By Youlu Zheng &
Shakil Akhtar

bandwidth as the straight NRZ encoding. This restricts the applications of a biphase coding scheme from higher speed networks, such as the 100 Mbps FDDI protocol (Chapter 7).

3.1.2 Multiplexing Schemes

A network may consist of millions of computers, and it is virtually impossible to completely connect them. However, complete connectivity may still be provided via sharing the physical links. Multiplexing (MUX) techniques, which allow more than one communications channel to be carried over a single physical connection, are just like the multitasking and time-sharing concepts used in computer science to share CPU resources. There are many elegant methods for multiplexing. In traditional telephone systems that use analog transmission, multiplexing is implemented by means of the frequency division multiplexing (FDM) method. For data communications and networks, time division multiplexing (TDM) allows time-sharing. For digital wireless telephone system, TDM is a maturing technology and code division multiplexing (CDM) is gaining popularity.

3.1.2.1 Frequency Division Multiplexing (FDM)

The idea for the FDM technique originated from radio broadcasting. The air is treated like a huge communications pipe. Different radio wave signals can be transmitted through the air without interfering one another because they use carrier waves with different frequencies. The difference of carrier frequencies is what separates one channel from another at the receiver end. FDM analog signals are transmitted over the physical link at different frequencies in much the same way as the signals for different television stations are transmitted at different frequencies through the air or via a television cable.

For the traditional analog telephone system, twelve 4000 Hz voice channels are grouped together and modulated and mixed (multiplexed) into the 60–108 kHz carrier band. In each 4000 Hz channel, only 3000 Hz (from 300 to 3300 Hz) is used for real signal, while two sidebands of 500 Hz each guard against possible interference between adjacent channels. A low- and a high-frequency filter at the receiver side separate the useful signal from the sidebands and carrier wave. In addition, the filter separates (demultiplexes) the signal from one channel from other channel signals (Figure 3.3).

The next FDM level for the telephone system is the 60-channel supergroup comprising five groups of signals ranging from 312 to 552 kHz. Each group is modulated as a single signal with a 48 kHz bandwidth in a supergroup. Then 5 supergroups (CCITT standard) or 10 supergroups (Bell System standard) are merged into a master group that accommodate 300 or 600 voice channels, respectively.

3.1.2.2 Time Division Multiplexing (TDM)

Digitized signals, such as voice, images, and data, are often transmitted through a time division multiplexing (TDM) system, in which a large data rate connection is divided into many small time slots. TDM is implemented by interleaving portions of each channel in time slots. In TDM, many communications channels share a single connection with a large data rate (sometimes inaccurately called bandwidth), which dedicates an individual band-

Networks for computer scientists and Engineers"

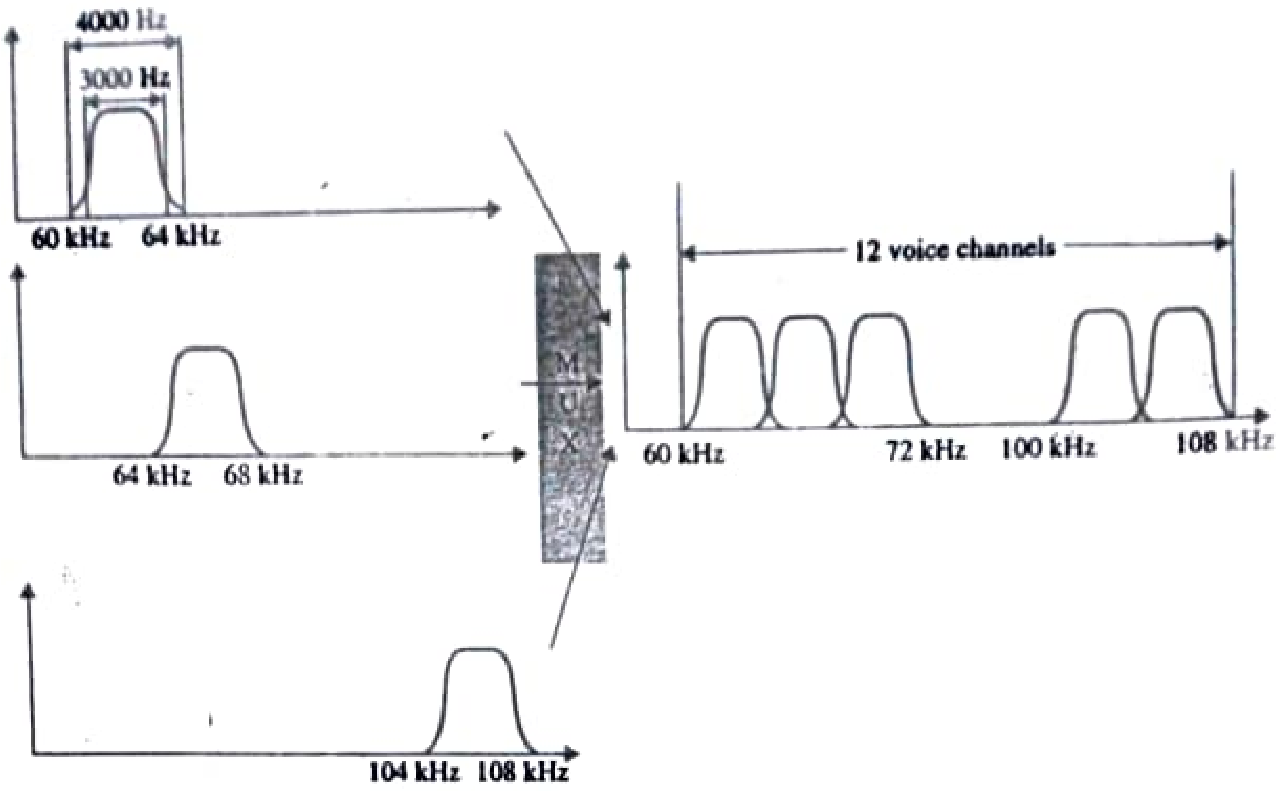

Figure 3.3 FDM divides the transmission bandwidth into smaller bands (subchannels).

width channel to each application or user. The time division multiplexer (MUX) cyclically scans the incoming data from the multiple input ports. Data are "picked" in bits, bytes, or block units and interleaved into frames on a single high-speed communications line. Thus, TDM systems take digital data as input. Consequently a channel bank that performs sampling quantization and encoding, using PCM (pulse coding modulation) to convert the analog signals into digital, is required to interface with the common carrier analog telephone network (Figure 3.4).

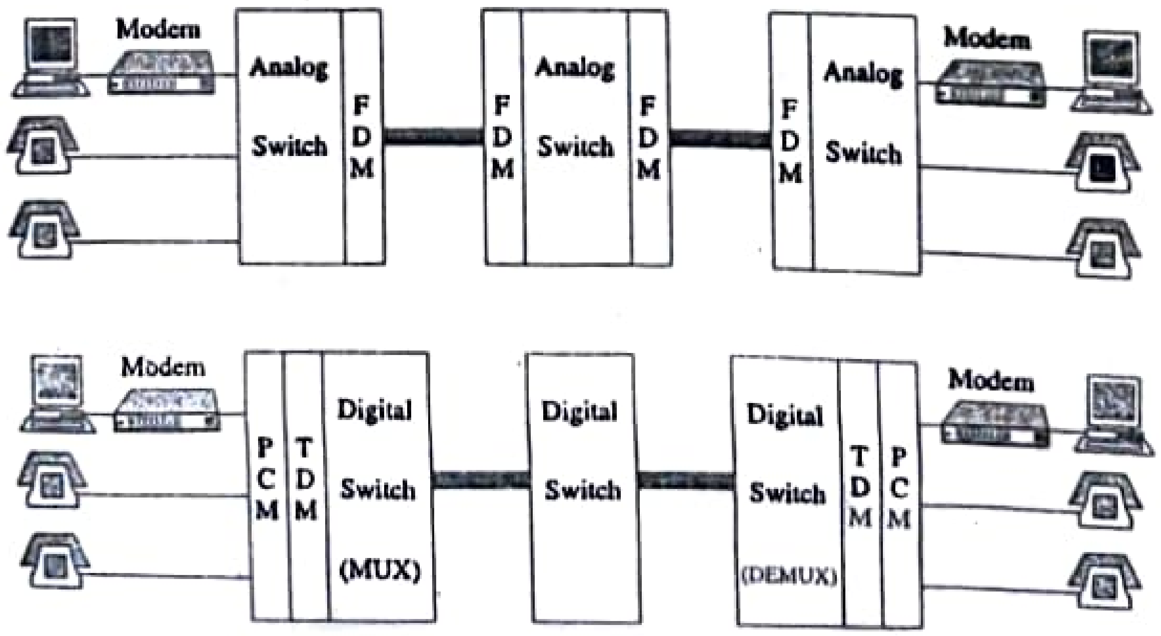

Figure 3.4 Comparison between FDM and TDM.

