each of k intermediate stage $N/n \times N/n$ switches. Each intermediate switch in turn has one line connecting it to each of the N/n switches in the third stage. The latter switches are $k \times n$. In effect each set of n input lines *shares* k *possible paths to any one of the switches at the last stage*; that is, the first path goes through the first intermediate switch, the second path goes through the second intermediate switch, and so on. The resulting multistage switch is not necessarily nonblocking. For example, if $k < n$, then as soon as a switch in the first stage has k connections, all other connections will be blocked.

◆ CLOS NONBLOCKING SWITCHING FABRIC

The question of determining when a multistage switch becomes nonblocking was answered by [Clos 1953]. Consider any desired input and any desired output such as those shown in Figure 4.36. The worst case for the desired input is when all the other inputs in its group have already been connected. Similarly, the worst case for the desired output is when all the other outputs in its group have already been connected. The set of routes that maximize the number of intermediate switches already in use by the given input and output groups is shown in the figure. That is, each existing connection uses a different intermediate switch. Therefore, the maximum number of intermediate switches not available to connect the desired input to the desired output is $2(n - 1)$. Now suppose that $k = 2n - 1$; then k paths are available from any input group to any output group. Because $2(n - 1)$ of these paths are already in use, it then

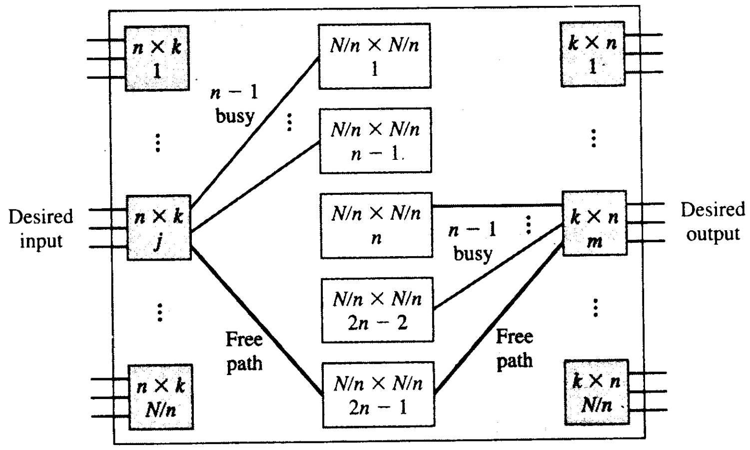


FIGURE 4.36 A multistage switch is nonblocking if $k = 2n - 1$.

follows that a single path remains available to connect the desired input to the desired output. Thus the multistage switch with $k = 2n - 1$ is nonblocking.

The number of crosspoints required in a three-stage switch is the sum of the following components:

- N/n input switches × nk crosspoints/input switch.
- k intermediate switches × $(N/n)^2$ crosspoints/intermediate switch.
- N/n output switches × nk crosspoints/output switch.

In this case the total number of crosspoints is $2Nk + k(N/n)^2$. The number of crosspoints required to make the switch nonblocking is $2N(2n-1) + (2n-1)(N/n)^2$. The number of crosspoints can be minimized through the choice of group size n. By differentiating the above expression with respect to n, we find that the number of crosspoints is minimized if $n \approx (N/2)^{1/2}$. The minimum number of crosspoints is then $4N((2N)^{1/2} - 1)$. We then see that the *minimum number of crosspoints using a Clos*

CROSSBAR CHIPS AND CLOS NETWORKS

The current generation of microelectronic chips makes it possible to build very large circuit switching fabrics. For example, a 144 × 144 crossbar switch is currently available where each port handles 3.125 Gbps, which is sufficient to handle an STS-48 signal. The throughput of this single chip is an amazing 450 Gbps!

The Clos three-stage construction allows us to build even bigger switches. For example, a three-stage arrangement that uses 144 8 × 16 switches in the first stage, 16 144 × 144 switches in the middle stage, and 144 16 × 8 switches in the third stage gives a circuit switch with size 1152 × 1152 (since 8 × 144 = 1152), and throughput 3.6 terabits/second.

nonblocking three-stage switch grows at a rate proportional to $N^{1.5}$, which is less than the N^2 growth rate of a crossbar switch.

When $k < 2n - 1$, there is a nonzero probability that a connection request will be blocked. The methods for calculating these probabilities can be found in [Bellamy 1991, pp. 234–242].

4.4.2 Time-Division Switches

In Section 4.1.2, we explained how TDM could replace multiple physical lines by a single high-speed line. In TDM a slot within a frame corresponds to a single connection. The **time-slot interchange (TSI)** technique replaces the crosspoints in a space switch with the reading and writing of a slot into a memory. Suppose we have a number of pairs of speakers in conversation. The speech of each speaker is digitized to produce a sequence of 8000 bytes/second. Suppose that the bytes from all the speakers are placed into a T-1 carrier, as shown in Figure 4.37. Suppose also that the first pair of speakers has been assigned slots 1 and 23. For the speakers to hear each other, we need to route slots 1 and 23 in the incoming frames to slots 23 and 1 in the outgoing frames. Similarly, if the second pair of speakers is assigned slots 2 and 24, we need to interchange incoming slots 2 and 24 with outgoing slots 24 and 2, respectively.

Figure 4.37 shows the interchange technique: The octets in each incoming frame are written into a register. The call setup procedure has set a permutation table that controls the order in which the contents of the register are read out. Thus the outgoing frame begins by reading the contents of slot 23, followed by slot 24, and so on until slots 1 and 2 are read, as shown in the figure. This procedure can connect any input to any available output. Because frames come in at a rate of 8000 times a second and the time-slot interchange requires one memory write and one memory read operation per slot, the maximum number of slots per frame that can be handled is

$$\text{Maximum number of slots} = \frac{125\ \mu\text{sec}}{2 \times \text{memory cycle time}} \tag{4.4}$$

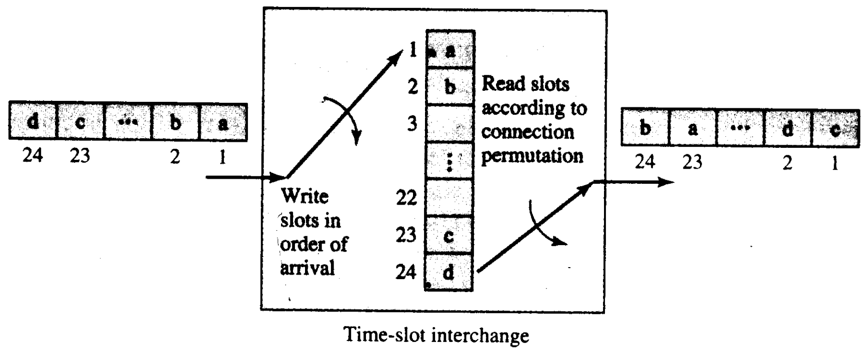


FIGURE 4.37 Time-slot interchange technique: Crosspoints in a space switch are replaced with the reading and writing of slots into memory.

For example, if the memory cycle time is 125 nanoseconds, then the maximum number of slots is 500, which can accommodate 250 connections.

The development of the TSI technique was crucial in completing the digitization of the telephone network. Starting in 1961 digital transmission techniques were introduced in the trunks that interconnected telephone central offices. Initially, at each office the digital streams would be converted back to analog form and switched by using space switches of the type discussed in the previous section. The introduction of TSI in digital time-division switches led to significant reductions in cost and to improvements in performance by obviating the need to convert back to analog form. Most modern telephone backbone networks are now entirely digital in terms of transmission and switching.

◆ TIME-SPACE-TIME SWITCHES

We now consider a hybrid switch design in which TSI switches are used at the input and output stages and a crossbar space switch is used at the intermediate stage. These switches are called **time-space-time switches.** *The design approach is to establish an exact correspondence between the input lines in a space-division switch in the first stage and time slots in a TSI switch.* Suppose we replace the $n \times k$ switch in the first stage of a multistage space switch by an $n \times k$ TSI switch, as shown in Figure 4.38. Each input line to the switch corresponds to a slot, so the TSI switch has input frames of size n slots. Similarly, the output frame from the TSI switch has k slots. Thus the operation of the TSI switch involves taking the n slots from the incoming frame and reading them out in a frame of size k, according to some preset permutation table. Note that for the system to operate in synchronous fashion, the transmission time of an input frame must be equal to the transmission time of an output frame. Thus, for example, if $k = 2n - 1$, then the internal speed is nearly double the speed of the incoming line.

Consider now the flow of slots between the switches in the first stage and the switches in the intermediate stage. We assume that frames coming out of the TSI switches in the first stage are synchronized. Consider what happens as the first slot

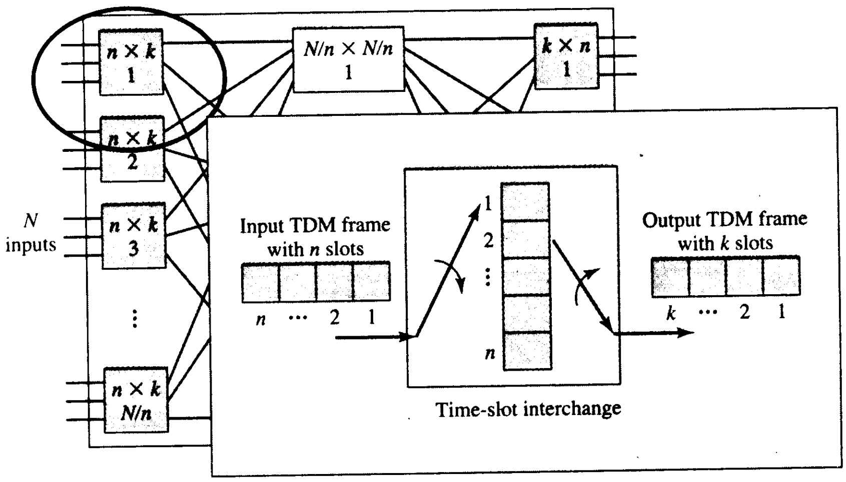


FIGURE 4.38 Hybrid switches: The input and output stages are TSI switches and the intermediate stage is a crossbar space switch.

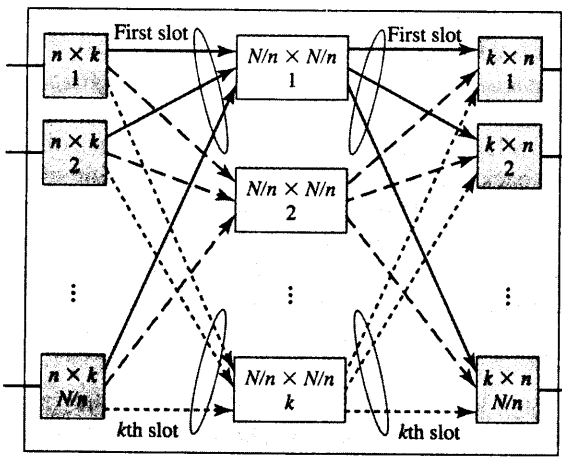


FIGURE 4.39 Flow of slots between switches in a hybrid switch.

in a frame comes out of the first stage. This first slot corresponds to the first output line out of each of the first stage switches. Recall from Figure 4.35 that the first line out of each first stage switch is connected to the first intermediate switch. Thus the first slot in each intermediate frame will be directed to intermediate switch 1, as shown in Figure 4.39. Recall that this switch is a crossbar switch, and so it will transfer the N/n input slots into N/n output slots according to the crosspoint settings. Note that all the other intermediate switches are idle during the first time slot.

Next consider what happens with the second slot in a frame. These slots are now directed to crossbar switch 2, and all other intermediate switches are idle. It thus becomes apparent that only one of the crossbar switches is active during any given time slot. This situation makes it possible to replace the k intermediate crossbar switches with a single crossbar switch that is time-shared among the k slots in a frame, as shown in Figure 4.40. To replace the k intermediate original crossbar switches, the time-shared

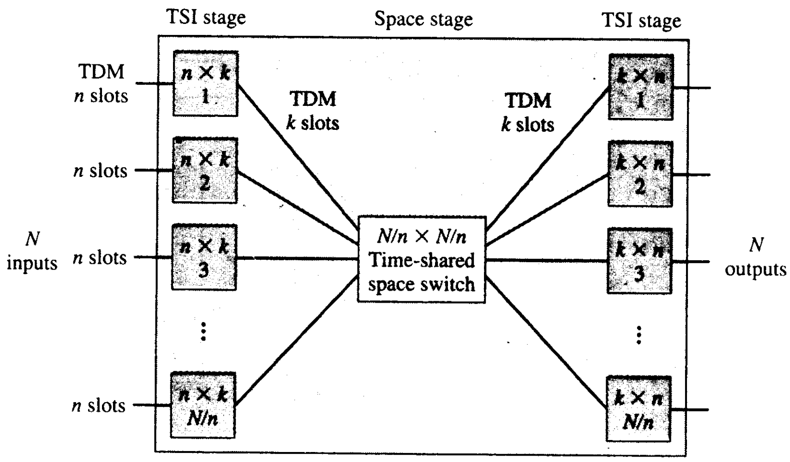


FIGURE 4.40 Time-space-time switches: The hybrid switch in Figure 4.38 requires only a single intermediate crossbar switch.

crossbar switch must be reconfigured to the interconnection pattern of the corresponding original switch at every time slot. This approach to sharing a space switch is called **time-division switching.**

EXAMPLE A 4 × 4 Time-Space-Time Switch

Figure 4.41 shows a simple 4 × 4 switch example that is configured for the connection pattern that takes inputs (A, B, C, D) and outputs (C, A, D, B). Part (a) of the figure shows a configuration of a three-stage space switch that implements this permutation. Part (b) shows the TST implementation where the inputs arrive in frames of size 2 that are mapped into frames of size 3 by the first-stage TSI switches. Note the correspondence between the arrangement of the first-stage space switches in part (a) and the TSI switches in part (b). For example, the top first-stage switch takes inputs A and B and outputs A in the first line, B in the second line, and nothing in the third line. The corresponding TSI switch takes the frame with A in the first slot and B in the second slot and outputs the frame with A in the first slot, B in the second slot, and nothing in the third slot.

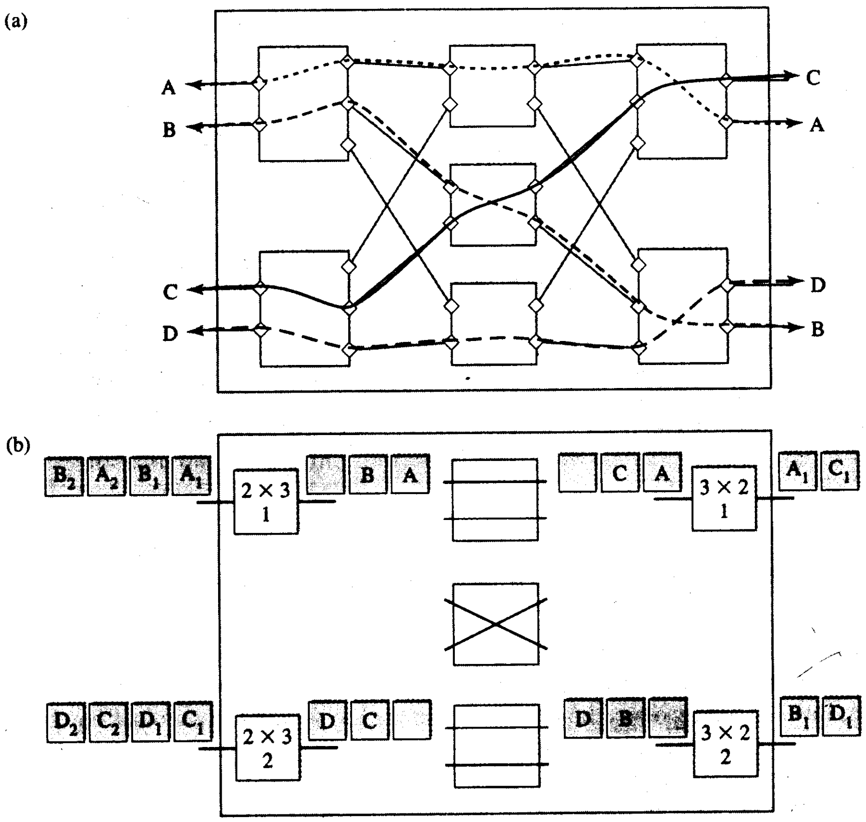


FIGURE 4.41 Example of a time-space-time switch.

The interconnection pattern for each of the three time slots is shown for the middle-stage space switch is also shown in part (b). Note that parts (a) and (b) have the same configurations. Finally, the mapping in the third stage from the three-slot frame to the two-slot frame is shown. By tracing the handling of each slot, we can see how the overall interconnection pattern (A, B, C, D) to (C, A, D, B) is accomplished. Thus for example, the output of the top third-stage space switch is C and A, and the output of the corresponding TSI switch is the frame with C in the first slot and A in the second slot.

The introduction of TSI switches at the input and output stages and the introduction of a single time-shared crossbar switch result in a much more compact design than space switches. The replacement of the N original input lines by the N/n TDM lines also leads to a compact design. For example, consider the design of a nonblocking 4096×4096 time-space-time switch that has input frames with 128 slots. Because $N = 4096$ and $n = 128$, we see that $N/n = 32$ input TSI switches are required in the first stage. The nonblocking requirement implies that the frames between the input stage and the intermediate stage must be of size $k = 2n - 1 = 255$ slots. The internal speed of the system is then approximately double that of the input lines. The time-shared crossbar switch is of size 32×32. The resulting switch can be seen to be quite compact.

TSI CHIPS

TSI functionality is incorporated into many microelectronic chips used in SONET equipment. For example, framer chips in an OC-48 SONET line card take an STS-48 stream, break it into its STS-1 components, and perform time-slot interchange prior to transferring the rearranged stream into a second stage switch.

Standalone time-space-time chips are also available. For example, a recent chip has 64 inputs and 64 outputs, with each input and output consisting of an STS-12 SONET signal. This single chip has a throughput of 40 Gbps and is equivalent to a 768×768 STS-1 switch.

4.5 THE TELEPHONE NETWORK

The modern telephone network was developed to provide basic telephone service, which involves the two-way, real-time transmission of voice signals. In its most basic form, this service involves the transfer of an analog signal of a nominal bandwidth of 4 kHz across a sequence of transmission and switching facilities. We saw in Chapter 3 that the digital transmission capacity of this 4 kHz channel is about 45 kbps, which is miniscule in relation to the speed of modern computers. Nevertheless, the ubiquity and low cost of the telephone network make it an essential component of computer communications.

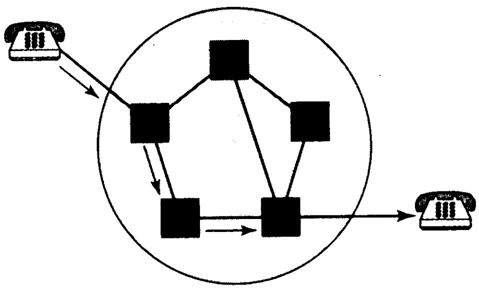


FIGURE 4.42 Circuit switching.

Telephone networks operate on the basis of *circuit switching*. Initially, circuit switching involved the setting up of a physical path from one telephone all the way across the network to the other telephone, as shown in Figure 4.42. At the telephone offices operators would make physical connections that would allow electric current to flow from one telephone to the other. The physical resources, such as wires and switch connections, were dedicated to the call for its entire duration. Modern digital telephone networks combine this circuit-switching approach to operating a network with digital transmission and digital switching. In the following discussion we explain how a telephone call is set up in a modern digital network.

Figure 4.43 shows the *three phases of connection-oriented communications*. When a user picks up the phone, a current flow is initiated that alerts equipment in a switch at the local telephone office of a call request. The switch prepares to accept the dialed digits and then provides the user with a dial tone. The user then enters the telephone number either by turning a dial that generates a sequence of pulses or by pushing a series of buttons that generates a sequence of tones. The switch equipment converts these pulses or tones into a telephone number.

In North America each telephone is given a 10-digit number according to the North American numbering plan. The first three digits are the area code, the next three digits are the central office code, and the last four digits are the station number. The first six digits of the telephone number uniquely identify the central office of the destination telephone. To set up a call, the source switch uses the telephone-signaling network to find a route and allocate resources across the network to the destination office.[9] This step is indicated by the propagation of signaling messages in Figure 4.43. The destination office next alerts the destination user of an incoming call by ringing the

[9]The telephone signaling network is discussed in Section 4.6.

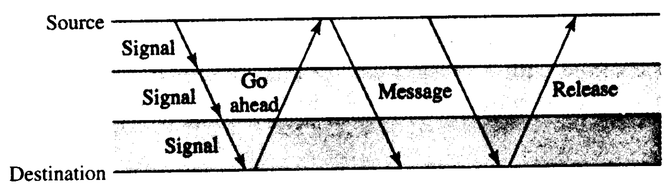


FIGURE 4.43 Telephone call setup.

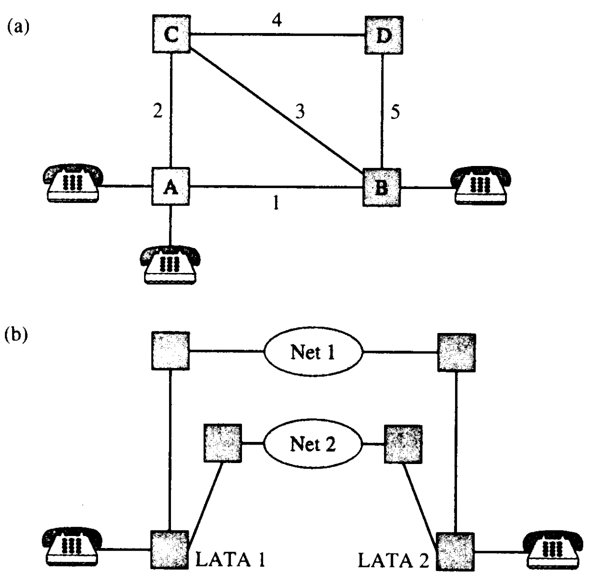


FIGURE 4.44 Routing in local and long-distance connections: (a) within a typical metropolitan area; (b) between two LATAs.

phone. This ring is sent back to the source telephone, and conversation can begin when the destination phone is picked up. The call is now in the message transfer phase. The call is terminated and the resources released when the users hang up their telephones.

The call setup procedure involves finding a path from the source to the destination. Figure 4.44a shows a typical arrangement in a metropolitan area. Central offices are connected by high-speed digital transmission lines that correspond to a group of trunks. If the two telephones are connected to the same central office, that is, the two phones are attached to switch A, then they are connected directly by the local switch. If the two telephones are connected to different central offices (that is, A and B), then a route needs to be selected.[10]

In the United States the telephone network is divided into *local access and transport areas (LATAs)* that are served by the *local exchange carriers (LECs)*. The LECs consist of *regional Bell operating companies (RBOCs)* such as Verizon, BellSouth, and SBC. Communication between LATAs is provided by separate independent *interexchange carriers (IXCs)*, that is, long-distance service providers such as AT&T, Sprint, and MCI.[11] Figure 4.44b shows how long distance calls are routed from a switch in the LATA to a facility that connects the call to the network of one of a number of interexchange carriers. The call is then routed through the network of the interexchange carrier to the destination LATA and then to the destination telephone.

Now let us consider the end-to-end path that is set up between the source and the destination telephones. In the majority of cases, the telephones are connected to their local telephone office with a twisted pair of copper wires: The voice signal flows in analog form from the telephone to the telephone office. The voice signal is converted

[10]The routing control procedures are discussed in Section 4.7.

[11]Further deregulation now allows the RBOCs to provide long-distance services in some states.

into digital form using PCM at the line card interface where the copper wires connect to the local telephone switch. The digitized voice signal flows from that point onward as a sequence of PCM samples over a path that has been set up across the network. This path consists of reserved time slots in transmission links that use TDM. The transmission links connect digital switches in which TSI arrangements have been made during call setup. Finally, at the destination switch the received PCM signal is converted back to analog form and transmitted to the destination telephone over the pair of copper wires.

The pair of copper wires that connect the user telephones to their local offices is called the "last mile," which constitutes the remaining obstacle to providing end-to-end *digital* connectivity. In the next section we discuss the structure of the last mile in more detail.

4.5.1 Transmission Facilities

The user's telephone is connected to its local telephone office by twisted-pair copper wires that are called the *local loop*. The wire pairs are stranded into groups, and the groups are combined in cables that can hold up to 2700 pairs. Wire pairs connect to the telephone office at the *distribution frame* as shown in Figure 4.45. From the local office the feeder cables extend in a star topology to the various geographic serving areas. Each feeder cable connects to a *serving area interface*. Distribution cables in turn extend in a star topology from the serving area interface to the *pedestals* that connect to the user's telephone.

