COMPUTER NETWORKS

FOURTH EDITION

PROBLEM SOLUTIONS

ANDREW S. TANENBAUM

Vrije Universiteit
Amsterdam, The Netherlands

PRENTICE HALL PTR

UPPER SADDLE RIVER, NJ 07458

Publishing as Prentice Hall PTR
Upper Saddle River, New Jersey 07458

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

ISBN 0-13-046002-8

Pearson Education LTD.
Pearson Education Australia PTY, Limited
Pearson Education Singapore, Pte. Ltd.
Pearson Education North Asia Ltd.
Pearson Education Canada, Ltd.
Pearson Educación de Mexico, S.A. de C.V.
Pearson Education — Japan
Pearson Education Malaysia, Pte. Ltd.

SOLUTIONS TO CHAPTER 1 PROBLEMS

1. The dog can carry 21 gigabytes, or 168 gigabits. A speed of 18 km/hour equals 0.005 km/sec. The time to travel distance x km is $x/0.005 = 200x$ sec, yielding a data rate of $168/200x$ Gbps or $840/x$ Mbps. For $x < 5.6$ km, the dog has a higher rate than the communication line.

2. The LAN model can be grown incrementally. If the LAN is just a long cable. it cannot be brought down by a single failure (if the servers are replicated) It is probably cheaper. It provides more computing power and better interactive interfaces.

3. A transcontinental fiber link might have many gigabits/sec of bandwidth, but the latency will also be high due to the speed of light propagation over thousands of kilometers. In contrast, a 56-kbps modem calling a computer in the same building has low bandwidth and low latency.

4. A uniform delivery time is needed for voice, so the amount of jitter in the network is important. This could be expressed as the standard deviation of the delivery time. Having short delay but large variability is actually worse than a somewhat longer delay and low variability.

5. No. The speed of propagation is 200,000 km/sec or 200 meters/μsec. In 10 μsec the signal travels 2 km. Thus, each switch adds the equivalent of 2 km of extra cable. If the client and server are separated by 5000 km, traversing even 50 switches adds only 100 km to the total path, which is only 2%. Thus, switching delay is not a major factor under these circumstances.

6. The request has to go up and down, and the response has to go up and down. The total path length traversed is thus 160,000 km. The speed of light in air and vacuum is 300,000 km/sec, so the propagation delay alone is 160,000/300,000 sec or about 533 msec.

7. There is obviously no single correct answer here, but the following points seem relevant. The present system has a great deal of inertia (checks and balances) built into it. This inertia may serve to keep the legal, economic, and social systems from being turned upside down every time a different party comes to power. Also, many people hold strong opinions on controversial social issues, without really knowing the facts of the matter. Allowing poorly reasoned opinions be to written into law may be undesirable. The potential effects of advertising campaigns by special interest groups of one kind or another also have to be considered. Another major issue is security. A lot of people might worry about some 14-year kid hacking the system and falsifying the results.

8. Call the routers A, B, C, D, and E. There are ten potential lines: AB, AC, AD, AE, BC, BD, BE, CD, CE, and DE. Each of these has four possibilities (three speeds or no line), so the total number of topologies is $4^{10} = 1{,}048{,}576$. At 100 ms each, it takes 104,857.6 sec, or slightly more than 29 hours to inspect them all.

9. The mean router-router path is twice the mean router-root path. Number the levels of the tree with the root as 1 and the deepest level as n. The path from the root to level n requires $n - 1$ hops, and 0.50 of the routers are at this level. The path from the root to level $n - 1$ has 0.25 of the routers and a length of $n - 2$ hops. Hence, the mean path length, l, is given by

$$l = 0.5 \times (n - 1) + 0.25 \times (n - 2) + 0.125 \times (n - 3) + \cdots$$

or

$$l = \sum_{i=1}^{\infty} n\ (0.5)^i - \sum_{i=1}^{\infty} i(0.5)^i$$

This expression reduces to $l = n - 2$. The mean router-router path is thus $2n - 4$.

10. Distinguish $n + 2$ events. Events 1 through n consist of the corresponding host successfully attempting to use the channel, i.e., without a collision. The probability of each of these events is $p(1 - p)^{n-1}$. Event $n + 1$ is an idle channel, with probability $(1 - p)^n$. Event $n + 2$ is a collision. Since these $n + 2$ events are exhaustive, their probabilities must sum to unity. The probability of a collision, which is equal to the fraction of slots wasted, is then just $1 - np(1 - p)^{n-1} - (1 - p)^n$.

11. Among other reasons for using layered protocols, using them leads to breaking up the design problem into smaller, more manageable pieces, and layering means that protocols can be changed without affecting higher or lower ones,

12. No. In the ISO protocol model, physical communication takes place only in the lowest layer, not in every layer.

13. Connection-oriented communication has three phases. In the establishment phase a request is made to set up a connection. Only after this phase has been successfully completed can the data transfer phase be started and data transported. Then comes the release phase. Connectionless communication does not have these phases. It just sends the data.

14. Message and byte streams are different. In a message stream, the network keeps track of message boundaries. In a byte stream, it does not. For example, suppose a process writes 1024 bytes to a connection and then a little later writes another 1024 bytes. The receiver then does a read for 2048 bytes. With a message stream, the receiver will get two messages, of 1024 bytes

each. With a byte stream, the message boundaries do not count and the receiver will get the full 2048 bytes as a single unit. The fact that there were originally two distinct messages is lost.

15. Negotiation has to do with getting both sides to agree on some parameters or values to be used during the communication. Maximum packet size is one example, but there are many others.

16. The service shown is the service offered by layer k to layer $k + 1$. Another service that must be present is below layer k, namely, the service offered to layer k by the underlying layer $k - 1$.

17. The probability, P_k, of a frame requiring exactly k transmissions is the probability of the first $k - 1$ attempts failing, p^{k-1}, times the probability of the k-th transmission succeeding, $(1 - p)$. The mean number of transmission is then just

$$\sum_{k=1}^{\infty} kP_k = \sum_{k=1}^{\infty} k(1-p)p^{k-1} = \frac{1}{1-p}$$

18. (a) Data link layer. (b) Network layer.

19. Frames encapsulate packets. When a packet arrives at the data link layer, the entire thing, header, data, and all, is used as the data field of a frame. The entire packet is put in an envelope (the frame), so to speak (assuming it fits).

20. With n layers and h bytes added per layer, the total number of header bytes per message is hn, so the space wasted on headers is hn. The total message size is $M + nh$, so the fraction of bandwidth wasted on headers is $hn/(M + hn)$.

21. Both models are based on layered protocols. Both have a network, transport, and application layer. In both models, the transport service can provide a reliable end-to-end byte stream. On the other hand, they differ in several ways. The number of layers is different, the TCP/IP does not have session or presentation layers, OSI does not support internetworking, and OSI has both connection-oriented and connectionless service in the network layer.

22. TCP is connection oriented, whereas UDP is a connectionless service.

23. The two nodes in the upper-right corner can be disconnected from the rest by three bombs knocking out the three nodes to which they are connected. The system can withstand the loss of any two nodes.

24. Doubling every 18 months means a factor of four gain in 3 years. In 9 years, the gain is then 4^3 or 64, leading to 6.4 billion hosts. My intuition says that is much too conservative, since by then probably every television in the world and possibly billions of other appliances will be on home LANs connected to

the Internet. The average person in the developed world may have dozens of Internet hosts by then.

25. If the network tends to lose packets, it is better to acknowledge each one separately, so the lost packets can be retransmitted. On the other hand, if the network is highly reliable, sending one acknowledgement at the end of the entire transfer saves bandwidth in the normal case (but requires the entire file to be retransmitted if even a single packet is lost).

26. Small, fixed-length cells can be routed through switches quickly, and completely in hardware. Small, fixed-size cells also make it easier to build hardware that handles many cells in parallel. Also, they do not block transmission lines for very long, making it easier to provide quality-of-service guarantees.

27. The speed of light in coax is about 200,000 km/sec, which is 200 meters/μsec. At 10 Mbps, it takes 0.1 μsec to transmit a bit. Thus, the bit lasts 0.1 μsec in time, during which it propagates 20 meters. Thus, a bit is 20 meters long here.

28. The image is $1024 \times 768 \times 3$ bytes or 2,359,296 bytes. This is 18,874,368 bits. At 56,000 bits/sec, it takes about 337.042 sec. At 1,000,000 bits/sec, it takes about 18.874 sec. At 10,000,000 bits/sec, it takes about 1.887 sec. At 100,000,000 bits/sec, it takes about 0.189 sec.

29. Think about the hidden terminal problem. Imagine a wireless network of five stations, *A* through *E*, such that each one is in range of only its immediate neighbors. Then *A* can talk to *B* at the same time *D* is talking to *E*. Wireless networks have potential parallelism, and in this way differ from Ethernet.

30. One disadvantage is security. Every random delivery man who happens to be in the building can listen in on the network. Another disadvantage is reliability. Wireless networks make lots of errors. A third potential problem is battery life, since most wireless devices tend to be mobile.

31. One advantage is that if everyone uses the standard, everyone can talk to everyone. Another advantage is that widespread use of any standard will give it economies of scale, as with VLSI chips. A disadvantage is that the political compromises necessary to achieve standardization frequently lead to poor standards. Another disadvantage is that once a standard has been widely adopted, it is difficult to change,, even if new and better techniques or methods are discovered. Also, by the time it has been accepted, it may be obsolete.

32. There are many examples, of course. Some systems for which there is international standardization include compact disc players and their discs, Walkman tape players and audio cassettes, cameras and 35mm film, and automated

teller machines and bank cards. Areas where such international standardization is lacking include VCRs and videotapes (NTSC VHS in the U.S., PAL VHS in parts of Europe, SECAM VHS in other countries), portable telephones, lamps and lightbulbs (different voltages in different countries), electrical sockets and appliance plugs (every country does it differently), photocopiers and paper (8.5 x 11 inches in the U.S., A4 everywhere else), nuts and bolts (English versus metric pitch), etc.

SOLUTIONS TO CHAPTER 2 PROBLEMS

1. $a_n = \dfrac{-1}{\pi n}$, $b_n = 0$, $c = 1$.

2. A noiseless channel can carry an arbitrarily large amount of information, no matter how often it is sampled. Just send a lot of data per sample. For the 4 kHz channel, make 8000 samples/sec. If each sample is 16 bits, the channel can send 128 kbps. If each sample is 1024 bits, the channel can send 8.2 Mbps. The key word here is "noiseless." With a normal 4 kHz channel, the Shannon limit would not allow this.

3. Using the Nyquist theorem, we can sample 12 million times/sec. Four-level signals provide 2 bits per sample, for a total data rate of 24 Mbps.

4. A signal-to-noise ratio of 20 dB means $S/N = 100$. Since $\log_2 101$ is about 6.658, the Shannon limit is about 19.975 kbps. The Nyquist limit is 6 kbps. The bottleneck is therefore the Nyquist limit, giving a maximum channel capacity of 6 kbps.

5. To send a T1 signal we need $H\log_2(1 + S/N) = 1.544 \times 10^6$ with $H = 50{,}000$. This yields $S/N = 2^{30} - 1$, which is about 93 dB.

6. A passive star has no electronics. The light from one fiber illuminates a number of others. An active repeater converts the optical signal to an electrical one for further processing.

7. Use $\Delta f = c\Delta\lambda/\lambda^2$ with $\Delta\lambda = 10^{-7}$ meters and $\lambda = 10^{-6}$ meters. This gives a bandwidth (Δf) of 30,000 GHz.

8. The data rate is $480 \times 640 \times 24 \times 60$ bps, which is 442 Mbps. For simplicity, let us assume 1 bps per Hz. From Eq. (2-3) we get $\Delta\lambda = \lambda^2\Delta f/c$. We have $\Delta f = 4.42 \times 10^8$, so $\Delta\lambda = 2.5 \times 10^{-6}$ microns. The range of wavelengths used is very short.

9. The Nyquist theorem is a property of mathematics and has nothing to do with technology. It says that if you have a function whose Fourier spectrum does not contain any sines or cosines above f, then by sampling the function at a

frequency of $2f$ you capture all the information there is. Thus, the Nyquist theorem is true for all media.

10. In the text it was stated that the bandwidths (i.e., the frequency ranges) of the three bands were approximately equal. From the formula $\Delta f = c\Delta\lambda/\lambda^2$, it is clear that to get a constant Δf, the higher the frequency, the larger $\Delta\lambda$ has to be. The x-axis in the figure is λ, so the higher the frequency, the more $\Delta\lambda$ you need. In fact, $\Delta\lambda$ is quadratic in λ. The fact that the bands are approximately equal is an accidental property of the kind of silicon used.

11. Start with $\lambda f = c$. We know that c is 3×10^8 m/s. For $\lambda = 1$ cm, we get 30 GHz. For $\lambda = 5$ m, we get 60 MHz. Thus, the band covered is 60 MHz to 30 GHz.

