CHAPTER 1

Introduction

Solutions to Review Questions and Exercises

Review Questions

1. The five components of a data communication system are the ***sender***, **receiver**, ***transmission medium***, ***message***, and ***protocol***.
2. The advantages of distributed processing are ***security***, ***access to distributed data bases***, ***collaborative processing***, and ***faster problem solving***.
3. The three criteria are ***performance***, ***reliability***, and ***security***.
4. Advantages of a multipoint over a point-to-point configuration (type of connec tion) include ***ease of installation*** and ***low cost***.
5. Line configurations (or types of connections) are ***point-to-point*** and ***multipoint***.
6. We can divide line configuration in two broad categories:
   a. ***Point-to-point***: ***mesh***, ***star***, and ***ring***.
   b. ***Multipoint***: ***bus***
7. In ***half-duplex transmission***, only one entity can send at a time; in a ***full-duplex transmission***, both entities can send at the same time.
8. We give an advantage for each of four network topologies:
   a. ***Mesh***: secure
   b. ***Bus***: easy installation
   c. ***Star***: robust
   d. ***Ring***: easy fault isolation
9. The number of cables for each type of network is:
   a. ***Mesh***: $n (n - 1) / 2$
   b. ***Star***: n
   c. ***Ring***: $n - 1$
   d. ***Bus***: one backbone and n drop lines
10. The general factors are ***size***, ***distances*** (covered by the network), ***structure***, and ***ownership***.

11. An ***internet*** is an interconnection of networks. The ***Internet*** is the name of a spe cific worldwide network

12. A ***protocol*** defines ***what*** is communicated, in ***what way*** and ***when***. This provides accurate and timely transfer of information between different devices on a net work.

13. ***Standards*** are needed to create and maintain an open and competitive market for manufacturers, to coordinate protocol rules, and thus guarantee compatibility of data communication technologies.

Exercises

14. ***Unicode*** uses **32** bits to represent a symbol or a character. We can define $\mathbf{2^{32}}$ differ ent symbols or characters.

15. With **16** bits, we can represent up to $\mathbf{2^{16}}$ different colors.

16.

a. Cable links: n $(n - 1) / 2 = (6 \cdot 5) / 2 =$ **15**

b. Number of ports: $(n - 1) =$ **5** ports needed per device

17.

a. ***Mesh topology***: If one connection fails, the other connections will still be work ing.

b. **Star topology**: The other devices will still be able to send data through the hub; there will be no access to the device which has the failed connection to the hub. c. **Bus Topology**: All transmission stops if the failure is in the bus. If the drop-line fails, only the corresponding device cannot operate.

d. ***Ring Topology***: The failed connection may disable the whole network unless it is a dual ring or there is a by-pass mechanism.

18. This is a ***LAN***. The Ethernet hub creates a LAN as we will see in Chapter 13. 19. Theoretically, in a ***ring topology***, unplugging one station, interrupts the ring. How ever, most ring networks use a mechanism that bypasses the station; the ring can continue its operation.

20. In a ***bus topology***, no station is in the path of the signal. Unplugging a station has no effect on the operation of the rest of the network.

21. See Figure 1.1

22. See Figure 1.2.

23.

a. E-mail is not an interactive application. Even if it is delivered immediately, it may stay in the mail-box of the receiver for a while. It is not sensitive to delay. b. We normally do not expect a file to be copied immediately. It is not very sensi tive to delay.

c. Surfing the Internet is the an application very sensitive to delay. We except to get access to the site we are searching.

24. In this case, the communication is only between a caller and the callee. A dedi cated line is established between them. The connection is ***point-to-point.***

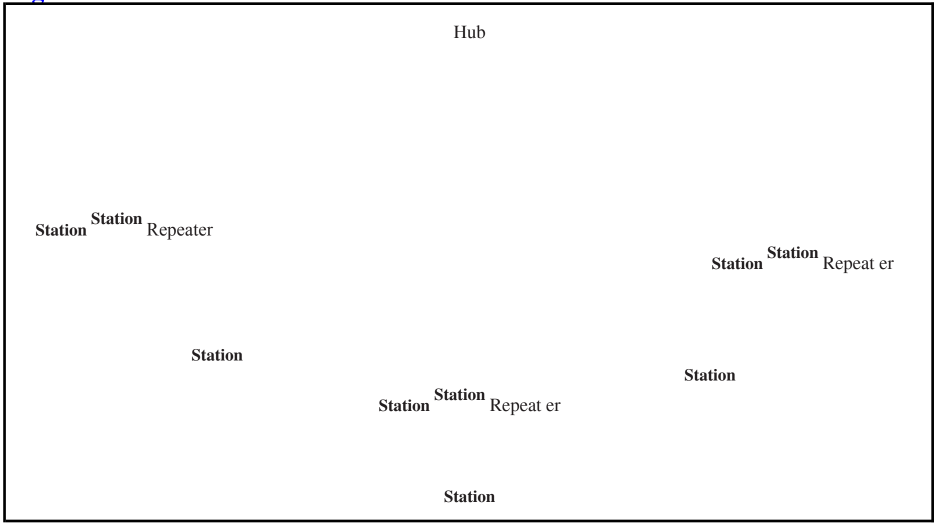


Figure 1.1 *Solution to Exercise 21*

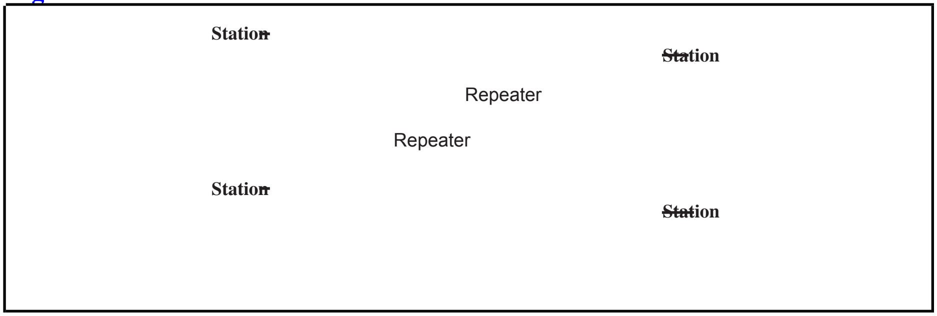


Figure 1.2 *Solution to Exercise 22*

25. The telephone network was originally designed for voice communication; the Internet was originally designed for data communication. The two networks are similar in the fact that both are made of interconnections of small networks. The telephone network, as we will see in future chapters, is mostly a circuit-switched network; the Internet is mostly a packet-switched network.

CHAPTER 2

Network Models

Solutions to Review Questions and Exercises

Review Questions

1. The Internet model, as discussed in this chapter, include ***physical***, ***data link***, ***network***, ***transport***, and ***application*** layers.
2. The network support layers are the ***physical***, ***data link***, and ***network*** layers.
3. The ***application*** layer supports the user.
4. The *transport layer* is responsible for ***process-to-process*** delivery of the entire message, whereas the network layer oversees ***host-to-host*** delivery of individual packets.
5. ***Peer-to-peer processes*** are processes on two or more devices communicating at a same layer
6. Each layer calls upon the ***services*** of the layer just below it using interfaces between each pair of adjacent layers.
7. ***Headers*** and ***trailers*** are control data added at the beginning and the end of each data unit at each layer of the sender and removed at the corresponding layers of the receiver. They provide source and destination addresses, synchronization points, information for error detection, etc.
8. The *physical layer* is responsible for transmitting a bit stream over a physical medium. It is concerned with

   a. ***physical characteristics of the media***

   b. ***representation of bits***

   c. ***type of encoding***

   d. ***synchronization of bits***

   e. ***transmission rate and mode***

   f. ***the way devices are connected with each other and to the links***
9. The *data link layer* is responsible for

   a. ***framing data bits***

   b. ***providing the physical addresses of the sender/receiver***

   c. ***data rate control***

   d. ***detection and correction of damaged and lost frames***
10. The *network layer* is concerned with delivery of a packet across multiple networks; therefore its responsibilities include

    a. ***providing host-to-host addressing***

    b. ***routing***
11. The *transport layer* oversees the process-to-process delivery of the entire message.

It is responsible for

a. ***dividing the message into manageable segments***

b. ***reassembling it at the destination***

c. ***flow and error control***

12. The *physical address* is the local address of a node; it is used by the data link layer to deliver data from one node to another within the same network. The ***logical address*** defines the sender and receiver at the network layer and is used to deliver messages across multiple networks. The port address (service-point) identifies the application process on the station.

13. The *application layer services* include ***file transfer***, ***remote access***, ***shared data base management***, and ***mail services***.

14. The ***application***, ***presentation***, and ***session*** layers of the OSI model are represented by the *application* layer in the Internet model. The lowest four layers of OSI corre spond to the Internet model layers.

Exercises

15. The ***International Standards Organization***, or the ***International Organization of Standards***, (**ISO**) is a multinational body dedicated to worldwide agreement on international standards. An ISO standard that covers all aspects of network com munications is the ***Open Systems Interconnection*** (**OSI**) model.

16.

a. Route determination: ***network*** layer

b. Flow control: ***data link*** and ***transport*** layers

c. Interface to transmission media: ***physical*** layer

d. Access for the end user: ***application*** layer

17.

a. Reliable process-to-process delivery: **transport** layer

b. Route selection: ***network*** layer

c. Defining frames: ***data link*** layer

d. Providing user services: ***application*** layer

e. Transmission of bits across the medium: ***physical*** layer

18.

a. Communication with user's application program: ***application*** layer

b. Error correction and retransmission: ***data link*** and ***transport*** layers

c. Mechanical, electrical, and functional interface: ***physical layer***

d. Responsibility for carrying frames between adjacent nodes: ***data link*** layer

19.

a. Format and code conversion services: ***presentation*** layer

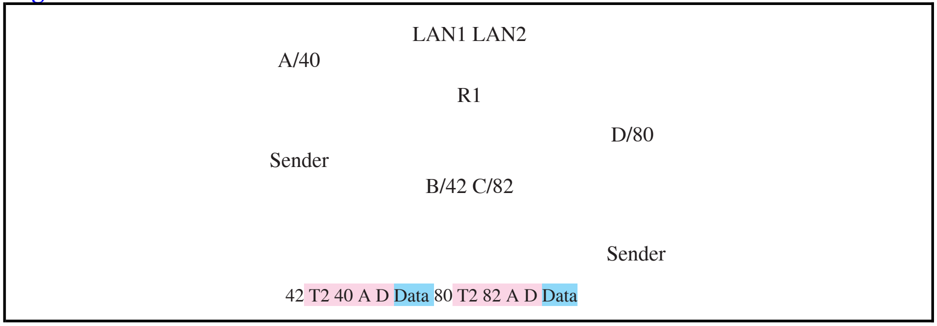
b. Establishing, managing, and terminating sessions: ***session*** layer c.
Ensuring reliable transmission of data: ***data link*** and ***transport*** layers d.

Log-in and log-out procedures: ***session*** layer

e. Providing independence from different data representation: ***presentation*** layer

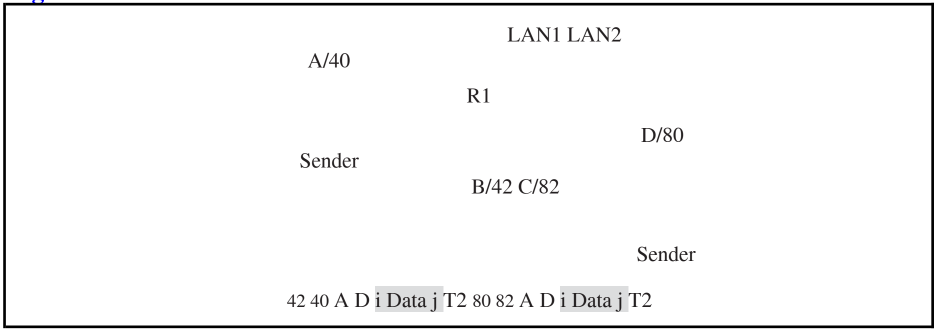
20. See Figure 2.1.

Figure 2.1 *Solution to Exercise 20*



21. See Figure 2.2.

Figure 2.2 *Solution to Exercise 21*



22. If the corrupted destination address does not match any station address in the network, the packet is lost. If the corrupted destination address matches one of the stations, the frame is delivered to the wrong station. In this case, however, the error detection mechanism, available in most data link protocols, will find the error and discard the frame. In both cases, the source will somehow be informed using one of the data link control mechanisms discussed in Chapter 11.

23. Before using the destination address in an intermediate or the destination node, the packet goes through error checking that may help the node find the corruption (with a high probability) and discard the packet. Normally the upper layer protocol will inform the source to resend the packet.

24. Most protocols issue a ***special error message*** that is sent back to the source in this case.

25. The errors ***between*** the nodes can be detected by the data link layer control, but the error ***at*** the node (between input port and output port) of the node cannot be detected by the data link layer.

CHAPTER 3

Data and Signals

Solutions to Review Questions and Exercises

Review Questions

1. ***Frequency*** and ***period*** are the inverse of each other. T = 1/ f and f = 1/T.
2. The ***amplitude*** of a signal measures the value of the signal at any point. The ***fre quency*** of a signal refers to the number of periods in one second. The phase describes the position of the waveform relative to time zero.
3. Using Fourier analysis. ***Fourier series*** gives the frequency domain of a periodic signal; ***Fourier analysis*** gives the frequency domain of a nonperiodic signal.
4. Three types of transmission impairment are ***attenuation***, ***distortion***, and ***noise***. 5. ***Baseband transmission*** means sending a digital or an analog signal without modu lation using a low-pass channel. ***Broadband transmission*** means modulating a digital or an analog signal using a band-pass channel.
6. A ***low-pass channel*** has a bandwidth starting from zero; a ***band-pass*** channel has a bandwidth that does not start from zero.
7. The ***Nyquist theorem*** defines the maximum bit rate of a noiseless channel.
8. The ***Shannon capacity*** determines the theoretical maximum bit rate of a noisy channel.
9. ***Optical signals*** have very high frequencies. A high frequency means a short wave length because the wave length is inversely proportional to the frequency (λ = v/f), where v is the propagation speed in the media.
10. A signal is ***periodic*** if its frequency domain plot is ***discrete***; a signal is **nonperi odic** if its frequency domain plot is ***continuous***.
11. The frequency domain of a voice signal is normally ***continuous*** because voice is a ***nonperiodic*** signal.
12. An alarm system is normally ***periodic***. Its frequency domain plot is therefore ***dis crete***.
13. This is ***baseband transmission*** because no modulation is involved.
14. This is ***baseband transmission*** because no modulation is involved.
15. This is ***broadband transmission*** because it involves modulation.

Exercises

16.

a. **T = 1 / f** = 1 / (24 Hz) = 0.0417 s = 41.7 · 10^{-3} s = **41.7 ms**

b. **T = 1 / f** = 1 / (8 MHz) = 0.000000125 = 0.125 · 10^{-6} s = **0.125 μs**

c. **T = 1 / f** = 1 / (140 KHz) = 0.00000714 s = 7.14 · 10^{-6} s = **7.14 μs**

17.

a. f = 1 / T = 1 / (5 s) = 0.2 Hz

b. **f = 1 / T** = 1 / (12 μs) =83333 Hz = 83.333 · 10^{3} Hz = **83.333 KHz**

c. **f = 1 / T** = 1 / (220 ns) = 4550000 Hz = 4.55· 10^{6} Hz = **4.55 MHz**

18.

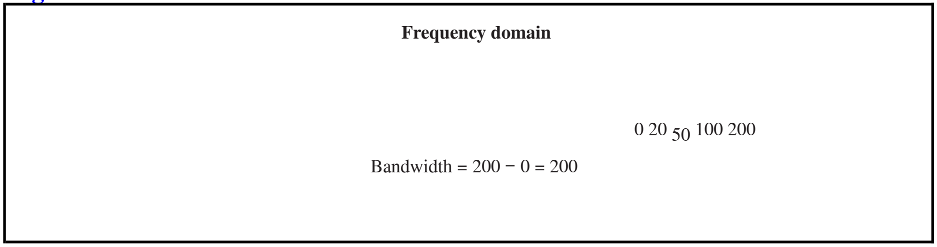
a. 90 degrees (π/2 radian)

b. 0 degrees (0 radian)

c. 90 degrees (π/2 radian)

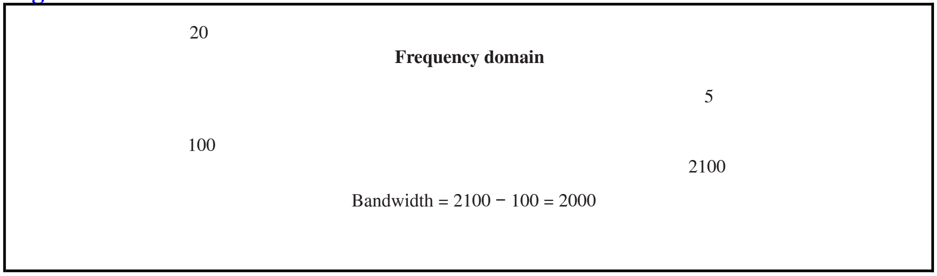
19. See Figure 3.1

Figure 3.1 *Solution to Exercise 19*



20. We know the lowest frequency, 100. We know the bandwidth is 2000. The highest frequency must be 100 + 2000 = **2100 Hz**. See Figure 3.2

Figure 3.2 *Solution to Exercise 20*



21. Each signal is a simple signal in this case. The bandwidth of a simple signal is zero. So the bandwidth of both signals are the same.

22.

a. bit rate = 1/ (bit duration) = 1 / (0.001 s) = 1000 bps = **1 Kbps**

b. bit rate = 1/ (bit duration) = 1 / (2 ms) = **500 bps**

c. bit rate = 1/(bit duration) = 1 / (20 μs/10) = 1 / (2 μs) = **500 Kbps**

23.

a. (10 / 1000) s = **0.01 s**

b. (8 / 1000) s = 0. 008 s = **8 ms**

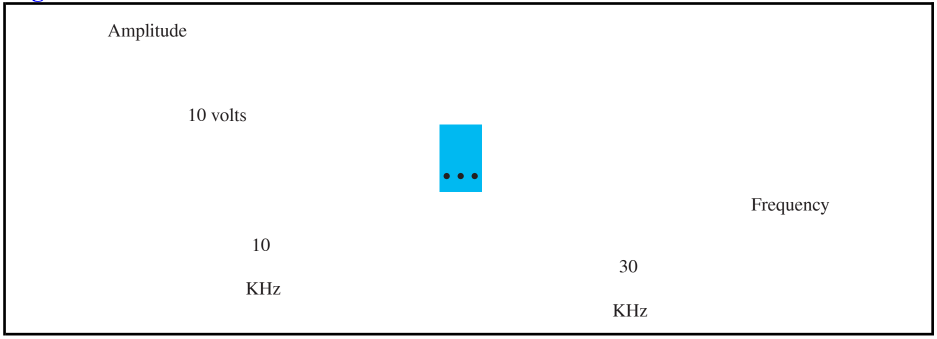
c. ((100,000 · 8) / 1000) s = **800 s**

24. There are 8 bits in 16 ns. Bit rate is 8 / (16 · 10^{-9}) = 0.5 · 10^{-9} = **500 Mbps**

25. The signal makes 8 cycles in 4 ms. The frequency is 8 /(4 ms) = **2 KHz** 26.
The bandwidth is 5 · 5 = **25 Hz**.

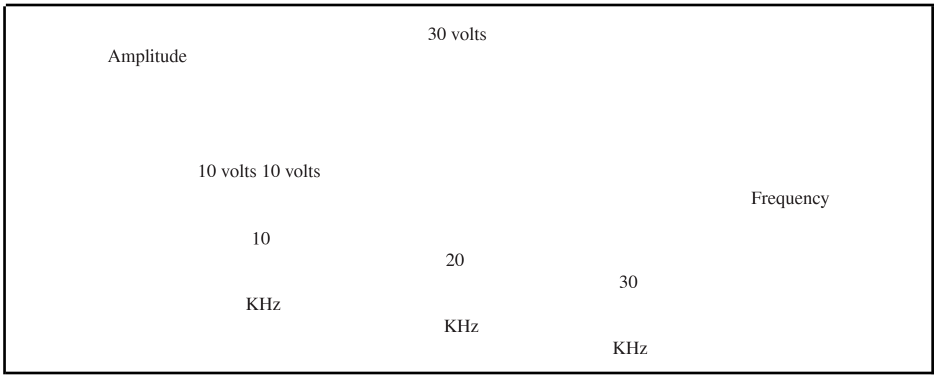
27. The signal is periodic, so the frequency domain is made of discrete frequencies. as shown in Figure 3.3.

Figure 3.3 *Solution to Exercise 27*



28. The signal is nonperiodic, so the frequency domain is made of a continuous spectrum of frequencies as shown in Figure 3.4.

Figure 3.4 *Solution to Exercise 28*



29\.

Using the first harmonic, data rate = 2 · 6 MHz = **12 Mbps**
Using three harmonics, data rate = (2 · 6 MHz) /3 = **4 Mbps**
Using five harmonics, data rate = (2 · 6 MHz) /5 = **2.4 Mbps**

30. dB = 10 $\log_{10}(90 / 100)$ = **–0.46 dB**

31. –10 = 10 $\log_{10}(P_2 / 5) \rightarrow \log_{10}(P_2 / 5) = -1 \rightarrow (P_2 / 5) = 10^{-1} \rightarrow P_2 =$ **0.5 W** 32.
The total gain is 3 · 4 = 12 dB. The signal is amplified by a factor $10^{1.2}$ = **15.85.**

4

33. 100,000 bits / 5 Kbps = **20 s**

34. 480 s · 300,000 km/s = **144,000,000 km**

35. 1 μm · 1000 = 1000 μm = **1 mm**

36. We have

$$\mathbf{4{,}000 \log_2 (1 + 1{,}000) \approx 40\ Kbps}$$

37. We have

$$\mathbf{4{,}000 \log_2 (1 + 10 / 0.005) = 43{,}866\ bps}$$

38. The file contains 2,000,000 · 8 = 16,000,000 bits. With a 56-Kbps channel, it takes 16,000,000/56,000 = **289 s**. With a 1-Mbps channel, it takes **16 s**.

39. To represent 1024 colors, we need $\log_2 1024$ = 10 (see Appendix C) bits. The total number of bits are, therefore,

$$\mathbf{1200 \cdot 1000 \cdot 10 = 12{,}000{,}000\ bits}$$

40. We have

$$\mathbf{SNR = (200\ mW) / (10 \cdot 2 \cdot \mu W) = 10{,}000}$$

We then have

$$\mathbf{SNR_{dB} = 10 \log_{10} SNR = 40}$$

41. We have

SNR= (signal power)/(noise power).

