Tanenbaum solutions 6th edition for computer networks

Computer and Network Security (جامعة عين شمس)

COMPUTER NETWORKS

FIFTH EDITION

PROBLEM SOLUTIONS

ANDREW S. TANENBAUM

Vrije Universiteit
Amsterdam, The Netherlands

and

DAVID WETHERALL

University of Washington
Seattle, WA

PRENTICE HALL

Upper Saddle River, NJ

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.

(i) If dog's speed is doubled, maximum value of x is also doubled.

(ii) If tape capacity is doubled, value of x is also doubled.

(iii) If data rate of the transmission line is doubled, value of x is halved.

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 as well as video, 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. For financial transaction traffic, reliability and security are very important.

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. 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 $1 - np(1 - p)^{n-1} - (1 - p)^n$.

10. 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. One possible disadvantage is the performance of a layered system is likely to be worse than the performance of a monolithic system, although it is extremely difficult to implement and manage a monolithic system.

11. In the ISO protocol model, physical communication takes place only in the lowest layer, not in every layer.

12. 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.

13. 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.

14. 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 $k - 1$.

15. 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

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

16. 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 $hn/(M + hn)$.

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

18. 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.

19. 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 38.4 billion hosts. That sounds like a lot, but if every television, cellphone, camera, car, and appliance in the world is online, maybe it is plausible. The average person may have dozens of hosts by then.

20. 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).

21. Having mobile phone operators know the location of users lets the operators learn much personal information about users, such as where they sleep, work, travel and shop. This information might be sold to others or stolen; it could let the government monitor citizens. On the other hand, knowing the location of the user lets the operator send help to the right place in an emergency. It might also be used to deter fraud, since a person who claims to be you will usually be near your mobile phone.

22. 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.

23. The image is $1600 \times 1200 \times 3$ bytes or 5,760,000 bytes. This is 46,080,000 bits. At 56,000 bits/sec, it takes about 822.857 sec. At 1,000,000 bits/sec, it takes 46.080 sec. At 10,000,000 bits/sec, it takes 4.608 sec. At 100,000,000

bits/sec, it takes about 0.461 sec. At 1,000,000,000 bits/sec it takes about 46 msec.

24. 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.

25. 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.

26. There are many examples, of course. Some systems for which there is international standardization include compact disc players and their discs, digital cameras and their storage cards, 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.

27. This has no impact on the operations at layers k-1 or k+1.

28. There is no impact at layer k-1, but operations in k+1 have to be reimplemented.

29. One reason is request or response messages may get corrupted or lost during transmission. Another reason is the processing unit in the satellite may get overloaded processing several requests from different clients.

30. Small-sized cells result in large header-to-payload overhead. Fixed-size cells result in wastage of unused bytes in the payload.

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. A signal-to-noise ratio of 30 dB means S/N = 1000. So, the Shannon limit is about 39.86 kbps.

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. Fiber has many advantages over copper. It can handle much higher bandwidth than copper. It is not affected by power surges, electromagnetic interference, power failures, or corrosive chemicals in the air. It does not leak light and is quite difficult to tap. Finally, it is thin and lightweight, resulting in much lower installation costs. There are some downsides of using fiber over copper. First, it can be damaged easily by being bent too much. Second, optical communication is unidirectional, thus requiring either two fibers or two frequency bands on one fiber for two-way communication. Finally, fiber interfaces cost more than electrical interfaces.

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 $2560 \times 1600 \times 24 \times 60$ bps, which is 5898 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 $\Delta f = 5.898 \times 10^9$, so $\Delta\lambda = 3.3 \times 10^{-5}$ 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, 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. 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.

11. 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.

12. 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.

13. Transit time = 2 × (Altitude/Speed of light). The speed of light in air or vacuum is 300,000 km/sec. This evaluates to 239 msec for GEO, 120 msec for MEO, and 5 msec for LEO satellites.

14. The call travels from the North Pole to the satellite directly overhead, and then transits through four other satellites to reach the satellite directly above the South Pole. Down it goes down to earth to the South Pole. The total distance traveled is $2 \times 750 + 0.5 \times$ circumference at altitude 750 km. Circumference at altitude 750 km is $2 \times \pi \times (6371 + 750) = 44,720$ km. So, the total distance traveled is 23,860 km. Time to travel this distance $= 23860/300000 = 79.5$ msec. In addition, switching occurs at six satellites. So, the total switching time is 60 *u*sec. So, the total latency is about 79.56 msec.

15. In NRZ, the signal completes a cycle at most every 2 bits (alternating 1s and 0s). So, the minimum bandwidth need to achieve *B* bits/sec data rate is B/2 Hz. In MLT-3, the signal completes a cycle at most every 4 bits (a sequence of 1s), thus requiring at least *B*/4 Hz to achieve B bits/sec data rate. Finally, in Manchester encoding, the signal completes a cycle in every bit, thus requiring at least *B* Hz to achieve *B* bits/sec data rate.