12. At 1 GHz, the waves are 30 cm long. If one wave travels 15 cm more than the other, they will arrive out of phase. The fact that the link is 50 km long is irrelevant.

13. If the beam is off by 1 mm at the end, it misses the detector. This amounts to a triangle with base 100 m and height 0.001 m. The angle is one whose tangent is thus 0.00001. This angle is about 0.00057 degrees.

14. With 66/6 or 11 satellites per necklace, every 90 minutes 11 satellites pass overhead. This means there is a transit every 491 seconds. Thus, there will be a handoff about every 8 minutes and 11 seconds.

15. The satellite moves from being directly overhead toward the southern horizon, with a maximum excursion from the vertical of 2ϕ. It takes 24 hours to go from directly overhead to maximum excursion and then back.

16. The number of area codes was $8 \times 2 \times 10$, which is 160. The number of prefixes was $8 \times 8 \times 10$, or 640. Thus, the number of end offices was limited to 102,400. This limit is not a problem.

17. With a 10-digit telephone number, there could be 10^{10} numbers, although many of the area codes are illegal, such as 000. However, a much tighter limit is given by the number of end offices. There are 22,000 end offices, each with a maximum of 10,000 lines. This gives a maximum of 220 million telephones. There is simply no place to connect more of them. This could never be achieved in practice because some end offices are not full. An end office in a small town in Wyoming may not have 10,000 customers near it, so those lines are wasted.

18. Each telephone makes 0.5 calls/hour at 6 minutes each. Thus, a telephone occupies a circuit for 3 minutes/hour. Twenty telephones can share a circuit, although having the load be close to 100% ($\rho = 1$ in queueing terms) implies very long wait times). Since 10% of the calls are long distance, it takes 200 telephones to occupy a long-distance circuit full time. The interoffice trunk

has 1,000,000/4000 = 250 circuits multiplexed onto it. With 200 telephones per circuit, an end office can support $200 \times 250 = 50,000$ telephones.

19. The cross-section of each strand of a twisted pair is $\pi/4$ square mm. A 10-km length of this material, with two strands per pair has a volume of $2\pi/4 \times 10^{-2}$ m^3. This volume is about 15,708 cm^3. With a specific gravity of 9.0, each local loop has a mass of 141 kg. The phone company thus owns 1.4×10^9 kg of copper. At 3 dollars each, the copper is worth about 4.2 billion dollars.

20. Like a single railroad track, it is half duplex. Oil can flow in either direction, but not both ways at once.

21. Traditionally, bits have been sent over the line without any error correcting scheme in the physical layer. The presence of a CPU in each modem makes it possible to include an error correcting code in layer 1 to greatly reduce the effective error rate seen by layer 2. The error handling by the modems can be done totally transparently to layer 2. Many modems now have built in error correction.

22. There are four legal values per baud, so the bit rate is twice the baud rate. At 1200 baud, the data rate is 2400 bps.

23. The phase shift is always 0, but two amplitudes are used, so this is straight amplitude modulation.

24. If all the points are equidistant from the origin, they all have the same amplitude, so amplitude modulation is not being used. Frequency modulation is never used in constellation diagrams, so the encoding is pure phase shift keying.

25. Two, one for upstream and one for downstream. The modulation scheme itself just uses amplitude and phase. The frequency is not modulated.

26. There are 256 channels in all, minus 6 for POTS and 2 for control, leaving 248 for data. If 3/4 of these are for downstream, that gives 186 channels for downstream. ADSL modulation is at 4000 baud, so with QAM-64 (6 bits/baud) we have 24,000 bps in each of the 186 channels. The total bandwidth is then 4.464 Mbps downstream.

27. A 5-KB Web page has 40,000 bits. The download time over a 36 Mbps channel is 1.1 msec. If the queueing delay is also 1.1 msec, the total time is 2.2 msec. Over ADSL there is no queueing delay, so the download time at 1 Mbps is 40 msec. At 56 kbps it is 714 msec.

28. There are ten 4000 Hz signals. We need nine guard bands to avoid any interference. The minimum bandwidth required is $4000 \times 10 + 400 \times 9 = 43,600$ Hz.

29. A sampling time of 125 μsec corresponds to 8000 samples per second. According to the Nyquist theorem, this is the sampling frequency needed to capture all the information in a 4 kHz channel, such as a telephone channel. (Actually the nominal bandwidth is somewhat less, but the cutoff is not sharp.)

30. The end users get $7 \times 24 = 168$ of the 193 bits in a frame. The overhead is therefore 25/193 = 13%.

31. In both cases 8000 samples/sec are possible. With dibit encoding, two bits are sent per sample. With T1, 7 bits are sent per period. The respective data rates are 16 kbps and 56 kbps.

32. Ten frames. The probability of some random pattern being 0101010101 (on a digital channel) is 1/1024.

33. A coder accepts an arbitrary analog signal and generates a digital signal from it. A demodulator accepts a modulated sine wave only and generates a digital signal.

34. (a) 64 kbps. (b) 32 kbps. (c) 8 kbps.

35. The signal must go from 0 to A in one quarter of a wave—that is, in a time $T/4$. In order to track the signal, 8 steps must fit into the quarter wave, or 32 samples per full wave. The time per sample is $1/x$ so the full period must be long enough to contain 32 samples—that is, $T > 32/x$ or $f_{\max} = x/32$.

36. A drift rate of 10^{-9} means 1 second in 10^9 seconds or 1 nsec per second. At OC-1 speed, say, 50 Mbps, for simplicity, a bit lasts for 20 nsec. This means it takes only 20 seconds for the clock to drift off by one bit. Consequently, the clocks must be continuously synchronized to keep them from getting too far apart. Certainly every 10 sec, preferably much more often.

37. Of the 90 columns, 86 are available for user data in OC-1. Thus, the user capacity is $86 \times 9 = 774$ bytes/frame. With 8 bits/byte, 8000 frames/sec, and 3 OC-1 carriers multiplexed together, the total user capacity is $3 \times 774 \times 8 \times 8000$, or 148.608 Mbps.

38. VT1.5 can accommodate 8000 frames/sec × 3 columns × 9 rows × 8 bits = 1.728 Mbps. It can be used to accommodate DS-1. VT2 can accommodate 8000 frames/sec × 4 columns × 9 rows × 8 bits = 2.304 Mbps. It can be used to accommodate European CEPT-1 service. VT6 can accommodate 8000 frames/sec × 12 columns × 9 rows × 8 bits = 6.912 Mbps. It can be used to accommodate DS-2 service.

39. Message switching sends data units that can be arbitrarily long. Packet switching has a maximum packet size. Any message longer than that is split up into multiple packets.

40. The OC-12c frames are $12 \times 90 = 1080$ columns of 9 rows. Of these, $12 \times 3 = 36$ columns are taken up by line and section overhead. This leaves an SPE of 1044 columns. One SPE column is taken up by path overhead, leaving 1043 columns for user data. Since each column holds 9 bytes of 8 bits, an OC-12c frame holds 75,096 user data bits. With 8000 frames/sec, the user data rate is 600.768 Mbps.

41. The three networks have the following properties:
star: best case = 2, average case = 2, worst case = 2
ring: best case = 1, average case = $n/4$, worst case = $n/2$
full interconnect: best case = 1, average case = 1, worst case = 1

42. With circuit switching, at $t = s$ the circuit is set up; at $t = s + x/b$ the last bit is sent; at $t = s + x/b + kd$ the message arrives. With packet switching, the last bit is sent at $t = x/b$. To get to the final destination, the last packet must be retransmitted $k - 1$ times by intermediate routers, each retransmission taking p/b sec, so the total delay is $x/b + (k - 1)p/b + kd$. Packet switching is faster if $s > (k - 1)p/b$.

43. The total number of packets needed is x/p, so the total data + header traffic is $(p + h)x/p$ bits. The source requires $(p + h)x/pb$ sec to transmit these bits. The retransmissions of the last packet by the intermediate routers take up a total of $(k - 1)(p + h)/b$ sec. Adding up the time for the source to send all the bits, plus the time for the routers to carry the last packet to the destination, thus clearing the pipeline, we get a total time of $(p + h)x/pb + (p + h)(k - 1)/b$ sec. Minimizing this quantity with respect to p, we find $p = \sqrt{hx/(k - 1)}$.

44. Each cell has six neighbors. If the central cell uses frequency group A, its six neighbors can use B, C, B, C, B, and C respectively. In other words, only 3 unique cells are needed. Consequently, each cell can have 280 frequencies.

45. First, initial deployment simply placed cells in regions where there was high density of human or vehicle population. Once they were there, the operator often did not want to go to the trouble of moving them. Second, antennas are typically placed on tall buildings or mountains. Depending on the exact location of such structures, the area covered by a cell may be irregular due to obstacles near the transmitter. Third, some communities or property owners do not allow building a tower at a location where the center of a cell falls. In such cases, directional antennas are placed at a location not at the cell center.

46. If we assume that each microcell is a circle 100 m in diameter, then each cell has an area of 2500π. If we take the area of San Francisco, 1.2×10^8 m^2 and divide it by the area of 1 microcell, we get 15,279 microcells. Of course, it is impossible to tile the plane with circles (and San Francisco is decidedly three-dimensional), but with 20,000 microcells we could probably do the job.

47. Frequencies cannot be reused in adjacent cells, so when a user moves from one cell to another, a new frequency must be allocated for the call. If a user moves into a cell, all of whose frequencies are currently in use, the user's call must be terminated.

48. It is not caused directly by the need for backward compatibility. The 30 kHz channel was indeed a requirement, but the designers of D-AMPS did not have to stuff three users into it. They could have put two users in each channel, increasing the payload before error correction from $260 \times 50 = 13$ kbps to $260 \times 75 = 19.5$ kbps. Thus, the quality loss was an intentional trade-off to put more users per cell and thus get away with bigger cells.

49. D-AMPS uses 832 channels (in each direction) with three users sharing a single channel. This allows D-AMPS to support up to 2496 users simultaneously per cell. GSM uses 124 channels with eight users sharing a single channel. This allows GSM to support up to 992 users simultaneously. Both systems use about the same amount of spectrum (25 MHz in each direction). D-AMPS uses 30 KHz × 892 = 26.76 MHz. GSM uses 200 KHz × 124 = 24.80 MHz. The difference can be mainly attributed to the better speech quality provided by GSM (13 Kbps per user) over D-AMPS (8 Kbps per user).

50. The result is obtained by negating each of *A*, *B*, and *C* and then adding the three chip sequences. Alternatively the three can be added and then negated. The result is (+3 +1 +1 −1 −3 −1 −1 +1).

51. By definition

$$\mathbf{S} \bullet \mathbf{T} \equiv \frac{1}{m} \sum_{i=1}^{m} S_i T_i$$

If *T* sends a 0 bit instead of 1 bit, its chip sequence is negated, with the *i*-th element becoming $-T_i$. Thus,

$$\mathbf{S} \bullet \mathbf{T} \equiv \frac{1}{m} \sum_{i=1}^{m} S_i(-T_i) = -\frac{1}{m} \sum_{i=1}^{m} S_i T_i = 0$$

52. When two elements match, their product is +1. When they do not match, their product is −1. To make the sum 0, there must be as many matches as mismatches. Thus, two chip sequences are orthogonal if exactly half of the corresponding elements match and exactly half do not match.

53. Just compute the four normalized inner products:

$$(-1\ +1\ -3\ +1\ -1\ -3\ +1\ +1) \bullet (-1\ -1\ -1\ +1\ +1\ -1\ +1\ +1)/8 = 1$$
$$(-1\ +1\ -3\ +1\ -1\ -3\ +1\ +1) \bullet (-1\ -1\ +1\ -1\ +1\ +1\ +1\ -1)/8 = -1$$

$(-1\ +1\ -3\ +1\ -1\ -3\ +1\ +1) \bullet (-1\ +1\ -1\ +1\ +1\ +1\ -1\ -1)/8 = 0$
$(-1\ +1\ -3\ +1\ -1\ -3\ +1\ +1) \bullet (-1\ +1\ -1\ -1\ -1\ -1\ +1\ -1)/8 = 1$

The result is that *A* and *D* sent 1 bits, *B* sent a 0 bit, and *C* was silent.

54. Ignoring speech compression, a digital PCM telephone needs 64 kbps. If we divide 10 Gbps by 64 kbps we get 156,250 houses per cable. Current systems have hundreds of houses per cable.

55. It is both. Each of the 100 channels is assigned its own frequency band (FDM), and on each channel the two logical streams are intermixed by TDM. This example is the same as the AM radio example given in the text, but neither is a fantastic example of TDM because the alternation is irregular.

56. A 2-Mbps downstream bandwidth guarantee to each house implies at most 50 houses per coaxial cable. Thus, the cable company will need to split up the existing cable into 100 coaxial cables and connect each of them directly to a fiber node.

57. The upstream bandwidth is 37 MHz. Using QPSK with 2 bits/Hz, we get 74 Mbps upstream. Downstream we have 200 MHz. Using QAM-64, this is 1200 Mbps. Using QAM-256, this is 1600 Mbps.