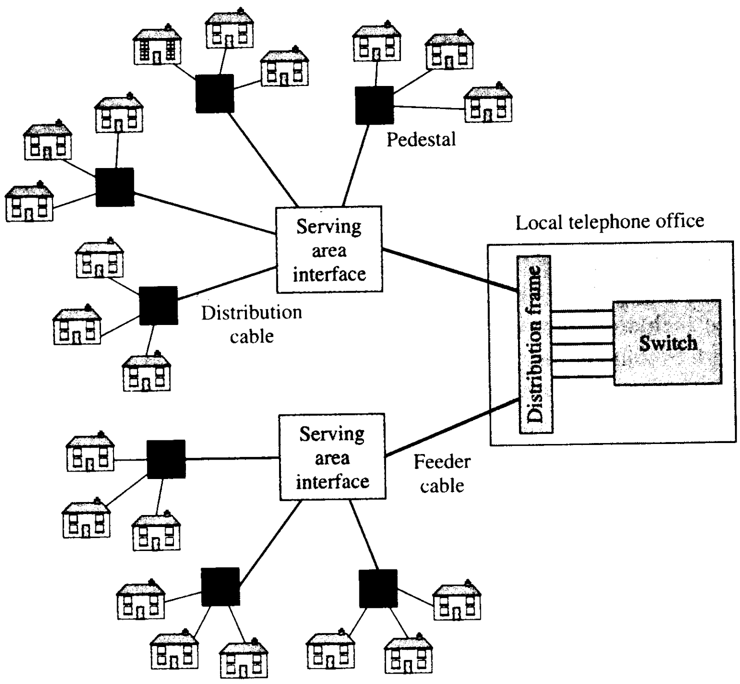


FIGURE 4.45 Access transmission facilities.

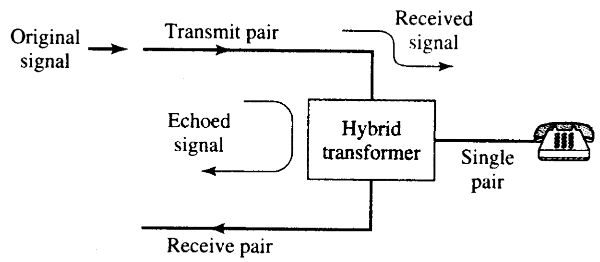


FIGURE 4.46 Two- and four-wire connections in the telephone network.

In the local loop a single pair of wires carries the information in both directions, from the user to the local switch and from the local switch to the user. Inside the network a separate pair of wires is used for each direction of signal flow. As shown in Figure 4.46, at the interface to the switch a device called a *hybrid transformer* converts the signals from the incoming wire pair to the four-wire connection that is used inside the network. These hybrid transformers tend to reflect some of the arriving signal resulting in *echoes* that under certain conditions can be disturbing to speakers and that can impair digital transmission. For these reasons, echo-cancellation devices have been developed that estimate the echo delay and use a properly delayed and scaled version of the transmitted signal to cancel the arriving echoes.

The twisted-wire pairs that connect the user to the local switch can be used to carry high-speed digital signals using ADSL and other digital transmission technologies as discussed in Chapter 3. In many instances the feeder cable that connects the serving area interface to the local switch is being replaced with an optical fiber that carries time-division multiplexed traffic. In this case the service area interface carries out the conversion to and from PCM. As optical fiber extends closer to the user premises, the distance that needs to be covered by the twisted-wire pairs is reduced. Table 3.5 in Chapter 3 shows that at sufficiently short distances it is then possible to deliver bit rates in the tens of megabits/second to the user premises. In the future this deployment of optical fiber is likely to be one of the approaches that will be used to provide much higher bit rates to the user premises.

In addition to handling the analog telephone traffic from its serving areas, a local telephone switching office must handle a multiplicity of other types of transmission lines. These include digital lines of various speeds from customer premises, private lines, foreign exchange lines, lines from cellular telephone networks, and high-speed digital transmission lines to the backbone of the network. A digital cross-connect system is used to organize the various types of lines that arrive and depart from a telephone office. A **digital cross-connect (DCC)** system is simply a digital time-division switch that is used to manage the longer-term flows in a network. DCCs typically deal with signals at DS1 and DS3 rates. Unlike the time-division switches that deal with individual telephone calls, DCC systems are not controlled by the signaling process associated with call setup. Instead they are controlled by the network operator to meet network configuration requirements.

Figure 4.47 shows how a DCC system can be used to interconnect various types of traffic flows in a telephone central office. The local analog telephone signals on the left side are digitized and then input into the DCC system. The other traffic on the left

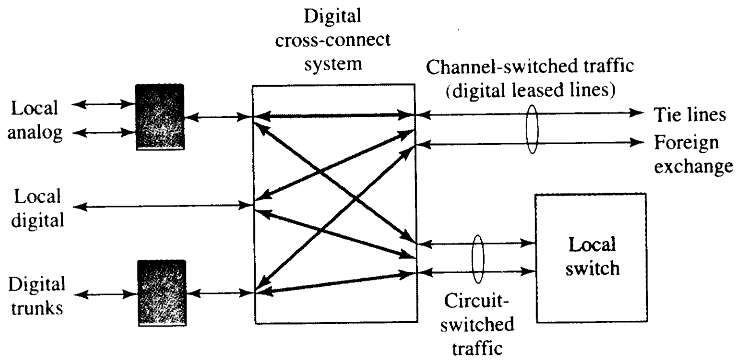


FIGURE 4.47 Digital cross-connect system.

side are local digital traffic and digital trunks connected to other offices. Some of these inputs are permanently connected as "tie lines" that interconnect customer sites; some of the other inputs may originate in other central offices and be permanently connected to foreign exchange lines that access an international gateway. Voice traffic that needs to be switched is routed from the input lines to a digital switch that is attached to the DCC system. The DCC system provides a flexible means for managing these connections on a semipermanent basis.

The transmission facilities between switches in the telephone network are becoming dominated by SONET-based optical transmission equipment using linear and ring topologies as discussed earlier in Section 4.2. DCC systems can be combined with SONET transmission systems to configure the topology "seen" by the telephone switches during call setup. As an example, in Figure 4.48 we show a SONET network that has a "stringy" physical topology. The SONET tributaries can be configured using add-drop multiplexers and a DCC system so tributaries interconnect the switches to produce the topology shown in part (b). In particular note that the maximum number of hops between any pair of switches is two hops instead of the four hops in the physical

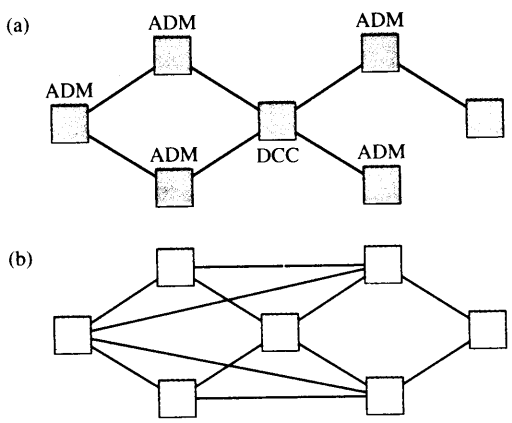


FIGURE 4.48 Digital cross-connect and SONET: (a) a physical SONET topology using ADMs and DCC systems; (b) a logical topology as seen by switches connected to ADMs.

topology. One benefit of decreasing the number of hops between switches is to simplify the routing procedure that is implemented by the switches.

DCC and SONET equipment can also be combined to provide network recovery from faults. For example, standby tributaries may be set up to protect against failures. Should a fault occur, the standby tributaries are activated to restore the original logical topology seen by the switches. This approach reduces the actions that the network switches must take in response to the failure.

4.5.2 End-to-End Digital Services

Since the inception of telephony, the telephone user has been connected to the telephone office by a pair of copper wires in the local loop. This last mile of the network has remained analog even as the backbone of the network was converted to all-digital transmission and switching. In the mid-1970s it became apparent that the demand for data services would necessitate making the telephone network entirely digital as well as capable of providing access to a wide range of services including voice and data. In the early 1980s the CCITT developed the **Integrated Services Digital Network (ISDN)** standards for providing end-to-end digital connectivity.

The ISDN standards define two *interfaces* between the user and the network as shown in Figure 4.49. The purpose of the interfaces is to provide *access* to various services that are possibly supported by different networks. The *basic rate interface (BRI)* provides the user with two 64 kbps bearer (B) channels and one 16 kbps data (D) channel. The *primary rate interface (PRI)* provides users with 23B+1D channels in North America and Japan and 30B+1D channels in Europe. The primary rate formats were clearly chosen to fit the T-1 and CEPT-1 frame formats in the digital multiplexing

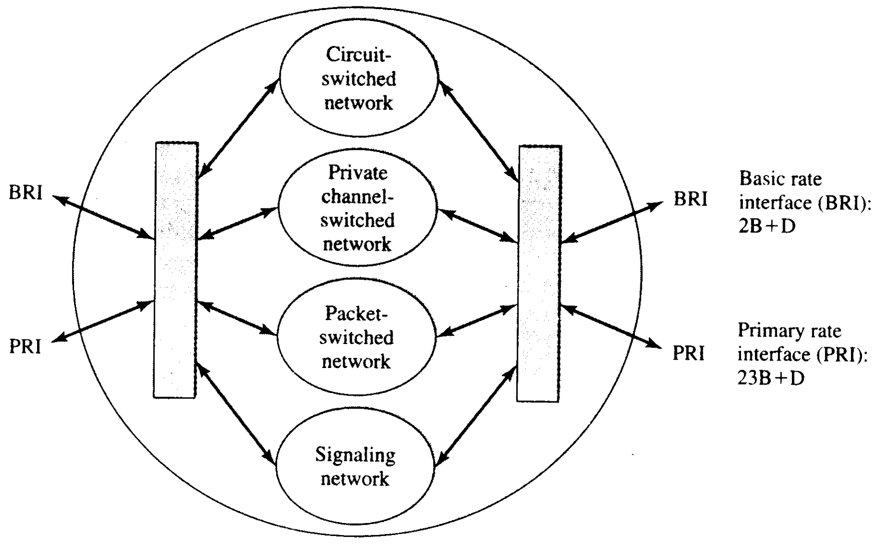


FIGURE 4.49 Integrated Services Digital Network.

HOW DOES THE OSI REFERENCE MODEL APPLY TO THE TELEPHONE NETWORK?

The telephone network was established well before the OSI reference model, and so there is not a perfect fit between the model and the telephone network. However, the ISDN standards do help to clarify this relationship. The telephone network (and circuit switching in general) require two basic functions: (1) signaling to establish and release the call and (2) end-to-end transmission to transfer the information between the users. ISDN views these two functions as separate and indeed as being provided by different networks (see Figure 4.49).

The signaling function is in fact a distributed computer application that involves the exchange of signaling messages between users and the network and between switches in the network to control the setup and release of calls. This process involves the use of all the layers in the OSI reference model. The set of protocols that implement signaling in the telephone network is said to constitute the *control plane*. We discuss these protocols in the next section.

The set of protocols that implement the transfer of information between users is called the *user plane*. In the case of an end-to-end 64 kbps connection, the control plane sets up the connection across the network. The user plane consists of an end-to-end 64 kbps flow and so involves only the physical layer. End users are free to use any higher-layer protocols on this digital stream.

This ISDN framework has been generalized so that the control plane is used to establish virtual connections in the user plane that involve more than just the physical layer. For example, the original ISDN standards defined a "packet mode" service in which an OSI layer-3 connection is set up across a packet-switching network. Subsequently a "frame relay mode" was defined to establish an end-to-end data-link layer connection between two terminals attached to a public network.

hierarchy. Each B channel is bidirectional and provides a 64 kbps end-to-end digital connection that can carry PCM voice or data. The primary function of the D channel is to carry signaling information for the B channels at the interface between the user and the network. Unlike traditional signaling between the user and the network, ISDN signaling is *out of band*, that is, the signaling messages are carried in a separate channel than the user information. The D channel can also be used to access a packet network. The D channel is 64 kbps in the primary rate interface. Additional H channels have been defined by combining B channels to provide rates of 384 kbps (H0), 1.536 Mbps (H11), and 1.920 Mbps (H12).

The basic rate interface was intended as a replacement for the basic telephone service. In North America the associated transmission format over twisted-wire pairs was selected to operate at 160 kbps and use the band that is occupied by traditional analog voice. The two B channels could provide two digital voice connections, and the D channel would provide signaling and access to a packet network. In practice, the 2B channels in the basic rate interface found use in digital videoconferencing applications and in providing access to Internet service providers at speeds higher than those available over conventional modems. The primary rate interface was intended for providing access from user premises equipment such as private branch exchanges (PBXs) and multiplexers to the network.[12]

ISDN was very slow in being adopted primarily because few services could use it. The growth of the Internet as a result of the World Wide Web stimulated the deployment of ISDN as an alternative to telephone modems. However, as discussed in Chapter 3, alternative ADSL transmission techniques for providing much higher speeds over the local loop are currently being introduced instead to meet the demand for high-speed Internet access. These techniques coexist with the regular analog telephone signal and do not interact with the telephone network.

As the ISDN standard was being completed, the interest in high-definition television and high-speed data interconnection prompted new work on a *broadband ISDN (BISDN)* standard. The very high bit rate required by these applications necessitated access speeds much higher than those provided by ISDN. The BISDN effort resulted in much more than an interface standard. An entirely new network architecture based on the connection-oriented transfer and switching of small fixed-length packets, known as asynchronous transfer mode (ATM), emerged as a target network for supporting a very wide range of services. ATM networks are discussed in detail later in Chapter 9.

4.6 SIGNALING

To set up a connection in a circuit-switching network a series of transmission lines, multiplexers, and switches must be configured so that information can flow in uninterrupted fashion from sender to receiver. Some form of signaling to coordinate the configuration is always required. In the very earliest telephone systems, end-to-end dedicated circuits were set up manually to connect each pair of users. The "signaling" here consisted of pieces of paper directing craftsmen to configure the appropriate connections. This approach is in use, even today, in configuring SONET paths where work orders direct the manual configuration of ADMs. In the next section, we discuss how the telephone network evolved to provide automated signaling. We also give an overview of areas where signaling is being introduced in new networks.

4.6.1 Signaling in the Telephone Network

The development of the telephone switchboard to establish connections dynamically between users fostered the development of new signaling procedures. Ringing sounds and flashing lights were incorporated into telephone sets to allow users to signal for a connection. An operator would then speak to the user and determine what connection was required. If more than one telephone office was involved, operators would speak

[12] A PBX is a switch in the user premises (i.e., an office) that provides connection services for intrapremise calls and that controls access to public and private network services.

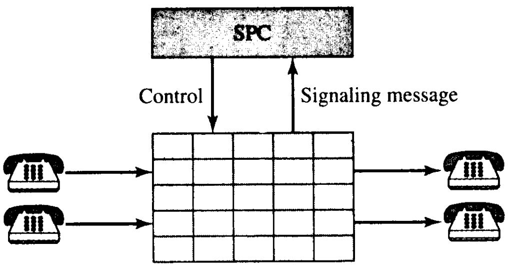


FIGURE 4.50 Stored-program control switches.

to each other to coordinate the establishment of a connection. For example, the earliest long-distance connections were established manually; a connection from San Francisco to New York would take 23 minutes to establish by hand.

Eventually various standard methods were developed to automate the establishment of connections. For example, the lifting of the handset in a traditional phone closes a circuit that allows current to flow from the telephone office to the telephone and back. This flow is used to signal a request for a call. In traditional phones, the turning of a dial generates a series of current pulses that can be counted to determine the desired telephone number. The electromechanical Strowger switch could be controlled by a sequence of these pulses to automatically establish a telephone circuit.

As shown in Figure 4.50, in traditional networks signaling information would arrive in telephone lines and be routed to the control system. Initially, hard-wired electromechanical or electronic logic was used to process these signaling messages. Later **stored-program control (SPC)** switches were introduced in which computers would control the setting up of connections in the switch, as shown in Figure 4.50. Through the intervention of the stored-program control computer, a request for a call would come in, a check would be made to see whether the destination was available, and if so, the appropriate connection would be made. The use of a program to control the switch provided great flexibility in modifying the control and in introducing new features.

As shown in Figure 4.51, setting up a call also required that the computers controlling the switches communicate with each other to exchange the signaling information.

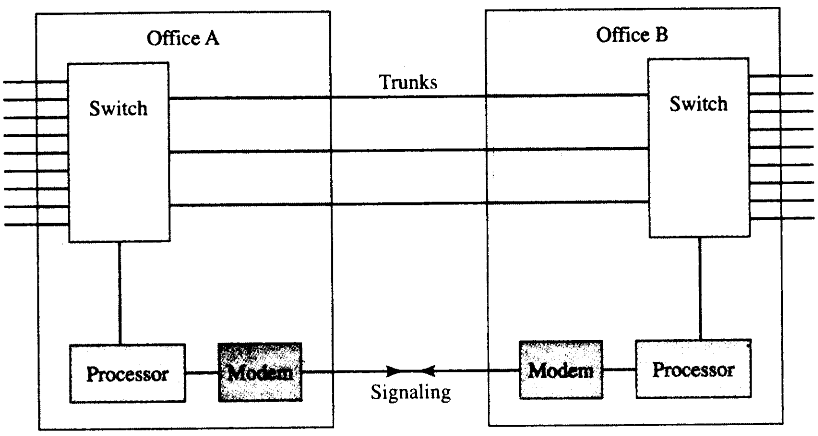
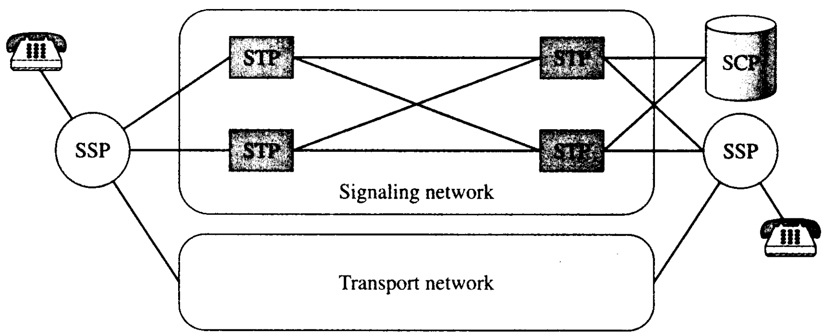


FIGURE 4.51 Common channel signaling.



SSP = Service switching point (signal to message)
STP = Signal transfer point (message transfer)
SCP = Service control point (processing)

FIGURE 4.52 Signaling network: Messages are transmitted in the signaling network to set up connections in the transport network.

A modem and separate communication lines were introduced to interconnect these computers. This situation eventually led to the introduction of a separate computer communications network to carry the signaling information.

Consider the operation of the **signaling network.** Its purpose is to implement connectivity between the computers that control the switches in the telephone network by providing for the exchange of messages. Figure 4.52 shows the telephone network as consisting of two parts: a signaling network that carries the information to control connections and a transport network that carries the user information. Communications from the user are split into two streams at the service switching point (SSP). The signaling information is directed toward the signaling network where it is routed and processed as required. The signaling system then issues commands to the switches to establish the desired connection. The signaling network functions much like the "nervous system" of the telephone network, directing the switches and communication lines in the network to be configured to handle the various connection requests. The second stream in the SSP consists of the user information that is directed to the transport network where it flows from one user to the other. Note that the signaling network does not extend to the user because of security concerns. Separate user-to-network signaling procedures are in place.

The function of the signaling network is to provide communications between the computers that control the switches. The computers communicate through the exchange of discrete messages. The best way of implementing such a network is through a *packet-switching network* that transfers information in the form of packets between network elements. The introduction of a packet-switching signaling network in digital telephony is important, since it is at this point that the evolution of the signaling network for telephony converges with the evolution of computer networks. In the next section we discuss the layered architecture of the telephone signaling system.

Because uninterrupted availability of telephone service is extremely important, *reliability* was built into the packet-switching network for signaling. The packet-switching

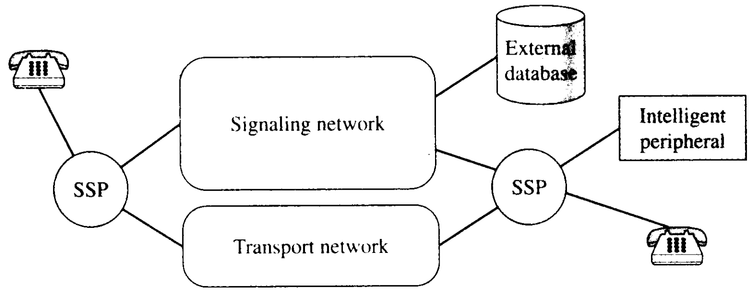


FIGURE 4.53 Intelligent network.

nodes (signal transfer points or STPs) are interconnected as shown in Figure 4.52. Any given region has two STPs that can reach any given office, so if one STP goes down the other is still available.

The processing of a connection request may involve going to *databases* and *special purpose processors* at the service control points (SCPs) in Figure 4.52. As signaling messages enter the network, they are routed to where a decision can be made or where information that is required can be retrieved. In the early 1970s telephone companies realized that the signaling network (and its computer control) could be used to enhance the basic telephone service. Credit-card calls, long-distance calls, 800 calls, and other services could all be implemented by using this more capable signaling network. In the case of credit-card calls, a recorded message could request the credit-card number. The digits would be collected, and a message would be sent to a database to check the credit-card number; if authorized, the call would then be set up.

Telephone companies use the term **intelligent network** to denote the use of an enhanced signaling network that provides a broad array of services. These services include identification of the calling person, screening out of certain callers, callback of previous callers, and voice mail, among others. As shown in Figure 4.53, the addition of new devices, "intelligent peripherals," to the intelligent network enables other new services. For example, one such device can provide voice recognition. When making a call, your voice message might be routed to this intelligent peripheral, which then decodes what you are saying and translates it into a set of actions that the signaling network has to perform in order to carry out your transaction.

Another service that intelligent networks provide is personal mobility. *Personal mobility* allows the user who subscribes to the service to have a personal ID. Calls to the user are not directed to a specific location in the network. Instead the network dynamically keeps track of where the user is at any given time and routes calls accordingly.

4.6.2 Signaling System #7 Architecture

From the above discussion we see that two classes of functions are involved in operating a telephone network. The first class of functions is associated with the transfer of the user's information and belongs to the *user* or *data plane*. The second class of functions

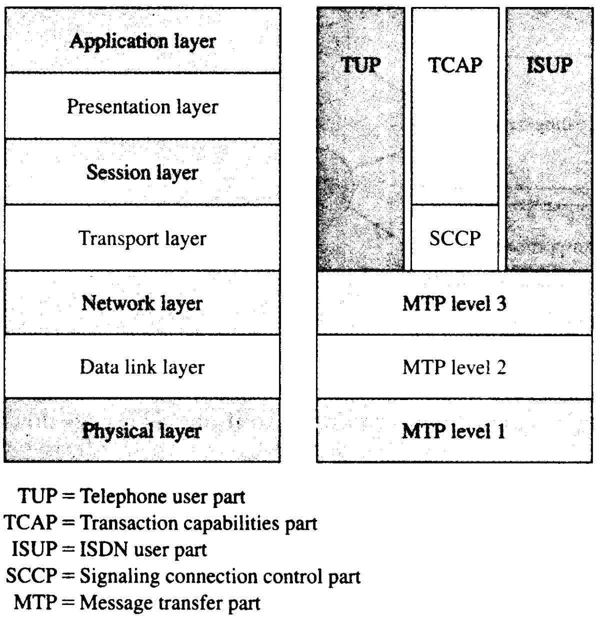


FIGURE 4.54 OSI reference model and SS7 network architecture.

is associated with the setting up, monitoring, and tearing down of connections and belongs to the *control plane*. In the telephone network the user plane is very simple; it consists of physical layer connections for the transfer of voice signals. The control plane of the telephone network is quite sophisticated and involves an entire packet-switching network, the Signaling System #7. The Signaling System #7 (SS7) network is a packet network that controls the setting up, managing, and releasing of telephone calls. The network also provides support for intelligent networks, mobile cellular networks, and ISDN. The SS7 network architecture is shown in Figure 4.54.

This architecture uses "parts" instead of "layers." The message transfer part (MTP) corresponds to the lower three layers of the OSI reference model. Level 1 of the MTP corresponds to the physical layer of the signaling links in the SS7 networks. Physical links have been defined for the following transmission speeds: E-1 (2.048 Mbps = 32 channels at 64 kbps each), T-1 (1.544 Mbps = 24 channels at 64 kbps each), V-35 (64 kbps), DS-0 (64 kbps), and DS-0A (56 kbps). These telephone transmission standards were discussed earlier in this chapter.

MTP level 2 ensures that messages are delivered reliably across a signaling link. This level corresponds to the data link layer in the OSI reference model. The MTP level 3 ensures that messages are delivered between signaling points across the SS7 network. Level 3 provides routing and congestion control that reroutes traffic away from failed links and signaling points.

The ISDN user part (ISUP) protocols perform the basic setup, management, and release of telephone calls. The telephone user part (TUP) is used instead in some countries.

The MTP addresses the signaling points, but is not capable of addressing the various applications that may reside within a signaling point. These applications include

800-call processing, calling-card processing, call management services, and other intelligent network services. The signaling connection control part (SCCP) allows these applications to be addressed by building on the MTP to provide connectionless and connection-oriented service. The SCCP can also translate "global titles" (e.g., a dialed 800 number or a mobile subscriber number) into an application identifier at a destination signaling point. This feature of SCCP is similar to the Internet DNS and assists in the routing of messages to the appropriate processing point.