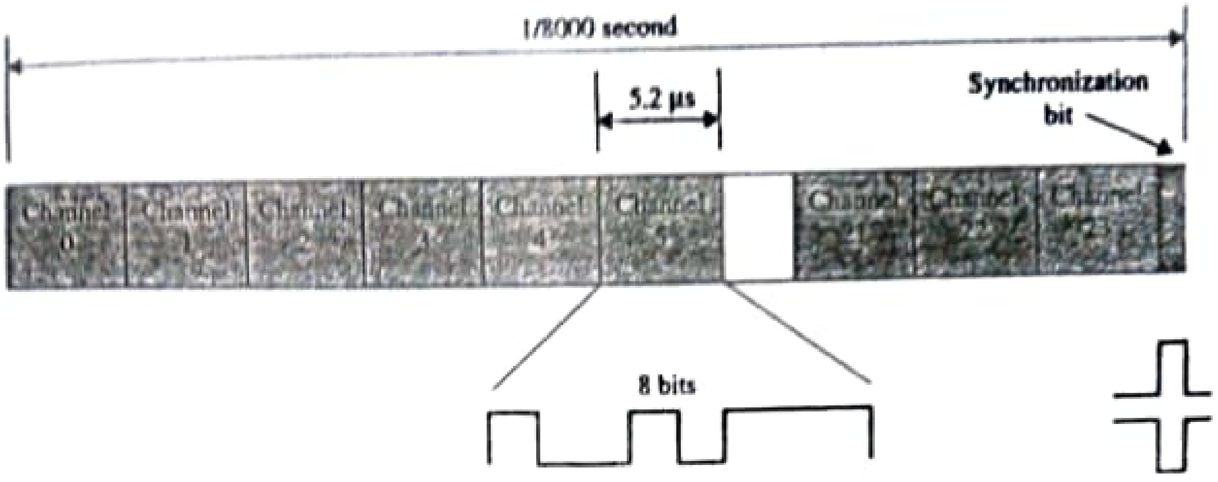

Figure 3.5 A T1 frame interleaves 24 channels of 8 bits each, plus one synchronization bit. The total data rate of the T1 is 8000(8 × 24 + 1) = 1,544,000 bps (1.544 Mbps).

The most widely used TDM system is the North American (AT&T) digital TDM hierarchy. The T1 system, first offered as tariff services in 1983, is at the bottom of the hierarchy. Common carriers transmit 24 voice channels together with a T1 digital channel. Because a voice channel uses a 4000 Hz bandwidth (including the two 500 Hz sidebands), a sampling rate of 8000/s is sufficient to transmit all the information in the original analog signal according to the Nyquist theorem. Each sampling takes 8 bits (256 levels), and 24 channels plus one synchronization bit are combined to form a 193-bit frame. A rate of 8000 fps, therefore, requires a T1 data rate of 1,544,000 bps (1.544 Mbps). In summary, a T1 frame has a duration of 1/8000 second, which contains 24 samplings plus one synchronization bit. Each sampling has a signal width of 5.2 μs and contains 8 bits of data for a single voice channel. Each voice channel has a data rate of 8000 × 8 (bits) = 64 Kbps (Figure 3.5).

The synchronization bit establishes and maintains synchronization between the sending and receiving devices and performs a function similar to the start bit in an asynchronous communications system and a synchronous byte in a synchronous transmission system. The bit alternates as 1 and 0 in each succeeding frame and forms the pattern 10101010 . . . , which can be easily scanned by the receiver to resynchronize when necessary. When 7-bit data format is used, the eighth bit is used for signaling and control, resulting in a 56 Kbps data rate. The formal name of T1 is DS-1, while a digitized voice channel (64 Kbps) is called DS-0. Two DS-1 lines can be combined to form a DS-1C (T1C) channel (3.152 Mbps), and four DS-1 lines are combined to form a DS-2 (T2) channel (6.312 Mbps). After that, 28 DS-1 lines or 7 DS-2 lines form the DS-3 (T3) channel (44.736 Mbps), and 6 DS-3 channels are multiplexed into a DS-4 (T4M) trunk (274.176 Mbps) to provide 4284 voice channels of 64 Kbps each (Figure 3.6).

When fewer than 24 voice channels are needed, fractional T1 service is provided. In Europe and China, the basic carrier data rate corresponding to T1 is E1, with a 2048 Kbps data rate. Furthermore, frame length, placement of the synchronization bit, and synchronization bit functions are also different. These differences highlighted the serious need for integrating international digital networks and resulted in the ISDN (integrated services digital network) standard (Chapter 6).

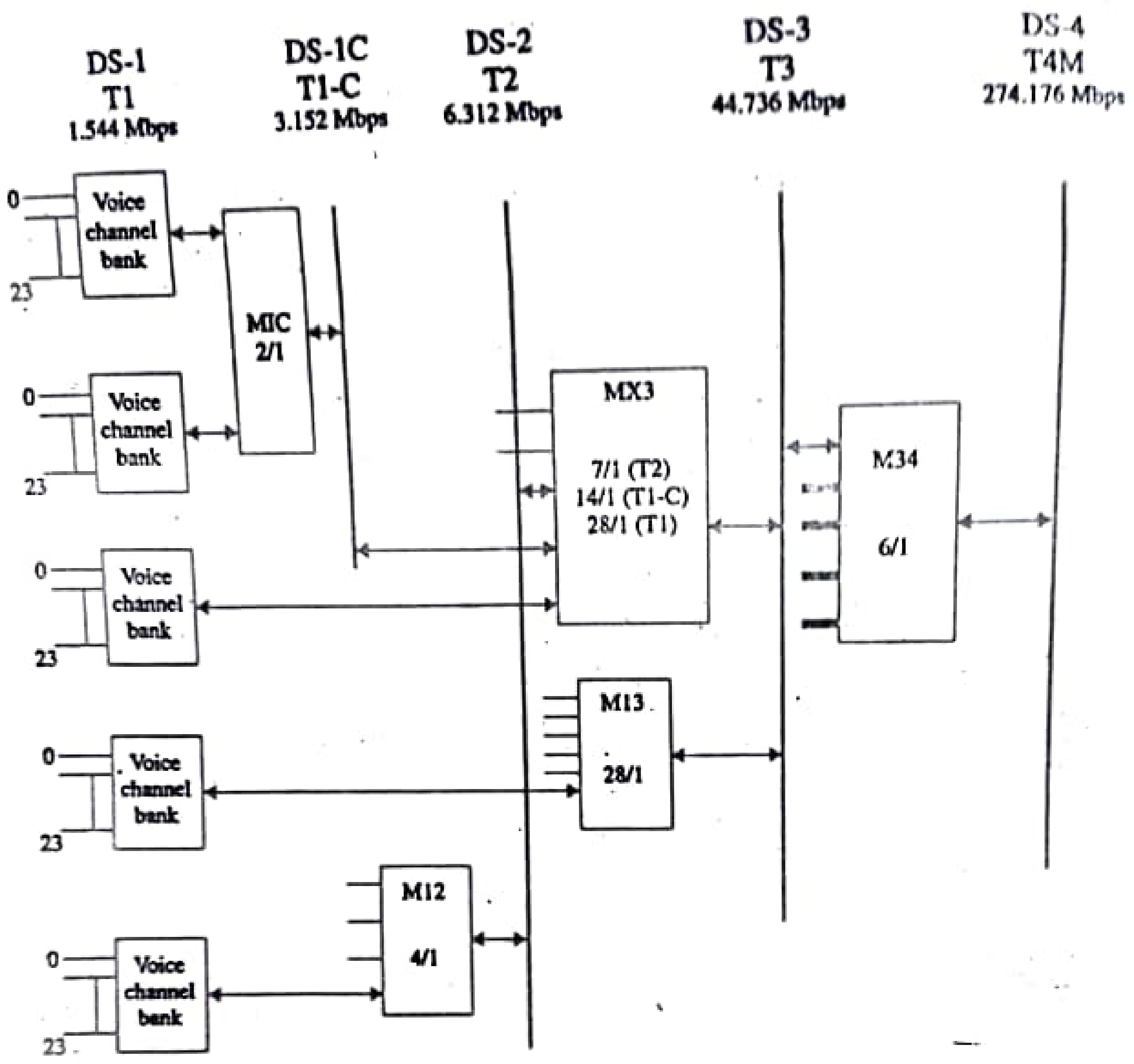

Figure 3.6 The North American digital hierarchy: DS-01, DS-1C, DS-2, DS-3, and DS-4M.

A multiplexer sends out control signals to input devices to keep the data streams synchronized. The control signals are generally issued from a master clock in the MUX. Because different communication devices operate with different physical transmission times and delays, the MUX may provide internal buffers for the time-independent data streams to allow the data to arrive at random intervals, to be multiplexed later into the high-speed link. This technique, called isochronous TDM, provides elastic buffers that can expand or contract to accommodate dynamic traffic flows.