16. Since 4B/5B encoding uses NRZI, there is a signal transition every time a 1 is sent. Furthermore, the 4B/5B mapping (see Figure 2-21) ensures that a sequence of consecutive 0s cannot be longer than 3. Thus, in the worst case, the transmitted bits will have a sequence 10001, resulting in a signal transition in 4 bits.

17. 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.

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 queuing 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. Supporting such a large number of telephones may result in significantly long

wait times. For example, if 5,000 (10% of 50,000) users decide to make a long-distance telephone call at the same time and each call lasts 3 minutes, the worst-case wait time will be 57 minutes. This will clearly result in unhappy customers.

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 $6 each, the copper is worth about 8.4 billion dollars.

20. Like a single railroad track, it is half duplex. Oil can flow in either direction, but not both ways at once. A river is an example of a simplex connection while a walkie-talkie is another example of a half-duplex connection.

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. While this significantly reduces the effective error rate seen at layer 2, errors at layer 2 are still possible. This can happen, for example, because of loss of data as it is transferred from layer 1 to layer 2 due lack of buffer space.

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. Since there are 32 symbols, 5 bits can be encoded. At 1200 baud, this provides $5 \times 1200 = 6000$ bps.

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

25. There are 10 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.

26. 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.)

27. The end users get $7 \times 24 = 168$ of the 193 bits in a frame. The overhead is therefore 25/193 = 13%. From Figure 2-41, percent overhead in OC-1 is (51.84 − 49.536)/51.84 = 3.63%. In OC-768, percent overhead is (39813.12 −

38043.648)/39813.12 = 4.44%.

28. In both cases 8000 samples/sec are possible. With dibit encoding, 2 bits are sent per sample. With T1, 7 bits are sent per period. The respective data rates are 16 kbps and 56 kbps.

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

30. 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.

31. 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 1 bit. Consequently, the clocks must be continuously synchronized to keep them from getting too far apart. Certainly every 10 sec, preferably much more often.

32. The lowest bandwidth link (1 Mbps) is the bottleneck.
One-way latency = $4 \times (35800/300000) = 480$ msec.
Total time = 1.2 + 233/220 + 0.48 = 8193.68 sec.

33. Again, the lowest-bandwidth link is the bottleneck.
Number of packets = 230/216 = 214.
One way latency = $480 + 3 \times 0.001 = 480.003$ msec.
Total bits transmitted = 233 + 214 * 28 = 233 + 222.
Total time = (233 + 222) / 220 + 0.48 = 8196.48 sec.

34. 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. For an OC-3072 line:

Gross data rate = $51.84 \times 3072 = 159252.48$ Mbps.
SPE data rate = $50.112 \times 3072 = 153944.064$ Mbps.
User data rate = $49.536 \times 3072 = 152174.592$ Mbps.

35. 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.

36. 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.

37. 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.

38. 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, with each retransmission taking p/b sec, so the total delay is $x/b + (k - 1)p/b + kd$. Packet switching is faster if $s > (k - 1)p/b$. In addition to the faster transmission under these conditions, packet switching is preferable when fault-tolerant transmission in the presence of switch failures is desired.

39. 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 $p = \sqrt{hx/(k - 1)}$.

40. 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 three unique cells are needed. Consequently, each cell can have 280 frequencies.

41. First, initial deployment simply placed cells in regions where there was a high density of human or vehicle population. Once they were there, the operators 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 a structure, 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. In the case of regular shapes, there is typically a buffer two cells wide where a frequency assigned to a cell is not reused in order to provide good separation and low interference. When the shapes of cells are irregular, the number of cells separating two cells that are using the same frequency is variable, depending on the width of the intermediate cells. This makes frequency

assignment much more complicated.

42. If we assume that each microcell is a circle 100 m in diameter, 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.

43. 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.

44. 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).

45. 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.

46. 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.

47. Here are the chip sequences:

(+1 +1 +1 +1 +1 +1 +1 +1)
(+1 −1 +1 −1 +1 −1 +1 −1)
(+1 +1 −1 −1 +1, +1 −1 −1)
(+1 −1 −1 +1 +1 −1 −1 +1)

48. 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.

49. 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.

50. 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.

51. The downstream data rate of a cable user is the smaller of the downstream cable bandwidth and the bandwidth of the communication medium between the cable modem and the user's PC. If the downstream cable channel works at 27 Mbps, the downstream data rate of the cable user will be

(a) 10 Mbps.
(b) 27 Mbps.
(c) 27 Mbps.

This is assuming that the communication medium between cable modem and the user's PC is not shared with any other user. 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. The maximum overhead occurs when the payload consists of only ESC and FLAG bytes. In this case, there will be 100% overhead.

5. 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.

6. The output is 011110111110011111010.

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 its 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, 5 parity bits are sufficient. The bit pattern transmitted is 01101011001100111010 1.

10. 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.