However, power is proportional to the square of voltage. This means we have

$$\textbf{SNR} = [(\text{signal voltage})^2] / [(\text{noise voltage})^2] = [(\text{signal voltage}) / (\text{noise voltage})]^2 = 20^2 = \mathbf{400}$$

We then have

$$\text{SNR}_{\text{dB}} = 10 \log_{10} \text{SNR} \approx \mathbf{26.02}$$

42. We can approximately calculate the capacity as

a. $C = B \cdot (\text{SNR}_{\text{dB}} /3) = 20$ KHz $\cdot$ $(40 /3) =$ **267 Kbps**

b. $C = B \cdot (\text{SNR}_{\text{dB}} /3) = 200$ KHz $\cdot$ $(4 /3) =$ **267 Kbps**

c. $C = B \cdot (\text{SNR}_{\text{dB}} /3) = 1$ MHz $\cdot$ $(20 /3) =$ **6.67 Mbps**

43.

a. The data rate is doubled ($C_2 = 2 \cdot C_1$).

b. When the SNR is doubled, the data rate increases slightly. We can say that, approximately, ($C_2 = C_1 + 1$).

44. We can use the approximate formula

$$C = B \cdot (\text{SNR}_{\text{dB}} /3) \text{ or } \text{SNR}_{\text{dB}} = (3 \cdot C) /B$$

We can say that the minimum

$$\text{SNR}_{\text{dB}} = 3 \cdot 100 \text{ Kbps} / 4 \text{ KHz} = 75$$

5

This means that the minimum

$$\text{SNR} = 10^{\text{SNRdB}/10} = 10^{7.5} \approx \mathbf{31{,}622{,}776}$$

45. We have

$$\textbf{transmission time} = (\text{packet length})/(\text{bandwidth}) = (8{,}000{,}000 \text{ bits}) / (200{,}000 \text{ bps}) = \mathbf{40\ s}$$

46. We have

$$(\text{bit length}) = (\text{propagation speed}) \cdot (\text{bit duration})$$

The bit duration is the inverse of the bandwidth.

a. Bit length = $(2 \cdot 10^8$ m$) \cdot [(1 / (1$ Mbps$)] =$ **200 m**. This means a bit occupies 200 meters on a transmission medium.

b. Bit length = $(2 \cdot 10^8$ m$) \cdot [(1 / (10$ Mbps$)] =$ **20 m**. This means a bit occupies 20 meters on a transmission medium.

c. Bit length = $(2 \cdot 10^8$ m$) \cdot [(1 / (100$ Mbps$)] =$ **2 m**. This means a bit occupies 2 meters on a transmission medium.

47.

a. Number of bits = bandwidth · delay = 1 Mbps · 2 ms = **2000 bits** b. Number of bits = bandwidth · delay = 10 Mbps · 2 ms = **20,000 bits** c. Number of bits = bandwidth · delay = 100 Mbps · 2 ms = **200,000 bits** 48. We have

$$\textbf{Latency} = \text{processing time} + \text{queuing time} + \text{transmission time} + \text{propagation time}$$

Processing time = 10 · 1 μs = 10 μs = **0.000010 s**

Queuing time = 10 · 2 μs = 20 μs = **0.000020 s**

Transmission time = 5,000,000 / (5 Mbps) = **1 s**

Propagation time = (2000 Km) / (2 · 10^8) = **0.01 s**

Latency = 0.000010 + 0.000020 + 1 + 0.01 = 1.01000030 s

The transmission time is dominant here because the packet size is huge.

6

CHAPTER 4

Digital Transmission

Solutions to Review Questions and Exercises

Review Questions

1. The three different techniques described in this chapter are ***line coding***, ***block coding***, and ***scrambling***.
2. A ***data element*** is the smallest entity that can represent a piece of information (a bit). A ***signal element*** is the shortest unit of a digital signal. Data elements are what we need to send; signal elements are what we can send. Data elements are being carried; signal elements are the carriers.
3. The ***data rate*** defines the number of data elements (bits) sent in 1s. The unit is bits per second (bps). The ***signal rate*** is the number of signal elements sent in 1s. The unit is the baud.
4. In decoding a digital signal, the incoming signal power is evaluated against the ***baseline*** (a running average of the received signal power). A long string of 0s or 1s can cause ***baseline wandering*** (a drift in the baseline) and make it difficult for the receiver to decode correctly.
5. When the voltage level in a digital signal is constant for a while, the spectrum creates very low frequencies, called ***DC components***, that present problems for a system that cannot pass low frequencies.
6. A ***self-synchronizing*** digital signal includes timing information in the data being transmitted. This can be achieved if there are transitions in the signal that alert the receiver to the beginning, middle, or end of the pulse.
7. In this chapter, we introduced ***unipolar***, ***polar***, ***bipolar***, ***multilevel***, and ***multitransition*** coding.
8. ***Block coding*** provides redundancy to ensure synchronization and to provide inherent error detecting. In general, block coding changes a block of *m* bits into a block of *n* bits, where *n* is larger than *m*.
9. ***Scrambling,*** as discussed in this chapter, is a technique that substitutes long zero

level pulses with a combination of other levels without increasing the number of bits.

10. Both ***PCM*** and ***DM*** use sampling to convert an analog signal to a digital signal. PCM finds the value of the signal amplitude for each sample; DM finds the change between two consecutive samples.

11. In ***parallel transmission*** we send data *several* bits at a time. In ***serial transmission*** we send data *one* bit at a time.

12. We mentioned ***synchronous***, ***asynchronous***, and ***isochronous***. In both synchro nous and asynchronous transmissions, a bit stream is divided into independent frames. In synchronous transmission, the bytes inside each frame are synchro nized; in asynchronous transmission, the bytes inside each frame are also indepen dent. In isochronous transmission, there is no independency at all. All bits in the whole stream must be synchronized.

Exercises

13. We use the formula **s = c · N · (1/r)** for each case. We let c = 1/2.

    a. r = 1 → s = (1/2) · (1 Mbps) · 1/**1** = **500 kbaud**

    b. r = 1/2 → s = (1/2) · (1 Mbps) · 1/(**1/2**) = **1 Mbaud**

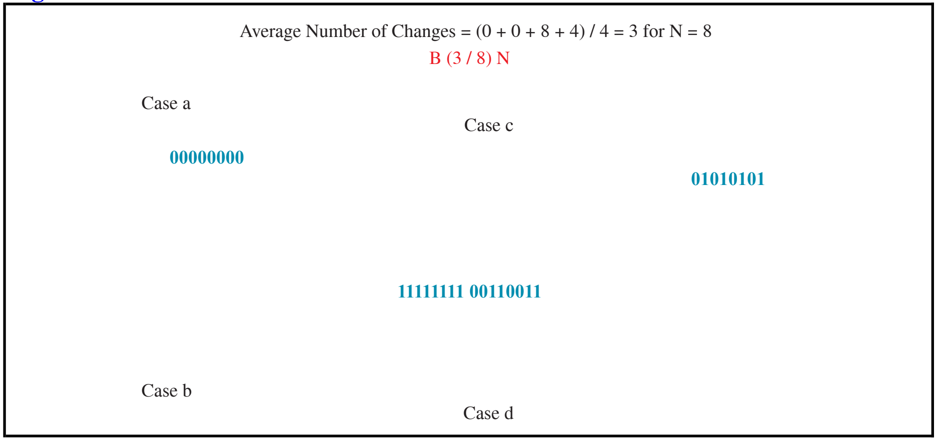
    c. r = 2 → s = (1/2) · (1 Mbps) · 1/**2** = **250 Kbaud**

    d. r = 4/3 → s = (1/2) · (1 Mbps) · 1/(**4/3**) = **375 Kbaud**

14. The number of bits is calculated as (0.2 /100) · (1 Mbps) = **2000 bits**

15. See Figure 4.1. Bandwidth is proportional to **(3/8)N** which is within the range in Table 4.1 (B = 0 to N) for the NRZ-L scheme.

Figure 4.1 *Solution to Exercise 15*



16. See Figure 4.2. Bandwidth is proportional to **(4.25/8)N** which is within the range in Table 4.1 (B = 0 to N) for the NRZ-I scheme.

17. See Figure 4.3. Bandwidth is proportional to **(12.5 / 8) N** which is within the range in Table 4.1 (B = N to B = 2N) for the Manchester scheme.

18. See Figure 4.4. B is proportional to **(12/8) N** which is within the range in Table 4.1 (B = N to 2N) for the differential Manchester scheme.

Figure 4.2 *Solution to Exercise 16*

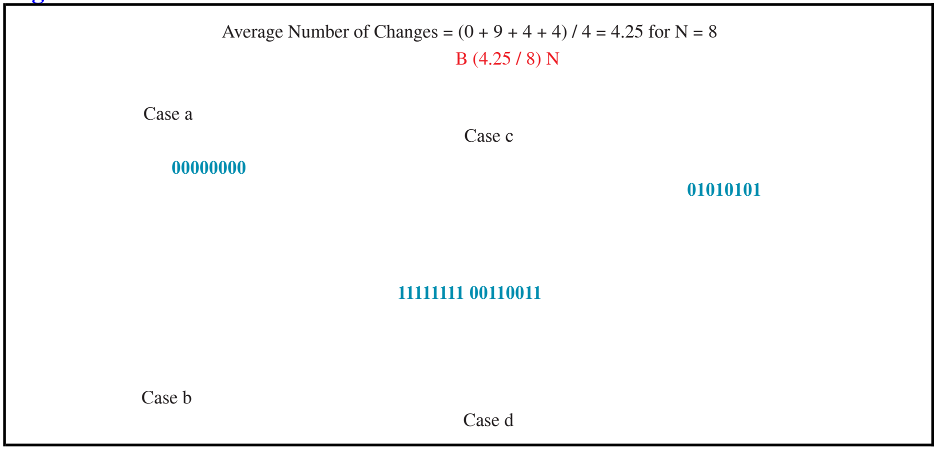


Figure 4.3 *Solution to Exercise 17*

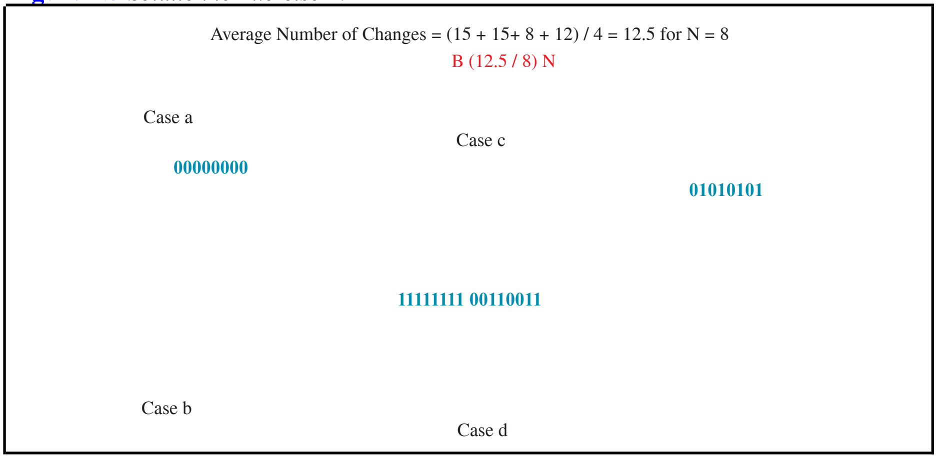
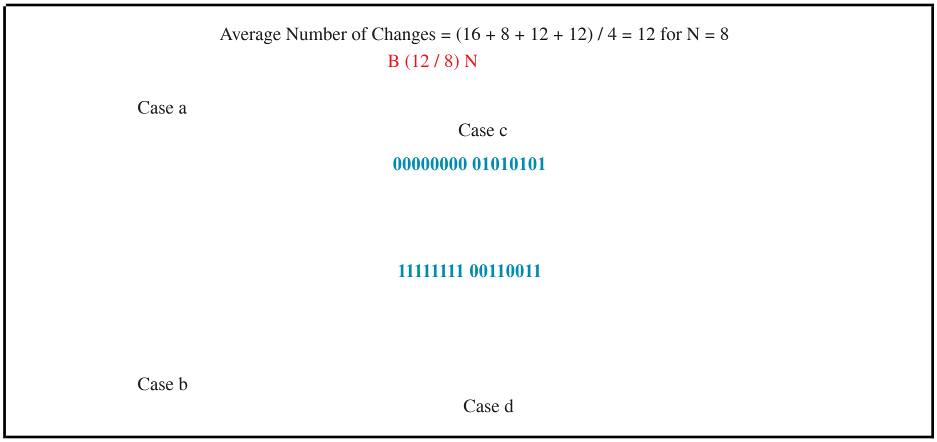


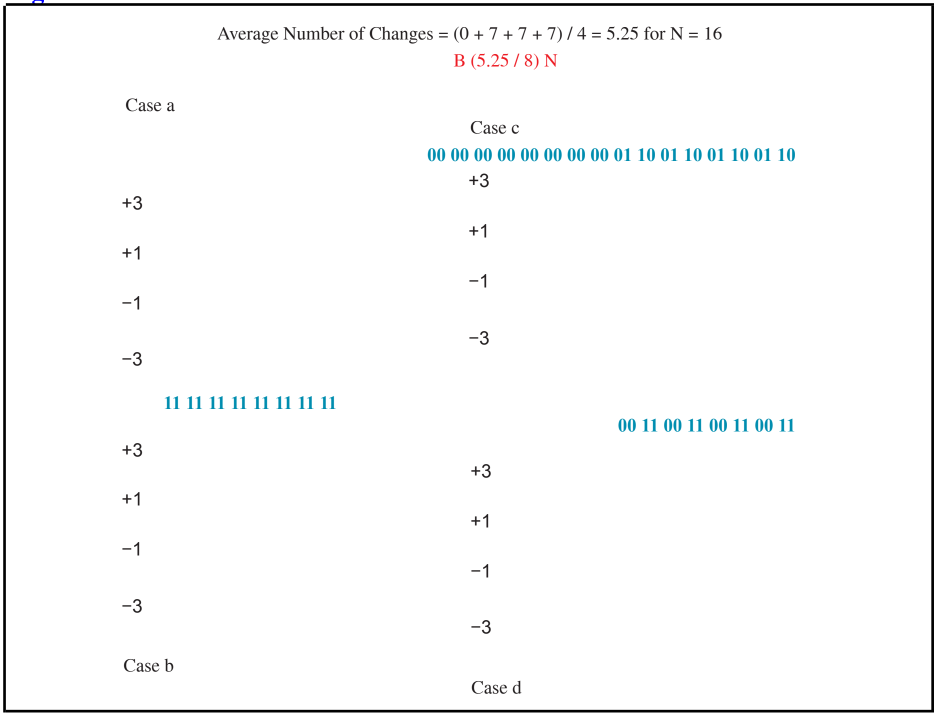
Figure 4.4 *Solution to Exercise 18*



4

19. See Figure 4.5. B is proportional to **(5.25 / 16) N** which is inside range in Table 4.1 (B = 0 to N/2) for 2B/1Q.

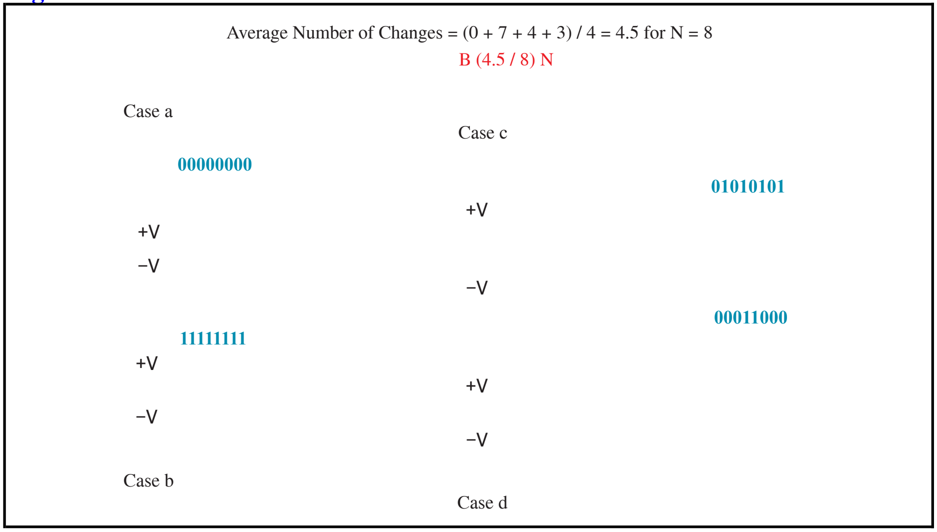
Figure 4.5 *Solution to Exercise 19*



20. See Figure 4.6. B is proportional to **(5.25/8)** · **N** which is inside the range in Table

4.1 (B = 0 to N/2) for MLT-3.

Figure 4.6 *Solution to Exercise 20*



21. The data stream can be found as

    a. NRZ-I: **10011001**.

    b. Differential Manchester: **11000100**.

    c. AMI: **01110001**.

22. The data rate is 100 Kbps. For each case, we first need to calculate the value f / N. We then use Figure 4.6 in the text to find P (energy per Hz). All calculations are approximations.

    a. f /N = 0/100 = 0 → **P = 1.0**

    b. f /N = 50/100 = 1/2 → **P = 0.5**

    c. f /N = 100/100 = 1 → **P = 0.0**

    d. f /N = 150/100 = 1.5 → **P = 0.2**

23. The data rate is 100 Kbps. For each case, we first need to calculate the value f/N. We then use Figure 4.8 in the text to find P (energy per Hz). All calculations are approximations.

    a. f /N = 0/100 = 0 → **P = 0.0**

    b. f /N = 50/100 = 1/2 → **P = 0.3**

    c. f /N = 100/100 = 1 → **P = 0.4**

    d. f /N = 150/100 = 1.5 → **P = 0.0**

24.

    a. The output stream is **01010 11110 11110 11110 11110 01001**.

    b. The maximum length of consecutive 0s in the input stream is **21**. c. The maximum length of consecutive 0s in the output stream is **2**. 25. In 5B/6B, we have 2^5 = 32 data sequences and 2^6 = 64 code sequences. The number of unused code sequences is 64 − 32 = **32**. In 3B/4B, we have 2^3 = 8 data sequences and 2^4 = 16 code sequences. The number of unused code sequences is 16 − 8 = **8**.

26. See Figure 4.7. Since we specified that the last non-zero signal is positive, the first bit in our sequence is positive.

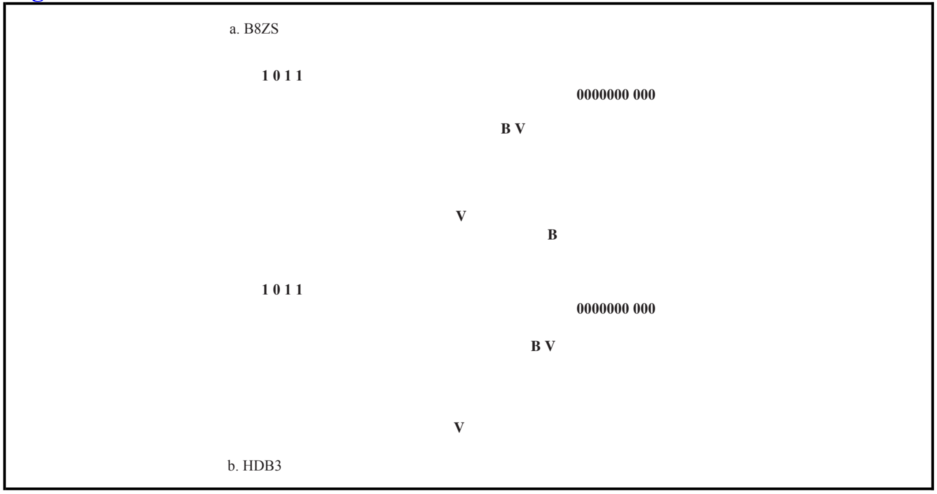


Figure 4.7 *Solution to Exercise 26*

27.

a. In a low-pass signal, the minimum frequency 0. Therefore, we have

$$f_{max} = 0 + 200 = 200 \text{ KHz.} \rightarrow f_s = 2 \cdot 200{,}000 = \mathbf{400{,}000 \text{ samples/s}}$$

b. In a bandpass signal, the maximum frequency is equal to the minimum frequency plus the bandwidth. Therefore, we have

$$f_{max} = 100 + 200 = 300 \text{ KHz.} \rightarrow f_s = 2 \cdot 300{,}000 = \mathbf{600{,}000 \text{ samples /s}}$$

28.

a. In a lowpass signal, the minimum frequency is 0. Therefore, we can say

$$f_{max} = 0 + 200 = 200 \text{ KHz} \rightarrow f_s = 2 \cdot 200{,}000 = \mathbf{400{,}000 \text{ samples/s}}$$

The number of bits per sample and the bit rate are

$$n_b = \log_2 1024 = 10 \text{ bits/sample } N = 400 \text{ KHz} \cdot 10 = \mathbf{4 \text{ Mbps}}$$

b. The value of n_b = 10. We can easily calculate the value of SNR_{dB}

$$SNR_{dB} = 6.02 \cdot n_b + 1.76 = \mathbf{61.96}$$

c. The value of n_b = 10. The minimum bandwidth can be calculated as

$$B_{PCM} = n_b \cdot B_{analog} = 10 \cdot 200 \text{ KHz} = \mathbf{2 \text{ MHz}}$$

29. The maximum data rate can be calculated as

$$N_{max} = 2 \cdot B \cdot n_b = 2 \cdot 200\text{ KHz} \cdot \log_2 4 = \mathbf{800\ kbps}$$

30. We can first calculate the sampling rate (fs) and the number of bits per sample (nb)

$$f_{max} = 0 + 4 = 4\text{ KHz} \rightarrow f_s = 2 \cdot 4 = \mathbf{8000\ sample/s}$$

We then calculate the number of bits per sample.

$$\rightarrow n_b = 30000 / 8000 = 3.75$$

We need to use the next integer $n_b = 4$. The value of SNR_{dB} is

$$SNR_{dB} = 6.02 \cdot n_b + 1.72 = \mathbf{25.8}$$

31. We can calculate the data rate for each scheme:

a. NRZ → $N = 2 \cdot B = 2 \cdot 1\text{ MHz} =$ **2 Mbps**

b. Manchester → $N = 1 \cdot B = 1 \cdot 1\text{ MHz} =$ **1 Mbps**

c. MLT-3 → $N = 3 \cdot B = 3 \cdot 1\text{ MHz} =$ **3 Mbps**

d. 2B1Q → $N = 4 \cdot B = 4 \cdot 1\text{ MHz} =$ **4 Mbps**

32.

a. For synchronous transmission, we have $1000 \cdot 8 =$ **8000** bits.

b. For asynchronous transmission, we have $1000 \cdot 10 =$ **10000** bits. Note that we assume only one stop bit and one start bit. Some systems send more start bits.

c. For case a, the redundancy is 0%. For case b, we send 2000 extra for 8000 required bits. The redundancy is **25%**.

CHAPTER 5

Analog Transmission

Solutions to Review Questions and Exercises

Review Questions

1. Normally, ***analog transmission*** refers to the transmission of analog signals using a band-pass channel. Baseband digital or analog signals are converted to a complex analog signal with a range of frequencies suitable for the channel.

2. A ***carrier*** is a single-frequency signal that has one of its characteristics (amplitude, frequency, or phase) changed to represent the baseband signal.