TDM may be used together with FDM: that is, FDM may be used to multiplex multiple channels, which are then merged into a larger TDM channel. Like FDM systems, the conventional TDM wastes the bandwidth of the communication channel because time slots in the frames are often unused. For example, vacant slots occur with idle terminals and conversation gaps. Various efforts have been made, and some variations of TDM systems, such as the *statistical TDM*, have been developed to improve channel utilization. However, because of the bursty nature of communications, much expensive bandwidth was wasted carrying idle patterns in case of data and silence in case of voice. Furthermore, these channels represented strict boundaries that limited the performance of such high-bandwidth traffic as image transfer, LAN, and system network architecture (SNA) (Section 3.3) users. As communications became even more bursty, widespread, and mission critical, it became clear that the original TDM architecture was not quite appropriate for the new networking requirements.

3.1.2.3 Code Division Multiple Access (CDMA)

Thanks to the availability of very low cost, high-density digital integrated circuits, which reduce the size, weight, and cost of communication devices, an "old' multiplexing technology—code division multiplexing access (CDMA), proposed 40 years ago—is finding new life in cellular telephone communications.

CDMA has been used by the military in antijamming devices and to ensure secure communications because the spread-spectrum signal used by CDMA is difficult to detect or jam, and resists interference. An important issue for any cellular telephone system is its multiple-*access* capacity, meaning support for multiple, simultaneous users. That is, in cellular communications a large number of users must be able to share a common pool of radio channels, and any user can gain access to any channel, although a user is not always assigned to the same channel. A channel can be thought of as merely a portion of the limited radio resource, which is allocated, temporarily, for a specific phone call. CDMA uses an innovative multiple-access method to define how the radio spectrum is divided into channels and how channels are allocated to each user of the communication system.

The CDMA approach uses coding theory and complicated algorithms to spread information about multiple channels over the entire frequency spectrum. A CDMA cellular phone call starts with a standard rate of 9600 bps. It then spreads to a transmitted rate of about 1.23 Mbps. "Spreading" means that digital codes are applied to the data bits associated with users in a cell. These data bits are transmitted along with the signals of all the other users in that cell. When the signal is received, the codes are removed from the desired signal, separating the users and returning the call to a rate of 9600 bps. These codes, which are shared by both the mobile station—the cellular phone—and the base station, are called pseudo-random code sequences. All users share the same range of the radio spectrum.

Now we explain CDMA with the help of an example. Assume that a bit of 1 or 0 is represented by a vector (code) c_i or $\bar{c}_i$ for channel i to be multiplexed using CDMA. All the channels to be transmitted are *orthogonal* with respect to one another. This can be mathematically expressed as follows:

$$c_i \cdot c_j = \frac{1}{n}\sum_{k=1}^{n} c_{ik}c_{jk} = \begin{cases} 1 \text{ if } i = j \\ 0, \text{ otherwise} \end{cases} \tag{3.1}$$

The operator "·" is called the inner product. One way to implement the orthogonal relation is to use bipolar coding, that is, to use a special sequence of -1 and $+1$—vector or code—to represent a single bit of signal for each channel. For code $\bar{c}_i$, we simply replace every -1 with $+1$ and replace every $+1$ with -1 in the sequence for c_i. Apparently

$$c_i \cdot \bar{c}_j = \begin{cases} -1 \text{ if } i = j \\ 0, \text{ otherwise} \end{cases} \tag{3.2}$$

To preserve orthogonality for any pair of vectors from different channels, the number of pairs of identical elements must equal the number of pairs of different elements. Adding all the code sequences arithmetically will simultaneously transmit multiple channels. For channel i to transmit a 1, the code sequence for c_i is simply transmitted, and to transmit a 0 the code sequence for $\bar{c}_i$ is transmitted. To retrieve signals for channel i from signal S, where

$$S = C_1 + C_2 + \cdots + C_i + \cdots$$

and $C = c_i$ or $\bar{c}_i$, we find the inner product of c_i and S. Based on Equations (3.1) and (3.2),

$$c_i \cdot C_j = 0, \text{ when } i \neq j$$

we have

$$C_i \cdot S = c_i \cdot C_i = \begin{cases} +1, & \text{if } C_i = c_i \\ -1, & \text{if } C_i = \bar{c}_i \\ 0, \text{ if channel } i \text{ is silent} \end{cases}$$

In the following example, we use the vectors (+1, +1, +1, +1), (−1, −1, +1, +1), (−1, +1, −1, +1), and (−1, +1, +1, −1) to represent 1 for channels 1, 2, 3, and 4, respectively. Thus 0s are expressed as (−1, −1, −1, −1), (+1, +1, −1, −1), (+1, −1, +1, −1), and (+1, −1, −1, +1), respectively.

	Bit 1	Bit 0
Channel 1	(+1, +1, +1, +1)	(−1, −1, −1, −1)
Channel 2	(−1, −1, +1, +1)	(+1, +1, −1, −1)
Channel 3	(−1, +1, −1, +1)	(+1, −1, +1, −1)
Channel 4	(−1, +1, +1, −1)	(+1, −1, −1, +1)

Note that all the channels are orthogonal from one another, for example,

$$c_1 \cdot c_3 = \frac{-1 + 1 - 1 + 1}{4} = 0$$

$$c_2 \cdot c_4 = \frac{-1 + 1 - 1 + 1}{4} = 0$$

while

$$c_1 \cdot c_1 = \frac{+1 + 1 + 1 + 1}{4} = 1$$

$$c_4 \cdot c_4 = \frac{-1 - 1 - 1 - 1}{4} = -1$$

Suppose that at a given moment, channel 1 is transmitting 0, channel 2 is transmitting 1, channel 3 is not transmitting (silent), and channel 4 is transmitting 0. (Note that transmitting a signal 0 is different from not transmitting at all.) The resulting signal S would be

$$S = (-1, -1, -1, -1) + (-1, -1, +1, +1) + (0, 0, 0, 0) + (+1, -1, -1, +1) = (-1, -3, -1, +1)$$

and we have

$$c_1 \cdot S = \frac{(+1, +1, +1, +1) \cdot (-1, -3, -1, +1)}{4} = -1$$

$$c_2 \cdot S = \frac{(-1, -1, +1, +1) \cdot (-1, -3, -1, +1)}{4} = +1$$

$$c_3 \cdot S = \frac{(-1, +1, -1, +1) \cdot (-1, -3, -1, +1)}{4} = 0$$

$$c_4 \cdot S = \frac{(-1, +1, +1, -1) \cdot (-1, -3, -1, +1)}{4} = -1$$

Increased privacy is inherent in CDMA technology. CDMA phone calls will be secure from the casual eavesdropper since, unlike an analog conversation, a simple radio receiver will not be able to pick individual digital conversations out of the overall RF radiation in a frequency band.

In the final stages of the encoding of the radio link from the base station to the mobile unit, CDMA adds to the signal a special "pseudo-random code" that repeats itself at finite intervals. Base stations in the system distinguish themselves from each other by transmitting different portions of the code at certain times. In other words, the base stations transmit time-offset versions of the same pseudo-random code. To assure that each time offset used remains unique, CDMA stations must remain synchronized to a common time reference.

This precise common time reference is provided by the Global Positioning System, the satellite-based radio navigation system discussed in Chapter 2. GPS is capable of providing a practical and affordable means of determining continuous position, velocity, and time to an unlimited number of users. For station identification mechanisms to work reliably and without ambiguity, base stations must be synchronized within a few microseconds. Any convenient mechanism can be used for this purpose, but the system was designed under the assumption that the GPS itself would be used. GPS is a system made of a family of low-earth-orbit satellites that broadcast a spread-spectrum signal and ephemeris information from which a sophisticated Kalman filter algorithm in a receiver can derive both a very accurate position and a very accurate time.

CDMA (and spread spectrum in general) was always dismissed as unworkable in the mobile radio environment because of what was called the "near–far problem." It was always assumed that all the stations transmitted constant power. In the mobile radio environment, some users may be located near the base station while others are far away. The propagation path loss difference between the extreme users can be many tens of decibels.

If there is, say, a 30 dB difference between the largest and smallest path losses, then there is a 60 dB difference between the signal-to-noise ratio (SNR) of the closest user and the farthest user, because these are the received powers. To accommodate the farthest users, the spreading bandwidth would have to be perhaps 40 dB, or 10,000 times the data rate. If the data rate were 10,000 bps, then $W = 100$ MHz. The spectral efficiency is abysmal, far worse than even the most inefficient FDMA or TDMA system. Conversely, if a more reasonable bandwidth is chosen, remote users will receive no service.

This observation was, for years, the rationale for not even attempting any sort of spread spectrum in any but geosynchronous satellite environments, where the path loss spread was relatively small.

The key to the high capacity of commercial CDMA is extremely simple. If, rather than using constant power, the transmitters can be controlled in such a way that the received powers from all users are roughly equal, then the benefits of spreading are realized. If the received power is controlled, the subscribers can occupy the same spectrum, and the hoped-for benefits of interference averaging accrue.

