CHAPTER 1

Introduction

Solutions to Odd-Numbered Review Questions and Exercises

Review Questions

1. The five components of a data communication system are the ***sender***, **receiver**, ***transmission medium***, ***message***, and ***protocol***.
3. The three criteria are ***performance***, ***reliability***, and ***security***.
5. Line configurations (or types of connections) are ***point-to-point*** and ***multipoint***.
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.
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
11. An ***internet*** is an interconnection of networks. The ***Internet*** is the name of a specific worldwide network
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

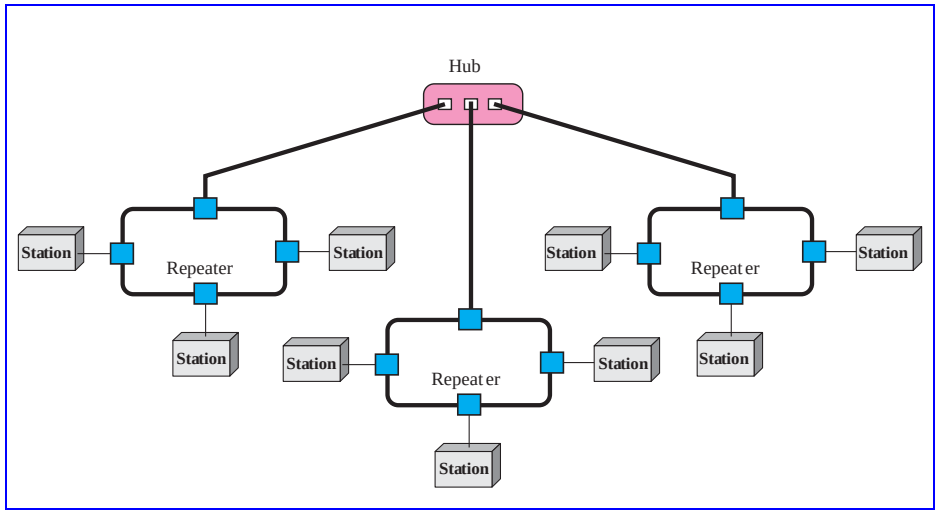
15. With **16** bits, we can represent up to $\mathbf{2^{16}}$ different colors.
17.
    a. ***Mesh topology***: If one connection fails, the other connections will still be working.
    b. **Star topology**: The other devices will still be able to send data through the hub; there will be no access to 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.

19. Theoretically, in a ***ring topology***, unplugging one station, interrupts the ring. However, most ring networks use a mechanism that bypasses the station; the ring can continue its operation.

21. See Figure 1.1

Figure 1.1 *Solution to Exercise 21*



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 sensitive 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.

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 Odd-Numbered Review Questions and Exercises

Review Questions

1. The Internet model, as discussed in this chapter, include ***physical***, ***data link***, ***network***, ***transport***, and ***application*** layers.
3. The ***application*** layer supports the user.
5. ***Peer-to-peer processes*** are processes on two or more devices communicating at a same layer
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.
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***
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***
13. The *application layer services* include ***file transfer***, ***remote access***, ***shared database management***, and ***mail services***.

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 communications is the ***Open Systems Interconnection*** (**OSI**) model.

17.

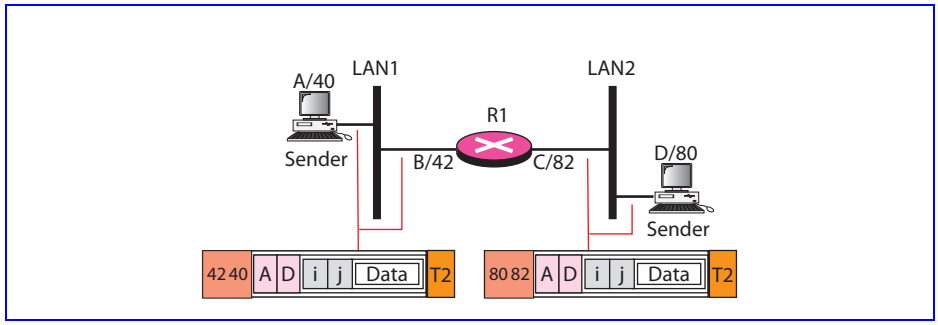
a. Reliable process-to-process delivery: **transport** layer
b. Route selection: ***network*** layer
c. Defining frames: ***data link*** layer
d. Providing user services: ***application*** layer
e. Transmission of bits across the medium: ***physical*** layer

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

21. See Figure 2.1.

Figure 2.1 *Solution to Exercise 21*



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.

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 Odd-Numbered Review Questions and Exercises

Review Questions

1. ***Frequency*** and ***period*** are the inverse of each other. T = 1/ f and f = 1/T.
3. Using Fourier analysis. ***Fourier series*** gives the frequency domain of a periodic signal; ***Fourier analysis*** gives the frequency domain of a nonperiodic signal.
5. ***Baseband transmission*** means sending a digital or an analog signal without modulation using a low-pass channel. ***Broadband transmission*** means modulating a digital or an analog signal using a band-pass channel.
7. The ***Nyquist theorem*** defines the maximum bit rate of a noiseless 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 ($\lambda = v/f$), where v is the propagation speed in the media.
11. The frequency domain of a voice signal is normally ***continuous*** because voice is a ***nonperiodic*** signal.
13. This is ***baseband transmission*** because no modulation is involved.
15. This is ***broadband transmission*** because it involves modulation.

Exercises

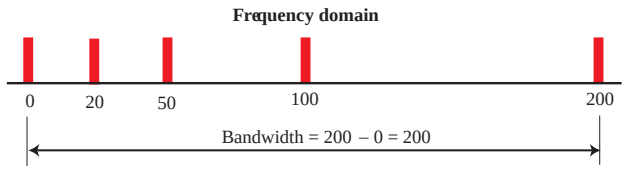
17.

    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**

19. See Figure 3.1
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.
23.

    a. (10 / 1000) s = **0.01 s**
    b. (8 / 1000) s = 0. 008 s = **8 ms**

Figure 3.1 *Solution to Exercise 19*

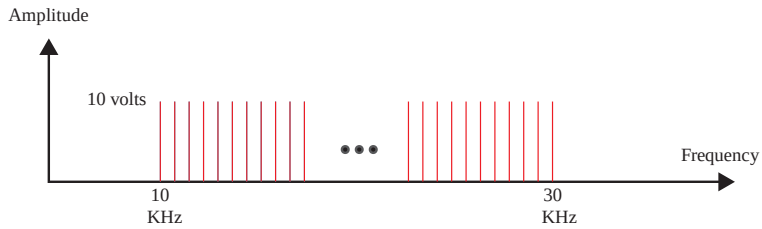


c. ((100,000 × 8) / 1000) s = **800 s**

25. The signal makes 8 cycles in 4 ms. The frequency is 8 /(4 ms) = **2 KHz**

27. The signal is periodic, so the frequency domain is made of discrete frequencies. as shown in Figure 3.2.

Figure 3.2 *Solution to Exercise 27*



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**

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**

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

35. 1 μm × 1000 = 1000 μm = **1 mm**

37. We have

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

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

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

41. We have

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

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

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

We then have

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

43.

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

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

45. We have

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

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**

4

CHAPTER 4

Digital Transmission

Solutions to Odd-Numbered Review Questions and Exercises

Review Questions

1. The three different techniques described in this chapter are ***line coding***, ***block coding***, and ***scrambling***.
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.
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.
7. In this chapter, we introduced ***unipolar***, ***polar***, ***bipolar***, ***multilevel***, and ***multitransition*** coding.
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.
11. In ***parallel transmission*** we send data *several* bits at a time. In ***serial transmission*** we send data *one* bit at a time.

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**
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.
17. See Figure 4.2. 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.

Figure 4.1 *Solution to Exercise 15*

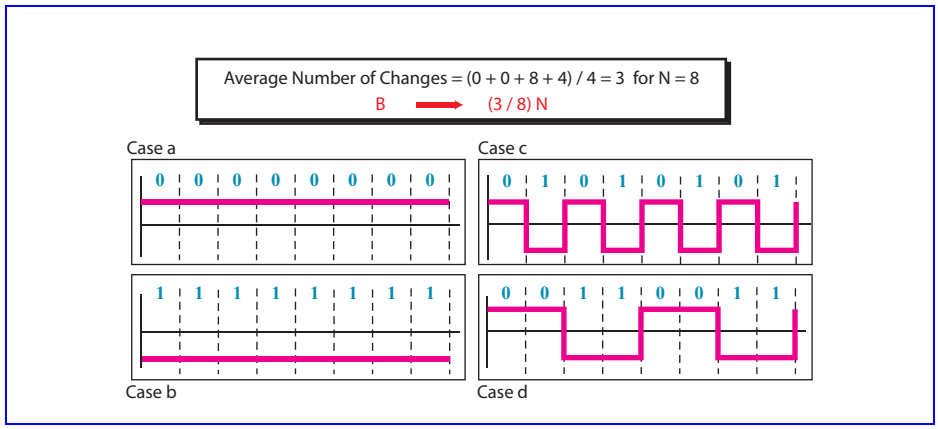
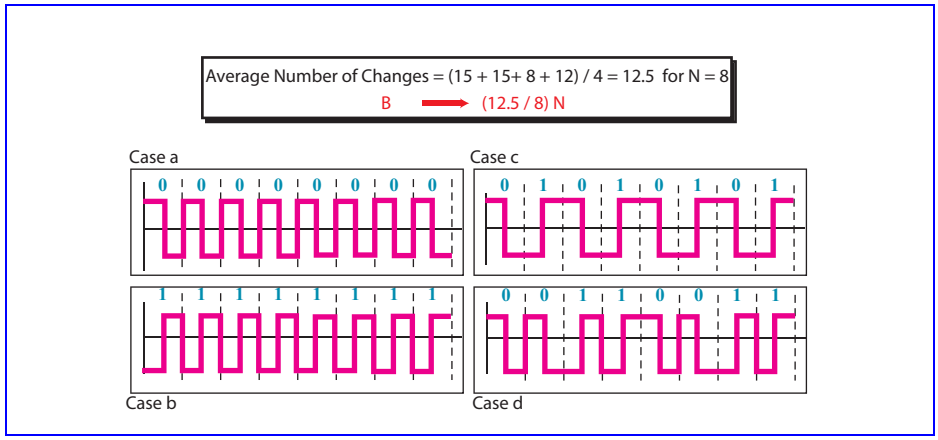


Figure 4.2 *Solution to Exercise 17*

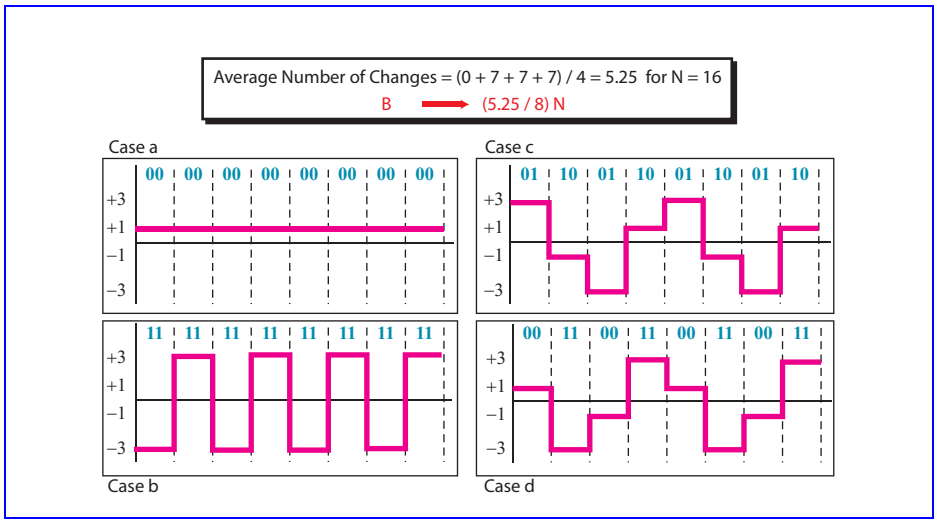


19. See Figure 4.3. B is proportional to **(5.25 / 16) N** which is inside range in Table 4.1 (B = 0 to N/2) for 2B/1Q.

21. The data stream can be found as
    a. NRZ-I: **10011001**.
    b. Differential Manchester: **11000100**.
    c. AMI: **01110001**.

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**

Figure 4.3 *Solution to Exercise 19*



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**.

27

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

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

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

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

29. The maximum data rate can be calculated as

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

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

a. NRZ → N = 2 × B = 2 × 1 MHz = **2 Mbps**
b. Manchester → N = 1 × B = 1 × 1 MHz = **1 Mbps**
c. MLT-3 → N = 3 × B = 3 × 1 MHz = **3 Mbps**
d. 2B1Q → N = 4 × B = 4 × 1 MHz = **4 Mbps**

4

CHAPTER 5

Analog Transmission

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

Exercises

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

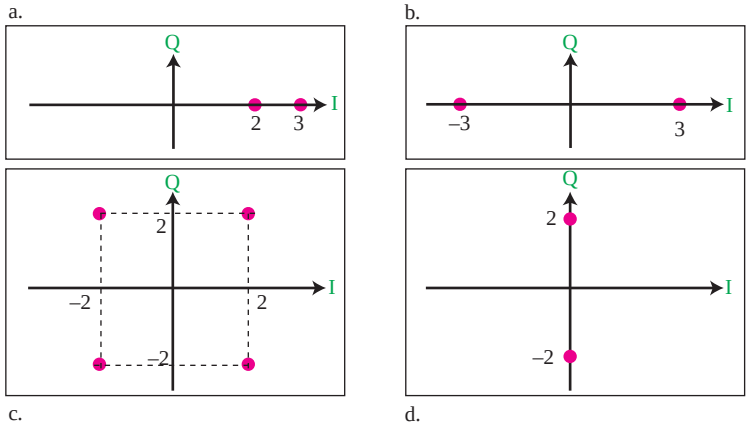
a. $r = \log_2 2$	$= 1$	$\rightarrow$	$S = (1/1) \times (2000 \text{ bps})$	$=$ **2000 baud**
b. $r = \log_2 2$	$= 1$	$\rightarrow$	$S = (1/1) \times (4000 \text{ bps})$	$=$ **4000 baud**
c. $r = \log_2 4$	$= 2$	$\rightarrow$	$S = (1/2) \times (6000 \text{ bps})$	$=$ **3000 baud**
d. $r = \log_2 64$	$= 6$	$\rightarrow$	$S = (1/6) \times (36{,}000 \text{ bps})$	$=$ **6000 baud**

13. We use the formula **$r = \log_2 L$** 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. $\log_2 128 = 7$

15. See Figure 5.1

Figure 5.1 *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.

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

a. r = 1 → B= (1 + 1) × (1/1) × (4000 bps) = **8000 Hz**
b. r = 1 → B = (1 + 1) × (1/1) × (4000 bps) + 4 KHz = **8000 Hz**
c. r = 2 → B = (1 + 1) × (1/2) × (4000 bps) = **2000 Hz**
d. r = 4 → B = (1 + 1) × (1/4) × (4000 bps) = **1000 Hz**

19.

