Chapter 1

6. The answer is in the book.

8. The answer is in the book.

13. 1 Gbps = 10^9 bps, meaning each bit is 10^{-9} sec (1 ns) wide. The length in the wire of such a bit is 1 ns $\times$ 2.3×10^8 m/sec = 0.23 m

16. The answer is in the book.

17. (a) Delay-sensitive; the messages exchanged are short.

    (b) Bandwidth-sensitive, particularly for large files. (Technically this does presume that the underlying protocol uses a large message size or window size; stop-and-wait transmission (as in Section 2.5 of the text) with a small message size would be delay-sensitive.)

    (c) Delay-sensitive; directories are typically of modest size.

    (d) Delay-sensitive; a file's attributes are typically much smaller than the file itself (even on NT filesystems).

19. The answer is in the book.

23. (a) Without compression the total time is 1 MB/$bandwidth$. When we compress the file, the total time is

$$compression\_time + compressed\_size/bandwidth$$

    Equating these and rearranging, we get

$$bandwidth = compression\_size\_reduction/compression\_time$$

    = 0.5 MB/1 sec = 0.5 MB/sec for the first case,
    = 0.6 MB/2 sec = 0.3 MB/sec for the second case.

    (b) Latency doesn't affect the answer because it would affect the compressed and uncompressed transmission equally.

32. (a) In the absence of any packet losses or duplications, when we are expecting the Nth packet we *get* the Nth packet, and so we can keep track of N locally at the receiver.

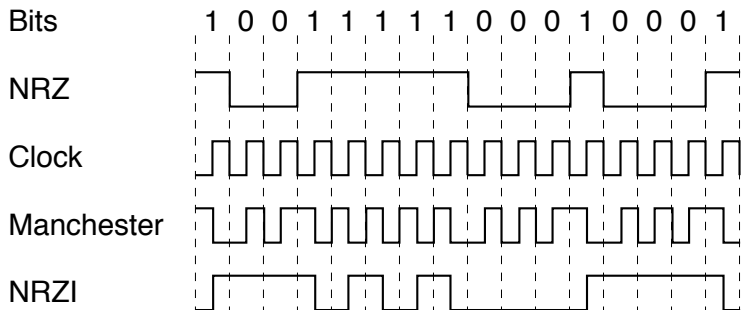
    (b) The scheme outlined here is the stop-and-wait algorithm of Section 2.5; as is indicated there, a header with at least one bit of sequence number is needed (to distinguish between receiving a new packet and a duplication of the previous packet).

    (c) With out-of-order delivery allowed, packets up to 1 minute apart must be distinguishable via sequence number. Otherwise a very old packet might arrive and be accepted as current. Sequence numbers would have to count as high as

$$bandwidth \times 1\,\text{minute}\,/packet\_size$$

Chapter 2

1.



8. ..., DLE, DLE, DLE, ETX, ETX

12. If we flip the bits corresponding to the corners of a rectangle in the 2-D layout of the data, then all parity bits will still be correct. Furthermore, if four bits change and no error is detected, then the bad bits must form a rectangle: in order for the error to go undetected, each row and column must have no errors or exactly two errors.

18. (a) We take the message 11001001, append 000 to it, and divide by 1001. The remainder is 011; what we transmit is the original message with this remainder appended, or 1100 1001 011.

    (b) Inverting the first bit gives 0100 1001 011; dividing by 1001 ($x^3 + 1$) gives a quotient of 0100 0001 and a remainder of 10.

23. (a) Propagation delay = $20 \times 10^3\,\mathrm{m}/(2 \times 10^8\,\mathrm{m/sec})$ = 100 μs.

    (b) The roundtrip time would be about 200 μs. A plausible timeout time would be twice this, or 0.4 ms. Smaller values (but larger than 0.2 ms!) might be reasonable, depending on the amount of variation in actual RTTs. See Section 5.2.5 of the text.

    (c) The propagation-delay calculation does not consider processing delays that may be introduced by the remote node; it may not be able to answer immediately.

24. Bandwidth×(roundtrip)delay is about 125KB/sec × 2.5 sec, or 312 packets. The window size should be this large; the sequence number space must cover twice this range, or up to 624. 10 bits are needed.

35. (a) The smallest working value for MaxSeqNum is 8. It suffices to show that if DATA[8] is in the receive window, then DATA[0] can no longer arrive at the receiver. We have that DATA[8] in receive window
$\Rightarrow$ the earliest possible receive window is DATA[6]..DATA[8]
$\Rightarrow$ ACK[6] has been received
$\Rightarrow$ DATA[5] was delivered.

But because SWS=5, all DATA[0]'s sent were sent before DATA[5]
$\Rightarrow$ by the no-out-of-order arrival hypothesis, DATA[0] can no longer arrive.

(b) We show that if MaxSeqNum=7, then the receiver can be expecting DATA[7] and an old DATA[0] can still arrive. Because 7 and 0 are indistinguishable mod MaxSeqNum, the receiver cannot tell which actually arrived. We follow the strategy of Exercise 27.