3. The process of changing one of the characteristics of an analog signal based on the information in digital data is called ***digital-to-analog conversion***. It is also called modulation of a digital signal. The baseband digital signal representing the digital data modulates the carrier to create a broadband analog signal.

4.

a. ASK changes the ***amplitude*** of the carrier.

b. FSK changes the ***frequency*** of the carrier.

c. PSK changes the ***phase*** of the carrier.

d. QAM changes both the ***amplitude*** and the ***phase*** of the carrier.

5. We can say that the most susceptible technique is ***ASK*** because the amplitude is more affected by noise than the phase or frequency.

6. A ***constellation diagram*** can help us define the amplitude and phase of a signal element, particularly when we are using two carriers. The diagram is useful when we are dealing with multilevel ASK, PSK, or QAM. In a constellation diagram, a signal element type is represented as a dot. The bit or combination of bits it can carry is often written next to it.The diagram has two axes. The horizontal *X* axis is related to the in-phase carrier; the vertical *Y* axis is related to the quadrature carrier.

7. The two components of a signal are called ***I*** and ***Q***. The I component, called in phase, is shown on the horizontal axis; the Q component, called quadrature, is shown on the vertical axis.

8. The process of changing one of the characteristics of an analog signal to represent the instantaneous amplitude of a baseband signal is called ***analog-to-analog conversion***. It is also called the *modulation* of an analog signal; the baseband analog signal modulates the carrier to create a broadband analog signal.

9.

a. AM changes the ***amplitude*** of the carrier

b. FM changes the ***frequency*** of the carrier

c. PM changes the ***phase*** of the carrier

10. We can say that the most susceptible technique is ***AM*** because the amplitude is more affected by noise than the phase or frequency.

Exercises

11. We use the formula **S = (1/r) · N**, but first we need to calculate the value of r for each case.

a. $r = \log_2 2 = 1$ → S = (1/1) · (2000 bps) = **2000 baud**

b. $r = \log_2 2 = 1$ → S = (1/1) · (4000 bps) = **4000 baud**

c. $r = \log_2 4 = 2$ → S = (1/2) · (6000 bps) = **3000 baud**

d. $r = \log_2 64 = 6$ → S = (1/6) · (36,000 bps) = **6000 baud**

12. We use the formula **N = r · S**, but first we need to calculate the value of r for each case.

a. $r = \log_2 2 = 1 \rightarrow N = (1) \cdot (1000 \text{ bps}) =$ **1000 bps**
b. $r = \log_2 2 = 1 \rightarrow N = (1) \cdot (1000 \text{ bps}) =$ **1000 bps**
c. $r = \log_2 2 = 1 \rightarrow N = (1) \cdot (1000 \text{ bps}) =$ **1000 bps**
d. $r = \log_2 16 = 4 \rightarrow N = (4) \cdot (1000 \text{ bps}) =$ **4000 bps**

13. We use the formula $\mathbf{r = \log_2 L}$ to calculate the value of r for each case.

a. $\log_2 4 = 2$
b. $\log_2 8 = 3$
c. $\log_2 4 = 2$
d. $\log_2 128 = 7$

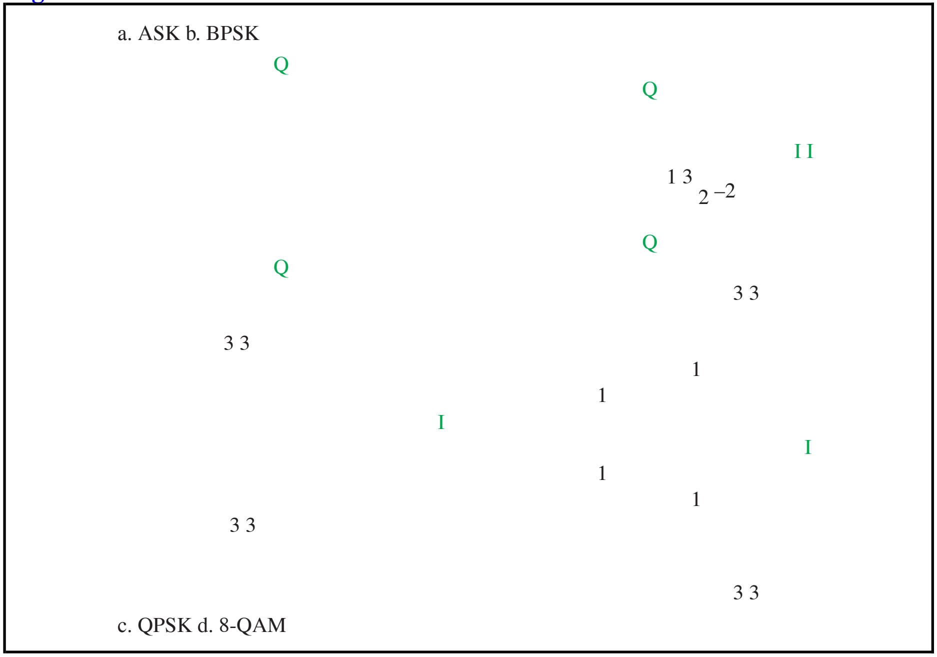
14. See Figure 5.1.

a. We have two signal elements with peak amplitudes 1 and 3. The phase of both signal elements are the same, which we assume to be 0 degrees.

b. We have two signal elements with the same peak amplitude of 2. However, there must be 180 degrees difference between the two phases. We assume one phase to be 0 and the other 180 degrees.

c. We have four signal elements with the same peak amplitude of 3. However, there must be 90 degrees difference between each phase. We assume the first phase to be at 45, the second at 135, the third at 225, and the fourth at 315 degrees. Note that this is one out of many configurations. The phases can be at

Figure 5.1 *Solution to Exercise 14*

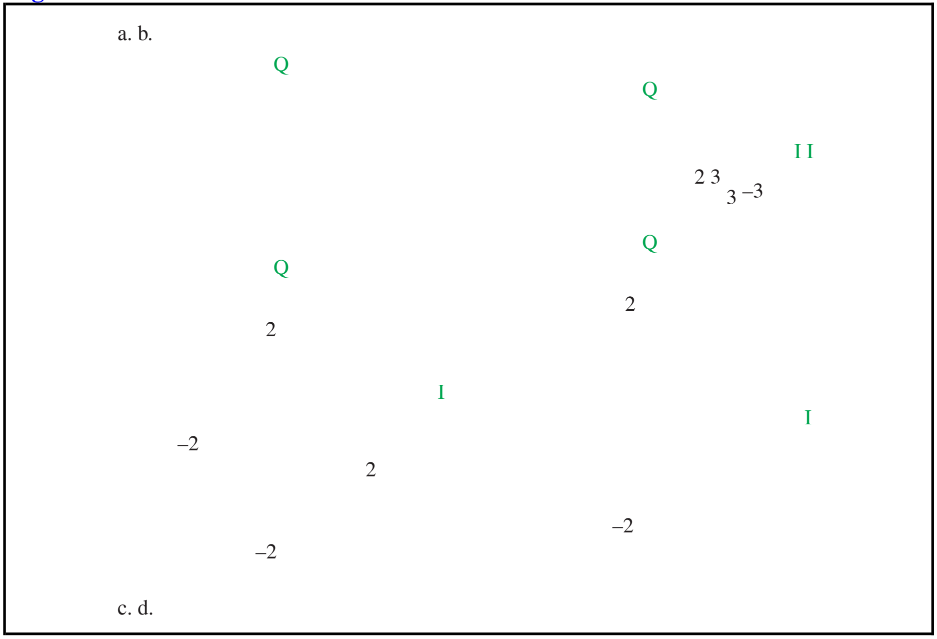


0, 90, 180, and 270. As long as the differences are 90 degrees, the solution is satisfactory.

d. We have four phases, which we select to be the same as the previous case. For each phase, however, we have two amplitudes, 1 and 3 as shown in the figure. Note that this is one out of many configurations. The phases can be at 0, 90, 180, and 270. As long as the differences are 90 degrees, the solution is satisfactory.

15. See Figure 5.2

Figure 5.2 *Solution to Exercise 15*



a. This is ASK. There are two peak amplitudes both with the same phase (0 degrees). The values of the peak amplitudes are $A_1 = 2$ (the distance between

the first dot and the origin) and $A_2 = 3$ (the distance between the second dot and the origin).

b. This is BPSK, There is only one peak amplitude (3). The distance between each dot and the origin is 3. However, we have two phases, 0 and 180 degrees.

c. This can be either QPSK (one amplitude, four phases) or 4-QAM (one amplitude and four phases). The amplitude is the distance between a point and the origin, which is $(2^2 + 2^2)^{1/2} = 2.83$.

d. This is also BPSK. The peak amplitude is 2, but this time the phases are 90 and 270 degrees.

16. The number of points define the number of levels, L. The number of bits per baud is the value of r. Therefore, we use the formula **$r = \log_2 L$** for each case.

a. $\log_2 2 = 1$
b. $\log_2 4 = 2$
c. $\log_2 16 = 4$
d. $\log_2 1024 = 10$

17. We use the formula **$B = (1 + d) \cdot (1/r) \cdot N$**, but first we need to calculate the value of r for each case.

a. r = 1 → $B= (1 + 1) \cdot (1/1) \cdot (4000 \text{ bps}) =$ **8000 Hz**
b. r = 1 → $B = (1 + 1) \cdot (1/1) \cdot (4000 \text{ bps}) + 4 \text{ KHz} =$ **8000 Hz**
c. r = 2 → $B = (1 + 1) \cdot (1/2) \cdot (4000 \text{ bps}) =$ **2000 Hz**
d. r = 4 → $B = (1 + 1) \cdot (1/4) \cdot (4000 \text{ bps}) =$ **1000 Hz**

18. We use the formula **$N = [1/(1 + d)] \cdot r \cdot B$**, but first we need to calculate the value of r for each case.

a. $r = \log_2 2 = 1$ → $N= [1/(1 + 0)] \cdot 1 \cdot (4 \text{ KHz}) =$ **4 kbps**
b. $r = \log_2 4=2$ → $N = [1/(1 + 0)] \cdot 2 \cdot (4 \text{ KHz}) =$ **8 kbps**
c. $r = \log_2 16= 4$ → $N = [1/(1 + 0)] \cdot 4 \cdot (4 \text{ KHz}) =$ **16 kbps**
d. r = $\log_2 64= 6$ → $N = [1/(1 + 0)] \cdot 6 \cdot (4 \text{ KHz}) =$ **24 kbps**

19.

First, we calculate the bandwidth for each channel = (1 MHz) / 10 = 100 KHz. We then find the value of r for each channel:

$$B = (1 + d) \cdot (1/r) \cdot (N) \rightarrow r = N / B \rightarrow r = (1 \text{ Mbps}/100 \text{ KHz}) = 10$$

We can then calculate the number of levels: $L = 2^r = 2^{10} =$ **1024.** This means that that we need a **1024-QAM** technique to achieve this data rate.

20. We can use the formula: $N = [1/(1 + d)] \cdot r \cdot B = 1 \cdot 6 \cdot 6 \text{ MHz} = 36 \text{ Mbps}$ 21.

a. $B_{AM} = 2 \cdot B = 2 \cdot 5 =$ **10 KHz**

b. $B_{FM} = 2 \cdot (1 + \beta) \cdot B = 2 \cdot (1 + 5) \cdot 5 =$ **60 KHz**
c.$B_{PM} = 2 \cdot (1 + \beta) \cdot B = 2 \cdot (1 + 1) \cdot 5 =$ **20 KHz**

22. We calculate the number of channels, not the number of coexisting stations.

a. n = (1700 - 530) KHz / 10 KHz = **117**
b. n = (108 - 88) MHz / 200 KHz = **100**

CHAPTER 6

Bandwidth Utilization:

Solutions to Review Questions and Exercises

Review Questions

1. ***Multiplexing*** is the set of techniques that allows the simultaneous transmission of multiple signals across a single data link.
2. We discussed ***frequency-division multiplexing*** (**FDM**), ***wave-division multiplexing*** (**WDM**), and ***time-division multiplexing*** (**TDM**).
3. In ***multiplexing***, the word ***link*** refers to the physical path. The word ***channel*** refers to the portion of a link that carries a transmission between a given pair of lines. One link can have many (n) channels.
4. ***FDM*** and ***WDM*** are used to combine ***analog signals***; the bandwidth is shared. ***TDM*** is used to combine ***digital signals***; the time is shared.
5. To maximize the efficiency of their infrastructure, telephone companies have traditionally multiplexed analog signals from lower-bandwidth lines onto higher-bandwidth lines. The ***analog hierarchy*** uses voice channels (4 KHz), ***groups*** (48 KHz), ***supergroups*** (240 KHz), ***master groups*** (2.4 MHz), and ***jumbo groups*** (15.12 MHz).
6. To maximize the efficiency of their infrastructure, telephone companies have traditionally multiplexed digital signals from lower data rate lines onto higher data rate lines. The ***digital hierarchy*** uses ***DS-0*** (64 Kbps), ***DS-1*** (1.544 Mbps), ***DS-2*** (6.312 Mbps), ***DS-3*** (44.376 Mbps), and ***DS-4*** (274.176 Mbps).
7. ***WDM*** is common for multiplexing ***optical signals*** because it allows the multiplexing of signals with a very high frequency.
8. In ***multilevel TDM***, some lower-rate lines are combined to make a new line with the same data rate as the other lines. ***Multiple slot TDM***, on the other hand, uses multiple slots for higher data rate lines to make them compatible with the lower data rate line. ***Pulse stuffing TDM*** is used when the data rates of some lines are not an integral multiple of other lines.
9. In ***synchronous TDM***, each input has a reserved slot in the output frame. This can be inefficient if some input lines have no data to send. In ***statistical TDM***, slots are

dynamically allocated to improve bandwidth efficiency. Only when an input line has a slot's worth of data to send is it given a slot in the output frame.

10. In ***spread spectrum***, we spread the bandwidth of a signal into a larger bandwidth. Spread spectrum techniques add redundancy; they spread the original spectrum needed for each station. The expanded bandwidth allows the source to wrap its message in a protective envelope for a more secure transmission. We discussed ***frequency hopping spread spectrum*** (***FHSS***) and direct sequence spread spectrum (***DSSS***).

11. The ***frequency hopping spread spectrum*** (***FHSS***) technique uses *M* different carrier frequencies that are modulated by the source signal. At one moment, the signal modulates one carrier frequency; at the next moment, the signal modulates another carrier frequency.

12. The ***direct sequence spread spectrum*** (***DSSS***) technique expands the bandwidth of the original signal. It replaces each data bit with *n* bits using a spreading code.

Exercises

13. To multiplex 10 voice channels, we need nine guard bands. The required bandwidth is then B = (4 KHz) · 10 + (500 Hz) · 9 = **44.5 KHz**

14. The bandwidth allocated to each voice channel is 20 KHz / 100 = 200 Hz. As we saw in the previous chapters, each digitized voice channel has a data rate of 64 Kbps (8000 sample · 8 bit/sample). This means that our modulation technique uses 64,000/200 = 320 bits/Hz.

15.
   a. Group level: overhead = 48 KHz − (12 · 4 KHz) = **0 Hz**.
   b. Supergroup level: overhead = 240 KHz − (5 · 48 KHz) **= 0 Hz**.
   c. Master group: overhead = 2520 KHz − (10 · 240 KHz) = **120 KHz**.
   d. Jumbo Group: overhead = 16.984 MHz − (6 · 2.52 MHz) = **1.864 MHz**.

16.
   a. Each output frame carries 1 bit from each source plus one extra bit for synchronization. Frame size = 20 · 1 + 1 = **21 bits.**
   b. Each frame carries 1 bit from each source. Frame rate = **100,000 frames/s.**
   c. Frame duration = 1 /(frame rate) = 1 /100,000 = **10 μs**.
   d. Data rate = (100,000 frames/s) · (21 bits/frame) = **2.1 Mbps**
   e. In each frame 20 bits out of 21 are useful. Efficiency = 20/21= **95%**

17.
   a. Each output frame carries 2 bits from each source plus one extra bit for synchronization. Frame size = 20 · 2 + 1 = **41 bits.**
   b. Each frame carries 2 bit from each source. Frame rate = 100,000/2 **= 50,000 frames/s.**
   c. Frame duration = 1 /(frame rate) = 1 /50,000 = **20 μs**.
   d. Data rate = (50,000 frames/s) · (41 bits/frame) = **2.05 Mbps.** The output data rate here is slightly less than the one in Exercise 16.

   e. In each frame 40 bits out of 41 are useful. Efficiency = 40/41= **97.5%**. Effi ciency is better than the one in Exercise 16.

18.

a. Frame size = 6 · (8 + 4) = **72 bits**.

b. We can assume that we have only 6 input lines. Each frame needs to carry one character from each of these lines. This means that the frame rate is **500 frames/s**.

c. Frame duration = 1 /(frame rate) = 1 /500 = **2 ms**.

d. Data rate = (500 frames/s) · (72 bits/frame) = **36 kbps**.

19. We combine six 200-kbps sources into three 400-kbps. Now we have seven 400-kbps channel.

a. Each output frame carries 1 bit from each of the seven 400-kbps line. Frame size = 7 · 1 = **7 bits.**

b. Each frame carries 1 bit from each 400-kbps source. Frame rate = **400,000 frames/s.**

c. Frame duration = 1 /(frame rate) = 1 /400,000 = **2.5 μs**.

d. Output data rate = (400,000 frames/s) · (7 bits/frame) = **2.8 Mbps.** We can also calculate the output data rate as the sum of input data rate because there is no synchronizing bits. Output data rate = 6 · 200 + 4 · 400 = **2.8 Mbps**. 20.

a. The frame carries 4 bits from each of the first two sources and 3 bits from each of the second two sources. Frame size = 4 · 2 + 3 · 2 = **14 bits.**

b. Each frame carries 4 bit from each 200-kbps source or 3 bits from each 150 kbps. Frame rate = 200,000 / 4 = 150,000 /3 = **50,000 frames/s.**

c. Frame duration = 1 /(frame rate) = 1 /50,000 = **20 μs**.

d. Output data rate = (50,000 frames/s) · (14 bits/frame) = **700 kbps.** We can also calculate the output data rate as the sum of input data rates because there are no synchronization bits. Output data rate = 2 · 200 + 2 · 150 = 700 kbps.

21. We need to add extra bits to the second source to make both rates = 190 kbps. Now we have two sources, each of 190 Kbps.

a. The frame carries 1 bit from each source. Frame size = 1 + 1 = **2 bits.** b. Each frame carries 1 bit from each 190-kbps source. Frame rate = **190,000 frames/s.**

c. Frame duration = 1 /(frame rate) = 1 /190,000 = **5.3 μs**.

d. Output data rate = (190,000 frames/s) · (2 bits/frame) = **380 kbps.** Here the output bit rate is greater than the sum of the input rates (370 kbps) because of extra bits added to the second source.

22.

a. T-1 line sends 8000 frames/s. Frame duration = 1/8000 = **125 μs**.

b. Each frame carries one extra bit. Overhead = 8000 · 1 = **8 kbps**

23. See Figure 6.1.

24. See Figure 6.2.

Figure 6.1 *Solution to Exercise 23*

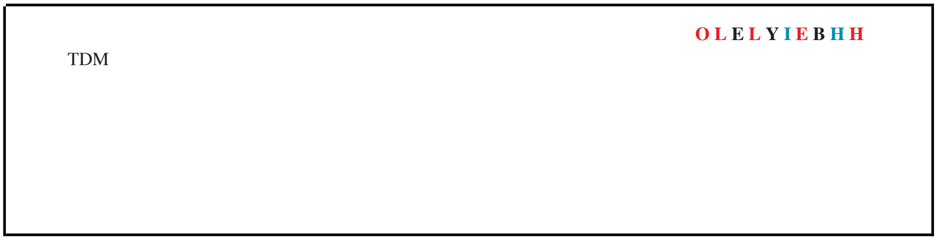
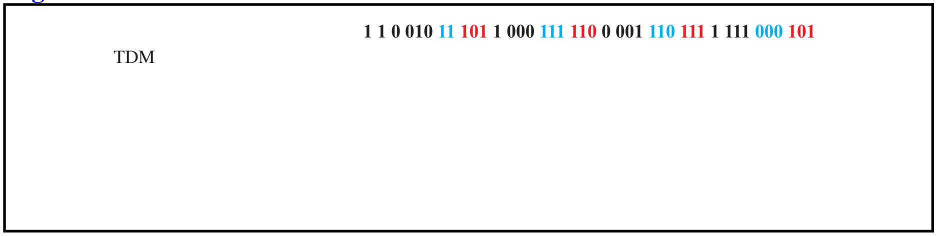
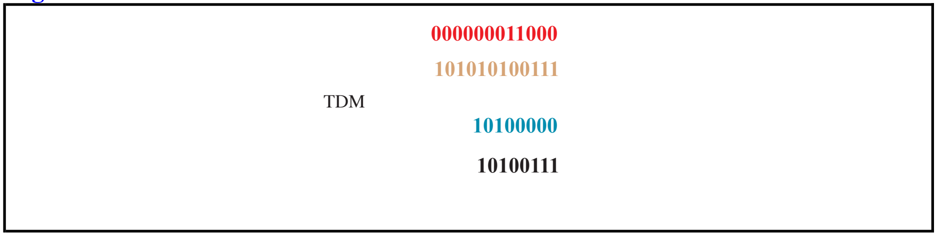


Figure 6.2 *Solution to Exercise 24*



25. See Figure 6.3.

Figure 6.3 *Solution to Exercise 25*



26.

a. DS-1 overhead = 1.544 Mbps − (24 · 64 kbps) = **8 kbps**.

b. DS-2 overhead = 6.312 Mbps − (4 · 1.544 Mbps) = **136 kbps**.

c. DS-3 overhead = 44.376 Mbps − (7 · 6.312 Mbps) = **192 kbps**.

d. DS-4 overhead = 274.176 Mbps − (6 · 44.376 Mbps) = **7.92 Mbps**.

27. The number of hops = 100 KHz/4 KHz = 25. So we need $\log_2 25 = 4.64 \approx$ **5 bits**

28.

a. 2^4 = **16 hops**

b. (64 bits/s) / 4 bits = **16 cycles**

29. Random numbers are 11, 13, 10, 6, 12, 3, 8, 9 as calculated below:

$N_1 = \mathbf{11}$
$N_2 = (5 + 7 \cdot \mathbf{11}) \bmod 17 - 1 = \mathbf{13}$
$N_3 = (5 + 7 \cdot \mathbf{13}) \bmod 17 - 1 = \mathbf{10}$

$N_4 = (5 + 7 \cdot \mathbf{10}) \bmod 17 - 1 = \mathbf{6}$

$N_5 = (5 + 7 \cdot \mathbf{6}) \bmod 17 - 1 = \mathbf{12}$
$N_6 = (5 + 7 \cdot \mathbf{12}) \bmod 17 - 1 = \mathbf{3}$
$N_7 = (5 + 7 \cdot \mathbf{3}) \bmod 17 - 1 = \mathbf{8}$
$N_8 = (5 + 7 \cdot \mathbf{8}) \bmod 17 - 1 = \mathbf{9}$

30. The Barker chip is 11 bits, which means that it increases the bit rate 11 times. A voice channel of 64 kbps needs 11 · 64 kbps = 704 kbps. This means that the bandpass channel can carry (10 Mbps) / (704 kbps) or approximately **14 channels**.