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

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

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.

21.

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

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

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

4

CHAPTER 6

Bandwidth Utilization:

Solutions to Odd-Numbered 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.
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.
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).
7. ***WDM*** is common for multiplexing ***optical signals*** because it allows the multiplexing of signals with a very high frequency.
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.
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.

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**
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**.

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%**. Efficiency is better than the one in Exercise 16.

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**.

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.**

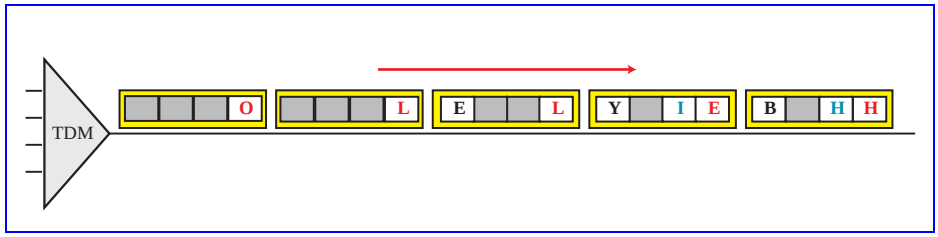
b. Each frame carries 1 bit from each 190-kbps source. Frame rate = **190,000 frames/s.**

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.

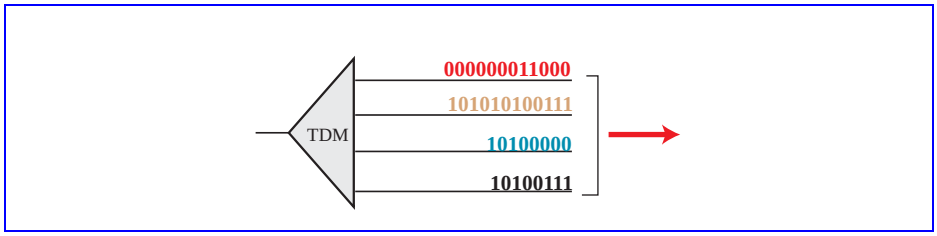
23. See Figure 6.1.

Figure 6.1 *Solution to Exercise 23*



25. See Figure 6.2.

Figure 6.2 *Solution to Exercise 25*



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

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

$$N_1 = \mathbf{11}$$
$$N_2 = (5 + 7 \times \mathbf{11}) \bmod 17 - 1 = \mathbf{13}$$
$$N_3 = (5 + 7 \times \mathbf{13}) \bmod 17 - 1 = \mathbf{10}$$
$$N_4 = (5 + 7 \times \mathbf{10}) \bmod 17 - 1 = \mathbf{6}$$
$$N_5 = (5 + 7 \times \mathbf{6}) \bmod 17 - 1 = \mathbf{12}$$
$$N_6 = (5 + 7 \times \mathbf{12}) \bmod 17 - 1 = \mathbf{3}$$
$$N_7 = (5 + 7 \times \mathbf{3}) \bmod 17 - 1 = \mathbf{8}$$
$$N_8 = (5 + 7 \times \mathbf{8}) \bmod 17 - 1 = \mathbf{9}$$

4

CHAPTER 7

Transmission Media

Solutions to Odd-Numbered Review Questions and Exercises

Review Questions

1. The ***transmission media*** is located ***beneath the physical layer*** and controlled by the physical layer.
3. ***Guided media*** have physical boundaries, while ***unguided media*** are unbounded.
5. ***Twisting*** ensures that both wires are equally, but ***inversely***, affected by external influences such as noise.
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.
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.

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**

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

1 KHz	dB = –3	$P_2 = P_1 \times 10^{-3/10}$	= **100.23 mw**
10 KHz	dB = –5	$P_2 = P_1 \times 10^{-5/10}$	= **63.25 mw**

100 KHz dB = –7 $P_2 = P_1 \times 10^{-7/10}$ = **39.90 mw**

The table shows that the power for 100 KHz is reduced almost 5 times, which may not be acceptable for some applications.

15. We first make Table 7.2 from Figure 7.9 (in the textbook).

Table 7.2 *Solution to Exercise 15*

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**

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

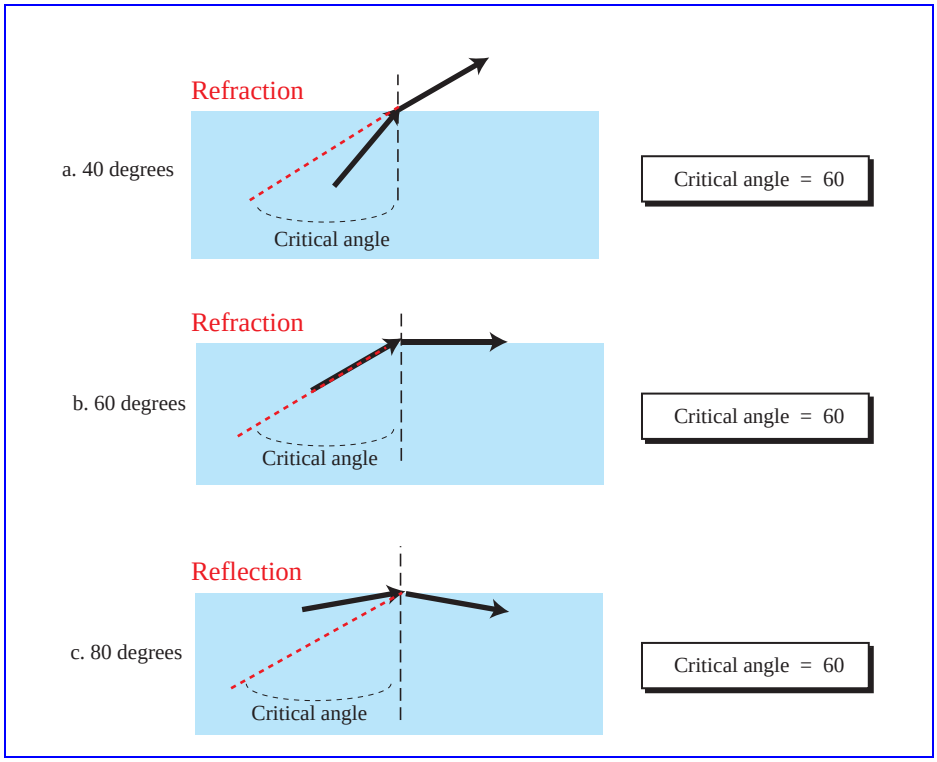
17. We can use the formula $f = c / \lambda$ to find the corresponding frequency for each wave length as shown below (c is the speed of propagation):
    a. $B = [(2 \times 10^8)/1000 \times 10^{-9}] - [(2 \times 10^8)/ 1200 \times 10^{-9}]$ = **33 THz**
    b. $B = [(2 \times 10^8)/1000 \times 10^{-9}] - [(2 \times 10^8)/ 1400 \times 10^{-9}]$ = **57 THz**
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**

21. See Figure 7.1.
    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.

Figure 7.1 *Solution to Exercise 21*



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.

4

CHAPTER 8

Switching

Solutions to Odd-Numbered 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.
3. There are two approaches to packet switching: ***datagram approach*** and ***virtual-circuit approach***.
5. The address field defines the ***end-to-end*** (source to destination) 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.
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.

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

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

    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.

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 transfer phase, each packet needs to carry a virtual-circuit identifier to show which virtual-circuit that particular packet follows.

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 teardown 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 considered a ***connectionless*** network.

17.

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

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.

21.

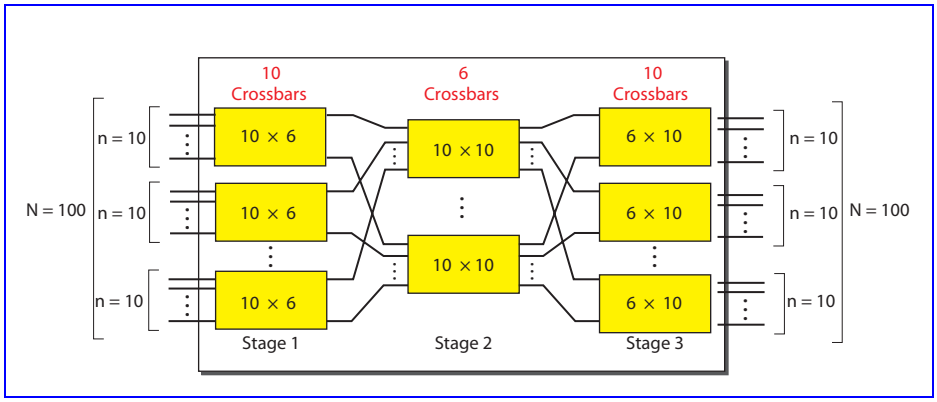
a. If $n > k$, an $n \times k$ crossbar is like a ***multiplexer*** that combines n inputs into k outputs. However, we need to know that a regular multiplexer discussed in Chapter 6 is $n \times 1$.

b. If $n < k$, an $n \times k$ 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 $1 \times n$.

23.

a. See Figure 8.1.

Figure 8.1 *Solution to Exercise 23 Part a*



b. The total number of crosspoints are

$$\text{Number of crosspoints} = 10\,(10 \times 6) + 6\,(10 \times 10) + 10\,(6 \times 10) = \mathbf{1800}$$

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.***

25.

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

b. Total crosspoints $\geq 4N[(2N)^{1/2} - 1] \geq$ ***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***.

4

CHAPTER 9

Using Telephone and Cable Networks

Solutions to Odd-Numbered Review Questions and Exercises

Review Questions

1. The telephone network is made of three major components: ***local loops***, ***trunks***, and ***switching 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.

5. Telephone companies provide two types of services: ***analog*** and ***digital***.

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 subscriber line access multiplexer (DSLAM)*** at the telephone company site.

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.

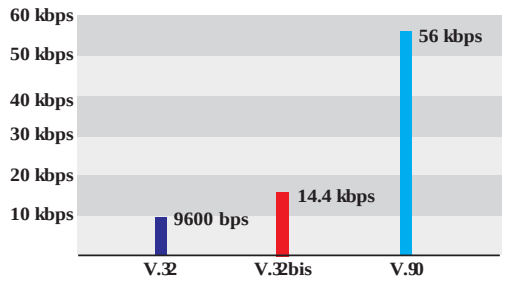
Exercises

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.

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.

15. See Figure 9.1.

Figure 9.1 *Solution to Exercise 15*



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**

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

$$\text{Time} = (1{,}000{,}000 \times 8) / 10{,}000{,}000 \approx 0.8 \text{ seconds}$$

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 bandwidth. This means if all neighbors try to transfer data, the effective data rate will be decreased.

CHAPTER 10

Error Detection and Correction

Solutions to Odd-numbered 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).
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).
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.
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.
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 (intentionally 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.

Exercises

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

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

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.

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.

17.

a. **01**
b. **error**
c. **00**
d. **error**

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)

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

a. Dataword: 0100 → Codeword: 0100011 → Corrupted: **1**100011 → $s_2s_1s_0 = 110$
Change b_3 (Table 10.5) → Corrected codeword: **0100011** → dataword: 0100
The dataword is correctly found.

b. Dataword: 0111 → Codeword: 0111001 → Corrupted: **00**11001 → $s_2s_1s_0 = 011$
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.

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 $\mathbf{d_{min} = 3}$.

25.
    a. 101110 → $x^5 + x^3 + x^2 + x$
    b. 101110 → 101110**000** (Three 0s are added to the right)
    c. $x^3 \times (x^5 + x^3 + x^2 + x) = x^8 + x^6 + x^5 + x^4$
    d. 101110 → 10 (The four rightmost bits are deleted)
    e. $x^{-4} \times (x^5 + x^3 + x^2 + x) = x$ (Note that negative powers are deleted)

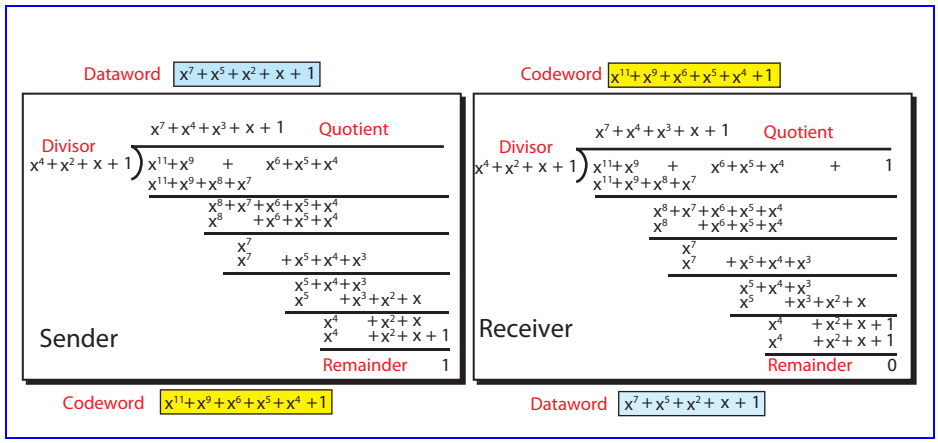
27. CRC-8 generator is $x^8 + x^2 + x + 1$.
    a. It has more than one term and the coefficient of x^0 is 1. It can detect a single-bit error.
    b. The polynomial is of degree 8, which means that the number of checkbits (remainder) r = 8. It will detect all burst errors of size 8 or less.
    c. Burst errors of size 9 are detected most of the time, but they slip by with probability $(1/2)^{r-1}$ 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 probability $(1/2)^r$ or $(1/2)^8 \approx 0.004$. This means **4 out of 1000** burst errors of size 15 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

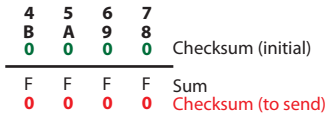
31. Figure 10.1 shows the generation of the codeword at the sender and the checking of the received codeword at the receiver using polynomial division.

Figure 10.1 *Solution to Exercise 31*



33. Figure 10.2 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 10.2 *Solution to Exercise 33*



CHAPTER 11

Data Link Control

Solutions to Odd-Numbered 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.

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.

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.

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***.

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

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.

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*

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)
{
    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
```

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**.

19. See Algorithm 11.3. Note that we have assumed that both events (request and arrival) have the same priority.

Algorithm 11.3 *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 ();
        ExtractData ();
        DeliverData ();     // Deliver data to network layer
    }
}  // End Repeat forever
```

21. Algorithm 11.4 shows one design. This is a very simple implementation in which we assume that both sites always have data to send.

