EXERCISES

1. Show the NRZ, Manchester, and NRZI encodings for the bit pattern shown in Figure 2.36. Assume that the NRZI signal starts out low.

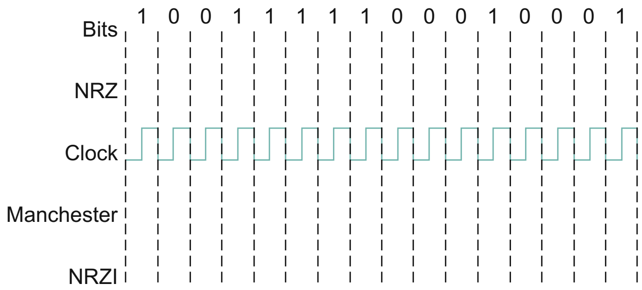

FIGURE 2.36 Diagram for Exercise 1.

2. Show the 4B/5B encoding, and the resulting NRZI signal, for the following bit sequence:

1110 0101 0000 0011

3. Show the 4B/5B encoding, and the resulting NRZI signal, for the following bit sequence:

1101 1110 1010 1101 1011 1110 1110 1111

4. In the 4B/5B encoding (Table 2.2), only two of the 5-bit codes used end in two 0s. How many possible 5-bit sequences are there (used by the existing code or not) that meet the stronger restriction of having at most one leading and at most one trailing 0? Could all 4-bit sequences be mapped to such 5-bit sequences?

5. Assuming a framing protocol that uses bit stuffing, show the bit sequence transmitted over the link when the frame contains the following bit sequence:

110101111101011111101011111110

Mark the stuffed bits.

6. Suppose the following sequence of bits arrives over a link:

   1101011111010111110010111110110

   Show the resulting frame after any stuffed bits have been removed. Indicate any errors that might have been introduced into the frame.

7. Suppose the following sequence of bits arrives over a link:

   01101011111010100111111011001111110

   Show the resulting frame after any stuffed bits have been removed. Indicate any errors that might have been introduced into the frame.

8. Suppose you want to send some data using the BISYNC framing protocol and the last 2 bytes of your data are DLE and ETX. What sequence of bytes would be transmitted immediately prior to the CRC?

9. For each of the following framing protocols, give an example of a byte/bit sequence that should never appear in a transmission:
   (a) BISYNC
   (b) HDLC

10. Assume that a SONET receiver resynchronizes its clock whenever a 1 bit appears; otherwise, the receiver samples the signal in the middle of what it believes is the bit's time slot.
    (a) What relative accuracy of the sender's and receiver's clocks is required in order to receive correctly 48 zero bytes (one ATM cell's worth) in a row?
    (b) Consider a forwarding station A on a SONET STS-1 line, receiving frames from the downstream end B and retransmitting them upstream. What relative accuracy of A's and B's clocks is required to keep A from accumulating more than one extra frame per minute?

11. Show that two-dimensional parity allows detection of all 3-bit errors.

12. Give an example of a 4-bit error that would not be detected by two-dimensional parity, as illustrated in Figure 2.14. What is the general set of circumstances under which 4-bit errors will be undetected?

13. Show that two-dimensional parity provides the receiver enough information to correct any 1-bit error (assuming the receiver knows only 1 bit is bad), but not any 2-bit error.

14. Show that the Internet checksum will never be 0xFFFF (that is, the final value of `sum` will not be 0x0000) unless every byte in the buffer is 0. (Internet specifications in fact require that a checksum of 0x0000 be transmitted as 0xFFFF; the value 0x0000 is then reserved for an omitted checksum. Note that, in ones complement arithmetic, 0x0000 and 0xFFFF are both representations of the number 0.)

