CHAPTER 12
ROUTING IN SWITCHED NETWORKS

ANSWERS TO QUESTIONS

12.1 The average load expected over the course of the busiest hour of use during the course of a day.

12.2 The tradeoff is between efficiency and resilience.

12.3 A static routing strategy does not adapt to changing conditions on the network but uses a fixed strategy developed ahead of time. With alternate routing, there are a number of alternate routes between source and destination and a dynamic choice of routes is made.

12.4 Correctness, simplicity, robustness, stability, fairness, optimality, and efficiency.

12.5 For fixed routing, a single, permanent route is configured for each source-destination pair of nodes in the network.

12.6 With flooding, a packet is forwarded to all other switches so that eventually all routes between source and destination are traversed.

12.7 Advantages: (1) An adaptive routing strategy can improve performance, as seen by the network user. (2) An adaptive routing strategy can aid in congestion control. Because an adaptive routing strategy tends to balance loads, it can delay the onset of severe congestion. Disadvantages: (1) The routing decision is more complex; therefore, the processing burden on network nodes increases. (2) In most cases, adaptive strategies depend on status information that is collected at one place but used at another. There is a tradeoff here between the quality of the information and the amount of overhead. The more information that is exchanged, and the more frequently it is exchanged, the better will be the routing decisions that each node makes. On the other hand, this information is itself a load on the constituent networks, causing a performance degradation. (3) An adaptive strategy may react too quickly, causing congestion-producing oscillation, or too slowly, being irrelevant.

12.8 Given a network of nodes connected by bidirectional links, where each link has a cost associated with it in each direction, define the cost of a path between two nodes as the sum of the costs of the links traversed. For each pair of nodes, find a path with the least cost.

12.9 The Bellman-Ford algorithm uses only on information from its neighbors and knowledge of its link costs, to update it costs and paths. Dijkstra's algorithm requires that each node must have complete topological information about the network; that is, each node must know the link costs of all links in the network.

ANSWERS TO PROBLEMS

12.1 The number of hops is one less than the number of nodes visited.

a. The fixed number of hops is 2.

b. The furthest distance from a station is half-way around the loop. On average, a station will send data half this distance. For an N-node network, the average number of hops is (N / 4) – 1.

c. 1.

12.2 The mean node-node path is twice the mean node-root path. Number the levels of the tree with the root as 1 and the deepest level as N. The path from the root to level N requires N – 1 hops and 0.5 of the nodes are at this level. The path from the root to level N – 1 has 0.25 of the nodes and a length of N – 2 hops. Hence the mean path length, L, is given by

$$L = 0.5 \times (N - 1) + 0.25 \times (N - 2) + 0.125 \times (N - 3) + \ldots$$

$$L = \sum_{i=1}^{\infty} N(0.5)^i - \sum_{i=1}^{\infty} i(0.5)^i = N - 2$$

Thus the mean node-node path is 2N – 4

12.3
```
for  n := 1 to N do
    begin
        d[n] := ∞;
        p[n] := -1
    end;
d[srce] := 0              {initialize Q to contain srce only 0}
insert srce at the head of Q;
{initialization over }

while  Q is not empty do
    begin
        delete the head node j from Q;
        for each link jk that starts at j do
            begin
                newdist := d[j] + c[j,k];
                if newdist < d[k] then
                  begin
```

```
                    d[k] := newdist;
                    p[nk] := j
                    if  $k \notin Q$ then  insert K at the tail of Q;
               end
          end;
     end;
```

12.4 This proof is based on one in [BERT92]. Let us claim that

(1) $L(i) \leq L(j)$ for all $i \in T$ and $j \notin T$

(2) For each node j, L(j) is the shortest distance from s to j using paths with all nodes in T except possibly j.

Condition (1) is satisfied initially, and because $w(i, j) \geq 0$ and $L(i) = \min_{j \notin T} L(j)$, it is preserved by the formula in step 3 of the algorithm. Condition (2) then can be shown by induction. It holds initially. Suppose that condition (2) holds at the beginning of some iteration. Let i be the node added to T at that iteration, and let L(k) be the label of each node k at the beginning of the iteration. Condition (2) holds for i = j by the induction hypothesis, and it holds for all $j \in T$ by condition (1) and the induction hypothesis. Finally for a node $j \notin T \cup i$, consider a path from s to j which is shortest among all those in which all nodes of the path belong to $T \cup i$ and let L'(j) be the distance. Let k be the last node of this path before node j. Since k is in $T \cup i$, the length of this path from s to k is L(k). So we have

$$L'(j) = \min_{k \notin T \cup i}[w(k, j) + L(k)] = \min[\min_{k \notin T}[w(k, j) + L(k)], w(i, j) + L(i)]$$

The induction hypothesis implies that $L(j) = \min_{k \notin T}[w(k, j) + L(k)]$, so we have

$$L'(j) = \min[L(j) , w(i, j) + L(i)]$$

Thus in step 3, L(j) is set to the shortest distance from s to j using paths with all nodes except j belonging to $T \cup i$.

