CHAPTER 5

PRACTICE SET

Questions

Q5-1. 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).

Q5-2. A linear block code is a block code in which the exclusive-OR of any two codewords results in another codeword.

Q5-3. In this case, $k = 20$, $r = 5$, and $n = 20$. Five redundant bits are added to the dataword to create the corresponding codeword.

Q5-4. We have $k = 8$, $r = 2$, and $n = 8 + 2 = 10$.

a. The number of valid codewords is $2^k = 2^8 = 256$.

b. The number of invalid codewords is $2^n - 2^k = 2^{10} - 2^8 = 768$.

Q5-5. Communication at the network layer is host-to-host; communication at the data-link layer is node-to-node.

Q5-6. A point-to-point link is dedicated to the two devices connecting at the two ends of the link. A broadcast link shares its capacity between pairs of devices that need to use the link.

Q5-7. In variable-size framing, flags are needed to separate a frame from the previous one and the next one.

Q5-8. In character-oriented framing, the smallest unit of data for the link layer is a byte. In other words, the link layer is handling bytes as atomic data units that cannot be split. If something is supposed to be added to the datagram received from the network layer, it is a byte or a set of bytes.

Q5-9. We have $n = 2^r - 1 = 7$ and $k = n - 3 = 7 - 3 = 4$. A dataword has four bits and a codeword has seven bits. Although it is not asked in the question, we give the datawords and valid codewords below. Note that the minimum distance between the two valid codewords is 3.

Data	*Code*	*Data*	*Code*	*Data*	*Code*	*Data*	*Code*
0000	0000000	0100	0100011	1000	1000110	1100	1100101
0001	0001101	0101	0101110	1001	1001011	1101	1101000
0010	0010111	0110	0110100	1010	1010001	1110	1110010
0011	0011010	0111	0111001	1011	1011100	1111	1111111

Q5-10. We have $k = 5$ and $n = 8$. The size of the dividend is the same as the size of the codeword (8 bits). We need to augment the dataword with three 0s. The size of the remainder is $r = n - k = 3$ bits. The divisor is $r + 1 = 4$ bits.

Q5-11. The minimum Hamming distance is the smallest Hamming distance between all possible pairs in a set of words.

Q5-12. The Hamming distance $d_{min} = s + 1$. Since $s = 2$, we have $d_{min} = 3$.

Q5-13. In this case $r = 7 - 1 = 6$.

a. The length of the error is L = 5, which means $L \leq r$. All burst errors of this size will be detected.

b. The length of the error is L = 7, which means $L = r + 1$. This CRC will detect all burst errors of this size with the probability $1 - (0.5)^5 \approx 0.9688$. Almost 312 out of 10,000 errors of this length may be passed undetected.

c. The length of the error is L = 10, which means $L > r$. This CRC will detect all burst errors of this size with the probability $1 - (0.5)^6 \approx 0.9844$. Almost 156 out of 10,000 errors of this length may be passed undetected. Although the length of the burst error is increased, the probability of errors being passed undetected is decreased.

Q5-14. The error cannot be detected because the sum of items is not affected in this swapping.

Q5-15.

a. The generator has three bits (more than required). Both the rightmost bit and leftmost bits are 1s; it can detect all single-bit errors.

b. This cannot be used as a generator: the rightmost bit is 0.

c. This cannot be used as a generator; it has only one bit.

Q5-16.

a. The generator 10111 is qualified and divisible by 11 (the quotient is 1101); it can always detect an odd number of errors.

b. The generator 101101 is qualified and divisible by 11 (the result is 11011); it can always detect an odd number of errors.

c. The generator 111 is qualified, but not divisible by 11; it can detect an odd number of errors sometimes, but not always.

Q5-17. The transmission rate of this network is T_{fr} = (1000 bits)/(1Mbps) = 1 ms. The vulnerable time in pure Aloha is $2 \times T_{fr} = 2$ ms.

Q5-18. In a pure Aloha, the maximum throughput (18.4%) is achieved when G is 1/2. In both cases given, the throughput is decreased. When G = 1, the throughput is decreased to 13.5%; when G = 1/4, it is decreased to 15.2%.

Q5-19. The use of K in the figure decreases the probability that a station can immediately send when the number of failures increases. This means decreasing the probability of collision.

a. After one failure (K = 1), the value of R is 0 or 1. The probability that the station gets R = 0 (send immediately) is 1/2 or 50%.

b. After three failures (K = 3), the value of R is 0 to 7. The probability that the station gets R = 0 (send immediately) is 1/8 or 12.5%.

Q5-20. *Success* in an Aloha network is interpreted as receiving an acknowledgment for a frame.

Q5-21. The last bit is 10 μs behind the first bit.

a. It takes 5 μs for the first bit to reach the destination.

b. The last bit arrives at the destination 10 μs after the first bit.

c. The network is involved with this frame for 5 + 10 = 15 μs.

Q5-22. The sender needs to detect the collision before the last bit of the frame is sent out. If the collision occurs near the destination, it takes 2 × 3 = 6 μs for the collision news to reach the sender. The sender has already sent out the whole frame; it is not listening for a collision anymore.

Q5-23. The value of a checksum can be all 0s (in binary). This happens when the value of the sum (after wrapping) becomes all 1s (in binary).

Q5-24. In the first iteration, we have $R = 0 + D_1 = D_1$ and $L = 0 + R = D_1$. In the second iteration, we have $R = R + D_2 = D_1 + D_2$ and $L = L + R = D_1 + D_1 + D_2 = 2D_1 + D_2$. After n iterations we have the following:

$$\mathbf{R = D_1 + D_2 + \ldots + D_n} \quad \rightarrow \quad \mathbf{L = nD_1 + (n-1)D_2 + \ldots + D_n}$$

This shows that L is the weighted sum of the data items.