Maximum capacity is achieved if we adjust the power control so that SNR is exactly what it needs to be for an acceptable error rate. [If we set the SNR in Equation (2.8) to the target SNR, we can find the basic capacity equation for CDMA.]

3.1.2.4 Wave Division Multiplexing (WDM)

Dense WDM switches give users full system flexibility and use of conventional fiber for very high-speed transmission. Each optical channel is capable of carrying information over a single optical fiber at a rate of up to 10 Gbps. For a 128-channel dense WDM switch at full capacity, each fiber pair can deliver 1.28 terabits per second (tbps) bandwidth, the rough equivalent of 16 million simultaneous telephone calls.

Wave division multiplexing is similar in principle to frequency division multiplexing. With WDM, discrete colors of light become high-bandwidth channels for carrying data over a strand of fiber. WDM allows multiple channels to share the same fiber connection and boosts the speed and capacity of fiber optic links without installing more fibers to provide high-speed interconnection among widely scattered sites. WDM is also important as a means of setting up multiple data centers, and for purposes of disaster recovery and CPU redundancy.

The primary value of WDM lies in expanding the capacity of existing fiber networks without laying more fiber cables. All the user has to do is install wave division multiplexers at both ends of existing fiber spans. This is a step up from older fiber optic systems that require a fiber pair for each channel. WDM is particularly popular among carriers that otherwise would have to install thousands of miles of new fiber to boost capacity. In many situations, running new fiber is practically impossible because of rights of way.

WDM requires no repeaters over a 6-mile span. WDM switches are protocol independent. You can send ATM through them, FDDI, or any other type of frame. The multiplexers do not open up the packets or cells passing through them. They just send units down the line at up to 200 Mbps per channel. WDM might be useful for enterprise applications in which high bandwidth is essential, such as linking dual corporate data centers, multilocation manufacturing businesses, hospitals, and financial institutions. Some enterprise users consider WDM even if they do not have an existing fiber network.

3.2 DATA LINK LAYER

3.2.1 Asynchronous and Synchronous Communications

In a computer system including the I/O subsystems, most data transfers occur in parallel. For instance, the transfers from random access memory (RAM) to CPU, and from disk to memory, happen in parallel. Parallel transmission is fast because it allows 8, 16, 32, and 64 bits of data to be transferred at once. However, parallel transmission needs multiple circuits for the interface of transmitter and receiver. In addition, the expense of running the multiple-conductor cables over long distances is prohibitive. Therefore, parallel transmission is appropriate for short distance.

In contrast to computer systems, in digital communications and in network environments data are transmitted almost exclusively serially (i.e., over links in a single communications channel instead of a set of parallel lines). Many more devices may be connected to a serial link than to a parallel bus. Also, a serial connection may continue across multiple media (e.g., copper to fiber optic, then to copper), which is virtually impossible with a parallel bus. In a serial communications system, signals are sent through the communications line one at a time. When nonbinary logic is used, more than one bit of data may be transmitted at once by a serial communications system like the quadrature amplitude modulation (QAM) modem, described in Chapter 2.

Asynchronous serial transmission, the simplest communications format, is found primarily in point-to-point communications systems. A character or a byte of data is the unit of transmission between two devices. An asynchronous transmission system copes with timing and synchronization problems by splitting a character into a fixed number of bits and sending one character, or byte of data, at a time. To keep the communications synchronized, extra bits—a start bit and one or two stop bits—are added to the original data, creating a small frame. A single optional parity bit maybe added to the data, inserted between the data bits and the stop bit, for error checking. This parity bit may be

- Even parity: making the number of 1s in the 8 bits always an even number.
- Odd parity: making the number of 1s in the 8 bits always an odd number.
- Mark: always 1.
- Space: always 0.
- No parity: this bit is part of the data.

When the parity bit is used, at most 7 bits of data can be sent at a time. Parity check is performed on the data bits only, not the start and stop bits. Because 2 or 3 bits may be required to form the frame for each character (7 or 8 bits), a synchronous transmission is not very efficient. However, asynchronous communication requires little hardware support and has the advantage of simplicity. This mode is still often used in software for communications of microcomputers to host computers (e.g., Kermit). The most often used format is *N-8-1-1*, meaning no parity, 8 data bits, one start bit, and one stop bit. Figure 3.7 shows how the character G is transmitted in an asynchronous communications system. Note that in such a system, the least significant bit is transmitted first.

Synchronous transmission is more efficient and is appropriate for the fast serial transmission available in most high-speed digital networks. In a synchronous system, a stream

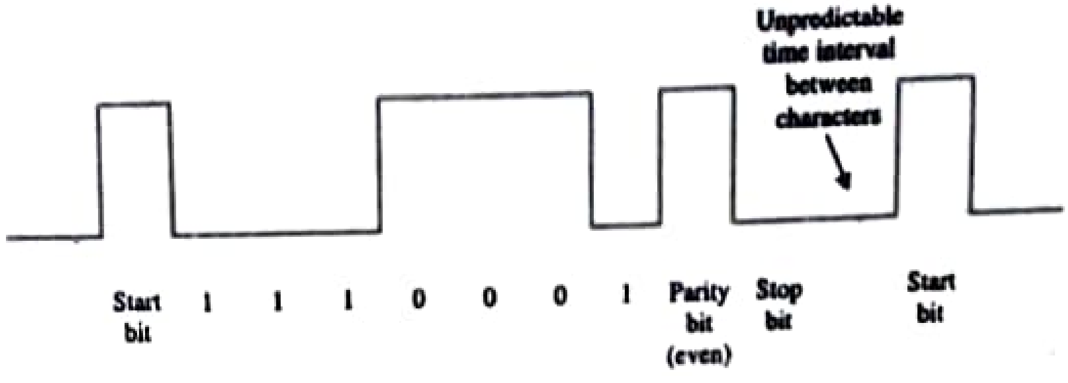

Figure 3.7 The ASCII character G (47_{16}) is transmitted in E-7-1-1 asynchronous transmission format with one start bit, one even parity bit, and seven data bits. The least significant bit is transmitted first. The zeros represent high voltage.

of bits or blocks of characters are transmitted without a start and stop bit for every character. Exact timing and synchronization are archived in three ways at different levels:

• A synchronization signal and a clock may be embedded inside the data signal and encoding mechanism as discussed before. Biphase encoding, such as Manchester and differential Manchester coding, is a good example of such synchronization schemes.

• Separate clock signal lines may be included in the interface between the DTE and DCE devices. Recall the EIA standard 232-D for serial line communications (Chapter 2). Pin 15 (or 24), transmit clock (TC) and pin 17 receive clock (RC) may be used along with pin 2, transmit data (TD) and pin 3, receive data (RD) respectively, to synchronize the transmission and reception of the data frame. Note that the synchronization is not between the transmitting modem and the receiving modem, but between the transmitting computer and its modem, and between the receiving computer and its modem. These clock signal lines are not used in asynchronous communications.

At the higher level, for a block of data (bits or characters), a preamble and postamble bit patterns or flags can be added in the front and end of the data block to form a frame. These flags contain control/status information and the information about the enclosed data pertaining to the sender and receiver. The data block may be bit-oriented or character-oriented, as we discuss in detail shortly. The exact format of the frame and the preamble/postamble depends on the network environment and communications protocols used. Usually, these starting and ending flags are multiple of 8 bits. For example, an Ethernet frame has a preamble 8 bytes long.

For large blocks of data, synchronous transmission is far more efficient than asynchronous. For example, for an Ethernet frame having a maximum length of 1500 bytes, the overhead for synchronization is only 8/1500.

3.2.2 Error Detection and Correction

Electromagnetic waves traveling over a transmission medium may encounter noise. A very common example is a lightning surge, which may cause corruption and loss of data. We see the effect of noise due to lightning, operation of common household appliances, and firing spark plugs from a passing vehicle as streaks and blank screens on our television screens. Both analog and digital signals are subject to noise that may result in changes to data and/or data loss. In computer communications, changing a bit may have severe effects on data. For example, the 7-bit binary ASCII code of character a is 1100001. If the code is changed to 1100011 during transmission, the data string no longer represents an a, but now represents a c. The presence of such errors may have adverse end effects on the data. A file containing the financial records of a company may have figures such as $1.# million dollars instead of the expected numeric value $1.3. Notice that this happened because of only a single incorrect bit (0100011 for # instead of 0110011 for 3).

Fortunately, single-bit errors are the most common type in data communication designs featuring simple error detection and correction techniques. However, multiple-bit errors or burst errors are possible too. A multiple-bit error would involve changing two or more bits during a transmission. So, for example, a 1100001 may change to 1000000. Burst errors are caused when a noise interferes with the transmission for a longer period of time, possibly causing a change in several consecutive bits. For example, if the sending bit pattern is 1000 0111 1001 1100, a burst error involving bits 3–9 (bit 0 being the rightmost bit) would result in the following received string: 1000 0100 0110 0100.