12.5 Consider the node i which has path length K+1, with the immediately preceding node on the path being j. The distance to node i is w(j, i) plus the distance to reach node j. This latter distance must be L(j), the distance to node j along the optimal route, because otherwise there would be a route with shorter distance found by going to j along the optimal route and then directly to i.

12.6 Not possible. A node will not be added to T until its least-cost route is found. As long as the least-cost route has not been found, the last node on that route will be eligible for entry into T before the node in question.

12.7 We show the results for starting from node 2.

	M	L(1)	Path	L(3)	Path	L(4)	Path	L(5)	Path	L(6)	Path
1	{2}	3	2-1	3	2-3	2	2-4	∞	—	∞	—

2	{2, 4}	3	2-1	3	2-3	2	2-4	3	2-4-5	∞	—
3	{2, 4, 1}	3	2-1	3	2-3	2	2-4	3	2-4-5	∞	—
4	{2, 4, 1, 3}	3	2-1	3	2-3	2	2-4	3	2-4-5	8	2-3-6
5	{2, 4, 1, 3, 5}	3	2-1	3	2-3	2	2-4	3	2-4-5	5	2-4-5-6
6	{2, 4, 1, 3, 5, 6}	3	2-1	3	2-3	2	2-4	3	2-4-5	5	2-4-5-6

12.8 We show the results for starting from node 2.

h	$L_h(1)$	Path	$L_h(3)$	Path	$L_h(4)$	Path	$L_h(5)$	Path	$L_h(6)$	Path
0	∞	—	∞	—	∞	—	∞	—	∞	—
1	3	2-1	3	2-3	2	2-4	∞	—	∞	—
2	3	2-1	3	2-3	2	2-4	3	2-4-5	8	2-3-6
3	3	2-1	3	2-3	2	2-4	3	2-4-5	5	2-4-5-6
4	3	2-1	3	2-3	2	2-4	3	2-4-5	5	2-4-5-6

12.9 a. We provide a table for node 1 of network a; the figure is easily generated.

	M	L(2)	Path	L(3)	Path	L(4)	Path	L(5)	Path	L(6)	Path
1	{1}	1	1-2	∞	—	4	1-4	∞	—	∞	—
2	{1,2}	1	1-2	4	1-2-3	4	1-4	2	1-2-5	∞	—
3	{1,2,5}	1	1-2	3	1-2-5-3	3	1-2-5-4	2	1-2-5	6	1-2-5-6
4	{1,2,5,3}	1	1-2	3	1-2-5-3	3	1-2-5-4	2	1-2-5	5	1-2-5-3-6
5	{1,2,5,3,4}	1	1-2	3	1-2-5-3	3	1-2-5-4	2	1-2-5	5	1-2-5-3-6
6	{1,2,5,3,4,6}	1	1-2	3	1-2-5-3	3	1-2-5-4	2	1-2-5	5	1-2-5-3-6

b. The table for network b is similar in construction but much larger. Here are the results for node A:

A to B: A-B
A to C: A-B-C
A to D: A-E-G-H-D
A to E: A-E
A to F: A-B-C-F
A to G: A-E-G
A to H: A-E-G-H
A to J: A-B-C-J
A to K: A--E-G-H-D-K

12.10

h	$L_h(2)$	Path	$L_h(3)$	Path	$L_h(4)$	Path	$L_h(5)$	Path	$L_h(6)$	Path
0	∞	—	∞	—	∞	—	∞	—	∞	—
1	1	1-2	∞	—	4	1-4	∞	—	∞	—
2	1	1-2	4	1-2-3	4	1-4	2	1-2-5	∞	—
3	1	1-2	3	1-2-5-3	3	1-2-5-4	2	1-2-5	6	1-2-3-6
4	1	1-2	3	1-2-5-3	3	1-2-5-4	2	1-2-5	5	1-2-5-3-6

12.11 If there is a unique least-cost path, the two algorithms will yield the same result because they are both guaranteed to find the least-cost path. If there are two or more equal least-cost paths, the two algorithms may find different least-cost paths, depending on the order in which alternatives are explored.