CHAPTER 7

Transmission Media

Solutions to Review Questions and Exercises

Review Questions

1. The ***transmission media*** is located ***beneath the physical layer*** and controlled by the physical layer.

2. The two major categories are ***guided*** and ***unguided*** media.

3. ***Guided media*** have physical boundaries, while ***unguided media*** are unbounded. 4. The three major categories of guided media are ***twisted-pair***, ***coaxial***, and ***fiber optic*** cables.

5. ***Twisting*** ensures that both wires are equally, but ***inversely***, affected by external influences such as noise.

6. ***Refraction*** and ***reflection*** are two phenomena that occur when a beam of light travels into a less dense medium. When the angle of incidence is less than the critical angle, ***refraction*** occurs. The beam crosses the interface into the less dense medium. When the angle of incidence is greater than the critical angle, ***reflection*** occurs. The beam changes direction at the interface and goes back into the more dense medium.

7. The ***inner core*** of an optical fiber is surrounded by ***cladding***. The core is denser than the cladding, so a light beam traveling through the core is reflected at the boundary between the core and the cladding if the incident angle is more than the critical angle.

8. We can mention three advantages of optical fiber cable over twisted-pair and coaxial cables: ***noise resistance***, ***less signal attenuation***, and ***higher bandwidth.***

9. In ***sky propagation*** radio waves radiate upward into the ionosphere and are then reflected back to earth. In ***line-of-sight propagation*** signals are transmitted in a

straight line from antenna to antenna.

10. ***Omnidirectional*** waves are propagated in all directions; ***unidirectional*** waves are propagated in one direction.

Exercises

11. See Table 7.1 (the values are approximate).

Table 7.1 *Solution to Exercise 11*

Distance	*dB at 1 KHz*	*dB at 10 KHz*	*dB at 100 KHz*
1 Km	**−3**	**−5**	**−7**
10 Km	**−30**	**−50**	**−70**
15 Km	**−45**	**−75**	**−105**
20 Km	**−60**	**−100**	**−140**

12. As the Table 7.1 shows, for a specific maximum value of attenuation, the highest frequency decreases with distance. If we consider the bandwidth to start from zero, we can say that the bandwidth decreases with distance. For example, if we can tol erate a maximum attenuation of 50 dB (loss), then we can give the following list ing of distance versus bandwidth.

Distance	**Bandwidth**
1 Km	100 KHz
10 Km	50 KHz
15 Km	1 KHz
20 Km	0 KHz

13. We can use Table 7.1 to find the power for different frequencies:

$$1 \text{ KHz} \quad \text{dB} = -3 \quad P_2 = P_1 \cdot 10^{-3/10} = \mathbf{100.23\ mw}$$
$$10 \text{ KHz} \quad \text{dB} = -5 \quad P_2 = P_1 \cdot 10^{-5/10} = \mathbf{63.25\ mw}$$
$$100 \text{ KHz} \quad \text{dB} = -7 \quad P_2 = P_1 \cdot 10^{-7/10} = \mathbf{39.90\ mw}$$

The table shows that the power is reduced 5 times, which may not be acceptable for some applications.

14. See Table 7.2 (the values are approximate).

Table 7.2 *Solution to Exercise 14*

Distance	*dB at 1 KHz*	*dB at 10 KHz*	*dB at 100 KHz*
1 Km	**−3**	**−7**	**−20**
10 Km	**−30**	**−70**	**−200**
15 Km	**−45**	**−105**	**−300**
20 Km	**−60**	**−140**	**−400**

15. As Table 7.2 shows, for a specific maximum value of attenuation, the highest frequency decreases with distance. If we consider the bandwidth to start from zero, we can say that the bandwidth decreases with distance. For example, if we can tolerate a maximum attenuation of 50 dB (loss), then we can give the following listing of distance versus bandwidth.

Distance	**Bandwidth**
1 Km	100 KHz
10 Km	1 KHz
15 Km	1 KHz
20 Km	0 KHz

16. We can use Table 7.2 to find the power for different frequencies:

1 KHz dB = −3 $P_2 = P_1 \cdot 10^{-3/10}$ = **100.23 mw** 10 KHz dB = −7 $P_2 = P_1$
$\cdot 10^{-7/10}$ = **39.90 mw** 100 KHz dB = −20 $P_2 = P_1 \cdot 10^{-20/10}$ = **2.00 mw**

The table shows that power is decreased 100 times for 100 KHz, which is unacceptable for most applications.

17. We can use the formula f = c / λ to find the corresponding frequency for each wave length as shown below (c is the speed of propagation):

a. $B = [(2 \cdot 10^8)/1000 \cdot 10^{-9}] - [(2 \cdot 10^8)/\, 1200 \cdot 10^{-9}]$ = **33 THz** b.
$B = [(2 \cdot 10^8)/1000 \cdot 10^{-9}] - [(2 \cdot 10^8)/\, 1400 \cdot 10^{-9}]$ = **57 THz** 18.

a. The ***wave length*** is the ***inverse*** of the ***frequency*** if the propagation speed is fixed (based on the formula λ = c / f). This means all three figures represent the same thing.

b. We can change the wave length to frequency. For example, the value 1000 nm can be written as 200 THz.

c. The vertical-axis units may not change because they represent dB/km.

d. The curve must be flipped horizontally.

19. See Table 7.3 (The values are approximate).

Table 7.3 *Solution to Exercise 19*

Distance	*dB at 800 nm*	*dB at 1000 nm*	*dB at 1200 nm*
1 Km	**−3**	**−1.1**	**−0.5**
10 Km	**−30**	**−11**	**−5**
15 Km	**−45**	**−16.5**	**−7.5**
20 Km	**−60**	**−22**	**−10**

20. The delay = distance / (propagation speed). Therefore, we have:

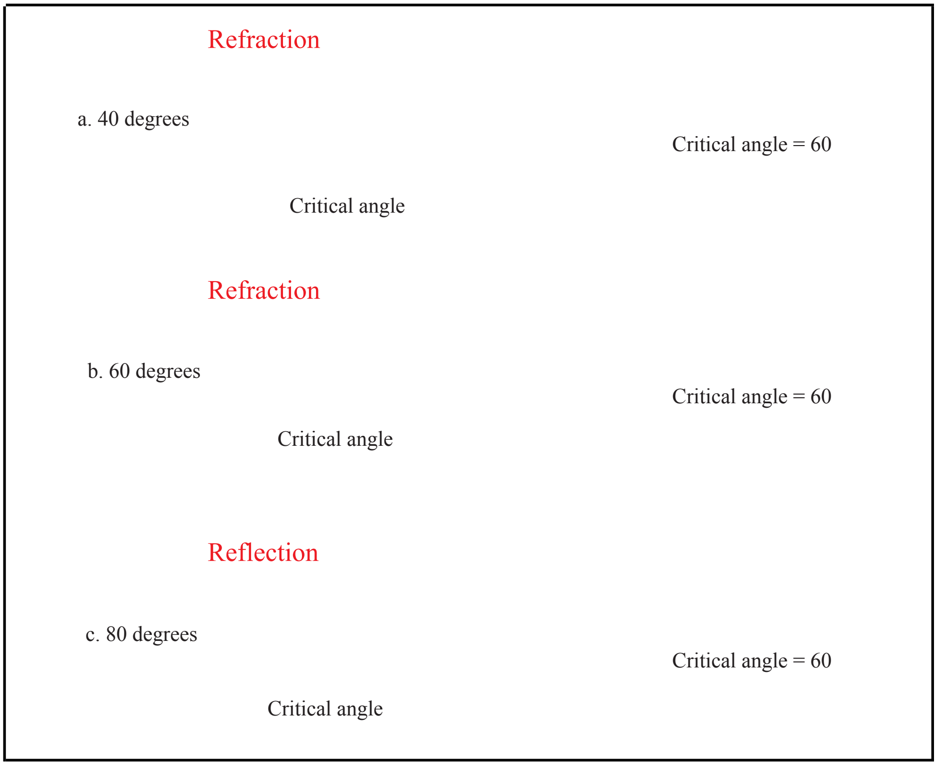
a. Delay = $10/(2 \cdot 10^8)$ = **0.05 ms**

b. Delay = $100/(2 \cdot 10^8)$ = **0.5 ms**

c. Delay = $1000/(2 \cdot 10^8)$ = **5 ms**

21. See Figure 7.1.

Figure 7.1 *Solution to Exercise 21*



a. The incident angle (40 degrees) is smaller than the critical angle (60 degrees). We have ***refraction***.The light ray enters into the less dense medium.

b. The incident angle (60 degrees) is the same as the critical angle (60 degrees). We have ***refraction***. The light ray travels along the interface.

c. The incident angle (80 degrees) is greater than the critical angle (60 degrees). We have ***reflection***. The light ray returns back to the more dense medium.

CHAPTER 8

Switching

Solutions to Review Questions and Exercises

Review Questions

1. ***Switching*** provides a practical solution to the problem of connecting multiple devices in a network. It is more practical than using a bus topology; it is more efficient than using a star topology and a central hub. Switches are devices capable of

creating temporary connections between two or more devices linked to the switch.

2. The three traditional switching methods are ***circuit switching***, ***packet switching***, and ***message switching***. The most common today are ***circuit switching*** and ***packet switching***.
3. There are two approaches to packet switching: ***datagram approach*** and ***virtual circuit approach***.
4. In a ***circuit-switched network***, data are not packetized; data flow is somehow a continuation of bits that travel the same channel during the data transfer phase. In a ***packet-switched network*** data are packetized; each packet is somehow an independent entity with its local or global addressing information.
5. The address field defines the ***end-to-end*** (source to destination) addressing.
6. The address field defines the ***virtual circuit number*** (local) addressing.
7. In a ***space-division*** switch, the path from one device to another is spatially separate from other paths. The inputs and the outputs are connected using a grid of electronic microswitches. In a ***time-division*** switch, the inputs are divided in time using TDM. A control unit sends the input to the correct output device.
8. ***TSI*** (time-slot interchange) is the most popular technology in a time-division switch. It used random access memory (RAM) with several memory locations. The RAM fills up with incoming data from time slots in the order received. Slots are then sent out in an order based on the decisions of a control unit.
9. In multistage switching, ***blocking*** refers to times when one input cannot be connected to an output because there is no path available between them—all the possible intermediate switches are occupied. One solution to blocking is to increase the number of intermediate switches based on the Clos criteria.

10. A packet switch has four components: ***input ports***, ***output ports***, the ***routing processor***, and the ***switching fabric***. An input port performs the physical and data link functions of the packet switch. The output port performs the same functions as the input port, but in the reverse order. The routing processor performs the function of table lookup in the network layer. The switching fabric is responsible for moving the packet from the input queue to the output queue.

Exercises

11. We assume that the setup phase is a two-way communication and the teardown phase is a one-way communication. These two phases are common for all three cases. The delay for these two phases can be calculated as three propagation delays and three transmission delays or

$$3\,[(5000 \text{ km})/\ (2 \cdot 10^8 \text{ m/s})] + 3\,[(1000 \text{ bits}/1 \text{ Mbps})] = 75 \text{ ms} + 3 \text{ ms} = \mathbf{78\ ms}$$

We assume that the data transfer is in one direction; the total delay is then delay for setup and teardown + propagation delay + transmission delay

a. 78 + 25 + 1 = ***104 ms***

b. 78 + 25 + 100 = ***203 ms***

c. 78 + 25 + 1000 = ***1103 ms***

d. In case a, we have 104 ms. In case b we have 203/100 = 2.03 ms. In case c, we have 1103/1000 = 1.101 ms. The ratio for case c is the smallest because we use one setup and teardown phase to send more data.

12. We assume that the transmission time is negligible in this case. This means that we suppose all datagrams start at time 0. The arrival timed are calculated as:

First: (3200 Km) / ($2 \cdot 10^8$ m/s) + (3 + 20 + 20) = **59.0 ms**

Second: (11700 Km) / ($2 \cdot 10^8$ m/s) + (3 + 10 + 20) = **91.5 ms**

Third: (12200 Km) / ($2 \cdot 10^8$ m/s) + (3 + 10+ 20 + 20) = **114.0 ms**

Fourth: (10200 Km) / ($2 \cdot 10^8$ m/s) + (3 + 7 + 20) = **81.0 ms**

Fifth: (10700 Km) / ($2 \cdot 10^8$ m/s) + (3 + 7 + 20 + 20) = **103.5 ms**

The order of arrival is: **3** → **5** → **2** → **4** → **1**

13.

a. In a ***circuit-switched*** network, end-to-end addressing is needed during the setup and teardown phase to create a connection for the whole data transfer phase. After the connection is made, the data flow travels through the already-reserved resources. The switches remain connected for the entire duration of the data transfer; there is no need for further addressing.

b. In a ***datagram network***, each packet is independent. The routing of a packet is done for each individual packet. Each packet, therefore, needs to carry an end to-end address. There is no setup and teardown phases in a datagram network (connectionless transmission). The entries in the routing table are somehow permanent and made by other processes such as routing protocols.

c. In a ***virtual-circuit*** network, there is a need for end-to-end addressing during the setup and teardown phases to make the corresponding entry in the switching table. The entry is made for each request for connection. During the data trans fer phase, each packet needs to carry a virtual-circuit identifier to show which virtual-circuit that particular packet follows.

14. A ***datagram*** or ***virtual-circuit*** network handles packetized data. For each packet, the switch needs to consult its table to find the output port in the case of a datagram network, and to find the combination of the output port and the virtual circuit iden tifier in the case of a virtual-circuit network. In a ***circuit-switched*** network, data are not packetized; no routing information is carried with the data. The whole path is established during the setup phase.

15. In ***circuit-switched*** and ***virtual-circuit*** networks, we are dealing with connections. A connection needs to be made before the data transfer can take place. In the case of a circuit-switched network, a physical connection is established during the setup phase and the is broken during the teardown phase. In the case of a

virtual-circuit network, a virtual connection is made during setup and is broken during the tear down phase; the connection is virtual, because it is an entry in the table. These two types of networks are considered ***connection-oriented***. In the case of a ***datagram*** network no connection is made. Any time a switch in this type of network receives a packet, it consults its table for routing information. This type of network is con sidered a ***connectionless*** network.

16. The switching or routing in a ***datagram network*** is based on the final destination address, which is global. The minimum number of entries is two; one for the final destination and one for the output port. Here the input port, from which the packet has arrived is irrelevant. The switching or routing in a ***virtual-circuit*** network is based on the virtual circuit identifier, which has a local jurisdiction. This means that two different input or output ports may use the same virtual circuit number. Therefore, four pieces of information are required: input port, input virtual circuit number, output port, and output virtual circuit number.

17.

Packet 1: **2**
Packet 2: **3**
Packet 3: **3**
Packet 4: **2**

18.

Packet 1: **2, 70**
Packet 2: **1, 45**
Packet 3: **3, 11**
Packet 4: **4, 41**

19.

a. In a ***datagram*** network, the destination addresses are unique. They cannot be duplicated in the routing table.

b. In a ***virtual-circuit*** network, the VCIs are local. A VCI is unique only in relationship to a port. In other words, the (port, VCI) combination is unique. This means that we can have two entries with the same input or output ports. We can

have two entries with the same VCIs. However, we cannot have two entries with the same (port, VCI) pair.

20. When a packet arrives at a router in a ***datagram*** network, the only information in the packet that can help the router in its routing is the ***destination address*** of the packet. The table then is sorted to make the searching faster. Today's routers use some sophisticated searching techniques. When a packet arrives at a switch in a ***virtual-circuit*** network, the pair ***(input port, input VCI)*** can uniquely determined how the packet is to be routed; the pair is the only two pieces of information in the packet that is used for routing. The table in the virtual-circuit switch is sorted based on the this pair. However, since the number of port numbers is normally much smaller than the number of virtual circuits assigned to each port, sorting is done in two steps: first according to the input port number and second according to the input VCI.

21.

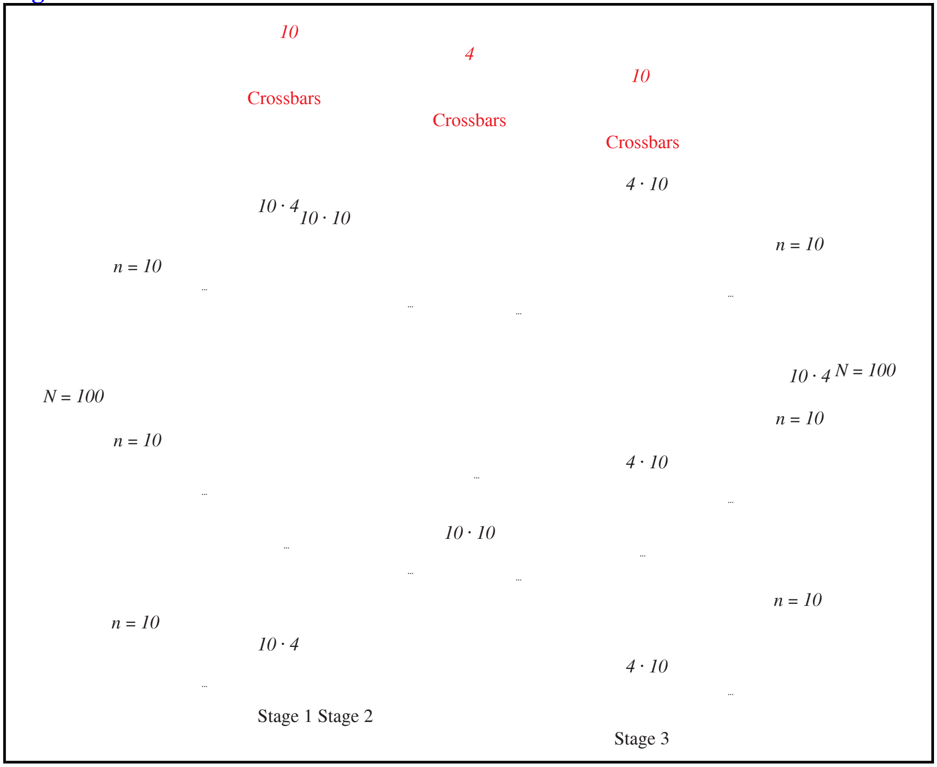
a. If $n > k$, an $n \cdot k$ crossbar is like a ***multiplexer*** that combines n inputs into k out puts. However, we need to know that a regular multiplexer discussed in Chapter 6 is $n \cdot 1$.

b. If $n < k$, an $n \cdot k$ crossbar is like a ***demultiplexer*** that divides n inputs into k outputs. However, we need to know that a regular demultiplexer discussed in Chapter 6 is $1 \cdot n$.

22.

a. See Figure 8.1.

Figure 8.1 *Solution to Exercise 22 Part a*



b. The total number of crosspoints are

$$\text{Number of crosspoints} = 10\,(10 \cdot 4) + 4\,(10 \cdot 10) + 10\,(4 \cdot 10) = \mathbf{1200}$$

c. Only four simultaneous connections are possible for each crossbar at the first stage. This means that the total number of simultaneous connections is **40.**

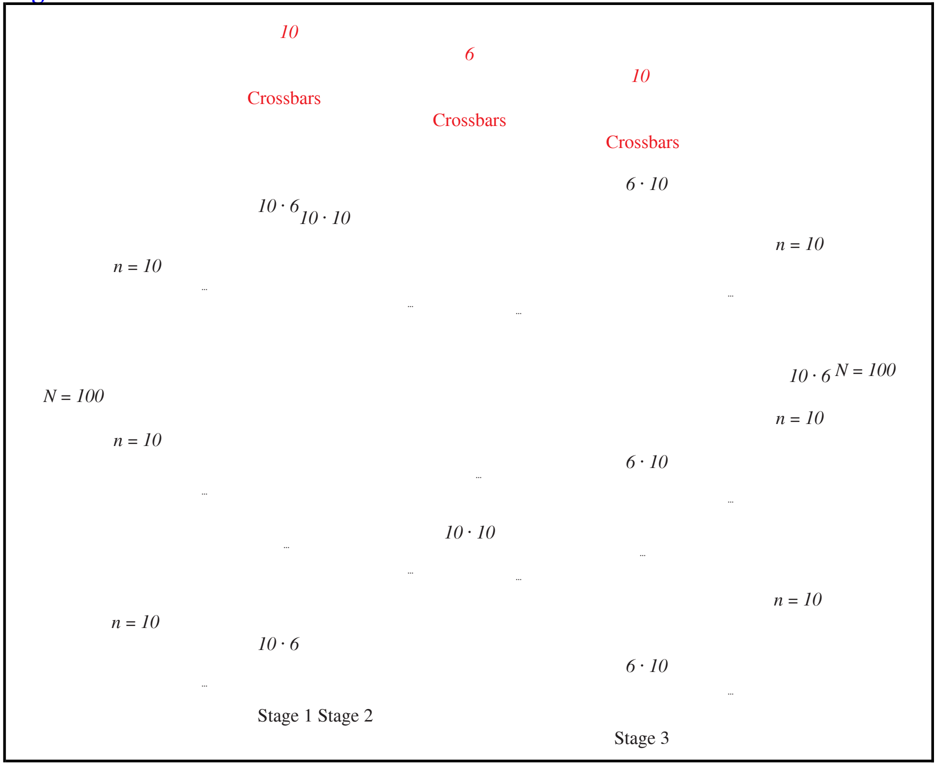
d. If we use one crossbar ($100 \cdot 100$), all input lines can have a connection at the same time, which means **100** simultaneous connections.

e. The blocking factor is 40/100 or ***40 percent***.

23.

a. See Figure 8.2.

Figure 8.2 *Solution to Exercise 23 Part a*



b. The total number of crosspoints are

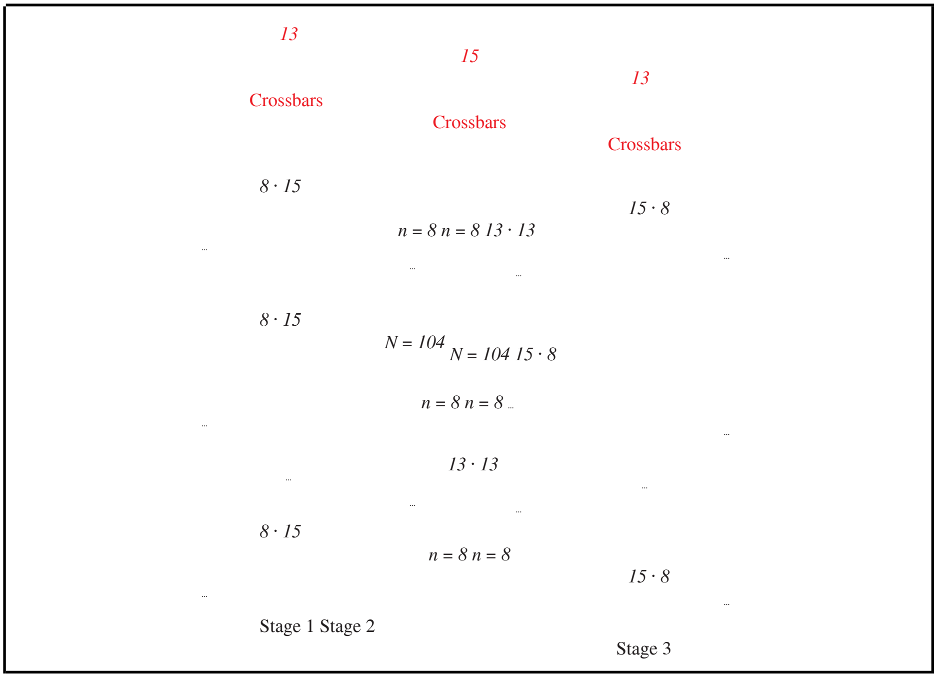
Number of crosspoints = 10 (10 · 6) + 6 (10 · 10) + 10 (6 · 10) = **1800**

c. Only six simultaneous connections are possible for each crossbar at the first stage. This means that the total number of simultaneous connections is **60**. d. If we use one crossbar (100 · 100), all input lines can have a connection at the same time, which means **100** simultaneous connections.

e. The blocking factor is 60/100 or ***60 percent.***

24. According to Clos, n = $(N/2)^{1/2}$ = 7.07. We can choose n = 8. The number of cross bars in the first stage can be 13 (to have similar crossbars). Some of the input lines can be left unused. We then have $k = 2n - 1 = 15$. Figure 8.3 shows the configura tion.