To avoid passing the effects of bit errors to higher OSI layers, the data link layer must detect any errors in a received message. Or, preferably, the bit error is fixed subsequent to the detection, thus correcting the error. One simple mechanism of error detection would be to transmit every bit twice. As soon as the receiver finds a pair of bits to be unequal, it can flag an error to higher layers. However, this redundancy technique effectively reduces the transmission efficiency by half, since there is one overhead bit for every bit transmitted.

A more practical method would be one for which overhead is less. Mainly, there are three redundancy-based techniques discussed in this section. The parity-based techniques append a single bit to a block of data to indicate the total number of bits; the parity bit can be even (for even parity) or odd (for odd parity). The second technique is based on the calculation of arithmetic checksums, using the binary ones-complement arithmetic. The calculated checksum is appended to the end of the message, and the sum of the data and the checksum is obtained upon receipt of the message. If the resulting sum is all 1s, then the message is considered to be error free; otherwise, there is an error in the message. The third redundancy method is based on using binary division of polynomials in modulo 2 to calculate the cyclic redundancy checksum (CRC).

Error detection at the data link layer is usually sufficient, and the frames in error may be retransmitted by using a protocol between the sender and receiver. However, some codes are designed to correct errors at the receiver end without need for retransmission. In one such code, called Hamming code, the even parity of specific bits in the data is included in such a way that in case of a single bit error, the checksum bits obtained at the receiver point to the erroneous bit.

3.2.2.1 *Parity*

Parity-based redundancy systems use either odd or even parity. The parity bit is obtained based on the count of 1s in the data block. The number of 1s, including the parity bit in the encoded data, must be even for even parity. For example, even parity of the ASCII character c is 1, since ASCII a is 1100001. Single-bit parity for a data block may not de-

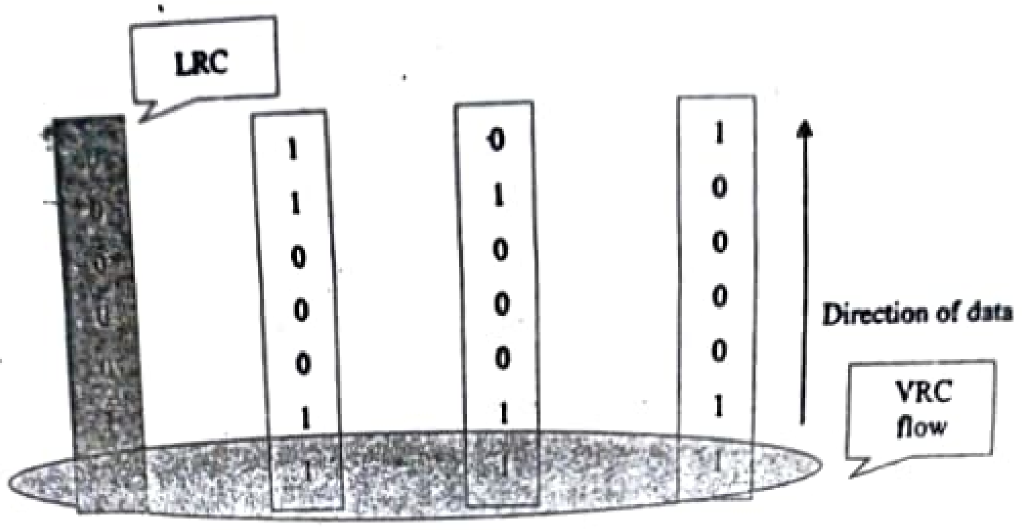

Figure 3.8 Parity using LRC and VRC.

tect an even number of errors (i.e., errors in even numbers of bits may go undetected). If our example of ASCII a, the number of errors had been odd (say the received bit pattern, with the leftmost bit as even parity, had been 11011001), the error would have been detected because of the odd number of 1s. To solve the problem of undetected errors when the number of errors is even, a two-dimensional parity mechanism may be used. Here the even parity for each vertical block of data is called the vertical redundancy check (VRC). Parity in the second dimension, called the longitudinal redundancy check (LRC), provides a double check. For transmitted data consisting of the ASCII character set abc, the corresponding check is shown in Figure 3.8.

The VRC/LRC technique shows how even parity in two dimensions may be used to accomplish single-bit error detection. Another redundancy technique that is based on even parity and allows for error correction is called Hamming code. In this code the number of redundancy bits required to correct any single error in m data bits is termed p. The resulting code is $m + p$ bits long. This code may be applied to data blocks of any length. However, p increases rapidly with the number of data bits. In general, $2^p \geq m + p + 1$. So for an ASCII character of seven bits ($m = 7$), p must be at least 4 bits, resulting in a total of 11 bits. The check bits (or redundancy bits) are positioned at 1, 2, 4, 8, . . . , 2^n, where n is an integer. The position number starts at 1 and continues until all the positions have been filled with either data bits or parity bits. Figure 3.9 shows the positions of bits in a 7-bit ASCII character.

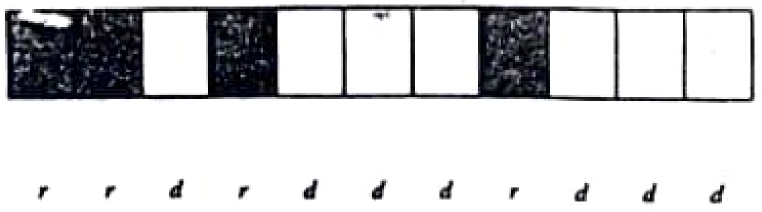

Figure 3.9 Positions of parity and data bits in Hamming code.

The r bits are obtained as even parity of d bits at specific positions. Actually the position numbers to be included in determining the parity is obtained by breaking down the d-bit positions as powers of 2:

$$
\begin{aligned}
3 &= 1 + 2 \\
5 &= 1 + 4 \\
6 &= 2 + 4 \\
7 &= 1 + 2 + 4 \\
9 &= 1 + 8 \\
10 &= 2 + 8 \\
11 &= 1 + 2 + 8
\end{aligned}
$$

and so on

The r bit at position 1 is obtained by locating 1s in the foregoing breakdown. Wherever a 1 appears at the right-hand side, the corresponding data bits are used in determining the r bit. We see that r at position 1 is obtained as even parity of data bits at positions 3, 5, 7, 9, and 11. Similarly for r at position 2, we locate 2s at the right-hand side. The data bits used in this case will be at positions 3, 6, 7, 10, and 11. The process is repeated for the r bits at positions 4, 8, and so on.

Now let us apply the technique to get the r bits for the 7-bit data string 1 1 0 0 0 0 1 with respective positions at 3, 5, 6, 7, 9, 10, and 11. Put together, it will be 1(3) 1(5) 0(6) 0(7) 0(9) 0(10) 1(11), where the numbers in parentheses are the bit positions. The r bit at position 1 will be an even parity of 1, 1, 0, 0, and 1, giving a 1. Similarly, the even parity at position 2 will be an even parity of 1, 0, 0, 0, and 1, giving a 0. The r bits at positions 4 and 8 will both be 1s. The resulting bit sequence will be (1) (0) 1 (1) 1 0 0 (1) 0 0 1, in which the r bits are in parentheses.

Error correction at the receiving end is done by calculating c bits, which are r bits as obtained just shown, but this time the bit at position r is also included in the calculation. Let us say that the received bit pattern is 1 0 0 1 1 0 0 1 0 0 1. Notice that there is an error at position 3. The c bit at position 1 will be an even parity of bits at positions 1, 3, 5, 7, 9, and 11 (i.e., even parity of 1, 0, 1, 0, 0, 1 equal to 1). Similarly the c bit at position 2 will be a parity of 0, 0, 0, 0, 0, 1 equal to 1, and the c bit at position 4 will be a parity of 1, 1, 0, 0 equal to 0. The c bit at position 8 will be a parity of 1, 0, 0, 1 equal to 0. Now arranging the c bits as positions 8, 4, 2, and 1, we get 0 0 1 1, which is the binary combination of 3 pointing to the erroneous bit in the received message.

Notice that the Hamming code technique as presented here works well as long as there is only one bit in error. The method must be modified to correct multiple errors in the data, resulting in higher number of r bits. The main idea is to produce enough overlap in the parity bit calculations to allow error correction to take place at the receiving end. Since, however, the number of r bits increases dramatically to allow multiple error correction, this method is seldom practical for multiple error correction.

3.2.2.2 Arithmetic Checksum

In the arithmetic checksum method, the sender divides the sending data unit into equal segments of n bits. Then ones-complement arithmetic is used to add the segments together to get the result in n-bit form. This sum is complemented and appended to the data as the checksum field. The two fields, being complements of each other, must give all 1s when added together.