15. Prove that the Internet checksum computation shown in the text is independent of byte order (host order or network order) except that the bytes in the final checksum should be swapped later to be in the correct order. Specifically, show that the sum of 16-bit words can be computed in either byte order. For example, if the one's complement sum (denoted by $+'$) of 16-bit words is represented as follows,

$$[\mathrm{A,B}] +' [\mathrm{C,D}] +' \cdots +' [\mathrm{Y,Z}]$$

the following swapped sum is the same as the original sum above:

$$[\mathrm{B,A}] +' [\mathrm{D,C}] +' \cdots +' [\mathrm{Z,Y}]$$

16. Suppose that one byte in a buffer covered by the Internet checksum algorithm needs to be decremented (e.g., a header hop count field). Give an algorithm to compute the revised checksum without rescanning the entire buffer. Your algorithm should consider whether the byte in question is low order or high order.

☆ **17.** Show that the Internet checksum can be computed by first taking the 32-bit ones complement sum of the buffer in 32-bit units, then taking the 16-bit ones complement sum of the upper and lower halfwords, and finishing as before by complementing the result. (To take a 32-bit ones complement sum on 32-bit twos complement hardware, you need access to the "overflow" bit.)

18. Suppose we want to transmit the message `11100011` and protect it from errors using the CRC polynomial $x^3 + 1$.

   (a) Use polynomial long division to determine the message that should be transmitted.

(b) Suppose the leftmost bit of the message is inverted due to noise on the transmission link. What is the result of the receiver's CRC calculation? How does the receiver know that an error has occurred?

19. Suppose we want to transmit the message 1011 0010 0100 1011 and protect it from errors using the CRC8 polynomial $x^8 + x^2 + x^1 + 1$.

(a) Use polynomial long division to determine the message that should be transmitted.

(b) Suppose the leftmost bit of the message is inverted due to noise on the transmission link. What is the result of the receiver's CRC calculation? How does the receiver know that an error has occurred?

20. The CRC algorithm as presented in this chapter requires lots of bit manipulations. It is, however, possible to do polynomial long division taking multiple bits at a time, via a table-driven method, that enables efficient software implementations of CRC. We outline the strategy here for long division 3 bits at a time (see Table 2.5); in practice, we would divide 8 bits at a time, and the table would have 256 entries.

Let the divisor polynomial $C = C(x)$ be $x^3 + x^2 + 1$, or 1101. To build the table for C, we take each 3-bit sequence, p, append three trailing 0s, and then find the quotient $q = p\frown 000 \div C$,

Table 2.5 Table-Driven CRC Calculation

p	$q = p\frown 000 \div C$	$C \times q$
000	000	000 000
001	001	001 101
010	011	010 ____
011	0___	011 ____
100	111	100 011
101	110	101 110
110	100	110 ____
111	____	111 ____

ignoring the remainder. The third column is the product $C \times q$, the first 3 bits of which should equal p.

(a) Verify, for $p = 110$, that the quotients $p^\frown 000 \div C$ and $p^\frown 111 \div C$ are the same; that is, it doesn't matter what the trailing bits are.

(b) Fill in the missing entries in the table.

(c) Use the table to divide 101 001 011 001 100 by C. Hint: The first 3 bits of the dividend are $p = 101$, so from the table the corresponding first 3 bits of the quotient are $q = 110$. Write the 110 above the second 3 bits of the dividend, and subtract $C \times q = 101\ 110$, again from the table, from the first 6 bits of the dividend. Keep going in groups of 3 bits. There should be no remainder.

☆ **21.** With 1 parity bit we can detect all 1-bit errors. Show that at least one generalization fails, as follows:

(a) Show that if messages m are 8 bits long, then there is no error detection code $e = e(m)$ of size 2 bits that can detect all 2-bit errors. Hint: Consider the set M of all 8-bit messages with a single 1 bit; note that any message from M can be transmuted into any other with a 2-bit error, and show that some pair of messages m_1 and m_2 in M must have the same error code e.

(b) Find an N (not necessarily minimal) such that no 32-bit error detection code applied to N-bit blocks can detect all errors altering up to 8 bits.