11. 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 will also be detected. If they are in the same row or column, that row's or column's parity will catch them. If two errors are in the same row, the column parity of at least one of them will catch the error. If two errors are in the same column, the row parity of at least one of them will catch the error. A 4-bit error in which the four error bits lie on the four corners of a rectangle cannot be caught.

12. From Eq. (3-1), we know that 10 check bits are needed for each block in case of using Hamming code. Total bits transmitted per block is 1010 bits. In case of error detection mechanism, one parity bit is transmitted per block. Suppose

error rate is x per bit. However, a block may encounter a bit error $1000x$ times. Every time an error is encountered, 1001 bits have to be retransmitted. So, total bits transmitted per block is $1001 + 1000x \times 1001$ bits. For error detection and retransmission to be better, $1001 + 1000x \times 1001 \prec 1010$. So, the error rate must be less than 9×10^{-6}.

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 $0 \le x \prec k$ and $0 \le y \prec n$. Suppose that the bit closest to the origin (the lower-left vertex) is at (p, q). The number of legal rectangles is $(k-p-1)(n-q-1)$. 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 4 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. When the first 1 goes in, 11 comes out and S_1 stores the 1. Then 0 goes in and 01 comes out, with S_2 now storing a 1 and S_1 storing the 0. The complete output sequence, including these initial values is 11 01 00 10 10 00 11 00.

15. To obtain the checksum, we need to calculate the ones complement sum of words, which is same as sum modulo 2^4 and adding any overflow of high order bits back into low-order bits:

$0011 + 1010 = 1101$
$1101 + 1100 = 1001 + 1 = 1010$
$1010 + 1001 = 0011 + 1 = 1100.$

So, the Internet checksum is 1100.

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

17. 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 of 100, which is different from 0. Thus, the receiver detects the error and can ask for a retransmission. If the

transmitted bit stream is converted to any multiple of 1001, the error will not be detected. A trivial example is if all ones in the bit stream are inverted to zeros.

18. The CRC checksum polynomial is or degree 32, so (a) Yes. CRC catches all single-bit errors.
(b) Yes. CRC catches all double-bit errors for any reasonably long message.
(c) No. CRC may not be able catch all even number of isolated bit errors.
(d) Yes. CRC catches all odd number of isolated bit errors.
(e) Yes. CRC catches all burst errors with burst lengths less than or equal to 32.
(f) No. CRC may not be able to catch a burst error with burst length greater than 32.

19. Yes, it is possible. The reason is that an acknowledgement frame may arrive correctly, but after the sender's timer has expired. This can happen if the receiver gets delayed in sending the acknowledgement frame, because its CPU is overloaded processing other jobs in the system.

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

21. 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.

22. To operate efficiently, the sequence space (actually, the sender's 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, so 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.

23. 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:

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

24. 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.

25. Yes. It might lead to deadlock. Suppose that a batch of frames arrived correctly and was accepted. 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 then send a NAK. If this packet 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.

26. 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.

27. Link utilization $= (1/(1 + 2BD))$
BD = bandwidth-delay product / frame size
delay = $(9 \times 10^{10)/(3} \times 10^{8)} = 300$ sec
bandwidth-delay product = 64 ×300 = 19.2 Gb
$BD = 19200000 / 256 = 75000$
So, link utilization is $6.67 \times 10^{-4}\%$

28. For a send window size w, link utilization is $w/(1 + 2BD)$. So, for 100% link utilization, $w = 150001$.

29. 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 resent r.ack + 1, it would be sending frame 1, which it has not gotten yet.

30. 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 it would be below *FrameExpected*. Consequently, it would send a NAK, which carries an acknowledgement number. Each frame would be sent exactly two times.

31. 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] will be set to *true*. The loop in the code for *FrameArrival* will be executed once, and an out-of-order message will be 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.

32. 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, for

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

33. 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, giving 80.4/(3960 + 80.4) = 1.99%.

34. 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.

35. 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.

36. PPP was clearly designed to be implemented in software, not in hardware as bit-stuffing protocols such as HDLC nearly always are. 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 2 flag bytes, 1 protocol byte, and 2 checksum bytes, for a total of 5 overhead bytes per frame. For maximum overhead, 2 flag bytes, 1 byte each for address and control, 2 bytes for protocol and 4 bytes for checksum. This totals to 10 overhead bytes.

38. The AAL5 frame will consist of 2 PPP protcol bytes, 100 PPP payload bytes, some padding bytes, and 8 trailer bytes. To make this frame size a multiple of 48, the number of padding bytes will be 34. This will result in an AAL5 frame of size 144 bytes. This can fit in three ATM cells. The first ATM cell will contain the 2 PPP protcol bytes and 46 bytes of the IP packet, the second cell will contain the next 48 bytes of the IP packet, and finally, the third ATM cell will contain the last 6 bytes of IP packet, 34 padding bytes, and 8 AAL5 trailer bytes.