12.12 This explanation is taken from [BERT92]. The Floyd-Warshall algorithm iterates on the set of nodes that are allowed as intermediate nodes on the paths. It starts like both Dijkstra's algorithm and the Bellman-Ford algorithm with single arc distances (i.e., no intermediate nodes) as starting estimates of shortest path lengths. It then calculates shortest paths under the constraint that only node 1 can be used as an intermediate node, and then with the constraint that only nodes 1 and 2 can be used, and so forth.

For n = 0, the initialization clearly gives the shortest path lengths subject to the constraint of no intermediate nodes on paths. Now, suppose for a given n, $L_n(i, j)$ in the above algorithm gives the shortest path lengths using nodes 1 to n as intermediate nodes. Then the shortest path length from i to j, allowing nodes 1 to n+1 as possible intermediate nodes, either contains node n+1 on the shortest path or doesn't contain node n+1. For the first case, the constrained shortest path from i to j goes from i to n+1 and then from n+1 to j, giving the length in the final term of the equation in step 2 of the problem. For the second case, the constrained shortest path is the same as the one using nodes 1 to n as possible intermediate nodes, yielding the length of the first term in the equation in step 2 of the problem.

12.13 a. Consider Figure 12.4, which is valid through part (b). On the third stage, only 5 and 6 are receiving new packets. Node 5 retransmits only to node 6. Thus the total count is 13 packets.

b. Continue the process beyond Figure 12.4c.

12.14 No. Although it is true that the first packet to reach node 6 has experienced the minimum delay, this delay was experienced under a condition of network flooding, and cannot be considered valid for other network conditions.

12.15 The destination node may be unreachable.

12.16 If a node sees a packet arriving on line k from node H with hop count 4, it knows that H is at most four hops away via line k. If its current best route to H is estimated at more than four hops, it marks line k as the choice for traffic to H and records the estimated distance as four hops.

The advantage of this algorithm is that, since it is an isolated technique, minimal node-node cooperation is needed. The disadvantage occurs if a line goes down or is overloaded. The algorithm as described only records improvements, not changes for the worse.

12.17 a.

From Node

		1	2	3	4	5	6
	1	—	1	5	5	2	3
	2	2	—	5	5	2	3
To	3	2	5	—	5	3	3
Node	4	2	5	5	—	4	3
	5	2	5	5	5	—	3
	6	2	5	6	5	3	—

b.

		From Node									
		A	B	C	D	E	F	G	H	J	K
	A	—	B	B	H	A	C	E	G	C	D
	B	B	—	B	H	A	C	C	G	C	F
To	C	B	C	—	H	G	C	C	G	C	F
Node	D	E	C	G	—	G	K	H	D	D	D
	E	E	A	G	H	—	K	E	G	D	D
	F	B	C	F	K	G	—	C	D	D	F
	G	E	C	G	H	G	K	—	G	D	D
	H	E	C	G	H	G	K	H	—	D	D
	J	B	C	J	J	G	K	H	D	—	D
	K	E	C	F	K	G	K	H	D	D	—

12.18 Yes. With flooding, all possible paths are used. So at least one path that is the minimum-hop path to the destination will be used.

CHAPTER 13
CONGESTION IN DATA NETWORKS

ANSWERS TO QUESTIONS

13.1 Two general strategies can be adopted. The first such strategy is to discard any incoming packet for which there is no available buffer space. The alternative is for the node that is experiencing these problems to exercise some sort of flow control over its neighbors so that the traffic flow remains manageable.

13.2 Here is a simple intuitive explanation of why delay must go to infinity. Suppose that each node in the network is equipped with buffers of infinite size and suppose that the input load exceeds network capacity. Under ideal conditions, the network will continue to sustain a normalized throughput of 1.0. Therefore, the rate of packets leaving the network is 1.0. Because the rate of packets entering the network is greater than 1.0, internal queue sizes grow. In the steady state, with input greater than output, these queue sizes grow without bound and therefore queuing delays grow without bound.