58. Even if the downstream channel works at 27 Mbps, the user interface is nearly always 10-Mbps Ethernet. There is no way to get bits to the computer any faster than 10-Mbps under these circumstances. If the connection between the PC and cable modem is fast Ethernet, then the full 27 Mbps may be available. Usually, cable operators specify 10 Mbps Ethernet because they do not want one user sucking up the entire bandwidth.

SOLUTIONS TO CHAPTER 3 PROBLEMS

1. Since each frame has a chance of 0.8 of getting through, the chance of the whole message getting through is 0.8^{10}, which is about 0.107. Call this value *p*. The expected number of transmissions for an entire message is then

$$E = \sum_{i=1}^{\infty} ip(1-p)^{i-1} = p\sum_{i=1}^{\infty} i(1-p)^{i-1}$$

To reduce this, use the well-known formula for the sum of an infinite geometric series,

$$S = \sum_{1=1}^{\infty} \alpha^i = \frac{1}{1-\alpha}$$

Differentiate both sides with respect to α to get

$$S' = \sum_{i=1}^{\infty} i\alpha^{i-1} = \frac{1}{(1-\alpha)^2}$$

Now use $\alpha = 1 - p$ to get $E = 1/p$. Thus, it takes an average of 1/0.107, or about 9.3 transmissions.

2. The solution is
(a) 00000100 01000111 11100011 11100000 01111110
(b) 01111110 01000111 11100011 11100000 11100000 11100000 01111110 01111110
(c) 01111110 01000111 110100011 111000000 011111010 01111110

3. After stuffing, we get A B ESC ESC C ESC ESC ESC FLAG ESC FLAG D.

4. If you could always count on an endless stream of frames, one flag byte might be enough. But what if a frame ends (with a flag byte) and there are no new frames for 15 minutes. How will the receiver know that the next byte is actually the start of a new frame and not just noise on the line? The protocol is much simpler with starting and ending flag bytes.

5. The output is 011110111110011111010.

6. It is possible. Suppose that the original text contains the bit sequence 01111110 as data. After bit stuffing, this sequence will be rendered as 011111010. If the second 0 is lost due to a transmission error, what is received is 01111110, which the receiver sees as end of frame. It then looks just before the end of the frame for the checksum and verifies it. If the checksum is 16 bits, there is 1 chance in 2^{16} that it will accidentally be correct, leading to an incorrect frame being accepted. The longer the checksum, the lower the probability of an error getting through undetected, but the probability is never zero.

7. If the propagation delay is very long, as in the case of a space probe on or near Mars or Venus, forward error correction is indicated. It is also appropriate, in a military situation in which the receiver does not want to disclose his location by transmitting. If the error rate is low enough that an error-correcting code is good enough, it may also be simpler. Finally, real-time systems cannot tolerate waiting for retransmissions.

8. Making one change to any valid character cannot generate another valid character due to the nature of parity bits. Making two changes to even bits or two changes to odd bits will give another valid character, so the distance is 2.

9. Parity bits are needed at positions 1, 2, 4, 8, and 16, so messages that do not extend beyond bit 31 (including the parity bits) fit. Thus, five parity bits are sufficient. The bit pattern transmitted is 01101011001100111010 1

10. The encoded value is 101001001111.

11. If we number the bits from left to right starting at bit 1, in this example, bit 2 (a parity bit) is incorrect. The 12-bit value transmitted (after Hamming encoding) was 0xA4F. The original 8-bit data value was 0xAF.

12. A single error will cause both the horizontal and vertical parity checks to be wrong. Two errors will also be easily detected. If they are in different rows, the row parity will catch them. If they are in the same row, the column parity will catch them. Three errors might slip by undetected, for example, if some bit is inverted along with its row and column parity bits. Even the corner bit will not catch this.

13. Describe an error pattern as a matrix of n rows by k columns. Each of the correct bits is a 0, and each of the incorrect bits is a 1. With four errors per block, each block will have exactly four 1s. How many such blocks are there? There are nk ways to choose where to put the first 1 bit, $nk - 1$ ways to choose the second, and so on, so the number of blocks is $nk(nk-1)(nk-2)(nk-3)$. Undetected errors only occur when the four 1 bits are at the vertices of a rectangle. Using Cartesian coordinates, every 1 bit is at a coordinate (x, y), where $0 \le x < k$ and $0 \le y < n$. Suppose that the bit closest to the origin (the lower-left vertex) is at (p, q). The number of legal rectangles is $(k - p - 1)(n - q - 1)$. Then the total number of rectangles can be found by summing this formula for all possible p and q. The probability of an undetected error is then the number of such rectangles divided by the number of ways to distribute the four bits:

$$\frac{\sum_{p=0}^{k-2} \sum_{q=0}^{n-2} (k - p - 1)(n - q - 1)}{nk(nk - 1)(nk - 2)(nk - 3)}$$

14. The remainder is $x^2 + x + 1$.

15. The frame is 10011101. The generator is 1001. The message after appending three zeros is 10011101000. The remainder on dividing 10011101000 by 1001 is 100. So, the actual bit string transmitted is 10011101100. The received bit stream with an error in the third bit from the left is 10111101100. Dividing this by 1001 produces a remainder 100, which is different from zero. Thus, the receiver detects the error and can ask for a retransmission.

16. The CRC is computed during transmission and appended to the output stream as soon as the last bit goes out onto the wire. If the CRC were in the header, it would be necessary to make a pass over the frame to compute the CRC before transmitting. This would require each byte to be handled twice—once for checksumming and once for transmitting. Using the trailer cuts the work in half.

17. Efficiency will be 50% when the time to transmit the frame equals the round-trip propagation delay. At a transmission rate of 4 bits/ms, 160 bits takes 40 ms. For frame sizes above 160 bits, stop-and-wait is reasonably efficient.

18. To operate efficiently, the sequence space (actually, the send window size) must be large enough to allow the transmitter to keep transmitting until the first acknowledgement has been received. The propagation time is 18 ms. At T1 speed, which is 1.536 Mbps (excluding the 1 header bit), a 64-byte frame takes 0.300 msec. Therefore, the first frame fully arrives 18.3 msec after its transmission was started. The acknowledgement takes another 18 msec to get back, plus a small (negligible) time for the acknowledgement to arrive fully. In all, this time is 36.3 msec. The transmitter must have enough window space to keep going for 36.3 msec. A frame takes 0.3 ms, so it takes 121 frames to fill the pipe. Seven-bit sequence numbers are needed.

19. It can happen. Suppose that the sender transmits a frame and a garbled acknowledgement comes back quickly. The main loop will be executed a second time and a frame will be sent while the timer is still running.

20. Let the sender's window be (S_l, S_u) and the receiver's be (R_l, R_u). Let the window size be W. The relations that must hold are:

$$0 \le S_u - S_l + 1 \le W1$$
$$R_u - R_l + 1 = W$$
$$S_l \le R_l \le S_u + 1$$

21. The protocol would be incorrect. Suppose that 3-bit sequence numbers are in use. Consider the following scenario:

A just sent frame 7.
B gets the frame and sends a piggybacked ACK.
A gets the ACK and sends frames 0–6, all of which get lost.
B times out and retransmits its current frame, with the ACK 7.

Look at the situation at *A* when the frame with $r.ack = 7$ arrives. The key variables are $AckExpected = 0$, $r.ack = 7$, and $NextFrameToSend = 7$. The modified *between* would return *true*, causing *A* to think the lost frames were being acknowledged.

22. Yes. It might lead to deadlock. Suppose that a batch of frames arrived correctly and were accepted. Then the receiver would advance its window. Now suppose that all the acknowledgements were lost. The sender would eventually time out and send the first frame again. The receiver would send a NAK. Suppose that this were lost. From that point on, the sender would keep timing out and sending a frame that had already been accepted, but the receiver would just ignore it. Setting the auxiliary timer results in a correct acknowledgement being sent back eventually instead, which resynchronizes.

23. It would lead to deadlock because this is the only place that incoming acknowledgements are processed. Without this code, the sender would keep timing out and never make any progress.

24. It would defeat the purpose of having NAKs, so we would have to fall back to timeouts. Although the performance would degrade, the correctness would not be affected. The NAKs are not essential.

25. Consider the following scenario. *A* sends 0 to *B*. *B* gets it and sends an ACK, but the ACK gets lost. *A* times out and repeats 0, but now *B* expects 1, so it sends a NAK. If *A* merely re-sent r.ack+1, it would be sending frame 1, which it has not got yet.

26. No. The maximum receive window size is 1. Suppose that it were 2. Initially, the sender transmits frames 0–6. All are received and acknowledged, but the acknowledgement is lost. The receiver is now prepared to accept 7 and 0. When the retransmission of 0 arrives at the receiver, it will be buffered and 6 acknowledged. When 7 comes in, 7 and 0 will be passed to the host, leading to a protocol failure.

27. Suppose *A* sent *B* a frame that arrived correctly, but there was no reverse traffic. After a while *A* would time out and retransmit. *B* would notice that the sequence number was incorrect, since the sequence number is below *FrameExpected*. Consequently, it would send a NAK, which carries an acknowledgement number. Each frame would be sent exactly two times.

28. No. This implementation fails. With $MaxSeq = 4$, we get $NrBufs = 2$. The even sequence numbers use buffer 0 and the odd ones use buffer 1. This mapping means that frames 4 and 0 both use the same buffer. Suppose that frames 0–3 are received and acknowledged. The receiver's window now contains 4 and 0. If 4 is lost and 0 arrives, it will be put in buffer 0 and *arrived*[0] set to *true*. The loop in the code for *FrameArrival* will be executed once, and an out-of-order message delivered to the host. This protocol requires *MaxSeq* to be odd to work properly. However, other implementations of sliding window protocols do not all have this property

29. Let $t = 0$ denote the start of transmission. At $t = 1$ msec, the first frame has been fully transmitted. At $t = 271$ msec, the first frame has fully arrived. At $t = 272$ msec, the frame acknowledging the first one has been fully sent. At $t = 542$ msec, the acknowledgement-bearing frame has fully arrived. Thus, the cycle is 542 msec. A total of k frames are sent in 542 msec, for an efficiency of $k/542$. Hence

(a) $k = 1$, efficiency = 1/542 = 0.18%
(b) $k = 7$, efficiency = 7/542 = 1.29%
(c) $k = 4$, efficiency = 4/542 = 0.74%

30. With a 50-kbps channel and 8-bit sequence numbers, the pipe is always full. The number of retransmissions per frame is about 0.01. Each good frame wastes 40 header bits, plus 1% of 4000 bits (retransmission), plus a 40-bit NAK once every 100 frames. The total overhead is 80.4 bits per 3960 data bits, for a fraction 80.4/(3960 + 80.4) = 1.99 percent.

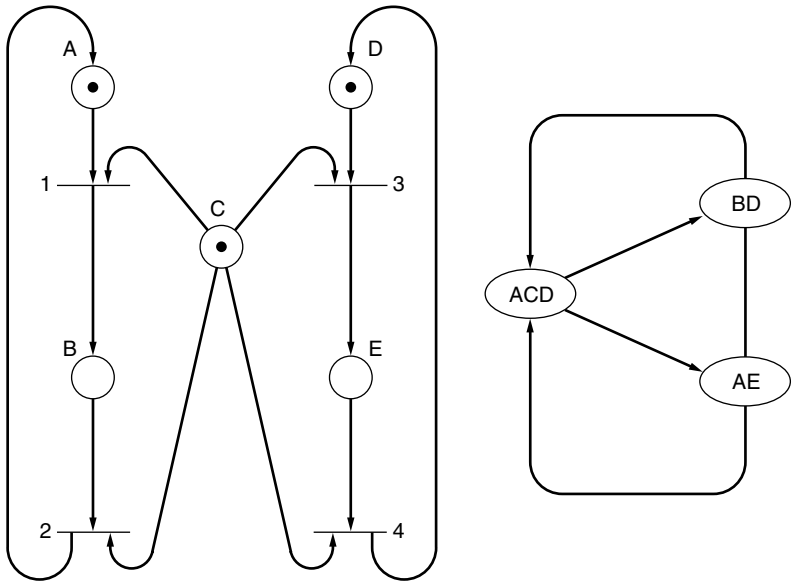
31. The transmission starts at $t = 0$. At $t = 4096/64000$ sec = 64 msec, the last bit is sent. At $t = 334$ msec, the last bit arrives at the satellite and the very short ACK is sent. At $t = 604$ msec, the ACK arrives at the earth. The data rate here is 4096 bits in 604 msec or about 6781 bps. With a window size of 7 frames, transmission time is 448 msec for the full window, at which time the sender has to stop. At 604 msec, the first ACK arrives and the cycle can start again. Here we have $7 \times 4096 = 28{,}672$ bits in 604 msec. The data rate is 47,470.2 bps. Continuous transmission can only occur if the transmitter is still sending when the first ACK gets back at $t = 604$ msec. In other words, if the window size is greater than 604 msec worth of transmission, it can run at full speed. For a window size of 10 or greater, this condition is met, so for any window size of 10 or greater (e.g., 15 or 127), the data rate is 64 kbps.