SOLUTIONS TO CHAPTER 4 PROBLEMS

1. The formula is the standard formula for Markov queueing given in Sec. 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, and (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. (a) With $G = 2$ Poisson's 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 $e^G = 7.4$.

5. 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.

6. (a) Signal propagation speed in twin lead is 2.46×10^8 m/sec. Signal propagation time for 2 km is 8.13 μsec. So, the length of contention slot is 16.26 μsec. (b) Signal propagation speed in multimode fiber is 1.95×10^8 m/s. Signal propagation time for 40 km is 205.13 μsec. So, the length of contention slot is 410.26 μsec.

7. The worst case is where 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.

8. If a higher-numbered station and a lower-numbered station have packets to send at the same time, the higher-numbered station will always win the bid. Thus, a lower-numbered station will be starved from sending its packets if there is a continuous stream of higher-numbered stations ready to send their packets.

9. 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

10. (a) Since all stations will see A's packet, it will interfere with receipt of any other packet by any other station. So, no other communication is possible in this case.
(b) B/s packet will be seen by E, A and C, by not by D. Thus, E can send to D, or A can send to D, or C can send to D at the same time.
(c) This scenario is same as (b).

11. 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.

12. (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

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

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

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

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

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

1. Transmitter seizes cable (10 μsec)
2. Transmit data (25.6 μsec)
3. Delay for last bit to get to the end (5.0 μsec)
4. Receiver seizes cable (10 μsec)
5. Acknowledgement sent (3.2 μsec)
6. 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.

16. 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 will 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 $\sum kP_k$.

17. 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.

18. The maximum wire delay in fast Ethernet is 1/10 as long as in Ethernet.

19. The payload is 1500 bytes, but when the destination address, source address, type/length, and checksum fields are counted, plus the VLAN header, the total is indeed 1522. Prior to VLANs, the total was 1518.

20. 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.

21. 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.

22. 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.

23. RTS/CTS in 802.11 does not help with the exposed terminals problem. So, given the scenario in Figure 4-11(b), MACA protocol will allow simultaneous communication, *B* to *A* and *C* to *D*, but 802.11 will allow only one of these communications to take place at a time.

24. (a) Each set of ten frames will include one frame from each station. So, all stations will experience a data rate of 54/50 Mbps = 1.08 Mbps. (b) Each station gets the same amount of time to transmit. So, the 6 Mbps stations will get 0.6 Mbps, 18 Mbps stations will get 1.8 Mbps, and 54 Mbps stations will get 5.4 Mbps.

25. 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.

26. It depends how far away the subscriber is. If the subscriber is close, 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.

27. One reason is the need for real-time quality of service. If an error is discovered, there is no time for 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 retransmissions would probably damaged as well. To avoid this, forward error correction is used to increase the fraction of frames that arrive correctly.

28. Like 802.11, WiMAX wirelessly connects devices, including mobile devices to the Internet at Mbps speeds. Also, like 802.11, WiMAX is based on OFDM and MIMO technologies. However, unlike 802.11, WiMAX base stations are much more powerful than 802.11 access points. Also, transmissions in WiMAX are carefully scheduled by the base station for each subscriber

without any possibility of collisions unlike CSMA/CA used in 802.11.

29. It is impossible for a device to be master in two piconets at the same time. Allowing this would create 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.

30. A Bluetooth frame has an overhead of 126 bits for access code and header, and a settling time of 250 to 260 μsec. At the basic data rate, 1 Mbps, a settling time of 250 to 260 μsec corresponds to 250 to 260 bits. A slot is 625 μsec long, which corresponds to 625 bits at 1 Mbps. So, a maximum of 1875 bits can be transmitted in a 3-slot frame. Out of this, 376 to 386 bits are overhead bits, leaving a maximum of 1499 to 1509 bits for the data field.

31. 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.

32. In a 5-slot Bluetooth frame, a maximum of 3125 (625×5) bits can transmitted at basic rate. Out of this, a maximum of 2744 bits are for data. In case of repetition encoding, data is replicated thrice, so the actual data transmitted is about 914 bits. This results in about 29% efficiency.

33. 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.

34. We want to maximize the probability that one (and only one) tag responds in a given slot. Consulting Sec. 4.2.4, the best tag probability for 10 tags is 1/10. This occurs when the reader sets Q equal to 10 slots. Consulting Fig. 4-0, the probability that one tag responds is roughly 40%.

35. One key security concern is unauthorized tracking of RFID tags. An adversary with an appropriate RFID reader can track the locations of the items tagged using RFID tags. This becomes quite serious if the item is sensitive in nature, for example, a passport, and the tag can be used to retrieve further information, for example, the nationality and other personal information of the person holding the passport. Another security concern is the ability of a reader to change tag information. This can be used by an adversary to, for example, change the price of a tagged item he plans to buy.

36. 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.

37. 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.