The receiver keeps receiving the n-bit data units and adds them to the checksum received at the end. If there is no error, the result will be all 1s. In the event of an error, the data unit is rejected. As an example, consider four 4-bit data units as 1000 (unit 1), 1101 (unit 2), 0101 (unit 3), and 1110 (unit 4). The checksum for these would be obtained by subsequent one complement additions, in which the final carry is added to the resulting binary sum: 1000 + 1101 = 0110, followed by 0110 + 0101 = 1011, followed by 1011 + 1110 = 1010. For an error-free transmission, the resulting arithmetic checksum should be 0101, which gives all 1s when added to the last sum, 1010. Notice that if there are errors in any of the received data units or the checksum, the sum results will not give all 1s.

3.2.2.3 *Cyclic Redundancy Checksum (CRC)*

Like other data link layer error detection methods, the CRC of data is generated at the transmitter end by means of a hardware mechanism that involves sequential circuits using shift registers and flip-flops. A modulo-2 division is implemented by means of a generator polynomial $G(x)$ of degree n over the message polynomial $M(x)$ representing the message. The n-bit frame check sequence or the checksum is the remainder modulo-2 division of $M(x)$ by $G(x)$. However, n bits of 0 are appended at the end of $M(x)$.

As an example, consider $M(x) = X^7 + X^6 + X^5 + X^2 + X$, where $M(x)$ represents the message bit sequence 1 1 1 0 0 1 1 0. For $G(x) = X^4 + X^3 + 1$, the degree is $n = 4$, resulting in the division of 1 1 1 0 0 1 1 0 0 0 0 0 by 1 1 0 0 1. Figure 3.10 illustrates the division. The 4-bit remainder is obtained as 0 1 1 0, which when appended to the message gives us 1 1 1 0 0 1 1 0 0 1 1 0.

The CRC hardware implementation using $G(x) = X^4 + X^3 + 1$ consists of two exclusive-or gates and a couple of shift registers, as illustrated in Figure 3.11. Initially, the content of the shift registers is 0. When the data stream to be transmitted is coming sequentially, the results of the exclusive-or gates are pushed back into the shift registers one bit at a time. For each bit transmitted, the shift registers shift left one bit. At the end of the operation, the remainder is kept in the shift registers, which is attached to the end of the data (payload) and transmitted without change.

At the receiver side, the same device is used. The shift registers are set to zero initially. The incoming data is fed into the exclusive-or gate and pushed back into the shift registers through the exclusive-or gates. The difference is that after the incoming data including the attached CRC remainder is received, the shift registers contain all 0s. Otherwise, transmitting errors occurred.

3.2.3 Framing and Flow Control

Asynchronous transmission consists of the movement of short sequences of fixed-length data. To correctly recover the data, synchronization is needed between the sender and the receiver, and since the data length is fixed, clocks of reasonable tolerance can be used to achieve it. This simple synchronization mechanism does not work well for long data se-

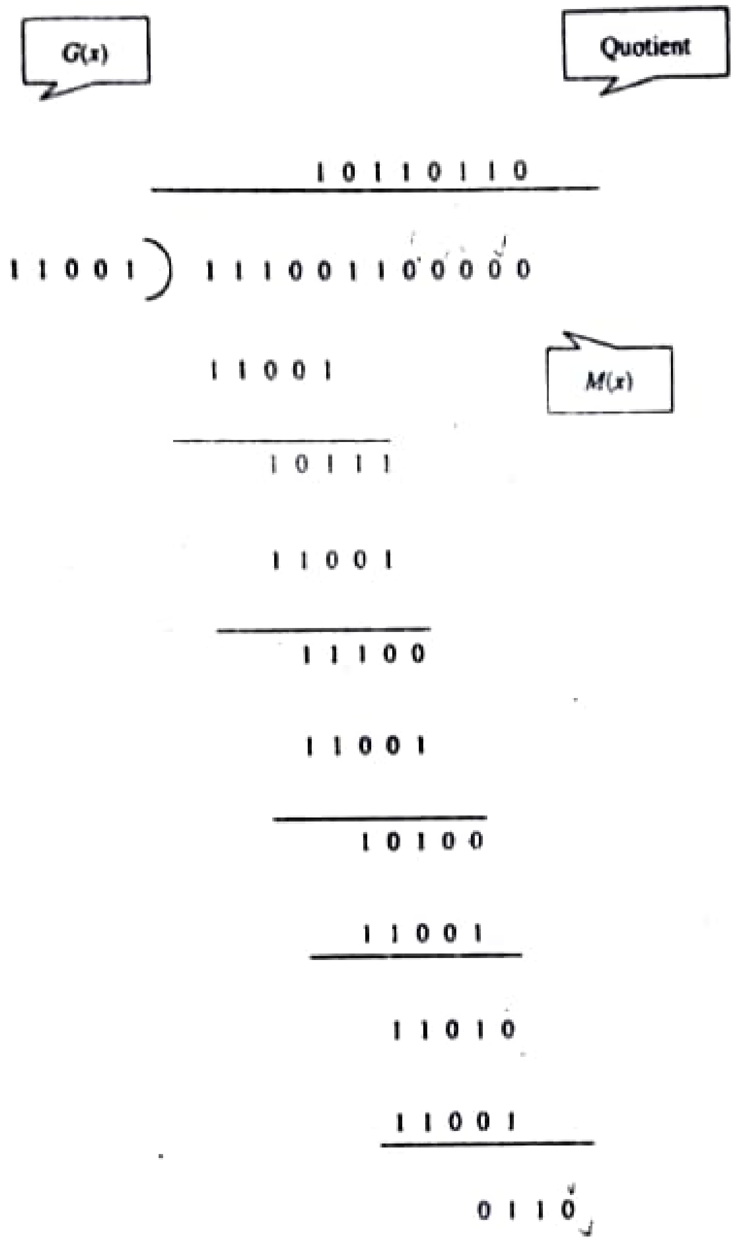

Figure 3.10 Evaluation of CRC using modulo-2 division.

quences, however. Thus in practice, sequences of bits called packets are used to increase transmission efficiency. The transmission of long packets requires improved synchronization of the receiver, which in turn requires special framing structures. The frames help indicate the start and end of packets for the receiver, allowing an arbitrarily long sequence of data. Two types of transmission may be done with a built-in synchronization and flow control mechanism: bit-oriented and character-oriented transmission. The two methods differ in the way they frame the bits. Bit-oriented transmission uses bit sequences to represent characters, and character-oriented transmission uses characters.

The bit-oriented transmission mechanism is illustrated in Figure 3.12. A special bit pattern 0 1 1 1 1 1 1 0 called a flag is used to indicate the frame's start and end. The frame contains the address, control, CRC, and data, in addition to the start and end flags. The length of each field depends on the protocol used. High-level data link control (HDLC), a popular bit-oriented protocol, is presented later in Section 3.2.4.