32. The propagation speed in the cable is 200,000 km/sec, or 200 km/msec, so a 100-km cable will be filled in 500 μsec. Each T1 frame is 193 bits sent in 125 μsec. This corresponds to four frames, or 772 bits on the cable.

33. Each machine has two key variables: *next_frame_to_send* and *frame_expected*, each of which can take on the values 0 or 1. Thus, each machine can be in one of four possible states. A message on the channel contains the sequence number of the frame being sent and the sequence number of the frame being ACKed. Thus, four types of messages exist. The channel may contain 0 or 1 message in either direction. So, the number of states the channel can be in is 1 with zero messages on it, 8 with one message on it, and 16 with two messages on it (one message in each direction). In total there are 1 + 8 + 16 = 25 possible channel states. This implies $4 \times 4 \times 25 = 400$ possible states for the complete system.

34. The firing sequence is 10, 6, 2, 8. It corresponds to acceptance of an even frame, loss of the acknowledgement, timeout by the sender, and regeneration of the acknowledgement by the receiver.

35. The Petri net and state graph are as follows:



The system modeled is mutual exclusion. *B* and *E* are critical sections that may not be active simultaneously, i.e., state *BE* is not permitted. Place *C* represents a semaphore that can be seized by either *A* or *D* but not by both together.

36. PPP was clearly designed to be implemented in software, not in hardware as HDLC nearly always is. With a software implementation, working entirely with bytes is much simpler than working with individual bits. In addition, PPP was designed to be used with modems, and modems accept and transmit data in units of 1 byte, not 1 bit.

37. At its smallest, each frame has two flag bytes, one protocol byte, and two checksum bytes, for a total of five overhead bytes per frame.

SOLUTIONS TO CHAPTER 4 PROBLEMS

1. The formula is the standard formula for Markov queueing given in section 4.1.1, namely, $T = 1/(\mu C - \lambda)$. Here $C = 10^8$ and $\mu = 10^{-4}$, so $T = 1/(10000 - lambda)$ sec. For the three arrival rates, we get (a) 0.1 msec, (b) 0.11 msec, (c) 1 msec. For case (c) we are operating a queueing system with $\rho = \lambda/\mu C = 0.9$, which gives the 10× delay.

2. With pure ALOHA the usable bandwidth is 0.184 × 56 kbps = 10.3 kbps. Each station requires 10 bps, so $N = 10300/10 = 1030$ stations.

3. With pure ALOHA, transmission can start instantly. At low load, no collisions are expected so the transmission is likely to be successful. With slotted ALOHA, it has to wait for the next slot. This introduces half a slot time of delay.

4. Each terminal makes one request every 200 sec, for a total load of 50 requests/sec. Hence $G = 50/8000 = 1/160$.

5. (a) With $G = 2$ the Poisson law gives a probability of e^{-2}.
(b) $(1 - e^{-G})^k e^{-G} = 0.135 \times 0.865^k$.
(c) The expected number of transmissions is $e^G = 7.4$.

6. (a) From the Poisson law again, $P_0 = e^{-G}$, so $G = -\ln P_0 = -\ln 0.1 = 2.3$.
(b) Using $S = Ge^{-G}$ with $G = 2.3$ and $e^{-G} = 0.1$, $S = 0.23$.
(c) Whenever $G > 1$ the channel is overloaded, so it is overloaded.

7. The number of transmissions is $E = e^G$. The E events are separated by $E - 1$ intervals of four slots each, so the delay is $4(e^G - 1)$. The throughput is given by $S = Ge^{-G}$. Thus, we have two parametric equations, one for delay and one for throughput, both in terms of G. For each G value it is possible to find the corresponding delay and throughput, yielding one point on the curve.

8. (a) The worst case is: all stations want to send and s is the lowest numbered station. Wait time N bit contention period + $(N - 1) \times d$ bit for transmission of frames. The total is $N + (N - 1)d$ bit times. (b) The worst case is: all stations have frames to transmit and s has the lowest virtual station number. Consequently, s will get its turn to transmit after the other $N - 1$ stations have transmitted one frame each, and N contention periods of size $\log_2 N$ each. Wait time is thus $(N - 1) \times d + N \times \log_2$ bits.

9. When station 4 sends, it becomes 0, and 1, 2, and 3 are increased by 1. When station 3 sends, it becomes 0, and 0, 1, and 2 are increased by 1. Finally, when station 9 sends, it becomes 0 and all the other stations are incremented by 1. The result is 9, 1, 2, 6, 4, 8, 5, 7, 0, and 3.

10. Stations 2, 3, 5, 7, 11, and 13 want to send. Eleven slots are needed, with the contents of each slot being as follows:

slot 1: 2, 3, 5, 7, 11, 13
slot 2: 2, 3, 5, 7
slot 3: 2, 3
slot 4: 2
slot 5: 3
slot 6: 5, 7
slot 7: 5
slot 8: 7
slot 9: 11, 13

slot 10: 11
slot 11: 13

11. The number of slots required depends on how far back in the tree one must go to find a common ancestor of the two stations. If they have the same parent (i.e., one level back), which happens with probability 2^{-n}, it takes $2n + 1$ slots to walk the tree. If the stations have a common grandparent, which happens with probability 2^{-n+1}, the tree walk takes $2n - 1$ slots, etc. The worst case is $2n + 1$ (common parent), and the best case is three slots (stations in different halves of the tree). The mean, m, is given by

$$m = \sum_{i=0}^{n-1} 2^{-(n-i)}(2n + 1 - 2i)$$

This expression can be simplified to

$$m = (1 - 2^{-n})(2n + 1) - 2^{-(n-1)} \sum_{i=0}^{n-1} i2^i$$

12. Radios cannot receive and transmit on the same frequency at the same time, so CSMA/CD cannot be used. If this problem could be solved (e.g., by equipping each station with two radios), there is still the problem of not all stations being within radio range of each other. Only if both of these problems can be solved, is CSMA/CD a candidate.

13. Both of them use a combination of FDM and TDM. In both cases dedicated frequency (i.e., wavelength) bands are available, and in both cases these bands are slotted for TDM.

14. Yes. Imagine that they are in a straight line and that each station can reach only its nearest neighbors. Then A can send to B while E is sending to F.

15. (a) Number the floors 1-7. In the star configuration, the router is in the middle of floor 4. Cables are needed to each of the $7 \times 15 - 1 = 104$ sites. The total length of these cables is

$$4 \sum_{i=1}^{7} \sum_{j=1}^{15} \sqrt{(i-4)^2 + (j-8)^2}$$

The total length is about 1832 meters.
(b) For 802.3, 7 horizontal cables 56 m long are needed, plus one vertical cable 24 m long, for a total of 416 m.

16. The Ethernet uses Manchester encoding, which means it has two signal periods per bit sent. The data rate of the standard Ethernet is 10 Mbps, so the baud rate is twice that, or 20 megabaud.

17. The signal is a square wave with two values, high (H) and low (L). The pattern is LHLHLHHLHLHLLHHLLHHL.

18. The pattern this time is HLHLHLLHHLLHLHHLHLLH.

19. The round-trip propagation time of the cable is 10 μsec. A complete transmission has six phases:

transmitter seizes cable (10 μsec)
transmit data (25.6 μsec)
Delay for last bit to get to the end (5.0 μsec)
receiver seizes cable (10 μsec)
acknowledgement sent (3.2 μsec)
Delay for last bit to get to the end (5.0 μsec)

The sum of these is 58.8 μsec. In this period, 224 data bits are sent, for a rate of about 3.8 Mbps.

20. Number the acquisition attempts starting at 1. Attempt i is distributed among 2^{i-1} slots. Thus, the probability of a collision on attempt i is $2^{-(i-1)}$. The probability that the first $k-1$ attempts fail, followed by a success on round k is

$$P_k = (1 - 2^{-(k-1)}) \prod_{i=1}^{k-1} 2^{-(i-1)}$$

which can be simplified to

$$P_k = (1 - 2^{-(k-1)})\, 2^{-(k-1)(k-2)/2}$$

The expected number of rounds is then just $\sum kP_k$.

21. For a 1-km cable, the one-way propagation time is 5 μsec, so $2\tau = 10$ μsec. To make CSMA/CD work, it must be impossible to transmit an entire frame in this interval. At 1 Gbps, all frames shorter than 10,000 bits can be completely transmitted in under 10 μsec, so the minimum frame is 10,000 bits or 1250 bytes.

22. The minimum Ethernet frame is 64 bytes, including both addresses in the Ethernet frame header, the type/length field, and the checksum. Since the header fields occupy 18 bytes and the packet is 60 bytes, the total frame size is 78 bytes, which exceeds the 64-byte minimum. Therefore, no padding is used.

23. The maximum wire length in fast Ethernet is 1/10 as long as in Ethernet.

24. The payload is 1500 bytes, but when the destination address, source address, type/length, and checksum fields are counted too, the total is indeed 1518.

25. The encoding is only 80% efficient. It takes 10 bits of transmitted data to represent 8 bits of actual data. In one second, 1250 megabits are transmitted, which means 125 million codewords. Each codeword represents 8 data bits, so the true data rate is indeed 1000 megabits/sec.

26. The smallest Ethernet frame is 512 bits, so at 1 Gbps we get 1,953,125 or almost 2 million frames/sec. However, this only works when frame bursting is operating. Without frame bursting, short frames are padded to 4096 bits, in which case the maximum number is 244,140. For the largest frame (12,144 bits), there can be as many as 82,345 frames/sec.

27. Gigabit Ethernet has it and so does 802.16. It is useful for bandwidth efficiency (one preamble, etc.) but also when there is a lower limit on frame size.

28. Station *C* is the closest to *A* since it heard the RTS and responded to it by asserting its NAV signal. *D* did not respond so it must be outside *A*'s radio range.

29. A frame contains 512 bits. The bit error rate is $p = 10^{-7}$. The probability of all 512 of them surviving correctly is $(1 - p)^{512}$, which is about 0.9999488. The fraction damaged is thus about 5×10^{-5}. The number of frames/sec is $11 \times 10^6/512$ or about 21,484. Multiplying these two numbers together, we get about 1 damaged frame per second.

30. It depends how far away the subscriber is. If the subscriber is close in, QAM-64 is used for 120 Mbps. For medium distances, QAM-16 is used for 80 Mbps. For distant stations, QPSK is used for 40 Mbps.

31. Uncompressed video has a constant bit rate. Each frame has the same number of pixels as the previous frame. Thus, it is possible to compute very accurately how much bandwidth will be needed and when. Consequently, constant bit rate service is the best choice.

32. One reason is the need for real-time quality of service. If an error is discovered, there is no time to get a retransmission. The show must go on. Forward error correction can be used here. Another reason is that on very low quality lines (e.g., wireless channels), the error rate can be so high that practically all frames would have to be retransmitted, and the retransmission would probably damaged as well. To avoid this, forward error correction is used to increase the fraction of frames that arrive correctly.

33. It is impossible for a device to be master in two piconets at the same time. There are two problems. First, only 3 address bits are available in the header while as many as seven slaves could be in each piconet. Thus, there would be no way to uniquely address each slave. Second, the access code at the start of the frame is derived from the master's identity. This is how slaves tell which

message belongs to which piconet. If two overlapping piconets used the same access code, there would be no way to tell which frame belonged to which piconet. In effect, the two piconets would be merged into one big piconet instead of two separate ones.

34. Bluetooth uses FHSS, just as 802.11 does. The biggest difference is that Bluetooth hops at a rate of 1600 hops/sec, far faster than 802.11.

35. An ACL channel is asynchronous, with frames arriving irregularly as data are produced. An SCO channel is synchronous, with frames arriving periodically at a well-defined rate.

36. They do not. The dwell time in 802.11 is not standardized, so it has to be announced to new stations that arrive. In Bluetooth this is always 625 μsec. There is no need to announce this. All Bluetooth devices have this hardwired into the chip. Bluetooth was designed to be cheap, and fixing the hop rate and dwell time leads to a simpler chip.

37. The first frame will be forwarded by every bridge. After this transmission, each bridge will have an entry for destination *a* with appropriate port in its hash table. For example, *D*'s hash table will now have an entry to forward frames destined to *a* on LAN 2. The second message will be seen by bridges *B*, *D*, and *A*. These bridges will append a new entry in their hash table for frames destined for *c*. For example bridge *D*'s hash table will now have another entry to forward frames destined to *c* on LAN 2. The third message will be seen by bridges *H*, *D*, *A*, and *B*. These bridges will append a new entry in their hash table for frames destined for *d*. The fifth message will be seen by bridges *E*, *C*, *B*, *D*, and *A*. Bridges *E* and *C* will append a new entry in their hash table for frames destined for *d*, while bridges *D*, *B*, and *A* will update their hash table entry for destination *d*.