Figure 8.3 *Solution to Exercise 24 Part a*



We can calculate the total number of crosspoints as

6

$$13\ (8 \cdot 15) + 15\ (13 \cdot 13) + 13\ (15 \cdot 8) = \mathbf{\mathit{5655}}$$

The number of crosspoints is still much less than the case with one crossbar (10,000). We can see that there is no blocking involved because each 8 input line has 15 intermediate crossbars. The total number of crosspoints here is a little greater than the minimum number of crosspoints according to Clos using the formula $4N[(2N)^{1/2} - 1]$, which is 5257.

25.

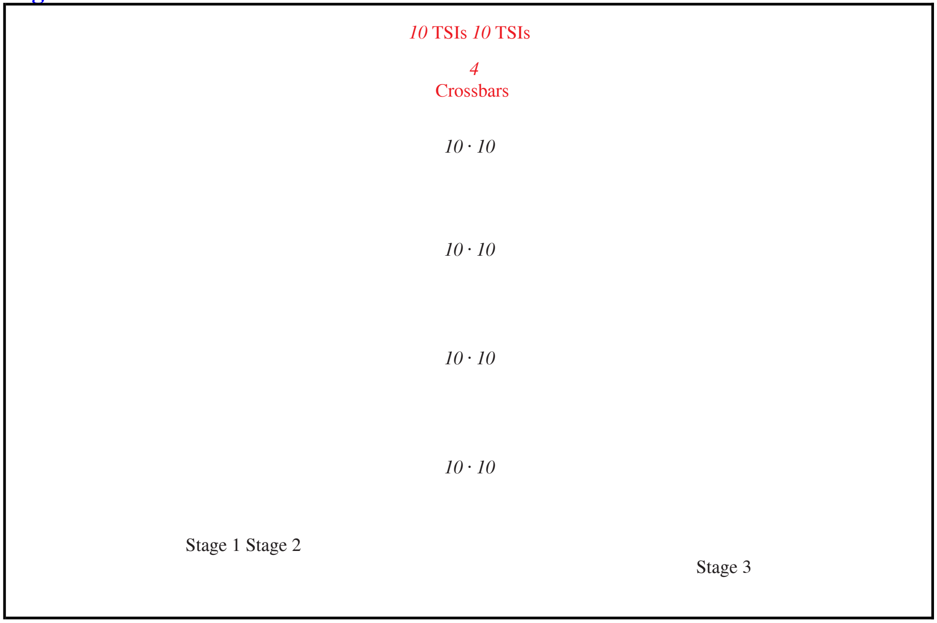
a. Total crosspoints = $N^2 = 1000^2 =$ ***1,000,000***

b. Total crosspoints ≥ $4N[(2N)^{1/2} - 1]$ ≥ ***174,886.*** With less than 200,000 crosspoints we can design a three-stage switch. We can use n = $(N/2)^{1/2}$ =23 and choose k = 45. The total number of crosspoints is ***178,200***.

26. We give two solutions.

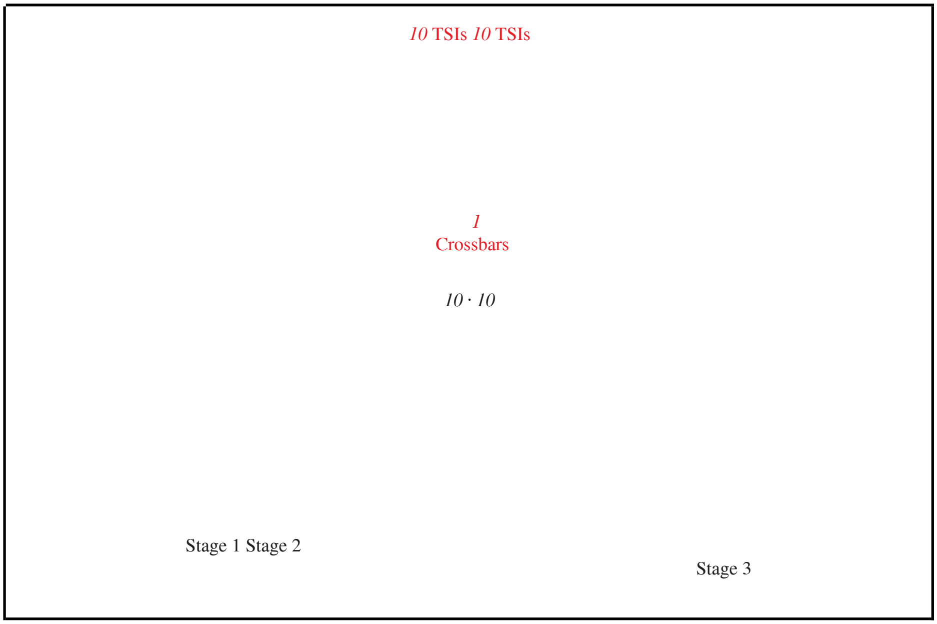
a. We first solve the problem using only crossbars and then we replace the crossbars at the first and the last stage with TSIs. Figure 8.1 shows the solution using only crossbars. We can replace the crossbar at the first and third stages with TSIs as shown in Figure 8.4. The total number of crosspoints is ***400*** and the total number of memory locations is ***200***. Each TSI at the first stage needs one TDM multiplexer and one TDM demultiplexer. The multiplexer is 10 · 1, but the demultiplexer is 1 · 4. In other words, the input frame has 10 slots and the output frame has only 4 slots. The data in the first slot of all input TSIs are directed to the first switch, the output in the second slot are directed to the second switch, and so on.

Figure 8.4 *First solution to Exercise 26*



b. We can see the inefficiency in the first solution. Since the slots are separated in time, only one of the switches at the middle stage is active at each moment. This means that, instead of 4 crossbars, we could have used only one with the same result. Figure 8.5 shows the new design. In this case we still need ***200*** memory locations but only ***100*** crosspoints.

Figure 8.5 *Second solution to Exercise 26*



CHAPTER 9

Using Telephone and Cable Networks ***Solutions to Review Questions and Exercises***

Review Questions

1. The telephone network is made of three major components: ***local loops***, ***trunks***, and ***switching offices***.
2. The telephone network has several levels of switching offices such as ***end offices***, ***tandem offices***, and ***regional offices***.
3. A ***LATA*** is a small or large metropolitan area that according to the divestiture of 1984 was under the control of a single telephone-service provider. The services offered by the common carriers inside a LATA are called intra-LATA services. The services between LATAs are handled by interexchange carriers (IXCs). These carriers, sometimes called long-distance companies, provide communication services between two customers in different LATAs.

4. ***Signaling System Seven (SS7)*** is the protocol used to provide signaling services in the telephone network. It is very similar to the five-layer Internet model.
5. Telephone companies provide two types of services: ***analog*** and ***digital***.
6. ***Dial-up modems*** use part of the bandwidth of the local loop to transfer data. The latest dial-up modems use the V-series standards such as V.32 and V.32bis (9600 bps), V.34bis (28,800 or 33,600 bps), V.90 (56 kbps for downloading and 33.6 kbps for uploading), and V.92. (56 kbps for downloading and 48 kbps for uploading.
7. Telephone companies developed ***digital subscriber line (DSL)*** technology to provide higher-speed access to the Internet. DSL technology is a set of technologies, each differing in the first letter (ADSL, VDSL, HDSL, and SDSL). The set is often referred to as xDSL, where x can be replaced by A, V, H, or S. DSL uses a device called ***ADSL modem*** at the customer site. It uses a device called a ***digital sub scriber line access multiplexer (DSLAM)*** at the telephone company site.
8. The ***traditional cable networks*** use only coaxial cables to distribute video information to the customers. The ***hybrid fiber-coaxial (HFC) networks*** use a combination of fiber-optic and coaxial cable to do so.

9. To provide Internet access, the cable company has divided the available bandwidth of the coaxial cable into three bands: video, downstream data, and upstream data. The ***downstream-only video band*** occupies frequencies from 54 to 550 MHz. ***The downstream data*** occupies the upper band, from 550 to 750 MHz. The ***upstream data*** occupies the lower band, from 5 to 42 MHz.
10. The ***cable modem (CM)*** is installed on the subscriber premises. The ***cable modem transmission system (CMTS)*** is installed inside the distribution hub by the cable company. It receives data from the Internet and passes them to the combiner, which sends them to the subscriber. The CMTS also receives data from the subscriber and passes them to the Internet.

Exercises

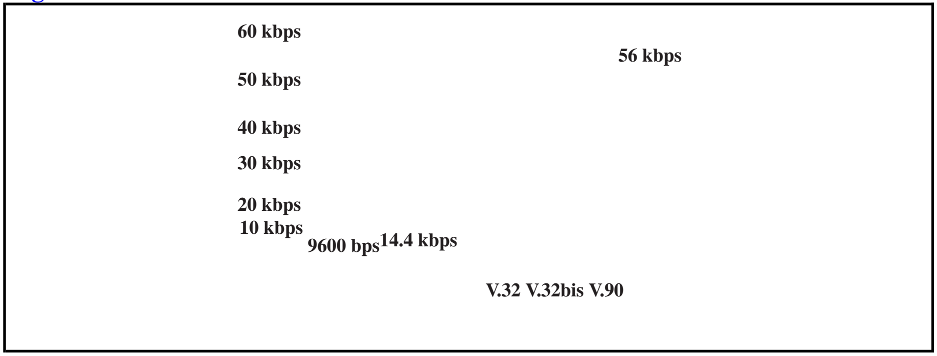
11. ***Packet-switched*** networks are well suited for carrying data in packets. The end-to-end addressing or local addressing (VCI) occupies a field in each packet. Telephone networks were designed to carry voice, which was not packetized. A ***circuit-switched*** network, which dedicates resources for the whole duration of the conversation, is more suitable for this type of communication.
12. ***The setup phase*** can be matched to the dialing process. After the callee responds, the ***data transfer phase*** (here voice transfer phase) starts. When any of the parties hangs up, the data transfer is terminated and the ***teardown phase*** starts. It takes a while before all resources are released.
13. In a telephone network, the ***telephone numbers*** of the caller and callee are serving

as source and destination addresses. These are used only during the setup (dialing) and teardown (hanging up) phases.

14. The ***delay*** can be attributed to the fact that some telephone companies use ***satellite*** networks for overseas communication. In these case, the signals need to travel sev eral thousands miles (earth station to satellite and satellite to earth station).

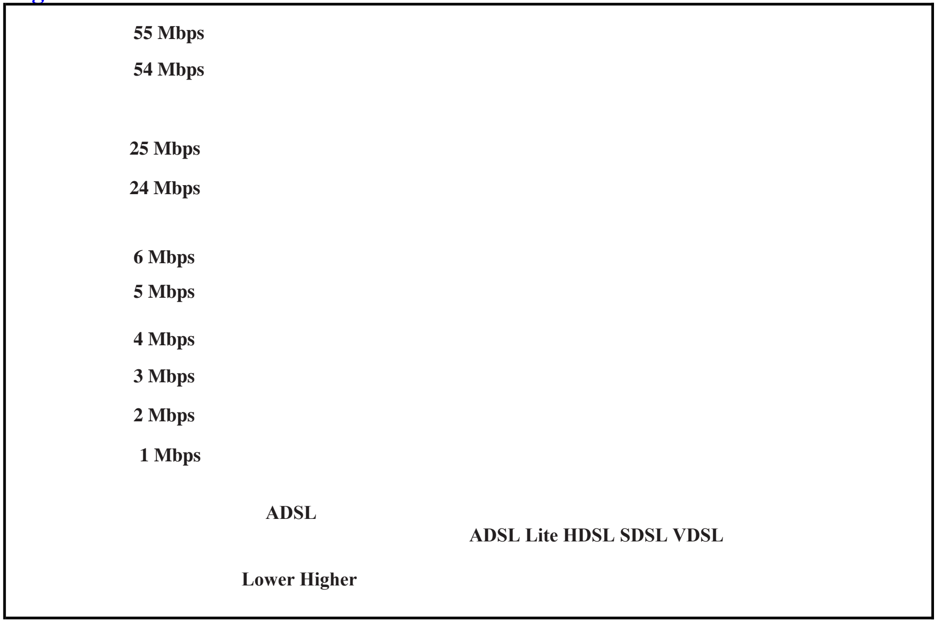
15. See Figure 9.1.

Figure 9.1 *Solution to Exercise 15*



16. See Figure 9.2.

Figure 9.2 *Solution to Exercise 16*



17.

a. V.32 → Time = (1,000,000 · 8) /9600 ≈ 834 s b. V.32bis → Time = (1,000,000 · 8) / 14400 ≈ 556 s c. V.90 → Time = (1,000,000 · 8) / 56000 ≈ 143 s

18.

≈ 5.3 s

a. ADSL → Time = (1,000,000 · 8) / 1,500,000

b. ADSL Lite → ≈ 5.3 s

Time = (1,000,000 · 8) / 1,500,000

c. HDSL → ≈ 5.3 s

Time = (1,000,000 · 8) / 1,500,000

d. SDSL → ≈ 10.42 s

Time = (1,000,000 · 8) / 768,000

e. VDSL → ≈ 0.32 sTime = (1,000,000 · 8) / 25,000,000

19. We can calculate time based on the assumption of 10 Mbps data rate:

Time = (1,000,000 · 8) / 10,000,000 ≈ 0.8 seconds

20. The ***DSL*** technology is based on ***star*** topology with the hub at the telephone office. The local loop connects each customer to the end office. This means that there is no sharing; the allocated bandwidth for each customer is not shared with neigh bors. The data rate does not depend on how many people in the area are transfer ring data at the same time.

21. The ***cable modem*** technology is based on the ***bus*** (or rather tree) topology. The cable is distributed in the area and customers have to share the available band width. This means if all neighbors try to transfer data, the effective data rate will be decreased.

CHAPTER 10

Error Detection and Correction

Solutions to Review Questions and Exercises

Review Questions

1. In a ***single bit error*** only one bit of a data unit is corrupted; in a **burst error** more than one bit is corrupted (not necessarily contiguous).
2. ***Redundancy*** is a technique of adding extra bits to each data unit to determine the accuracy of transmission.
3. In ***forward error correction***, the receiver tries to correct the corrupted codeword; in ***error detection by retransmission***, the corrupted message is discarded (the sender needs to retransmit the message).
4. A ***linear block code*** is a block code in which the exclusive-or of any two code words results in another codeword. A ***cyclic code*** is a linear block code in which the rotation of any codeword results in another codeword.
5. The ***Hamming distance*** between two words (of the same size) is the number of differences between the corresponding bits. The Hamming distance can easily be found if we apply the XOR operation on the two words and count the number of 1s in the result. The ***minimum Hamming distance*** is the smallest Hamming distance between all possible pairs in a set of words.
6. The ***single parity check*** uses one redundant bit for the whole data unit. In a ***two dimensional parity check***, original data bits are organized in a table of rows and columns. The parity bit is then calculated for each column and each row.
7.
    a. The only relationship between the size of the codeword and dataword is the one based on the definition: **n = k + r**., where ***n*** is the size of the codeword, ***k*** is the size of the dataword, and ***r*** is the size of the remainder.

    b. The ***remainder*** is always ***one bit smaller*** than the ***divisor***.

    c. The ***degree*** of the generator polynomial is ***one less than*** the size of the ***divisor***. For example, the CRC-32 generator (with the polynomial of degree 32) uses a 33-bit divisor.

    d. The ***degree*** of the generator polynomial is the ***same as*** the size of the remainder (length of checkbits). For example, CRC-32 (with the polynomial of degree 32) creates a remainder of 32 bits.

8. ***One’s complement arithmetic*** is used to add data items in checksum calculation. In this arithmetic, when a number needs more than *n* bits, the extra bits are wrapped and added to the number. In this arithmetic, the complement of a number is made by inverting all bits.
9. ***At least three types of error*** cannot be detected by the current checksum calculation. First, if two data items are swapped during transmission, the sum and the checksum values will not change. Second, if the value of one data item is increased (intentionally or maliciously) and the value of another one is decreased (intention ally or maliciously) the same amount, the sum and the checksum cannot detect these changes. Third, if one or more data items is changed in such a way

that the change is a multiple of $2^{16} - 1$, the sum or the checksum cannot detect the changes.

10. ***The value of a checksum can be all 0s*** (in binary). This happens when the value of the sum (after wrapping) becomes all 1s (in binary). ***It is almost impossible for the value of a checksum to be all 1s***. For this to happen, the value of the sum (after wrapping) must be all 0s which means all data units must be 0s.

Exercises

11. We can say that **(vulnerable bits) = (data rate) · (burst duration)**

a. vulnerable bits = $(1{,}500) \cdot (2 \cdot 10^{-3})$ = **3 bits**
b. vulnerable bits = $(12 \cdot 10^{3}) \cdot (2 \cdot 10^{-3})$ = **24 bits**
c. vulnerable bits = $(100 \cdot 10^{3}) \cdot (2 \cdot 10^{-3})$ = **200 bits**
d. vulnerable bits = $(100 \cdot 10^{6}) \cdot (2 \cdot 10^{-3})$ = **200,000 bits**

Comment: The last example shows how a noise of small duration can affect so many bits if the data rate is high.

12.

a. $(10001) \oplus (10000) =$ **00001**
b. $(10001) \oplus (10001) =$ **00000**
c. $(11100) \oplus (00000) =$ **11100**
d. $(10011) \oplus (11111) =$ **01100**

Comment: The above shows three properties of the exclusive-or operation. First, the result of XORing two equal patterns is an all-zero pattern (part b). Second, the result of XORing of any pattern with an all-zero pattern is the original non-zero pattern (part c). Third, the result of XORing of any pattern with an all-one pattern is the complement of the original non-one pattern.

13. The codeword for dataword **10** is **101**. This codeword will be changed to **010** if a 3-bit burst error occurs. This pattern is not one of the valid codewords, so the receiver detects the error and discards the received pattern.

14. The codeword for dataword **10** is **10101**. This codeword will be changed to **01001** if a 3-bit burst error occurs. This pattern is not one of the valid codewords, so the receiver discards the received pattern.

15.

a. d (10000, 00000) = **1**

b. d (10101, 10000) = **2**

c. d (1111, 1111) = **0**

d. d (000, 000) = **0**

Comment: Part c and d show that the distance between a codeword and itself is 0.

16.

a. For error detection → $d_{min} = s + 1 = 2 + 1 =$ **3**

b. For error correction → $d_{min} = 2t + 1 = 2 \cdot 2 + 1 = \mathbf{5}$

c.

For error section → $d_{min} = s + 1 = 3 + 1 = \mathbf{4}$

For error correction → $d_{min} = 2t + 1 = 2 \cdot 2 + 1 = \mathbf{5}$

Therefore d_{min} should be **5**.

d.

For error detection → $d_{min} = s + 1 = 6 + 1 = \mathbf{7}$

For error correction → $d_{min} = 2t + 1 = 2 \cdot 2 + 1 = \mathbf{5}$

Therefore d_{min} should be **7**.

17.

a. **01**

b. **error**

c. **00**

d. **error**

18. We show that the exclusive-or of the second and the third code word

$$(01011) \oplus (10111) = \mathbf{11100}$$

is not in the code. The code is not linear.

19. We check five random cases. All are in the code.

I. (1st) ⊕ (2nd) = (2nd) II. (2nd) ⊕ (3th) = (4th) III. (3rd) ⊕ (4th) = (2nd) IV. (4th) ⊕ (5th) = (8th) V. (5th) ⊕ (6th) = (2nd)

20. We show the dataword, the codeword, the corrupted codeword, and the interpretation of the receiver for each case:

a. Dataword: **0100** → Codeword: **0100011** → Corrupted: **0010011**
This pattern is not in the table. → **Correctly discarded**.

b. Dataword: **0111** → Codeword: **0111001** → Corrupted: **1111000**
This pattern is not in the table. → **Correctly discarded.**

c. Dataword: **1111** → Codeword: **1111111** → Corrupted: **0101110**
This pattern is in the table. → **Erroneously accepted as 0101.**

d. Dataword: **0000** → Codeword: **0000000** → Corrupted: **1101000**
This pattern is in the table. → **Erroneously accepted as 1101.**

Comment: The above result does not mean that the code can never detect three errors. The last two cases show that it may happen that three errors remain undetected.

21. We show the dataword, codeword, the corrupted codeword, the syndrome, and the interpretation of each case:

a. Dataword: 0100 → Codeword: 0100011 → Corrupted: **1100011** → $s_2s_1s_0 = 110$
Change b_3 (Table 10.5) → Corrected codeword: **0100011** → dataword: 0100
The dataword is correctly found.

b. Dataword: 0111 → Codeword: 0111001 → Corrupted: **0011001** → $s_2s_1s_0 = 011$
Change b_2 (Table 10.5) → Corrected codeword: **0111001** → dataword: 0111

The dataword is correctly found.

c. Dataword: 1111 → Codeword: 1111111 → Corrupted: **0111110** → $s_2s_1s_0 = 111$
Change b_1 (Table 10.5) → Corrected codeword: **0101110** → dataword: 0101
The dataword is found, but it is **incorrect**. C(7,4) cannot correct two errors.

d. Dataword: 0000 → Codeword: 0000000 → Corrupted: **1100001** → $s_2s_1s_0 = 100$
Change q_2 (Table 10.5) → Corrected codeword: **1100101** → dataword: 1100
The dataword is found, but it is **incorrect**. C(7,4) cannot correct three errors.

22.

a. If we rotate **0101100** one bit, the result is **0010110**, which is in the code. If we rotate **0101100** two bits, the result is **0001011**, which is in the code. And so on. b. The XORing of the two codewords $(0010110) \oplus (1111111) = 1101001$, which is in the code.