Q5-25. The address field in the HDLC network defines the address of the secondary station (as the sender or receiver); the primary station, which is always unique, does not need an address.

Q5-26. Only CSMA/CD is a random-access protocol. Polling is a controlled- access protocol. TDMA is a channelization protocol.

Q5-27. We do not need a multiple access protocol in this case. The DSL provides a dedicated point-to-point connection to the telephone office.

Q5-28. The size of an ARP packet is variable, depending on the length of the protocol and hardware addresses used.

Q5-29. ARP Packet Size = 2 + 2 + 1 + 1 + 2 + 6 + 4 + 6 + 4 = 28 bytes (Figure 5. 48).

Q5-30. We need to pad the data to achieve the minimum size of 46 bytes. The size of the packet in the Ethernet frame is then calculated as 6 + 6 + 2 + 46 + 4 = 64 bytes (without preamble and SFD). See Figure 5.55.

Q5-31. The answer is theoretically yes. A link-layer address has a local jurisdiction. This means that two hosts in different networks can have the same link-layer address, although this does not occur today because each NIC has a unique MAC address.

Q5-32. In random access methods, there is no control over channel access and there are no predefined channels. Each station can transmit when it desires. This liberty may create collisions.

Q5-33. In a full-duplex Ethernet, each station is connected to the switch and the media is divided into two channels for sending and receiving. No two stations compete to access the channels; each channel is dedicated.

Q5-34. The rates are as follows:

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

Q5-35. The common traditional Ethernet implementations are 10Base5, 10Base2, 10-Base-T, and 10Base-F.

Q5-36. The common Gigabit Ethernet implementations are 1000Base-SX, 1000Base-LX, 1000Base-CX, and 1000Base-T4.

Q5-37. Dial-up modems use part of the bandwidth of the local loop to transfer data. The latest dial-up modems use the V-series standards such as V.90 (56 kbps for downloading and 33.6 kbps for uploading), and V.92 (56 kbps for downloading and 48 kbps for uploading).

Q5-38. The traditional cable networks use only coaxial cables to distribute video information to the customers. The hybrid fiber-coaxial (HFC) networks use a combination of fiber-optic and coaxial cable to do so.

Q5-39. For calculating T_p, we need to consider the maximum length of frame transmission between any two stations. In this case, the maximum length is 500 + 700 = 1200 m.

Q5-40. A link-layer switch checks the destination link-layer address and sends out only one copy of the frame to the destination station. The other stations receive no copy. This is referred to as filtering. Filtering eliminates the need for the CSMA/CD protocol.

Q5-41. 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.

Q5-42. An NIC provides an Ethernet station with a 6-byte link-layer address. Most of the physical and data-link layer duties are also implemented in the NIC.

Q5-43. A single clock handles the timing of transmission and equipment across the entire network.

Q5-44. 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. 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.

Q5-45. A VLAN saves time and money because reconfiguration is done through software. Physical reconfiguration is not necessary.

Q5-46. A VLAN creates virtual workgroups. Each workgroup member can send broadcast messages to others in the workgroup. This eliminates the need for multicasting and all the overhead messages associated with it.

Q5-47. A TP (transmission path) is the physical connection between a user and a switch or between two switches. It is divided into several VPs (virtual paths), which provide a connection or a set of connections between two switches. VPs in turn consist of several VCs (virtual circuits) that logically connect two points.

Problems

P5-1. We have (vulnerable bits) = (data rate) × (burst duration). The last example shows how a noise of small duration can affect a large number of bits if the data rate is high.

a. **vulnerable bits** $= (1500) \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**

P5-2. This is a binomial distribution if we think of each bit as the outcome of tossing a coin. A corrupted bit is the *head* outcome; an uncorrupted bit is the *tail* outcome. The probability of x errors in an n-bit data unit is the probability of tossing a non-fair coin n times and expecting x heads:

$$\mathbf{P}\ [x \text{ bits in error}] = \mathbf{C}\ (n, x)\ p^x\ (1 - p)^{n-x}$$

P [1-bit error in 8-bit unit]	=	$C\ (8, 1)\ (0.2)^1\ (0.8)^7$	≈	**0.34**
P [3-bit error in 16-bit unit]	=	$C\ (16, 3)\ (0.3)^3\ (0.7)^{13}$	≈	**0.15**
P [10-bit error in 32-bit unit]	=	$C\ (32, 10)\ (0.4)^{10}\ (0.6)^{22}$	≈	**0.09**

P5-3. Each escape or flag byte must be pre-stuffed with an escape byte. The following shows the result:

D	**E**	**E**	D	D	**E**	**F**	D	D	**E**	**E**	**E**	**E**	D	**E**	**F**	D

P5-4. The following shows the result. We inserted three extra 1s.

`000111111111001111110100011111111111110000111`

P5-5. Answers are given below:

a. error **b.** error **c.** 0000 **d.** 1101

P5-6. The exclusive-OR of the second and the third codewords (01011) ⊕ (10111) is **11100,** which is not in the code. The code is not linear.

P5-7. The following shows the results. In the interpretation, 0 means a word of all 0 bits, 1 means a word of all 1 bits, and ~X means the complement of X.

a. **(10001) ⊕ (10001) = (00000)** Interpretation: X ⊕ X → 0

b. **(11100) ⊕ (00000) = (11100)** Interpretation: X ⊕ 0 → X

c. **(10011) ⊕ (11111) = (01100)** Interpretation: X ⊕ 1 → ~X

P5-8. The codeword for dataword 10 is 101. If a 3-bit burst error occurs, the codeword will be changed to 010. This pattern is not one of the valid codewords, so the receiver detects the error and discards the received pattern.

P5-9. The following shows the errors and how they are detected.