22. Consider an ARQ protocol that uses only negative acknowledgments (NAKs), but no positive acknowledgments (ACKs). Describe what timeouts would have to be scheduled. Explain why an ACK-based protocol is usually preferred to a NAK-based protocol.

23. Consider an ARQ algorithm running over a 40-km point-to-point fiber link.

(a) Compute the one-way propagation delay for this link, assuming that the speed of light is 2×10^8 m/s in the fiber.

(b) Suggest a suitable timeout value for the ARQ algorithm to use.

(c) Why might it still be possible for the ARQ algorithm to time out and retransmit a frame, given this timeout value?

24. Suppose you are designing a sliding window protocol for a 1-Mbps point-to-point link to the moon, which has a one-way latency of 1.25 seconds. Assuming that each frame carries 1 KB of data, what is the minimum number of bits you need for the sequence number?

25. Suppose you are designing a sliding window protocol for a 1-Mbps point-to-point link to the stationary satellite revolving around the Earth at an altitude of 3×10^4 km. Assuming that each frame carries 1 KB of data, what is the minimum number of bits you need for the sequence number in the following cases? Assume the speed of light is 3×10^8 m/s.
   (a) RWS=1
   (b) RWS=SWS

26. The text suggests that the sliding window protocol can be used to implement flow control. We can imagine doing this by having the receiver delay ACKs, that is, not send the ACK until there is free buffer space to hold the next frame. In doing so, each ACK would simultaneously acknowledge the receipt of the last frame and tell the source that there is now free buffer space available to hold the next frame. Explain why implementing flow control in this way is not a good idea.

27. Implicit in the stop-and-wait scenarios of Figure 2.17 is the notion that the receiver will retransmit its ACK immediately on receipt of the duplicate data frame. Suppose instead that the receiver keeps its own timer and retransmits its ACK only after the next expected frame has not arrived within the timeout interval. Draw timelines illustrating the scenarios in Figure 2.17(b) to (d); assume the receiver's timeout value is twice the sender's. Also redraw (c) assuming the receiver's timeout value is half the sender's.

28. In stop-and-wait transmission, suppose that both sender and receiver retransmit their last frame immediately on receipt of a duplicate ACK or data frame; such a strategy is superficially reasonable because receipt of such a duplicate is most likely to mean the other side has experienced a timeout.
   (a) Draw a timeline showing what will happen if the first data frame is somehow duplicated, but no frame is lost. How long

will the duplications continue? This situation is known as the Sorcerer's Apprentice bug.

(b) Suppose that, like data, ACKs are retransmitted if there is no response within the timeout period. Suppose also that both sides use the same timeout interval. Identify a reasonably likely scenario for triggering the Sorcerer's Apprentice bug.

29. Give some details of how you might augment the sliding window protocol with flow control by having ACKs carry additional information that reduces the SWS as the receiver runs out of buffer space. Illustrate your protocol with a timeline for a transmission; assume the initial SWS and RWS are 4, the link speed is instantaneous, and the receiver can free buffers at the rate of one per second (i.e., the receiver is the bottleneck). Show what happens at $T = 0, T = 1, \ldots, T = 4$ seconds.

30. Describe a protocol combining the sliding window algorithm with selective ACKs. Your protocol should retransmit promptly, but not if a frame simply arrives one or two positions out of order. Your protocol should also make explicit what happens if several consecutive frames are lost.

31. Draw a timeline diagram for the sliding window algorithm with SWS $=$ RWS $= 3$ frames, for the following two situations. Use a timeout interval of about $2 \times$ RTT.

(a) Frame 4 is lost.

(b) Frames 4 to 6 are lost.

32. Draw a timeline diagram for the sliding window algorithm with SWS $=$ RWS $= 4$ frames in the following two situations. Assume the receiver sends a duplicate acknowledgment if it does not receive the expected frame. For example, it sends DUPACK[2] when it expects to see Frame[2] but receives Frame[3] instead. Also, the receiver sends a cumulative acknowledgment after it receives all the outstanding frames. For example, it sends ACK[5] when it receives the lost frame Frame[2] after it already received Frame[3], Frame[4], and Frame[5]. Use a timeout interval of about $2 \times$ RTT.