23. We need to find $k = 2^m - 1 - m \geq 11$. We use *trial and error* to find the right answer:

a. Let m = 1 $k = 2^m - 1 - m = 2^1 - 1 - 1 = 0$ (not acceptable)

b. Let m = 2 $k = 2^m - 1 - m = 2^2 - 1 - 2 = 1$ (not acceptable)

c. Let m = 3 $k = 2^m - 1 - m = 2^3 - 1 - 3 = 4$ (not acceptable)

d. Let m = 4 $k = 2^m - 1 - m = 2^4 - 1 - 4 = 11$ (acceptable)

Comment: The code is **C(15, 11)** with **$d_{min} = 3$**.

24.

a. $(x^3 + x^2 + x + 1) + (x^4 + x^2 + x + 1) = x^4 + x^3$

b. $(x^3 + x^2 + x + 1) - (x^4 + x^2 + x + 1) = x^4 + x^3$

c. $(x^3 + x^2) \cdot (x^4 + x^2 + x + 1) = x^7 + x^6 + x^5 + x^2$

d. $(x^3 + x^2 + x + 1) / (x^2 + 1) = x + 1$ (remainder is 0)

25.

a. 101110 → $x^5 + x^3 + x^2 + x$

b. 101110 → 101110**000** (Three 0s are added to the right)

c. $x^3 \cdot (x^5 + x^3 + x^2 + x) = x^8 + x^6 + x^5 + x^4$

d. 101110 → 10 (The four rightmost bits are deleted)

e. $x^{-4} \cdot (x^5 + x^3 + x^2 + x) = x$ (Note that negative powers are deleted) 26. To detect single bit errors, a CRC generator must have at least two terms and the coefficient of x^0 must be nonzero.

a. $x^3 + x + 1$ → It meets both critic.

b. $x^4 + x^2$ → It meets the first criteria, but not the second.

c. 1 → It meets the second criteria, but not the first.

d. $x^2 + 1$ → It meets both criteria.

27. CRC-8 generator is $x^8 + x^2 + x + 1$.

a. It has more than one term and the coefficient of x^0 is 1. It can detect a single-bit error.

b. The polynomial is of degree 8, which means that the number of checkbits (remainder) r = 8. It will detect all burst errors of size 8 or less.

c. Burst errors of size 9 are detected most of the time, but they slip by with proba bility $(1/2)^{r-1}$ or $(1/2)^{8-1} \approx 0.008$. This means **8 out of 1000** burst errors of size 9 are left undetected.

d. Burst errors of size 15 are detected most of the time, but they slip by with prob ability $(1/2)^{r}$ or $(1/2)^{8} \approx 0.004$. This means **4 out of 1000** burst errors of size 15 are left undetected.

28. This generator is $x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^{8} + x^{7} + x^{5} + x^{4} + x^{2} + x + 1$.

a. It has more than one term and the coefficient of x^0 is 1. It detects all single-bit error.

b. The polynomial is of degree 32, which means that the number of checkbits (remainder) r = 32. It will detect all burst errors of size 32 or less. c. Burst errors of size 33 are detected most of the time, but they are slip by with probability $(1/2)^{r-1}$ or $(1/2)^{32-1} \approx 465 \cdot 10^{-12}$. This means **465 out of 10^{12}** burst errors of size 33 are left undetected.

d. Burst errors of size 55 are detected most of the time, but they are slipped with probability $(1/2)^{r}$ or $(1/2)^{32} \approx 233 \cdot 10^{-12}$. This means **233 out of 10^{12}** burst errors of size 55 are left undetected.

29. We need to add all bits modulo-2 (XORing). However, it is simpler to count the number of 1s and make them even by adding a 0 or a 1. We have shown the parity bit in the codeword in color and separate for emphasis.

Dataword Number of 1s Parity Codeword a. 1001011 → 4 (even)
→ **0 0** 1001011 b. 0001100 → 2 (even) → **0 0** 0001100 c. 1000000 → 1
(odd) → **1 1** 1000000 d. 1110111 → 6 (even) → **0 0** 1110111

30. Figure 10.1 shows the calculation of the checksum both at the sender and receiver site using binary division.

6

Figure 10.1 *Solution to Exercise 30*

Codeword [redacted]

Dataword

[redacted]

1 111 0 0 1

1 111 0 0 1

Quotient

Quotient

1 111 0

1 111 0

11 1

0 0 0

1

1

0 0

0

0

11 1

0 0 0

1

1

0 0

0

1

Divisor

Divisor

1 111 0

1 111 0

1

1

0 011

0 011

000 0 0

000 0 0

0 111

0 111

1

1

000 0 0

000 0 0

1

1

1 111

1 111

1 111 0

1 111 0

0

0

1 000

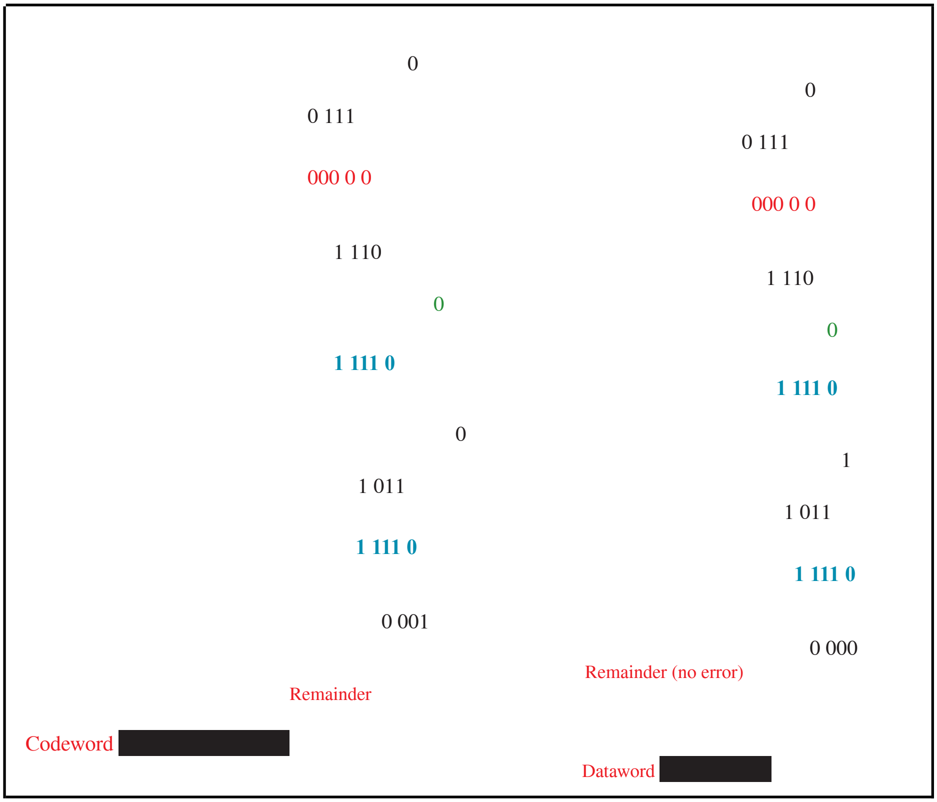
1 000

Sender

Receiver

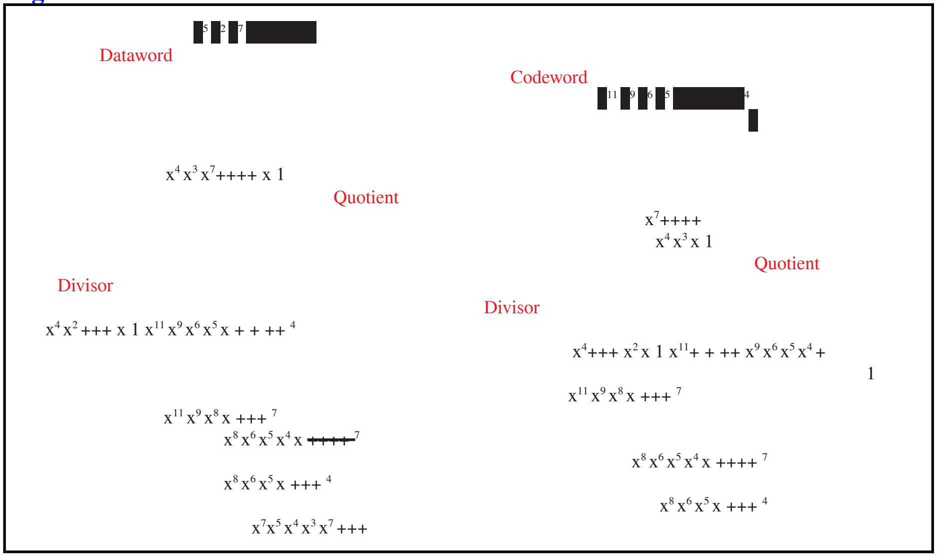
1 111 0

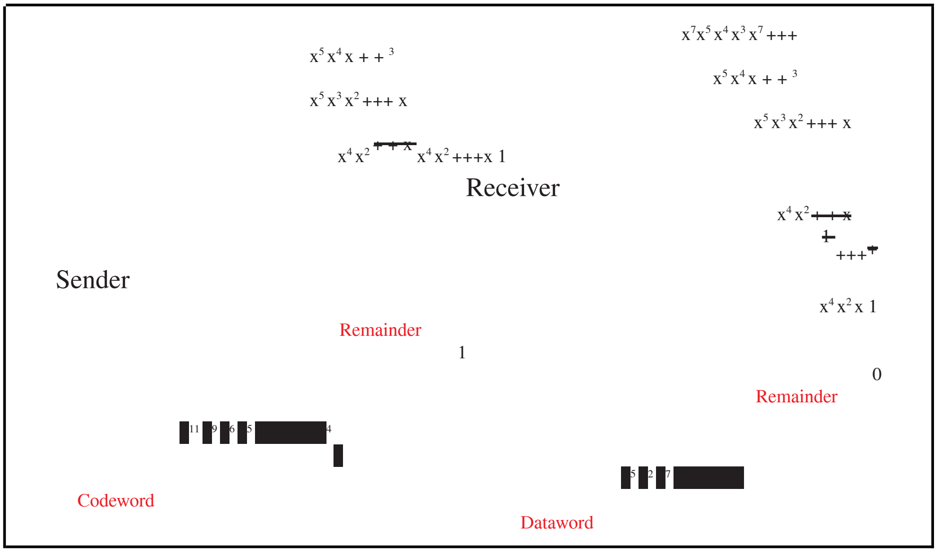
1 111 0



31. Figure 10.2 shows the generation of the codeword at the sender and the checking of the received codeword at the receiver using polynomial division.

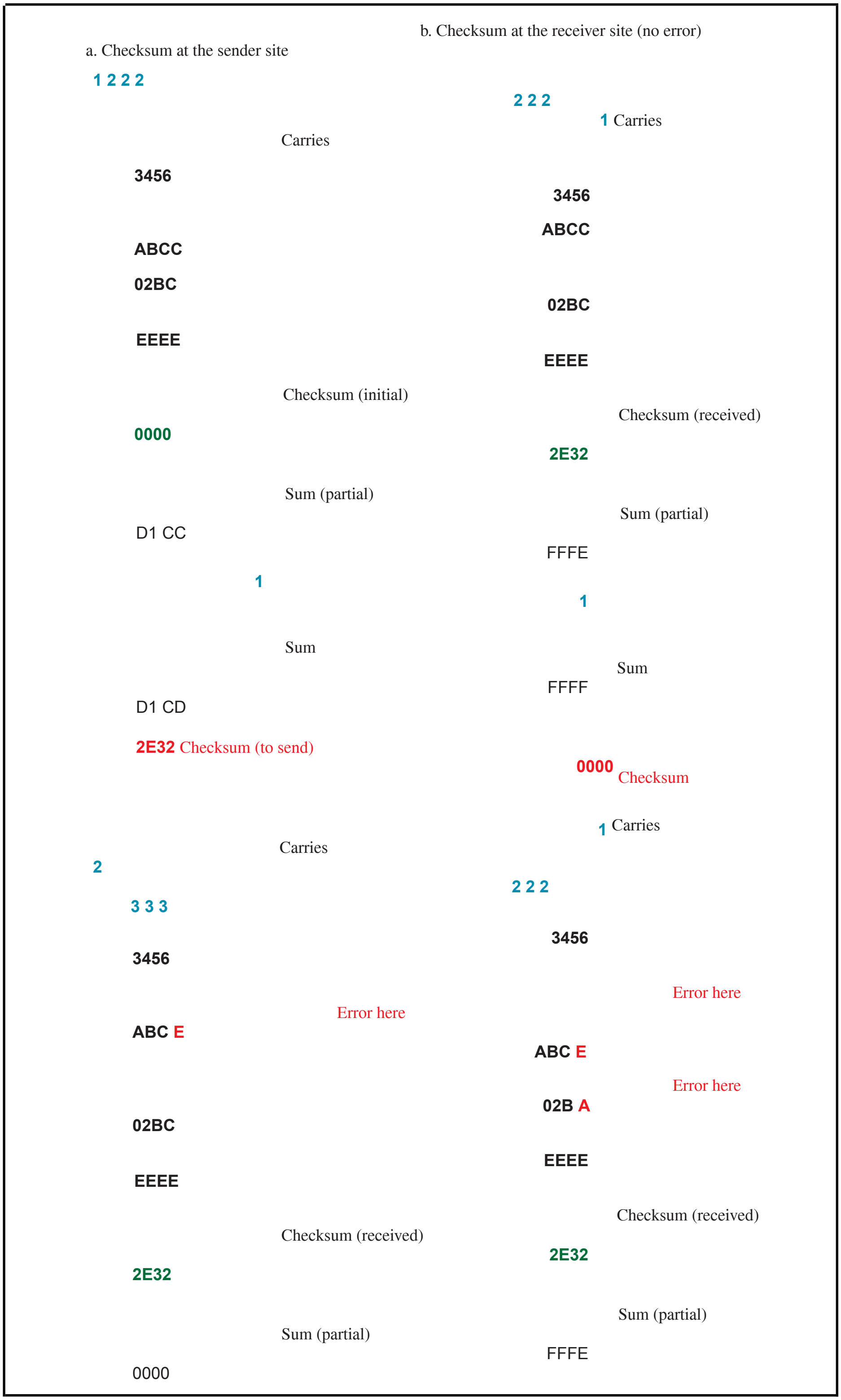
Figure 10.2 *Solution to Exercise 31*





32. Figure 10.3 shows the four situations.

Figure 10.3 *Solution to Exercise 32*



a. Checksum at the sender site

1 2 2 2 Carries

3456

ABCC

02BC

EEEE

0000 Checksum (initial)

D1 CC Sum (partial)

1

D1 CD Sum

2E32 Checksum (to send)

b. Checksum at the receiver site (no error)

2 2 2

1 Carries

3456

ABCC

02BC

EEEE

2E32 Checksum (received)

FFFE Sum (partial)

1

FFFF Sum

0000 Checksum

2 Carries

3 3 3

3456

ABC E Error here

02BC

EEEE

2E32 Checksum (received)

0000 Sum (partial)

1 Carries

2 2 2

3456

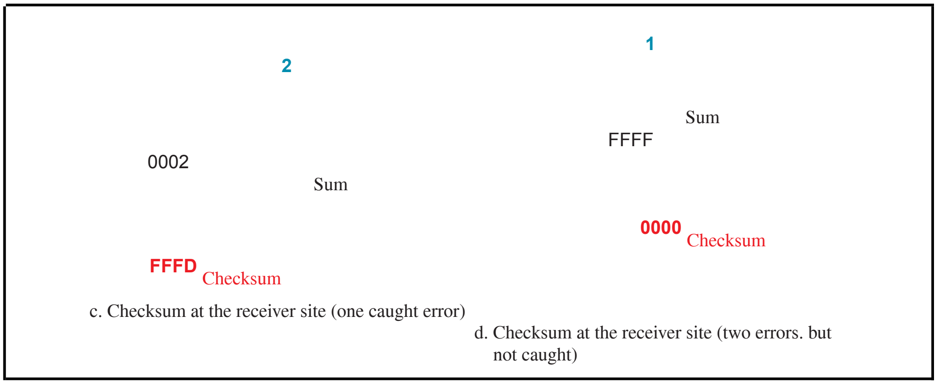
ABC E Error here

02B A Error here

EEEE

2E32 Checksum (received)

FFFE Sum (partial)


2
1
0002
Sum
FFFF
Sum
FFFD Checksum
0000 Checksum
c. Checksum at the receiver site (one caught error)
d. Checksum at the receiver site (two errors. but not caught)

a. In part a, we calculate the checksum to be sent (0x2E32)

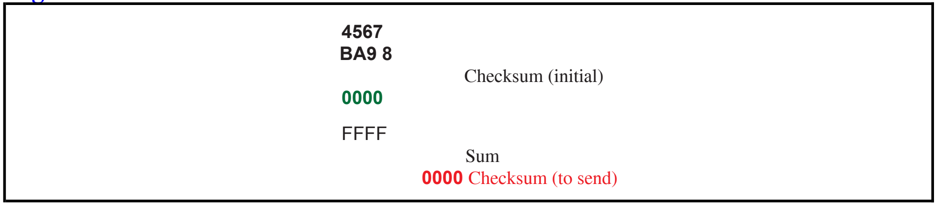
b. In part b, there is no error in transition. The receiver recalculates the checksum to be all 0x0000. The receiver correctly assumes that there is no error. c. In part c, there is one single error in transition. The receiver calculates the checksum to be 0FFFD. The receiver correctly assumes that there is some error and discards the packet.

d. In part d, there are two errors that cancel the effect of each other. The receiver calculates the checksum to be 0x0000. The receiver erroneously assumes that there is no error and accepts the packet. This is an example that shows that the checksum may slip in finding some types of errors.

8

33. Figure 10.4 shows the checksum to send (0x0000). This example shows that the checksum ***can be all 0s***. ***It can be all 1s only if all data items are all 0, which means no data at all***.

Figure 10.4 *Solution to Exercise 33*



CHAPTER 11

Data Link Control

Solutions to Review Questions and Exercises

Review Questions

1. The two main functions of the data link layer are ***data link control*** and ***media access control***. Data link control deals with the design and procedures for communication between two adjacent nodes: node-to-node communication. Media access control deals with procedures for sharing the link.
2. The data link layer needs to pack bits into ***frames***. Framing divides a message into smaller entities to make flow and error control more manageable.
3. In a ***byte-oriented protocol***, data to be carried are 8-bit characters from a coding system. Character-oriented protocols were popular when only text was exchanged by the data link layers. In a ***bit-oriented protocol***, the data section of a frame is a sequence of bits. Bit-oriented protocols are more popular today because we need to send text, graphic, audio, and video which can be better represented by a bit pattern than a sequence of characters.
4. Character-oriented protocols use ***byte-stuffing*** to be able to carry an 8-bit pattern that is the same as the flag. Byte-stuffing adds an extra character to the data section of the frame to escape the flag-like pattern. Bit-oriented protocols use ***bit-stuffing*** to be able to carry patterns similar to the flag. Bit-stuffing adds an extra bit to the data section of the frame whenever a sequence of bits is similar to the flag.
5. ***Flow control*** refers to a set of procedures used to restrict the amount of data that the sender can send before waiting for acknowledgment. ***Error control*** refers to a set of procedures used to detect and correct errors.
6. In this chapter, we discussed two protocols for noiseless channels: the ***Simplest*** and the ***Stop-and-Wait***.
7. In this chapter, we discussed three protocols for noisy channels: the ***Stop-and-Wait ARQ***, the ***Go-Back-N ARQ***, and the ***Selective-Repeat ARQ***.
8. Go-Back-N ARQ is more efficient than Stop-and-Wait ARQ. The second uses ***pipelining***, the first does not. In the first, we need to wait for an acknowledgment for each frame before sending the next one. In the second we can send several frames before receiving an acknowledgment.

9. In the ***Go-Back-N ARQ Protocol***, we can send several frames before receiving acknowledgments. If a frame is lost or damaged, all outstanding frames sent before that frame are resent. In the ***Selective- Repeat ARQ protocol*** we avoid unnecessary transmission by sending only the frames that are corrupted or missing. Both Go Back-*N* and Selective-Repeat Protocols use ***sliding windows***. In Go-Back-*N* ARQ, if m is the number of bits for the sequence number, then the size

of the send win dow must be at most $\mathbf{2^{m}-1}$; the size of the receiver window is always 1. In Selec tive-Repeat ARQ, the size of the sender and receiver window must be at most $\mathbf{2^{m-1}}$.

10. ***HDLC*** is a *bit-oriented protocol* for communication over point-to-point and multi point links. ***PPP*** is a byte-oriented protocol used for point-to-point links.

11. ***Piggybacking*** is used to improve the efficiency of bidirectional transmission. When a frame is carrying data from A to B, it can also carry control information about frames from B; when a frame is carrying data from B to A, it can also carry control information about frames from A.

12. Only ***Go-Back-N*** and ***Selective-Repeat*** protocols use ***pipelining***.

Exercises

13. We give a very simple solution. Every time we encounter an ESC or flag character, we insert an extra ESC character in the data part of the frame (see Figure 11.1).

Figure 11.1 *Solution to Exercise 13*

```
ESC ESC ESC Flag ESC ESC ESC ESC ESC ESC ESC Flag
```

14. Figure 11.2 shows data to be encapsulated in the frame.

Figure 11.2 *Solution to exercise 14*

```
00011111 0 110011111 0 0100011111 0 11111 0 1000011111 0
```

15. We write two very simple algorithms. We assume that a frame is made of a one byte beginning flag, variable-length data (possibly byte-stuffed), and a one-byte ending flag; we ignore the header and trailer. We also assume that there is no error during the transmission.

    a. Algorithm 11.1 can be used at the sender site. It inserts one ESC character whenever a flag or ESC character is encountered.

Algorithm 11.1 *Sender's site solution to Exercise 15*

```
InsertFrame (one-byte flag); // Insert beginning flag
while (more characters in data buffer)
{
```

Algorithm 11.1 *Sender's site solution to Exercise 15*

```
    ExtractBuffer (character);
    if (character is flag or ESC) InsertFrame (ESC); // Byte stuff
    InsertFrame (character);
}
InsertFrame (one-byte flag); // Insert ending flag
```

b. Algorithm 11.2 can be used at the receiver site.

Algorithm 11.2 *Receiver's site solution to Exercise 15*

```
ExtractFrame (character); // Extract beginning flag
Discard (character); // Discard beginning flag
while (more characters in the frame)
{
    ExtractFrame (character);
    if (character = = flag) exit(); // Ending flag is extracted

    if (character = = ESC)
    {
        Discard (character); // Un-stuff
        ExtractFrame (character); // Extract flag or ESC as data
    }
    InsertBuffer (character);
}
Discard (character); // Discard ending flag
```

16. We write two very simple algorithms. We assume that a frame is made of an 8-bit flag (01111110), variable-length data (possibly bit-stuffed), and an 8-bit ending flag (0111110); we ignore header and trailer. We also assume that there is no error during the transmission.

    a. Algorithm 11.3 can be used at the sender site.