38. Bridges *G*, *I* and *J* are not used for forwarding any frames. The main reason for having loops in an extended LAN is to increase reliability. If any bridge in the current spanning tree fails, the (dynamic) spanning tree algorithm reconfigures the spanning tree into a new one that may include one or more of these bridges that were not a part of the previous spanning tree.

39. The simplest choice is to do nothing special. Every incoming frame is put onto the backplane and sent to the destination card, which might be the source card. In this case, intracard traffic goes over the switch backplane. The other choice is to recognize this case and treat it specially, sending the frame out directly and not going over the backplane.

40. The worst case is an endless stream of 64-byte (512-bit) frames. If the backplane can handle 10^9 bps, the number of frames it can handle is $10^9/512$. This is 1,953,125 frames/sec.

41. The port on *B1* to LAN 3 would need to be relabeled as GW.

42. A store-and-forward switch stores each incoming frame in its entirety, then examines it and forwards it. A cut-through switch starts to forward incoming frames before they have arrived completely. As soon as the destination address is in, the forwarding can begin.

43. Store-and-forward switches store entire frames before forwarding them. After a frame comes in, the checksum can be verified. If the frame is damaged, it is discarded immediately. With cut=through, damaged frames cannot be discarded by the switch because by the time the error is detected, the frame is already gone. Trying to deal with the problem is like locking the barn door after the horse has escaped.

44. No. Hubs just connect all the incoming lines together electrically. There is nothing to configure. No routing is done in a hub. Every frame coming into the hub goes out on all the other lines.

45. It would work. Frames entering the core domain would all be legacy frames, so it would be up to the first core switch to tag them. It could do this by using MAC addresses or IP addresses. Similarly, on the way out, that switch would have to untag outgoing frames.

SOLUTIONS TO CHAPTER 5 PROBLEMS

1. File transfer, remote login, and video on demand need connection-oriented service. On the other hand, credit card verification and other point-of-sale terminals, electronic funds transfer, and many forms of remote database access are inherently connectionless, with a query going one way and the reply coming back the other way.

2. Yes. Interrupt signals should skip ahead of data and be delivered out of sequence. A typical example occurs when a terminal user hits the quit (kill) key. The packet generated from the quit signal should be sent immediately and should skip ahead of any data currently queued up for the program, i.e., data already typed in but not yet read.

3. Virtual circuit networks most certainly need this capability in order to route connection setup packets from an arbitrary source to an arbitrary destination.

4. The negotiation could set the window size, maximum packet size, data rate, and timer values.

5. Four hops means that five routers are involved. The virtual-circuit implementation requires tying up $5 \times 8 = 40$ bytes of memory for 1000 sec. The datagram implementation requires transmitting $12 \times 4 \times 200 = 9600$ bytes of header over and above what the virtual-circuit implementation needs. Thus,

the question comes down to the relative cost of 40,000 byte-sec of memory versus 9600 byte-hops of circuit capacity. If memory is depreciated over $2 \times 52 \times 40 \times 3600 = 1.5 \times 10^7$ sec, a byte-sec costs 6.7×10^{-8} cents, and 40,000 of them cost just over 2 millicents. If a byte-hop costs 10^{-6} cents, 9600 of them cost 9.6 millicents. Virtual circuits are cheaper for this set of parameters.

6. Yes. A large noise burst could garble a packet badly. With a k-bit checksum, there is a probability of 2^{-k} that the error is undetected. If the destination field or, equivalently, virtual-circuit number, is changed, the packet will be delivered to the wrong destination and accepted as genuine. Put in other words, an occasional noise burst could change a perfectly legal packet for one destination into a perfectly legal packet for another destination.

7. It will follow all of the following routes: *ABCD*, *ABCF*, *ABEF*, *ABEG*, *AGHD*, *AGHF*, *AGEB*, and *AGEF*. The number of hops used is 24.

8. Pick a route using the shortest path. Now remove all the arcs used in the path just found, and run the shortest path algorithm again. The second path will be able to survive the failure of any line in the first path, and vice versa. It is conceivable, though, that this heuristic may fail even though two line-disjoint paths exist. To solve it correctly, a max-flow algorithm should be used.

9. Going via *B* gives (11, 6, 14, 18, 12, 8).
Going via *D* gives (19, 15, 9, 3, 9, 10).
Going via *E* gives (12, 11, 8, 14, 5, 9).

Taking the minimum for each destination except *C* gives (11, 6, 0, 3, 5, 8). The outgoing lines are (*B*, *B*, –, *D*, *E*, *B*).

10. The routing table is 400 bits. Twice a second this table is written onto each line, so 800 bps are needed on each line in each direction.

11. It always holds. If a packet has arrived on a line, it must be acknowledged. If no packet has arrived on a line, it must be sent there. The cases 00 (has not arrived and will not be sent) and 11 (has arrived and will be sent back) are logically incorrect and thus do not exist.

12. The minimum occurs at 15 clusters, each with 16 regions, each region having 20 routers, or one of the equivalent forms, e.g., 20 clusters of 16 regions of 15 routers. In all cases the table size is 15 + 16 + 20 = 51.

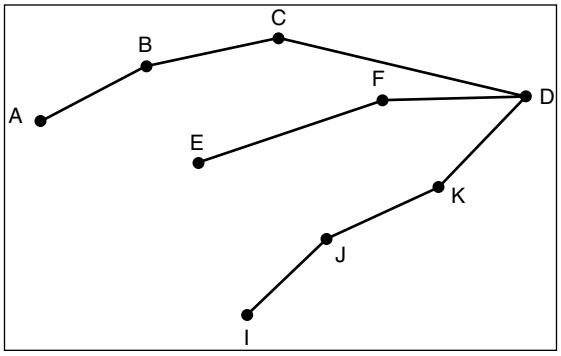
13. Conceivably it might go into promiscuous mode, reading all frames dropped onto the LAN, but this is very inefficient. Instead, what is normally done is that the home agent tricks the router into thinking it is the mobile host by responding to ARP requests. When the router gets an IP packet destined for the mobile host, it broadcasts an ARP query asking for the 802.3 MAC-level address of the machine with that IP address. When the mobile host is not

around, the home agent responds to the ARP, so the router associates the mobile user's IP address with the home agent's 802.3 MAC-level address.

14. (a) The reverse path forwarding algorithm takes five rounds to finish. The packet recipients on these rounds are *AC*, *DFIJ*, *DEGHIJKN*, *GHKN*, and *LMO*, respectively. A total of 21 packets are generated.
(b) The sink tree needs four rounds and 14 packets.

15. Node *F* currently has two descendants, *A* and *D*. It now acquires a third one, *G*, not circled because the packet that follows *IFG* is not on the sink tree. Node *G* acquires a second descendant, in addition to *D*, labeled *F*. This, too, is not circled as it does not come in on the sink tree.

16. Multiple spanning trees are possible. One of them is:



17. When *H* gets the packet, it broadcasts it. However, *I* knows how to get to *I*, so it does not broadcast.

18. Node *H* is three hops from *B*, so it takes three rounds to find the route.

19. It can do it approximately, but not exactly. Suppose that there are 1024 node identifiers. If node 300 is looking for node 800, it is probably better to go clockwise, but it could happen that there are 20 actual nodes between 300 and 800 going clockwise and only 16 actual nodes between them going counter-clockwise. The purpose of the cryptographic hashing function SHA-1 is to produce a very smooth distribution so that the node density is about the same all along the circle. But there will always be statistical fluctuations, so the straightforward choice may be wrong.

20. The node in entry 3 switches from 12 to 10.

21. The protocol is terrible. Let time be slotted in units of *T* sec. In slot 1 the source router sends the first packet. At the start of slot 2, the second router has received the packet but cannot acknowledge it yet. At the start of slot 3, the third router has received the packet, but it cannot acknowledge it either, so all the routers behind it are still hanging. The first acknowledgement can

only be sent when the destination host takes the packet from the destination router. Now the acknowledgement begins propagating back. It takes two full transits of the subnet, $2(n-1)T$ sec, before the source router can send the second packet. Thus, the throughput is one packet every $2(n-1)T$ sec.

22. Each packet emitted by the source host makes either 1, 2, or 3 hops. The probability that it makes one hop is p. The probability that it makes two hops is $p(1-p)$. The probability that it makes 3 hops is $(1-p)^2$. The mean path length a packet can expect to travel is then the weighted sum of these three probabilities, or $p^2 - 3p + 3$. Notice that for $p = 0$ the mean is 3 hops and for $p = 1$ the mean is 1 hop. With $0 < p < 1$, multiple transmissions may be needed. The mean number of transmissions can be found by realizing that the probability of a successful transmission all the way is $(1-p)^2$, which we will call α. The expected number of transmissions is just

$$\alpha + 2\alpha(1-\alpha) + 3\alpha(1-\alpha)^2 + \cdots = \frac{1}{\alpha} = \frac{1}{(1-p)^2}$$

Finally, the total hops used is just $(p^2 - 3p + 3)/(1-p)^2$.

23. First, the warning bit method explicitly sends a congestion notification to the source by setting a bit, whereas RED implicitly notifies the source by simply dropping one of its packets. Second, the warning bit method drops a packet only when there is no buffer space left, whereas RED drops packets before all the buffer are exhausted.

24. The router has to do approximately the same amount of work queueing a packet, no matter how big it is. There is little doubt that processing 10 packets of 100 bytes each is much more work than processing 1 packet of 1000 bytes.

25. It is not possible to send any packets greater than 1024 bytes, ever.

26. With a token every 5 μsec, 200,000 cells/sec can be sent. Each cell holds 48 data bytes or 384 bits. The net data rate is then 76.8 Mbps.

27. The naive answer says that at 6 Mbps it takes 4/3 sec to drain an 8 megabit bucket. However, this answer is wrong, because during that interval, more tokens arrive. The correct answer can be obtained by using the formula $S = C/(M - \rho)$. Substituting, we get $S = 8/(6-1)$ or 1.6 sec.

28. Call the length of the maximum burst interval Δt. In the extreme case, the bucket is full at the start of the interval (1 Mbyte) and another $10\Delta t$ Mbytes come in during the interval. The output during the transmission burst contains $50\Delta t$ Mbytes. Equating these two quantities, we get $1 + 10\Delta t = 50\Delta t$. Solving this equation, we get Δt is 25 msec.

29. The bandwidths in MB/sec are as follows: *A*: 2, *B*: 0, *C*: 1, *E*: 3, *H*: 3, *J*: 3, *K*: 2, and *L*: 1.

30. Here μ is 2 million and λ is 1.5 million, so $\rho = \lambda/\mu$ is 0.75, and from queueing theory, each packet experiences a delay four times what it would in an idle system. The time in an idle system is 500 nsec, here it is 2 μsec. With 10 routers along a path, the queueing plus service time is 20 μsec.

31. There is no guarantee. If too many packets are expedited, their channel may have even worse performance than the regular channel.

32. It is needed in both. Even in a concatenated virtual-circuit network, some networks along the path might accept 1024-byte packets, and others might only accept 48-byte packets. Fragmentation is still needed.

33. No problem. Just encapsulate the packet in the payload field of a datagram belonging to the subnet being passed through and send it.

34. The initial IP datagram will be fragmented into two IP datagrams at I1. No other fragmentation will occur.

Link *A-R1*:
Length = 940; *ID* = x; *DF* = 0; *MF* = 0; *Offset* = 0

Link *R1-R2*:
(1) *Length* = 500; *ID* = x; *DF* = 0; *MF* = 1; *Offset* = 0
(2) *Length* = 460; *ID* = x; *DF* = 0; *MF* = 0; *Offset* = 60

Link *R2-B*:
(1) *Length* = 500; *ID* = x; *DF* = 0; *MF* = 1; *Offset* = 0
(2) *Length* = 460; *ID* = x; *DF* = 0; *MF* = 0; *Offset* = 60

35. If the bit rate of the line is *b*, the number of packets/sec that the router can emit is *b*/8192, so the number of seconds it takes to emit a packet is 8192/*b*. To put out 65,536 packets takes $2^{29}/b$ sec. Equating this to the maximum packet lifetime, we get $2^{29}/b = 10$. Then, *b* is about 53,687,091 bps.

36. Since the information is needed to route every fragment, the option must appear in every fragment.

37. With a 2-bit prefix, there would have been 18 bits left over to indicate the network. Consequently, the number of networks would have been 2^{18} or 262,144. However, all 0s and all 1s are special, so only 262,142 are available.

38. The address is 194.47.21.130.

39. The mask is 20 bits long, so the network part is 20 bits. The remaining 12 bits are for the host, so 4096 host addresses exist.

40. To start with, all the requests are rounded up to a power of two. The starting address, ending address, and mask are as follows: *A:* 198.16.0.0 – 198.16.15.255 written as 198.16.0.0/20
B: 198.16.16.0 – 198.23.15.255 written as 198.16.16.0/21
C: 198.16.32.0 – 198.47.15.255 written as 198.16.32.0/20
D: 198.16.64.0 – 198.95.15.255 written as 198.16.64.0/19

41. They can be aggregated to 57.6.96/19.

42. It is sufficient to add one new table entry: 29.18.0.0/22 for the new block. If an incoming packet matches both 29.18.0.0/17 and 29.18.0.0./22, the longest one wins. This rule makes it possible to assign a large block to one outgoing line but make an exception for one or more small blocks within its range.