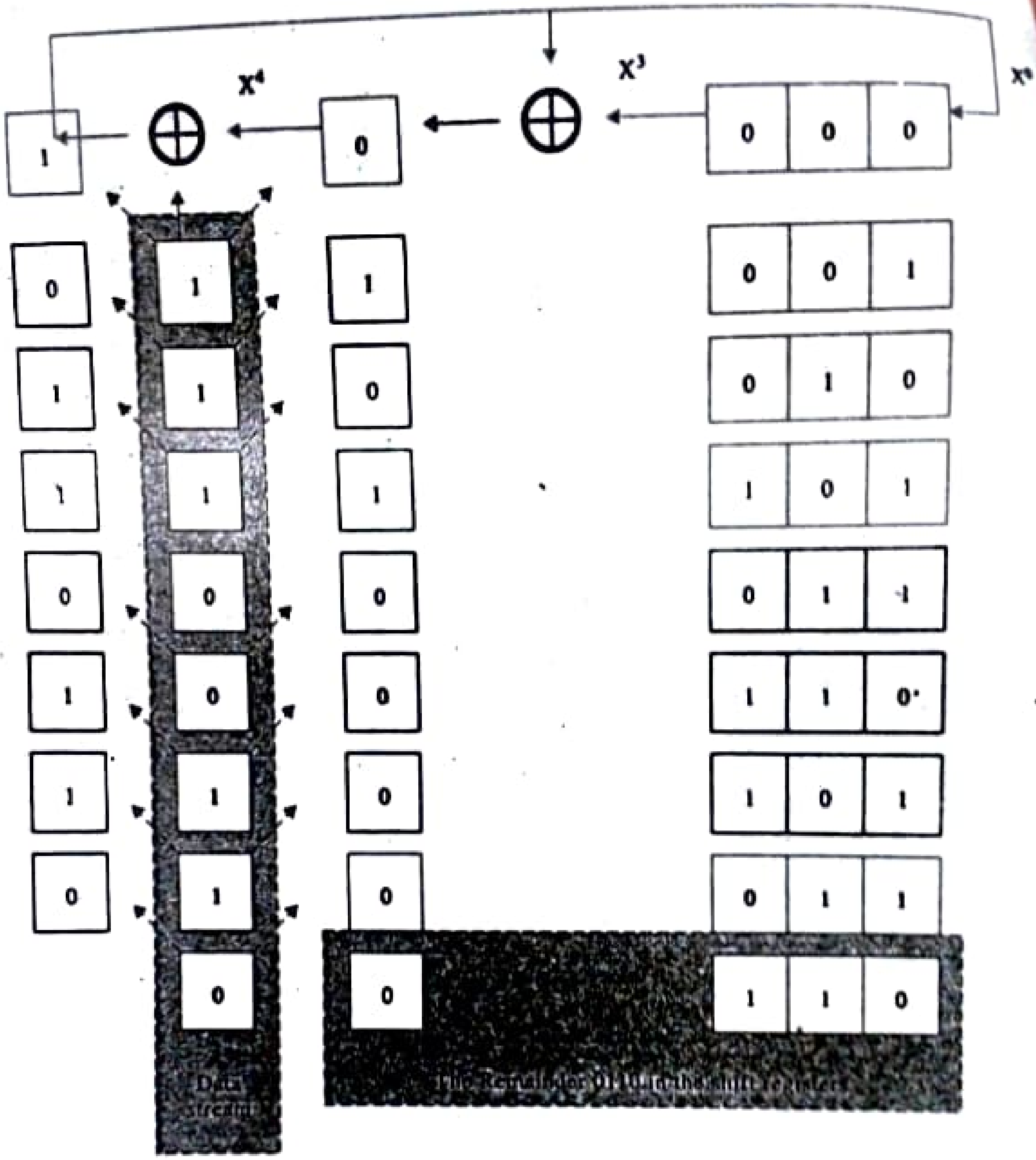

Figure 3.11 The CRC hardware implementation using $G(x) = X^4 + X^3 + 1$.

In character-oriented transmission, the packets are assumed to consist of an integral number of (8-bit) bytes. A typical frame structure is shown in Figure 3.13. The frame start is indicated by a special synchronization character SYN followed by a DLE (data link escape) character and an STX (start-of-text) character. The end of the frame is indicated by DLE, followed by an ETX (end-of-text) character. Two bytes of CRC and a byte of SYN are appended at the end to complete the frame.

Flag	Address	Control	Data (0 or more bytes)	CRC	Flag

Figure 3.12 Use of a bit-oriented protocol to construct a frame.

SYN	DLE	STX	Header	Data (0 or more bytes)	DLE	ETX	CRC	SYN

Figure 3.13 Use of a character-oriented protocol to construct a frame.

3.2.3.1 *Bit Stuffing and Character Stuffing*

A flag in the bit-oriented protocol determines the start and end of a frame. However, the data and other fields may also have the same sequence. Unless special measures are taken at the transmitter end, the receiver may take the wrong sequence as the end-of-frame flag. To avoid such an error, the sender end must be designed to ensure that no bit sequence identical to the flag will occur anywhere else in the data. If any such sequence is noticed, it may be modified by the process called bit stuffing of 0 bits. For example, if any sequence of five is found at the sender side after the start flag, a 0 is inserted, as shown in Figure 3.14. The reverse of stuffing, called de-stuffing, is done at the receiver end if a 0 is found in the received data after five consecutive 1s to recover the original data.

Just as in bit stuffing in bit-oriented protocols, it is necessary to perform character stuffing at the sender to avoid confusion at the receiver end. Character stuffing consists of replacing every occurrence of the pattern DLE in the frame between the two legitimate DLEs by DLE DLE. This prevents the pattern DLE ETX from appearing anywhere in the frame except at the end. A de-stuffing mechanism at the receiver simply removes a DLE from the frame if two consecutive DLEs are received.

3.2.3.2 *Flow Control*

Flow control defines both the way in which many frames are sent and tracked and how the stations do error control. Error control defines both how a station checks frames for

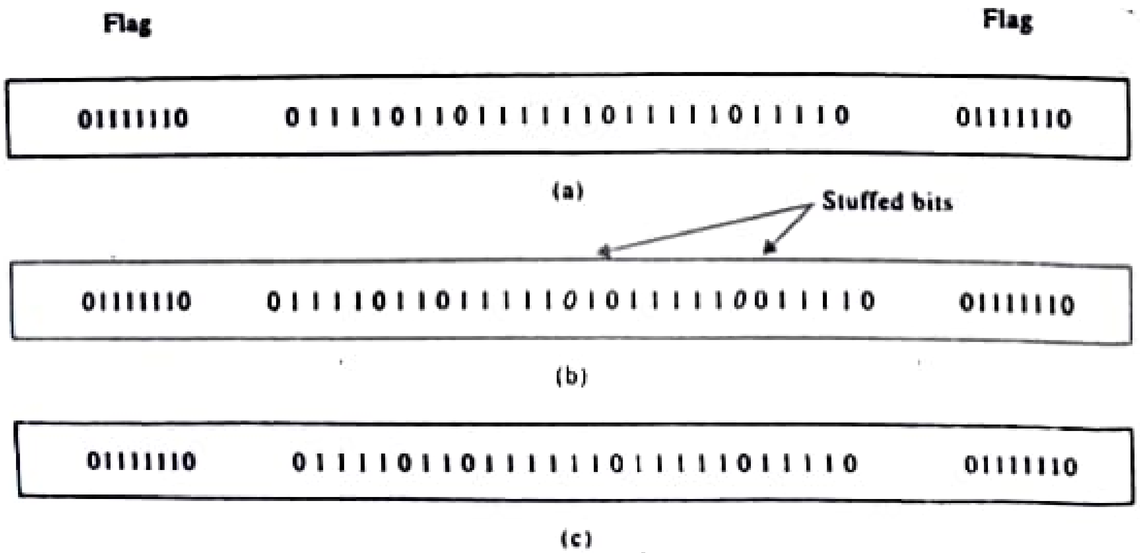

Figure 3.14 (a) Original frame at the sender, with flags. (b) Stuffed frame. (c) De-stuffed frame at the receiver.

errors and what it does if it finds them. A common error control approach is to have the receiver send a message to the transmitter indicating that an error has occurred in the preceding transmission. The message is effectively a request to resend the erroneous frame. Thus this type of error control is often called automatic repeat request (ARQ).

In general, the flow control protocols ensure that all the related frames arrive at their destination accurately and in order. Many physical limitations of the stations must be taken into consideration to ensure a proper flow control. Buffer space, processing capability, and transmission line errors are some limitations that must be considered in obtaining proper flow control. For example, the receiver's buffer capacity is usually limited, and if a sender keeps transmitting the frames, the buffers at the receiver may in time fill up, resulting in buffer overrun, which can loosen and damage the frames. One method to stop the sender from overrunning the receiver's buffer is to insert special ASCII XON/XOFF characters in the data. Thus, a receiver may send an XOFF message to the sender along with its own data to stop the flow of incoming data. When the receiver is ready again, it can send an XON message to resume the transmission. It may be noticed that this method is more applicable to character-oriented asynchronous transmission.

Bit-oriented or frame-oriented transmission requires better organization than the character-oriented kind, since the information is sent and received in larger pieces rather than bytes or characters. Two well-known protocols that provide the flow control in synchronous transmission are the stop-and-wait and sliding window protocols.

3.2.3.3 Stop-and-Wait Protocol

In the stop-and-wait protocol, the receiver buffer space is assumed to be limited, such that the sender may not continue to transmit until an okay signal (an acknowledgment) is received. The acknowledgment is a frame sent by the receiver to the sender to indicate a successful transmission. The sender sends one frame and then waits for an acknowledgment from the receiver before resuming transmission, hence the name stop-and-wait protocol. The sender does not proceed with the next frame until a positive acknowledgment, indicating an error-free transmission, has been obtained from the receiver. If the acknowledgment message indicates an error in the transmission just completed, the sender transmits the same frame again instead of getting a new one from its buffer.

The receiver checks each received frame for errors (using CRC, checksum, etc.), and if no error is found, the recovered data are passed to the upper layer and a positive acknowledgment is sent to the sender. In the case of an error, the receiver sets the error field to 1 in the acknowledgment message, and the frame is retransmitted.

While this protocol seems to work, it has some shortcomings. For instance, it does not allow the sender to perceive transmission errors that have resulted in frame loss. What happens if the frame sent by the sender is lost on the line? Obviously if the receiver never got it, no acknowledgment for that frame will be generated, and sender has no way of knowing of the problem. Similarly, if an acknowledgment is lost on the line or damaged, the error field will be changed to indicate (incorrectly) an unsuccessful transmission. Somehow both sender and receiver must learn about this system break down.

Fortunately, the solution to these problems is a timer mechanism at the sender end and a capability for the receiver to determine the presence of duplicate frames in the case of an acknowledgment loss. The timer mechanism allows the sender to time-out after a certain time and retransmit the frame. If the cause of time-out was frame loss, then the receiver gets the frame and normal operation resumes. However, if the cause was acknowledgment loss or damage, the receiver will end up with the same frame twice, resulting in duplicate frames. Since there can be only one pending frame at one time, a duplicate can be detected easily by checking the sequence number bit that alternates between 0 and 1. (The protocol is also called the alternate bit protocol for this reason.) If two consecutive frames at the receiver have the same sequence, acknowledgment loss or damage must have occurred, and the receiver simply discards one frame.