(a) Frame 2 is lost. Retransmission takes place upon timeout (as usual).

(b) Frame 2 is lost. Retransmission takes place either upon receipt of the first DUPACK or upon timeout. Does this scheme reduce the transaction time? (Note that some end-to-end protocols, such as variants of TCP, use similar schemes for fast retransmission.)

33. Suppose that we attempt to run the sliding window algorithm with SWS $= $ RWS $= 3$ and with MaxSeqNum $= 5$. The Nth packet DATA[N] thus actually contains N mod 5 in its sequence number field. Give an example in which the algorithm becomes confused; that is, a scenario in which the receiver expects DATA[5] and accepts DATA[0]—which has the same transmitted sequence number—in its stead. No packets may arrive out of order. Note that this implies MaxSeqNum ≥ 6 is necessary as well as sufficient.

34. Consider the sliding window algorithm with SWS $=$ RWS $= 3$, with no out-of-order arrivals and with infinite-precision sequence numbers.

(a) Show that if DATA[6] is in the receive window, then DATA[0] (or in general any older data) cannot arrive at the receiver (and hence that MaxSeqNum $= 6$ would have sufficed).

(b) Show that if ACK[6] may be sent (or, more literally, that DATA[5] is in the sending window), then ACK[2] (or earlier) cannot be received.

These amount to a proof of the formula given in Section 2.5.2, particularized to the case SWS $= 3$. Note that part (b) implies that the scenario of the previous problem cannot be reversed to involve a failure to distinguish ACK[0] and ACK[5].

35. Suppose that we run the sliding window algorithm with SWS $= 5$ and RWS $= 3$, and no out-of-order arrivals.

(a) Find the smallest value for MaxSeqNum. You may assume that it suffices to find the smallest MaxSeqNum such that if DATA[MaxSeqNum] is in the receive window, then DATA[0] can no longer arrive.

(b) Give an example showing that MaxSeqNum $- 1$ is not sufficient.

(c) State a general rule for the minimum MaxSeqNum in terms of SWS and RWS.

FIGURE 2.37 Diagram for Exercises 36 to 38.

36. Suppose A is connected to B via an intermediate router R, as shown in Figure 2.37. The A–R and R–B links each accept and transmit only one packet per second in each direction (so two packets take 2 seconds), and the two directions transmit independently. Assume A sends to B using the sliding window protocol with SWS $= 4$.

(a) For Time $= 0, 1, 2, 3, 4, 5$, state what packets arrive at and leave each node, or label them on a timeline.

(b) What happens if the links have a propagation delay of 1.0 second, but accept immediately as many packets as are offered (i.e., latency $= 1$ second but bandwidth is infinite)?

37. Suppose A is connected to B via an intermediate router R, as in the previous problem. The A–R link is instantaneous, but the R–B link transmits only one packet each second, one at a time (so two packets take 2 seconds). Assume A sends to B using the sliding window protocol with SWS $= 4$. For Time $= 0, 1, 2, 3, 4$, state what packets arrive at and are sent from A and B. How large does the queue at R grow?

38. Consider the situation in the previous exercise, except this time assume that the router has a queue size of 1; that is, it can hold one packet in addition to the one it is sending (in each direction). Let A's timeout be 5 seconds, and let SWS again be 4. Show what happens at each second from Time $= 0$ until all four packets from the first window-full are successfully delivered.

39. What kind of problems can arise when two hosts on the same Ethernet share the same hardware address? Describe what happens and why that behavior is a problem.