	C1	C2	C3	C4	C5	C6	C7	
R1	1	1	0	0	1	1	1	1
R2	1	0	1	1	1	0	1	1
R3	0	1	$\boxed{1}$	1	0	0	1	0
R4	0	1	0	1	0	0	1	1
	0	1	0	1	0	1	0	1

a. Detected and corrected

	C1	C2	C3	C4	C5	C6	C7	
R1	1	1	0	0	1	1	1	1
R2	1	0	1	1	1	0	1	1
R3	0	1	1	$\boxed{0}$	0	$\boxed{1}$	1	0
R4	0	1	0	1	0	0	1	1
	0	1	0	1	0	1	0	1

b. Detected

	C1	C2	C3	C4	C5	C6	C7	
R1	1	1	0	0	1	1	1	1
R2	1	0	1	$\boxed{0}$	$\boxed{0}$	0	1	1
R3	0	1	1	$\boxed{0}$	0	0	1	0
R4	0	1	0	1	0	0	1	1
	0	1	0	1	0	1	0	1

c. Detected

	C1	C2	C3	C4	C5	C6	C7	
R1	1	$\boxed{0}$	0	0	1	$\boxed{0}$	1	1
R2	1	0	1	1	1	0	1	1
R3	0	$\boxed{0}$	1	1	0	$\boxed{1}$	1	0
R4	0	1	0	1	0	0	1	1
	0	1	0	1	0	1	0	1

d. Not detected

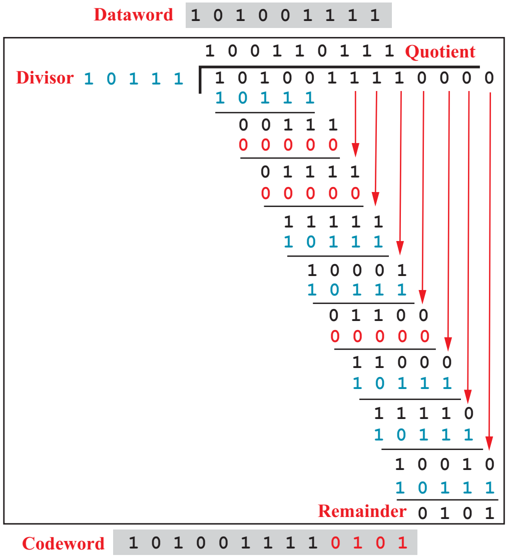
a. In the case of one error, it can be detected and corrected because the two affected parity bits can define where the error is.

b. Two errors can definitely be detected because they affect two bits of the column parity.The receiver knows that the message is somewhat corrupted (although not where). It discards the whole message.

c. Three errors are detected because they affect two parity bits, one of the column parity and one of the row parity. The receiver knows that the message is somewhat corrupted (although not where). It discards the whole message.

d. The last case cannot be detected because none of the parity bits are affected.

P5-10. The following shows the calculation of the codeword.



P5-11. The following shows the result. Part d shows that the Hamming distance between a word and itself is 0.

a. d (10000, 00000) = 1

b. d (10101, 10000) = 2

c. d (00000, 11111) = 5

d. d (00000, 00000) = 0

P5-12.

a. If we rotate **0101100** one bit, the result is **0010110**, which is in the code.

b. The XORing of the two codewords (0010110) ⊕ (1111111) = 1101001, which is in the code.

P5-13. The CRC-8 is 9 bits long, which means $r = 8$.

a. It has more than one bit and the rightmost and leftmost bits are 1s; it can detect a single-bit error.

b. Since $6 \leq 8$, a burst error of size 6 is detected.

c. Since $9 = 8 + 1$, a burst error of size 9 is detected most of the time; it may be left undetected with probability $(1/2)^{r-1}$ or $(1/2)^{8-1} \approx 0.008$.

d. Since $15 > 8 + 1$, a burst error of size 15 is detected most of the time; it may be left undetected with probability $(1/2)^{r}$ or $(1/2)^{8} \approx 0.004$.

P5-14. 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 separated for emphasis.

	Dataword		Number of 1s		Parity	Codeword
a.	1001011	→	4 (even)	→	0	10010110
b.	0001100	→	2 (even)	→	0	00011000
c.	1000000	→	1 (odd)	→	1	10000001
d.	1110111	→	6 (even)	→	0	11101110

P5-15. The sum in this case is $(FFFF)_{16}$ and the checksum is $(0000)_{16}$. The problem shows that the checksum can be all 0s in hexadecimal. It can be all Fs in the hexadecimal only if all data items are all 0s, which makes no sense.

P5-16.

a. We calculate R and L values in each iteration of the loop and then concatenate L and R to get the checksum. All calculations are in hexadecimal and modulo 256 or $(FF)_{16}$. Note that R needs to be calculated before L in each iteration ($L = L_{previous} + R$).

Initial values:	R = 00	L = 00
Iteration 1:	R = 00 + 2B = 2B	L = 00 + 2B = 2B
Iteration 2:	R = 2B + 3F = 6A	L = 2B + 6A = 95
Iteration 3:	R = 6A + 6A = D4	L = 95 + D4 = 69
Iteration 4:	R = D4 + AF = 83	L = 69 + 83 = EC

Checksum = **EC83**

b. The L and R values can be calculated as shown below (D_i is the corresponding bytes), which shows that L is the weighted sum of bytes.

$$R = D_1 + D_2 + D_3 + D_4 = 2B + 3F + 6A + AF = 83$$
$$L = 4 \times D_1 + 3 \times D_2 + 2 \times D_3 + 1 \times D_4 = EC$$

P5-17. The following shows the steps:

a. We first add the numbers to get $(0002A3BE)_{16}$. This corresponds to the first loop in Figure 5.17.

b. We extract the leftmost four digits, $(0002)_{16}$, and the rightmost four digits, (A3BE)16, and add them together to simulate the second loop in Figure 5.17. The result is $(A3C0)_{16}$. We stop here because the result does not create a carry.

c. Finally, we complement the result to get the checksum as $(5C3F)_{16}$.