38. (a) *B1* will forward this packet on ports 2, 3, and 4. *B2* will forward it on 1, 2 and 3.
(b) *B2* will forward this packet on ports 1, 3, and 4. *B1* will forward it on 1, 2 and 3.
(c) *B2* will not forward this packet on any of its ports, and *B1* will not see it.
(d) *B2* will forward this packet on port 2. *B1* will not see it.
(e) *B2* will forward this packet on port 4 and *B1* will forward it on port 1.
(f) *B1* will forward this packet on ports 1, 3 and 4. *B2* will forward it on port 2.

39. 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.

40. A bridge that does not have any station directly connected to any of its ports and is part of a loop is a candidate for not being a part of the spanning tree bridges. This can happen if the shortest paths to the root for all bridges connected to this bridge does not include this bridge.

41. 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.

42. 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 ac-

cess are inherently connectionless, with a query going one way and the reply coming back the other way.

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

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

4. 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.

5. 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.

6. 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).

7. 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.

8. 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.

9. 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.

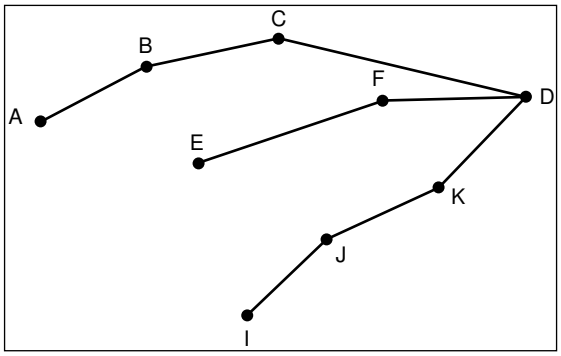
10. Conceivably it might go into promiscuous mode, reading all frames dropped onto the LAN, but this is very inefficient. Instead, what is normally done is that the home agent tricks the router into thinking it is the mobile host by responding to ARP requests. When the router gets an IP packet destined 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.

11. (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.

12. 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.

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



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

15. 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 network, $2(n-1)T$ sec, before the source router can send the second packet. Thus, the throughput is one packet every $2(n-1)T$ sec.

16. 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 $(p^2 - 3p + 3)/(1-p)^2$.

17. First, the ECN 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 ECN method drops a packet only when there is no buffer space left, whereas RED drops packets before all the buffer are exhausted.

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

19. 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.

20. 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.

21. 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.

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

23. 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

24. 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.

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

26. 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.

27. The address is 194.47.21.130.

28. 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.

29. Each Ethernet adapter sold in stores comes hardwired with an Ethernet (MAC) address in it. When burning the address into the card, the manufacturer has no idea where in the world the card will be used, making the address useless for routing. In contrast, IP addresses are either assigned either statically or dynamically by an ISP or company, which knows exactly how to get to the host getting the IP address.

30. 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

31. They can be aggregated to 57.6.96.0/19.

32. 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.

33. The packets are routed as follows:

(a) Interface 1
(b) Interface 0
(c) Router 2
(d) Router 1
(e) Router 2

34. 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 map-

ping can be done correctly and multihoming with NAT can be made to work.

35. 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.

36. 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.

37. 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.

38. 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.

39. 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.

40. 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.

41. 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.

42. 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. Since the two end points are peers, a separate application-level mechanism is needed that informs the end points at run time about which end will act as server and which end will act as client, as well as their addresses. One way to do this is to have a separate coordinator process that provides this information to the end points before a connection between the end points is established.

3. 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.

4. 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.

5. The connect( ) may fail if the server hasn't yet executed its listen( ) call.

6. One other criteria is how the client is affected by extra delay involved in process server technique. The server for the requested service has to be loaded and probably has to be initialized before the client request can be serviced.

7. (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.

8. 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.

9. 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.

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

11. 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.

12. Allocation for flow *A* will be 1/2 on links *R1R2* and *R2R3*. Allocation for flow *E* will 1/2 on links *R1R2* and *R2R6*. All other allocations remain the same.

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 segments 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. In AIAD and MIMD, the users will oscillate along the efficiency line, but will not converge. MIAD will converge just like AIMD. None of these policies are stable. Decrease policy in AIAD and MIAD is not aggressive, and increase policy in MIAD and MIMD is not gentle.

15. 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.

16. 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 the 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.

17. 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.

18. 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.

19. 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.

20. A client will use RPC over UDP if the operation is idempotent and the length of all parameters or results is small enough to fit in a single UDP packet. On the other hand if the parameters or results are large, or the operation is not idempotent, he will use RPC over TCP.

21. In *N*, since the maximum delay is 10 seconds, an appropriate buffer can be chosen to store a little more than 10 seconds of data at destination *D*. This will ensure that there will be no jitter experienced. On the other hand, in *N* 2, a smaller buffer, perhaps 2-3 seconds will be used, but some frames (that experience larger delays) will be dropped.

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

23. 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.

24. 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.

25. 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.

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

27. 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.

28. 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.

29. 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.

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

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

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

33. 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.

34. 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.