The transaction capabilities part (TCAP) defines the messages and protocols that are used to communicate between applications that use the SS7 network. The TCAP uses the connectionless service provided by the SCCP to support database queries that are used in intelligent networks.

TRENDS IN SIGNALING

The notions of signaling and control plane were developed around the telephone network, but are currently being adapted and evolved in the context of the Internet. Recent work on Internet protocols has developed *generic* signaling protocols that can be used to allocate transmission and switching resources to given flows, that is, connections, along a path across a network. The RSVP protocol was developed for the establishment of packet connections with service guarantees across an Internet, but the protocol has been extended so that it can be used to set up SONET paths across a SONET network as well as lightpaths across an optical network. We discuss this type of signaling in Chapter 10.

A second type of signaling involves the exchange of messages between end systems prior to the establishment of a connection. The Session Initiation Protocol, also discussed in Chapter 10, is an application protocol that can be used to establish, modify, and terminate multimedia sessions involving two or more participants. SIP can provide the rich set of features of the telephone system, but supports an even richer and more versatile set of connection types.

◆ 4.7 TRAFFIC AND OVERLOAD CONTROL IN TELEPHONE NETWORKS

In this section we consider the *dynamic* aspects of multiplexing the information flows from various users into shared high-speed digital transmission lines. We begin by looking at the problem of concentration, which involves the sharing of a number of trunks by a set of users. Here we examine the problem of ensuring that there are sufficient resources, namely, trunks, to provide high availability, that is, low probability of blocking. We find that concentration can lead to very efficient usage of network resources if the volume of traffic is sufficiently large. We next discuss how this result influences routing methods in circuit-switching networks. Finally we consider approaches for dealing with overload conditions.

4.7.1 Concentration

Telephone networks are engineered so that, under normal circumstances, there is very little chance that a request for a connection cannot be met. In fact telephone networks make extensive use of concentration to make efficient use of their transmission facilities and to turn a healthy profit from offering telephone service. In this section we explain the statistical basis for the "magic" behind concentration.

Concentration addresses the situation where a large number of users alternate between periods when they need connections and periods when they are idle. In this situation it does not make sense to assign channels to all users all the time. Concentration involves the dynamic sharing of a number of communication channels among a *larger* community of users.

In Figure 4.55 numerous users at a given site, each connected to a local concentrator, share expensive trunks provided by a high-speed digital transmission line to connect to another location, for example, a telephone central office or another user site. When a user on one end wishes to communicate with a user at the other end, the concentrator assigns a communication line or *trunk* for the duration of the call. When the call is completed, the transmission line is returned to the pool that is available to meet new connection requests. Note that signaling between concentrators is required to set up and terminate each call. The number of trunks in use varies randomly over time but is typically much smaller than the total number of lines. We say that a connection request is *blocked* when no trunks are available. The objective here is to minimize the number of trunks, while keeping the probability of blocking to some specified level.

Figure 4.56 shows the occupancy of a set of seven trunks over time. The shaded rectangles indicate periods when a given trunk is in use. The upper part of the figure shows the corresponding $N(t)$, the number of trunks in use at time t. In this example the system is in a blocking state when $N(t) = 7$.

The users require trunk connections in a sporadic and unscheduled manner. Nevertheless, the statistical behavior of the users can be characterized. In particular it has been found that the users *as a group* make requests for connections according to a Poisson process with connection request or **arrival rate** λ calls/second. A Poisson process is characterized by the following two properties:

1. In a very small time interval of duration Δ seconds, only two things can happen: There is a request for one call, with probability $\lambda\Delta$, or there are no requests for calls, with probability $1 - \lambda\Delta$.
2. The arrivals of connection requests in different intervals are statistically independent.

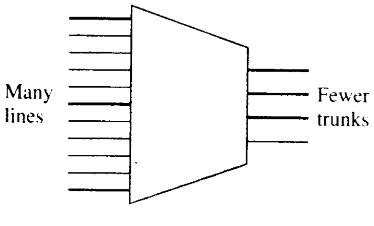


FIGURE 4.55 Concentration (bold lines indicate lines/trunks that are in use).

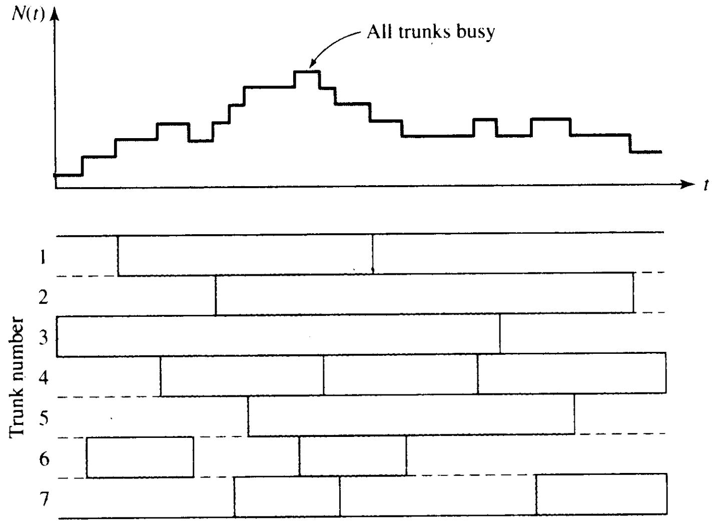


FIGURE 4.56 Number of trunks in use as a function of time.

The first property states that the rate at which call requests are made is uniform over time. For example, it does not depend on the number of users already connected. The second property states that the pattern of prior requests does not influence the likelihood of a request occurring in a given time interval. Both of these properties are reasonable when dealing with the aggregate behavior of a large number of people. The analysis of the trunk concentration problem is carried out in Appendix A. We only present the results of the analysis here.

The time that a user maintains a connection is called the **holding time.** In general, the holding time X is a random variable. The average holding time $E[X]$ can be viewed as the amount of "work" that the transmission system has to do for a typical user. In telephone systems typical conversations have a mean holding time of several minutes. The **offered load** $\boldsymbol{a}$ is defined as the total rate at which work is offered by the community of users to the multiplexing system, as measured in Erlangs:

$$a = \lambda \times E[X] \qquad \text{(Erlangs)} \tag{4.5}$$

One Erlang corresponds to an offered load that would occupy a single trunk 100 percent of the time, for example, an arrival rate of $\lambda = 1$ call/second and a call holding time of $E[X] = 1$ second/call would occupy a single trunk all of the time. Typically telephone systems are designed to provide a certain level or "grade" of service during the *busy hour* of the day. Measurements of call attempts reveal clear patterns of activity and relatively stable patterns of call attempts. For example, in office settings the call rate peaks in mid-morning, drops during lunch hour, and peaks again in mid-afternoon. In the subsequent discussion the offered load should be interpreted as the load during the busy hour.

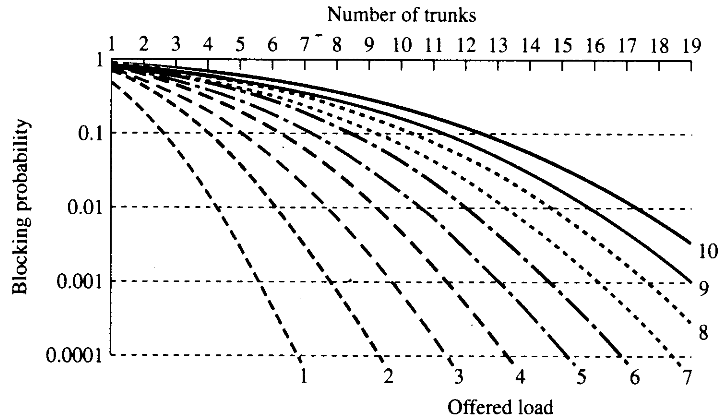


FIGURE 4.57 Blocking probability for offered load versus number of trunks: Blocking probability decreases with the number of trunks.

The blocking probability P_b for a system with c trunks and offered load a is given by the **Erlang B formula:**

$$P_b = B(c, a) = \frac{a^c/c!}{\sum_{k=0}^{c} a^k/k!} \tag{4.6}$$

where $k! = 1 \times 2 \times 3 \ldots \times (k-1) \times k$.

Figure 4.57 shows the blocking probability for various offered loads as the number of trunks c is increased. As expected, the blocking probability decreases with the number of trunks. A 1% blocking probability is typical in the design of trunk systems. Thus from the figure we can see that four trunks are required to achieve this P_b requirement when the offered load is one Erlang. On the other hand, only 16 trunks are required for an offered load of nine Erlangs. This result shows that the system becomes more efficient as the size of the system increases, in terms of offered load. The efficiency can be measured by trunk **utilization** that is defined as the average number of trunks in use divided by the total number of trunks. The utilization is given by

$$\text{Utilization} = \lambda(1 - P_b)E[X]/c = (1 - P_b)a/c \tag{4.7}$$

Table 4.2 shows the trunk utilization for the various offered loads and $P_b = 0.01$. Note that for small loads the utilization is relatively low. In this case extra trunks are required to deal with surges in connection requests. However, the utilization increases as the size of the systems increases in terms of offered load. For a load of two Erlangs, a total of 7 trunks is required; however, if the load is tripled to six Erlangs, the number of trunks required, 13, is less than double. The entry in Table 4.2 for offered loads of 50 and 100 Erlangs shows that high utilization is possible when the offered loads are large. These examples demonstrate how the sharing of network resources becomes more efficient as the scale or size of the system increases. The improvement in system performance that results from aggregating traffic flow is called *multiplexing gain*.

Telephone networks have traditionally been engineered for probability of blocking of 1% during the busy hour of the day. The actual call request rates at other times of

TABLE 4.2 Trunk utilization.

Load	Trunks @ 1%	Utilization
1	5	0.20
2	7	0.29
3	8	0.38
4	10	0.40
5	11	0.45
6	13	0.46
7	14	0.50
8	15	0.53
9	17	0.53
10	18	0.56
30	42	0.71
50	64	0.78
60	75	0.80
90	106	0.85
100	117	0.85

the day can be much lower, so the probability of blocking is even lower. In fact, users have come to expect that the network is always available to complete a connection.

4.7.2 Routing Control

Routing control refers to the procedures for assigning paths in a network to connections. Clearly, connections should follow the most direct route, since this approach uses the fewest network resources. However, we saw in Section 4.7.1 that when traffic flows are not large the required set of resources to provide high availability, that is, a blocking probability of 1%, will be used inefficiently. Economic considerations lead to an approach that provides direct trunks between switches that have large traffic flows between them and that provide indirect paths through tandem switches for smaller flows.

A hierarchical approach to routing is desirable when the volume of traffic between switches is small. This approach entails aggregating traffic flows onto paths that are shared by multiple switches. Consider the situation in Figure 4.58 where switches A, B, and C have 10 Erlangs of traffic to D, E, and F. Suppose that switches A, B, and C are close to each other and that they have access to tandem switch 1. Similarly, suppose

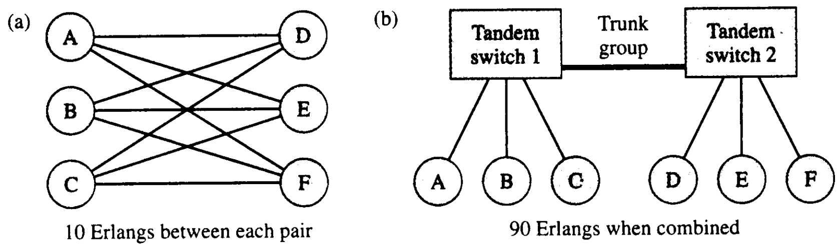


FIGURE 4.58 Hierarchical routing control.

that switches D, E, and F are close to each other and have access to tandem switch 2. Furthermore, suppose that the distances between switches A, B, and C and D, E, and F are large. From Table 4.2, each pair of switches requires 18 long distance trunks to handle the 10 Erlangs of traffic at 1% blocking probability. Thus the approach in Figure 4.58a requires $9 \times 18 = 162$ trunks. Concentrating the traffic flows through the tandems reduces to 106 the number of trunks required to handle the combined 90 Erlangs of traffic. The second approach does require the use of local trunks to the tandem switch, and so the choice depends on the relative costs of local and long-distance trunks.

The increase in efficiency of trunk utilization that results from the increased offered load introduces a sensitivity problem. The higher efficiency implies that a smaller number of spare circuits is required to meet the 1% blocking probability. However, the smaller number of spare circuits makes the system more sensitive to traffic overload conditions. For example, if each trunk group in part (a) is subjected to an overload of 10% the resulting offered load of 11 Erlangs on the 17 trunks results in an increased blocking probability of 2.45%. On the other hand, the 10% overload on the trunk group in part (b) results in an offered load of 99 Erlangs to 106 trunks. The blocking probability of the system increases dramatically to 9.5%. In other words, the blocking probability for large systems is quite sensitive to traffic overloads, and therefore the selection of the trunk groups must provide a margin for some percentage of overload.

Figure 4.59 shows a typical approach for routing connections between two switches that have a significant volume of traffic between them. A set of trunks is provided to directly connect these two switches. A request for a connection between the two switches first attempts to engage a trunk in the direct path. If no trunk is available in the direct path, then an attempt is made to secure an alternative path through the tandem switch. The number of trunks in the direct route is selected to have a high usage and hence a blocking probability higher than 1% say, 10%. The number of trunks available in the alternative route needs to be selected so that the overall blocking probability is 1%. Note that because only 10% of the traffic between the switches attempts the alternative route, a 10% blocking probability on the alternative path is sufficient to bring the overall blocking probability to 1%. It should be noted also that the Erlang formula cannot be applied directly in the calculation of blocking probability on the alternative route. The reason is that the requests for routes to the tandem switch arrive only during periods when the high-usage route is unavailable. The method for handling this case is discussed in [Cooper 1981].

Figure 4.60 shows a more realistic scenario where the tandem switch handles overflow traffic from some high-usage trunk groups and direct traffic between switches that have small volumes of traffic between them. Note that in this case the traffic

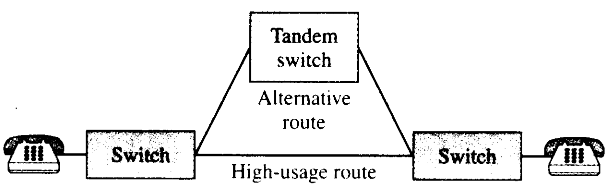


FIGURE 4.59 Alternative routing.

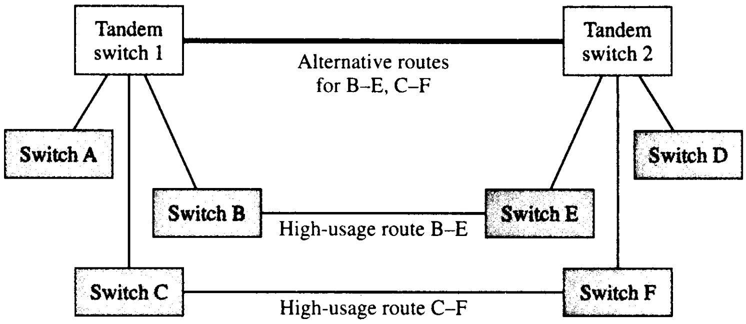


FIGURE 4.60 Typical routing scenario.

between switches A and D must be provided with a 1% blocking probability, while a 10% blocking probability is sufficient for the other pairs of switches. To achieve this blocking probability the traffic from A to D must receive a certain degree of preferential access to the trunks between the tandem switches.

Traffic flows vary according to the time of day, the day of the week, and even the time of year. The ability to determine the state of network links and switches provides an opportunity to assign routes in more dynamic fashion. For example, time/day differences between the East Coast and the West Coast in North America allow the network resources at one coast to provide alternative routes for traffic on the other coast during certain times of the day. *Dynamic nonhierarchical routing (DNHR)* is an example of this type of dynamic approach to routing calls. As shown in Figure 4.61, the first route attempt between two switches consists of a direct route. A certain number of tandem switches is capable of providing a two-hop alternative route. The order in which tandem switches are attempted as alternative routes is determined dynamically according to the state of the network. The AT&T long-distance network consists of approximately 100 switches almost interconnected entirely with direct links. This topology allows the use of DNHR [Carne 1995].

4.7.3 Overload Controls

Traffic and routing control are concerned with the handling of traffic flows during normal predictable network conditions. **Overload control** addresses the handling of traffic flows during unexpected or unusual conditions, such as occur during holidays

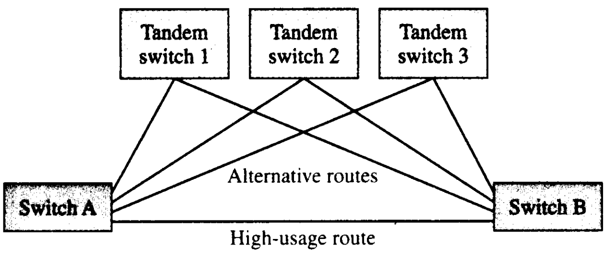


FIGURE 4.61 Dynamic nonhierarchical routing: If the high-usage route between two switches is unavailable, several alternative routes may be used as needed.

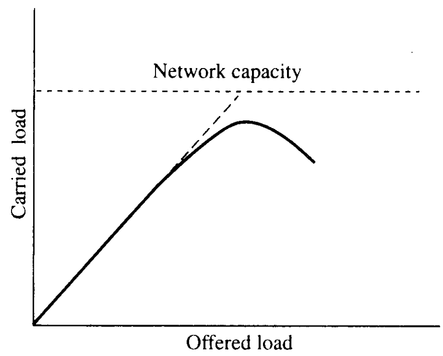


FIGURE 4.62 Traffic overload: As the offered load reaches network capacity, the carried load decreases.

(Christmas, New Year's Day, and Mother's Day), catastrophes (e.g., earthquakes), or equipment failures (e.g., a fire in a key switch or a cut in a key large-capacity optical fiber).

Overload conditions result in traffic levels that the network equipment has not been provisioned for and if not handled properly can result in a degradation in the level of service offered to all network customers. The situation can be visualized as shown in Figure 4.62. Under normal conditions the traffic carried by the network increases or decreases with the traffic that is offered to it. As the offered traffic approaches network capacity, the carried traffic may begin to fall. The reason for this situation is that as network resources become scarce, many call attempts manage to seize only some of the resources they need and ultimately end up uncompleted. One purpose of overload control is to ensure that a maximum number of calls are completed so that the carried load can approach the network capacity under overload conditions.

Network monitoring is required to identify overload conditions. Clearly the traffic loads at various links and switches need to be measured and tracked. In addition, the success ratio of call attempts to a given destination also needs to be monitored. The answer/bid ratio measures this parameter. The traffic load measurement in combination with the answer/bid ratio is useful in diagnosing fault conditions. For example, the failure of switch A will result in an increased traffic level at other switches due to reattempts from callers to switch A. The increased traffic load indicates a problem condition but is not sufficient to identify the problem. The answer/bid ratio provides the information that identifies switch A as the location of the problem. Network monitoring software is used to process alarms that are set by the monitoring system to diagnose problems in the network.

Once an overload condition has been identified, several types of actions can be taken, depending on the nature of the problem. One type of overload control addresses problems by allocating additional resources. Many transmission systems include backup redundant capacity that can be activated in response to failures. For example, SONET transmission systems use a ring topology of add-drop multiplexers to provide two paths between any two stations on the ring. Additional redundancy can be provided by interconnecting SONET rings using DCCs. Dynamic alternative routing provides another approach for allocating resources between areas experiencing high levels of traffic.

Certain overload conditions cannot be addressed by allocation of additional resources. The overload controls in this case act to maximize the efficiency with which the available resources are utilized. For example, in the case of networkwide congestion the routing procedures could be modified so that all call attempts, if accepted, are met using direct routes. Alternative routes are disallowed because they require more resources to complete calls. As a result, the traffic carried by the network is maximized.

Another overload condition occurs when a certain area experiences extreme levels of inbound and outbound traffic as may result from the occurrence of a natural disaster. A number of overload controls have been devised to deal with this situation. One approach involves allowing only outbound traffic to seize the available trunks. This approach relieves the switches in the affected area from having to process incoming requests for calls while allowing a maximum of outbound calls to be completed. A complementary control involves code blocking, where distant switches are instructed to block calls destined for the affected area. A less extreme measure is to pace the rate at which call requests from distant switches to the affected area are allowed to proceed.

It should be noted that all of the above overload controls make extensive use of the signaling system. This dependence on the signaling system is another potential source of serious problem conditions. For example, faulty signaling software can result in abnormal levels of signaling traffic that in turn can incapacitate the network. Clearly, overload control mechanisms are also essential for the signaling system.

4.8 CELLULAR TELEPHONE NETWORKS

The first generations of cellular telephone networks extend the basic telephone service to *mobile* users with *portable* telephones. Unlike conventional telephone service where the call to a telephone number is directed to a specific line that is connected to a specific switch, in cellular telephony the telephone number specifies a specific subscriber's mobile station (telephone). Much of the complexity in cellular telephony results from the need to track the location of the mobile station. In this section we discuss how radio transmission systems and the telephone network infrastructure are organized to make this service possible.

Radio telephony was first demonstrated in the early 1900s when an analog voice signal was modulated onto a radio wave. Because electromagnetic waves propagate over a wide geographical area, they are ideally suited for a *radio broadcasting service* where information from a source or station is transmitted to a community of receivers that is within range of the signal. The economics of this type of communication dictate that the cost can be high for the station equipment but that the cost of the receivers must be low so that the service can become available to a large number of users. Commercial broadcast radio was introduced in the early 1920s, and within a few years the service was used by millions of homes.

The introduction of commercial radio resulted in intense competition for frequency bands. The signals from different stations that use the same frequency band will interfere with each other, and neither signal will be received clearly. A limited number of frequencies are available, so in the 1930s it became clear the *regulation* was needed to

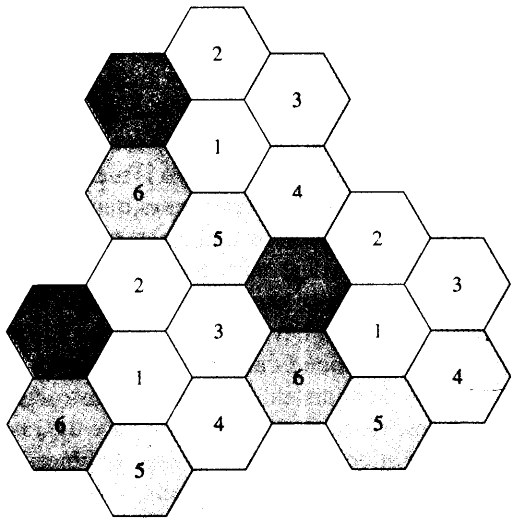


FIGURE 4.63 Cellular network structure.

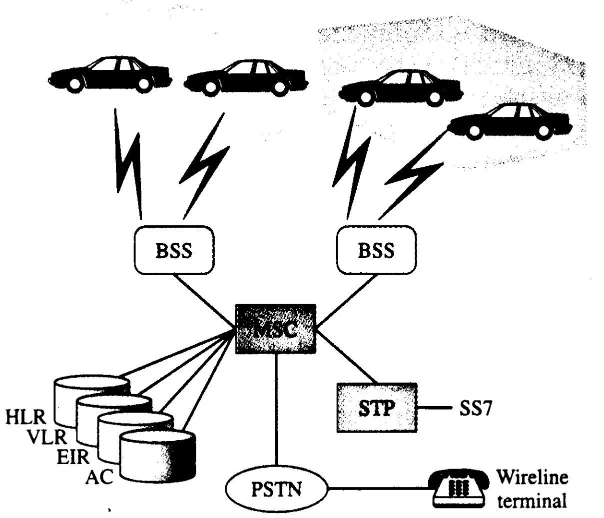
control the use of frequency bands. Governments established agencies responsible for determining the use and the allocation of frequency bands to various users.

Radio transmission makes communications possible to *mobile* users. Early mobile radio telephone systems used a radio antenna installed on a hill and equipped with a high-power multichannel transmitter. Transmission from the mobile users to the antenna made use of the power supplied by the car battery. These systems provided communications for police, taxis, and ambulance services. The limited amount of available bandwidth restricted the number of calls that could be supported and hence the number of subscribers that could use such systems was limited. For example, in the late 1940s, the mobile telephone service for the entire New York City could only support 543 users [CSTB 1997].

The scarcity of the radio frequency bands that are available and the high demand for their use make frequency spectrum a precious resource. Transmission of a radio signal at a certain power level results in a coverage area composed of the region where the signal power remains significant. By reducing the power level, the coverage area can be reduced and the frequency band can then be reused in nearby adjacent areas. This **frequency-reuse** principle forms the basis for cellular radio communications.

In **cellular radio communications**, a region, for example, a city, is divided into a number of geographical areas called **cells** and users within a cell communicate using a band of frequencies.[13] Figure 4.63 shows how a region can be partitioned in a honeycomb pattern using hexagonal cells. Cell areas are established based on the density of subscribers. Large cells are used in rural areas, and small cells are used in urban areas. As shown in Figure 4.64, a **base station** is placed near the center of each cell. The base station has an antenna that is used to communicate with mobile users in its vicinity.