1. Sender sends DATA[0]...DATA[4]. All arrive.

2. Receiver sends ACK[5] in response, but it is slow. The receive window is now DATA[5]..DATA[7].

3. Sender times out and retransmits DATA[0]. The receiver accepts it as DATA[7].

(c) MaxSeqNum $\geq$ SWS + RWS.

43. (a) Assuming 48 bits of jam signal was still used, the minimum packet size would be 4640+48 bits = 586 bytes.

(b) This packet size is considerably larger than many higher-level packet sizes, resulting in considerable wasted bandwidth.

(c) The minimum packet size could be smaller if maximum collision domain diameter were reduced, and if sundry other tolerances were tightened up.

46. If the hosts are not perfectly synchronized the preamble of the colliding packet will interrupt clock recovery.

58. It takes a host 82 μs to send a packet. With immediate release, it sends a token upon completion; the earliest it can then transmit again is 200 μs later, when the token has completed a circuit. The station can thus transmit at most 82/282 = 29% of the time, for an effective bandwidth of 29Mbps.

With delayed release, the sender waits 200 μs after beginning the transmission for the beginning of the frame to come around again; at this point the sender sends the token. The token takes another 200 μs to travel around before the original station could transmit again (assuming no other stations transmit). This yields an efficiency of 82/400 = 20%.

15. When A sends to C, all bridges see the packet and learn where A is. However, when C then sends to A, the packet is routed directly to A and B4 does not learn where C is. Similarly, when D sends to C, the packet is routed by B2 towards B3 only, and B1 does not learn where D is.

B1:	A-interface:	A	B2-interface:	C (not D)		
B2:	B1-interface:	A	B3-interface:	C	B4-interface:	D
B3:	B2-interface:	A,D	C-interface:	C		
B4:	B2-interface:	A (not C)	D-interface:	D		

19. (a) The packet will circle endlessly, in both the M→B2→L→B1 and M→B1→L→B2 directions.

(b) Initially we (potentially) have four packets: one from M clockwise, one from M counterclockwise, and a similar pair from L.

Suppose a packet from L arrives at an interface to a bridge Bi, followed immediately via the same interface by a packet from M. As the first packet arrives, the bridge adds ⟨L,arrival-interface⟩ to the table (or, more likely, updates an existing entry for L). When the second packet arrives, addressed to L, the bridge then decides not to forward it, because it arrived from the interface recorded in the table as pointing towards the destination, and so it dies.

Because of this, we expect that in the long run only one of the pair of packets traveling in the same direction will survive. We may end up with two from M, two from L, or one from M and one from L. A specific scenario for the latter is as follows, where the bridges' interfaces are denoted "top" and "bottom":

1. L sends to B1 and B2; both place ⟨L,top⟩ in their table. B1 already has the packet from M in the queue for the top interface; B2 this packet in the queue for the bottom.

2. B1 sends the packet from M to B2 via the top interface. Since the destination is L and ⟨L,top⟩ is in B2's table, it is dropped.

3. B2 sends the packet from M to B1 via the bottom interface, so B1 updates its table entry for M to ⟨M,bottom⟩

4. B2 sends the packet from L to B1 via the bottom interface, causing it to be dropped.

The packet from M now circulates counterclockwise, while the packet from L circulates clockwise.

Chapter 4

3. All 0's or 1's over the entire packet will change the *Version* and *HLen* fields, resulting in non-IPv4 packets.

10. IPv4 effectively requires that, if reassembly is to be done at the downstream router, then it be done at the link layer, and will be transparent to IPv4. IP-layer fragmentation is only done when such link-layer fragmentation isn't practical, in which case IP-layer reassembly might be expected to be even less practical, given how busy routers tend to be. See RFC791, page 23.

    IPv6 uses link-layer fragmentation exclusively; experience had by then established reasonable MTU values, and also illuminated the performance problems of IPv4-style fragmentation. (TCP path-MTU discovery is also mandatory, which means the sender always knows just how large TCP segments can be to avoid fragmentation.)

    Whether or not link-layer fragmentation is feasible appears to depend on the nature of the link; neither version of IP therefore requires it.

12. The answer is no in practice, but yes in theory. MAC address is statically assigned to each hardware. ARP mapping enables indirection from IP addresses to the hardware MAC addresses. This allows IP addresses to be dynamically reallocated when the hardware moves to the different network. So using MAC addresses as IP addresses would mean that we would have to use static IP addresses.

    Since the Internet routing takes advantage of address space hierarchy (use higher bits for network addresses and lower bits for host addresses), if we would have to use static IP addresses, the routing would be much less efficient. Therefore this design is practically not feasible.

13. After B broadcasts any ARP query, all stations that had been sending to A's physical address will switch to sending to B's. A will see a sudden halt to all arriving traffic. (To guard against this, A might monitor for ARP broadcasts purportedly coming from itself; A might even immediately follow such broadcasts with its own ARP broadcast in order to return its traffic to itself. It is not clear, however, how often this is done.)

    If B uses self-ARP on startup, it will receive a reply indicating that its IP address is already in use, which is a clear indication that B should not continue on the network until the issue is resolved.

22. The answer is in the book.