35. IP is a network level protocol while TCP is an end-to-end transport level protocol. Any change in the protocol specification of IP must be incorporated on all routers in the Internet. On the other hand, TCP can works fine as long as the two end points are running compatible versions. Thus, it is possible to have many different versions of TCP running at the same time on different hosts, but not this is not the case with IP.

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

37. Compute the average: (270,000 × 0 + 730,000 × 1 msec)/1,000,000. It takes 730 μsec.

38. 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 handling 200 MB/sec or 1600 Mbps. It can handle a 1-Gbps line if no other bottleneck is present. The size of the sequence space is 2^{64} bytes, which is about

39. 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.

40. 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.

41. 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.

42. The answers 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.

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

3. The DNS servers provide a mapping between domain names and IP addresses, such that when a request for a Web page is received, the browser can look up in the DNS server the IP address corresponding to the domain name of the requested page, and then download the requested page from that IP address.
If all the DNS servers in the world were to crash at the same time, one would not be able to map between domain names and IP addresses. Therefore, the only way to access Web pages would be by using the IP address of the host server instead of the domain name. Since most of us do not know the IP addresses of the servers we access, this type of situation would make use of the Internet extremely inefficient, if not virtually impossible for most users.

4. 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.

5. The generated name would probably be unique, and should therefore be allowed. However, DNS names *must* be shorter than 256 bytes, as required by the standard. Since together with the *com* ending the generated name would be longer than 256 characters, it is not permissible.

6. Yes. In fact, in Fig. 7-4 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.

7. 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.

8. It belongs to the envelope because the delivery system needs to know its value to handle email that cannot be delivered.

9. 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 names. 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

10. Naturally, the firm does not want to provide an additional email account for each employee. However, the only thing that needs to be done is to associate the alias *firstname.lastname* with a user's existing email account. This way, when incoming email at the SMTP daemon with a *TO* address of the form *firstname.lastname@lawfirm.com*, all it needs to do is look up what login name this alias corresponds to, and point that email to the mailbox *login@lawfirm.com*.

11. The base64 encoding will break the message into 1520 units of 3 bytes each. Each of these will be encoded as 4 bytes, for a total of 6080 bytes. If these are then broken up into lines of 110 bytes, 56 such lines will be needed, adding 56 CRs and 56 LFs. The total length will then be 6192 bytes.

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

13. Yes. Use the *message/external-body* subtype and just send the URL of the file instead of the actual file.

14. Each message received in John's work email inbox will be forwarded to his personal inbox, thereby generating an autoreply by the vacation agent, sent to his work inbox. This reply will be seen by the work computer as a new message, and thus be forwarded to the personal mailbox, which in turn, will send another reply to the work inbox. As a result there will be an endless string of messages for each message received in John's work email address (unless the vacation agent is smart enough to reply just once to each sender it sees).
However, assuming that the vacation agent logs email addresses to which it has already responded, a single auto-reply will be received by the work email inbox and forwarded back to the personal inbox, and no more canned messages will be generated.

15. The first one is any sequence of one or more spaces and/or tabs. The 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.

16. 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 agent is set up to respond to the incoming email, 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.

17. 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 counterclockwise. 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.

18. No. The IMAP program does not actually touch the remote mailbox. It sends commands to the IMAP 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 IMAP daemon so it understands the new format.

19. In the finger table for node 1, the node in entry 4 switches from 20 to 18. In the finger table for node 12, the node in entry 2 switches from 20 to 18. The finger table for node 4 is not affected by the change.

20. 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.

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

22. Yes, it is possible. Which helper is started depends on the configuration tables inside the browser, and Firefox 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.

23. As mentioned, an IP address is a set of four numbers separated by dots. An example of using an IP address is *http://192.31.231.66/index.html*. The browser uses the fact that a DNS name cannot end with a digit in order to distinguish between a URL using a DNS name and a URL using an IP address, which would always end with a digit.

24. The URL is probably *ftp://www.ma.stanford.edu/ftp/pub/forReview/newProof.pdf*.

25. 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.

26. 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 ID number is safe, but the customer should be required to enter a password to prove his identity.

27. (a) The browser uses the *TITLE* attribute when a user hovers with the mouse over the words "HEADER 1", and displays the value of that attribute as "this is the header".
(b) The *ALT* attribute is only useful for images, whereas the *TITLE* attribute can be included in any HTML tag. Additionally, the *ALT* attribute is used when the browser cannot find the image which should be displayed, whereas the *TITLE* attribute is used during hover-over. Due to these different uses, an <img> tag may include both *ALT* and *TITLE* attributes, though their values would typically be identical.