[13]Unfortunately, the term *cell* appears in two separate and unrelated areas in networks. The geographic "cells" in a cellular network have nothing to do with the fixed-packet "cells" that are found in ATM networks. The context is usually sufficient to determine what kind of cell is being discussed.



AC = Authentication center
BSS = Base station subsystem
EIR = Equipment identity register
HLR = Home location register
MSC = Mobile switching center
PSTN = Public switched telephone network
STP = Signal transfer point
VLR = Visitor location register

FIGURE 4.64 Components of a cellular network.

Each base station has a number of *forward channels* available to transmit to its mobile users and an equal number of *reverse channels* to receive from its mobile users.[14]

Base stations are connected by a wireline transmission link or by point-to-point microwave radio to a telephone switch, called the *mobile switching center (MSC)*, which is sometimes also called the *mobile telephone switching office (MTSO)*. The MSC handles connections between cells as well as to the public switched telephone network. As a mobile user moves from one cell to another, a **handoff** procedure is carried out that transfers the connection from one base station to the other, allowing the call to continue without interruption.

In general, immediately adjacent cells cannot use the same set of frequency channels because doing so may result in interference in transmissions to users near their boundary.[15] The set of available radio channels are reused following the *frequency-reuse pattern*. For example, Figure 4.63 shows a seven-cell reuse pattern in which seven disjoint sets of frequency channels are reused as shown. This pattern introduces a minimum distance of one cell between cells using the same frequency channels. Other reuse patterns have reuse factors of 4 and 12. As traffic demand grows, additional capacity can be provided by splitting a cell into several smaller cells.

As an example consider the *Advanced Mobile Phone Service (AMPS)*, which is an analog cellular system still in use in North America. In this system the frequency

[14]The channels are created by using frequency-division multiple access (FDMA), time-division multiple access (TDMA), or code-division multiple access (CDMA). These techniques are explained in Chapter 6.

[15]An exception is CDMA, which allows the same "code division" channel to be reused in adjacent cells.

band 824 to 849 MHz is allocated to transmissions from the mobile station to the base station, and the band 869 to 894 MHz is allocated to transmissions from the base station to the mobile station. AMPS uses a 30 kHz channel to carry one voice signal, so the total number of channels available in each direction is 25 MHz/30 kHz = 832 channels. The bands are divided equally between two independent service providers, so each cellular network has 416 bidirectional channels. Each forward and reverse channel pair has frequency assignments that are separated by 45 MHz. This separation between transmit and receive channels reduces the interference between the transmitted signal and the received signal.

A small number of channels within each cell have been designated to function as **setup channels**. For example, the AMPS system allocates 21 channels for this purpose. These channels are used in the setting up and handing off of calls as follows. When a mobile user turns on his or her unit, the unit scans the setup channels and selects the one with the strongest signal. Henceforth it monitors this setup channel as long as the signal remains above a certain threshold. To establish a call from the public telephone network or from another mobile user to a mobile user, the MSC sends the call request to all of its base stations, which in turn broadcast the request in all the forward setup channels, specifying the mobile user's telephone number. When the desired mobile station receives the request message, it replies by identifying itself on a reverse setup channel. The corresponding base station forwards the reply to the MSC and assigns a forward and reverse voice channel. The base station instructs the mobile station to begin using these channels, and the mobile telephone is rung.

To initiate a call, the mobile station sends a request in the reverse setup channel. In addition to its phone number and the destination phone number, the mobile station also transmits a serial number and possible password information that is used by the MSC to validate the request. This call setup involves consulting the *home location register*, which is a database that contains information about subscribers for which this is the home area. The validation involves the *authentication center*, which contains authentication information about subscribers. The MSC then establishes the call to the public telephone network by using conventional telephone signaling, and the base station and mobile station are moved to the assigned forward and reverse voice channels.

As the call proceeds, the signal level is monitored by the base station. If the signal level falls below a specified threshold, the MSC is notified and the mobile station is instructed to transmit on the setup channel. All base stations in the vicinity are instructed to monitor the strength of the signal level in the prescribed setup channel. The MSC uses this information to determine the best cell to which the call should be handed off. The current base station and the mobile station are instructed to prepare for a handoff. The MSC then releases its connection to the first base station and establishes a connection to the new base station. The mobile station changes its channels to those selected in the new cell. The connection is interrupted for the brief period that is required to execute the handoff.[16]

---

[16]Again CDMA systems differ in that they carry out a "soft" handoff that uses a "make before break" connection approach.

When **roaming users** enter an area outside their home region, special procedures are required to provide the cellular phone service. First, business arrangements must be in place between the home and visited cellular service providers. To automatically provide roaming service, a series of interactions is required between the home network and the visited network, using the telephone signaling system. When the roamer enters a new area, the roamer registers in the area by using the setup channels. The MSC in the new area uses the information provided by the roamer to request authorization from the roamer's home location register. The *visitor location register* contains information about visiting subscribers. After registering, the roamer can receive and place calls inside the new area.

Two sets of standards have been developed for the signaling required to support cellular telephone service. The **Global System for Mobile Communications (GSM)** signaling was developed as part of a standard for pan-European public land mobile system. The **Interim Standard 41 (IS-41)** was developed later in North America, using much of the GSM framework. In the following section we describe the protocol layered architecture of GSM.

In the GSM system the *base station subsystem (BSS)* consists of the *base transceiver station (BTS)* and the *base station controller (BSC)*. The BTS consists of the antenna and transceiver to communicate with the mobile telephone. The BTS is also concerned with the measurement of signal strength. The BSC manages the radio resources of one or more BTSs. The BSC is concerned with the setup of frequency channels as well as with the handling of handoffs. Each BTS communicates with the mobile switching center through the BSC, which provides an interface between the radio segment and the switching segment as shown in Figure 4.65. The subsystems must exchange signaling messages to coordinate their activities. The GSM signaling stack was developed for this purpose.

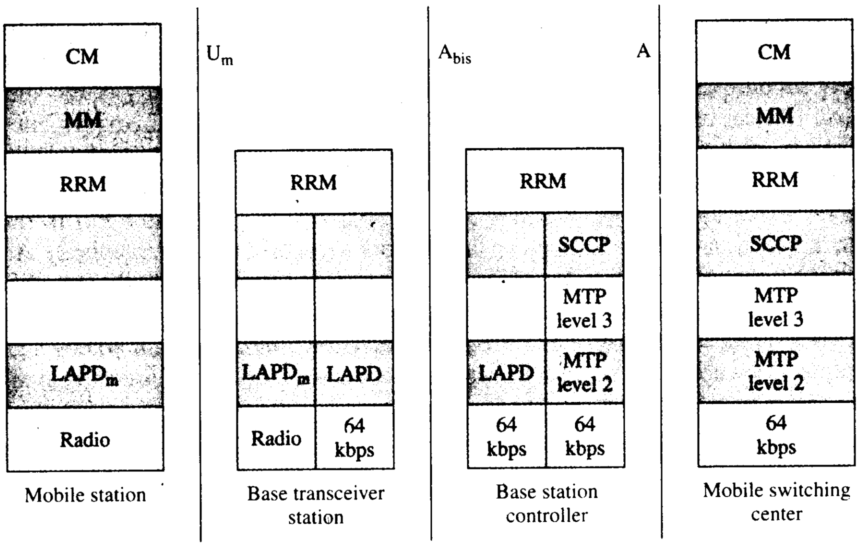


FIGURE 4.65 Protocol stacks in the cellular network.

The GSM signaling protocol stack has three layers as shown in Figure 4.65. Layer 1 corresponds to the physical layer, and layer 2 to the data link layer. GSM layer 3 corresponds to the application layer and is divided into three sublayers: radio resources management (RRM), mobility management (MM), and call management (CM). Different subsets of these layers/sublayers are present in different elements in the GSM network. We discuss these stacks proceeding from the mobile station to the MSC in Figure 4.65.

The radio air interface between the mobile station and the BTS is denoted as U_m. The physical layer across the U_m interface is provided by the radio transmission system. The LAPD protocol is a data link protocol that is part of the ISDN protocol stack and is similar to asynchronous balanced mode in HDLC discussed in Chapter 5. $LAPD_m$ denotes a "mobile" version of LAPD. The radio resources management sublayer between the mobile station and the BTS deals with setting up the radio channels and with handover (the GSM term for handoff).

The interface between the BTS and its BSC is denoted as the A_{bis} interface. The physical layer consists of a 64 kbps link with LAPD providing the data link layer. A BSC can handle a handover if the handover involves two cells under its control. This approach relieves the MSC of some processing load. The interface between the BSC and the MSC is denoted as the A interface that uses the protocol stack of SS7. The RRM sublayer in the MSC is involved in handovers between cells that belong to different BSCs but that are under the control of the MSC.

Only the mobile station and the MSC are involved in the mobility management and call management sublayers. Mobility management deals with the procedures to locate mobile stations so that calls can be completed. In GSM, cells are grouped into location areas. A mobile station is required to send update messages to notify the system when it is moving between location areas. When a call arrives for a mobile station, it is paged in all cells in the current location area. The MSC, the HLR, and the VLR are involved in the procedures for updating location and for routing incoming calls. The mobility sublayer also deals with the authentication of users. In GSM, unlike other standards, the mobile station includes a smart card, called the Subscriber Identity Module (SIM), which identifies the subscriber, independently of the specific terminal device, and provides a secret authorization key. The call management sublayer deals with the establishment and release of calls. It is based on the signaling procedures of ISDN with modifications to deal with the routing of calls to mobile users.

In this section we have focused on the network aspects of cellular telephony. A very important aspect of cellular networks is the access technique that is used to provide the channels that are used for individual connections. Here we have discussed an example of the AMPS standard that is based on frequency division multiplexing of analog signals. In Chapter 6 we consider the various types of access techniques that are used in digital cellular telephone networks.

WHAT IS PERSONAL COMMUNICATIONS?

You have probably heard of Personal Communication Services (PCS) phones and of Personal Digital Assistants (PDAs), two types of "personal" devices. PCS indicates the class of digital cellular telephone service that has become available in the 1850–1990 MHz band in North America. The *personal* in PCS attempts to contrast the *portability* of the service from that provided by the early analog cellular telephone service that was initially designed for car phones. The PCS user can carry the phone and hence enjoys greater *mobility*. PDA denotes a small, portable, handheld computer that can be used for personal and business information, such as electronic schedules and calendars, address and phone books, and with the incorporation of modems, e-mail transmission and retrieval. The "personal" in PDA denotes the *private* nature of the use of the device. Therefore, we find that personal communications involves *terminal portability, personal mobility, communications in a variety of modes (voice, data, fax, and so on) and personal profile/customization/privacy.*

What's involved in providing personal communications in the broad sense of the term? An essential component of personal communications is terminal portability, that is, the ability to carry the phone/device with you, and this feature is clearly provided by radio communications and cellular technology. However, this is not all. Personal mobility, the ability to access the network at any point possible by using different terminal devices, is provided by intelligence built into a network. In the case of telephone networks, the intelligence is provided by the signaling system. In the case of other networks, for example, the Internet, this intelligence may be provided in other ways. For example, it is now possible for a person to read or send e-mail by going to his or her ISP's web page from any computer in the world. In any case, personal mobility implies that it is the *individual* who accesses the communication service, not the terminal. Thus there is a need to provide the individual with the *universal personal ID*. Such an ID could be a form of telephone number or address or it could be provided by a smart card such as in GSM. The mode of communications would depend on the capabilities of the terminal device and access network that is in use at a particular time. It would also depend on the personal profile of the individual that would describe a personalized set of communications and other services to which the individual has subscribed. The personal profile could be stored in a database akin to the home location register in cellular networks and/or in a personal smart card.

PCS phones and PDAs have already become consumer market products and devices that combine both are already available. In the near future they will be incorporated into a broad array of *wearable* devices, much as radios and tape and CD players in the 1980s and 1990s. We may find them in earrings, in tie clasps, or for those who hate ties, in cowboy-style turquoise-clasp string ties!

SUMMARY

The purpose of this chapter was to provide an introduction to circuit-switched networks and their two major tasks: (1) providing circuits for the flow of user information and (2) providing the signaling required to establish circuits dynamically.

A long-term trend in communications is availability of communication systems with higher bit rates and greater reliability. We began the chapter with a discussion of multiplexing techniques that allow the bandwidth of such systems to be shared among multiple users. We considered FDM, TDM, and WDM, and introduced the digital multiplexing hierarchy that forms the backbone of current transmission systems.

Optical fiber technology figures prominently in current and future backbone networks. We introduced the SONET standard for optical fiber transmission, and we discussed its role in the design of transmission systems that are flexible in terms of configuration and robust with respect to faults. Linear, ring, and mesh topology approaches to providing reliable transport connections were introduced. We also discussed the role of WDM in all optical networks. The availability of huge bandwidths from WDM will change both the access and the backbone architecture of future networks.

In the next few sections of the chapter we discussed the basic concepts of circuit switching and the modern telephone network. We discussed the design of circuit switches and explained their role in providing end-to-end physical connections on demand. We discussed the structure of the telephone network and paid particular attention to the local loop that represents a major challenge to providing very high bandwidth to the user. The role of signaling in setting up telephone connections was discussed, and the layered architecture of the signaling system was examined. The management of traffic flows in telephone networks through routing plans as well as overload controls was also discussed.

The student may ask why one should bother studying "old" networks and "old" ways of doing things. The reason is that certain fundamental concepts underlie all networks, and that many of these concepts are built into the telephone network. Nevertheless, it is also true that the current way of operating the telephone network is not the only way and not always the right way. The trick, of course, is to know which concepts apply in which context, and only a good grasp of the fundamentals and a careful examination of any given situation can provide the best solution.

As an example of the generality of telephony concepts consider the cellular networks that were discussed in the last section of the chapter. The value of terminal portability and user mobility and the relatively low infrastructure cost are all factors in the explosive growth of cellular communications. We saw how the key control functions that enable mobility are built on the signaling infrastructure that was developed for telephone networks. Thus the "new" frequently builds on the "old".

CHECKLIST OF IMPORTANT TERMS

add-drop multiplexer (ADM)
◆ arrival rate
automatic protection switching (APS)
base station
bidirectional line switched ring (BLSR)
cellular radio communications

circuit switch
circuit-switching network
◆ concentration
crossbar switch
digital cross-connect (DCC)
digital multiplexing hierarchy
digital signal 1 (DS1)
◆ Erlang B formula
frequency-division multiplexing (FDM)
frequency reuse
Global System for Mobile Communications (GSM)
handoff
◆ holding time
intelligent network
Integrated Services Digital Network (ISDN)
multiplexing
◆ multistage switch
nonblocking switch
◆ offered load *a*
optical add-drop multiplexer (OADM)
optical carrier level (OC)
◆ overload control
pointer
roaming user
◆ routing control
setup channels
signaling
space-division switch
stored program control
synchronous digital hierarchy (SDH)
synchronous optical network (SONET)
synchronous payload envelope (SPE)
synchronous transfer module (STM)
synchronous transport signal level (STS)
time-division multiplexing (TDM)
◆ time-division switching
time-slot interchange (TSI)
◆ time-space-time (TST) switch
◆ traffic management
transport network
unidirectional path switched ring (UPSR)
◆ utilization
wavelength-division multiplexing (WDM)

FURTHER READING

Bellamy, J., *Digital Telephony*, John Wiley & Sons, Inc., New York, 2000.
Carne, E. B., *Telecommunications Primer: Signals, Building, Blocks, and Networks*, Prentice-Hall PTR, Upper Saddle River, New Jersey, 1995.
Clark, M. P., *Network and Telecommunications: Design and Operation*, John Wiley & Sons, New York, 1997.
Clos, C., "A Study of Non-Blocking Switching Networks," *Bell System Technical Journal*, Vol. 32: 406–424, March 1953.
Cooper, R. B., *Introduction to Queueing Theory*, North Holland, Amsterdam, 1981.
Computer Science and Telecommunications Board (CSTB), *The Evolution of Untethered Communications*, National Academy of Sciences, Washington D.C., 1997.
Gallagher, M. D. and R. A. Snyder, *Mobile Telecommunications Networking with IS-41*, McGraw-Hill, New York, 1997.
Goodman, D. J., *Wireless Personal Communications Systems*, Addison-Wesley, Reading, Massachusetts, 1997.
Liu, M., *Principles and Applications of Optical Communications*, McGraw-Hill, New York, 1996.
Mukherjee, B., *Optical Communication Networks*, McGraw-Hill, New York, 1997.
Pecar, J. A., R. J. O'Connor, and D. A. Garbin, *The McGraw-Hill Telecommunications Factbook*, McGraw-Hill, New York, 1993.

Ramaswami, R. and K. N. Sivaraian, *Optical Networks: A Practical Perspective*, Morgan Kaufmann, San Francisco, 1998.

Russell, T., *Signaling System #7*, McGraw-Hill, New York, 1995.

Scourias, J., "Overview of the Global System for Mobile Communications," Department of Computer Science, University of Waterloo, Waterloo, Ontario, Canada. Technical Report CS-96-15, March 1996.

Sexton, M. and A. Reid, *Transmission Networking: SONET and the Synchronous Digital Hierarchy*, Artech House, Boston, 1992.

See our website for additional references available through the Internet.

PROBLEMS

4.1. A television transmission channel occupies a bandwidth of 6 MHz.
   (a) How many two-way 30 kHz analog voice channels can be frequency-division multiplexed in a single television channel?
   (b) How many two-way 200 kHz GSM channels can be frequency-division multiplexed in a single television channel?
   (c) Discuss the tradeoffs involved in converting existing television channels to cellular telephony channels?

4.2. A cable sheath has an inner diameter of 2.5 cm.
   (a) Estimate the number of wires that can be contained in the cable if the wire has a diameter of 5 mm.
   (b) Estimate the diameter of a cable that holds 2700 wire pairs.

4.3. Suppose that a frequency band W Hz wide is divided into M channels of equal bandwidth.
   (a) What bit rate is achievable in each channel? Assume all channels have the same SNR.
   (b) What bit rate is available to each of M users if the entire frequency band is used as a single channel and TDM is applied?
   (c) How does the comparison of (a) and (b) change if we suppose that FDM requires a guard band between adjacent channels? Assume the guard band is 10% of the channel bandwidth.

4.4. In a cable television system (see Section 3.8.2), the frequency band from 5 MHz to 42 MHz is allocated to upstream signals from the user to the network, and the band from 550 MHz to 750 MHz is allocated for downstream signals from the network to the users.
   (a) How many 2 MHz upstream channels can the system provide? What bit rate can each channel support if a 16-point QAM constellation modem is used?
   (b) How many 6 MHz downstream channels can the system provide? What bit rates can each channel support if there is an option of 64-point or 256-point QAM modems?

4.5. Suppose a radio transmission system has a large band of available bandwidth, say, 1 GHz, that is to be used by a central office to transmit and receive from a large number of users. Compare the following two approaches to organizing the system:
   (a) A single TDM system.
   (b) A hybrid TDM/FDM system in which the frequency band is divided into multiple channels and TDM is used within each channel.

4.6. Suppose an organization leases a T-1 line between two sites. Suppose that 32 kbps speech coding is used instead of PCM. Explain how the T-1 line can be used to carry twice the number of calls.

4.7. A basic rate ISDN transmission system uses TDM. Frames are transmitted at a rate of 4000 frames/second. Sketch a possible frame structure. Recall that basic rate ISDN provides two 64 kbps channels and one 16 kbps channel. Assume that one-fourth of the frame consists of overhead bits.

4.8. The T-1 carrier system uses a framing bit to identify the beginning of each frame. This is done by alternating the value of the framing bit at each frame, assuming that no other bits can sustain an alternating pattern indefinitely. Framing is done by examining each of 193 possible bit positions successively until an alternating pattern of sufficient duration is detected. Assume that each information bit takes a value of 0 or 1 independently and with equal probability.

(a) Consider an information bit position in the frame. Calculate the average number of times this bit position needs to be observed before the alternating pattern is found to be violated.

(b) Now suppose that the frame synchronizer begins at a random bit position in the frame. Suppose the synchronizer observes the given bit position until it observes a violation of the alternating pattern. Calculate the average number of bits that elapse until the frame synchronizer locks onto the framing bit.

4.9. The CEPT-1 carrier system uses a framing *byte* at the beginning of a frame.

(a) Suppose that all frames begin with the same byte pattern. What is the probability that this pattern occurs elsewhere in the frame? Assume that each information bit takes a value of 0 or 1 independently and with equal probability.

(b) Consider an arbitrary information bit position in the frame. Calculate the average number of times that the byte beginning in this bit position needs to be observed before it is found to not be the framing byte.

(c) Now suppose that the frame synchronizer begins at a random bit position in the frame. Suppose the synchronizer observes the byte beginning in the given bit position until it observes a violation of the alternating pattern. Calculate the average number of bits that elapse until the frame synchronizer locks onto the framing byte.

4.10. Suppose a multiplexer has two input streams, each at a nominal rate of 1 Mbps. To accommodate deviations from the nominal rate, the multiplexer transmits at a rate of 2.2 Mbps as follows. Each group of 22 bits in the output of the multiplexer contains 18 positions that always carry information bits, nine from each input. The remaining four positions consist of two flag bits and two data bits. Each flag bit indicates whether the corresponding data bit carries user information or a stuff bit because user information was not available at the input.

(a) Suppose that the two input lines operate at exactly 1 Mbps. How frequently are the stuff bits used?

(b) How much does this multiplexer allow the input lines to deviate from their nominal rate?

4.11. Calculate the number of voice channels that can be carried by an STS-1, STS-3, STS-12, STS-48, and STS-192 Calculate the number of MPEG2 video channels that can be carried by these systems.

4.12. SONET allows positive or negative byte stuffing to take place at most once every four frames. Calculate the minimum and maximum rates of the payload that can be carried within an STS-1 SPE.

4.13. Consider a SONET ring with four stations. Suppose that tributaries are established between each pair of stations to produce a logical topology. Find the capacity required in each hop of the SONET ring in the following three cases, assuming first that the ring is unidirectional and then that the ring is bidirectional.
(a) The traffic between each pair of stations is one STS-1.
(b) Each station produces three STS-1's worth of traffic to the next station in the ring and no traffic to other stations.
(c) Each station produces three STS-1's worth of traffic to the farthest station along the ring and no traffic to other stations.

4.14. Consider a set of 16 sites organized into a two-tier hierarchy of rings. At the lower tier a bidirectional SONET ring connects four sites. At the higher tier, a bidirectional SONET ring connects the four lower-level SONET rings. Assume that each site generates traffic that requires an STS-3.
(a) Discuss the bandwidth requirements that are possible if 80% of the traffic generated by each site is destined to other sites in the same tier ring.
(b) Discuss the bandwidth requirements that are possible if 80% of the traffic generated by each site is destined to sites in other rings.

4.15. Compare the efficiency of BLSR and UPSR rings in the following two cases:
(a) All traffic originating at the nodes in the ring are destined for a given central node.
(b) Each node originates an equal amount of traffic to all other nodes.

4.16. Consider the operation of the dual gateways for interconnecting two bidirectional SONET rings shown in Figure 4.29. The primary gateway transmits the desired signal to the other ring and simultaneously transmits the signal to the secondary gateway that also routes the signal across the ring and then to the primary gateway. A service selector switch at the primary gateway selects between the primary and secondary signals. Explain how this setup recovers from failures in the link between the primary gateways.

4.17. Consider the synchronous multiplexing in Figure 4.15. Explain how the pointers in the outgoing STS-1 signals are determined.

4.18. Draw a sketch to explain the relationship between a virtual tributary and the synchronous payload envelope. Show how 28 T-1 signals can be carried in an STS-1.

4.19. Do a web search for DWDM equipment available from telecommunications equipment vendors. What specifications do you see in terms of number of channels, spacing between channels, bit rates and formats of the signals that can be carried?

4.20. Consider WDM systems with 100, 200, and 400 wavelengths operating at the 1550 nm region and each carrying an STS-48 signal.
(a) How close do these systems come to using the available bandwidth in the 1550 nm range?
(b) How many telephone calls can be carried by each of these systems? How many MPEG2 television signals?

4.21. Consider the SONET fault protection schemes described earlier in the chapter. Explain whether these schemes can be used with WDM rings.

4.22. Calculate the spacing between the WDM component signals in Figure 4.18. What is the spacing in hertz, and how does it compare to the bandwidth of each component signal?

4.23. How does WDM technology affect the hierarchical SONET ring topology in Figure 4.29? In other words, what are the consequences of a single fiber providing large number of high-bandwidth channels?

4.24. WDM and SONET can be used to create various logical topologies over a given physical topology. Discuss how WDM and SONET differ and explain what impact these differences have in the way logical topologies can be defined.

4.25. Compare the operation of a multiplexer, an add-drop multiplexer, a switch, and a digital cross-connect.

4.26. Consider a crossbar switch with n inputs and k outputs.
(a) Explain why the switch is called a concentrator when $n > k$. Under what traffic conditions is this switch appropriate?
(b) Explain why the switch is called an expander when $n < k$. Under what traffic conditions is this switch appropriate?
(c) Suppose an $N \times N$ switch consists of three stages: an $N \times k$ concentration stage; a $k \times k$ crossbar stage; and a $k \times N$ expansion stage. Under what conditions is this arrangement appropriate?
(d) When does the three-stage switch in part (c) fail to provide a connection between an idle input and an idle output line?