P5-18. The following shows the steps:

a. We first add the numbers in two's complement to get 212,947.

b. We divide the above result by 65,536 (or 2^{16}). The quotient is 3 and the remainder is 16,339. The sum of the quotient and the remainder is 16,342.

c. Finally, we subtract the sum from 65,535 (or $2^{16} - 1$), simulating the complement operation, to get 49,193 as the checksum.

P5-19. We first calculate the sum modulo 10 of all digits. We then let the check digit to be 10 – sum. In this way, when the check digit is added to the sum, the result is 0 modulo 10.

$$C = [(1 \times 9) + (3 \times 7) + (1 \times 8) + (3 \times 0) + (1 \times 0) + (3 \times 7) + (1 \times 2) + (3 \times 9) + (1 \times 6) + (3 \times 7) + (1 \times 7) + (3 \times 5)] \bmod 10 = 137 \bmod 10 = 7 \quad \rightarrow \quad C = 10 - 7 = 3$$

P5-20. In both pure and slotted Aloha networks, the average number of frames created during a frame transmission time (T_{fr}) is G.

a. For a pure Aloha network, the vulnerable time is ($2 \times T_{fr}$), which means that $\lambda = 2G$.

$$p[x] = (e^{-\lambda} \times \lambda^x) / (x!) = (e^{-2G} \times (2G)^x) / (x!)$$

b. For a slotted Aloha network, the vulnerable time is (T_{fr}), which means that $\lambda = G$.

$$p[x] = (e^{-\lambda} \times \lambda^x) / (x!) = (e^{-G} \times (G)^x) / (x!)$$

P5-21. The probability of success for a station is the probability that the rest of the network generates no frame during the vulnerable time. However, since the number of stations is very large, it means that the network generates no frame. In other words, we are looking for *p*[0] in the Poisson distribution.

a. For a pure Aloha network, the vulnerable time is ($2 \times T_{fr}$), which means that $\lambda = 2G$.

$$P\,[\text{success for a frame}] = p[0] = (e^{-\lambda} \times \lambda^0) / (0!) = e^{-\lambda} = e^{-2G}$$

b. For a slotted Aloha network, the vulnerable time is (T_{fr}), which means that $\lambda = G$.

$$P\,[\text{success for a frame}] = p[0] = (e^{-\lambda} \times \lambda^0) / (0!) = e^{-\lambda} = e^{-G}$$

P5-22. The throughput for each network is $S = G \times P[\text{success for a frame}]$.

a. For a pure Aloha network, $P[\text{success for a frame}] = e^{-2G}$.

$$S = G \times P[\text{success for a frame}] = Ge^{-2G}$$

b. For a pure Aloha network, $P[\text{success for a frame}] = e^{-G}$.

$$S = G \times P[\text{success for a frame}] = Ge^{-G}$$

P5-23. We find dS/dG for each network and set the derivative to 0 to find the value of G. We then insert the G in the expression for S to find the maximum. It can be seen that the maximum throughput is the same for each network as we discussed in the text.

a. For a pure Aloha network, $S = Ge^{-2G}$.

$dS/dG = e^{-2G} - 2Ge^{-2G} = 0$	$\rightarrow$	$G_{max} = 1/2$
$S = Ge^{-2G}$	$\rightarrow$	If $G = 1/2$, $S_{max} = (e^{-1}) / 2 \approx 0.184$

b. For a slotted Aloha network, $S = Ge^{-G}$.

$dS/dG = e^{-G} - Ge^{-G} = 0$	$\rightarrow$	$G_{max} = 1$
$S = Ge^{-G}$	$\rightarrow$	If $G = 1$, $S_{max} = e^{-1} \approx 0.3678$

P5-24. We can find the probability for each network type separately:

a. In a pure Aloha network, a station can send a frame successfully if no other station has a frame to send during two frame transmission times (vulnerable time). The probability that a station has no frame to send is $(1 - p)$. The probability that none of the $N - 1$ stations have a frame to send is definitely $(1 - p)^{N - 1}$. The probability that none of the $N - 1$ stations have a frame to send during a vulnerable time is $(1 - p)^{2(N - 1)}$. The probability of success for a station is then

$$P[\text{success for a particular station}] = p\,(1 - p)^{2(N-1)}$$

b. In a slotted Aloha network, a station can send a frame successfully if no other station has a frame to send during one frame transmission time (vulnerable time).

$$P[\text{success for a particular station}] = p\,(1 - p)^{(N-1)}$$

P5-25. We found the success probability for each network type in the previous problem. If we multiply the success probability in each case by N, we have the throughput.

a. In a pure Aloha network with a limited number of stations, the throughput is

$$S = N \times P[\text{success for a particular station}] = Np\,(1 - p)^{2(N-1)}$$

b. In a slotted Aloha network with a limited number of stations, the throughput is

$$S = N \times P[\text{success for a particular station}] = Np\,(1 - p)^{(N-1)}$$

P5-26. To find the value of p that maximizes the throughput, we need to find the derivative of S with respect to p, dS/dp, and set the derivative to zero. Note that for large N, we can say $N - 1 \approx N$.