13.3 Backpressure: This technique produces an effect similar to backpressure in fluids flowing down a pipe. It involves link-by-link use of flow control in a direction toward the source. Choke packet: A choke packet is a control packet generated at a congested node and transmitted back to a source node to restrict traffic flow. Implicit congestion signaling: If a source is able to detect increased delays and packet discards, then it has implicit evidence of network congestion. Explicit congestion signaling: In general terms, for explicit congestion avoidance, the network alerts end systems to growing congestion within the network and the end systems take steps to reduce the offered load to the network. Policing: A node in the network, typically the node to which the end system attaches, monitors the traffic flow and compares it to the traffic contract. Excess traffic is either discarded or marked to indicate that it is liable to discard or delay.

13.4 Backward: Notifies the source that congestion avoidance procedures should be initiated where applicable for traffic in the opposite direction of the received notification. It indicates that the packets that the user transmits on this logical connection may encounter congested resources. Forward: Notifies the user that congestion avoidance procedures should be initiated where applicable for traffic in the same direction as the received notification. It indicates that this packet, on this logical connection, has encountered congested resources.

13.5 Binary: A bit is set in a data packet as it is forwarded by the congested node. When a source receives a binary indication of congestion on a logical connection, it may reduce its traffic flow. Credit based: These schemes are based on providing an explicit credit to a source over a logical connection. The credit indicates how many octets or how many packets the source may transmit. When the credit is exhausted, the source must await additional credit before sending additional data. Rate based: These schemes are based on providing an explicit data rate limit to the source over a logical connection. The source may transmit data at a rate up to the set limit. To control congestion, any node along the path of the connection can reduce the data rate limit in a control message to the source.

13.6 This is a rate, in bits per second, that the network agrees to support for a particular frame-mode connection. Any data transmitted in excess of the CIR are vulnerable to discard in the event of congestion.

13.7 If a constant data rate with no variability is desired, cell delay variation complicates the satisfaction of this requirement.

13.8 (1) Control of peak cell rate and the associated cell delay variation (CDV).
(2) Control of sustainable cell rate and the associated burst tolerance

13.9 Traffic policing occurs when a flow of data is regulated so that cells (or frames or packets) that exceed a certain performance level are discarded or tagged. It may be desirable to supplement a traffic-policing policy with a traffic-shaping policy. Traffic shaping is used to smooth out a traffic flow and reduce cell clumping. This can result in a fairer allocation of resources and a reduced average delay time.

Answers to Problems

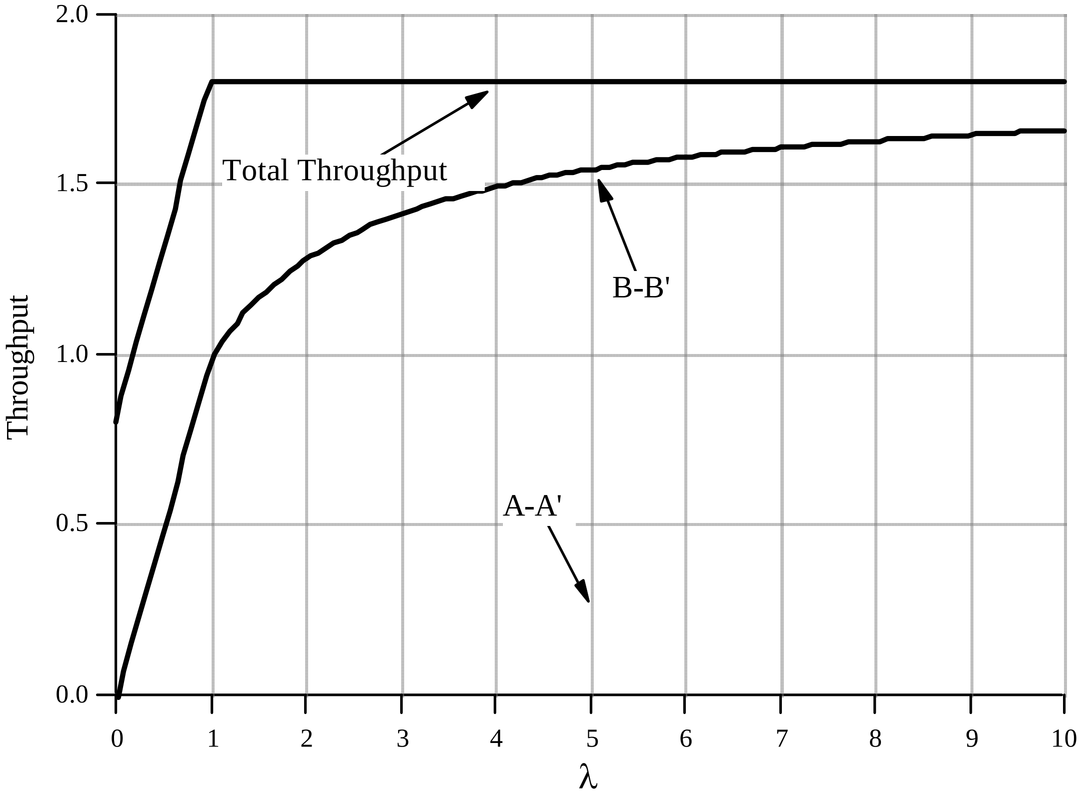
13.1
1. It does not guarantee that a particular node will not be swamped with frames.
2. There is no good way of distributing permits where they are most needed.
3. If a permit is accidentally destroyed, the capacity of the network is inadvertently reduced.