4.27. Consider the multistage switch in Figure 4.35 with $N = 16, n = 4, k = 2$.
(a) What is the maximum number of connections that can be supported at any given time? Repeat for $k = 4$ and $k = 10$.
(b) For a given set of input-output pairs, is there more than one way to arrange the connections over the multistage switch?

4.28. In the multistage switch in Figure 4.35, an input line is busy 10% of the time.
(a) Estimate the percent of time p that a line between the first and second stage is busy.
(b) How is p affected by n and k?
(c) How does this p affect the blocking performance of the intermediate crossbar switch?
(d) Supposing that the blocking probability of the intermediate crossbar is small, what is the proportion of time p' that a line between the second and third stage is busy?
(e) For a given input and output line, what is the probability that none of the N/n paths between the input and output lines are available?

4.29. Consider the multistage switch in Figure 4.35 with $N = 32$. Compare the number of crosspoints required by a nonblocking switch with $n = 16, n = 8, n = 4$, and $n = 2$.

4.30. A multicast connection involves transmitting information from one source user to several destination users.

(a) Explain how a multicast connection may be implemented in a crossbar switch.

(b) How does the presence of multicast connections affect blocking performance of a crossbar switch? Are unicast calls adversely affected?

(c) Explain how multicast connections should be implemented in a multistage switch. Should the multicast branching be done as soon as possible or as late as possible?

4.31. Do a web search for crosspoint switching chips. What specifications do you find for number of ports, bit rate/port? Are dualcasting and multicasting supported? Do the crosspoints need to be changed together or can they be changed independently? What considerations determine how fast the crosspoints can be changed?

4.32. What is the delay incurred in traversing a TSI switch?

4.33. Explain how the TSI method can be used to build a time-division multiplexer that takes four T-1 lines and combines them into a single time-division multiplexed signal. Be specific in terms of the number of registers required and the speeds of the lines involved.

4.34. Suppose that the TDM frame structure is changed so that each frame carries two PCM samples. Does this change affect the maximum number of channels that can be supported using TSI switching?

4.35. Examine the time-space-time circuit-switch architecture and explain the elements that lead to greater compactness, that is, smaller physical size in the resulting switching system.

4.36. Consider the three-stage switch in Problem 4.26c. Explain why a space-time-space implementation of this switch makes sense. Identify the factors that limit the size of the switches that can be built using this approach.

4.37. Consider n digital telephones interconnected by a unidirectional ring. Suppose that transmissions in the ring are organized into frames with slots that can hold one PCM sample.

(a) Suppose each telephone has designated slot numbers into which it inserts its PCM sample in the outgoing direction and from which it extracts its received PCM sample from the incoming direction. Explain how a TSI system can provide the required connections in this system.

(b) Explain how the TSI system can be eliminated if the pairs of users are allowed to share a time slot.

(c) How would connections be established in parts (a) and (b)?

4.38. Consider the application of a crossbar structure for switching optical signals.

(a) What functions are the crosspoints required to implement?

(b) Consider a 2×2 crossbar switch and suppose that the switch connection pattern is $(1 \rightarrow 1, 2 \rightarrow 2)$ for T seconds and $(1 \rightarrow 2, 2 \rightarrow 1)$ for T seconds. Suppose it takes τ seconds to change between connection patterns, so the incoming optical signals must have guard bands to allow for this gap. Calculate the relationship between the bit rate R of the information in the optical signals, the number of bits in each "frame," and the values T and τ. For R in the range from 1 gigabit/second to 1 terabit/second and τ in the range of 1 microsecond to 1 millisecond, find values of T that yield 50% efficiency in the use of the transmission capacity.

4.39. Suppose an optical signal contains n wavelength-division multiplexed signals at wavelengths $\lambda_1, \lambda_2, \ldots, \lambda_n$. Consider the multistage switch structures in Figure 4.35 and suppose that the first-stage switches consist of an element that splits the incoming optical signal into n separate optical signals each at one of the incoming wavelengths. The jth such signal is routed to the jth crossbar switch in the middle stage. Explain how the resulting switch structure can be used to provide a rearrangeable optical switch.

4.40. Consider the equipment involved in providing a call between two telephone sets.

(a) Sketch a diagram showing the various equipment and facilities between the originating telephone through a single telephone switch and on to the destination telephone. Suppose first that the local loop carries analog voice; then suppose it carries digital voice.

(b) Repeat part (a) in the case where the two telephone calls are in different LATAs.

(c) In parts (a) and (b) identify the points at which a call request during setup can be blocked because resources were unavailable.

4.41. Suppose that an Internet service provider has a pool of modems located in a telephone office and that a T-1 digital leased line is used to connect to the ISP's office. Explain how the 56 K modem (that was discussed in Chapter 3) can be used to provide a 56 kbps transfer rate from the ISP to the user. Sketch a diagram showing the various equipment and facilities involved.

4.42. Why does a conventional telephone still work when the electrical power is out?

4.43. In Figure 4.44b, how does the network know which interexchange carrier is to be used to route a long-distance call?

4.44. ADSL was designed to provide high-speed digital access using existing telephone facilities.

(a) Explain how ADSL is deployed in the local loop.

(b) What happens after the twisted pairs enter the telephone office?

(c) Can ADSL and ISDN services be provided together? Explain why or why not.

4.45. In this problem we compare the local loop topology of the telephone network with the coaxial cable topology of cable television networks (discussed in Chapter 3).

(a) Explain how telephone service may be provided by using the cable television network.

(b) Explain how cable television service may be provided by using the local loop.

(c) Compare both topologies in terms of providing Internet access service.

4.46. The local loop was described as having a star topology in the feeder plant and a star topology in the distribution plant.

(a) Compare the star-star topology with a star-ring topology and a ring-ring topology. Explain how information flows in these topologies and consider issues such as efficiency in use of bandwidth and robustness with respect to faults.

(b) What role would SONET transmission systems play in the above topologies?

4.47. Suppose that the local loop is upgraded so that optical fiber connects the central office to the pedestal and twisted pair of length at most 1000 feet connects the user to the pedestal.

(a) What bandwidth can be provided to the user?

(b) What bit rate does the optical fiber have to carry if each pedestal handles 500 users?

(c) How can SONET equipment be used in this setting?

4.48. Let's consider an approach for providing fiber-to-the-home connectivity from the central office to the user. The telephone conversations of users are time-division multiplexed at the telephone office and *broadcast* over a "passive optical network" that operates as follows. The TDM signal is broadcast on a fiber up to a "passive optical splitter" that transfers the optical signal to N optical fibers that are connected to N users. Each user receives the entire TDM signal and retrieves its own signal.
(a) Trace the flow of PCM samples to and from the user. What bit rates are required if $N = 10$? 100? Compare these to the bit rate that is available.
(b) Discuss how Internet access service might be provided by using this approach.
(c) Discuss how cable television service might be provided by using this approach.
(d) What role could WDM transmission play in the connection between the central office and the optical splitter? in the connection all the way to the user?

4.49. Explain where the following fit in the OSI reference model:
(a) A 4 kHz analog connection across the telephone network.
(b) A 33.6 kbps modem connection across the telephone network.
(c) A 64 kbps digital connection across the telephone network.

4.50. For the following examples of connections, explain the signaling events that take place inside the network as a connection is set up and released. Identify the equipment and facilities involved in each phase of the connection.
(a) A voice call in the telephone network.
(b) A frame relay connection.
(c) A SONET connection in metropolitan network as shown in Figure 4.29.
(d) An optical connection across the continent.

4.51. Sketch the sequence of events that take place in the setting up of a credit-card call over the intelligent network. Identify the equipment involved in each phase.

4.52. Explain how the intelligent network can provide the following services:
(a) Caller identification—A display on your phone gives the telephone number or name of the incoming caller.
(b) Call forwarding—Allows you to have calls transferred from one phone to another where you can be reached.
(c) Call answer—Takes voice mail message if you are on the phone or do not answer the phone.
(d) Call waiting—If you are on the phone, a distinctive tone indicates that you have an incoming local or long-distance call. You can place your current call on hold, speak briefly to the incoming party, and return to the original call.
(e) Called-party identification—Each member of a family has a different phone number. Incoming calls have a different ring for each phone number.

4.53. Consider a 64 kbps connection in ISDN.
(a) Sketch the layers of the protocol stack in the user plane that are involved in the connection. Assume that two switches are involved in the connection.
(b) Sketch the layers of the protocol stack in the control plane that are involved in setting up the connection. Assume that the two switches in part (a) are involved in the call setup.

4.54. Explain how the control plane from ISDN can be used to set up a virtual connection in a packet-switching network. Is a separate signaling network required in this case?

4.55. Identify the components in the delay that transpires from when a user makes a request for a telephone connection to when the connection is set up. Which of these components increase as the volume of connection requests increases?

4.56. Discuss the fault tolerance properties of the STP interconnection structure in the signaling system in Figure 4.52.

4.57. A set of trunks has an offered load of 10 Erlangs. How many trunks are required to obtain a blocking probability of 2%? Use the following recursive expression for the Erlang B blocking probability:

$$B(c, a) = \frac{aB(c-1, a)}{c + aB(c-1, a)} \qquad B(0, a) = 1$$

4.58. Compare the blocking probabilities of a set of trunks with offered load $a = 9$ and $c = 10$ trunks to a system that is obtained by scaling up by a factor of 10, that is, $a = 90$ and $c = 100$. Hint: Use the recursion in Problem 4.57 in a spreadsheet or program.

4.59. Calls arrive to a pool of 50 modems according to a Poisson process. Calls have an average duration of 25 minutes.
(a) What is the probability an arriving call finds all modems busy if the arrival rate is two calls per minute?
(b) What is the maximum arrival rate that can be handled if the maximum acceptable blocking probability is 1%? 10%?

4.60. Consider dynamic nonhierarchical routing (DNHR).
(a) Explain how DNHR can be used to exploit the time differences between different time zones in a continent.
(b) Explain how DNHR can be used to exploit different business and residential activity patterns during the day.

4.61. Suppose that setting up a call requires reserving N switch and link segments.
(a) Suppose that each segment is available with probability p. What is the probability that a call request can be completed?
(b) In allocating switch and transmission resources, explain why it makes sense to give priority to call requests that are almost completed rather than to locally originating call requests.

4.62. Consider a cellular telephone system with the following parameters: B is the total bandwidth available for the system for communications in both directions; b is the bandwidth required by each channel, including guard bands; R is the reuse factor; and a is the fraction of channels used for set up.
(a) Find an expression for the number of channels available in each cell.
(b) Evaluate the number of channels in each cell for the AMPS system.

4.63. Consider the AMPS system in Problem 4.62.
(a) How many Erlangs of traffic can be supported by the channels in a cell with a 1% blocking probability? 5%?
(b) Explain why requests for channels from handoffs should receive priority over requests for channels from new calls. How does this change the Erlang load calculations?

4.64. Suppose that an analog cellular telephone system is converted to digital format by taking each channel and converting it into n digital telephone channels.

(a) Find an expression for the number of channels that can be provided in the digital system using the parameters introduced in Problem 4.62.

(b) Consider the AMPS system and assume that it is converted to digital format using $n = 3$. How many Erlangs of traffic can the new system support at 1% blocking probability? 5%?

4.65. Suppose that a CDMA system has the same number of channels as the digital system in Problem 4.64, but with a reuse factor of 1.

(a) How many Erlangs of traffic can be supported in each cell by this system at 1% blocking probability? 5%?

(b) Suppose the per capita traffic generated in a city is 0.10 Erlangs during the busiest hour of the day. The city has a population of 1 million residents, and the traffic is generated uniformly throughout the city. Estimate the number of cells required to meet the city's traffic demand using the system in part (a).

4.66. Consider the equipment involved in providing a call between mobile and wireline telephones.

(a) Sketch a diagram showing the various equipment and facilities between an originating mobile telephone to a wireline destination telephone. Suppose first that the mobile station carries analog voice; then suppose it carries digital voice.

(b) Repeat part (a) in the case where the two telephone calls are mobile.

(c) In parts (a) and (b) identify the points at which a call request can be blocked during call setup because resources are unavailable.

4.67. Explain the signaling events that take place when a call is set up and released in a cellular telephone system. Identify the equipment and facilities involved in each phase of the call. Consider the following cases.

(a) The source and destination mobile phones are in the same cell.

(b) The source and destination mobile phones are in different cells but in the same MSC.

(c) The source and destination mobile phones are in different cellular networks.

4.68. Explain the signaling events that take place when a call is handed off from one cell to another cell. Suppose first that the two cells are under the same MSC and then that they are under different MSCs.

4.69. Bcb Smiley, star salesman, has just arrived in Los Angeles from his home base in Chicago. He turns on his cell phone to contact his Los Angeles client.

(a) Sketch the sequence of events that take place to set up his call. Assume he subscribes to roaming service.

(b) Next he calls home to Chicago to inform his wife that he forgot to take out his son's hockey gear from the trunk of his car and to give her the parking spot where he left the car in the airport ("somewhere on the third level of parking lot A"). Sketch the sequence of events that take place to set up his call. (Don't concern yourself with the specifics of the conversation.)

(c) In the meantime, Bob's college roommate, Kelly, who now works in Hollywood, calls Bob's cell phone. Note that Bob's cell phone begins with the Chicago area code. Sketch the sequence of events that take place to set up this call. Should this call be billed as local or long distance?

4.70. Compare cellular wireless networks to the local loop and to the coaxial cable television system in terms of their suitability to serve as an integrated access network. In particular, comment on the ability to support telephone service, high-speed Internet access, and digital television service. Consider the following two cases:

(a) The band of frequencies available spans 300 MHz.

(b) The band of frequencies available spans 2 GHz.

CHAPTER 5

Peer-to-Peer Protocols and Data Link Layer

We have seen that the overall communications process can be broken up into layers consisting of communications functions that can be grouped together. In Chapter 2 we introduced the notion of a layer service and protocol. Each layer provides a service to the layer above and it does so by executing a peer-to-peer protocol that uses the services of the layer below. The objective of Part I of this chapter is to answer the basic question: "How does a peer-to-peer protocol deliver a service?" First we provide several examples of services that can be provided by a layer. To answer the question, we then discuss in detail the peer-to-peer protocols that provide reliable data transfer service across unreliable transmission lines and/or networks. Part II of the chapter deals with standard data link layer protocols that incorporate the results of the peer-to-peer protocols developed in Part I to provide reliable data transfer service. The chapter is organized as follows.

Part I:

1. *Peer-to-peer protocols and service models.* We provide examples of services that can be provided by a protocol layer. We also examine the two cases where peer-to-peer protocols occur, across a single hop in a network and end-to-end across multiple hops in a network, and we explain the different requirements that must be met in these two cases.
2. *ARQ protocols and reliable data transfer service.* We consider Automatic Repeat Request (ARQ) protocols that provide reliable data transfer service. These protocols are essential when transmitting over channels and networks that are prone to errors. This detailed section is very important because it highlights an essential challenge in network protocols, namely, the coordination of the actions of two or more geographically separate machines with different state information. We also examine the efficiency of these protocols in various scenarios and identify the delay-bandwidth product as a key parameter in network performance.

3. *Flow control, reliable data transfer, and timing recovery.* We consider other services that are provided by peer-to-peer protocols: flow control so that fast end-systems do not overrun the buffers of slow end-systems; and synchronization and timing recovery for applications that involve voice, audio, and video. We also introduce Transmission Control Protocol (TCP), which uses ARQ techniques to provide reliable stream service and flow control end-to-end across connectionless packet networks.[1]

Part II:

4. *Framing in data link layer.* We introduce two techniques that are used to identify the boundaries of frames of information within a digital bit stream: Flags and bit stuffing and CRC-based framing.
5. *Data link layer protocols.* We examine the data link layer and its essential functions. We discuss two data link control standards that are in widespread use: Point-to-Point Protocol (PPP) and High-Level Data Link Control (HDLC).
6. *Statistical multiplexing.* We examine the performance of statistical multiplexers that enable packets from multiple flows to share a common data link. We show how these multiplexers allow links to be used efficiently despite the fact that the individual packet flows are bursty.

PART I: Peer-to-Peer Protocols

In Chapter 2 we saw that the communications process can be broken into layers as shown in Figure 5.1. At each layer two or more entities or **peer processes** execute a protocol that delivers the service that is provided to the layer above. The communications between the layer n + 1 peer processes is virtual and in fact is carried out by using a service provided by layer n. In this chapter we examine the peer-to-peer protocol that is carried out by the layer-n peer processes to provide the desired service. We consider the processing that takes place from when layer n + 1 requests a transfer of a **service data unit (SDU)** to when the SDU is delivered to the destination layer n + 1. In particular, we examine how the layer n peer processes construct **protocol data units (PDUs)** and convey control information through **headers**. We show how in certain protocols each peer process maintains a state that dictates what actions are to be performed when certain events occur. The implementation of the layer n protocol uses the services of layer n − 1 and so involves an interaction between layer n and layer n − 1.

The discussion so far is fairly abstract. In Section 5.1 we introduce concrete examples of what we mean by a "service." In Section 5.2 we provide a detailed development of peer-to-peer protocols that deliver reliable data transfer service. Section 5.3 introduces additional examples of peer-to-peer protocols.

[1]The TCP protocol is discussed in detail in Chapter 8.

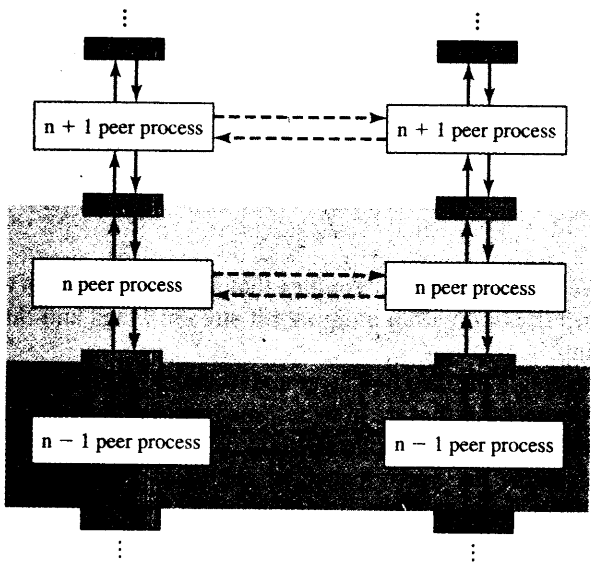


FIGURE 5.1 Layer-n peer processes carry out a protocol to provide service to layer n + 1. Layer-n protocol uses the services of layer n − 1.

5.1 PEER-TO-PEER PROTOCOLS AND SERVICE MODELS

In this section we consider peer-to-peer protocols and the services they provide. A **peer-to-peer protocol** involves the interaction of two or more processes or entities through the exchange of messages, called protocol data units (PDUs). The service provided by a protocol is described by a service model.

5.1.1 Service Models

The **service model** in a given layer specifies the manner in which information is transferred. There are two broad categories of service models: connection oriented and connectionless. In **connection-oriented services** a connection *setup* procedure precedes the transfer of information. This connection setup initializes state information in the two layer-n peer processes in Figure 5.1 and establishes a "pipe" for the transfer of information provided by the users of the service, the layer n + 1 peer processes, as shown in Figure 5.2. During the *data transfer phase*, this state information provides

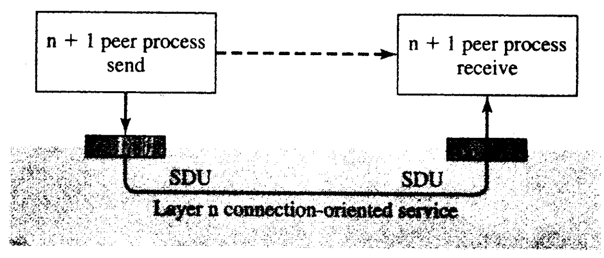


FIGURE 5.2 Connection-oriented transfer service.

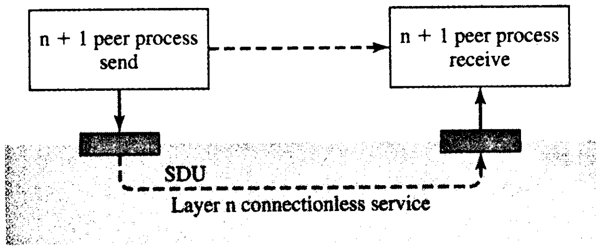


FIGURE 5.3 Connectionless transfer service.

a context that the layer-n peer processes use to track the exchange of PDUs between themselves as well as the exchange of SDUs with the higher layer. In particular, the context can be used to provide value-added services such as in-order delivery of SDUs to the higher layer. Connection-oriented services also involve a *connection release* procedure that removes the state information and releases the resources allocated to the connection.

Connectionless services do not involve a connection setup procedure. Instead, individual self-contained blocks of information are transmitted and delivered using appropriate address information as shown in Figure 5.3. Information blocks transmitted from the same user to the same destination are transmitted independently. In the simplest case the service does not provide an acknowledgment for transmitted information. Thus if information is lost during transmission, no effort is made to retransmit it. This type of service is appropriate when the transfer of each PDU is reliable or when the higher layer is more sensitive to delay than to occasional losses. Other applications require reliable transfer, so each PDU needs to be acknowledged and retransmitted if necessary. Note that the service is still connectionless so the order in which SDUs are delivered may differ from the order in which they were transmitted.

A service model may also specify a type of transfer capability. For example, connectionless services necessarily involve the transfer of clearly defined *blocks* of information. Some services place an upper limit on the size of the blocks that are transferred while others insist that the blocks consist of an integer number of octets. On the other hand, connection-oriented services can transfer both a *stream* of information in which the individual bits or bytes are not grouped into blocks as well as sequences of blocks of information. Furthermore, some service models may be intended to transfer information at a *constant bit rate*, while others are designed to transfer information at a *variable bit rate*.

The service model can also include a **quality-of-service (QoS)** requirement that specifies a level of performance that can be expected in the transfer of information. For example, QoS may specify levels of reliability in terms of probability of errors, probability of loss, or probability of incorrect delivery. QoS may also address transfer delay. For example, a service could guarantee a fixed (nearly) constant transfer delay, or it could guarantee that the delay will not exceed some given maximum value. The variation in the transfer delay is also an important QoS measure for services such as real-time voice and video. The term **best-effort service** describes a service in which every effort is made to deliver information but without any guarantees.

5.1.2 Examples of Services

Service models differ according to the features of the data transfer they provide. A service offered by a given layer can include some of the following features:

- Arbitrary message size or structure
- Sequencing
- Reliability
- Timing
- Pacing
- Flow control
- Multiplexing
- Privacy, integrity, and authentication

Let's consider these features one at a time.

A common element of all user-to-user communication systems and indeed all communication networks is the transfer of a message from one point to another. The message may consist of a single bit, a block of bytes, or a stream of information. Thus different services will place different restrictions on the size and structure of the messages that are transferred. For example, in the case of e-mail, the message is a discrete, well-defined entity. In the case of computer data files, the size of the files can be very large, suggesting that the files be broken into smaller units that are sent as data blocks and reassembled at the far end to recreate the complete file.

In the case of telephony or video, the notion of a message is less clear. As shown in Figure 5.4a, in telephony one could view the telephone call as generating a single message that consists of the entire sequence of speech samples, but this approach fails to take into account the real-time nature of the application. A more natural view of telephony information is that of a stream of digital speech samples, as shown in Figure 5.4b, which motivates the view of the telephone call as consisting of a sequence of one-byte messages corresponding to each speech sample. (Note that a very large file can also be viewed as a stream of bits or bytes that, unlike telephony, does not

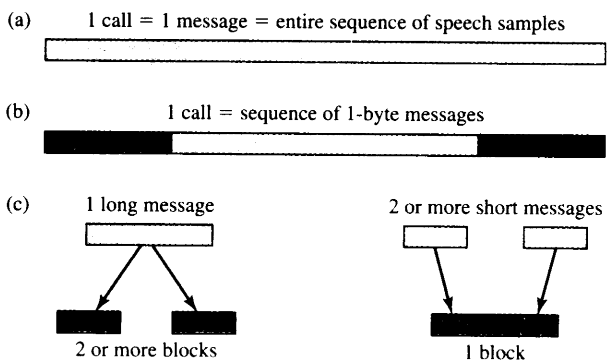


FIGURE 5.4 (a) Message, (b) stream, and (c) sequence of blocks.

have a real-time requirement.) Connection-oriented services as shown in Figure 5.2 are suitable for the transfer of stream information.

As shown in Figure 5.4c, for a service to accommodate messages of arbitrary size, its protocol may need to segment long messages into a sequence of blocks. In general, sequences of small messages can be handled either by converting each small message into a block or by combining one or more small messages into a block. We will encounter various examples of the segmentation and reassembly procedure when we discuss specific network architectures.

Many services involve the transfer of one or more messages or the transfer of streams of information that must be delivered error free, in order, and without duplication. On the other hand, many networks and transmission technologies are unreliable in the sense of introducing errors, delivering messages out of order, or losing and even duplicating messages. By combining error-detection coding, automatic retransmission, and sequence numbering, it is possible to obtain protocols that can provide *reliable and sequenced communication service over unreliable networks*. In the next section we present the details of such protocols.