a. The following shows that, in a pure Aloha network, for a maximum throughput $p = 1/(2N)$ and the value of the maximum throughput for a large N is $S_{max} = e^{-1}/ 2$, as we found using the Poisson distribution:

$$\mathbf{S = Np\,(1-p)^{2(N-1)} \;\rightarrow\; dS/dp = N\,(1-p)^{2(N-1)} - 2Np(N-1)(1-p)^{2(N-1)-1}}$$

$$\mathbf{dS/dp = 0 \;\rightarrow\; (1-p) - 2(N-1)p = 0 \;\rightarrow\; p = 1/(2N-1) \approx 1/(2N)}$$

$$\mathbf{S_{max} = N[1/(2N)]\,[1 - 1/(2N)]^{2N} = (1/2)\,[1 - 1/(2N)]^{2N} = (1/2)\,e^{-1}}$$

b. The following shows that, in a slotted Aloha network, for a maximum throughput $p = 1/N$ and the value of the maximum throughput for a large N is $S_{max} = e^{-1}$, as we found using the Poisson distribution:

$$\mathbf{S = Np\,(1-p)^{(N-1)} \;\rightarrow\; dS/dp = N\,(1-p)^{(N-1)} - Np(N-1)(1-p)^{(N-1)-1}}$$

$$\mathbf{dS/dp = 0 \;\rightarrow\; (1-p) - (N-1)p = 0 \;\rightarrow\; p = 1/N.}$$

$$\mathbf{S_{max} = N[1/(N)]\,[1 - 1/(N)]^{N} = [1 - 1/(N)]^{N} = e^{-1}}$$

P5-27. We can first find the throughput for each station. Throughput of the network is the sum of the throughputs.

a. The throughput of each station is the probability that the station has a frame to send and other stations have no frame to send.

$$\mathbf{S_A = p_A\,(1-p_B)\,(1-p_C) = 0.2 \times 0.7 \times 0.6 \approx 0.084}$$

$$\mathbf{S_B = p_B\,(1-p_A)\,(1-p_C) = 0.3 \times 0.8 \times 0.6 \approx 0.144}$$

$$\mathbf{S_B = p_C\,(1-p_A)\,(1-p_B) = 0.4 \times 0.8 \times 0.7 \approx 0.224}$$

b. The throughput of the network is the sum of the throughputs.

$$\mathbf{S = S_A + S_B + S_C \approx 0.452}$$

P5-28. We first find the probability of success for each station in any slot (P_{SA}, P_{SB}, and P_{SC}). A station is successful in sending a frame in any slot if it has a frame to send and the other stations do not.

$$\mathbf{P_{SA} = (p_A)\,(1 - p_B)\,(1 - p_C) = (0.2)\,(1 - 0.3)\,(1 - 0.4) = 0.084}$$

$$\mathbf{P_{SB} = (p_B)\,(1 - p_A)\,(1 - p_C) = (0.3)\,(1 - 0.2)\,(1 - 0.4) = 0.144}$$

$$\mathbf{P_{SC} = (p_C)\,(1 - p_A)\,(1 - p_B) = (0.4)\,(1 - 0.2)\,(1 - 0.3) = 0.224}$$

We then find the probability of failure for each station in any slot (P_{FA}, P_{FB}, and P_{FC}).

$$\mathbf{P_{FA} = (1 - P_{SA}) = 1 - 0.084 = 0.916}$$

$$\mathbf{P_{FB} = (1 - P_{SB}) = 1 - 0.144 = 0.856}$$

$$\mathbf{P_{FC} = (1 - P_{SC}) = 1 - 0.224 = 0.776}$$

a. Probability of success for any frame in any slot is the sum of probabilities of success.

$$\mathbf{P[success\ in\ first\ slot]} = P_{SA} + P_{SB} + P_{SC} = (0.084) + (0.144) + (0.224) \approx 0.452$$

b. Probability of success for the first time in the second slot is the product of failure in the first and success in the second.

$$\mathbf{P[success\ in\ second\ slot\ for\ A]} = P_{FA} \times P_{SA} = (0.916) \times (0.084) \approx 0.077$$

c. Probability of success for the first time in the third slot is the product of failure in two slots and success in the third.

$$\mathbf{P[success\ in\ third\ slot\ for\ C]} = P_{FC} \times P_{FC} \times P_{SC} = (0.776)^2 \times (0.224) \approx 0.135$$

P5-29. Adler is a byte-oriented algorithm; data needs to be divided into bytes. For this reason, we need to represent each 16-bit data words in the problem into two bytes. The result is $(FB)_{16}$, $(FF)_{16}$, $(EF)_{16}$, and $(AA)_{16}$.

a. We calculate R and L values in each iteration of the loop and then concatenate L and R to get the checksum. All calculations are in hexadecimal and modulo 65521 or $(FFF1)_{16}$. Note that R needs to be calculated before L in each iteration ($L = L_{previous} + R$). Since the result in each iteration is smaller than $(FFF1)_{16}$, modular calculation does not show here, but we need to remember that it needs to be applied continuously.

Initial:	`R = 0001`	`L = 0000`
Iteration 1:	`R = 0001 + FB = 00FC`	`L = 0000 + 00FC = 00FC`
Iteration 2:	`R = OOFC + FF = 01FB`	`L = 00FC + 01FB = 02F7`
Iteration 3:	`R = 01FB + EF = 02EA`	`L = 02F7 + 02EA = 05E1`
Iteration 4:	`R = 02EA + AA = 0394`	`L = 05E1 + 0394 = 0975`

Checksum = `09750394`

b. The L and R values can be calculated as shown below (D_i is the corresponding bytes), which shows that L is the weighted sum of bytes.

$$R = 1 + D_1 + D_2 + D_3 + D_4 = 1 + FB + FF + EF + AA = 0394$$

$$L = 4 + 4 \times D_1 + 3 \times D_2 + 3 \times D_2 + 3 \times D_2 = 0975$$

P5-30. We show the calculation in three numbering systems.