Algorithm 11.4 *A bidirectional algorithm for Stop-And-Wait ARQ*

```
Sn = 0;    // Frame 0 should be sent first
Rn = 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 (Sn , Rn);   // The seqNo of frame is Sn
        StoreFrame (Sn , Rn);    //Keep copy for possible resending
        SendFrame (Sn , Rn);
        StartTimer ();
        Sn = (Sn + 1) mod 2;
        canSend = false;
    }

    if (Event (ArrivalNotification))    // Data frame arrives
    {
        ReceiveFrame ();
        if (corrupted (frame))    sleep();
        if (seqNo = = Rn)     // Valid data frame
        {
            ExtractData ();
            DeliverData ();     // Deliver data
            Rn = (Rn + 1) mod 2;
        }
        if (ackNo = = Sn)     // Valid ACK
```

Algorithm 11.4 *A bidirectional algorithm for Stop-And-Wait ARQ*

```
        {
            StopTimer ();
            PurgeFrame (Sn–1 , Rn–1);    //Copy is not needed
            canSend = true;
        }
    }

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

23. Algorithm 11.5 shows one design. This is a very simple implementation in which we assume that both sites always have data to send.

Algorithm 11.5 *A bidirectional algorithm for Selective-Repeat ARQ*

```
Sw = 2m–1;
Sf = 0;
Sn = 0;
Rn = 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 (Sn–Sf >= Sw)    Sleep ();    // If window is full
      GetData ();
      MakeFrame (Sn , Rn);
      StoreFrame (Sn , Rn);
      SendFrame (Sn , Rn);
      Sn = Sn + 1;
      StartTimer (Sn);
   }

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

Algorithm 11.5 *A bidirectional algorithm for Selective-Repeat ARQ*

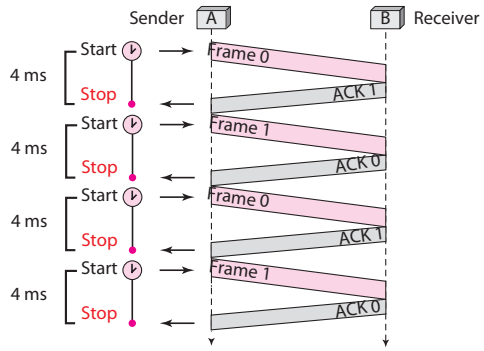
```
        if (FrameType is Data)
        {
            if (corrupted (Frame)) AND (NOT NakSent)
            {
                SendNAK (Rn);
                NakSent = true;
                Sleep();
            }

            if (ackNo between Sf and Sn)
            {
                while (Sf < ackNo)
                {
                    Purge (Sf);
                    StopTimer (Sf);
                    Sf = Sf + 1;
                }
            }

            if ((seqNo <> Rn) AND (NOT NakSent))
            {
                SendNAK (Rn);
                NakSent = true;
            }

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

    if (Event (TimeOut (t)))    // The timer expires
    {
        StartTimer (t);
        SendFrame (t);
    }
}  // End Repeat forever
```

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.

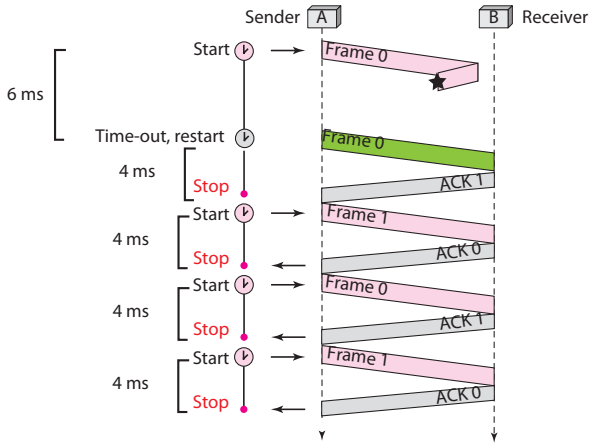
27. Figure 11.2 shows the situation. Since there are no lost or damaged frames and the round trip time is less than the time-out, each frame is sent only once.

Figure 11.2 *Solution to Exercise 27*



29. Figure 11.3 shows the situation. In this case, only the first frame is resent; the acknowledgment for other frames arrived on time.

Figure 11.3 *Solution to Exercise 29*



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**

CHAPTER 12

Multiple Access

Solutions to Odd-Numbered 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***.

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***.

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***.

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.

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.

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.

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

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.

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

$$T_{fr} = (1000 \text{ bits}) / 1 \text{ Mbps} = 1 \text{ ms}$$
$$G = \textbf{\textit{ns}} \times \textbf{\textit{nfs}} \times T_{fr} = 100 \times 10 \times 1 \text{ ms} = 1$$
$$\text{For } \textbf{pure ALOHA} \rightarrow \quad S = G \times e^{-2G} \approx \textbf{13.53 percent}$$

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

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

$$T_{fr} = (\text{frame size}) / (\text{data rate}) = 2 \times T_p = 2 \times \text{distance} / (\text{propagation speed})$$

or

$$(\text{frame size}) = [2 \times (\text{distance}) / (\text{propagation speed})] \times (\text{data rate})$$

or

$$\textbf{(frame size) = K} \times \textbf{(data rate)}$$

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

17. We have $t_1 = 0$ and $t_2 = 3\ \mu s$

    a. $t_3 - t_1 = (2000\ m) / (2 \times 10^8\ m/s) = 10\ \mu s \rightarrow t_3 = 10\ \mu s + t_1 = \mathbf{10\ \mu s}$

    b. $t_4 - t_2 = (2000\ m) / (2 \times 10^8\ m/s) = 10\ \mu s \rightarrow t_4 = 10\ \mu s + t_2 = \mathbf{13\ \mu s}$

    c. $T_{fr(A)} = t_4 - t_1 = 13 - 0 = 13\ \mu s \rightarrow Bits_A = 10\ Mbps \times 13\ \mu s = \mathbf{130\ bits}$

    d. $T_{fr(C)} = t_3 - t_2 = 10 - 3 = 07 \mu s \rightarrow Bits_C = 10\ Mbps \times 07\ \mu s = \mathbf{70\ bits}$

19. See Figure 12.1.

Figure 12.1 *Solution to Exercise 19*

$$W_8 = \begin{bmatrix} +1 & +1 & +1 & +1 & +1 & +1 & +1 & +1 \\ +1 & -1 & +1 & -1 & +1 & -1 & +1 & -1 \\ +1 & +1 & -1 & -1 & +1 & +1 & -1 & -1 \\ +1 & -1 & -1 & +1 & +1 & -1 & -1 & +1 \\ +1 & +1 & +1 & +1 & -1 & -1 & -1 & -1 \\ +1 & -1 & +1 & -1 & -1 & +1 & -1 & +1 \\ +1 & +1 & -1 & -1 & -1 & -1 & +1 & +1 \\ +1 & -1 & -1 & +1 & -1 & +1 & +1 & -1 \end{bmatrix}$$

21.

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

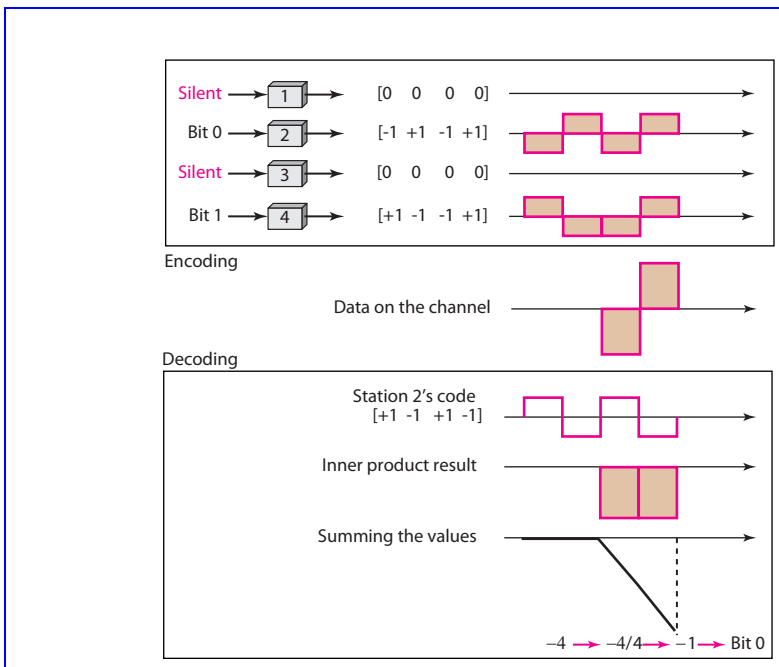
Row 1 • Row 1	[+1 +1 +1 +1]	•	[+1 +1 +1 +1]	= +1 + 1 + 1 + 1 = **4**
Row 2 • Row 2	[+1 −1 +1 −1]	•	[+1 −1 +1 −1]	= +1 + 1 + 1 + 1 = **4**
Row 3 • Row 1	[+1 +1 −1 −1]	•	[+1 +1 −1 −1]	= +1 + 1 + 1 + 1 = **4**
Row 4 • Row 4	[+1 −1 −1 +1]	•	[+1 −1 −1 +1]	= +1 + 1 + 1 + 1 = **4**

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**

23. Figure 12.2 shows the encoding, the data on the channel, and the decoding.

Figure 12.2 *Solution to Exercise 23*



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 Odd-Numbered 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.
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.
5. A ***layer-2 switch*** is an N-port bridge with additional sophistication that allows faster handling of packets.
7. The rates are as follows:

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

9. The common Fast Ethernet implementations are ***100Base-TX***, ***100Base-FX***, and ***100Base-T4***.
11. The common Ten-Gigabit Ethernet implementations are ***10GBase-S***, ***10GBase-L***, and ***10GBase-E***.

Exercises

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**

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.

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 following shows the breakdown:

    Data size for the first frame: **1500 bytes**

    Data size for the second frame: **46 bytes** (with padding)

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

CHAPTER 14

Wireless LANs

Solutions to Odd-Numbered Review Questions and Exercises

Review Questions

1. The ***basic service set (BSS)*** is the building block of a wireless LAN. A BSS without 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.

3. The ***orthogonal frequency-division multiplexing (OFDM)*** method for signal generation 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.

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.

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**

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

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 technique that prevents collision.

2

CHAPTER 15

Connecting LANs, Backbone Networks, and Virtual Networks

Solutions to Odd-Numbered Review Questions and Exercises

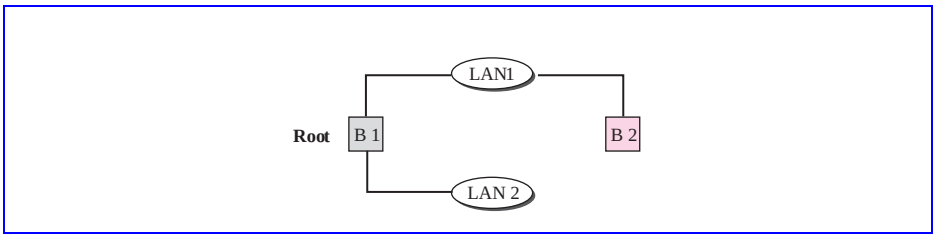
Review Questions

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.
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, reconfiguration of the stations is unnecessary.
5. A ***hub*** is a ***multiport repeater***.
7. In a ***bus backbone***, the topology of the backbone is a ***bus***; in a ***star backbone***, the topology is a ***star***.
9. Members of a ***VLAN*** can send broadcast messages with the assurance that users in other groups will not receive these messages.
11. Stations can be grouped by ***port number***, ***MAC address***, ***IP address***, or by a combination of these characteristics.

Exercises

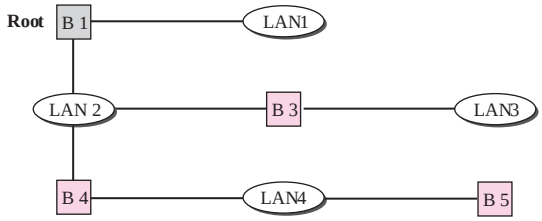
13. Figure 15.1 shows one possible solution. We made bridge B1 the root.

Figure 15.1 *Solution to Exercise 13*



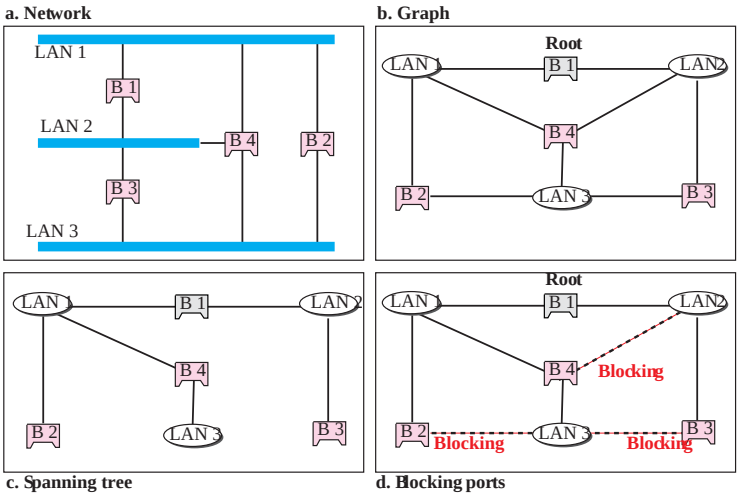
15. Figure 15.2 shows one possible solution.

Figure 15.2 *Solution to Exercise 15*



17. Although any ***router*** is also a ***bridge***, replacing bridges with routers has the following consequences:
    a. Routers are more expensive than bridges.
    b. Routers operate at the first three-layers; bridges operates at the first two layers. Routers are not designed to provide direct filtering the way the bridges do. A router needs to search a routing table which is normally longer and more time consuming than a filtering table.
    c. A router needs to decapsulate and encapsulate the frame and change physical addresses in the frame because the physical addresses in the arriving frame define the previous node and the current router; they must be changed to the physical addresses of the current router and the next hop. A bridge does not change the physical addresses. Changing addresses, and other fields, in the frame means much unnecessary overhead.
19. Figure 15.3 shows one possible solution. We have shown the network, the graph, the spanning tree, and the blocking ports.

Figure 15.3 *Solution to Exercise 19*



21. A ***bridge*** has more overhead than a ***repeater***. A ***bridge*** processes the packet at ***two layers***; a ***repeater*** processes a frame at ***only one layer***. A bridge needs to search a table and find the forwarding port as well as to regenerate the signal; a repeater only regenerates the signal. In other words, a bridge is also a repeater (and more); a repeater is not a bridge.

CHAPTER 16

Cellular Telephone and Satellite Networks

Solutions to Odd-Numbered Review Questions and Exercises

Review Questions

1. A ***mobile switching center*** coordinates communications between a ***base station*** and a ***telephone central office***.
3. A ***high reuse factor*** is better because the cells that use the same set of frequencies are farther apart (separated by more cells).
5. ***AMPS*** is an analog cellular phone system using FDMA.
7. ***GSM*** is a European standard that provides a common second-generation technology for all of Europe.
9. The three orbit types are ***equatorial***, ***inclined***, and ***polar***.
11. A ***footprint*** is the area on earth at which the satellite aims its signal.
13. Transmission from the earth to the satellite is called the ***uplink***. Transmission from the satellite to the earth is called the ***downlink***.
15. The main difference between ***Iridium*** and ***Globalstar*** is the relaying mechanism. Iridium requires relaying between satellites. Globalstar requires relaying between satellites and earth stations.