Reliable end-to-end communication involves not only the arrival of messages to the destination but also the actual delivery of the messages (e.g., to the listener for voice or to the computer program for a data file). A problem that can arise here is that the receiving system does not have sufficient buffering available to store the arriving message. In this case the arriving messages are lost. This problem tends to arise when the transmitter can send information at a rate higher than the receiver can accept it. The sliding-window protocols developed in the next section can also be used to provide flow control, where the receiver paces or controls the rate at which the transmitter sends new messages. We will see that in some situations pacing is also used to control congestion inside the network.

Applications such as speech, audio, and video involve the transfer of a stream of information in which a temporal relationship exists between the information elements. In particular, each of these applications requires a procedure for playing back the information at the receiver end. The system that carries out this playback needs to have appropriate timing information in order to reconstruct the original information signal. For example, in the case of digital speech the receiver must know the appropriate rate at which samples should be entered into the digital-to-analog converter. The situation is similar but quite a bit more complex in the case of digital video. In Section 5.3.2, we show how sequence numbering and timestamps can be used to reconstruct the necessary timing information.

In many situations the service offered by a layer is shared by several users. For example, different processes in a host may simultaneously share a given network connection. In these situations the user messages need to include addressing information to allow the host to separate the messages and forward them to the appropriate process. The sharing of connections is referred to as *multiplexing*.

Public networks increasingly have a new type of "impairment" in the form of security threats. Imposters attempt to impersonate legitimate clients or servers. Attempts to deny service to others are made by flooding a server with requests. In this context protocols may be called upon to act as guards at the gateways to the public network. We discuss protocols for authenticating messages and protecting the privacy and integrity of messages in Chapter 11.

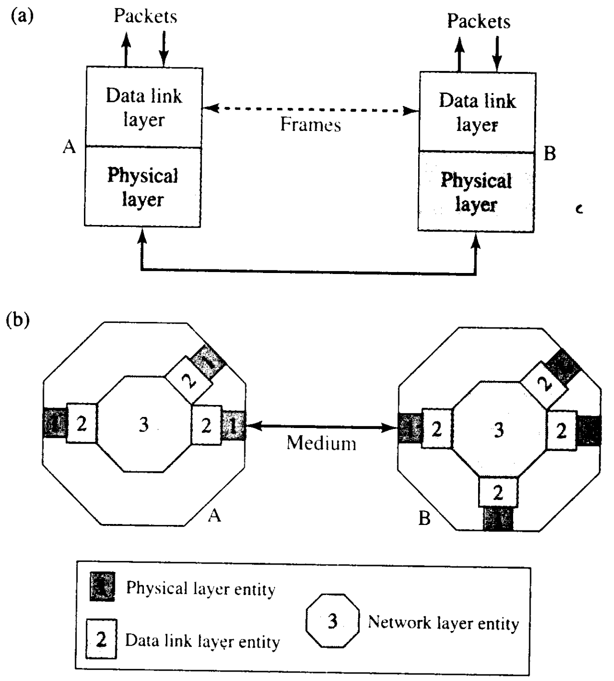


FIGURE 5.5 Peer-to-peer protocol across a single hop.

5.1.3 End to End versus Hop by Hop

Peer-to-peer protocols usually occur in two basic settings: across a single hop in the network or end to end across an entire network. These two settings can lead to different characteristics about whether PDUs arrive in order, about how long it takes for the PDUs to arrive, or about whether they arrive at all. The design of the corresponding protocols must take these characteristics into account.

Figure 5.5 shows a peer-to-peer protocol that operates across a single hop in a network. Part (a) of the figure uses the lower two layers of the OSI reference model to show how the data link layer protocol provides service for the transfer of packets across a single link in the network. The data link layer takes packets from the network layer, encapsulates them in frames that it transfers across the link, and delivers them to the network layer at the other end. Note that the frames arrive with small delay and in the order they were transmitted since the transfer involves only a single physical link. If the transmission medium is noisy or unreliable it is possible for transmitted frames to not arrive at the receiver. In Figure 5.5b we depict the data link that connects packet switch A and packet switch B in the broader context of a network.

Figure 5.6 shows a peer-to-peer protocol that operates end to end across a network. In the figure the transport layer peer processes at the end systems accept messages from their higher layer and transfer these messages by exchanging segments end to end across the network.[2] The exchange of segments is accomplished by using network

[2]We use the term *segment* for the transport layer PDU even though strictly speaking segment refers to the transport PDU for TCP only.

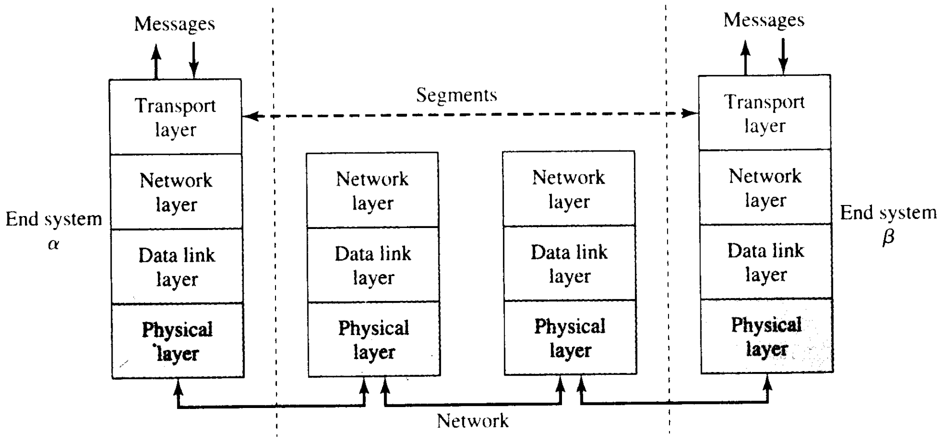


FIGURE 5.6 Peer-to-peer protocols operating end to end across a network—protocol stack view.

layer services. We use Figure 5.7 to show that the task of the peer-to-peer protocols in this case can be quite complicated. The figure shows the two end systems α and β operating across a three-node network. The segments that are exchanged by the end systems are encapsulated in packets that traverse the three-node network. Suppose that the network operates in datagram mode where packets from the same end system are

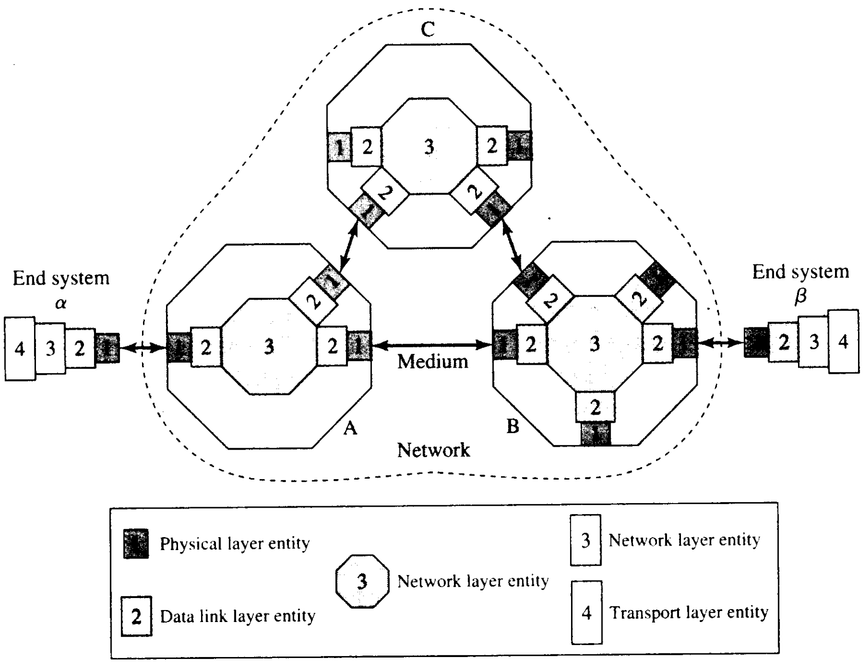


FIGURE 5.7 Peer-to-peer protocols operating end to end across a network—spatial view.

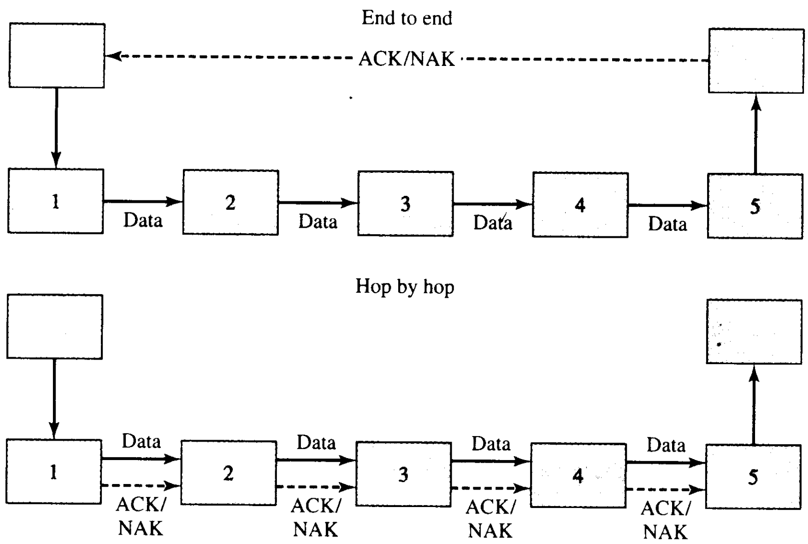


FIGURE 5.8 End-to-end versus hop-by-hop approaches.

routed independently. It is then possible that packets will follow different paths across the network, and so their corresponding segments may arrive out of order at their destination. Some packets and their segments may also be delayed for long periods or even lost if they are routed to a congested packet switch. The peer processes at the transport layer may need to take into account all the characteristics of the network transfer service to be able to provide the desired service to its higher layer. Note that end-to-end peer protocols can also be implemented at layers higher than the transport layer, for example, HTTP at the application layer.

In many situations an option exists for achieving a certain network transfer capability on an end-to-end basis or on a hop-by-hop basis, as shown in Figure 5.8. For example, to provide reliable communication, error-control procedures can be introduced at every hop, that is, between every pair of adjacent nodes in a path across the network. Every node is then required to implement a protocol that checks for errors and requests retransmission using ACK and NAK messages until a block of information is received correctly. Only then is the block forwarded along the next hop to the next node. An end-to-end approach, on the other hand, removes the error-recovery responsibility from the intermediate nodes. Instead blocks of information are forwarded across the path, and only the end systems are responsible for initiating error recovery.

There is a basic tradeoff in choosing between these two approaches. The hop-by-hop approach initiates error recovery more quickly and may give more reliable service. On the other hand, the processing in each node is more complex. In addition, for the hop-by-hop approach to be effective on an end-to-end basis *every* element in the end-to-end chain must operate correctly. For example, the hop-by-hop approach in Figure 5.8 is vulnerable to the introduction of errors within the switches.

The possibility of such errors would necessitate introducing end-to-end error-recovery procedures.[3]

In the case of reliable transfer service, both approaches have been implemented. In situations where errors are infrequent, end-to-end mechanisms are preferred. The TCP reliable stream service introduced later in this chapter provides an example of such an end-to-end mechanism. In situations where errors are likely, hop-by-hop error recovery becomes necessary. The HDLC data link control is an example of a hop-by-hop mechanism.

The end-to-end versus hop-by-hop option appears in other situations as well. For example, flow control and congestion control can be exercised on a hop-by-hop or an end-to-end basis. Security mechanisms provide another example in which this choice needs to be addressed. The approach selected typically determines which layer of the protocol stack provides the desired function. Thus, for example, mechanisms for providing congestion control and mechanisms for providing security are available at the data link layer, the network layer, and the transport layer.

5.2 ARQ PROTOCOLS AND RELIABLE DATA TRANSFER SERVICE

Reliable transfer of information is a critical requirement in communications between computers. In this section we consider specific peer-to-peer protocols, called the ARQ protocols, which provide reliable data transfer service. **Automatic Repeat Request (ARQ)** combines error detection and retransmission to ensure that data is delivered accurately to the user despite errors that occur during transmission. In this section we develop the three basic types of ARQ protocols, starting with the simplest and building up to the most complex. We also discuss the situations in which the three ARQ protocol types are applied.

Consider Figure 5.1 and suppose that the user at layer n + 1 generates information blocks, that is, SDUs, for transmission to a layer n + 1 peer process. Suppose also that the layer n + 1 processes require *reliable data transfer service*, that is, the information blocks need to be delivered in the correct sequence without errors, duplication, or gaps. We consider the development of a layer n protocol that offers this service. For simplicity we first consider the case of unidirectional transmission. This situation allows us to refer to one of the layer n peer processes as the transmitter and to the other peer process as the receiver. Without loss of generality we also refer to the layer n + 1 PDUs as *packets* and the layer n PDUs as *frames*. This terminology corresponds precisely to the case where the data link control layer provides service to the network layer, but the model applies equally well to other layers. For example, we could also have layer n + 1 be the application layer, layer n be the transport layer, and layer n − 1 be the network layer.

[3]We return to a discussion of the end-to-end argument for system design in Chapter 7.

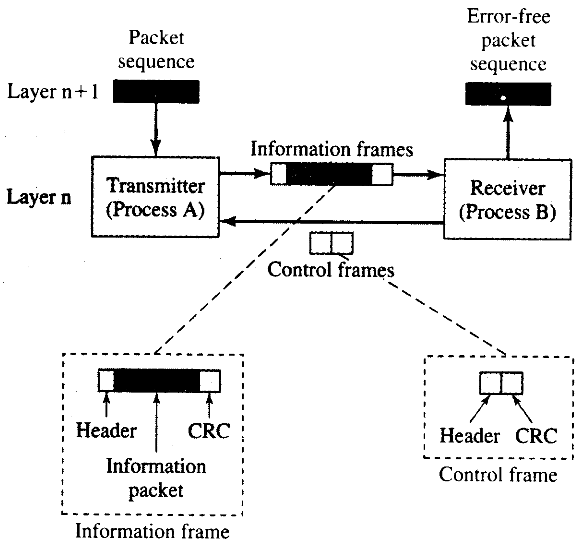


FIGURE 5.9 Basic elements of ARQ.

The ARQ mechanism requires the frame to contain a header with control information that is needed for its proper operation, as shown in Figure 5.9. The transmitter will also append CRC check bits that cover the header and the information bits to enable the receiver to determine whether errors have occurred during transmission. We assume that the design of the CRC ensures that transmission errors can be detected with very high probability, as discussed in Chapter 3. We refer to the set of rules that govern the operation of the transmitter and receiver as the **ARQ protocol**.

The ARQ protocol can be applied in a number of scenarios. Traditionally, the most common application has been in data link controls that operate over a single noisy communication channel. ARQ is introduced here to ensure a high level of reliability across a single transmission hop. As communication lines have become less noisy, the ARQ protocol has been implemented more often at the edges of the network in the transport layer to provide end-to-end reliability in the transmission of packets over multiple hops in a network, that is, over multiple communication channels and other network equipment. In this section we assume that the transfer service provided by layer n − 1 in Figure 5.1 is "wirelike" in the sense that the frames arrive at the receiver from layer n − 1, if they arrive at all, in the same order in which they were sent. In the case of multiple hops over a network, this assumption would hold when a connection is set up and where all frames follow the same path, as in frame relay or ATM networks. In particular, we assume that frames, while in transit, cannot pass previously transmitted frames.[4] In situations where these assumptions hold, the objective of the ARQ protocol is to ensure that packets are delivered error free to the destination, exactly once without duplicates, in the same order in which they were transmitted.

In addition to the error-detection code, the other basic elements of ARQ protocols consist of **information frames (I-frames)** that transfer the user packets, control frames,

[4] We consider the case where frames can arrive out of order later in this chapter.

and time-out mechanisms, as shown in Figure 5.9. **Control frames** are short binary blocks that consist of a header that provides the control information followed by the CRC. The control frames may include **ACKs**, which acknowledge the correct receipt of a given frame or group of frames; and **NAKs**, which indicate that a frame has been received in error and that the receiver is taking certain action. The headers contain fields that are used to identify the type of frame, that is, information or control, and ACK or NAK. We will see that the time-out mechanisms are required to prompt certain actions to maintain the flow of frames. We can visualize the transmitter and receiver as working jointly through the exchange of frames to provide the correct and orderly delivery of the sequence of packets provided by the sender.

It is instructive at this point to consider the basic operation of the packet transfer protocol in the absence of errors. This operation illustrates the interactions that take place between the various layers:

1. Each time there is information to send, the layer n + 1 process makes a call to the layer n service and passes the layer n + 1 PDU down.
2. Layer n takes the layer n SDU (layer n + 1 PDU), prepares a layer n PDU according to the layer n protocol, and then makes a call to the service of layer n − 1.
3. At the destination, layer n − 1 notifies the layer n process when a layer n − 1 SDU (layer n PDU) has arrived.
4. The layer n process accepts the PDU, executes the layer n protocol, and if appropriate, recovers the layer n SDU.
5. Layer n then notifies layer n + 1 that a layer n SDU has arrived.

Every protocol in the following sections includes these interlayer interactions.

5.2.1 Stop-and-Wait ARQ

The first protocol we consider is **Stop-and-Wait ARQ** where the transmitter and receiver work on the delivery of one frame at a time. The protocol begins with transmitter A sending an information frame to receiver B. The transmitter then stops and waits for an acknowledgment from the receiver. If no acknowledgment is received within some time-out period, the transmitter resends the frame, and once again stops and waits.

In Figure 5.10a we show how ACKs and time-outs can be used to provide recovery from transmission errors, in this case a lost frame. At the initial point in the figure, processes A and B are working on the transmission of frame 0. Note that each time A sends an I-frame, it starts an **I-frame timer** that will expire after some time-out period. The **time-out** period is selected so that it is greater than the time required to receive the corresponding ACK frame. Figure 5.10a shows the following sequence of events:

1. Process A transmits frame 0 and then waits for an ACK frame from the receiver.
2. Frame 0 is received without error, so process B transmits an ACK frame.
3. The ACK from B is also received without error, so process A knows the frame 0 has been received correctly.
4. Process A now proceeds to transmit frame 1 and then resets the timer.

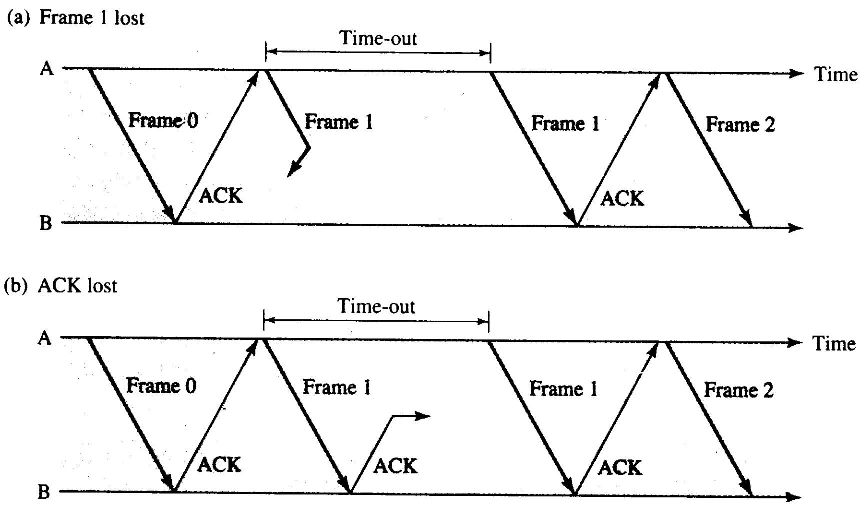


FIGURE 5.10 Possible ambiguities when frames are unnumbered. In parts (a) and (b) transmitting process A acts the same way, but part (b) receiving process B accepts frame 1 twice.

5. Frame 1 undergoes errors in transmission. It is possible that process B receives frame 1 and detects the errors through the CRC check; it is also possible that frame 1 was so badly garbled that process B is unaware of the transmission.[5] In either case process B does not take any action.
6. The time-out period expires, and frame 1 is retransmitted.

The protocol continues in this manner until frame 1 is received and acknowledged. The protocol then proceeds to frame 2, and so on.

THE NEED FOR SEQUENCE NUMBERS

Transmission errors in the reverse channel lead to ambiguities in the Stop-and-Wait protocol that need to be corrected. Figure 5.10b shows the situation that begins as in Figure 5.10a, but where frame 1 is received correctly, and its acknowledgment undergoes errors. After receiving frame 1 process B delivers its packet to its upper layer. Process A does not receive the acknowledgment for frame 1, so the time-out period expires. Note that at this point process A cannot distinguish between the sequence of events in parts (a) and (b) of Figure 5.10. Process A proceeds to retransmit the frame. If the frame is received correctly by process B, as shown in the figure, then process B will accept frame 1 as a new frame and redeliver its packet to the user. Thus we see that the loss of an ACK can result in the delivery of a duplicate packet. The ambiguity can be

[5]In general, when errors are detected in a frame, the frame is ignored. The receiver cannot trust any of the data in the frame and, in particular, cannot take any actions based on the contents of the frame header.

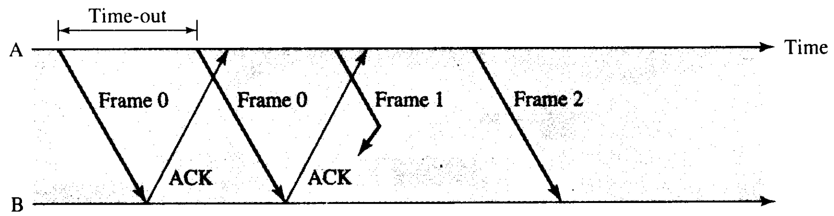


FIGURE 5.11 Possible ambiguities when ACKs are unnumbered: Transmitting process A misinterprets duplicate ACKs.

eliminated by including a **sequence number** in the header of each I-frame. Process B would then recognize that the second transmission of frame 1 was a duplicate, discard the frame, and resend the ACK for frame 1.

A second type of ambiguity arises if the ACKs do not contain a sequence number. In Figure 5.11 frame 0 is transmitted, but the time-out expires prematurely. Frame 0 is received correctly, and the (unnumbered) ACK is returned. In the meantime process A has resent frame 0. Shortly thereafter, process A receives an ACK and assumes it is for the last frame. Process A then proceeds to send frame 1, which incurs transmission errors. In the meantime the second transmission of frame 0 has been received and acknowledged by process B. When process A receives the second ACK, the process assumes the ACK is for frame 1 and proceeds to transmit frame 2. The mechanism fails because frame 1 is not delivered. This example shows that *premature time-outs (or delayed ACKs) combined with loss of I-frames can result in gaps in the delivered packet sequence*. This ambiguity is resolved by providing a sequence number in the acknowledgment frames that enables the transmitter to determine which frames have been received.

The sequence numbers cannot be allowed to become arbitrarily large because only a finite number of bits are available in the frame headers. We now show that a one-bit sequence number suffices to remove the above ambiguities in the Stop-and-Wait protocol. Figure 5.12 shows the information or "state" that is maintained by the transmitter and receiver. The transmitter must keep track of the sequence number S_{last} of the frame being sent, its associated timer, and the frame itself in case retransmission is required. The receiver keeps track only of the sequence number R_{next} of the next frame it is expecting to receive.

Suppose that initially the transmitter and receiver are synchronized in the sense that process A is about to send a frame with $S_{last} = 0$ and process B is expecting $R_{next} = 0$. In Figure 5.12 the global state of the system is defined by the pair (S_{last}, R_{next}), so initially the system is in state (0,0).

The system state will not change until process B receives an error-free version of frame 0. That is, process A will continue resending frame 0 as dictated by the time-out mechanism. Eventually process B receives frame 0, process B changes R_{next} to 1 and sends an acknowledgment to process A with $R_{next} = 1$ implicitly acknowledging the receipt of frame 0. At this point the state of the system is (0,1). Any subsequent received frames that have sequence number 0 are recognized as duplicates and discarded

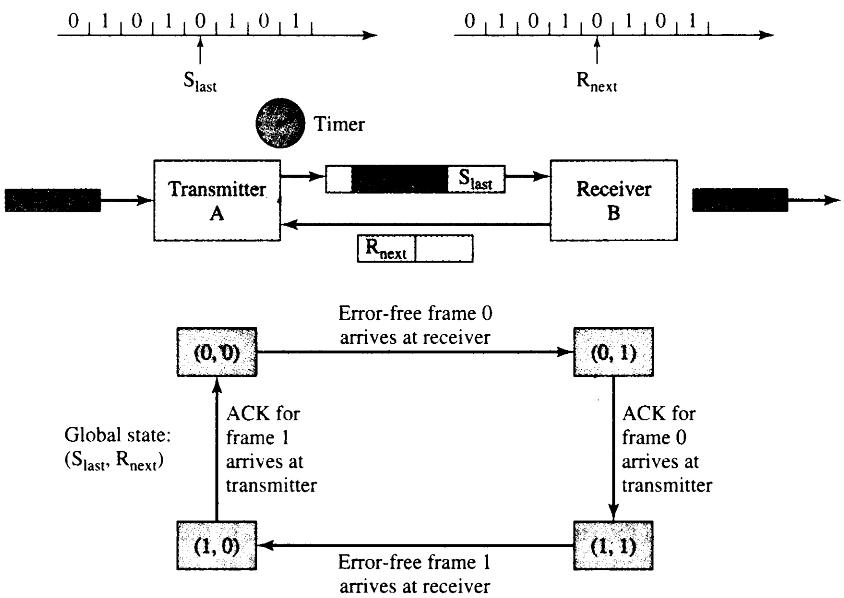


FIGURE 5.12 System state information in Stop-and-Wait ARQ.

by process B, and an acknowledgment with $R_{next} = 1$ is resent. Eventually process A receives an acknowledgment with $R_{next} = 1$ and then begins transmitting the next frame, using sequence number $S_{last} = 1$. The system is now in state (1,1). The transmitter and receiver are again synchronized, and they now proceed to work together on the transfer of frame 1. Therefore, a protocol that implements this mechanism (that follows the well-defined sequence of states shown in Figure 5.12) can ensure the correct and orderly delivery of frames to the destination.