Fields	Decimal	Hex	Binary
4, 5, and 0	17664	4500	01000101 00000000
36	36	0024	00000000 00100100
1	1	0001	00000000 00000001
0 and 0	0	0000	00000000 00000000
4 and 17	1041	0411	00000100 00010001
0	0	0000	00000000 00000000
180.124	46204	B47C	10110100 01111100
168.110	43118	A86E	10101000 01101110
201.126	51582	C97E	11001001 01111110
145.167	37287	91A7	10010001 10100111
Result	196933	30145	11 00000001 01000101
Sum	328	0148	00000001 01001000
Checksum	65207	FEB7	11111110 10110111

a. To find the checksum in decimal, we first add the fields to get the *result* in two's complement. We then divide the result by 65,536 (or 2^{16}) and add the quotient and the remainder to get the *sum in* one's complement. The checksum is found by subtracting the sum from 65,535 (or 2^{16}– 1).

b. To find the checksum in hexadecimal, we first add the fields to get the *result* in two's complement. We then wrap the extra digit (sum should be only four digits) and add it with the rest. The checksum is found by subtracting each digit from 15 (or 16 – 1).

c. To find the checksum in binary, we first add the fields to get the *result* in two's complement. We then wrap the extra two bits (sum should be only 16 bits) and add them with the rest. The checksum is found by flipping each bit.

P5-31. A slotted Aloha network is working with maximum throughput when G = 1.

a. The probability of an empty slot can be found by using the Poisson distribution when $x = 0$:

$$p[\text{empty slot}] = p[0] = (G^0\, e^{-G})/0! = e^{-1} = 0.3679$$

b. To calculate the average number of empty slots before getting a non-empty slot, we can use the Geometric distribution, which tells us that if a probability of an event is p, the number of experiments we need to try before getting that event is $1/p$. The following shows that we should wait on average 2.72 slots before getting an empty slot.

$$n = 1 / p[\text{empty slot}] \approx 2.72$$

P5-32. The data rate (R) defines how many bits are generated in one second and the propagation speed (V) defines how many meters each bit is moving per second. Therefore, the number of bits in each meter $n_{b/m} = R / V$. In this case,

$$n_{b/m} = R / V = (100 \times 10^6 \text{ bits/s}) / (2 \times 10^8 \text{ m/s}) = 1/2 \text{ bits/m}.$$

P5-33. In the previous problem, we defined the number of bits in one meter of the medium ($n_{b/m} = R/V$), in which R is the data rate and V is the propagation speed in the medium. If the length of the medium in meters is L_m, then $L_b = L_m \times n_{b/m}$. In this case, we have

$$n_{b/m} = R / V = (1 \times 2^8 \text{ m / s}) / (2 \times 2^8 \text{ m / s}) = 1/2 \text{ m/s}$$

$$L_b = L_m \times n_{b/m} = 200 \times (1/2) = 200 \times (1/2) = 100 \text{ bits}$$

P5-34. Let L_m be the length of the medium in meters, V the propagation speed, R the data rate, and $n_{b/m}$ the number of bits that can fit in each meter of the medium (defined in the previous problems). We can then proceed as follows:

$$a = (T_p) / (T_{fr}) = (L_m / V) / (F_b / R) = (L_m / F_b) \times (R / V)$$

$$\text{We have } (R / V) = n_{b/m} \rightarrow a = (L_m / F_b) \times (n_{b/m})$$

$$\text{Since } L_b = L_m \times n_{b/m} \rightarrow a = (L_b / F_b)$$

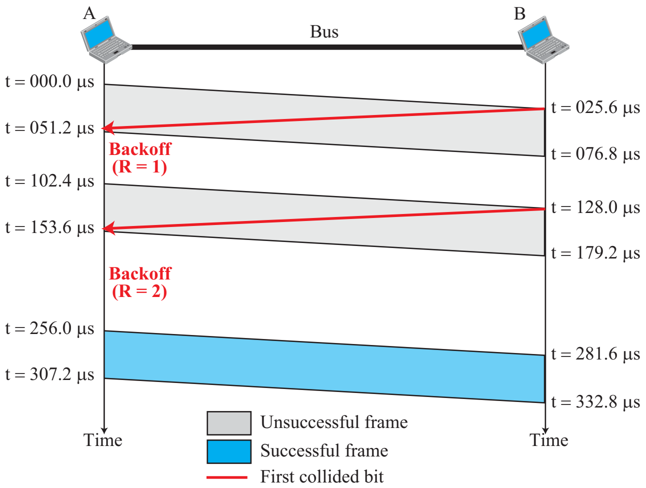
P5-35. We recalculate a new checksum with the value of the all fields as shown below. Since the checksum is zero, there is no corruption in the header.

Fields	Decimal	Hex	Binary
4, 5, and 0	17664	4500	01000101 00000000
36	36	0024	00000000 00100100
1	1	0001	00000000 00000001
0 and 0	0	0000	00000000 00000000
4 and 17	1041	0411	00000100 00010001
65207	65207	FEB7	11111110 10110111
180.124	46204	B47C	10110100 01111100
168.110	43118	A86E	10101000 01101110
201.126	51582	C97E	11001001 01111110
145.167	37287	91A7	10010001 10100111
Result	262140	3FFFC	11 11111111 11111100
Sum	65535	FFFF	11111111 11111111
Checksum	00000	0000	00000000 00000000

P5-36. We use modulo-11 calculation to find the check digit:

$$C = (1 \times 0) + (2 \times 0) + (3 \times 7) + (4 \times 2) + (5 \times 9) + (6 \times 6) + (7 \times 7) + (8 \times 7) + (9 \times 5) \bmod 11 = 7$$

P5-37. See the following figure.