43. The packets are routed as follows:
(a) Interface 1
(b) Interface 0
(c) Router 2
(d) Router 1
(e) Router 2

44. After NAT is installed, it is crucial that all the packets pertaining to a single connection pass in and out of the company via the same router, since that is where the mapping is kept. If each router has its own IP address and all traffic belonging to a given connection can be sent to the same router, the mapping can be done correctly and multihoming with NAT can be made to work.

45. You say that ARP does not provide a service to the network layer, it is part of the network layer and helps provide a service to the transport layer. The issue of IP addressing does not occur in the data link layer. Data link layer protocols are like protocols 1 through 6 in Chap. 3, HDLC, PPP, etc. They move bits from one end of a line to the other.

46. RARP has a RARP server that answers requests. ARP does not have this. The hosts themselves answer ARP queries.

47. In the general case, the problem is nontrivial. Fragments may arrive out of order and some may be missing. On a retransmission, the datagram may be fragmented in different-sized chunks. Furthermore, the total size is not known until the last fragment arrives. Probably the only way to handle reassembly is to buffer all the pieces until the last fragment arrives and the size is known. Then build a buffer of the right size, and put the fragments into the buffer, maintaining a bit map with 1 bit per 8 bytes to keep track of which bytes are present in the buffer. When all the bits in the bit map are 1, the datagram is complete.

48. As far as the receiver is concerned, this is a part of new datagram, since no other parts of it are known. It will therefore be queued until the rest show up. If they do not, this one will time out too.

49. An error in the header is much more serious than an error in the data. A bad address, for example, could result in a packet being delivered to the wrong host. Many hosts do not check to see if a packet delivered to them is in fact really for them. They assume the network will never give them packets intended for another host. Data is sometimes not checksummed because doing so is expensive, and upper layers often do it anyway, making it redundant here.

50. Yes. The fact that the Minneapolis LAN is wireless does not cause the packets that arrive for her in Boston to suddenly jump to Minneapolis. The home agent in Boston must tunnel them to the foreign agent on the wireless LAN in Minneapolis. The best way to think of this situation is that the user has plugged into the Minneapolis LAN, the same way all the other Minneapolis users have. That the connection uses radio instead of a wire is irrelevant.

51. With 16 bytes there are 2^{128} or 3.4×10^{38} addresses. If we allocate them at a rate of 10^{18} per second, they will last for 10^{13} years. This number is 1000 times the age of the universe. Of course, the address space is not flat, so they are not allocated linearly, but this calculation shows that even with an allocation scheme that has an efficiency of 1/1000 (0.1 percent), one will never run out.

52. The *Protocol* field tells the destination host which protocol handler to give the IP packet to. Intermediate routers do not need this information, so it is not needed in the main header. Actually, it is there, but disguised. The *Next header* field of the last (extension) header is used for this purpose.

53. Conceptually, there are no changes. Technically, the IP addresses requested are now bigger, so bigger fields are needed.

SOLUTIONS TO CHAPTER 6 PROBLEMS

1. The LISTEN call could indicate a willingness to establish new connections but not block. When an attempt to connect was made, the caller could be given a signal. It would then execute, say, OK or REJECT to accept or reject the connection. In our original scheme, this flexibility is lacking.

2. The dashed line from *PASSIVE ESTABLISHMENT PENDING* to *ESTABLISHED* is no longer contingent on an acknowledgement arriving. The transition can happen immediately. In essence, the *PASSIVE ESTABLISHMENT PENDING* state disappears, since it is never visible at any level.

3. If the client sends a packet to *SERVER_PORT* and the server is not listening to that port, the packet will not be delivered to the server.

4. (a) The clock takes 32768 ticks, i.e., 3276.8 sec to cycle around. At zero generation rate, the sender would enter the forbidden zone at 3276.8 − 60 = 3216.8 sec.
(b) At 240 sequence numbers/min, the actual sequence number is $4t$, where t is in sec. The left edge of the forbidden region is $10(t - 3216.8)$. Equating these two formulas, we find that they intersect at $t = 5361.3$ sec.

5. Look at the second duplicate packet in Fig. 6-11(b). When that packet arrives, it would be a disaster if acknowledgements to y were still floating around.

6. Deadlocks are possible. For example, a packet arrives at A out of the blue, and A acknowledges it. The acknowledgement gets lost, but A is now open while B knows nothing at all about what has happened. Now the same thing happens to B, and both are open, but expecting different sequence numbers. Timeouts have to be introduced to avoid the deadlocks.

7. No. The problem is essentially the same with more than two armies.

8. If the AW or WA time is small, the events AC(W) and WC(A) are unlikely events. The sender should retransmit in state *S1*; the receiver's order does not matter.

9. Yes. Both sides could simultaneously execute RECEIVE.

10. Yes, $n_2 + n_3 + n_6 + n_7 = 1$. The states *listening*, *waiting*, *sending*, and *receiving* all imply that the user is blocked and hence cannot also be in another state.

11. A zero-length message is received by the other side. It could be used for signaling end of file.

12. None of the primitives can be executed, because the user is blocked. Thus, only packet arrival events are possible, and not all of these, either. *CallReq*, *ClearReq*, *DataPkt*, and *Credit* are the only legal ones.

13. The sliding window is simpler, having only one set of parameters (the window edges) to manage. Furthermore, the problem of a window being increased and then decreased, with the TPDUs arriving in the wrong order, does not occur. However, the credit scheme is more flexible, allowing a dynamic management of the buffering, separate from the acknowledgements.

14. No. IP packets contain IP addresses, which specify a destination machine. Once such a packet arrived, how would the network handler know which process to give it to? UDP packets contain a destination port. This information is essential so they can be delivered to the correct process.

15. It is possible that a client may get the wrong file. Suppose client *A* sends a request for file *f1* and then crashes. Another client *B* then uses the same protocol to request another file *f2*. Suppose client *B*, running on the same machine as *A* (with same IP address), binds its UDP socket to the same port that *A* was using earlier. Furthermore, suppose *B*'s request is lost. When the server's reply (to *A*'s request) arrives, client *B* will receive it and assume that it is a reply its own request.

16. Sending 1000 bits over a 1 Gbps line takes 1 μsec. The speed of light in fiber optics is 200 km/msec, so it takes 0.5 msec for the request to arrive and another 0.5 msec for the reply to get back. In all, 1000 bits have been transmitted in 1 msec. This is equivalent to 1 megabit/sec, or 1/10 of 1% efficiency.

17. At 1 Gbps, the response time is determined by the speed of light. The best that can be achieved is 1 msec. At 1 Mbps, it takes about 1 msec to pump out the 1024 bits, 0.5 msec for the last one to get to the server, and 0.5 msec for the reply to get back in the best case. The best possible RPC time is then 2 msec. The conclusion is that improving the line speed by a factor of 1000 only wins a factor of two in performance. Unless the gigabit line is amazingly cheap, it is probably not worth having for this application.

18. Here are three reasons. First, process IDs are OS-specific. Using process IDs would have made these protocols OS-dependent. Second, a single process may establish multiple channels of communications. A single process ID (per process) as the destination identifier cannot be used to distinguish between these channels. Third, having processes listen on well-known ports is easy, but well-known process IDs are impossible.

19. The default segment is 536 bytes. TCP adds 20 bytes and so does IP, making the default 576 bytes in total.

20. Even though each datagram arrives intact, it is possible that datagrams arrive in the wrong order, so TCP has to be prepared to reassemble the parts of a message properly.

21. Each sample occupies 4 bytes. This gives a total of 256 samples per packet. There are 44,100 samples/sec, so with 256 samples/packet, it takes 44100/256 or 172 packets to transmit one second's worth of music.

22. Sure. The caller would have to provide all the needed information, but there is no reason RTP could not be in the kernel, just as UDP is.

23. No. A connection is identified only by its sockets. Thus, $(1, p) – (2, q)$ is the only possible connection between those two ports.

24. The *ACK* bit is used to tell whether the 32-bit field is used. But if it were not there, the 32-bit field would always have to be used, if necessary acknowledging a byte that had already acknowledged. In short, it is not absolutely essential for normal data traffic. However, it plays a crucial role during connection establishment, where it is used in the second and third messages of the three-way handshake.

25. The entire TCP segment must fit in the 65,515-byte payload field of an IP packet. Since the TCP header is a minimum of 20 bytes, only 65,495 bytes are left for TCP data.

26. One way starts out with a LISTEN. If a *SYN* is received, the protocol enters the *SYN RECD* state. The other way starts when a process tries to do an active open and sends a *SYN*. If the other side was opening too, and a *SYN* is received, the *SYN RECD* state is also entered.

27. Even though the user is typing at a uniform speed, the characters will be echoed in bursts. The user may hit several keys with nothing appearing on the screen, and then all of a sudden, the screen catches up with the typing. People may find this annoying.

28. The first bursts contain 2K, 4K, 8K, and 16K bytes, respectively. The next one is 24 KB and occurs after 40 msec.

29. The next transmission will be 1 maximum segment size. Then 2, 4, and 8. So after four successes, it will be 8 KB.

30. The successive estimates are 29.6, 29.84, 29.256.

31. One window can be sent every 20 msec. This gives 50 windows/sec, for a maximum data rate of about 3.3 million bytes/sec. The line efficiency is then 26.4 Mbps/1000 Mbps or 2.6 percent.

32. The goal is to send 2^{32} bytes in 120 sec or 35,791,394 payload bytes/sec. This is 23,860 1500-byte frames/sec. The TCP overhead is 20 bytes. The IP overhead is 20 bytes. The Ethernet overhead is 26 bytes. This means that for 1500 bytes of payload, 1566 bytes must be sent. If we are to send 23,860 frames of 1566 bytes every second, we need a line of 299 Mbps. With anything faster than this we run the risk of two different TCP segments having the same sequence number at the same time.

33. A sender may not send more than 255 TPDUs, i.e., $255 \times 128 \times 8$ bits, in 30 sec. The data rate is thus no more than 8.704 kbps.

34. Compute the average: (270,000 $\times$ 0 + 730,000 $\times$ 1 msec)/1,000,000. It takes 730 μsec.

35. It takes $4 \times 10 = 40$ instructions to copy 8 bytes. Forty instructions takes 40 nsec. Thus, each byte requires 5 nsec of CPU time for copying. The system is thus capable of handle 200 MB/sec or 1600 Mbps. It can handle a 1-Gbps line if no other bottleneck is present.

36. The size of the sequence space is 2^{64} bytes, which is about 2×10^{19} bytes. A 75 Tbps transmitter uses up sequence space at a rate of 9.375×10^{12} sequence numbers per second. It takes 2 million seconds to wrap around. Since there are 86,400 seconds in a day, it will take over 3 weeks to wrap around, even at 75 Tbps. A maximum packet lifetime of less than 3 weeks will prevent the problem. In short, going to 64 bits is likely to work for quite a while.

37. RPC over UDP takes only two packets instead of three. However, RPC has a problem if the reply does not fit in one packet.

38. Yes. Packet 6 acknowledges both the request and the FIN. If each one were acknowledged separately, we would have 10 packets in the sequence. Alternatively, Packet 9, which acknowledges the reply, and the FIN could also be split into two separate packets. Thus, the fact that there are nine packets is just due to good luck.

39. With a packet 11.72 times smaller, you get 11.72 times as many per second, so each packet only gets 6250/11.72 or 533 instructions.

40. The speed of light in fiber and copper is about 200 km/msec. For a 20-km line, the delay is 100 μsec one way and 200 μsec round trip. A 1-KB packet has 8192 bits. If the time to send 8192 bits and get the acknowledgement is 200 μsec, the transmission and propagation delays are equal. If B is the bit time, then we have $8192B = 2 \times 10^{-4}$ sec. The data rate, $1/B$, is then about 40 Mbps.

41. The answer are: (1) 18.75 KB, (2) 125 KB, (3) 562.5 KB, (4) 1.937 MB. A 16-bit window size means a sender can send at most 64 KB before having to wait for an acknowledgement. This means that a sender cannot transmit continuously using TCP and keep the pipe full if the network technology used is Ethernet, T3, or STS-3.

42. The round-trip delay is about 540 msec, so with a 50 Mbps channel the bandwidth-product delay is 27 megabits or 3,375,000 bytes. With packets of 1500 bytes, it takes 2250 packets to fill the pipe, so the window should be at least 2250 packets.

SOLUTIONS TO CHAPTER 7 PROBLEMS

1. They are the DNS name, the IP address, and the Ethernet address.

2. Its IP address starts with 130, so it is on a class B network. See Chap. 5 for the IP address mapping.

3. It is not an absolute name, but relative to *.cs.vu.nl*. It is really just a shorthand notation for *rowboat.cs.vu.nl*.