Algorithm 11.3 *Sender's site solution to Exercise 16*

```
InsertFrame (8-bit flag); // Insert beginning flag
counter = 0;
while (more bits in data buffer)
{
    ExtractBuffer (bit);
    InsertFrame (bit);
    if (bit = = 1) counter = counter + 1;
    else counter = 0;

    if (counter = = 5)
    {
        InsertFrame (bit 0); // Bit stuff
        counter = 0;
    }
}
InsertFrame (8-bit flag); // Insert ending flag
```

b. Algorithm 11.4 can be used at the receiver's site. Note that when the algorithm exits from the loop, there are six bits of the ending flag in the buffer, which need to be removed after the loop.

Algorithm 11.4 *Receiver's site solution to Exercise 16*

```
ExtractFrame (8 bits); // Extract beginning flag
counter = 0;
while (more bits in frame)
{
    ExtractFrame (bit);
    if (counter = = 5)
    {
        if (bit is 0) Discard (bit); counter = 0; // Un-stuff
        if (bit is 1) exit (); // Flag is encountered
    }

    if (counter < 5)
    {
        if (bit is 0) counter = 0;
        if (bit is 1) counter = counter + 1;
        InsertBuffer (bit);
    }
}
ExtractBuffer (last 6 bits); // These bits are part of flag
Discard (6 bits);
```

17. A five-bit sequence number can create sequence numbers from 0 to 31. The sequence number in the Nth packet is (N mod 32). This means that the 101th packet has the sequence number (101 mod 32) or **5**.

18.

Stop-And-Wait ARQ send window = **1** receive window = **1**
Go-Back-N ARQ send window = $2^5 - 1$ = **31** receive window = **1**
Selective-Repeat ARQ send window = 2^4 = **16** receive window = **16**

19. See Algorithm 11.5. Note that we have assumed that both events (request and arrival) have the same priority.

Algorithm 11.5 *Algorithm for bidirectional Simplest Protocol*

```
while (true) // Repeat forever
{
```

```
WaitForEvent (); // Sleep until an event occurs
if (Event (RequestToSend)) // There is a packet to send
{
GetData ();
MakeFrame ();
SendFrame (); // Send the frame
}

if (Event (ArrivalNotification)) // Data frame arrived
{
ReceiveFrame ();
```

Algorithm 11.5 *Algorithm for bidirectional Simplest Protocol*

```
 ExtractData ();
 DeliverData (); // Deliver data to network layer
 }
} // End Repeat forever
```

20. See Algorithm 11.6. Note that in this algorithm, we assume that the arrival of a frame by a site also means the acknowledgment of the previous frame sent by the same site.

Algorithm 11.6

```
while (true) // Repeat forever
{
canSend = true;
WaitForEvent (); // Sleep until an event occurs
if (Event (RequestToSend) AND canSend) // A packet can be sent
{
GetData ();
MakeFrame ();
SendFrame (); // Send the frame
canSend = false;
}

if (Event (ArrivalNotification)) // Data frame arrived
{
ReceiveFrame ();
ExtractData ();
DeliverData (); // Deliver data to network layer
canSend = true;
}
} // End Repeat forever
```

21. Algorithm 11.7 shows one design. This is a very simple implementation in which we assume that both sites always have data to send.

Algorithm 11.7 *A bidirectional algorithm for Stop-And-Wait ARQ*

```
$S_n = 0$; // Frame 0 should be sent first
```

```
$R_n$ = 0; // Frame 0 expected to arrive first
canSend = true; // Allow the first request to go
while (true) // Repeat forever
{
 WaitForEvent (); // Sleep until an event occurs
 if (Event (RequestToSend) AND canSend) // Packet to send
 {
 GetData ();
 MakeFrame ($S_n$ , $R_n$); // The seqNo of frame is $S_n$
 StoreFrame ($S_n$ , $R_n$); //Keep copy for possible resending
 SendFrame ($S_n$ , $R_n$);
 StartTimer ();
 $S_n$ = ($S_n$ + 1) mod 2;
 canSend = false;
 }
```

Algorithm 11.7 *A bidirectional algorithm for Stop-And-Wait ARQ*

```
 if (Event (ArrivalNotification)) // Data frame arrives
 {
 ReceiveFrame ();
 if (corrupted (frame)) sleep();
 if (seqNo = = $R_n$) // Valid data frame
 {
 ExtractData ();
 DeliverData (); // Deliver data
 $R_n$ = ($R_n$ + 1) mod 2;
 }
 if (ackNo = = Sn) // Valid ACK
 {
 StopTimer ();
 PurgeFrame ($S_n$−1 , $R_n$−1); //Copy is not needed
 canSend = true;
 }
 }

 if (Event(TimeOut)) // The timer expired
 {
 StartTimer ();
 ResendFrame ($S_n$-1 , $R_n$-1); // Resend a copy
 }
} // End Repeat forever
```

22. Algorithm 11.8 shows one design. This is a very simple implementation in which we assume that both sites always have data to send.

Algorithm 11.8 *Bidirectional algorithm for Go-Back-And-N algorithm*

```
$S_w = 2^m - 1$;
$S_f$ = 0;
$S_n$ = 0;
$R_n$ = 0;
while (true) // Repeat forever
{
 WaitForEvent ();
```

```
if (Event (RequestToSend)) // There is a packet to send
{
if ($S_n - S_f >= S_w$) Sleep(); // If window is full
GetData();
MakeFrame ($S_n$ , $R_n$);
StoreFrame ($S_n$, $R_n$);
SendFrame ($S_n$, $R_n$);
$S_n = S_n + 1$;
if (timer not running) StartTimer ();
}

if (Event (ArrivalNotification))
{
Receive (Frame);
if (corrupted (ACK)) Sleep();
if ((ackNo > $S_f$) AND (ackNo <= $S_n$)) // If a valid ACK
{
```

Algorithm 11.8 *Bidirectional algorithm for Go-Back-And-N algorithm*

```
while ($S_f$ <= ackNo)
{
PurgeFrame ($S_f$);
$S_f = S_f + 1$;
}
StopTimer ();
}
if (seqNo = = $R_n$) // If expected frame
{
DeliverData (); // Deliver data
$R_n = R_n + 1$; // Slide window one slot
SendACK ($R_n$);
}
}

if (Event (TimeOut)) // The timer expires
{
StartTimer ();
Temp = $S_f$;
while (Temp < $S_n$);
{
SendFrame ($S_f$);
$S_f = S_f + 1$;
}
}
} // End Repeat forever
```

23. Algorithm 11.9 shows one design. This is a very simple implementation in which we assume that both sites always have data to send.

Algorithm 11.9 *A bidirectional algorithm for Selective-Repeat ARQ*

```
$S_w = 2^{m-1}$;
$S_f = 0$;
$S_n = 0$;
```

```
$R_n = 0$;
NakSent = false;
AckNeeded = false;
Repeat (for all slots);
Marked (slot) = false;
while (true) // Repeat forever
{
    WaitForEvent ();
    if (Event (RequestToSend)) // There is a packet to send
    {
        if ($S_n - S_f >= S_w$) Sleep (); // If window is full
        GetData ();
        MakeFrame ($S_n$ , $R_n$);
        StoreFrame ($S_n$ , $R_n$);
        SendFrame ($S_n$ , $R_n$);
        $S_n = S_n + 1$;
        StartTimer ($S_n$);
    }
```

Algorithm 11.9 *A bidirectional algorithm for Selective-Repeat ARQ*

```
if (Event (ArrivalNotification))
{
    Receive (frame); // Receive Data or NAK
    if (FrameType is NAK)
    {
        if (corrupted (frame)) Sleep();
        if (nakNo between $S_f$ and $S_n$)
        {
            resend (nakNo);
            StartTimer (nakNo);
        }
    }

    if (FrameType is Data)
    {
        if (corrupted (Frame)) AND (NOT NakSent)
        {
            SendNAK ($R_n$);
            NakSent = true;
            Sleep();
        }

        if (ackNo between $S_f$ and $S_n$)
        {
            while ($S_f$ < ackNo)
            {
                Purge ($S_f$);
                StopTimer ($S_f$);
                $S_f = S_f + 1$;
            }
        }

        if ((seqNo <> $R_n$) AND (NOT NakSent))
        {
```

```
SendNAK ($R_n$);
NakSent = true;
}

if ((seqNo in window) AND (NOT Marked (seqNo))
{
StoreFrame (seqNo);
Marked (seqNo) = true;
while (Marked ($R_n$))
{
DeliverData ($R_n$);
Purge ($R_n$);
$R_n = R_n + 1$;
AckNeeded = true;
}
}
} // End if (FrameType is Data)
} // End if (arrival event)

if (Event (TimeOut (t))) // The timer expires
```

Algorithm 11.9 *A bidirectional algorithm for Selective-Repeat ARQ*

```
{
StartTimer (t);
SendFrame (t);
}
} // End Repeat forever
```

24. State $S_n = 0$ means the sender has sent Frame 1, but is waiting for the acknowledgment. State $S_n = 1$ means the sender has sent Frame 0, but is waiting for the acknowledgment. We can then say

Event A: **Sender Site:** ACK 0 received.
Event B: **Sender Site:** ACK 1 received.

25. State $R_n = 0$ means the receiver is waiting for Frame 0. State $R_n = 1$ means the receiver is waiting for Frame 1. We can then say

Event A: **Receiver Site:** Frame 0 received.
Event B: **Receiver Site:** Frame 1 received.

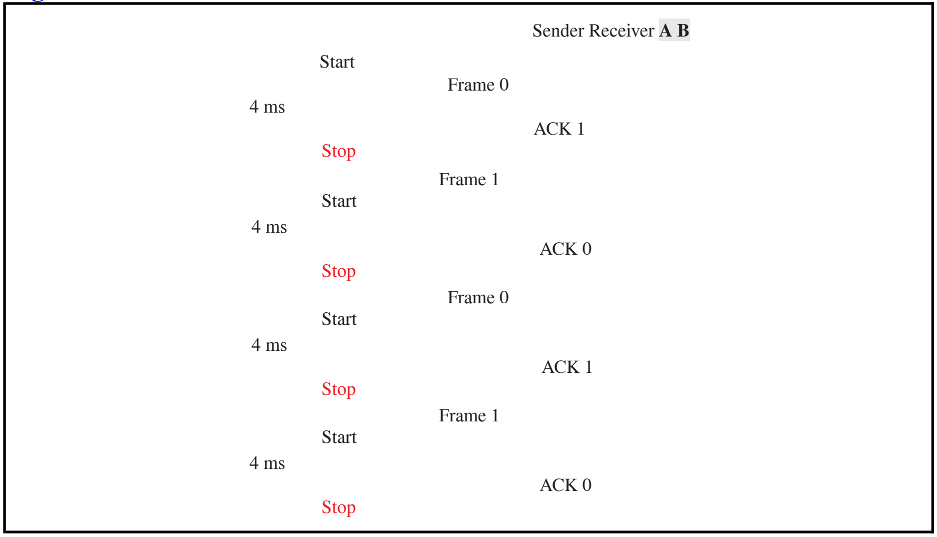
26. We can say that in this case, each state defines that a frame or an acknowledgment in transit. In other words,
(1 , 0) → Frame 0 is in transit (1 , 1) → ACK 1 is in transit (0 , 1) → Frame 1 is in transit (0 , 0) → ACK 0 is in transit

Event A: **Receiver Site:** Frame 0 arrives and ACK 1 is sent. **Event B**: **Sender Site:** ACK 1 arrives and Frame 1 is sent. **Event C**: **Receiver Site:** Frame 1 arrives and ACK 0 is sent. **Event D**:

Sender Site: ACK 0 arrives and Frame 0 is sent.

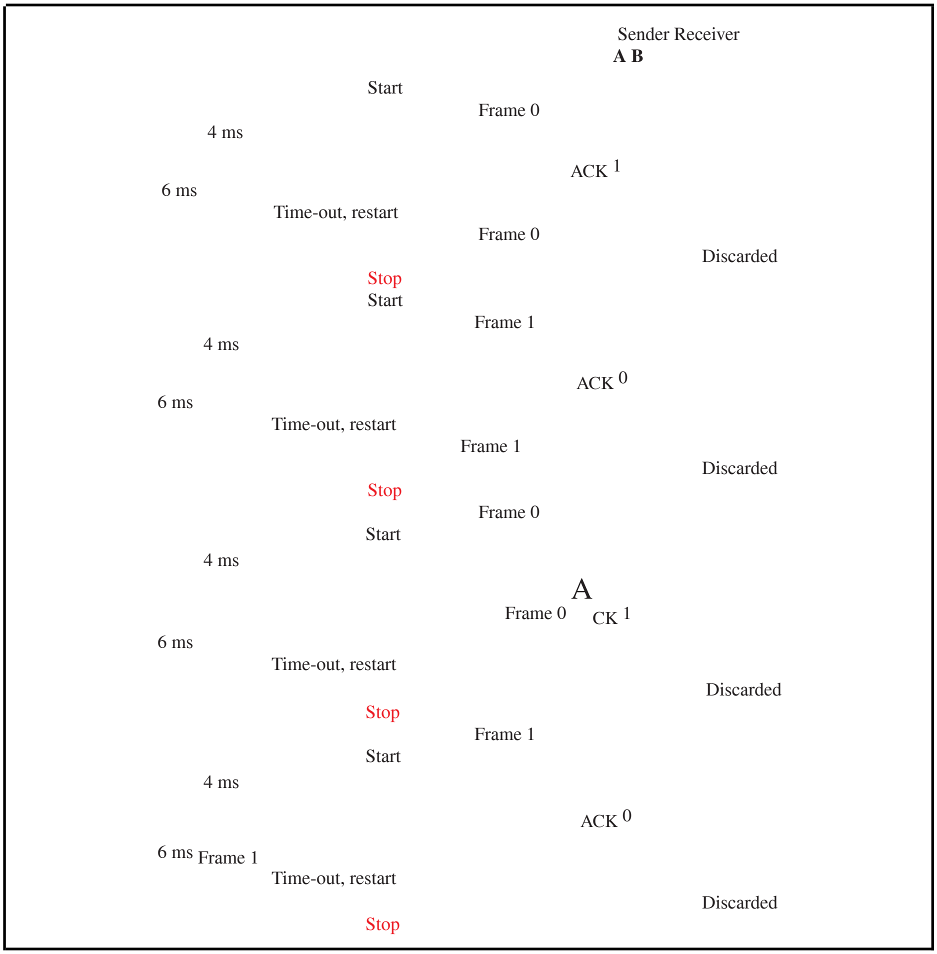
27. Figure 11.3 shows the situation. Since there are no lost or damaged frames and the round trip time is less than the time-out, each frame is sent only once.

Figure 11.3 *Solution to Exercise 27*



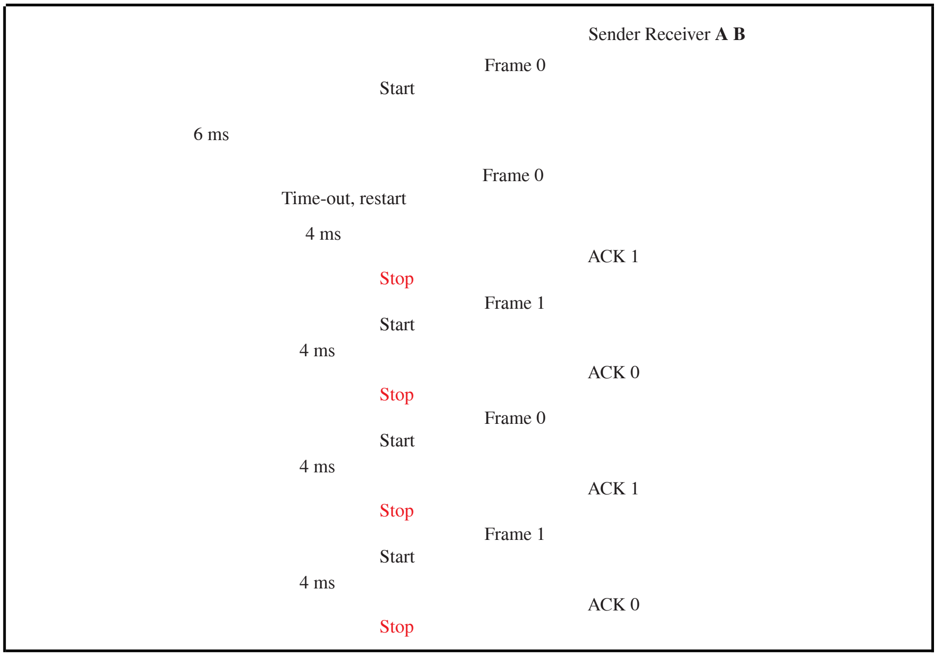
28. Figure 11.4 shows the situation. Here, we have a special situation. Although no frame is damaged or lost, the sender sends each frame twice. The reason is that the the acknowledgement for each frame reaches the sender after its timer expires. The sender thinks that the frame is lost.

Figure 11.4 *Solution to Exercise 28*



29. Figure 11.5 shows the situation. In this case, only the first frame is resent; the acknowledgment for other frames arrived on time.

Figure 11.5 *Solution to Exercise 29*



30. We need to send 1000 frames. We ignore the overhead due to the header and trailer.

    Data frame Transmission time = 1000 bits / 1,000,000 bits = 1 ms
    Data frame trip time = 5000 km / 200,000 km = 25 ms
    ACK transmission time = 0 (It is usually negligible)
    ACK trip time = 5000 km / 200,000 km = 25 ms
    Delay for 1 frame = 1 + 25 + 25 = 51 ms.
    Total delay = 1000 · 51 = 51 s

31. In the worst case, we send the a full window of size 7 and then wait for the acknowledgment of the whole window. We need to send 1000/7 ≈ 143 windows. We ignore the overhead due to the header and trailer.

    Transmission time for one window = 7000 bits / 1,000,000 bits = 7 ms
    Data frame trip time = 5000 km / 200,000 km = 25 ms
    ACK transmission time = 0 (It is usually negligible)
    ACK trip time = 5000 km / 200,000 km = 25 ms
    Delay for 1 window = 7 + 25 + 25 = 57 ms.
    Total delay = 143 · 57 ms = **8.151 s**

32. In the worst case, we send the a full window of size 4 and then wait for the acknowledgment of the whole window. We need to send 1000/4 = 250 windows.

We ignore the overhead due to the header and trailer.

Transmission time for one window = 4000 bits / 1,000,000 bits = 4 ms
Data frame trip time = 5000 km / 200,000 km = 25 ms
ACK transmission time = 0 (It is usually negligible)
ACK trip time = 5000 km / 200,000 km = 25 ms
Delay for 1 window = 4 + 25 + 25 = 54 ms.
Total delay = 250 · 54 ms = **13.5 s**

12

CHAPTER 12

Multiple Access

Solutions to Review Questions and Exercises

Review Questions

1. The three categories of multiple access protocols discussed in this chapter are ***random access***, ***controlled access***, and ***channelization***.
2. In ***random access*** methods, no station is superior to another station and none is assigned the control over another. Each station can transmit when it desires on the condition that it follows the predefined procedure. Three common protocols in this category are ***ALOHA***, ***CSMA/CD***, and ***CSMA/CA***.
3. In ***controlled access methods***, the stations consult one another to find which station has the right to send. A station cannot send unless it has been authorized by other stations. We discuss three popular controlled-access methods: ***reservation***, ***polling***, and ***token passing***.
4. ***Channelization*** is a multiple-access method in which the available bandwidth of a link is shared in time, frequency, or through code, between different stations. The common protocols in this category are ***FDMA***, ***TDMA***, and ***CDMA***.
5. In ***random access*** methods, there is no access control (as there is in controlled access methods) and there is no predefined channels (as in channelization). Each station can transmit when it desires. This liberty may create ***collision***.
6. In a ***random access*** method, there is no control; access is based on ***contention***. In a ***controlled access*** method, either a central authority (in polling) or other stations (in reservation and token passing) control the access. Random access methods have less administration overhead. On the other hand, controlled access method are collision free.
7. In a ***random access*** method, the whole available bandwidth belongs to the station that wins the contention; the other stations needs to wait. In a ***channelization*** method, the available bandwidth is divided between the stations. If a station does not have data to send, the allocated channel remains idle.

8. In a ***controlled access*** method, the whole available bandwidth belongs to the station that is granted permission either by a central authority or by other stations. In a ***channelization*** method, the available bandwidth is divided between the stations. If a station does not have data to send the allocated channel remains idle.

9. We do not need a multiple access method in this case. The local loop provides a dedicated ***point-to-point*** connection to the telephone office.
10. We do need a multiple access, because a channel in the CATV band is normally shared between several neighboring customers. The cable company uses the ***random access*** method to share the bandwidth between neighbors.

Exercises

11. To achieve the maximum efficiency in pure ALOHA, G = 1/2. If we let ***ns*** to be the number of stations and ***nfs*** to be the number of frames a station can send per second.

$$G = \boldsymbol{ns} \cdot \boldsymbol{nfs} \cdot T_{fr} = 100 \cdot \boldsymbol{nfs} \cdot 1\ \mu s = 1/2 \rightarrow \boldsymbol{nfs} = \textbf{5000 frames/s}$$

The reader may have noticed that the T_{fr} is very small in this problem. This means that either the data rate must be very high or the frames must be very small.

12. To achieve the maximum efficiency in slotted ALOHA, G = 1. If we let ***ns*** to be the number of stations and ***nfs*** to be the number of frames a station can send per second.

$$G = \boldsymbol{ns} \cdot \boldsymbol{nfs} \cdot T_{fr} = 100 \cdot \boldsymbol{nfs} \cdot 1\ \mu s = 1 \rightarrow \boldsymbol{nfs} = \textbf{10,000 frames/s}$$

The reader may have noticed that the T_{fr} is very small in this problem. This means that either the data rate must be very high or the frames must be very small.

13. We can first calculate T_{fr} and G, and then the throughput.

$$T_{fr} = (1000 \text{ bits}) / 1 \text{ Mbps} = 1 \text{ ms}$$

$$G = \boldsymbol{ns} \cdot \boldsymbol{nfs} \cdot T_{fr} = 100 \cdot 10 \cdot 1 \text{ ms} = 1$$

$$\text{For } \textbf{pure ALOHA} \rightarrow S = G \cdot e^{-2G} \approx \textbf{13.53 percent}$$

This means that each station can successfully send only 1.35 frames per second.

14. We can first calculate T_{fr} and G, and then the throughput.

$$T_{fr} = (1000 \text{ bits}) / 1 \text{ Mbps} = 1 \text{ ms}$$

$$G = \boldsymbol{ns} \cdot \boldsymbol{nfs} \cdot T_{fr} = 100 \cdot 10 \cdot 1 \text{ ms} = 1$$

$$\text{For } \textbf{slotted ALOHA} \rightarrow S = G \cdot e^{-G} \approx \textbf{36.7 percent}$$

This means that each station can successfully send only 3.67 frames per second.