THE PROTOCOL

Let us review the operation of the Stop-and-Wait protocol. The protocol is initialized with the transmitter in the *ready state*, the transmitter send-sequence number S_{last} set to 0, and the receiver sequence number R_{next} set to 0.

Transmitter:

- The transmitter in the ready state waits for a request for service from its higher layer. When a request occurs, a packet is received from the higher layer, and the transmitter prepares a frame that consists of a header, with sequence number S_{last}, the packet, and a CRC. A timer is started and the frame is then transmitted using the services of the layer below. The transmitter then enters a *wait state*.
- The transmitter remains in the wait state until an acknowledgment is received or the time-out period expires. The transmitter does not accept packets from the layer above while in the wait state. If the time-out expires, the frame is retransmitted, the timer is reset, and the process stays in the wait state. If an acknowledgment is received with an incorrect sequence number or with detected errors, then the acknowledgment

is ignored and the process stays in the wait state. If an acknowledgment with the correct sequence number is received, that is, $R_{next} = (S_{last} + 1)$ mod 2, then the send sequence number is updated to R_{next}, and the transmitter returns to the ready state.

Receiver:

- The receiver is always in the *ready state* waiting for a notification of an arriving frame from the layer below. When a frame arrives, the receiver accepts the frame and checks for errors. If no errors are detected and the received sequence number is the expected number, that is $S_{last} = R_{next}$, then the frame is accepted, the receive sequence number is updated to $(R_{next} + 1)$ mod 2, an acknowledgment frame is transmitted, and the packet is delivered to the higher layer. If an arriving frame has no errors but the wrong sequence number, then the frame is discarded, and an acknowledgment is sent with sequence number R_{next}. If an arriving frame has errors, then the frame is discarded and no further action is taken.

EXAMPLE Bisync

Stop-and-Wait ARQ was used in IBM's Binary Synchronous Communications (Bisync) protocol. Bisync is a character-oriented data link control that uses the ASCII character set, including several control characters. In Bisync, error detection was provided by two-dimensional block coding where each seven-bit character has a parity bit attached and where an overall block-check character is appended to a frame. Bisync has been replaced by data link controls based on HDLC, which is discussed later in this chapter.

EXAMPLE Xmodem

Xmodem, a popular file transfer protocol for modems, incorporates a form of Stop-and-Wait ARQ. Information is transmitted in fixed-length blocks consisting of a 3-byte header, 128 bytes of data, and a one-byte checksum. The header consists of a special start-of-header character, a one-byte sequence number, and a 2s complement of the sequence number. The check character is computed by taking the modulo 2 sum of the 128 data bytes. The receiver transmits an ACK or NAK character after receiving each block.

PERFORMANCE ISSUES

Stop-and-Wait ARQ works well on channels that have low propagation delay. However, the protocol becomes inefficient when the propagation delay is much greater than the time to transmit a frame as shown in Figure 5.13. For example, suppose we are transmitting frames that are 1000 bits long over a channel that has a speed of 1.5 megabits/second and suppose that the time that elapses from the beginning of the frame transmission to the receipt of its acknowledgment is 40 ms. The number of bits that can be transmitted over this channel in 40 ms is $40 \times 10^{-3} \times 1.5 \times 10^6 = 60{,}000$ bits. However, Stop-and-Wait ARQ can transmit only 1000 bits in this period time, so the efficiency is only $1000/60{,}000 = 1.6\%$. The **delay-bandwidth product** is the product of the bit

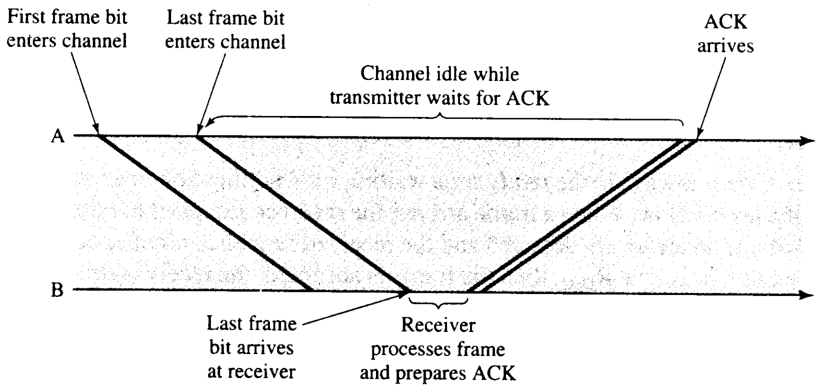


FIGURE 5.13 Stop-and-Wait ARQ is inefficient when the time to receive an ACK is large compared to the frame transmission time.

rate and the delay that elapses before an action can take place. In the preceding example the delay-bandwidth product is 60,000 bits. If we visualize the communication channel as a pipe, then the delay-bandwidth can be interpreted as the size of the pipe, that is, the maximum number of bits that can be in transit at any given time. In Stop-and-Wait ARQ the delay-bandwidth product can be viewed as a measure of lost opportunity in terms of transmitted bits. Let's examine the factors that lead to this inefficiency more closely.

Suppose that all information frames have the same length and that the transmitter always has frames to transmit to the receiver. Figure 5.14 shows the components in the basic delay t_0 that transpires in Stop-and-Wait ARQ from the instant a frame is transmitted into the channel to the instant when the acknowledgment is confirmed. The first bit that is input into the channel appears at the output of the channel after a propagation time t_{prop}; the end of the frame is received at process B after t_f additional seconds. Process B requires t_{proc} seconds to prepare an acknowledgment frame that will require t_{ack} seconds of transmission time. After an additional propagation delay, the acknowledgment frame is received at process A. Finally, t_{proc} additional seconds are required to carry out the CRC check. The basic time to send a frame and receive an

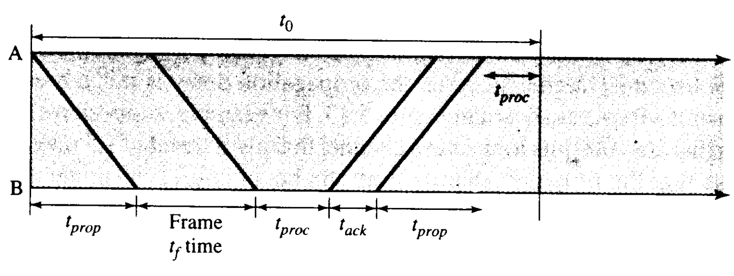


FIGURE 5.14 Delay components in Stop-and-Wait ARQ.

ACK, in the absence of errors, is then given by

$$t_0 = 2t_{prop} + 2t_{proc} + t_f + t_{ack} = 2t_{prop} + 2t_{proc} + \frac{n_f}{R} + \frac{n_a}{R} \tag{5.1}$$

where n_f is the number of bits in the information frame, n_a is the number of bits in the acknowledgment frame, and R is the bit rate of the transmission channel.

The effective information transmission rate of the protocol in the absence of errors is then given by

$$R^0_{eff} = \frac{\text{number of information bits delivered to destination}}{\text{total time required to deliver the information bits}} = \frac{n_f - n_o}{t_0} \tag{5.2}$$

where n_o is the number of overhead bits in a frame and is given by the total number of bits in the header and the number of CRC check bits. The **transmission efficiency** of Stop-and-Wait ARQ is given by the ratio R^0_{eff} to R:

$$\eta_0 = \frac{\frac{n_f - n_o}{t_0}}{R} = \frac{1 - \frac{n_o}{n_f}}{1 + \frac{n_a}{n_f} + \frac{2(t_{prop} + t_{proc})R}{n_f}} \tag{5.3}$$

Equation (5.3) identifies clearly the sources of inefficiency in transmission. In the numerator the ratio n_o/n_f represents the loss in transmission efficiency due to the need to provide headers and CRC checks. In the denominator the term n_a/n_f is the loss in efficiency due to the time required for the acknowledgment message, where n_a is the number of bits in an ACK/NAK frame. Finally, the term $2(t_{prop} + t_{proc})R$ is the delay-bandwidth product or reaction time. The following example shows the effect of this term.

EXAMPLE Effect of Delay-Bandwidth Product

Suppose that frames are 1250 bytes long including 25 bytes of overhead. Also assume that ACK frames are 25 bytes long. Calculate the efficiency of Stop-and-Wait ARQ in a system that transmits at $R = 1$ Mbps and with reaction times, $2(t_{prop} + t_{proc})$, of 1 ms, 10 ms, 100 ms, and 1 second. Repeat if $R = 1$ Gbps.

A frame is $n_f = 1250 \times 8 = 10{,}000$ bits long and the overhead is $25 \times 8 = 200$ bits long, so $n_o/n_f = n_a/n_f = 0.02$. For $R = 1$ Mbps, the delay bandwidth product is 10^3 bits, 10^4 bits, 10^5 bits, and 10^6 bits for 1 ms, 10 ms, 100 ms, and 1 second, respectively. The corresponding efficiencies are then 88%, 49%, 9%, and 1% respectively. The dramatic effect of increased propagation delay is evident.

An increase in bit rate has the same effect as an increase in propagation delay. Thus if $R = 1$ Gbps, then the delay bandwidth product is 1000 times greater than before: 10^6 bits, 10^7 bits, 10^8 bits, and 10^9 bits for reaction times of 1 ms, 10 ms, 100 ms, and 1 second, respectively. The corresponding efficiencies drop even further to the range from 1% for 1 ms delay to miniscule values for the other delays.

Now consider the effect of transmission errors on the efficiency of Stop-and-Wait ARQ. If a frame incurs errors during transmission, the time-out mechanism will cause retransmission of the frame. Each transmission or retransmission will take up t_0 seconds. Let P_f be the probability that a frame transmission has errors and needs to be retransmitted. Suppose 1 in 10 frame transmissions get through without error, that is $1 - P_f = 0.1$, then on average a frame will have to be transmitted 10 times to get through. In general we have that $1/(1 - P_f)$ frame transmissions are required for general values of P_f if frame transmission errors are independent. Thus Stop-and-Wait ARQ on average requires $t_{SW} = t_0/(1 - P_f)$ seconds to get a frame through. The efficiency is then obtained by modifying Equation (5.3):

$$\eta_{SW} = \frac{\frac{n_f - n_o}{t_{SW}}}{R} = \frac{1 - \frac{n_o}{n_f}}{1 + \frac{n_a}{n_f} + \frac{2(t_{prop} + t_{proc})R}{n_f}}(1 - P_f). \tag{5.4}$$

This equation is also derived in Appendix 5A.

EXAMPLE **Effect of Bit Error Rate**

Suppose that frames are 1250 bytes long including 25 bytes of overhead. Also assume that ACK frames are 25 bytes long. Calculate the efficiency of Stop-and-Wait ARQ in a system that transmits at $R = 1$ Mbps and with reaction time of 1 ms for channels with a *bit* error rate of 10^{-6}, 10^{-5}, and 10^{-4}.

As in the previous example, we have $n_f = 10{,}000$ bits and $n_o = n_a = 200$ bits. For $R = 1$ Mbps and 1 ms reaction time, the delay-bandwidth product is 10^3 bits. From Equation (5.4) we see that the efficiency is given by $\eta_0(1 - P_f)$ where η_0 is the efficiency with no errors. We need to find an expression that relates P_f, the probability that a frame has errors, to p, the probability that the transmission of a single bit is in error. A frame gets through correctly is there are no bit errors, that is, all n_f bits are transmitted without error. The probability that a single bit gets through without error is $(1 - p)$, so that probability $1 - P_f$ that all n_f get through assuming random bit errors is then given by:

$$1 - P_f = (1 - p)^{n_f} \tag{5.5}$$

We can then readily calculate that $1 - P_f = 0.99, 0.905$, and 0.368 for $p = 10^{-6}$, $p = 10^{-5}$, and $p = 10^{-4}$, respectively. The corresponding efficiencies are 86.6%, 79.2%, and 32.2%. We conclude that the bit error rate can have a dramatic effect on the performance of Stop-and-Wait ARQ.

5.2.2 Go-Back-N ARQ

The inefficiency of Stop-and-Wait ARQ can be overcome by allowing the transmitter to continue sending enough frames so that the channel is kept busy while the transmitter waits for acknowledgments. We now develop Go-Back-N ARQ that forms the basis for the HDLC data link protocol, which is discussed at the end of this chapter. Suppose for now that frames are numbered 0, 1, 2, 3, ... The transmitter has a limit on the number of frames W_S that can be outstanding without acknowledgment. The **window size** W_S

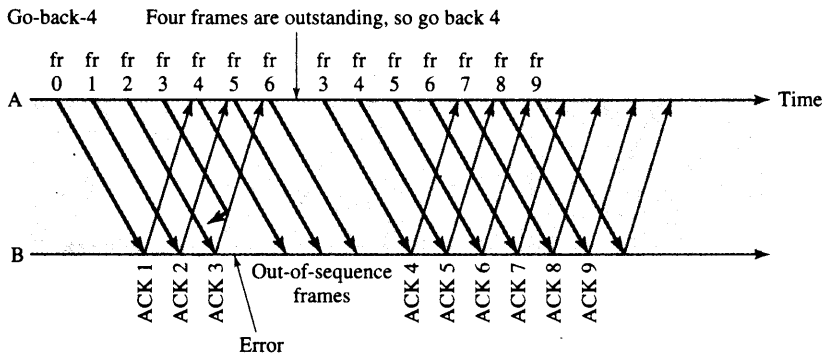


FIGURE 5.15 Basic Go-Back-N ARQ.

is chosen larger than the delay-bandwidth product to ensure that the channel or pipe is kept full.

The idea of the basic **Go-Back-N ARQ** is as follows: Consider the transfer of a reference frame, say, frame 0. After frame 0 is sent, the transmitter sends $W_S - 1$ additional frames into the channel, optimistic that frame 0 will be received correctly and not require retransmission. If things turn out as expected, an ACK for frame 0 will arrive in due course while the transmitter is still busy sending frames into the channel, as shown in Figure 5.15. The system is now done with frame 0. Note, however, that the handling of frame 1 and subsequent frames is already well underway. A procedure where the processing of a new task is begun before the completion of the previous task is said to be **pipelined**. In effect Go-Back-N ARQ pipelines the processing of frames to keep the channel busy.

Go-Back-N ARQ takes its name from the action that is taken when an error occurs. As shown in Figure 5.15, after frame 3 undergoes transmission errors, the receiver ignores frame 3 and all subsequent frames. Eventually the transmitter reaches the maximum number of outstanding frames. It is then forced to "go back N" frames, where $N = W_S$, and begin retransmitting all packets from 3 onwards.

In the previous discussion of Stop-and-Wait, we used the notion of a global state (S_{last}, R_{next}) to demonstrate the correct operation of the protocol in the sense of delivering the packets error free and in order. Figure 5.16 shows the similarities between Go-Back-N ARQ and Stop-and-Wait ARQ in terms of error recovery. In Stop-and-Wait ARQ, the occurrence of a frame-transmission error results in the loss of transmission time equal to the duration of the time-out period. In Go-Back-N ARQ, the occurrence of a frame-transmission error results in the loss of transmission time corresponding to W_S frames. In Stop-and-Wait the receiver is looking for the frame with sequence number R_{next}; in Go-Back-N the receiver is looking for a frame with a specific sequence number, say, R_{next}. If we identify the oldest outstanding (transmitted but unacknowledged) frame in Go-Back-N with the S_{last} frame in Stop-and-Wait, we see that the correct operation of Go-Back-N ARQ depends on ensuring that the oldest frame is eventually delivered successfully. The protocol will trigger a retransmission of S_{last} and the subsequent $W_S - 1$ frames each time the send window is exhausted. Therefore, as long as there is a nonzero probability of error-free frame transmission, the eventual error-free

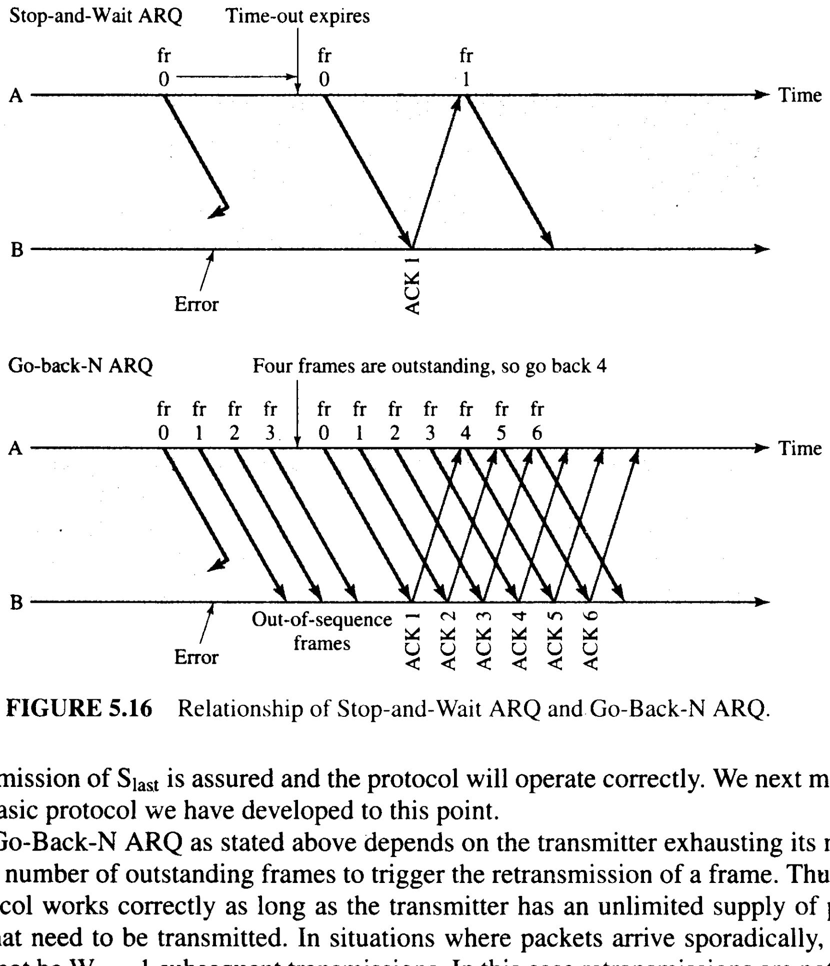


FIGURE 5.16 Relationship of Stop-and-Wait ARQ and Go-Back-N ARQ.

transmission of S_{last} is assured and the protocol will operate correctly. We next modify the basic protocol we have developed to this point.

Go-Back-N ARQ as stated above depends on the transmitter exhausting its maximum number of outstanding frames to trigger the retransmission of a frame. Thus this protocol works correctly as long as the transmitter has an unlimited supply of packets that need to be transmitted. In situations where packets arrive sporadically, there may not be $W_S - 1$ subsequent transmissions. In this case retransmissions are not triggered, since the window is not exhausted. This problem is easily resolved by modifying Go-Back-N ARQ such that a timer is associated with each transmitted frame.

Figure 5.17 shows how the modified Go-Back-N ARQ operates. The transmitter must now maintain a list of the frames it is processing, where S_{last} is the number of the last transmitted frame that remains unacknowledged and S_{recent} is the number of the most recently transmitted frame. The transmitter must also maintain a timer for each transmitted frame and must also buffer all frames that have been transmitted but have not yet been acknowledged. At any point in time the transmitter has a **send window** of available sequence numbers. The lower end of the window is given by S_{last}, and the upper limit of the transmitter window is $S_{last} + W_S - 1$. If S_{recent} reaches the upper limit of the window, the transmitter is not allowed to transmit further new frames until the send window "slides" forward with the receipt of a new acknowledgment.

Go-Back-N ARQ an example of a **sliding-window protocol**. The receiver maintains a **receive window** of size 1 that consists of the next frame R_{next} it expects to receive. If an arriving frame passes the CRC check and has the correct sequence number, that is,

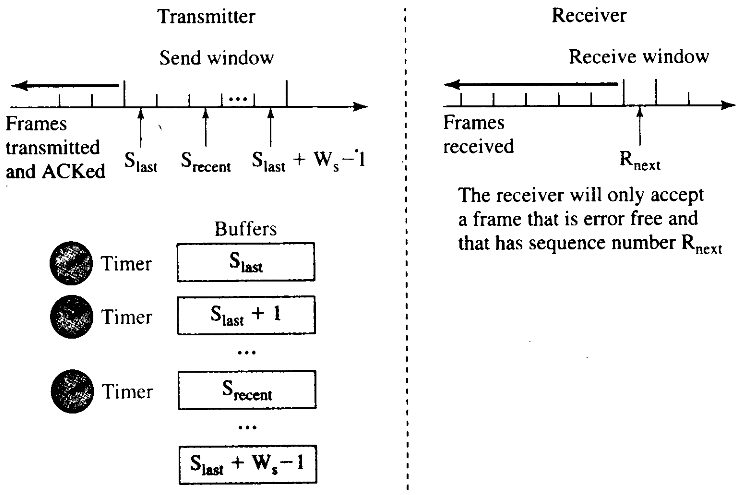


FIGURE 5.17 Go-Back-N ARQ.

R_{next}, then it is accepted and R_{next} is incremented. We say that the receive window slides forward. The receiver then sends an acknowledgment containing the incremented sequence number R_{next}, which implicitly acknowledges receipt of all frames prior to R_{next}. Note how we are making use of the assumption that the channel is wirelike: When the transmitter receives an ACK with a given value R_{next}, it can assume that all prior frames have been received correctly, even if it has not received ACKs for those frames, either because they were lost or because the receiver chose not to send them. Upon receiving an ACK with a given value R_{next}, the transmitter updates its value of S_{last} to R_{next} and in so doing the send window slides forward. (Note that this action implies that $S_{last} \leq R_{next}$. Note as well that $R_{next} \leq S_{recent}$, since S_{recent} is the last in the transmission frames.)

THE PROTOCOL

The protocol is initialized with the transmitter send window set to $\{0, 1, \ldots, W_S - 1\}$ and $S_{last} = 0$, and the receiver window set to $\{0\}$ and $R_{next} = 0$.

Transmitter:

- When the send window is nonempty, the transmitter is in the *ready state* waiting for a request for its service from its higher layer. When a request occurs, the transmitter accepts a packet from the higher layer, and the transmitter prepares a frame that consists of a header, with sequence number S_{recent} set to the lowest available number in the send window, the packet, and a CRC. A timer is started and the frame is then transmitted using the services of the layer below. If $S_{recent} = S_{last} + W_S - 1$, then the send window has become empty and the transmitter goes into the *blocking state*; otherwise the transmitter remains in the ready state. If an error-free ACK frame is received with R_{next} in the range between S_{last} and S_{recent}, then the window slides forward by setting $S_{last} = R_{next}$ and the maximum send window number to $S_{last} + W_S - 1$. If an error-free ACK frame is received with R_{next} outside the range from S_{last} to S_{recent}, then the frame is discarded and no further action is taken. If a timer expires, then the transmitter resends S_{last} and all subsequent frames, and resets the timer for each frame.

- The transmitter is in the *blocking state* when the send window is empty, that is, $S_{recent} = S_{last} + W_S - 1$, and the transmitter refuses to accept requests for packet transfers from the layer above. If a timer expires, then the transmitter resends S_{last} and all subsequent frames, and resets the timer for each frame. If an error-free ACK frame is received with R_{next} in the range between S_{last} and S_{recent}, then the window slides forward by setting $S_{last} = R_{next}$, and the maximum send window number to $S_{last} + W_S - 1$, and then changing to the ready state. If an error-free ACK frame is received with R_{next} outside the range from S_{last} to S_{recent}, then the frame is discarded and no further action is taken.

Receiver:

- The receiver is always in the ready state waiting for a notification of an arriving frame from the layer below. When a frame arrives, the receiver checks for errors. If no errors are detected and the received sequence number is the expected number, that is, R_{next}, then the frame is accepted, the receive sequence number is incremented to $R_{next} + 1$, an acknowledgment frame is transmitted, and the packet is delivered to the higher layer. If an arriving frame has no errors but the wrong sequence number, then the frame is discarded, and an acknowledgment is sent with sequence number R_{next}. If an arriving frame has errors, then the frame is discarded and no further action is taken.