28. 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>
```

29. Here is one way to do it:

```
<html>
<body>
<a href="mailto:username@DomainName.com''> Click Here to email me </a>
</body>
</html>
```

When a user clicks this link, the user's default email-writing program opens up a ''compose message'' window including the address ''username@DomainName.com'' in the *TO* field.

30. One way of writing the XML page is:

```
<?xml version="1.0" ?>
<?xml-stylesheet type="text/xsl" href="student list.xsl"?>
<student list>
    <student>
        <name> Jerry </name>
        <address> 50 Farmington Av </address>
        <sid> 11227766 </sid>
        <gpa> 4.0 </gpa>
    </student>
    <student>
        <name> Elaine </name>
        <address> 5 Gumdrop Lane</address>
        <sid> 37205639 </sid>
        <gpa> 3.0 </gpa>
    </student>
    <student>
        <name> Tessa </name>
        <address> 6 Waterfall St </address>
        <sid> 43720472 </sid>
        <gpa> 3.8 </gpa>
    </student>
</student list>
```

31. (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.

32. 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.

33. 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.

34. 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.

35. No. In the sports case, it is known months 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.

36. Sure. The ISP goes to a number of content providers and gets their permission to replicate their content on the ISP's site. The content provider might even pay for this service. 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.

37. Audio needs 1.4 Mbps, which is 175 KB/sec. Two hours are $2 \times 60 \times 60$=7,200 seconds. Therefore, the number of Mbit needed in the CD is 10,080 M-bit, which are 1,260 MB.

38. 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.

39. 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.

40. It takes 100 msec to get a pause command to the server, in which time 12,500 bytes will arrive, so the low-water mark should be way above 12,500, probably 50,000 to be safe. Similarly, the high-water mark should be at least 12,500 bytes from the top, but, say, 50,000 would be safer.

41. 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.

42. The number of bits/sec is just $1200 \times 800 \times 50 \times 16$ or 768 Mbps.

43. 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.

44. With 50,000 customers each getting two movies per month,, the server outputs 150,000 movies per month or about 5000 per day. If half of these are at 9 P.M., the server must handle about 3330 movies at once. If the server has to transmit 3330 movies at 6 Mbps each, the required bandwidth is 20 Gbps. Using OC-12 connections, with an SPE capacity of 594 Mbps each, at least 34 connections will be needed.

45. 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 for movies in memory. 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. will you walk a little faster said a whiting to a snail
theres a porpoise close behind us and hes treading on my tail
see how eagerly the lobsters and the turtles all advance
they are waiting on the shingle will you come and join the dance
will you wont you will you wont you will you join the dance
will you wont you will you wont you wont you join the dance

From *Alice in Wonderland* (A Whiting and a Snail).

2. Assume that the most frequent plaintext letter is e and the second most frequent letter is t. In the ciphertext, the most frequent letter is 'R', and the second most frequent letter is 'K'. Note that the numerical values are e = 4; K = 10; R = 17; and t = 19. The following equations therefore exist:

$17 = (4a+b) \bmod 26$
$10 = (19a+b) \bmod 26$
Thus, $-7 = 15a \bmod 26$, which is equivalent to $19=15a \bmod 26$. By trial and error, we solve: $a = 3$. Then $17 = (12 + b) \bmod 26$. By observation, $b = 5$.

3. 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.

4. By getting hold of the encrypted key, Trudy now knows the length of the key. She can therefore determine how many columns there were in the transposition cipher matrix, and can break the ciphertext into columns. Subsequently, all Trudy has to do in order to decipher the message is try out all the arrangements of the columns until she finds one that makes sense. Assuming that the length of the encrypted key is k characters, finding the correct arrangement of the columns would require at most 2^k attempts.

5. It is:

1010011 0001110 1100010 1010110 1001011 0100110 1111100 0111100 1001010 1111111 1100001

6. You could use ASCII representation of the characters in *Lord of the Rings* to encrypt your messages. This will give you a one-time pad which is as long as the number of bits required to represent all the characters in *Lord of the Rings*. When you are near the end of the book, and your key is almost used up, you use the last portion of the book to send a message announcing the name of the next book you will be using as your one-time pad, and switch to that book for your subsequent messages. By continuing in this routine, because you have an infinite number of books, you also have an infinitely long one-time pad.

7. At 250 Gbps, a bit takes 4×10^{-12} 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 0.8 mm or 800 microns. Since a photon is about 1 micron in length, the pulse is 800 photons long. Thus, we are nowhere near one photon per bit even at 250 Gbps. Only at 200 Tbps do we achieve 1 bit per photon.

8. 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.

9. 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.

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

11. 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.

12. The equation $2^n = 10^{16}$ tells us n, the number of doubling periods needed. Solving, we get $n = 16 \log_2 10$ or $n = 53.15$ doubling periods, which is 79.72 years. Just building that machine is quite a way off, and Moore's Law may

not continue to hold for nearly 80 more years.

13. 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.

14. DES mixes the bits pretty thoroughly, so a single bit error in block C_i will completely garble block P_i. However, a one bit error in block C_i will not affect any other blocks, and therefore a single bit error only affects one plaintext block.

15. 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.

16. 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).

17. (a) For these parameters, $z = 48$, so we must choose d to be relatively prime to 48. Possible values are: 5, 7, 11, 13, and 17.