40. The 1982 Ethernet specification allowed between any two stations up to 1500 m of coaxial cable, 1000 m of other point-to-point link cable, and two repeaters. Each station or repeater connects to the coaxial cable via up to 50 m of "drop

Table 2.6 Typical Delays Associated with Various Devices (Exercise 40)

Item	Delay
Coaxial cable	Propagation speed .77c
Link/drop cable	Propagation speed .65c
Repeaters	Approximately 0.6 μs each
Transceivers	Approximately 0.2 μs each

cable." Typical delays associated with each device are given in Table 2.6 (where $c =$ speed of light in a vacuum $= 3 \times 10^8$ m/s). What is the worst-case round-trip propagation delay, measured in bits, due to the sources listed? (This list is not complete; other sources of delay include sense time and signal rise time.)

☆ **41.** Coaxial cable Ethernet was limited to a maximum of 500 m between repeaters, which regenerate the signal to 100% of its original amplitude. Along one 500-m segment, the signal could decay to no less than 14% of its original value (8.5 dB). Along 1500 m, then, the decay might be $(0.14)^3 = 0.3\%$. Such a signal, even along 2500 m, is still strong enough to be read; why then are repeaters required every 500 m?

42. Suppose the round-trip propagation delay for Ethernet is 46.4 μs. This yields a minimum packet size of 512 bits (464 bits corresponding to propagation delay $+ 48$ bits of jam signal).

(a) What happens to the minimum packet size if the delay time is held constant, and the signalling rate rises to 100 Mbps?

(b) What are the drawbacks to so large a minimum packet size?

(c) If compatibility were not an issue, how might the specifications be written so as to permit a smaller minimum packet size?

☆ **43.** Let A and B be two stations attempting to transmit on an Ethernet. Each has a steady queue of frames ready to send; A's frames will be numbered A_1, A_2, and so on, and B's similarly. Let $T = 51.2$ μs be the exponential backoff base unit.

Suppose A and B simultaneously attempt to send frame 1, collide, and happen to choose backoff times of $0 \times T$ and $1 \times T$, respectively, meaning A wins the race and transmits A_1 while B waits. At the end of this transmission, B will attempt to retransmit B_1 while A will attempt to transmit A_2. These first attempts will collide, but now A backs off for either $0 \times T$ or $1 \times T$, while B backs off for time equal to one of $0 \times T, \ldots, 3 \times T$.

(a) Give the probability that A wins this second backoff race immediately after this first collision; that is, A's first choice of backoff time $k \times 51.2$ is less than B's.

(b) Suppose A wins this second backoff race. A transmits A_3, and when it is finished, A and B collide again as A tries to transmit A_4 and B tries once more to transmit B_1. Give the probability that A wins this third backoff race immediately after the first collision.

(c) Give a reasonable lower bound for the probability that A wins all the remaining backoff races.

(d) What then happens to the frame B_1?

This scenario is known as the Ethernet *capture effect*.

44. Suppose the Ethernet transmission algorithm is modified as follows: After each successful transmission attempt, a host waits one or two slot times before attempting to transmit again, and otherwise backs off the usual way.

(a) Explain why the capture effect of the previous exercise is now much less likely.

(b) Show how the strategy above can now lead to a pair of hosts capturing the Ethernet, alternating transmissions, and locking out a third.

(c) Propose an alternative approach, for example, by modifying the exponential backoff. What aspects of a station's history might be used as parameters to the modified backoff?

45. Ethernets use Manchester encoding. Assuming that hosts sharing the Ethernet are not perfectly synchronized, why does this allow collisions to be detected soon after they occur, without waiting for the CRC at the end of the packet?

46. Suppose A, B, and C all make their first carrier sense, as part of an attempt to transmit, while a fourth station D is transmitting. Draw a timeline showing one possible sequence of transmissions, attempts, collisions, and exponential backoff choices. Your timeline should also meet the following criteria: (i) initial transmission attempts should be in the order A, B, C but successful transmissions should be in the order C, B, A, and (ii) there should be at least four collisions.

47. Repeat the previous exercise, now with the assumption that Ethernet is p-persistent with $p = 0.33$ (that is, a waiting station transmits immediately with probability p when the line goes idle and otherwise defers one 51.2-μs slot time and repeats the process). Your timeline should meet criterion (i) of the previous problem, but in lieu of criterion (ii) you should show at least one collision and at least one run of four deferrals on an idle line. Again, note that many solutions are possible.