15. Let us find the relationship between the minimum frame size and the data rate. We know that

T_{fr} = (frame size) / (data rate) = 2 · T_p = 2 · distance / (propagation speed)
or
(frame size) = [2 · (distance) / (propagation speed)] · (data rate)]
or
(frame size) = K · (data rate)

This means that minimum frame size is proportional to the data rate (K is a constant). When the data rate is increased, the frame size must be increased in a network with a fixed length to continue the proper operation of the CSMA/CD. In Example 12.5, we mentioned that the minimum frame size for a data rate of 10 Mbps is 512 bits. We calculate the minimum frame size based on the above proportionality relationship

Data rate = 10 Mbps → minimum frame size = **512 bits** Data rate = 100 Mbps → minimum frame size = **5120 bits** Data rate = 1 Gbps → minimum frame size = **51,200 bits** Data rate = 10 Gbps → minimum frame size = **512,000 bits**

16. Let us find the relationship between the collision domain (maximum length of the network) and the data rate. We know that

T_{fr} = (frame size) / (data rate) = 2 · T_p = 2 · distance / (propagation speed) or
distance = [(frame size) (propagation speed)] / [2 · (data rate)]
or
distance = K / (data rate)

This means that distance is inversely proportional to the data rate (K is a constant). When the data rate is increased, the distance or maximum length of network or collision domain is decreased proportionally. In Example 12.5, we mentioned that the maximum distance for a data rate of 10 Mbps is 2500 meters. We calculate the maximum distance based on the above proportionality relationship.

Data rate = 10 Mbps → maximum distance = **2500 m** Data rate = 100 Mbps → maximum distance = **250 m**
Data rate = 1 Gbps → maximum distance = **25 m**
Data rate = 10 Gbps → maximum distance = **2.5 m**

This means that when the data rate is very high, it is almost impossible to have a network using CSMA/CD.

17. We have $t_1 = 0$ and $t_2 = 3$ μs
a. $t_3 - t_1$= (2000 m) / (2 · 10^8 m/s) =10 μs → t_3 = 10 μs + t_1 = **10 μs** b. $t_4 - t_2$ = (2000 m) / (2 · 10^8 m/s) =10 μs → t_4 = 10 μs + t_2 = **13 μs** c. $T_{fr(A)} = t_4 - t_1$ = 13 − 0 = 13 μs → Bits_A = 10 Mbps · 13 μs = **130 bits** d. $T_{fr(C)} = t_3 - t_2$ = 10 − 3 = 07μs → Bits_C = 10 Mbps · 07 μs = **70 bits** 18. We have $t_1 = 0$ and $t_2 = 3$ μs
a. $t_3 - t_1$= (2000 m) / (2 · 10^8 m/s) =10 μs → t_3 = 10 μs + t_1 = **10 μs** b. $t_4 - t_2$ = (2000 m) / (2 · 10^8 m/s) =10 μs → t_4 = 10 μs + t_2 = **13 μs** c. $T_{fr(A)} = t_4 - t_1$ = 13 − 0 = 13 μs → Bits_A = 100 Mbps · 13 μs = **1300 bits** d. $T_{fr(C)} = t_3 - t_2$ = 10 − 3 =

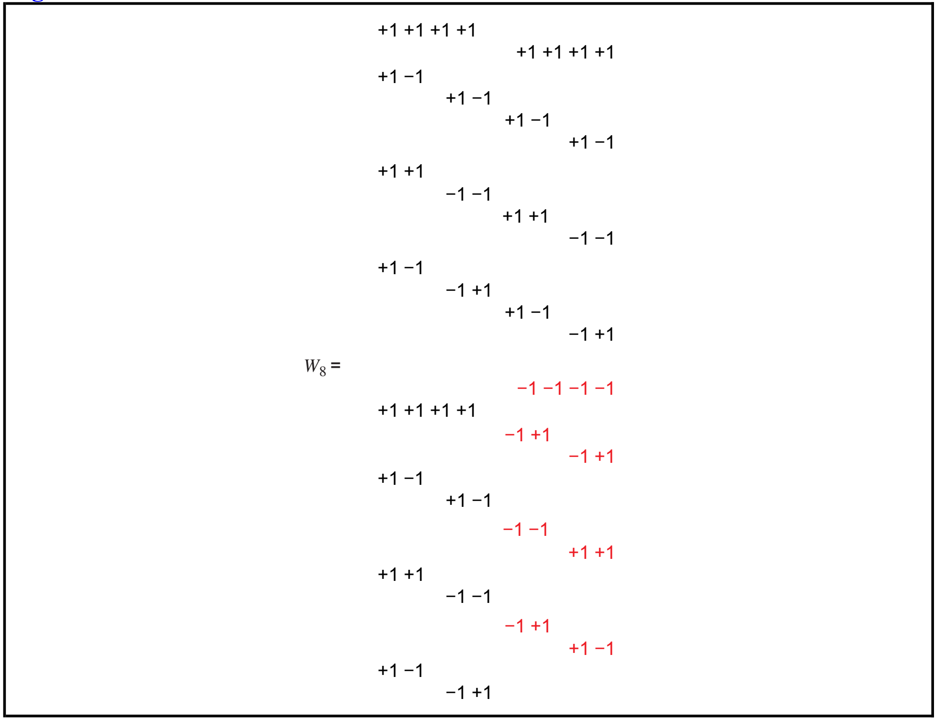
07 μs → Bits_C = 100 Mbps · 07 μs = **700 bits**

4

Note that in this case, both stations have already sent more bits than the minimum number of bits required for detection of collision. The reason is that with the 100 Mbps, the minimum number of bits requirement is feasible only when the maxi mum distance between stations is less than or equal to 250 meters as we will see in Chapter 13.

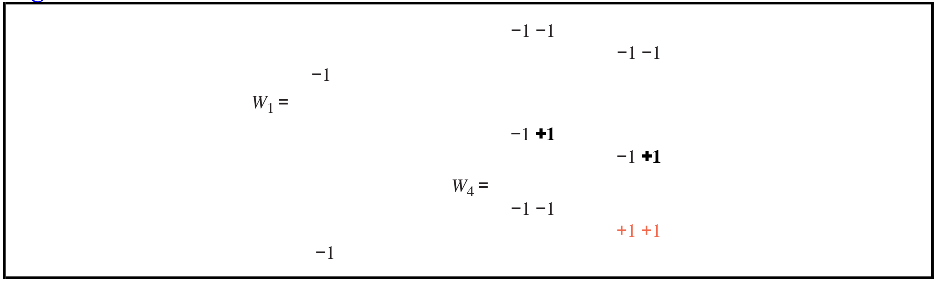
19. See Figure 12.1.

Figure 12.1 *Solution to Exercise 19*



20. See Figure 12.2.

Figure 12.2 *Solution to Exercise 20*



−1

$W_2 =$

−1 **+1**

+1 −1

−1 **+1**

21.

Third Property: we calculate the inner product of each row with itself:

Row 1 • Row 1 [+1 +1 +1 +1] • [+1 +1 +1 +1] = +1 + 1 + 1 + 1 = **4**
Row 2 • Row 2 [+1 −1 +1 −1] • [+1 −1 +1 −1] = +1 + 1 + 1 + 1 = **4**
Row 3 • Row 1 [+1 +1 −1 −1] • [+1 +1 −1 −1] = +1 + 1 + 1 + 1 = **4**
Row 4 • Row 4 [+1 −1 −1 +1] • [+1 −1 −1 +1] = +1 + 1 + 1 + 1 = **4**

Fourth Property: we need to prove 6 relations:

Row 1 • Row 2 [+1 +1 +1 +1] • [+1 −1 +1 −1] = +1 − 1 + 1 − 1 = **0**
Row 1 • Row 3 [+1 +1 +1 +1] • [+1 +1 −1 −1] = +1 + 1 − 1 − 1 = **0**
Row 1 • Row 4 [+1 +1 +1 +1] • [+1 −1 −1 +1] = +1 − 1 − 1 + 1 = **0**
Row 2 • Row 3 [+1 −1 +1 −1] • [+1 +1 −1 −1] = +1 − 1 − 1 + 1 = **0**
Row 2 • Row 4 [+1 −1 +1 −1] • [+1 −1 −1 +1] = +1 + 1 − 1 − 1 = **0**
Row 3 • Row 4 [+1 +1 −1 −1] • [+1 −1 −1 +1] = +1 − 1 + 1 − 1 = **0**

5

22.

Third Property: we calculate the inner product of each row with itself:

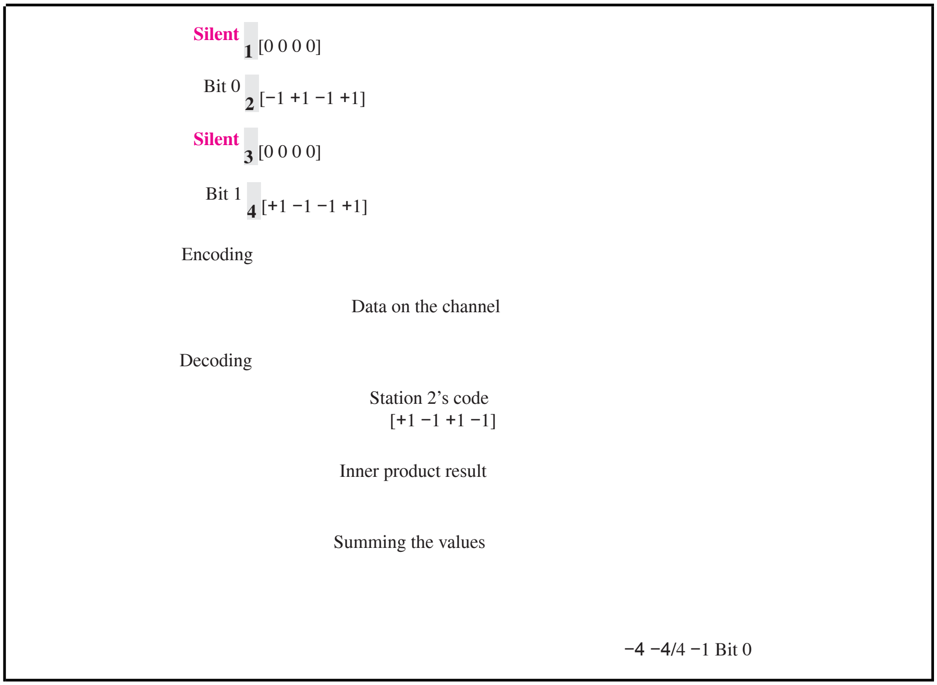
Row 1 • Row 1 [−1 −1 −1 −1] • [−1 −1 −1 −1] = +1 + 1 + 1 + 1 = **4** Row 2 • Row 2 [−1 +1 −1 +1] • [−1 +1 −1 +1] = +1 + 1 + 1 + 1 = **4** Row 3 • Row 1 [−1 −1 +1 +1] • [−1 −1 +1 +1] = +1 + 1 + 1 + 1 = **4** Row 4 • Row 4 [−1 +1 +1 −1] • [−1 +1 +1 −1] = +1 + 1 + 1 + 1 = **4**

Fourth Property: we neede to prove 6 relations:

Row 1 • Row 2 [−1 −1 −1 −1] • [−1 +1 −1 +1] = +1 − 1 + 1 − 1 = **0** Row 1 • Row 3 [−1 −1 −1 −1] • [−1 −1 +1 +1] = +1 + 1 − 1 − 1 = **0** Row 1 • Row 4 [−1 −1 −1 −1] • [−1 +1 +1 −1] = +1 − 1 − 1 + 1 = **0** Row 2 • Row 3 [−1 +1 −1 +1] • [−1 −1 +1 +1] = +1 − 1 − 1 + 1 = **0** Row 2 • Row 4 [−1 +1 −1 +1] • [−1 +1 +1 −1] = +1 + 1 − 1 − 1 = **0** Row 3 • Row 4 [−1 −1 +1 +1] • [−1 +1 +1 −1] = +1 − 1 + 1 − 1 = **0**

23. Figure 12.3 shows the encoding, the data on the channel, and the decoding.

Figure 12.3 *Solution to Exercise 23*



6

24. We can say:

Polling and Data Transfer

Station 1: [poll + 5 · (frame + ACK)]
Station 2: [poll + 5 · (frame + ACK)]
Station 3: [poll + 5 · (frame + ACK)]
Station 4: [poll + 5 · (frame + ACK)]

Polling and Sending NAKs

Station 1: [poll + NAK]
Station 2: [poll + NAK]
Station 3: [poll + NAK]
Station 4: [poll + NAK]

Total Activity:

8 polls + 20 frames + 20 ACKs + 4 NAKs = **21024 bytes**

We have 1024 bytes of overhead.

25. We can say:

Polling and Data Transfer

Frame 1 for all four stations: 4 · [poll + frame + ACK)]
Frame 2 for all four stations: 4 · [poll + frame + ACK)]
Frame 3 for all four stations: 4 · [poll + frame + ACK)]
Frame 4 for all four stations: 4 · [poll + frame + ACK)]

Frame 5 for all four stations: 4 · [poll + frame + ACK)]

Polling and Sending NAKs

Station 1: [poll + NAK]
Station 2: [poll + NAK]
Station 3: [poll + NAK]
Station 4: [poll + NAK]

Total Activity:

24 polls + 20 frames + 20 ACKs + 4 NAKs = **21536 bytes**

We have 1536 bytes of overhead which is 512 bytes more than the case in Exercise 23. The reason is that we need to send 16 extra polls.

CHAPTER 13

Local Area Networks: Ethernet

Solutions to Review Questions and Exercises

Review Questions

1. The ***preamble*** is a 56-bit field that provides an alert and timing pulse. It is added to the frame at the physical layer and is not formally part of the frame. SFD is a one byte field that serves as a flag.

2. An ***NIC*** provides an Ethernet station with a 6-byte physical address. Most of the physical and data-link layer duties are done by the NIC.

3. A ***multicast address*** identifies a group of stations; a ***broadcast address*** identifies all stations on the network. A ***unicast*** address identifies one of the addresses in a group.

4. A ***bridge*** can raise the bandwidth and separate collision domains.

5. A ***layer-2 switch*** is an N-port bridge with additional sophistication that allows faster handling of packets.

6. In a ***full-duplex Ethernet***, each station can send data without the need to sense the line.

7. The rates are as follows:

Standard Ethernet: **10 Mbps**
Fast Ethernet: **100 Mbps**
Gigabit Ethernet: Ten-Gigabit Ethernet:
1 Gbps 10 Gbps

8. The common traditional Ethernet implementations are ***10Base5***, ***10Base2***, ***10- Base-T***, and ***10Base-F***.

9. The common Fast Ethernet implementations are ***100Base-TX***, ***100Base-FX***, and ***100Base-T4***.

10. The common Gigabit Ethernet implementations are ***1000Base-SX***, 1000Base-LX, ***1000Base-CX***, and ***1000Base-T***.

11. The common Ten-Gigabit Ethernet implementations are ***10GBase-S***, ***10GBase-L***, and

10GBase-E.

1
2

Exercises

12. We interpret each four-bit pattern as a hexadecimal digit. We then group the hexa decimal digits with a colon between the pairs:

5A:11:55:18:AA:0F

13. The bytes are sent from left to right. However, the bits in each byte are sent from the least significant (rightmost) to the most significant (leftmost). We have shown the bits with spaces between bytes for readability, but we should remember that that bits are sent without gaps. The arrow shows the direction of movement.

← **01011000 11010100 00111100 11010010 01111010 11110110**

14. The first byte in binary is 0000011**1**. The least significant bit is 1. This means that the pattern defines a **multicast address.**

15. The first byte in binary is 0100001**1**. The least significant bit is 1. This means that the pattern defines a multicast address. ***A multicast address can be a destination address, but not a source address.*** Therefore, the receiver knows that there is an error, and discards the packet.

16. The minimum data size in the Standard Ethernet is 46 bytes. Therefore, we need to add **4 bytes of padding** to the data (46 − 42 = 4)

17. The maximum data size in the Standard Ethernet is 1500 bytes. The data of 1510 bytes, therefore, must be split between two frames. The standard dictates that the first frame must carry the maximum possible number of bytes (1500); the second frame then needs to carry only 10 bytes of data (it requires padding). The follow ing shows the breakdown:
Data size for the first frame: **1500 bytes**
Data size for the second frame: **46 bytes** (with padding)

18. The smallest Ethernet frame is 64 bytes and carries 46 bytes of data (and possible padding). The largest Ethernet frame is 1518 bytes and carries 1500 bytes of data. The ratio is (data size) / (frame size) in percent. We can then answer the question as follows:

Smallest Frame Frame size = 64 Data size ≤ 46 **Ratio ≤ 71.9%**
Largest Frame Frame size =1518 Data size = 1500 **Ratio = 98.8%**

19. We can calculate the propagation time as t = (2500 m) / (200,000.000) = **12.5** μ**s**. To get the total delay, we need to add propagation delay in the equipment (10 μs). This results in **T = 22.5 μs.**

20. The smallest frame is 64 bytes or 512 bits. With a data rate of 10 Mbps, we have T_{fr} = (512 bits) / (10 Mbps) = **51.2 μs**

This means that the time required to send the smallest frame is the same at the maximum time required to detect the collision.

CHAPTER 14

Wireless LANs

Solutions to Review Questions and Exercises

Review Questions

1. The ***basic service set (BSS)*** is the building block of a wireless LAN. A BSS with out an AP is called an ad hoc architecture; a BSS with an AP is sometimes referred to as an infrastructure network. An ***extended service set (ESS)*** is made up of two or more BSSs with APs. In this case, the BSSs are connected through a distribution system, which is usually a wired LAN.

2. A station with ***no-transition*** mobility is either stationary or moving only inside a BSS. A station with ***BSS-transition*** mobility can move from one BSS to anther, but the movement is confined inside one ESS. A station with ***ESS-transition*** mobility can move from one ESS to another.

3. The ***orthogonal frequency-division multiplexing (OFDM)*** method for signal gen eration in a 5-GHz ISM band is similar to ***frequency division multiplexing (FDM)***, with one major difference: All the subbands are used by one source at a given time. Sources contend with one another at the data link layer for access.

4. Stations on wireless LANs normally use ***CSMA/CA***.

5. ***Network Allocation Vector (NAV)*** forces other stations to defer sending their data if one station acquires access. In other words, it provides the collision avoidance aspect. When a station sends an RTS frame, it includes the duration of time that it needs to occupy the channel. The stations that are affected by this transmission create a timer called a NAV.

6. A Bluetooth network is called a ***piconet***. A ***scatternet*** is two or more piconets. 7. The following shows the relationship:

Radio layer → **Internet physical layer**
Baseband layer → **MAC sublayer of Internet data link layer**
L2CAP layer → **LLC sublayer of Internet data link layer**

8. A Bluetooth primary and secondary can be connected by a ***synchronous connec tion-oriented (SCO)*** link or an ***asynchronous connectionless (ACL)*** link. An SCO link is used when avoiding latency (delay in data delivery) is more important than

integrity (error-free delivery). An ACL link is used when data integrity is more important than avoiding latency.

9. The primary sends on the ***even-numbered*** slots; the secondary sends on the ***odd numbered*** slots.

10. In all types of frames, a duration of **259 μs** is used for hopping.

Exercises

11. In ***CSMA/CD***, the protocol allows collisions to happen. If there is a collision, it will be detected, destroyed, and the frame will be resent. ***CSMA/CA*** uses a tech nique that prevents collision.

12. See Table 14.1.

Table 14.1 *Exercise 12*

Fields	*802.3* *field size (bytes)*	*802.11* *field size (bytes)*
Destination Address	6	
Source Address	6	
Address 1		6
Address 2		6
Address 3		6
Address 4		6
FC		2
D/ID		2
SC		2
PDU Length	2	
Data and Padding	46 to 1500	
Frame Body	64-1518	0 to 2312
FCS (CRC)	4	4

CHAPTER 15

Connecting LANs, Backbone Networks, and Virtual Networks

Solutions to Review Questions and Exercises

Review Questions

1. An ***amplifier*** amplifies the signal, as well as noise that may come with the signal, whereas a ***repeater*** regenerates the signal, bit for bit, at the original strength.

2. ***Bridges*** have access to station ***physical addresses*** and can forward a packet to the appropriate segment of the network. In this way, they ***filter*** traffic and help control congestion.

3. A ***transparent bridge*** is a bridge in which the stations are completely unaware of the bridge's existence. If a bridge is added or deleted from the system, reconfigura tion of the stations is unnecessary.

4. A signal can only travel so far before it becomes corrupted. A ***repeater*** regenerates the original signal; the signal can continue to travel and the LAN length is thus extended.

5. A ***hub*** is a ***multiport repeater***.

6. A ***forwarding port*** forwards a frame that it receives; a ***blocking port*** does not. 7. In a ***bus backbone***, the topology of the backbone is a ***bus***; in a ***star backbone***, the topology is a ***star***.

8. A ***VLAN*** saves time and money because reconfiguration is done through software. Physical reconfiguration is not necessary.

9. Members of a ***VLAN*** can send broadcast messages with the assurance that users in other groups will not receive these messages.

10. A ***VLAN*** creates virtual workgroups. Each workgroup member can send broadcast messages to others in the workgroup. This eliminates the need for multicasting and all the overhead messages associated with it.

11. Stations can be grouped by ***port number***, ***MAC address***, ***IP address***, or by a com bination of these characteristics.

1
2

Exercises

12. Table 15.1 shows one possibility. We have sorted the table based on the physical address to make the searching faster.

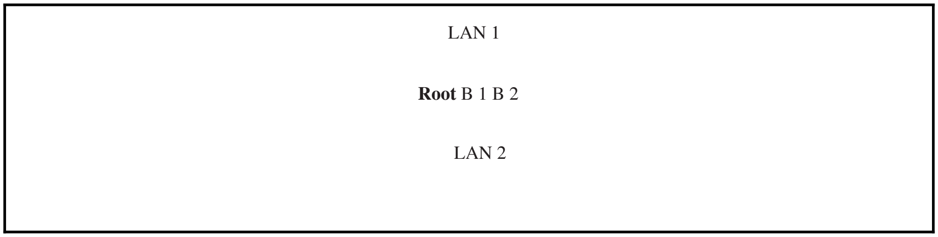
Table 15.1 *Solution to Exercise 12*

Address	*Port*
A	1

B	1
C	2
D	2
E	3
F	3

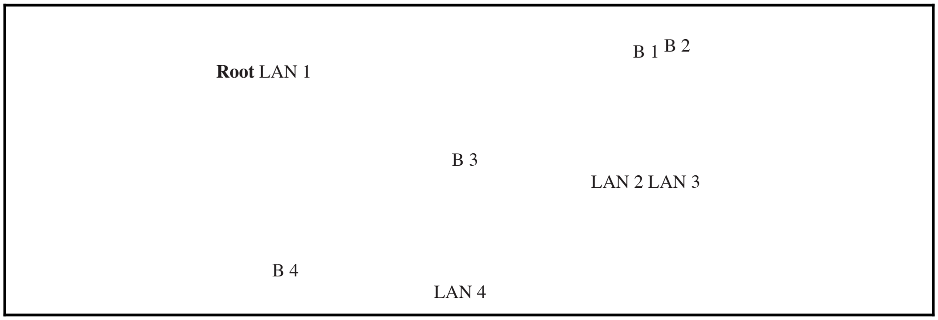
13. Figure 15.1 shows one possible solution. We made bridge B1 the root.

Figure 15.1 *Solution to Exercise 13*



14. Figure 15.2 shows one possible solution.

Figure 15.2 *Solution to Exercise 14*


B 1 B 2
Root LAN 1
B 3
LAN 2 LAN 3
B 4
LAN 4