(b) If e satisfies the equation $37e = 1 \bmod 120$, then 37 e must be 121, 241, 361, 481 etc. Dividing each of these in turn by 37 to see which is divisible by 37, we find that 481/37 = 13, hence $e = 13$.
(c) With these parameters, $e = 9$. To encrypt P we use the function $C = P^9 \bmod 55$. For P = 8, 5, 12, 12, and 15, C = 18, 20, 12, 12, and 25, respectively.

18. Trudy can look up Alice's and Bob's public key pairs, and retrieve n_a and n_b. Because of the properties of the RSA algorithm, Trudy knows that each of these numbers is a multiplication of two primes, and therefore has only two prime factors. As stated in the question, Trudy also knows that one of the prime factors is common to n_a and n_b. Thus, Trudy concludes that the Greatest Common Divisor (GCD) of n_a and n_b is the common prime factor, q. All Trudy needs to do in order to break Alice's code is to use the Euclidean algorithm to find the GCD of n_a and n_b to obtain q, and then divide n_a by the result, q, to obtain p_a. Trudy can look up e_a in Alice's public key pair, and

can then find a solution to the equation $d_a \times e_a = 1 \; mod \; (p-1)\,(q-1)$, thereby determining Alice's private key.

19. 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.

20. 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.

21. 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 forward 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.

22. With 20 students, there are $(25 \times 24)/2 = 300$ pairs of students. The probability that the students in any pair have the same birthday is 1/181, and the probability that they have different birthdays is 180/181. The probability that all 300 pairs have different birthdays is thus $(180/181)^{300}$. This number is about 0.190. If the probability that all pairs are mismatches is 0.190, then the probability that one or more pairs have the same birthday is about 0.810.

23. 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.

24. 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.

25. 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.

26. 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.

27. 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.

28. The recommended method would be by using HMACs, since they are computationally faster than using RSA. However, this requires establishing a shared key with Bob prior to the transmission of the message.

29. 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.

30. The VPN provides security for communication over the Internet, but not within the organization. Therefore, when communicating with Mary regarding R&D purchases, or any other communication which need only be secure from people outside the organization, Jim does not need to use additional encryption or security measures. However, if Jim wants his communication with Mary to be secure also with respect to people inside the organization, such as when communicating with Mary about his salary and the raise he had been promised, additional security measures should be used.

31. 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.

32. Bob knows that $g^x \bmod n = 82$. He computes $82^3 \bmod 227 = 155$. Alice knows that $g^y \bmod n = 125$. She computes $125^{12} \bmod 227 = 155$. The key is 155. The simplest way to do the above calculations is to use the UNIX *bc* program.

33. (a) The information transferred from Alice to Bob is not encrypted, and therefore, 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.
(b) If n or g are secret, and are not known to Trudy, she cannot pretend to be Bob using a man-in-the-middle attack, since she would not be able to perform the correct calculations in order to send a return message to Alice and/or to obtain the correct key.

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

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. Even if Trudy intercepted the message including R_B she has no way of using it, since this value will not be used again in the communication between Alice and Bob. Thus, there is no need for Alice and Bob to repeat the protocol with different values in order to ensure the security of their communication. However, Trudy can use the information she gleaned from the intercepted message (and multiple other such messages) to try and figure out how Bob is generating his random numbers. Therefore, next time Alice should remember to encrypt the last message of the protocol.

38. 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.

39. 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 an-

other 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.

40. In order to multicast a PGP message, one would have to encrypt the IDEA key with the public key for each of the users accessing the Internet address. However, if all the users to whom the message is multicast have the same public key, the message can be multicast effectively.

41. 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.

42. 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.

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. The image contains 2048×512 pixels. Since each pixel contains 3 low-order bits, the number of bits which can be used for steganographic purposes is $2048 \times 512 \times 3$, which equals 3,145,728 bits or 393,216 bytes. The fraction of the file which could be encrypted in the image is approximately 0.16. If the file were compressed to a quarter of its original size, the compressed version would be of size 0.625 Mbyte. Therefore the fraction of the file which could be hidden in the image would be approximately 0.63.

45. 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.

46. The number of bits to be encrypted is $60 \times 10^6 \times 8 = 480 \times 10^6$ bits. Each pixel of the image can hide 3 bits in it. Therefore, the number of pixels required in order to encrypt the entire file is $480 \times 10^6 \ / \ 3 = 160 \times 10^6 = 160{,}000{,}000$ pixels. We want the image to be 3:2 so let the width be $3x$ and the height be $2x$. The number of pixels is then $6x^2$ which must be 160,000,000. Solving, we get $x = 5164$ and an image of 15492×10328. If the file were compressed to a third of its original size, the number of bits to be encrypted would be 160×10^6, and the number of pixels needed would be a third of the uncompressed file or 53,333,333 pixels. The image would then be 8946×5962.

47. 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 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.