It is worth noting that the complexity of the receiver in Go-Back-N is the same as that of Stop-and-Wait ARQ. Only the complexity of the transmitter is increased.

MAXIMUM WINDOW SIZE

The number of bits that can be allotted within a header per sequence number is limited to some number, say, m, which then allows us to represent at most 2^m possible sequence numbers, so the sequence numbers must be counted using modulo 2^m. Thus if $m = 3$, then the sequence of frames would carry the sequence numbers 0, 1, 2, 3, 4, 5, 6, 7, 0, 1, 2, 3, ... When an error-free frame arrives, the receiver must be able to unambiguously determine which frame has been received, taking into account that the sequence numbers wrap around when the count reaches 2^m. We will next show that the receiver can determine the correct frame if the window size is less than 2^m.

The example in Figure 5.18 shows the ambiguities that arise when the window size is not less than 2^m. The example uses $2^m = 2^2 = 4$ sequence numbers. The transmitter initially sends four frames in a row. The receiver sends four corresponding acknowledgments, which are all obliterated in the return channel. When the transmitter exhausts its available frame numbers, it goes back four and begins retransmitting from frame 0. When frame 0 arrives at the receiver, the receiver has $R_{next} = 0$, so it accepts the frame. However, the receiver does not know whether the ACK for the previous frame 0 was received. Consequently, the receiver cannot determine whether this is a new frame 0 or an old frame 0. The second example in Figure 5.18 uses $2^m = 4$ sequence numbering, but a window size of 3. In this case we again suppose that the transmitter sends three consecutive frames and that acknowledgments are lost in the

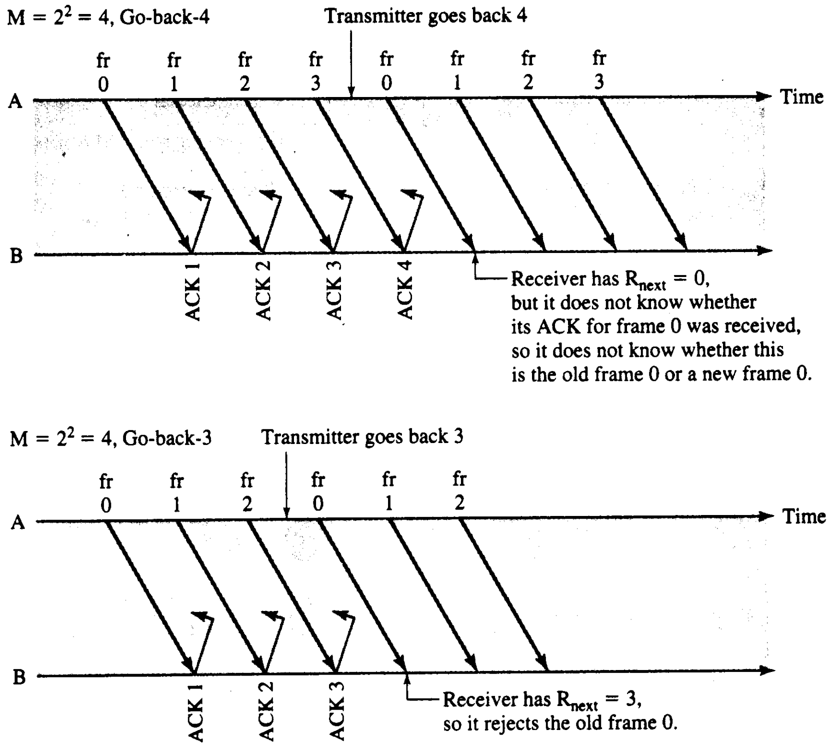


FIGURE 5.18 The window size should be less than 2^m.

return channel. When the retransmitted frame 0 arrives at the receiver, $R_{next} = 3$, so the frame is recognized to be an old one.

We can generalize Figure 5.18 as follows. In general, suppose the window size W_S is $2^m - 1$ or less and assume that the current send window is 0 up to $W_S - 1$. Suppose that frame 0 is received, but the acknowledgment for frame 0 is lost. The transmitter can only transmit new frames up to frame $W_S - 1$. Depending on which transmissions arrive without error, R_{next} will be in the range of 1 to W_S. *Crucially, the receiver will not receive frame W_S until the acknowledgment for frame 0 has been received at the transmitter.* Therefore, any receipt of frame 0 prior to frame W_S indicates a duplicate transmission of frame 0.

BIDIRECTIONAL LINKS

Finally we consider the case where the information flow is bidirectional. The transmitter and receiver functions of the modified Go-Back-N protocol are now implemented by both processes A and B. In the direct implementation the flow in each direction consists of information frames and control frames. Many of the control frames can be eliminated by **piggybacking** the acknowledgments in the headers of the information frames as shown in Figure 5.19.

This use of piggybacking can result in significant improvements in the use of bandwidth. When a receiver accepts an error-free frame, the receiver can insert the

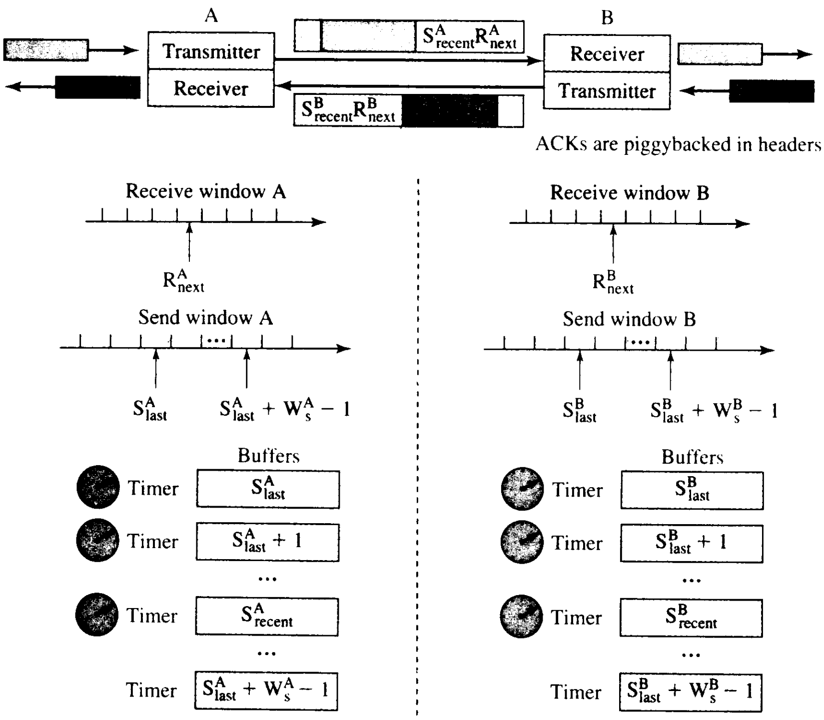


FIGURE 5.19 System parameters in bidirectional Go-Back-N ARQ.

acknowledgment into the next departing information frame. If no information frames are scheduled for transmission, the receiver can set an **ACK timer** that defines the maximum time it will wait for the availability of an information frame. When the time expires a control frame will be used to convey the acknowledgment.

In the bidirectional case the receiver handles out-of-sequence packets a little differently. A frame that arrives in error is ignored. Subsequent frames that are out of sequence but error free are discarded *after* the ACK (i.e., R_{next}) has been examined. Thus R_{next} is used to update the local S_{last}.

The I-frame time-out value should be selected so that it exceeds the time normally required to receive a frame acknowledgment. As shown in Figure 5.20, this time period includes the round-trip propagation delay $2T_{prop}$, two maximum-length frame transmission times on the reverse channel $2T_f^{max}$, and the frame processing time T_{proc}.[6] We assume that a frame transmission begins right before our frame arrives at the receiver. The ACK is then carried in the next frame that is transmitted by the receiver. The transmission time of the frame in the forward direction and the ACK time-out are absorbed by the aforementioned frame transmission times.

$$T_{out} = 2T_{prop} + 2T_f^{max} + T_{proc} \tag{5.6}$$

[6]Note that Figure 5.20 is more detailed than previous figures in that both the beginning and end of each frame transmission need to be shown explicitly.

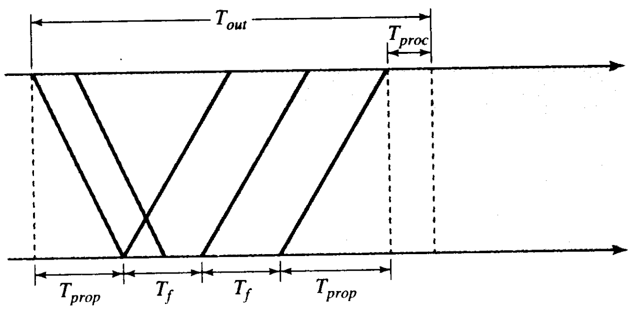


FIGURE 5.20 Calculation of time-out values.

As in the case of Stop-and-Wait ARQ, the performance of the modified Go-Back-N ARQ protocol is affected by errors in the information frames and errors in the acknowledgments. The transmission of long information frames at the receiver end can delay the transmission of acknowledgments, which can result in the expiry of time-outs at the transmitter end and in turn trigger unnecessary retransmissions. These long delays can be avoided by not using piggybacking and instead sending the short control frames ahead of the long information frames.

EXAMPLE HDLC

The High-level Data Link Control (HDLC) is a data link control standard developed by the International Standards Organization. HDLC is a bit-oriented protocol that uses the CRC-16 and CRC-32 error-detection codes discussed in Chapter 3. HDLC can be operated so that it uses Go-Back-N ARQ, as discussed in this section, or Selective Repeat ARQ, which is discussed in Section 5.2.3. HDLC is discussed in detail later in this chapter.

The most common use of HDLC is the Link Access Procedure—Balanced (LAPB) discussed later in this chapter. Several variations of the LAPB are found in different standards. For example, Link Access Procedure—Data (LAPD) is the data link standard for the data channel in ISDN. Another example is Mobile Data Link Protocol (MDLP), which is a variation of LAPD that was developed for Cellular Digital Packet Data systems that transmit digital information over AMPS and digital cellular telephone networks.

EXAMPLE V.42 Modem Standard

Error control is essential in telephone modem communications to provide protection against disturbances that corrupt the transmitted signals. The ITU-T V.42 standard was developed to provide error control for transmission over modem links. V.42 specifies the Link Access Procedure for Modems (LAPM) to provide the error control. LAPM is derived from HDLC and contains extensions for modem use. V.42 also supports the

Microcom Network Protocol (MNP) error-correction protocols as an option. Prior to the development of V.42, these early protocols were de facto standards for error control in modems.

PERFORMANCE ISSUES

Let's discuss briefly the improvement in efficiency performance that results from Go-Back-N ARQ. Consider the time t_{GBN} that it takes to get a frame through in the case where $1 - P_f = 0.1$, that is, where 1 in 10 frames get through without error: The first transmission takes $t_f = n_f/R$ seconds. With probability P_f (0.9) the first frame is in error, and so additional retransmissions are required. Each time a retransmission is required, Go-Back-N transmits W_S frames, each of duration t_f, and the average number of retransmissions is $1/(1 - P_f)$, that is, 10 retransmissions. Therefore the total average time required to transmit a frame in Go-Back-N is:

$$t_{GBN} = t_f + P_f \frac{W_S t_f}{1 - P_f} \tag{5.7}$$

Thus for the example, we have $t_{GBN} = t_f + 9W_S t_f$. If we substitute t_{GBN} into Equation (5.3), then we find that the efficiency for Go-Back-N is given by:

$$\eta_{GBN} = \frac{\frac{n_f - n_o}{t_{GBN}}}{R} = \frac{1 - \frac{n_o}{n_f}}{1 + (W_S - 1)P_f}(1 - P_f). \tag{5.8}$$

This equation is also derived in Appendix 5A. Note that if the channel is error-free, that is, $P_f = 0$, then Go-Back-N attains the best possible efficiency, namely, $1 - n_o/n_f$.

EXAMPLE **Effect of Bit Error Rate and Delay-Bandwidth Product**

Suppose that frames are 1250 bytes long including 25 bytes of overhead, and that ACK frames are 25 bytes long. Compare the efficiency of Stop-and-Wait and Go-Back-N in a system that transmits at $R = 1$ Mbps with reaction time of 100 ms, and for bit error rate of 10^{-6}, 10^{-5}, and 10^{-4}.

Consider Stop-and-Wait first. The delay-bandwidth product of this channel is 1 Mbps × 100 ms = 10^5 bits, which corresponds to about 10 frames and so the Equation (5.4) becomes

$$\eta_{SW} = \frac{0.98}{11.02}(1 - P_f) = 0.089(1 - P_f) \tag{5.9}$$

It can be seen that the best possible efficiency is 8.9% due to the large delay-bandwidth product. In the previous Stop-and-Wait example, we found that $1 - P_f = 0.99, 0.905$, and 0.368 for $p = 10^{-6}$, $p = 10^{-5}$, and $p = 10^{-4}$, respectively. We find the corresponding respective efficiencies from Equation (5.4) to be 8.8%, 8.0%, and 3.3%.

For Go-Back-N, the window size W_S must be larger than the delay-bandwidth product and so the minimum window size is 11 frames. From Equation (5.8), we find the efficiencies to be 88.2%, 45.4%, and 4.9% for $p = 10^{-6}$, $p = 10^{-5}$, and $p = 10^{-4}$, respectively. We see a substantial improvement relative to Stop-and-Wait for the first two bit error rates. The $p = 10^{-4}$ case is only marginally better than Stop-and-Wait ARQ. At this error rate Go-Back-N is retransmitting too many frames.

5.2.3 Selective Repeat ARQ

In channels that have high error rates, the Go-Back-N ARQ protocol is inefficient because of the need to retransmit the frame in error and all the subsequent frames. A more efficient ARQ protocol can be obtained by adding two new features: first, the receive window is made larger than one frame so that the receiver can accept frames that are out of order but error free; second, the retransmission mechanism is modified so that only individual frames are retransmitted. We refer to this protocol as **Selective Repeat ARQ**. We continue to work under the constraint that the ARQ protocol must deliver an error-free and ordered sequence of packets to the destination.

Figure 5.21 shows that the send window at the transmitter is unchanged but that the receive window now consists of a range of frame numbers spanning from R_{next} to $R_{next} + W_R - 1$, where W_R is the maximum number of frames that the receiver is willing to accept at a given time. As before, the basic objective of the protocol is to advance the values of R_{next} and S_{last} through the delivery of the oldest outstanding frame. Thus ACK frames carry R_{next}, the oldest frame that has not yet been received. The receive window is advanced with the receipt of an error-free frame with sequence number R_{next}. Unlike the case of Go-Back-N ARQ, the receive window may be advanced by several frames. This step occurs when one or more frames that follow R_{next} have already been received correctly and are buffered in the receiver. R_{next} and the following consecutive packets are delivered to the destination at this point.

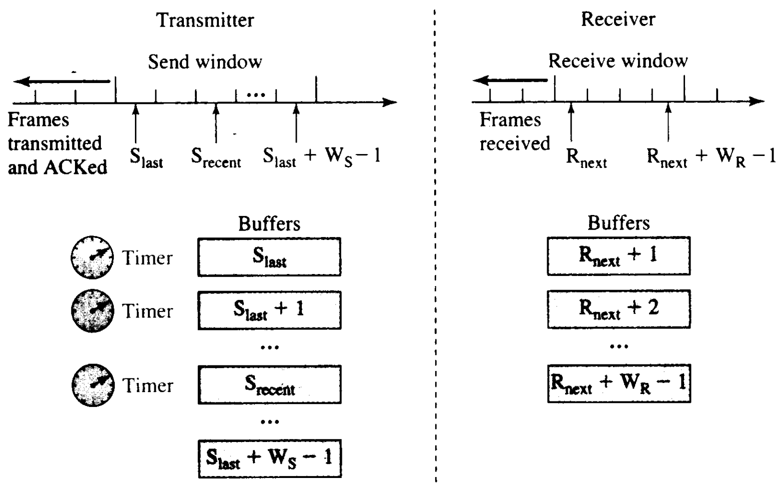


FIGURE 5.21 Selective Repeat ARQ.

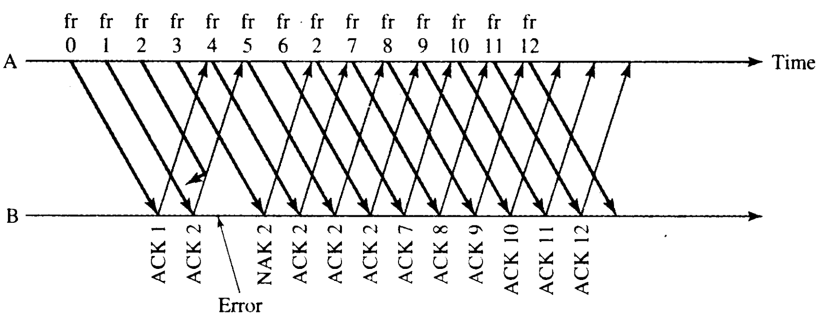


FIGURE 5.22 Error recovery in Selective Repeat ARQ.

Now consider the retransmission mechanism in Selective Repeat ARQ. The handling of timers at the transmitter is done as follows. When the timer expires, only the corresponding frame is retransmitted. There is no longer a clear correspondence between the age of the timers and the sequence numbers. This situation results in a considerable increase in complexity. Whenever an out-of-sequence frame is observed at the receiver, a NAK frame is sent with sequence number R_{next}. When the transmitter receives such a NAK frame, it retransmits the specific frame, namely, R_{next}. Finally, we note that the piggybacked acknowledgment in the information frames continues to carry R_{next}.

For example, in Figure 5.22 when the frame with sequence number 2 finally arrives correctly at the receiver, frames 3, 4, 5, and 6 have already been received correctly. Consequently, the receipt of frame 2 results in a sliding of the window forward by five frames.

THE PROTOCOL

The protocol is initialized with the transmitter send window set to $\{0, 1, \ldots, W_S - 1\}$ and $S_{last} = 0$, and the receiver window set to $\{0, \ldots, W_R - 1\}$ and $R_{next} = 0$.

Transmitter:

- When the send window is nonempty, the transmitter is in the *ready state* waiting for a request for its service from its higher layer. When a request occurs, the transmitter accepts a packet from above and prepares a frame that consists of a header, with sequence number S_{recent} set to the lowest available number in the send window, the packet, and a CRC. A timer is started and the frame is then transmitted using the services of the layer below. If $S_{recent} = S_{last} + W_S - 1$, then the send window has become empty and the transmitter goes into the blocking state; otherwise the transmitter remains in the ready state. If an error-free ACK frame is received with R_{next} in the range between S_{last} and S_{recent}, then the send window slides forward by setting $S_{last} = R_{next}$ and the maximum send window number to $S_{last} + W_S - 1$. If an error-free NAK frame is received with R_{next} in the range between S_{last} and S_{recent}, then the frame with sequence number R_{next} is retransmitted, and, if appropriate, the send window slides forward by setting $S_{last} = R_{next}$ and the maximum send window number to $S_{last} + W_S - 1$. If an error-free ACK frame is received with R_{next} outside the range from S_{last} to S_{recent}, then the frame is discarded and no further action is taken. If a timer expires, then the transmitter resends the corresponding frame and resets the timer.

- The transmitter is in the *blocking state* when the send window is empty, that is $S_{recent} = S_{last} + W_S - 1$, and the transmitter refuses to accept requests for packet transfers from the layer above. If a timer expires, then the transmitter resends the corresponding frame and resets the timer. If an error-free ACK frame is received with R_{next} in the range between S_{last} and S_{recent}, then the window slides forward by setting $S_{last} = R_{next}$ and the maximum send window number to $S_{last} + W_S - 1$, and then changing to the ready state. If an error-free NAK frame is received with R_{next} in the range between S_{last} and S_{recent}, then the frame with sequence number R_{next} is retransmitted, and, if appropriate, the send window slides forward by setting $S_{last} = R_{next}$ and the maximum send window number to $S_{last} + W_S - 1$, and the transmitter changes to the ready state. If an error-free ACK frame is received with R_{next} outside the range from S_{last} to S_{recent}, then the frame is discarded and no further action is taken.

Receiver:

- The receiver is always in the ready state waiting for a notification of an arriving frame from the layer below. When a frame arrives, the receiver checks the frame for errors. If no errors are detected and the received sequence number is the receive window, that is, in the range R_{next} to $R_{next} + W_R - 1$, then the frame is accepted and buffered, and an acknowledgment frame is transmitted. If in addition the received sequence number is R_{next}, and if $R_{next} + 1$ up to $R_{next} + k - 1$ have already been received, then the receive sequence number is incremented to $R_{next} + k$, the receive window slides forward, and the corresponding packet or packets are delivered to the higher layer. If an arriving frame has no errors but the sequence number is outside the receive window, then the frame is discarded, and an acknowledgment is sent with sequence number R_{next}. If an arriving frame has errors, then the frame is discarded and no further action is taken.

MAXIMUM WINDOW SIZE

Consider now the question of the maximum send window size that is allowed for a given sequence numbering 2^m. In Figure 5.23 we show an example in which the sequence numbering is $2^2 = 4$ and in which the send windows and receive windows are of size 3. Initially process A transmits frames 0, 1, and 2.

All three frames arrive correctly at process B, and so the receive window is advanced to $\{3, 0, 1\}$. Unfortunately all three acknowledgments are lost, so when the timer for frame 0 expires, frame 0 is retransmitted. The inadequacy of the window size now becomes evident. Upon receiving frame 0, process B cannot determine whether it is the old frame 0 or the new frame 0. So clearly, send and receive windows of size $2^m - 1$ are too large.

It turns out that the maximum allowable window size is $W_S = W_R = 2^{m-1}$, that is, half the sequence number space. To see this we retrace the arguments used in the case of Go-Back-N ARQ. Suppose the window size W_S is 2^{m-1} or less and assume that the current send window is 0 to $W_S - 1$. Suppose also that the initial receive window is 0 to $W_S - 1$. Now suppose that frame 0 is received correctly but that the acknowledgment

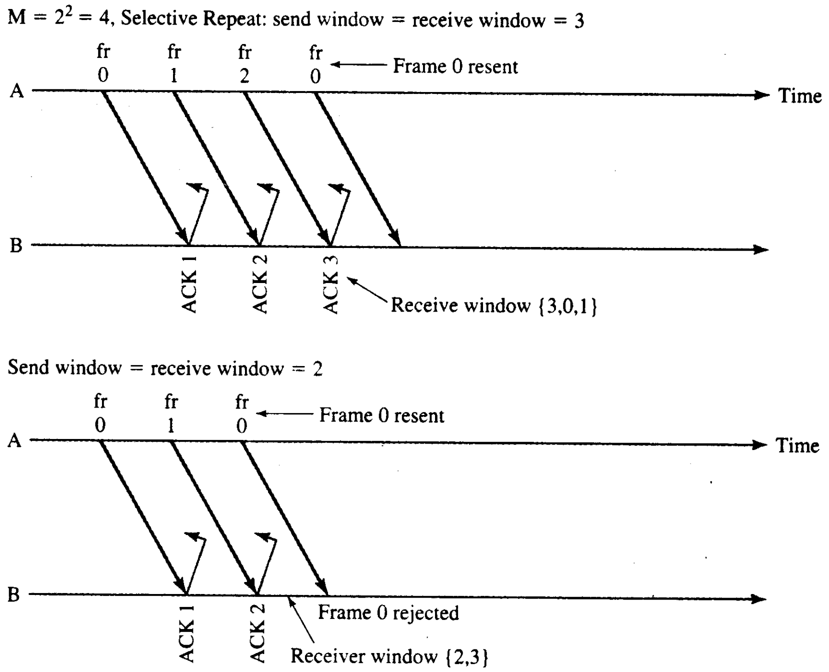


FIGURE 5.23 Maximum window size in Selective Repeat ARQ.

for frame 0 is lost. The transmitter can transmit new frames only up to frame $W_S - 1$. Depending on which transmissions arrive without error, R_{next} will be in the range between 1 and W_S while $R_{next} + W_R - 1$ will be in the range of 1 to $2W_S - 1$. The maximum value of R_{next} occurs when frames 0 through $W_S - 1$ are received correctly, so the value of R_{next} is W_S and the value of $R_{next} + W_R - 1$ increases to $2W_S - 1$. Crucially, the receiver will not receive frame $2W_S$ until the acknowledgment for frame 0 has been received at the transmitter. Any receipt of frame 0 prior to frame $2W_S$ indicates a duplicate transmission of frame 0. Therefore, the maximum size windows when $W_S = W_R$ is 2^{m-1}.

In the next section we develop performance models for the ARQ protocols. We show that Selective Repeat ARQ outperforms Go-Back-N and Stop-and-Wait ARQ. Of course, this performance level is in exchange for significantly greater complexity.

EXAMPLE **Transmission Control Protocol**

The **Transmission Control Protocol (TCP)** uses a form of Selective Repeat ARQ to provide end-to-end error control across a network. TCP is used over internets that use IP to transfer packets in connectionless mode. TCP must therefore contend with the transfer of packets that may arrive out of order, may be lost, or may experience a wide range of transfer delays. We explain how TCP uses ARQ to provide the user of the TCP service with a reliable, connection-oriented stream service in Section 5.3.3.