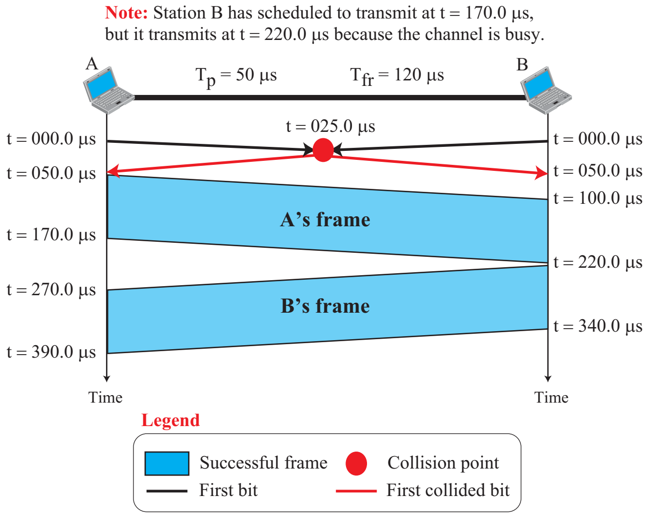
P5-38. The first bit of each frame needs at least 25 μs to reach its destination.

a. The frames collide because 2 μs before the first bit of A's frame reaches the destination, station B starts sending its frame. The collision of the first bit occurs at t = 24 μs.

b. The collision news reaches station A at time t = 24 μs + 24 μs = 48 μs. Station A has finished transmission at t = 0 + 40 = 40 μs, which means that the collision news reaches station A 8 μs after the whole frame is sent and station A has stopped listening to the channel for collision. Station A cannot detect the collision because $T_{fr} < 2 \times Tp$.

c. The collision news reaches station B at time t = 24 + 1 = 25 μs, just two μs after it has started sending its frame. Station B can detect the collision.

P5-39. Assume both stations start transmitting at t = 0 μs. The collision occurs at the middle of the bus at time t = 25 μs. Both stations hear the collision at time t = 50 μs. Station A (using R = 0) senses the medium and finds it free. It retransmits at time t = 50 μs. The frame arrives successfully. Station B (using R = 1) is scheduled to transmit at time t = 50 + 120 = 170 μs. The channel, however, is busy from t = 50 μs to t = 50 + 50 + 120 = 220 μs. This means that when station B senses the channel at t = 170, it finds it busy. It needs to continuously sense the channel. At t = 220 μs, it finds the channel free. This shows the benefit of creating a random number to make the stations schedule at different times and avoid the collision. (See the figure on the next page.)



P5-40. We calculate the probability in each case:

a. After the first collision (k = 1), R has the range (0, 1). There are four possibilities (00, 01, 10, and 11), in which 00 means that both station have come up with R = 0, and so on. In two of these four possibilities (00 or 11), a collision may occur. Therefore the probability of collision is 2/4 or 50 percent.

b. After the second collision (k = 2), R has the range (0, 1, 2, 3). There are sixteen possibilities (00, 01, 02, 03, 10, 11, …, 33). In four of these sixteen possibilities (00, 11, 22, 33), a collision may occur. Therefore the probability of collision is 4/16 or 25 percent.

P5-41. The probability of a free slot is the probability that a frame, generated from any station, is successfully transmitted. We discussed this in previous problems to be $Np(1 - p)^{N-1}$.

a. The probability of getting a free slot is then $P_{free} = Np(1 - p)^{N-1}$.

b. As we discussed in previous problems, the maximum occurs when $p = 1/N$ and the maximum of P_{free} is 1/e.

c. The probability that the jth slot is free is the probability that previous $(j - 1)$ slots were not free and the next one is free. $P_{jth} = j\ P_{free}\ (1 - P_{free})$.

d. The average number of slots that need to be passed is the average of P_{jth} when j is between 0 and infinity: $n = \Sigma\ P_{jth}$. Since P_{jth} is less than one, the series converges and the result is $n = 1/P_{free}$. The result is somewhat intuitive because, if the probability of the success for an event is P, the average number of times that the event should be repeated before getting a successful result is 1/P.

e. Since P_{free} = 1/e when *N* is a very large number, the value *n* = e in this case. In other words, a station needs to wait 2.7182 slots before being able to send a frame.

P5-42. We use the definition to find the throughput as S = 1 / (1 + 6.4*a*).

$$\mathbf{S = (T_{fr}) / (channel\ is\ occupied\ for\ a\ frame)}$$

$$\mathbf{S = (T_{fr}) / (\mathit{k} \times 2 \times T_p + T_{fr} + T_p)}$$

$$\mathbf{S = 1 / [2e\ (T_p) / (T_{fr}) + (T_{fr}) / (T_{fr}) + T_p / (T_{fr})]}$$

$$\mathbf{S = 1 / [2e\mathit{a} + 1 + \mathit{a}] = 1 / [1 + (2e + 1)\mathit{a}] = 1 / (1 + 6.4\mathit{a})}$$

P5-43. For the sender to detect the collision, the last bit of the frame should not have left the station. This means that the transmission delay (T_{fr}) needs to be greater than 40 μs (20 μs + 20 μs) or the frame length should be at least 10 Mbps × 40 μs = 400 bits.

P5-44. The propagation delay for this network is T_p = (2000 m) /(2 × 10^8 m/s) = 10 μs. The first bit of station A's frame reaches station B at (t_1 + 10 μs).

a. Station B has not received the first bit of A's frame at (t_1 + 10 μs). It senses the medium and finds it free. It starts sending its frame, which results in a collision.

b. At time (t_1 + 11 μs), station B has already received the first bit of station A's frame. It knows that the medium is busy and refrains from sending.

P5-45. The first byte in binary is 0000011<u>1</u>. The least significant bit is 1. This means that the pattern defines a multicast address.

P5-46.

a. Station A is successful if A is sending, but B is not sending. In other words

$$\mathbf{P_A = \mathit{p}_1 \times (1 - \mathit{p}_2) = 0.3 \times (1 - 0.4) = 0.18}$$

b. Station B is successful if B is sending, but A is not sending. In other words

$$\mathbf{P_B = \mathit{p}_2 \times (1 - \mathit{p}_1) = 0.4 \times (1 - 0.3) = 0.28}$$

c. The probability that a frame is transmitted, from A or B is the sum of the probabilities:

$$\mathbf{P = P_A + P_B = 0.18 + 0.28 = 0.46}$$

This means that almost half of the transmitted frames reach their destination.