16.1 EXERCISES

17. ***In AMPS***, there are two separate bands for each direction in communication. In each band, we have 416 analog channels. Out of this number, 21 channels are reserved for control. With a reuse factor of 7, the maximum number of simultaneous calls in each cell is

$$\text{Maximum number of simultaneous calls} = (416 - 21) / 7 = 56.4 \approx 56$$

19. In ***GSM***, separate bands are assigned for each direction in communication. This means 124 analog channels are available in each cell (assuming no control channels). Each analog channel carries 1 multiframe. Each multiframe carries 26 frames (2 frames are for control). Each frame allows 8 calls. With a reuse factor of 3, we have

Maximum number of simultaneous calls = [(124) × 24 × 8] / 3 = **7936**

21. In Exercise 17, we showed that the maximum simultaneous calls per cell for ***APMS*** is 56. Using the total bandwidth of 50 MHz (for both directions), we have

Efficiency = 56 / 50 = **1.12 calls/MHz**

23. In Exercise 19, we showed that the maximum simultaneous calls per cell for ***GSM*** is 7936. Using the total bandwidth of 50 MHz (for both directions), we have

Efficiency = 7936 / 50 = **158.72 calls/MHz**

25. A 3-KHz voice signal is modulated using FM to create a 30-KHz analog signal. As we learned in Chapter 5, the bandwidth required for FM can be determined from the bandwidth of the audio signal using the formula $B_{FM} = 2(1 + \beta)B$. ***AMPS*** uses $\beta = 5$. This means $\mathbf{B_{FM} = 10 \times B}$.

27. ***GPS*** satellites are orbiting at 18,000 km above the earth surface. Considering the radius of the earth, the radius of the orbit is then (18,000 km + 6378 km) = 24,378 km. Using the Kepler formula, we have

Period = (1/100) (distance) $^{1.5}$ = (1/100) $(24{,}378)^{1.5}$ = 38062 s = **10.58 hours**

29. ***Globalstar*** satellites are orbiting at 1400 km above the earth surface. Considering the radius of the earth, the radius of the orbit is then (1400 km + 6378 km) = 7778 km. Using the Kepler formula, we have

Period = (1/100) (distance) $^{1.5}$ = (1/100) $(7778)^{1.5}$ = 6860 s = **1.9 hours**

CHAPTER 17

SONET/SDH

Solutions to Odd-Numbered Review Questions and Exercises

Review Questions

1. The ANSI standard is called ***SONET*** and the ITU-T standard is called ***SDH***. The standards are nearly identical.
3. ***STS multiplexers/demultiplexers*** mark the beginning points and endpoints of a SONET link. An STS multiplexer multiplexes signals from multiple electrical sources and creates the corresponding optical signal. An STS demultiplexer demultiplexes an optical signal into corresponding electric signals. ***Add/drop multiplexers*** allow insertion and extraction of signals in an STS. An add/drop multiplexer can add an electrical signals into a given path or can remove a desired signal from a path.
5. ***Pointers*** are used to show the ***offset*** of the SPE in the frame or for ***justification***. SONET uses two pointers show the position of an SPE with respect to an STS. SONET use the third pointer for rate adjustment between SPE and STS.
7. A ***regenerator*** takes a received optical signal and regenerates it. The SONET regenerator also replaces some of the existing overhead information with new information.
9. The ***path layer*** is responsible for the movement of a signal from its source to its destination. The ***line layer*** is responsible for the movement of a signal across a physical line. The ***section layer*** is responsible for the movement of a signal across a physical section. The ***photonic layer*** corresponds to the physical layer of the OSI model. It includes physical specifications for the optical fiber channel. SONET uses NRZ encoding with the presence of light representing 1 and the absence of light representing 0.

Exercises

11. Each STS-*n* frame carries $(9 \times \boldsymbol{n} \times 86)$ bytes of bytes. SONET sends 8000 frames in each second. We can then calculate the user data rate as follows:

$$\text{STS-}\mathbf{3} \quad \rightarrow \quad 8000 \times (9 \times \mathbf{3} \times 86) \times 8 = \mathbf{148.608\ Mbps}$$

STS-**9** → 8000 × (9 × **9** × 86) × 8 = **445.824 Mbps**
STS-**12** → 8000 × (9 × **12** × 86) × 8 = **594.432 Mbps**

13. The user data rate of STS-1 is (8000 × 9 × 86 × 8) = 49.536 Mbps. To carry a load with a data rate 49.540, we need another 4 kbps. This means that we need to insert 4000 / 8 = 500 bytes into every 8000 frames. In other words, ***500 out of every 8000*** frames need to allow the H3 byte to carry data. For example, we can have sequences of 16 frames in which the first frame is an overloaded frame and then 15 frames are normal.

15. In answering this question, we need to think about the three upper layers in SONET. The path layer is responsible for end-to-end communication. The line layer is responsible between multiplexers. The section layer is responsible between devices.

    a. ***A1*** and ***A2*** are used as ***aligners*** (synchronizers). They perform the same job as a preamble or flag field in other networks. We can call them ***framing bytes***. These bytes are set and renewed at each device to synchronize the two adjacent devices. There is no need for these bytes at the line or path layer.

    b. ***C1*** is used at the section layer to identify multiplexed STSs. This idea can be compared to statistical TDM in which each slot needs an address. In other words, C1 is the address of each STS-1 in an STS-n. C2 is like the port numbers in other protocols. When different processes need to communicate over the same network, we need port addresses to distinguish between them. There is no need for C byte at the line layer.

    c. ***D*** bytes are used for SONET administration. SONET requires two separate channels at the section (device-to-device) and line (multiplexer-to-multiplexer) layers.No administration is provided at the line layer.

    d. ***E*** byte creates a voice communication channel between two devices at the ends of a section.

    e. ***F*** bytes also create a voice communication. F1 is used between two devices at the end of a section; F2 is used between two ends. No bytes are assigned at the line layer.

    f. The only ***G*** bytes are used for status reporting. A device at the end of the path reports its status to a device at the beginning of the path. No other layer needs this byte.

    g. ***H*** bytes are the pointers. H1 and H2 are used to show the offsetting of the SPE with respect to STS-1. H3 is used to compensate for a faster or slower user data. All three are used in the line layer because add/drop multiplexing is done at this layer. H4 is used at the path layer to show a multiframe payload. Obviously we do not need an H byte in the section layer because no multiplexing or demultiplexing happens at this layer.

    h. The only ***J*** byte is at the path layer to show the continuous stream of data at the path layer (end-to-end). The user uses a pattern that must be repeated to show the stream is going at the right destination. There is no need for this byte at the other layers.

i. As we discussed, ***K*** bytes are used for automatic protection switching, which happens at the line layer (multiplexing). Other layers do not need these bytes.

j. Z bytes are unused bytes. All of the bytes in SOH are assigned, but in LOH and POH some bytes are still unused.

CHAPTER 18

Virtual Circuit Switching: Frame Relay and ATM

Solutions to Odd-Numbered Review Questions and Exercises

Review Questions

1. ***Frame Relay*** does not use *flow* or *error control*, which means it does not use the sliding window protocol. Therefore, there is no need for ***sequence numbers***.
3. ***T-lines*** provide point-to-point connections, not many-to-many. In order to connect several LANs together using T-lines, we need a mesh with many lines. Using Frame Relay we need only one line for each LAN to get connected to the Frame Relay network.
5. ***Frame Relay*** does not define a specific protocol for the physical layer. Any protocol recognized by ANSI is acceptable.
7. A ***UNI*** (user network interface) connects a user access device to a switch inside the ATM network, while an ***NNI*** (network to network interface) connects two switches or two ATM networks.
9. An ATM virtual connection is defined by two numbers: a ***virtual path identifier (VPI)*** and a ***virtual circuit identifier (VCI).***
11. In an UNI, the total length of VPI+VCI is 24 bits. This means that we can define 2^{24} virtual circuits in an UNI. In an NNI, the total length of VPI+VCI is 28 bits. This means that we can define 2^{28} virtual circuits in an NNI.

Exercises

13. We first need to look at the EA bits. In each byte, the EA bit is the last bit (the eight bit from the left). If EA bit is 0, the address ends at the current byte; if it 1, the address continues to the next byte.

    Address → **1011000<u>0</u> 0001011<u>1</u>**

    The first EA bit is 0 and the second is 1. This means that the address is only two bytes (no address extension). DLCI is only 10 bits, bits 1 to 6 and 9 to 12 (from left).

Address → **101100**00 **0001**0111
DLCI → **1011000001** → **705**

15. We first need to look at the EA bits. In each byte, the EA bit is the last bit (the eight bit from the left). If the EA bit is 0, the address ends at the current byte; if it 1, the address continues to the next byte.

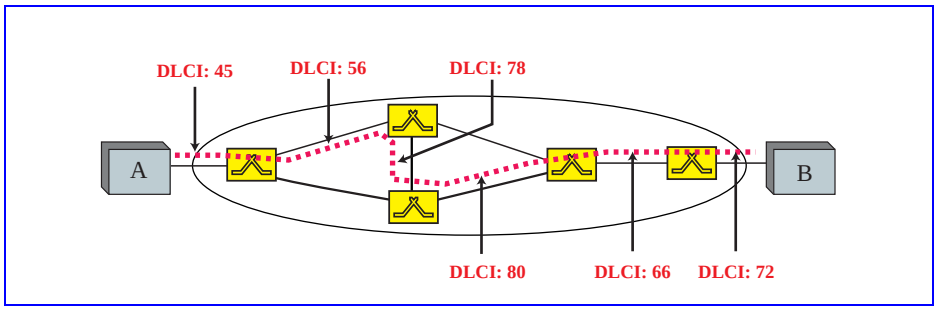
    Address → **0x7C74E1** → **0111110**0 **0111010**0 **1110000**1

    The first two EA bit are 0s and the last is 1. This means that the address is three bytes (address extension). DLCI is 16 bits, bits 1 to 6, 9 to 12, and 17 to 22.

    Address → **011111**00 **0111**0100 **111000**
    DLCI → **0111110111111000** → **32248**

17. See Figure 18.1.

Figure 18.1 *Solution to Exercise 17*



19. In AAL1, each cell carries only 47 bytes of user data. This means the number of cells sent per second can be calculated as [(2,000,000/8)/47] ≈ **5319.15**.

21.

    a. In AAL3/4, the CS layer needs to pass 44-byte data units to SAR layer. This means that the total length of the packet in the CS layer should be a multiple of 44. We can find the smallest value for padding as follows:

    H + Data + Padding + T = 0 mod 44
    4 + 47,787 + Padding + 4 = 0 mod 44
    Padding = **33 bytes**

    b. The number of data unit in the SAR layer is

    (4 + 47787 + 33 +4) / 44 = **1087**

    c. In AAL3/4, the number of cells in the ATM layer is the same as the number of data unit in the SAR layer. This means we have **1087 cells**.

23.

    a. The minimum number of cells is **1**. ***This happens when the data size ≤ 36 bytes***. Padding is added to make it exactly 36 bytes. Then 8 bytes of header creates a data unit of 44 bytes at the SAR layer.

b. The maximum number of cells can be determined from the maximum number of data units at the CS sublayer. If we assume no padding, the maximum size of the packet is 65535 + 8 = 65543. This needs 65543 / 44 ≈ 1489.61. The maximum number of cells is **1490**. ***This happens when the data size is between 65,509 and 65,535 (inclusive) bytes.*** We need to add between 17 to 43 (inclusive) bytes of padding to make the size 65552 bytes. The 8 byte overhead at the CS layer makes the total size 65560 which means 1490 data units of size 44.

25. AAL1 takes a ***continuous stream*** of bits from the user without any boundaries. There are always bits to fill the data unit; there is no need for padding. The other AALs take a bounded packet from the upper layer.

27. In AAL5 the number of bytes in CS, after adding padding and trailer must be multiple of 48.

    a. When user (**user data** + **8**) mod 48 = 0.

    b. When user (**user data** + **40** + **8**) mod 48 = 0.

    c. When user (**user data** + **43** + **8**) mod 48 = 0.

CHAPTER 19

Network Layer: Logical Addressing

Solutions to Odd-Numbered Review Questions and Exercises

Review Questions

1. An ***IPv4*** address is **32** bits long. An ***IPv6*** address is **128** bits long.
3. ***Classful addressing*** assigns an organization a Class A, Class B, or Class C block of addresses. ***Classless addressing*** assigns an organization a block of contiguous addresses based on its needs.
5. A ***block in class A*** address is ***too large*** for almost any organization. This means most of the addresses in class A are wasted and not used. ***A block in class C*** is probably ***too small*** for many organizations.
7. The ***network address*** in a block of addresses is the first address. The ***mask*** can be **ANDed** with any address in the block to find the network address.
9. Multicast addresses in ***IPv4*** are those that start with the **1110** pattern. Multicast addresses in ***IPv6*** are those that start with the **11111111** pattern.

Exercises

11.
    a. $2^8 =$ **256**
    b. $2^{16} =$ **65536**
    c. $2^{64} =$ $\mathbf{1.846744737 \times 10^{19}}$
13. $3^{10} =$ **59,049**
15.
    a. **127.240.103.125**
    b. **175.192.240.29**
    c. **223.176.31.93**
    d. **239.247.199.29**
17.
    a. **Class E** (first four bits are 1s)
    b. **Class B** (first bit is 1 and second bit is 0)

c. **Class C** (first two bits are 1s and the third bit is 0)

d. **Class D** (first three bits are 1s and the fourth bit is 0)

19. With the information given, the first address is found by ANDing the host address with the mask 255.255.0.0 (/16).

Host Address:	**25**	**.**	**34**	**.**	**12**	**.**	**56**
Mask (ANDed):	**255**	.	**255**	.	**0**	.	**0**
Network Address (First):	**25**	**.**	**34**	**.**	**0**	**.**	**0**

The last address can be found by ORing the host address with the mask complement 0.0.255.255.

Host Address:	**25**	**.**	**34**	**.**	**12**	**.**	**56**
Mask Complement (ORed):	**0**	.	**0**	.	**255**	.	**255**
Last Address:	**25**	**.**	**34**	**.**	**255**	**.**	**255**

However, we need to mention that this is the largest possible block with 2^{16} addresses. We can have many small blocks as long as the number of addresses divides this number.

21.

a. $\log_2 500 = 8.95$ Extra 1s = 9 Possible subnets: **512** Mask: **/17** (8+9)

b. $2^{32-17} = 2^{15} =$ **32,768** Addresses per subnet

c. **Subnet 1:** The first address in the this address is the beginning address of the block or **16.0.0.0**. To find the last address, we need to write 32,767 (one less than the number of addresses in each subnet) in base 256 (0.0.127.255) and add it to the first address (in base 256).