4. It means: my lips are sealed. It is used in response to a request to keep a secret.

5. DNS is idempotent. Operations can be repeated without harm. When a process makes a DNS request, it starts a timer. If the timer expires, it just makes the request again. No harm is done.

6. The problem does not occur. DNS names *must* be shorter than 256 bytes. The standard requires this. Thus, all DNS names fit in a single minimum-length packet.

7. Yes. In fact, in Fig. 7-3 we see an example of a duplicate IP address. Remember that an IP address consists of a network number and a host number. If a machine has two Ethernet cards, it can be on two separate networks, and if so, it needs two IP addresses.

8. It is possible. *www.large-bank.com* and *www.large-bank.ny.us* could have the same IP address. Thus, an entry under *com* and under one of the country domains is certainly possible (and common).

9. There are obviously many approaches. One is to turn the top-level server into a server farm. Another is to have 26 separate servers, one for names beginning with *a*, one for *b*, and so on. For some period of time (say, 3 years) after introducing the new servers, the old one could continue to operate to give people a chance to adapt their software.

10. It belongs to the envelope because the delivery system needs to know its value to handle e-mail that cannot be delivered.

11. This is much more complicated than you might think. To start with, about half the world writes the given names first, followed by the family name, and the other half (e.g., China and Japan) do it the other way. A naming system would have to distinguish an arbitrary number of given names, plus a family name, although the latter might have several parts, as in John von Neumann. Then there are people who have a middle initial, but no middle name. Various titles, such as Mr., Miss, Mrs., Ms., Dr., Prof., or Lord, can prefix the name. People come in generations, so Jr., Sr., III, IV, and so on have to be included. Some people use their academic titles in their names, so we need

B.A., B.Sc., M.A., M.Sc., Ph.D., and other degrees. Finally, there are people who include certain awards and honors in their name. A Fellow of the Royal Society in England might append FRS, for example. By now we should be able to please even the learned:

Prof. Dr. Abigail Barbara Cynthia Doris E. de Vries III, Ph.D., FRS

12. It is doable and relatively simple. When incoming e-mail arrives, the SMTP daemon that accepts it has to look up the login name in the *RCPT TO* message. There is certainly a file or database where these names are located. That file could be extended to have aliases of the form "Ellen.Johnson" that point to the person's mailbox. Then e-mail can always be sent using the person's actual name.

13. The base 64 encoding will break the message into 1024 units of 3 bytes each. Each of these will be encoded as 4 bytes, for a total of 4096 bytes. If these are then broken up into lines of 80 bytes, 52 such lines will be needed, adding 52 CRs and 52 LFs. The total length will then be 4200 bytes.

14. If a sequence beginning with an equal sign and followed by two hexadecimal digits happens to appear in the text, e.g., =FF, this sequence will be mistakenly interpreted as an escape sequence. The solution is to encode the equal sign itself, so all equal signs always start escape sequences.

15. Some examples and possible helpers are application/msexcel(Excel), application/ppt (PowerPoint), audio/midi (MIDI sound), image/tiff (any graphics previewer), video/x-dv (QuickTime player).

16. Yes, use the *message/external-body* subtype and just send the URL of the file instead of the actual file.

17. The message sent just before logout will generate a canned reply. Its arrival will also generate a canned reply. Assuming each machine logs e-mail addresses to which it has already responded, no more canned replies will be sent.

18. First one is any sequence of one or more spaces and/or tabs. Second one is any sequence of one or more spaces and/or tabs and/or backspaces subject to the condition that the net result of applying all the backspaces still leaves at least one space or tab over.

19. The actual replies have to be done by the message transfer agent. When an SMTP connection comes in, the message transfer agent has to check whether a vacation daemon is set up to respond to the incoming e-mail, and if so, send an answer. The user transfer agent cannot do this because it will not even be invoked until the user comes back from vacation.

20. No. The POP3 program does not actually touch the remote mailbox. It sends commands to the POP3 daemon on the mail server. As long as that daemon understands the mailbox format, it can work. Thus, a mail server could change from one format to another overnight without telling its customers, as long as it simultaneously changes its POP3 daemon so it understands the new format.

21. Storing users' e-mail takes up disk space, which costs money. This factor argues for using POP3. On the other hand, the ISP could charge for disk storage above a few megabytes, thus turning e-mail into a moneymaker. The latter argues for IMAP to encourage users to keep e-mail on the server (and pay for disk space).

22. It does not use either one. But it is fairly similar in spirit to IMAP because both of them allow a remote client to examine and manage a remote mailbox. In contrast, POP3 just sends the mailbox to the client for processing there.

23. The browser has to be able to know whether the page is text, audio, video, or something else. The MIME headers provide this information.

24. If a browser receives a page with a MIME type that it cannot handle, it calls an external viewer to display the page. It finds the viewer's name in a configuration table, or it gets it from the user.

25. Yes, it is possible. Which helper is started depends on the configuration tables inside the browser, and Netscape and IE may have been configured differently. Furthermore, IE takes the file extension more seriously than the MIME type, and the file extension may indicate a different helper than the MIME type.

26. If a module gets two requests, one will be a cache hit and one will be a cache miss on average. The total CPU time consumed is 1 msec, and the total wait time is 9 msec. This gives a 10% CPU utilization, so with 10 modules the CPU is kept busy.

27. The official RFC 1738 way to do this is *http://dns-name:port/file.*

28. DNS names may not end with a digit, so there is no ambiguity.

29. The URL is probably *ftp://www.cs.stanford.edu/ftp/pub/freebies/newprog.c*

30. Do it the way *toms-casino* does: just put a customer ID in the cookie and store the preferences in a database on the server indexed by customer ID. That way the size of the record is unlimited.

31. Technically, it will work but it is a terrible idea. All the customer has to do is modify the cookie to get access to someone else's bank account. Having the cookie provide the customer's identity is safe, but the customer should be required to enter a password to prove his identity.

32. If the user has turned off the automatic displaying of images or if images cannot be displayed for some other reason, then the text given in *ALT* is displayed instead of the image. Also, if the mouse hovers over the image, the text may be displayed.

33. A hyperlink consists of and . In between them is the clickable text. It is also possible to put an image here. For example:

```
<a href="http://www.abcd.com/foo"> <img src="http://www.abcd.com/im/im2"> </a>
```

34. It would be <a href="http://www.acm.org"> ACM <a> .

35. Here is one way to do it.

```
<html>
<head> <title> INTERBURGER </title> </head>
<body>
<h1> Interburger's order form </h1>
<form action="http://interburger.com/cgi-bin/burgerorder" method=POST>
<p> Name <input name="customer" size=46> </p>
<p> Street Address <input name="address" size=40> </p>
<p> City <input name="city" size=20> </p>
Burger size Gigantic <input name="size" type=radio value="gigantic">
Immense <input name="size" type=radio value="immense">
Cheese <input name="cheese" type=checkbox>
<p> <input type=submit value="submit order"> </p>
</form>
</body> </html>
```

36. The page that displays the form looks like this:

```
<html>
<head> <title> Adder </title> </head>
<body>
<form action="action.php" method="post">
<p> Please enter first number: <input type="text" name="first"> </p>
<p> Please enter second number: <input type="text" name="second"> </p>
<input type="submit">
</form>
</body>
</html>
```

The PHP script that does the processing looks like this:

```
<html>
<head> <title> Addition </title> </head>
<body>
The sum is <?PHP echo $first + $second; ?>
</body>
</html>
```

37. (a) There are only 14 annual calendars, depending on the day of the week on which 1 January falls and whether the year is a leap year. Thus, a JavaScript program could easily contain all 14 calendars and a small database of which year gets which calendar. A PHP script could also be used, but it would be slower.

(b) This requires a large database. It must be done on the server by using PHP.

(c) Both work, but JavaScript is faster.

38. There are obviously many possible solutions. Here is one.

```
<html>
<head> <title> JavaScript test </title> </head>
<script language="javascript" type="text/javascript">

function response(test_form) {
   var n = 2;
   var has_factors = 0;
   var number = eval(test_form.number.value);
   var limit = Math.sqrt(number);
   while (n++ < limit) if (number % n == 0) has_factors = 1;
   document.open();
   document.writeln("<html> <body>");
   if (has_factors > 0) document.writeln(number, " is not a prime");
   if (has_factors == 0) document.writeln(number, " is a prime");
   document.writeln("</body> </html>");
   document.close();

}
</script>
</head>

<body>
<form name="myform">
Please enter a number: <input type="text" name="number">
<input type="button" value="compute primality" onclick="response(this.form)">
</form>
</body>
</html>
```

Clearly, this can be improved in various ways, but these require a bit more knowledge of JavaScript.

39. The commands sent are as follows:

```
GET /welcome.html HTTP/1.1
Host: www.info-source.com
```

Note the blank line at the end. It is mandatory.

40. Most likely HTML pages change more often than JPEG files. Lots of sites fiddle with their HTML all the time, but do not change the images much. But the effectiveness relates to not only the hit rate but also the payoff. There is not much difference between getting a 304 message and getting 500 lines of HTML. The delay is essentially the same in both cases because HTML files are so small. Image files are large, so not having to send one is a big win.

41. No. In the sports case, it is known days in advance that there will be a big crowd at the Web site and replicas can be constructed all over the place. The essence of a flash crowd is that it is unexpected. There was a big crowd at the Florida Web site but not at the Iowa or Minnesota sites. Nobody could have predicted this in advance.

42. Sure. The ISP goes to a number of content providers and gets their permission to replicate the content on the ISP's site. The content provider might even pay for this. The disadvantage is that it is a lot of work for the ISP to contact many content providers. It is easier to let a CDN do this.

43. It is a bad idea if the content changes rapidly. Pages full of up-to-the second sports results or stock quotes are not good candidates, for example. Pages that are generated dynamically are not suitable.

44. Each Japanese kanji (word) has been assigned a number. There are about 20,000 of them in Unicode. For an all-English system, it would be possible to assign the 65,000 most common words a 16-bit code and just transmit the code. The terminal would automatically add a space between words. Words not in the list, would be spelled out in ASCII. Using this scheme, most words would take 2 bytes, far less than transmitting them character by character. Other schemes might involve using 8-bit codes for the most common words and longer codes for less frequent codes (primitive Huffman coding).

45. Audio needs 1.4 Mbps, which is 175 KB/sec. On a 650-MB device, there is room for 3714 sec of audio, which is just over an hour. CDs are never more than an hour long, so there is no need for compression and it is not used.

46. The true values are $\sin(2\pi i/32)$ for i from 1 to 3. Numerically, these sines are 0.195, 0.383, and 0.556. They are represented as 0.250, 0.500, and 0.500, respectively. Thus, the percent errors are 28, 31, and 10 percent, respectively.

47. In theory it could be used, but Internet telephony is real time. For music, there is no objection to spending 5 minutes to encode a 3-minute song. For real-time speech, that would not work. Psychoacoustic compression could work for telephony, but only if a chip existed that could do the compression on the fly with a delay of around 1 msec.

48. It takes 50 msec to get a pause command to the server, in which time 6250 bytes will arrive, so the low-water mark should be way above 6250, probably 50,000 to be safe. Similarly, the high-water mark should be at least 6250 bytes from the top, but, say, 50,000 would be safer.

49. It introduces extra delay. In the straightforward scheme, after 5 msec have elapsed, the first packet can be sent. In this scheme, the system has to wait until 10 msec until it can send the samples for the first 5 msec.

50. It depends. If the caller is not behind a firewall and the callee is at a regular telephone, there are no problems at all. If the caller is behind a firewall and the firewall is not picky about what leaves the site, it will also work. If the callee is behind a firewall that will not let UDP packets out, it will not work.

51. The number of bits/sec is just $800 \times 600 \times 40 \times 8$ or 153.6 Mbps.

52. Yes. An error in an I-frame will cause errors in the reconstruction of subsequent P-frames and B-frames. In fact, the error will continue to propagate until the next I-frame.

53. With 100,000 customers each getting two movies per month, the server outputs 200,000 movies per month or about 6600 per day. If half of these are at P.M., the server must handle about 3300 movies at once. If the server has to transmit 3300 movies at 4 Mbps each, the required bandwidth is 13.2 Gbps. Using OC-12 connections, with a SPE capacity of 594 Mbps each, at least 23 connections will be needed. A machine serving 3300 movies simultaneously over 23 OC-12 connections is not a small machine.

54. The fraction of all references to the first r movies is given by

$$C/1 + C/2 + C/3 + C/4 + \cdots + C/r$$

Thus, the ratio of the first 1000 to the first 10,000 is

$$\frac{1/1 + 1/2 + 1/3 + 1/4 + \cdots + 1/1000}{1/1 + 1/2 + 1/3 + 1/4 + \cdots + 1/10000}$$

because the Cs cancel out. Evaluating this numerically, we get 7.486/9.788. Thus, about 0.764 of all requests will be to movies on magnetic disk. Noteworthy is that Zipf's law implies that a substantial amount of the distribution is in the tail, compared, say, to exponential decay.