P5-47. The maximum efficiency in a pure Aloha network is 0.184.

S_{max} = **0.184** × **10 Mbps** = **1,840,000 bps**

Maximum number of frames per second = **1,840,000 / 1000** = **1840**

P5-48. Let us find the relationship between the minimum frame size and the data rate. We have

T_{fr} = **(frame size) / (data rate)** → **(frame size)** = T_{fr} × **(data rate)**

Since T_{fr} = [2 × (distance) / (propagation speed)], this means that T_{fr} is constant if the distance is constant (propagation speed is almost constant). Therefore, (frame size) = K × (data rate). We calculate the minimum frame size based on the above proportionality relationship.

a. **Data rate: 100 Mbps** → **minimum frame size = 5120 bits**

b. **Data rate: 1 Gbps** → **minimum frame size = 51,200 bits**

c. **Data rate:10 Gbps** → **minimum frame size = 512,000 bits**

P5-49. We interpret each four-bit pattern as a hexadecimal digit. We then group the hexadecimal digits with a colon between the pairs:

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

P5-50. 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 bits are sent without gaps. The arrow shows the direction of movement.

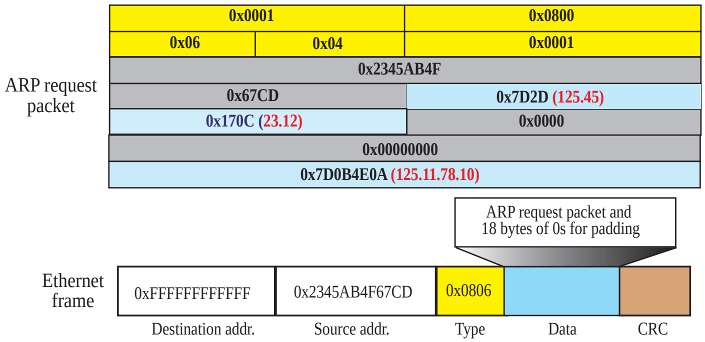
Bytes: `00011010 00101011 00111100 01001101 01011110 01101111`

← `01011000 11010100 00111100 10110010 01111010 11110110`

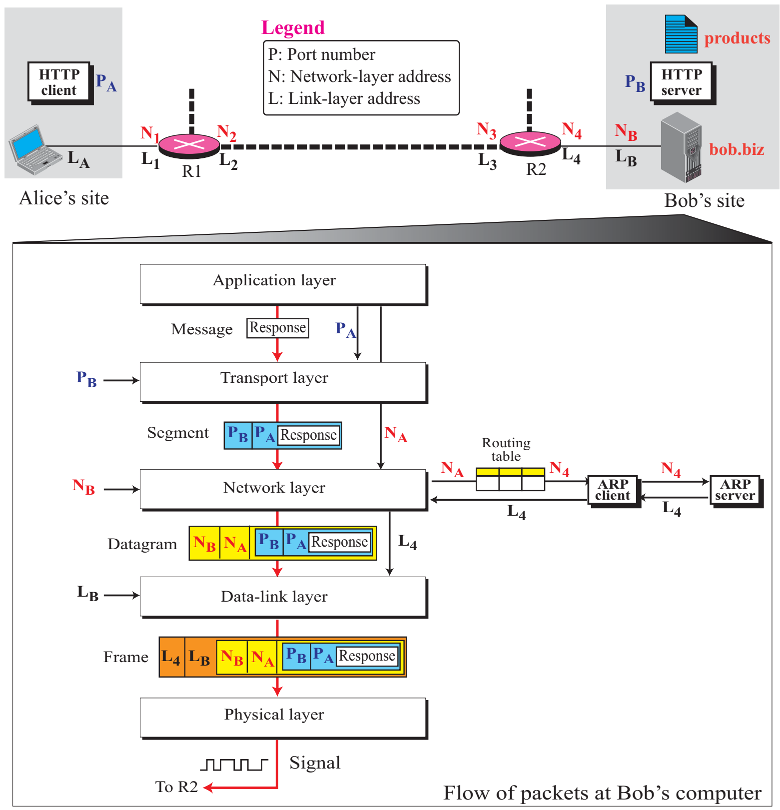
P5-51. We can calculate the propagation time as t = (2500 m) / (200,000,000 m/s) = 12.5 μs. To get the total delay, we need to add propagation delay in the equipment (10 μs). This results in T = 22.5 μs.

P5-52. The DSL technology is based on star topology with the hub at the telephone office. The local loop connects each customer to the end office. This means that there is no sharing; the allocated bandwidth for each customer is not shared with neighbors. The data rate does not depend on how many people in the area are transferring data at the same time.

P5-53. The following shows the ARP request packet and its encapsulation in an Ethernet frame (without the preamble and SFD fields for simplicity). Note that all values are in hexadecimal. We have also shown the IP addresses in dotted-decimal notation.



P5-54. We can repeat Figure 5.51 as shown below.



However, we need to remember that there is no need for calling DNS because Bob's computer can extract the IP address and the port number of the request received from Bob.

P5-55. The minimum data size in the Standard Ethernet is 46 bytes. Therefore, we need to add 4 bytes of padding to the data (46 – 42 = 4)

P5-56. The smallest Ethernet frame is 64 bytes and carries 46 bytes of data (and possible padding). The ratio is (data size) / (frame size) in percent. The Ratio is 71.9 percent.

P5-57. A filtering table is based on link-layer addresses; a forwarding table is based on the network-layer addresses.