First address in subnet 1:	**16**	**.**	**0**	**.**	**0**	**.**	**0**
Number of addresses:	**0**	.	**0**	.	**127**	.	**255**
Last address in subnet 1:	**16**	**.**	**0**	**.**	**127**	**.**	**255**

d. **Subnet 500:**
Note that the subnet 500 is not the last possible subnet; it is the last subnet used by the organization. To find the first address in subnet 500, we need to add 16,351,232 (499 × 32678) in base 256 (0. 249.128.0) to the first address in subnet 1. We have 16.0.0.0 + 0.249.128.0 = **16.249.128.0**. Now we can calculate the last address in subnet 500.

First address in subnet 500:	**16**	**.**	**249**	**.**	**128**	**.**	**0**
Number of addresses:	**0**	.	**0**	.	**127**	.	**255**
Last address in subnet 500:	**16**	**.**	**249**	**.**	**255**	**.**	**255**

23.

a. $\log_2 32 = 5$ Extra 1s = **5** Possible subnets: **32** Mask: **/29** (24 + 5)

b. $2^{32-29} =$ **8** Addresses per subnet

c. **Subnet 1:**
The first address is the beginning address of the block or **211.17.180.0**. To find the last address, we need to write 7 (one less than the number of addresses in each subnet) in base 256 (0.0.0.7) and add it to the first address (in base 256).

First address in subnet 1:	**211**	.	**17**	.	**180**	.	**0**
Number of addresses:	**0**	.	**0**	.	**0**	.	**7**
Last address in subnet 1:	**211**	.	**17**	.	**180**	.	**7**

d. **Subnet 32:**
To find the first address in subnet 32, we need to add 248 (31 × 8) in base 256 (0.0.0.248) to the first address in subnet 1. We have 211.17.180.0 + 0.0.0.248 or **211.17.180.248.** Now we can calculate the last address in subnet 32 as we did for the first address.

First address in subnet 32:	**211**	.	**17**	.	**180**	.	**248**
Number of addresses:	**0**	.	**0**	.	**0**	.	**7**
Last address in subnet 32:	**211**	.	**17**	.	**180**	.	**255**

25.

a. The number of address in this block is 2^{32-29} = **8**. We need to add 7 (one less) addresses (0.0.0.7 in base 256) to the first address to find the last address.

From:	**123**	.	**56**	.	**77**	.	**32**
	0	.	**0**	.	**0**	.	**7**
To:	**123**	.	**56**	.	**77**	.	**39**

b. The number of address in this block is 2^{32-27} = **32**. We need to add 31 (one less) addresses (0.0.0.31 in base 256) to the first address to find the last address.

From:	**200**	.	**17**	.	**21**	.	**128**
	0	.	**0**	.	**0**	.	**31**
To:	**200**	.	**17**	.	**21**	.	**159**

c. The number of address in this block is 2^{32-23} = 512. We need to add 511 (one less) addresses (0.0.1.255 in base 256) to the first address to find the last address.

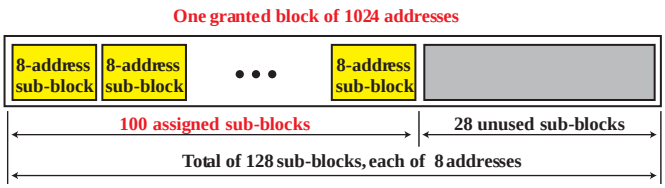
From:	**17**	.	**34**	.	**16**	.	**0**
	0	.	**0**	.	**1**	.	**255**
To:	**17**	.	**34**	.	**17**	.	**255**

d. The number of address in this block is $2^{32-30} = 4$. We need to add 3 (one less) addresses (0.0.0.3 in base 256) to the first address to find the last address.

From:	180	.	34	.	64	.	64
	0	.	**0**	.	**0**	.	**3**
To:	180	.	34	.	64	.	67

27. The site has $2^{32-22} = 2^{10} =$ **1024** from **120.60.4.0/22** to **120.60.7.255/22** addresses. One solution would be to divide this block into **128** 8-address sub-blocks as shown in Figure 19.1. The ISP can assign the first 100 sub-blocks to the current customers and keep the remaining 28 sub-blocks. Of course, this does not mean the future customer have to use 8-address subblocks. The remaining addresses can later be divided into different-size sub-blocks (as long as the three restrictions mentioned in this chapter are followed). Each sub-block has 8 addresses. The mask for each sub-block is **/29** ($32 - \log_2 8$). Note that the mask has changed from /22 (for the whole block) to /29 for each subblock because we have 128 sub-blocks ($2^7 = 128$).

Figure 19.1 *Solution to Exercise 27*



Sub-blocks:

1st subnet:	**120.60.4.0/29**	to	**120.60.4.7/29**
2nd subnet:	**120.60.4.8/29**	to	**120.60.4.15/29**
...	**...**		**...**
32nd subnet:	**120.60.4.248/29**	to	**120.60.4.255/29**
33rd subnet:	**120.60.5.0/29**	to	**120.60.5.7/29**
...	**...**		**...**
64th subnet:	**120.60.5.248/29**	to	**120.60.5.255/29**
...	**...**		**...**
99th subnet:	**120.60.7.16/29**	to	**120.60.7.23/29**
100th subnet:	**120.60.7.24/29**	to	**120.60.7.31/29**

1024 − 800 = **224** addresses left (from **120.60.7.31** to 120.60.7.155)

29.

a. **2340:1ABC:119A:A000::0**

b. **0:AA::119A:A231**

c. **2340::119A:A001:0**

d. **0:0:0:2340::0**

31.

a. ***Link local address***

b. ***Site local address***

c. ***Multicast address*** (permanent, link local)

d. ***Loopback address***

33. **58ABC1**

35.

a. **FE80:0000:0000:0000:0000:0000:0000:0123** or **FE80::123**

b. **FEC0:0000:0000:0000:0000:0000:0000:0123** or **FEC0::123**

37. The node identifier is **0000:0000:1211**. Assuming a 32-bit subnet identifier, the subnet address is **581E:1456:2314:ABCD:0000** where **ABCD:0000** is the subnet identifier.

CHAPTER 20

Network Layer: Internet Protocol

Solutions to Odd-Numbered Review Questions and Exercises

Review Questions

1. The delivery of a frame in the data link layer is ***node-to-node.*** The delivery of a packet at the network layer is ***host-to-host***.

3. Each data link layer protocol has a limit on the size of the packet it can carry. When a datagram is encapsulated in a frame, the total size of the datagram must be less than this limit. Otherwise, the datagram must be ***fragmented***. IPv4 allows fragmentation at the host and any router; IPv6 allows fragmentation only at the host.

5. ***Options*** can be used for network testing and debugging. We mentioned six options: no-operation, end-of-option, record-route, strict-source-route, loose-source-route, and timestamp. A ***no-operation*** option is a 1-byte option used as a filler between options. An ***end-of-option*** option is a 1-byte option used for padding at the end of the option field. A ***record-route*** option is used to record the Internet routers that handle the datagram. A ***strict-source-route*** option is used by the source to predetermine a route for the datagram. A ***loose-source-route*** option is similar to the strict source route, but it is less rigid. Each router in the list must be visited, but the datagram can visit other routers as well. A ***timestamp*** option is used to record the time of datagram processing by a router.

7. In IPv4, priority is handled by a field called ***service type*** (in the early interpretation) or ***differential services*** (in the latest interpretation). In the former interpretation, the three leftmost bits of this field define the priority or precedence; in the latter interpretation, the four leftmost bits of this field define the priority. In IPv6, the four-bit ***priority*** field handles two categories of traffic: *congestion-controlled* and *noncongestion-controlled*.

9. The ***checksum*** is eliminated in IPv6 because it is provided by upper-layer protocols; it is therefore not needed at this level.

Exercises

11. If no fragmentation occurs at the router, then the only field to change in the base header is the ***time to live*** field. If any of the multiple-byte options are present, then there will be changes in the option headers as well (to record the route and/or timestamp). If fragmentation does occur, the ***total length*** field will change to reflect the total length of each datagram. The ***more*** fragment bit of the flags field and the fragmentation ***offset*** field may also change to reflect the fragmentation. If options are present and fragmentation occurs, the ***header length*** field of the base header may also change to reflect whether or not the option was included in the fragments.

13.

Advantages of a large MTU:

- Good for transferring large amounts of data over long distances
- No fragmentation necessary; faster delivery and no reassembly
- Fewer lost datagrams
- More efficient (less overhead)

Advantages of a small MTU:

- Good for transferring time-sensitive data such as audio or video
- Better suited for multiplexing

15. The value of the header length field of an IP packet can never be less than 5 because every IP datagram must have at least a base header that has a fixed size of 20 bytes. The value of HLEN field, when multiplied by 4, gives the number of bytes contained in the header. Therefore the minimum value of this field is 5. This field has a value of exactly 5 when there are no options included in the header.

17. If the size of the option field is 20 bytes, then the total length of the header is 40 bytes (20 byte base header plus 20 bytes of options). The HLEN field will be the total number of bytes in the header divided by 4, in this case ten (1010 in binary).

19. Since there is no option information, the header length is 20, which means that the value of HLEN field is **5** or **0101** in binary. The value of total length is 1024 + 20 or **1044** (**00000100 00010100** in binary).

21. If the M (*more*) bit is zero, this means that the datagram is either the last fragment or the it is not fragmented at all. Since the *offset* is 0, it cannot be the last fragment of a fragmented datagram. ***The datagram is not fragmented***.

23. Let us first find the value of header fields before answering the questions:

    VER = 0x4 = **4**
    HLEN =0x5 = 5 $\rightarrow 5 \times 4 =$ **20**
    Service =0x00 = **0**
    Total Length = 0x0054 = **84**
    Identification = 0x0003 = **3**
    Flags and Fragmentation = 0x0000 $\rightarrow$ D = **0** M= **0** offset = **0**
    Time to live = 0x20 = **32**
    Protocol = 0x06 = **6**

Checksum = 0x5850
Source Address: 0x7C4E0302 = **124.78.3.2**
Destination Address: 0xB40E0F02 = **180.14.15.2**

We can then answer the questions:

a. If we calculate the checksum, we get 0x0000. ***The packet is not corrupted***.
b. Since the length of the header is 20 bytes, ***there are no options.***
c. Since *M* = 0 and *offset* = 0, ***the packet is not fragmented.***
d. The total length is 84. ***Data size is 64 bytes (84 –20)***.
e. Since the value of *time to live* = 32, ***the packet may visit up to 32 more routers***.
f. ***The identification number of the packet is 3***.
g. ***The type of service is normal***.

CHAPTER 21

Network Layer: Address Mapping, Error Reporting, and Multiplexing

Solutions to Odd-Numbered Review Questions and Exercises

Review Questions

1. The size of an ARP packet is ***variable***, depending on the length of the logical and physical addresses used.
3. The size of the ARP packet in Question 2 is 28 bytes. We need to pad the data to have the minimum size of **46**. The size of the packet in the Ethernet frame is then calculated as 6 + 6 + 2 + **46** + 4 = **64 bytes** (without preamble and SFD).
5. This restriction prevents ICMP packets from ***flooding*** the network. Without this restriction an endless flow of ICMP packets could be created.
7. A host would never receive a redirection message if there is only ***one router*** that connects the local network to the outside world.
9. The minimum size of an IP packet that carries an ICMP packet would be **28 bytes** (a 20 byte IP header + an 8 byte router solicitation packet). The maximum size would be **2068 bytes** (a 20 byte IP header + a 2048 byte router advertisement packet).
11. The minimum size would be **64 bytes** if we do not consider the preamble and SFD fields, which are added at the physical layer. The maximum size would be **1518** bytes, again not considering the preamble and SFD fields. Although the maximum size of an ICMP packet can be much more than 1500 bytes (for a router advertisement packet), Ethernet can carry only 1500 bytes of it.

Exercises

13. See Figure 21.1. Note that all values are in hexadecimal. Note also that the hardware addresses does not fit in the 4-byte word boundaries. We have also shown the IP address in parentheses.
15. See Figure 21.2. We have not shown the preamble and SFD fields, which are added in the physical layer.
17. It could happen that host B is unreachable, for some reason. The error message generated by an intermediate router could then be lost on its way back to host A.

Figure 21.1 *Solution to Exercise 13*

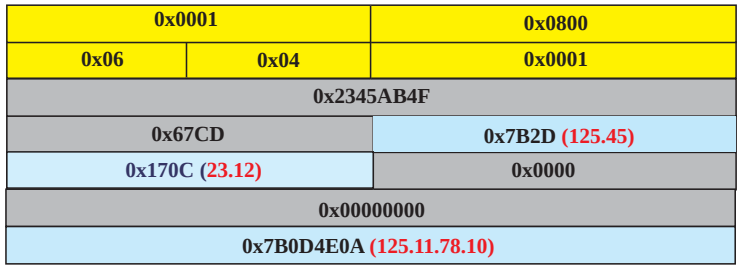
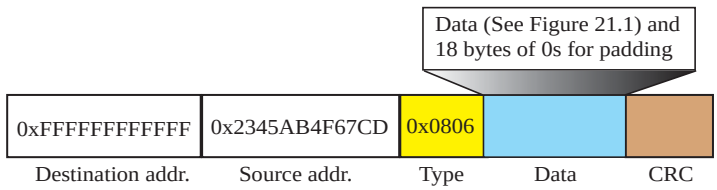


Figure 21.2 *Solution to Exercise 15*



Or perhaps the datagram was dropped due to congestion and the error message generated by an intermediate router was lost.

19. The appropriate ICMP message is ***destination unreachable*** message. This type of message has different types of codes to declare what is unreachable. In this case, the code is **0**, which means the network is unreachable (The codes are not discussed in the chapter; consult references for more information).

21. See the transformation process below:

IP: **11100111 0 0011000 00111100 00001001**
Ethernet: **00000001 00000000 01011110 0 0011000 00111100 00001001**

The Ethernet address in hexadecimal is **0x01005E183C09**

23. The host must send as many as **five different report messages** at random times in order to preserve membership in five different groups.

25. No action should be taken.

27. **Ethernet:**
Supported number of groups using 23 bits = 2^{23} = **8,388,608 groups**
IP:
Supported number of groups using 28 bits =2^{28} = **268,435,456 groups**
Address space lost:
268,435,456 – 8,388,608 = **260,046,848 groups**

CHAPTER 22

Network Layer: Delivery, Forwarding, and Routing

Solutions to Odd-Numbered Review Questions and Exercises

Review Questions

1. We discussed two different methods of delivery: direct and indirect. In a ***direct delivery***, the final destination of the packet is a host connected to the same physical network as the deliverer. In an ***indirect delivery*** the packet goes from router to router until it reaches the one connected to the same physical network as its final destination.

3. A routing table can be either static or dynamic. A ***static routing*** table contains information entered manually. A ***dynamic routing table*** is updated periodically by using one of the dynamic routing protocols such as RIP, OSPF, or BGP.