The mechanism just described seems to take care of the inherent problems in the stop-and-wait protocol. We see that the protocol works properly under different error conditions. Its efficiency may be very low, however, since there is only one frame pending at a time. For example, on a 10 Mbps line with frame and acknowledgment lengths of 1000 bits, if the signal propagation delay from sender to receiver is 100 µs (approximately 20 km), the efficiency of the stop-and-wait protocol will be

$$\frac{1000/10^7}{(1000 + 1000) / 10^7 + 2 \times 100 \times 10^{-6}} = 0.25$$

Thus the line running at 10 Mbps can be used effectively up to 2.5 Mbps only, which is quite low. For higher distances, this efficiency is even smaller. Also, note that the effect of processing delays at both ends is neglected in this evaluation.

3.2.3.4 Sliding Window Protocol

In the sliding window protocol, an extension of the stop-and-wait protocol, a sequence of packets may be sent and received simultaheously. The idea is to "keep the pipe full in both directions." However, there is a limiting factor. The number of pending frames between transmitter and receiver may not exceed the limit allowed by the sequence number. That is, if the sequence number consists of three bits, then the frames 0–7 may be generated by one end. The same sequence number must be used for the next group of frames. To ensure that the frames belonging to the first group can be distinguished from frames belonging to the second, the sender cannot send more than 8 frames at a time.

To keep track of sent and received frames, windows are implemented that open and close as frames are being sent or received. In the sender window shown in Figure 3.15,

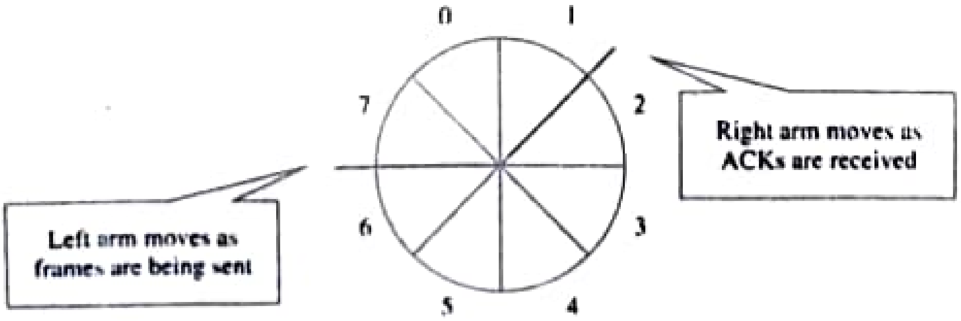
0
1
2
3
4
5
6
7
Right arm moves as ACKs are received
Left arm moves as frames are being sent

Figure 3.15 Sender's sliding window with two moving arms.

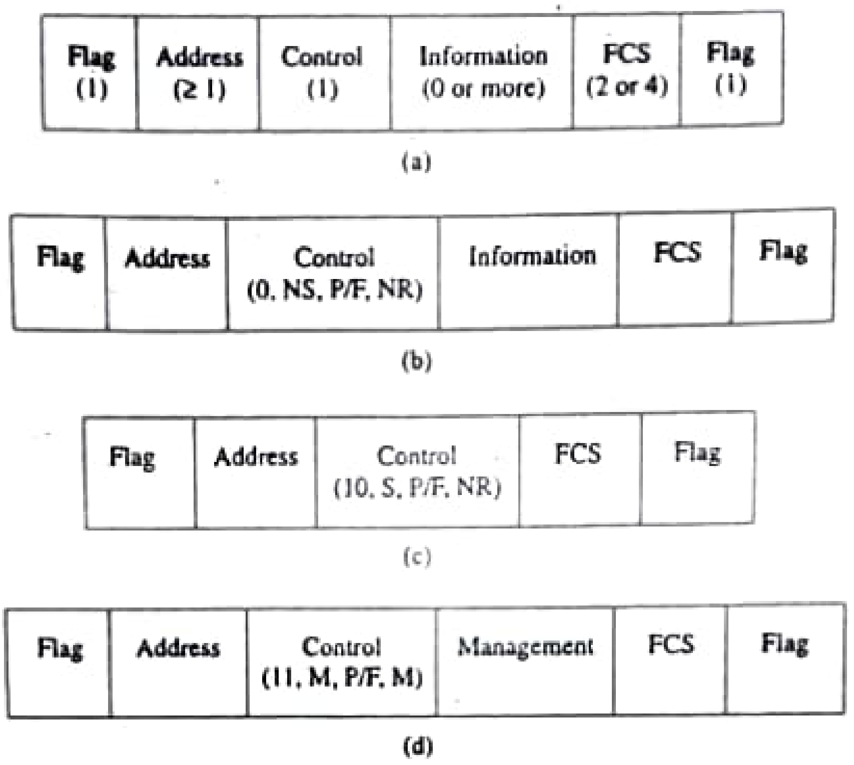

Figure 3.16 HDLC frame formats: (a) standard, (b) I frame, (c) S frame, and (d) U frame.

sion from primary before transmission can occur, as explained earlier. In ARM, a secondary station may initiate a communication by sending control packets to the primary. Unlike the primary station, however, the secondary may not control the line and issue commands. In ABM, only combined stations are connected. Therefore, any station may initiate transmission and may change its role as primary and secondary station.

Three types of frame are defined for HDLC: information frame (I frame), supervisory frame (S frame), and unnumbered frame (U frame) as shown in Figure 3.16. The number of bytes in each field is given in parentheses in Figure 3.16a. We note that the main difference between the I frame and the S frame is that the information (or data) part is not included in the S frame. Therefore an S frame acts like a control frame for exchange of control information between the stations. A U frame is a management frame that is used like an S frame to exchange control information.

In an HDLC frame, the flag is set to 01111110. Bit stuffing is used to avoid confusion with the end-of-frame flag. The FCS is either a 2- or a 4-byte CRC. The control field identifies the three frame types, starting with 0 for an I frame, 10 for an S frame, and 11 for a U frame, respectively. The 3-bit NS and NR fields in the control field indicate the sent frame sequence number and the expected frame sequence number, respectively. Hence, NR is used in a way to acknowledge the frames up to frame number (NR − 1). Notice that NR exists in both I and S frames. The inclusion of an acknowledgment on an information frame is called piggybacking; that is, the acknowledgment is not a separate frame by itself but gets a free ride by being part of an I frame. The 3-bit S control in the S frame is used for supervisory purposes. For instance, it may indicate the type of error control as go-back-*n* ARQ or selective-repeat ARQ, as explained earlier. Similarly the five *M* bits (2,3 combination) allow 32 types of management information to be exchanged in the U frame. More types of management information can be expressed by using the information field. Some examples of management information include set normal response mode (SNRM), request disconnect (RD), and reset (RST).

The stations use the P/F bit in the frame for polling and responses. When the primary station sets the bit to 1, it means that a secondary station with address NR has been polled. In response, the secondary station may transmit frames with the P/F bit set to 0. When the primary station transmits its last frame, it sets the P/F bit to 1 to indicate the final frame.

3.3 NETWORK LAYER

It is important that messages be delivered from source to destination in minimal time. This goal can be achieved relatively simply for a point-to-point network, since the shared link directly connects the two stations. However, the task is more complex for a multipoint network, in which there may be one or more active stations between the two ends. In addition, there may be more than one path connecting the two ends, and forwarding and routing strategies must be devised for each case. The network layer of OSI architecture deals with the connection of two ends via a switching mechanism to allow the use of network links in a predetermined manner. The two services used are called connection-oriented services (CONS) and connectionless network service (CLNS). In connection-oriented service there are three main phases of communication. In the first phase a connection is established between the sender and the receiver, followed by the second phase consisting of data transfer. The connection may be terminated by either side in the third phase when the data transfer is complete or for some other reason.

There are no connection establishment and termination phases in a connectionless service. Rather, the stations transfer the data directly. The packets forming the data may take different routes to reach the destination, resulting in different packet delays and unordered arrivals at the destination. In this case the higher layers must perform error checking to ensure an error-free and reliable end-to-end service. It may be noted that the CONS provides better network service than CLNS because in connection-oriented service a path is established between the two ends and the packets must follow the same path, minimizing the chances of loss and preventing any unordered delivery of packets. However, the delay will depend on the line conditions. Once the connection has been established in CONS, the network is responsible for delivering the data packets in order, without causing any extra delays. However, the connection establishment and termination phases cause the extra delay.

3.3.1 Subnet Concept

An internetwork consists of many smaller networks capable of performing switching, routing, and forwarding on their own. These smaller constituent networks of an internetwork are called subnetworks or subnets. Each subnet is capable of operating on its own. In addition, interconnection of subnets is possible by means of special switching devices called intermediate systems (ISs) by ISO. Two types of IS are routers connecting the subnets at