48. Suppose Ethernet physical addresses are chosen at random (using true random bits).

(a) What is the probability that on a 1024-host network, two addresses will be the same?

(b) What is the probability that the above event will occur on one or more of 2^{20} networks?

(c) What is the probability that, of the 2^{30} hosts in all the networks of (b), some pair has the same address?

Hint: The calculation for (a) and (c) is a variant of that used in solving the so-called Birthday Problem: Given N people, what is the probability that two of their birthdays (addresses) will be the same? The second person has probability $1 - \frac{1}{365}$ of having a different birthday from the first, the third has probability $1 - \frac{2}{365}$ of having a different birthday from the first two, and so on. The probability that all birthdays are different is thus

$$\left(1 - \frac{1}{365}\right) \times \left(1 - \frac{2}{365}\right) \times \cdots \times \left(1 - \frac{N-1}{365}\right)$$

which for smallish N is about

$$1 - \frac{1 + 2 + \cdots + (N-1)}{365}$$

49. Suppose five stations are waiting for another packet to finish on an Ethernet. All transmit at once when the packet is finished and collide.

(a) Simulate this situation up until the point when one of the five waiting stations succeeds. Use coin flips or some other genuine random source to determine backoff times. Make the following simplifications: Ignore inter-frame spacing, ignore variability in collision times (so that retransmission is always after an exact integral multiple of the 51.2-μs slot time), and assume that each collision uses up exactly one slot time.

(b) Discuss the effect of the listed simplifications in your simulation versus the behavior you might encounter on a real Ethernet.

50. Write a program to implement the simulation discussed above, this time with N stations waiting to transmit. Again, model time as an integer, T, in units of slot times, and again treat collisions as taking one slot time (so a collision at time T followed by a backoff of $k = 0$ would result in a retransmission attempt at time $\mathsf{T} + 1$). Find the average delay before *one* station transmits successfully, for $N = 20$, $N = 40$, and $N = 100$. Does your data support the notion that the delay is linear in N? Hint: For each station, keep track of that station's NextTimeToSend and CollisionCount. You are done when you reach a time T for which there is only one station with NextTimeToSend == T. If there is no such station, increment T. If there are two or more, schedule the retransmissions and try again.

51. Suppose that N Ethernet stations, all trying to send at the same time, require $N/2$ slot times to sort out who transmits next. Assuming the average packet size is 5 slot times, express the available bandwidth as a function of N.

52. Consider the following Ethernet model. Transmission attempts are at random times with an average spacing of λ slot times; specifically, the interval between consecutive attempts is an exponential random variable $x = -\lambda \log u$, where u is chosen randomly in the interval $0 \leq u \leq 1$. An attempt at time t results in a collision if there is another attempt in the range from $t - 1$ to $t + 1$, where t is measured in units of the 51.2-μs slot time; otherwise, the attempt succeeds.

(a) Write a program to simulate, for a given value of λ, the average number of slot times needed before a successful transmission, called the *contention interval*. Find the minimum value of the contention interval. Note that you will have to find one attempt past the one that succeeds in order to determine if there was a collision. Ignore retransmissions, which probably do not fit the random model above.

(b) The Ethernet alternates between contention intervals and successful transmissions. Suppose the average successful transmission lasts 8 slot times (512 bytes). Using your minimum length of the contention interval from above, what fraction of the theoretical 10-Mbps bandwidth is available for transmissions?

53. How can a wireless node interfere with the communications of another node when the two nodes are separated by a distance greater than the transmission range of either node?

54. Why is collision detection more complex in wireless networks than in wired networks such as Ethernet?

55. How can hidden terminals be detected in 802.11 networks?

56. Why might a wireless mesh topology be superior to a base station topology for communications in a natural disaster?

57. Why isn't it practical for each node in a sensor net to learn its location by using GPS? Describe a practical alternative.