5. A RIP ***message*** is used by a router to request and receive routing information about an autonomous system or to periodically share its knowledge with its neighbors.

7. The ***hop count limit*** helps RIP instability by limiting the number of times a message can be sent through the routers, thereby limiting the back and forth updating that may occur if part of a network goes down.

9. In OSPF, four types of links have been defined: point-to-point, transient, stub, and virtual. A ***point-to-point*** link connects two routers without any other host or router in between. A ***transient*** link is a network with several routers attached to it. The packets can enter and leave through any of the routers. A ***stub*** link is a network that is connected to only one router. The data packets enter the network through this single router and leave the network through this same router. This is a special case of the transient network. When the link between two routers is broken, the administrator may create a ***virtual*** link between them, using a longer path that probably goes through several routers.

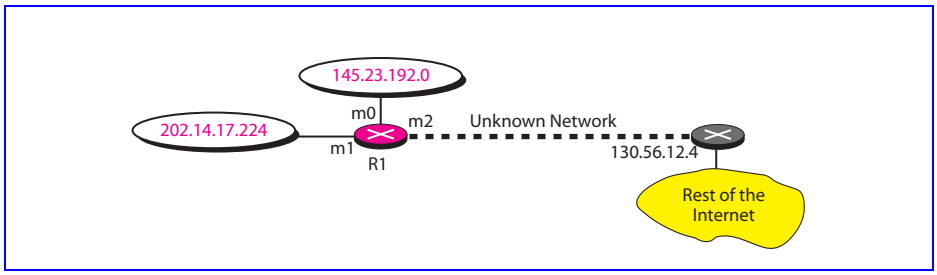
11. BGP is an ***interdomain*** routing protocol using path vector routing.

Exercises

13. A host that is totally isolated needs no routing information. ***The routing table has no entries***.

15. See Figure 22.1.

Figure 22.1 *Solution to Exercise 15*



17. R1 cannot receive a packet with this destination from ***m0*** because if any host in Organization 1 sends a packet with this destination address, the delivery is direct and does not go through R1. R1 can receive a packet with this destination from interfaces ***m1*** or ***m2***. This can happen when any host in Organization 2 or 3 sends a packet with this destination address. The packet arrives at R1 and is sent out through ***m0***. R1 can also receive a packet with this destination from interface **m3**. This happens in two cases. First, if R2 receives such a packet, the /24 is applied The packet is sent out from interface m0, which arrives at interface ***m3*** of R1. Second, if R3 receives such a packet, it applies the default mask and sends the packet from its interface ***m2*** to R2, which, in turn, applies the mask (/24) and sends it out from its interface ***m0*** to the interface ***m3*** of R1.

19. See Table 22.1.

Table 22.1 *Solution to Exercise 19: Routing table for local ISP 1*

Mask	Network address	Next-hop address	Interface
/23	120.14.64.0	---	**m0**
/23	120.14.66.0	---	**m1**
/23	120.14.68.0	---	**m2**
/23	120.14.70.0	---	**m3**
/23	120.14.72.0	---	**m4**
/23	120.14.74.0	---	**m5**
/23	120.14.76.0	---	**m6**
/23	120.14.78.0	---	**m7**
/0	0.0.0.0	**default router**	**m8**

21. See Table 22.2.

Table 22.2 *Solution to Exercise 21: Routing table for local ISP 3*

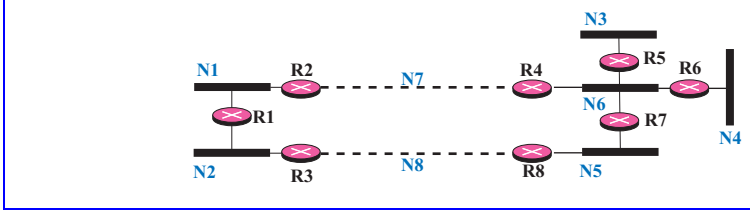
Mask	Network address	Next-hop address	Interface
/24	120.14.112.0	---	**m0**
/24	120.14.113.0	---	**m1**

Table 22.2 *Solution to Exercise 21: Routing table for local ISP 3*

Mask	Network address	Next-hop address	Interface
/24	120.14.114.0	---	**m2**
/24	120.14.115.0	---	**m3**
/24	120.14.116.0	---	**m4**
/24	120.14.117.0	---	**m5**
/24	120.14.118.0	---	**m6**
/24	120.14.119.0	---	**m7**
/24	120.14.120.0	---	**m8**
/24	120.14.121.0	---	**m9**
/24	120.14.122.0	---	**m10**
/24	120.14.123.0	---	**m11**
/24	120.14.124.0	---	**m12**
/24	120.14.125.0	---	**m13**
/24	120.14.126.0	---	**m14**
/24	120.14.127.0	---	**m15**
/0	0.0.0.0	**default router**	**m16**

23. In distance vector routing each router ***sends all of its knowledge about an autonomous system to all of the routers on its neighboring networks at regular intervals***. It uses a fairly simple algorithm to update the routing tables but results in a lot of unneeded network traffic. In link state routing a router ***floods an autonomous system with information about changes in a network only when changes occur***. It uses less network resources than distance vector routing in that it sends less traffic over the network but it uses the much more complex Dijkstra Algorithm to calculate routing tables from the link state database.
25. There are **2 + (10 × N)** = Empty bytes in a message advertising N networks
27. See Figure 22.2.

Figure 22.2 *Solution to Exercise 27*



29. **Transient networks**: **N1**, **N2**, **N5**, and **N6**. **Stub networks**: **N3** and **N4**
31. No, ***RPF*** does not create a shortest path tree because a network can receive more than one copy of the same multicast packet. RPF creates a graph instead of a tree.

33. Yes, ***RPM*** creates a shortest path tree because it is actually RPB (see previous answer) with pruning and grafting features. The leaves of the tree are the networks.

CHAPTER 23

Process-to-Process Delivery:

Solutions to Odd-Numbered Review Questions and Exercises

Review Questions

1. ***Reliability*** is not of primary importance in applications such as echo, daytime, BOOTP, TFTP and SNMP. In custom software, reliability can be built into the client/server applications to provide a more reliable, low overhead service.
3. ***Port addresses*** do not need to be universally unique as long as each IP address/port address pair uniquely identify a particular process running on a particular host. A good example would be a network consisting of 50 hosts, each running echo server software. Each server uses the well known port number 7, but the IP address, together with the port number of 7, uniquely identify a particular server program on a particular host. Port addresses are ***shorter*** than IP addresses because their domain, a single system, is smaller than the domain of IP addresses, all systems on the Internet.
5. The minimum size of a UDP datagram is **8** bytes at the transport layer and **28** bytes at the IP layer. This size datagram would contain no data–only an IP header with no options and a UDP header. The implementation may require padding.
7. The smallest amount of process data that can be encapsulated in a UDP datagram is **0** bytes.
9. See Table 23.1.

Table 23.1 *Answer to the Question 9.*

Fields in UDP	*Fields in TCP*	*Explanation*
Source Port Address	**Source Port Address**	
Destination Port Address	**Destination Port Address**	
Total Length		**There is no need for total length.**
Checksum	**Checksum**	
	Sequence Number	**UDP has no flow and error control.**
	Acknowledge Number	**UDP has no flow and error control.**
	Header Length	**UDP has no flow and error control.**

Table 23.1 *Answer to the Question 9.*

Fields in UDP	*Fields in TCP*	*Explanation*
	Reserved	**UDP has no flow and error control.**
	Control	**UDP has no flow and error control.**
	Window Size	**UDP has no flow and error control.**
	Urgent Pointer	**UDP cannot handle urgent data.**
	Options and Padding	**UDP uses no options.**

11.

a. None of the control bits are set. The segment is part of a data transmission without piggybacked acknowledgment.

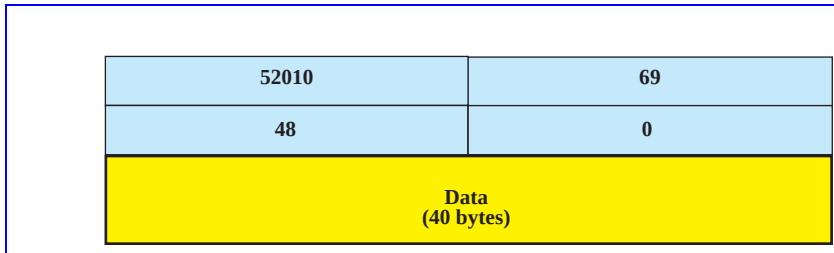
b. The ***FIN*** bit is set. This is a FIN segment request to terminate the connection.

c. The ***ACK*** and the ***FIN*** bits are set. This is a ***FIN+ACK*** in response to a received ***FIN*** segment.

Exercises

13. See Figure 23.1.

Figure 23.1 *Solution to Exercise 13*



15. The server would use the IP address **130.45.12.7**, combined with the well-known port number **69** for its source socket address and the IP address **14.90.90.33**, combined with an ephemeral port number as the destination socket address.

17. 16 bytes of data / 24 bytes of total length = **0.666**

19. 16 bytes of data / 72 byte minimum frame size = **0.222**

21. It looks as if both the destination IP address and the destination port number are corrupted. ***TCP calculates the checksum and drops the segment***.

23. See Figure 23.2.

25. Every second the counter is incremented by 64,000 × 2 = **128,000**. The sequence number field is 32 bits long and can hold only 2^{32}–1. So it takes (2^{32}–1)/(128,000) seconds or **33,554** seconds.

27. See Figure 23.3.

29. The largest number in the sequence number field is 2^{32} –1. If we start at 7000, it takes [(2^{32} – 1) –7000] / 1,000,000 = **4295** seconds.

Figure 23.2 *Solution to Exercise 23*

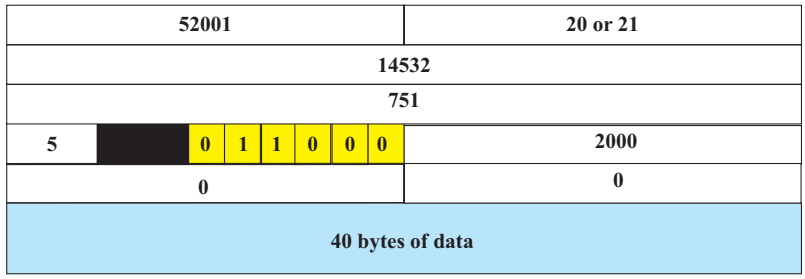
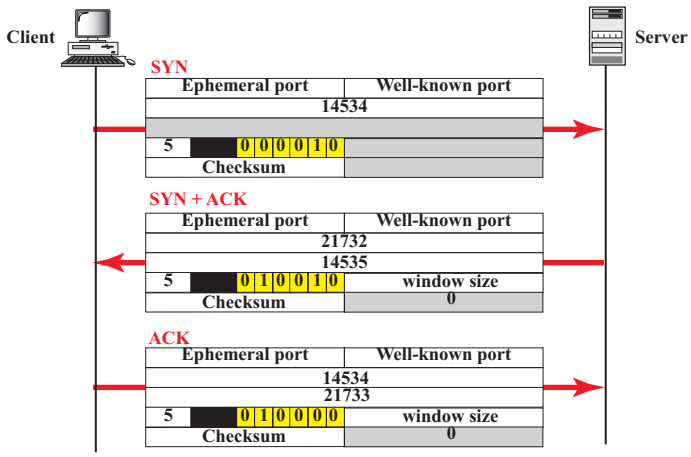


Figure 23.3 *Solution to Exercise 27*



31. See Figure 23.4.

33. See Figure 23.5.
    Note that the value of cumTSN must be updated to 8.

Figure 23.4 *Solution to Exercise 31*

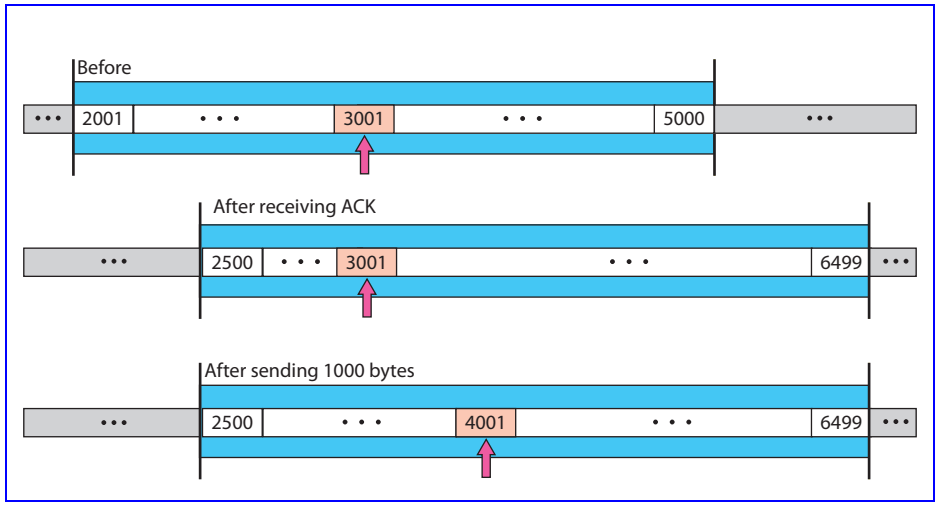
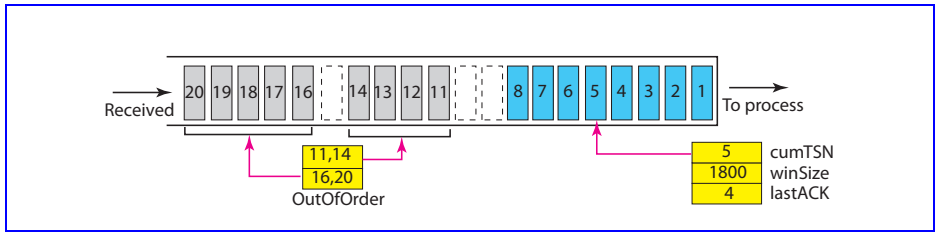


Figure 23.5 *Solution to Exercise 33*



CHAPTER 24

Congestion Control and Quality of Service

Solutions to Odd-Numbered Review Questions and Exercises

Review Questions

1. In ***congestion control***, the load on a network is prevented from exceeding the capacity. ***Quality of service*** refers to the characteristics that a flow of data seeks to attain. If there is good congestion control, then the QoS is also good and vice versa.

3. The ***average data rate*** is always less than or equal to the ***peak data rate***.

5. ***Open-loop*** congestion control policies try to prevent congestion. ***Closed-loop*** congestion control policies try to alleviate the effects of congestion.

7. Congestion can be alleviated by ***back pressure***, ***a choke point***, and ***explicit signaling***.

9. Frame Relay uses the ***BECN*** bit and the ***FECN*** bit to control congestion.

11. ***Scheduling***, ***traffic shaping***, ***admission control***, and ***resource reservation*** can improve QoS.

13. ***Differentiated Services*** was developed to handle the shortcomings of IntServ. The main processing was moved from the core of the network to the edge of the network. Also, the ***per-flow service*** was changed to ***per-class service***.