SOLUTIONS TO CHAPTER 8 PROBLEMS

1. the time has come the walrus said to talk of many things
of shoes and ships and sealing wax of cabbages and kings
and why the sea is boiling hot and whether pigs have wings
but wait a bit the oysters cried before we have our chat
for some of us are out of breath and all of us are fat
no hurry said the carpenter they thanked him much for that

From *Through the Looking Glass* (Tweedledum and Tweedledee).

2. The plaintext is: a digital computer is a machine that can solve problems for people by carrying out instructions given to it.

From *Structured Computer Organization* by A. S. Tanenbaum.

3. It is:

1011111 0000100 1110000 1011011 1001000 1100010 0001011 0010111 1001101 1110000 1101110

4. At 100 Gbps, a bit takes 10^{-11} sec to be transmitted. With the speed of light being 2×10^8 meters/sec, in 1 bit time, the light pulse achieves a length of 2 mm or 2000 microns. Since a photon is about 1 micron in length, the pulse is 2000 photons long. Thus, we are nowhere near one photon per bit even at 100 Gbps. Only at 200 Tbps do we achieve 1 bit per photon.

5. Half the time Trudy will guess right. All those bits will be regenerated correctly. The other half she will guess wrong and send random bits to Bob. Half of these will be wrong. Thus, 25% of the bits she puts on the fiber will be wrong. Bob's one-time pad will thus be 75% right and 25% wrong.

6. If the intruder had infinite computing power, they would be the same, but since that is not the case, the second one is better. It forces the intruder to do a computation to see if each key tried is correct. If this computation is expensive, it will slow the intruder down.

7. Yes. A contiguous sequence of P-boxes can be replaced by a single P-box. Similarly for S-boxes.

8. For each possible 56-bit key, decrypt the first ciphertext block. If the resulting plaintext is legal, try the next block, etc. If the plaintext is illegal, try the next key.

9. The equation $2^n = 10^{15}$ tells us n, the number of doubling periods needed. Solving, we get $n = 15 \log_2 10$ or $n = 50$ doubling periods, which is 75 years. Just building that machine is quite a way off, and Moore's law may not continue for 75 more years.

10. The equation we need to solve is $2^{256} = 10^n$. Taking common logarithms, we get $n = 256 \log 2$, so $n = 77$. The number of keys is thus 10^{77}. The number of stars in our galaxy is about 10^{12} and the number of galaxies is about 10^8, so there are about 10^{20} stars in the universe. The mass of the sun, a typical star, is 2×10^{33} grams. The sun is made mostly of hydrogen and the number of atoms in 1 gram of hydrogen is about 6×10^{23} (Avogadro's number). So the number of atoms in the sun is about 1.2×10^{57}. With 10^{20} stars, the number of atoms in all the stars in the universe is about 10^{77}. Thus, the number of 256-bit AES keys is equal to the number of atoms in the whole universe (ignoring the dark matter). Conclusion: breaking AES-256 by brute force is not likely to happen any time soon.

11. DES mixes the bits pretty thoroughly, so a single bit error in block C_i will completely garble block P_i. In addition, one bit will be wrong in block P_{i+1}. However, all subsequent plaintext blocks will be correct. A single bit error thus only affects two plaintext blocks.

12. Unfortunately, every plaintext block starting at P_{i+1} will be wrong now, since all the inputs to the XOR boxes will be wrong. A framing error is thus much more serious than an inverted bit.

13. Cipher block chaining produces 8 bytes of output per encryption. Cipher feedback mode produces 1 byte of output per encryption. Thus, cipher block chaining is eight times more efficient (i.e., with the same number of cycles you can encrypt eight times as much plaintext).

14. (a) For these parameters, $z = 60$, so we must choose d to be relatively prime to 60. Possible values are: 7, 11, 13, 17, and 19.
(b) If e satisfies the equation $7e = 1 \bmod 360$, then $7\,e$ must be 361, 721, 1081, 1441, etc. Dividing each of these in turn by 7 to see which is divisible by 7, we find that 721/7 = 103, hence $e = 103$.
(c) With these parameters, $e = 3$. To encrypt P we use the function $C = P^3 \bmod 55$. For P = 1 to 10, C = 1, 8, 27, 9, 15, 51, 13, 17, 14, and 10, respectively.

15. Maria should consider changing her keys. This is because it is relatively easy for Frances to figure out Maria's private key as follows. Frances knows Maria's public key is ($e\,1$, $n\,1$). Frances notices $n\,2 = n\,1$. Frances now can guess Maria's private key($d\,1$, $n\,1$) by simply enumerating different solutions of the equation $d\,1 \times e\,1 = 1\ mod n\,1$.

16. No. The security is based on having a strong crypto algorithm and a long key. The *IV* is not really essential. The key is what matters.

17. The R_As from the last message may still be in RAM. If this is lost, Trudy can try to replay the most recent message to Bob, hoping that he will not see that it is a duplicate. One solution is for Bob to write the R_A of every incoming

message to disk *before* doing the work. In this case, the replay attack will not work. However, there is now a danger that if a request is written to disk followed shortly by a crash, the request is never carried out.

18. If Trudy replaces both parts, when Bob applies Alice's public key to the signature, he will get something that is not the message digest of the plaintext. Trudy can put in a false message and she can hash it, but she cannot sign it with Alice's private key.

19. When a customer, say, Sam, indicates that he wants to buy some pornography, gamble, or whatever, the Mafia order a diamond on Sam's credit card from a jeweler. When the jeweler sends a contract to be signed (presumably including the credit card number and a Mafia post office box as address), the Mafia forwards the hash of the jeweler's message to Sam, along with a contract signing up Sam as a pornography or gambling customer. If Sam just signs blindly without noticing that the contract and signature do not match, the Mafia forward the signature to the jeweler, who then ships them the diamond. If Sam later claims he did not order a diamond, the jeweler will be able to produce a signed contract showing that he did.

20. With 20 students, there are $(20 \times 19)/2 = 190$ pairs of students. The probability that the students in any pair have the same birthday is 1/365, and the probability that they have different birthdays is 364/365. The probability that all 190 pairs have different birthdays is thus $(364/365)^{190}$. This number is about 0.594. If the probability that all pairs are mismatches is 0.594, then the probability that one or more pairs have the same birthday is about 0.406.

21. The secretary can pick some number (e.g., 32) spaces in the letter, and potentially replace each one by space, backspace, space. When viewed on the terminal, all variants will look alike, but all will have different message digests, so the birthday attack still works. Alternatively, adding spaces at the end of lines, and interchanging spaces and tabs can also be used.

22. It is doable. Alice encrypts a nonce with the shared key and sends it to Bob. Bob sends back a message encrypted with the shared key containing the nonce, his own nonce, and the public key. Trudy cannot forge this message, and if she sends random junk, when decrypted it will not contain Alice's nonce. To complete the protocol, Alice sends back Bob's nonce encrypted with Bob's public key.

23. Step 1 is to verify the X.509 certificate using the root CA's public key. If it is genuine, she now has Bob's public key, although she should check the CRL if there is one. But to see if it is Bob on the other end of the connection, she needs to know if Bob has the corresponding private key. She picks a nonce and sends it to him with his public key. If Bob can send it back in plaintext, she is convinced that it is Bob.

24. First Alice establishes a communication channel with X and asks X for a certificate to verify his public key. Suppose X provides a certificate signed by another CA Y. If Alice does not know Y, she repeats the above step with Y. Alice continues to do this, until she receives a certificate verifying the public key of a CA Z signed by A and Alice knows A's public key. Note that this may continue until a root is reached, that is, A is the root. After this Alice verifies the public keys in reverse order starting from the certificate that Z provided. In each step during verification, she also checks the CRL to make sure that the certificate provided have not been revoked. Finally, after verifying Bob's public key, Alice ensures that she is indeed to talking to Bob using the same method as in the previous problem.

25. No. AH in transport mode includes the IP header in the checksum. The NAT box changes the source address, ruining the checksum. All packets will be perceived as having errors.

26. HMACs are much faster computationally.

27. Incoming traffic might be inspected for the presence of viruses. Outgoing traffic might be inspected to see if company confidential information is leaking out. Checking for viruses might work if a good antivirus program is used. Checking outgoing traffic, which might be encrypted, is nearly hopeless against a serious attempt to leak information.

28. If Jim does not want to reveal who he is communicating with to anyone (including his own system administrator, then Jim needs to use additional security mechanisms. Remember that VPN provides security for communication only over the Internet (outside the organization). It does not provide any security for communication inside the organization. If Jim only wants to keep his communication secure from people outside the company, a VPN is sufficient.

29. Yes. Suppose that Trudy XORs a random word with the start of the payload and then XORs the same word with the checksum. The checksum will still be correct. Thus, Trudy is able to garble messages and not have them be detected because she can manipulate the checksum through the encryption.

30. In message 2, put R_B inside the encrypted message instead of outside it. In this way, Trudy will not be able to discover R_B and the reflection attack will not work.

31. Bob knows that $g^x \bmod n = 191$. He computes $191^{15} \bmod 719 = 40$. Alice knows that $g^y \bmod n = 543$. She computes $543^{16} \bmod n = 40$. The key is 40. The simplest way to do the above calculations is to use the UNIX *bc* program.

32. There is nothing Bob knows that Trudy does not know. Any response Bob can give, Trudy can also give. Under these circumstances, it is impossible for Alice to tell if she is talking to Bob or to Trudy.

33. The KDC needs some way of telling who sent the message, hence which decryption key to apply to it.

34. No. All Trudy has to do is capture two messages from or to the same user. She can then try decrypting both of those with the same key. If the random number field in both of them is the same, bingo, she has the right key. All this scheme does is increase her workload by a factor of two.

35. The two random numbers are used for different purposes. R_A is used to convince Alice she is talking to the KDC. R_{A2} is used to convince Alice she is talking to Bob later. Both are needed.

36. If AS goes down, new legitimate users will not be able to authenticate themselves, that is, get a TGS ticket. So, they will not be able to access any servers in the organization. Users that already have a TGS ticket (obtained from AS before it went down) can continue to access the servers until their TGS ticket lifetime expires. If TGS goes down, only those users that already have a server ticket (obtained from TGS before it went down) for a server S will be able to access S until their server ticket lifetime expires. In both cases, no security violation will occur.

37. It is not essential to send R_B encrypted. Trudy has no way of knowing it, and it will not be used again, so it is not really secret. On the other hand, doing it this way allows a tryout of K_S to make doubly sure that it is all right before sending data. Also, why give Trudy free information about Bob's random number generator? In general, the less sent in plaintext, the better, and since the cost is so low here, Alice might as well encrypt R_B.

38. The bank sends a challenge (a long random number) to the merchant's computer, which then gives it to the card. The CPU on the card then transforms it in a complex way that depends on the PIN code typed directly into the card. The result of this transformation is given to the merchant's computer for transmission to the bank. If the merchant calls up the bank again to run another transaction, the bank will send a new challenge, so full knowledge of the old one is worthless. Even if the merchant knows the algorithm used by the smart cards, he does not know the customer's PIN code, since it is typed directly into the card. The on-card display is needed to prevent the merchant from displaying: "Purchase price is 49.95" but telling the bank it is 499.95.

39. Compression saves bandwidth, but more important, it also wipes out the frequency information containined in the plaintext (e.g., that "e" is the most common letter in English text). In effect, it converts the plaintext into junk, increasing the amount of work the cryptanalyst must do to break the message.

40. No. Suppose the address was a mailing list. Each person would have his or her own public key. Encrypting the IDEA key with just one public key would not work. It would have to be encrypted with multiple public keys.

41. In step 3, the ISP asks for *www.trudy-the-intruder.com* and it is never supplied. It would be better to supply the IP address to be less conspicuous. The result should be marked as uncacheable so the trick can be used later if necessary.

42. The DNS code is public, so the algorithm used for ID generation is public. If it is a random number generator, using random IDs hardly helps at all. By using the same spoofing attack as shown in the text, Trudy can learn the current (random) ID. Since random number generators are completely deterministic, if Trudy knows one ID, she can easily calculate the next one. If the random number generated by the algorithm is XORed with the time, that makes it less predictable, except that Trudy also knows the time. XORing the random number with the time and also with the number of lookups the server has done in the past minute (something Trudy does not know) and then taking the SHA-1 hash of this is much better. The trouble here is that SHA-1 takes a nontrivial amount of time and DNS has to be fast.

43. The nonces guard against replay attacks. Since each party contributes to the key, if an intruder tries to replay old messages, the new key generated will not match the old one.

44. Easy. Music is just a file. It does not matter what is in the file. There is room for 294,912 bytes in the low-order bits. MP3s require roughly 1 MB per minute, so about 18 sec of music could fit.

45. Alice could hash each message and sign it with her private key. Then she could append the signed hash and her public key to the message. People could compare verify the signature and compare the public key to the one Alice used last time. If Trudy tried to impersonate Alice and appended Alice's public key, she would not be able to get the hash right. If she used her own public key, people would see it was not the same as last time.