15. The attributes are ***access rate***, ***committed burst size***, ***committed information rate***, and ***excess burst size***.

Exercises

17. The bit pattern is 10110000 0001**0**1**1**1. The ***FECN*** bit is **0** and the ***BECN*** bit is **1**. There is no congestion in the forward direction, but there is congestion in the backward direction.

19.

    Input: $(100/60) \times 12 + 0 \times 48 =$ **20** gallons
    Output: **5** gallons
    Left in the bucket: $20 - 5 =$ **15**

21.

a. The access rate is the rate of T-1 line (**1.544 Mbps**) that connects the user to the network. Obviously, the user cannot exceed this rate.

b. The user data rate cannot exceed the access rate, the rate of the T-1 line that connects the user to the network. The user should stay below this rate (**1.544 Mbps**).

c. The CIR is **1 Mbps**. This means that the user can send data at this rate all the time without worrying about the discarding of data.

d. The user can send data at the rate of **1.2 Mbps** because it is below the access rate. However, the user sends 6 million bits per 5 seconds, which is above B_c (5 million per 5 seconds), but below B_c+B_e (6 million per 5 seconds). The network will discard no data if there is no congestion, but it may discard data if there is congestion.

e. The user can send data at the rate of **1.4 Mbps** because it is below the access rate. However, the user sends 7 million bits per 5 seconds, which is above B_c and above B_c+B_e (6 million per 5 seconds). In other words, the user rate is beyond its share. The network will discard some data to limit the data rate.

f. To be sure that the network never discard her data, the user should stay at or below CIR rate all the time, which means below or at **1 Mbps**.

g. If the user can accept possible data discarding in case of congestion, she can send at a higher rate if the number of bits is below B_c+B_e (6 million per 5 seconds in this case). This mans that the user can send at **1.2 Mbps** all the time if she accepts this risk.

23. CTD is the average *cell transfer delay.* If each cell takes 10 μs to reach the destination, we can say that CTD = $[(10\ \mu s \times n) / n]$ in which n is the total number of cells transmitted in a period of time. This means that CTD = **10 μs**

CHAPTER 25

Domain Name System (DNS)

Solutions to Odd-Numbered Review Questions and Exercises

Review Questions

1. When the name space is large, searching a name in ***hierarchical*** structure (tree) is much faster that searching it in a ***flat*** structure (linear). The first can use a binary search; the second needs to use a sequential search.
3. ***Generic domain***, ***country domain***, and ***inverse domain***.
5. In ***recursive resolution*** the client queries just one server. In ***iterative resolution*** the client queries more than one server.
7. A ***PQDN*** is a domain name that does not include all the levels between the host and the root node.
9. ***Caching*** reduces the search time for a name.
11. ***DDNS*** is needed because the many address changes makes manual updating inefficient.

Exercises

13.
    a. **FQDN**
    b. **FQDN**
    c. **PQDN**
    d. **PQDN**
15. Remembering a ***name*** is often easier than remembering a ***number.***
17. There are ***three labels*** but ***four levels*** of hierarchy since the root is considered a level.
19. This is a ***generic domain***.
21. The number of question sections and answer sections must be the same. The relationship is ***one-to-one***.

2

CHAPTER 26

Remote Log-in, Electronic Mail and File Transfer

Solutions to Odd-Numbered Review Questions and Exercises

Review Questions

1. In ***local log-in***, the user terminal is directly connected to the target computer; in ***remote log-in***, the user computer is connected to the target computer through the Internet.

3. Options in TELNET are negotiated using four control characters **WILL**, **WONT**, **DO**, and **DONT**.

5. A ***user agent (UA)*** is a software package that composes, reads, replies to, and forwards messages.

7. SMTP is a ***push*** protocol; it pushes the message from the client to the server. In other words, the direction of the bulk data (messages) is from the client to the server. On the other hand, retrieving messages from mail boxes needs a ***pull*** protocol; the client must pull messages from the server. The direction of the bulk data is from the server to the client. The third stage uses a message access agent (MAA) such as POP3 or IMAP4.

9. One connection is for ***data transfer***, the other connection is for ***control information***.

11. The three transmission modes in FTP are ***stream***, ***block***, and ***compressed***.

13. ***Anonymous FTP*** allows a user to access files without an account or password on a remote server.

Exercises

15. There are **15** characters in the command (including the end of line). Each character is sent separately to the server and each is echoed and acknowledged by the server. Each echo from the server is then acknowledged by the client. A total of **45** packets must be sent.

17. Three transmissions, each with a minimum size of 72 bytes, mean a total of **216 bytes** or **1728 bits**.

19.

a. **IAC WILL ECHO**
b. **IAC DONT ECHO**
c. **IAC IP** (Interrupt Process)
d. **IAC GA** (Go Ahead)

21.

MIME-version: 1.1
Content-Type: Image/JPEG; name="something.jpg"
Content-Transfer-Encoding: base64

23. There should be limitations on ***anonymous FTP*** because it is unwise to grant the public complete access to a system. If the commands that an anonymous user could use were not limited, that user could do great damage to the file system (e.g., erase it completely).

CHAPTER 27

WWW and HTTP

Solutions to Odd-Numbered Review Questions and Exercises

Review Questions

1. ***HTTP*** is a file transfer protocol that facilitates access to the ***WWW***.
3. ***HTTP*** is like ***FTP*** because they both transfer files and use the services of TCP.
5. A ***proxy server*** is a computer that keeps copies of responses to recent requests. When an HTTP client has a request, the cache of the proxy server is checked before the request goes to the regular server.
7. A ***Web document*** can be classified as either ***static***, ***active***, or ***dynamic***.
9. A ***dynamic document*** is the product of a program run by a server as requested by a browser. An ***active document*** is the product of a program sent from the server to the client and run at the client site.
11. ***Java*** is one of the languages that is used to write an ***active document***.

Exercises

13. On the screen you see: The publisher of this book is **McGraw-Hill Publisher**
15.

    HTTP/1.1 200 OK
    Date: Fri, 13-Jan-06 08:45:25 GMT
    Server: Challenger
    MIME-version: 1.0
    Content-length: 4623

    (Body of document)
17.

    HTTP/1.1 400 Bad Request
    Date: Fri, 13-Jan-06 08:45:25 GMT
    Server: Challenger

19.

HEAD /bin/users/file HTTP /1.1
Date: Fri, 13-Jan-06 10:40:22 GMT
MIME-version: 1.0
From: mzzchen@sinonet.cn

21.

COPY /bin/usr/bin/file1 HTTP /1.1
Date: Fri, 13-Jan-06 10:52:12 GMT
MIME-version: 1.0
Location: /bin/file1

23.

DELETE /bin/file1 HTTP /1.1
Date: Fri, 13-Jan-06 11:04:22 GMT
Server: Challenger
Authentication: swd22899/3X4ake88rTfh (Not discussed in the book)

25.

GET /bin/etc/file1 HTTP /1.1
Date: Fri, 13-Jan-06 11:22:08 GMT
MIME-version: 1.0
Accept: */*
If-modified-since: 23-Jan-1999 00:00:00 GMT

27.

GET /bin/etc/file1 HTTP /1.1
Date: Fri, 13-Jan-06 11:41:02 GMT
MIME-version: 1.0
Accept: */*
Host: Mercury
Authentication: swd22899/3X4ake88rTfh (Not discussed in the book)

29.

PUT /bin/letter HTTP /1.1
Date: Fri, 13-Jan-06 11:47:00 GMT
MIME-version: 1.0
Accept: text/html
Accept: image/gif
Accept: image/jpeg
Location: /bin/letter

CHAPTER 28

Network Management: SNMP

Solutions to Odd-Numbered Review Questions and Exercises

Review Questions

1. ***Network management*** is defined as monitoring, testing, configuring, and troubleshooting network components to meet a set of requirements defined by an organization.
3. The ***configuration management*** system updates information about the status of each entity and its relation to other entities.
5. ***Fault management*** supervises the operation of the network, which depends on the proper operation of each individual component and its relation to other components.
7. ***Performance management*** monitors and controls the network to ensure that it is running as efficiently as possible.
9. ***Security management*** is responsible for controlling access to the network based on the predefined policy.

Exercises

11.

INTEGER tag: **02**
length: **04**
value: **00 00 05 B0**

Answer: **02 04 00 00 05 B0**

13.

OCTET STRING tag: **04**
length of the length field (2 bytes) (10000010) = **82**
length (1000 bytes) = **03 E8**
value (1000 character)

Answer: **04 82 03 E8 (Plus 1000 bytes of characters)**

15.

30 15	sequence, length
43 04 00 00 2E E0	TIME TICK, length, value (1200)
02 04 00 00 38 E4	INTEGER, length, value (14564)
06 07 01 03 06 01 02 01 07	Object ID, length, value (1.3.6.2.1.7)

17.

30 43	sequence, length
30 41	sequence, length
02 04 00 00 09 29	INTEGER, length, value (2345)
04 08 43 4F 4D 50 55 54 45 52	OCTET STRING, length, value (COMPUTER)
41 04 00 00 01 59	counter, length, value (345)
30 29	sequence, length
02 04 00 00 04 63	INTEGER, length, value (1123)
04 04 44 49 53 4B	OCTET STRING, length, value (DISK)
41 04 00 00 05 96	counter, length, value (1430)
30 15	sequence, length
02 04 00 00 0D 80	INTEGER, length, value (3456)
04 07 4D 4F 4E 49 54 4F 52	OCTET STRING, length, value (MONITOR)
41 04 00 00 09 09	counter, length, value (2313)

CHAPTER 29

Multimedia

Solutions to Odd-Numbered Review Questions and Exercises

Review Questions

1. In ***streaming stored audio/video***, a client first downloads a compressed file and then listens to or watches it. ***In streaming live audio/video***, a client listens to or watches a file while it is being downloaded.
3. A ***metafile*** contains information about a corresponding audio/video file.
5. ***Jitter*** manifests itself as a gap between what is heard or seen.
7. ***JPEG*** is used to compress images. ***MPEG*** is used to compress video.
9. The ***DCT*** reveals the number of redundancies of a block.

Exercises

11.

   a. **9** packets played; **11** packets left
   b. **12** packets played; **8** packets left
   c. **17** packets played; **3** packets left
   d. **22** packets played; **8** packets left

13. We can say that ***UDP*** plus ***RTP*** is more suitable than ***TCP*** for multimedia communication. The combination uses the appropriate features of UDP, such as timestamp, multicasting, and lack of retransmission, and appropriate features of ***RTP*** such as error control.
15. The ***web server*** and ***media server*** can be two distinct machines since it is the metafile-data file combination that is important.

17. Both ***SIP*** and ***H.323*** use the Internet as a telephone network. The main difference is that H.323 uses a gateway to transform a telephone network message to an Internet message. See Table 29.1.

Table 29.1 *Solution to Exercise 17*

Issues	*SIP*	*H.323*
Transport layer	UDP or TCP	UDP for data, TCP for control
Address format	IP address, e-mail address, or phone number	IP address
Establishment	3-way handshake	H.225, Q.931, H.245
Data exchange	UDP, TCP	RTP, RTCP, UDP, TCP
Termination	BYE message	Q.931

19. ***H.323*** can also be used for video, but it requires the use of videophones. Currently most people don't have videophones.

CHAPTER 30

Cryptography

Solutions to Odd-Numbered Review Questions and Exercises

Review Questions

1. Only ***one key*** (the shared secret key) is needed for two-way communication. However, for more security, it is recommended that a different key be used for each direction.

3. Each person in the first group needs to have **10** keys to communicate with all people in the second group. This means we need at least $10 \times 10 =$ **100** keys. Note that the same keys can be used for communication in the reverse direction. However, note that we are not considering the communication between the people in the same group. For this purpose, we would need more keys.

5. For two-way communication, **4** keys are needed. Alice needs a private key and a public key; Bob needs a private key and a public key.

7. For two-way communication, the people in the first group need 10 pairs of keys, and the people in the second group need a separate 10 pairs of keys. In other words, for two-way communication **40** keys are needed.

Exercises

9. If the two persons have two pairs of asymmetric keys, then they can send messages using these keys to create a ***session symmetric key***, a key which is valid for one session and should not be used again. Another solution is to use a ***trusted center*** that creates and send symmetric keys to both of them using the symmetric key or asymmetric key that has been already established between each person and the trusted center. We will discuss this mechanism in Chapter 31.

11.

    a. We can show the encryption character by character. We encode characters A to Z as 0 to 25. To wrap, we subtract 26.

T	**19 + 20 = 39 – 26 = 13**	→	**N**
H	**07 + 20 = 27 – 26 = 01**	→	**B**
I	**08 + 20 = 28 – 26 = 02**	→	**C**

S	**18 + 20 = 38 – 26 = 12**	→	**M**
I	**08 + 20 = 28 – 26 = 02**	→	**C**
S	**18 + 20 = 38 – 26 = 12**	→	**M**
A	**00 + 20 = 20**	→	**U**
N	**13 + 20 = 33 – 26 = 07**	→	**H**
E	**04 + 20 = 24**	→	**Y**
X	**23 + 20 = 43 – 26 = 17**	→	**R**
E	**04 + 20 = 24**	→	**Y**
R	**17 + 20 = 37 – 26 = 11**	→	**L**
C	**02 + 20 = 22**	→	**W**
I	**08 + 20 = 28 – 26 = 02**	→	**C**
S	**18 + 20 = 38 – 26 = 12**	→	**M**
E	**04 + 20 = 24**	→	**Y**

The encrypted message is ***NBCM CM UH YRYLWCMY***.

b. We can show the decryption character by character. We encode characters A to Z as 0 to 25. To wrap the negative numbers, we add 26.

N	**13 – 20 = –07 + 26 = 19**	→	**T**
B	**01 – 20 = –19 + 26 = 07**	→	**H**
C	**02 – 20 = –18 + 26 = 08**	→	**I**
M	**12 – 20 = –08 + 26 = 18**	→	**S**
C	**02 – 20 = –18 + 26 = 08**	→	**I**
M	**12 – 20 = –08 + 26 = 18**	→	**S**
U	**20 – 20 = 00**	→	**A**
H	**07– 20 = –13 + 26 = 13**	→	**N**
Y	**24 – 20 = 04**	→	**E**
R	**17 – 20 = –03 + 26 = 23**	→	**X**
Y	**24 – 20 = 04**	→	**E**
L	**11 – 20 = –09 + 26 = 17**	→	**R**
W	**22 – 20 = 02**	→	**C**
C	**02 – 20 = –18 + 26 = 08**	→	**I**
M	**12 – 20 = –08 + 26 = 18**	→	**S**
Y	**24 – 20 = 04**	→	**E**

The decrypted message is ***THIS IS AN EXERCISE***.

13. We can, ***but it is not safe at all***. The best we can do is to change a 0 sometimes to 0 and sometimes to 1 and to change a 1 sometimes to 0 and sometimes to 1. It can be easily broken using trial and error.

15. Input: 111001 → output: **001111**

17.

    a. Input: **1 1 0 0 1 0** → output: **0 1**

    b. Input: **1 0 1 1 0 1** → output: **0 0**

19.

    a. Input: 1011 (the leftmost bit is 1), the output is: **110**

    b. Input: 0110 (the leftmost bit is 0), the output is: **011**

21. We can follow the process until we find the value of *d*. For the last step, we need to use an algorithm defined in abstract algebra. We don't expect students know how to do it unless they have taken a course in abstract algebra or cryptography.

    a. $n = p \times q = 19 \times 23 =$ ***437***

    b. $\phi = (p - 1) \times (q - 1) = 18 \times 22 =$ ***396***

    c. $e =$ ***5*** $d =$ ***317***

    We can check that $e \times d = 5 \times 317 = 1 \bmod 396$

23. Bob knows *p* and *q*, so he can calculate $\phi = (p - 1) \times (q - 1)$ and find d such that $d \times e = 1 \bmod \phi$. Eve does not know the value of *p* or *q*. She just knows that $n = p \times q$. If *n* is very large (hundreds of digits), it is very hard to factor it to *p* and *q*. Without knowing one of these values, she cannot calculate ϕ. Without ϕ, it is impossible to find *d* given *e*. The whole idea of RSA is that *n* should be so large that it is impossible to factor it.

25. The value of $e = 1$ means no encryption at all because **$C = P^e = P$**. The ciphertext is the same as plaintext. Eve can intercept the ciphertext and use it as plaintext.

27. Although Eve can use what is called the *ciphertext attack* to find Bob's key, she could have done it by intercepting the message. In the ciphertext attack, the intruder can get several different ciphertexts (using the same pair of keys) and find the private key of the receiver. If the value of the public key and *n* are very large, this is a very time-consuming and difficult task.

29. Nothing happens in particular. Assume both Alice and Bob choose x = y = 9. We have the following situation with $g = 7$ and $p = 23$:
    R1 = $7^9 \bmod 23 =$ **15**
    R2 = $7^9 \bmod 23 =$ **15**
    Alice calculates K = $(R2)^9 \bmod 23 = 15^9 \bmod 23 =$ **14**
    Bob calculates K = $(R1)^9 \bmod 23 = 15^9 \bmod 23 =$ **14**

4

CHAPTER 31

Network Security

Solutions to Odd-Numbered Review Questions and Exercises

Review Questions

1. A ***nonce*** is a large random number that is used ***only once*** to help distinguish a fresh authentication request from a repeated one.
3. Both the ***Needham-Schroeder*** and the ***Otway-Rees*** protocols use a ***KDC*** for user authentication.
5. The ***Kerberos TGS*** issues a ticket for the real server and provides the session key between the sender and the receiver.
7. A ***certification authority (CA)*** is a federal or state organization that binds a public key to an entity and issues a certificate.
9. A ***frequently-changed password*** is more secure than a ***fixed password*** but less secure than a ***one-time password.*** However, a one-time password needs more effort from the system and the user. The system needs to check if the password is fresh every time the user tries to use the password. The user needs to be careful not to use the pervious one. A more frequently changed password can be used as an alternative. One solution is that the system initializes the process of changing the password by sending the new password, through a secure channel, and challenging the user to be sure that the right user has received the new password.

Exercises

11.

a. The algorithm meets the first criteria (***one-wayness***). It is not possible to find the original numbers if the digest is given. For example, if we know the digest is 76, we cannot find the original ten numbers. They can be any set of 10 numbers.

b. The algorithm does not meet the second criteria (***weak collision***). If the digest is given, we can create 10 numbers that hash to the same digest. For example, Eve, without knowing the original set of numbers, can intercept the digest of **51** and create the set {12, 23, 45, 12, 34, 56, 9, 12, 34, 14} and send it with the digest **51** to Bob. Bob is fooled and believes that the set is authentic.

c. The algorithm does not meet the third criteria (***strong collision***). If the digest is given, we can create at least two sets of 10 numbers that hash to the same digest. For example, Alice can create two sets {12, 23, 45, 12, 34, 56, 9, 12, 34, 14} and {12, 23, 45, 16, 34, 56, 9, 12, 34, 10} that both hash to **51**. Alice can send the first set and the digest to Bob, but later she can claimed that she sent the second set.

13. The possible number of digests is 2^N because each bit can be in one of the two values (0 or 1).

15. The second and third criteria for a hashing function are closely related to the solution found in problem 14. In the problem we try to related the number of people at the party to the number of days in a year. In a hashing function, we can relate the number of possible messages to the number of possible digests. To understand the problem assume that there are only 10 possible messages (number of people at the party) but there are 365 possible digests.

    a. If a particular digest is given (a particular birthday), the probability that Eve can find one of the ten messages (one of the ten people in the party) is 0.027 (2.7 percent).This is related to the weak collision. The probability is very weak. That is why it is called ***weak collision***.

    b. The probability that Alice can create two or more messages with the same digests is the probability of finding two or more people with the same birthday in a party. If the number of possible messages is 10 and the number of possible digest is 365, this probability is 0.117 or (11 percent). That is why this criterion is called ***strong collision***. The probability is higher. It is more probable that Alice can find two or messages with the same digest than Eve can find a message with a given digest.

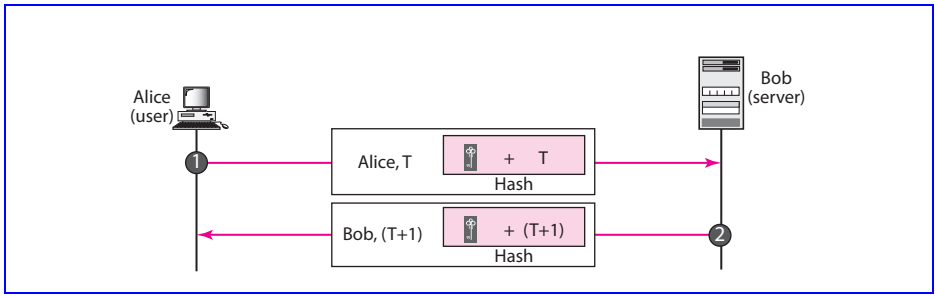
    The above discussion leads us to the point that we should worry more about the second criterion that the first. To decrease the probability of both criteria, we need to increase the number of possible digests and the number of possible messages. We need to increase the number of bits in a digest and impose a minimum number of bits on messages.

17. The whole idea of a sophisticated hash function such as ***SHA-1*** is that the partial digest of each block is dependent on the partial digest of the previous block and the message on the current block. Each block mingles and mixes the bits in a such a way that changing even one bit in the last block of the message may changed the whole final digest.

19. It is normally both. The entity authentication (based on the PIN) is needed to protect the person and the bank in case the money card is stolen. The message authentication is normally needed for the entity authentication.

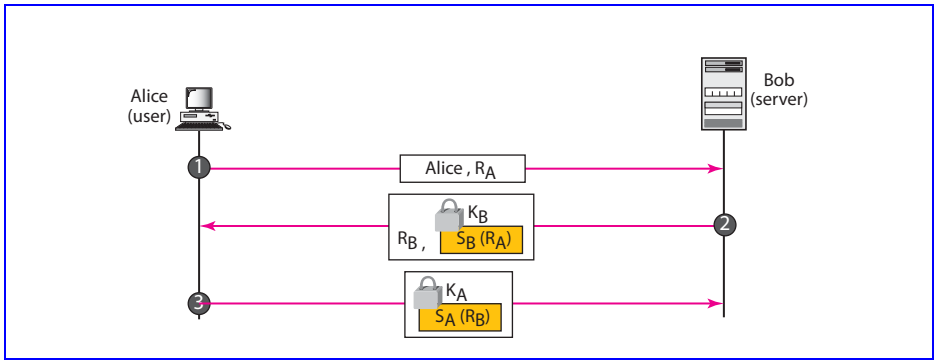
21. Figure 31.1. shows one scheme. Note that the scheme forces Bob to use the timestamp which is related to the timestamp used by Alice (T+1), this ensures that the two messages belongs to the same session.

Figure 31.1 *Solution to Exercise 21*



23. Figure 31.2 shows one simple scheme. Note that in the second message, Bob signs the message with his private key. When Alice verifies the message using Bob's public key, Bob is authenticated for Alice. In the third message, Alice signs the message with her private key. When Bob verifies the message using Alice's public key, Alice is authenticated for Bob.

Figure 31.2 *Solution to Exercise 23*



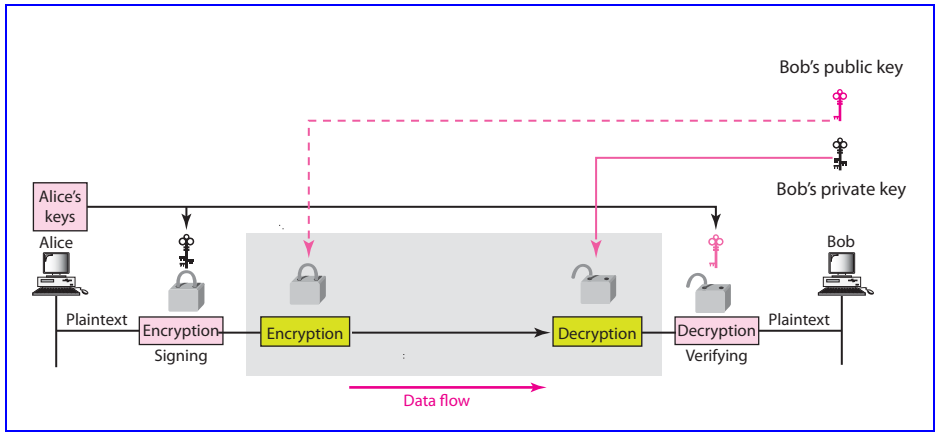
25. The ***timestamp*** definitely helps. If Alice adds a timestamp to the password before encrypting, the university, after decrypting, can check the freshness of the plaintext. In other words, adding a timestamp to a password, is like creating a new password each time.

27. If the ***KDC*** is down, nothing can take place. KDC is needed to create the session key for the two parties.

29. If the ***trusted center*** is down, Bob cannot obtain his certificate. Bob still can use his public key if the other party does not ask for a certificate.

31. See Figure 31.3. The shaded area shows the encryption/decryption layer.

Figure 31.3 *Solution to Exercise 31*



CHAPTER 32

Security In the Internet

Solutions to Odd-Numbered Review Questions and Exercises

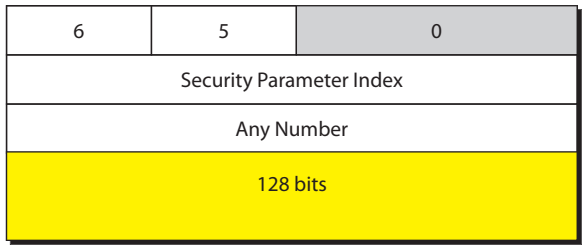
Review Questions

1. ***IPSec*** needs a set of security parameters before it can be operative. In IPSec, the establishment of the security parameters is done via a mechanism called ***security association (SA)***.
3. The two protocols defined by IPSec for exchanging datagrams are ***Authentication Header (AH)*** and ***Encapsulating Security Payload*** (ESP).
5. The ***Encapsulating Security Payload (ESP)*** protocol adds an ***ESP header***, ***ESP trailer***, and the ***digest***. The ESP header contains the security parameter index and the sequence number fields. The ESP trailer contains the padding, the padding length, and the next header fields. Note that the ***digest*** is a field separate from the header or trailer.
7. The two dominant protocols for providing security at the transport layer are the ***Secure Sockets Layer (SSL)*** Protocol and the ***Transport Layer Security (TLS)*** Protocol. The latter is actually an IETF version of the former.
9. A ***session*** between two systems is an association that can last for a long time; a ***connection*** can be established and broken several times during a session. Some of the security parameters are created during the session establishment and are in effect until the session is terminated. Some of the security parameters must be re-created (or occasionally resumed) for each connection.
11. One of the protocols designed to provide security for email is ***Pretty Good Privacy (PGP)***. ***PGP*** is designed to create authenticated and confidential e-mails.
13. The ***Handshake Protocol*** establishes a cipher set and provides keys and security parameters. It also authenticates the server to the client and the client to the server, if needed.
15. A ***firewall*** is a security mechanism that stands between the global Internet and a network. A firewall selectively filters packets.
17. A ***VPN*** is a technology that allows an organization to use the global Internet yet safely maintain private internal communication.

Exercises

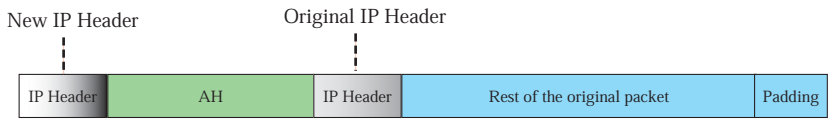
19. The only fields we can fill are the next header (assuming the packet encapsulates TCP) and the length field. The sequence number can be any number. Note that the length field defines the number of 32-bit words minus 2. See Figure 32.1.

Figure 32.1 *Solution to Exercise 19*



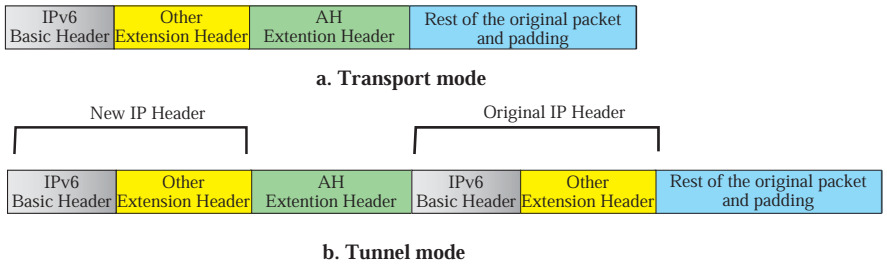
21. See Figure 32.2.

Figure 32.2 *Solution to Exercise 21*



23. See Figure 32.3.

Figure 32.3 *Solution to Exercise 23*



25. ***IPSec*** uses the services of IKE to create a security association that includes session keys. However, this does not start from scratch. Some kind of secret needs to exist between the two parties. In one of the methods used in IKE, the assumption is that

there is a ***shared secret key*** between the two parties. In this case, a ***KDC*** can be used to create this shared secret key.

27. Some ***SSL*** cipher suites need to use shared session keys. However, these session keys are created during hand-shaking. There is no need for a ***KDC***.

29. One of the purposes of ***PGP*** is to free the sender of the message from using a ***KDC***. In PGP, the session key is created and encrypted with the public key established between the sender and the receiver.

31. ***IPSec*** uses IKE to create security parameters. IKE has defined several methods to do so. Each method uses a different set of ciphers to accomplish its task However, the list of ciphers for each method is pre-defined. Although the two parties can choose any of the methods during negotiation, the cipher used for that particular method is predefined. In other words, we can say that IPSec has a list of method suites, but not a cipher suite.