13.2 Yes, but ATM does not include such sliding-window mechanisms. In some cases, a higher-level protocol above ATM will provide such mechanisms, but not in all cases.

14.3 Throughput is as follows:

	$0 \le \lambda \le 1$	$1 \le \lambda \le 10$	$10 \le \lambda$
A-A' traffic	0.8	$(1.8 \times 0.8) / (0.8 + \lambda)$	$1.44 / 10.8 = 0.13$
B-B' traffic	λ	$1.8\lambda / (0.8 + \lambda)$	$18 / 10.8 = 1.67$
Total throughput	$0.8 + \lambda$	1.8	1.8

The plot:



For $1 \le \lambda \le 10$, the fraction of throughput that is A-A' traffic is $0.8 / (0.8 + \lambda)$.

13.4 The terms are roughly similar, but the mathematical definitions differ, so one would expect that the practical application of the two sets of concepts would produce different results.

13.5 The average queue size over the previous cycle and the current cycle is calculated. This value is the threshold. By averaging over two cycles instead of just monitoring current queue length, the system avoids reacting to temporary surges that would not necessarily produce congestion. The average queue length may be computed by determining the area (product of queue size and time interval) over the two cycles and dividing by the time of the two cycles

CHAPTER 14
CELLULAR WIRELESS NETWORKS

ANSWERS TO QUESTIONS

14.1 Hexagon

14.2 For frequency reuse in a cellular system, the same set of frequencies are used in multiple cells, with these cells separated from one another by enough distance to avoid interference.

14.3 Adding new channels: Typically, when a system is set up in a region, not all of the channels are used, and growth and expansion can be managed in an orderly fashion by adding new channels. Frequency borrowing: In the simplest case, frequencies are taken from adjacent cells by congested cells. The frequencies can also be assigned to cells dynamically. Cell splitting: In practice, the distribution of traffic and topographic features is not uniform, and this presents opportunities of capacity increase. Cells in areas of high usage can be split into smaller cells. Cell sectoring: With cell sectoring, a cell is divided into a number of wedge-shaped sectors, each with its own set of channels, typically 3 or 6 sectors per cell. Each sector is assigned a separate subset of the cell's channels, and directional antennas at the base station are used to focus on each sector. Microcells: As cells become smaller, antennas move from the tops of tall buildings or hills, to the tops of small buildings or the sides of large buildings, and finally to lamp posts, where they form microcells. Each decrease in cell size is accompanied by a reduction in the radiated power levels from the base stations and the mobile units. Microcells are useful in city streets in congested areas, along highways, and inside large public buildings.

14.4 To complete a call to a mobile unit, the base stations in a number of cells will send out a page signal in an attempt to find the mobile unit and make the connection.

14.5 The term *fading* refers to the time variation of received signal power caused by changes in the transmission medium or path(s).

14.6 Diffraction occurs at the edge of an impenetrable body that is large compared to the wavelength of the radio wave. The edge in effect become a source and waves radiate in different directions from the edge, allowing a beam to bend around an obstacle. If the size of an obstacle is on the order of the wavelength of the signal or less, scattering occurs. An incoming signal is scattered into several weaker outgoing signals in unpredictable directions.

14.7 Fast fading refers to changes in signal strength between a transmitter and receiver as the distance between the two changes by a small distance of about one-half a wavelength. Slow fading refers to changes in signal strength between a transmitter and receiver as the distance between the two changes by a larger distance, well in excess of a wavelength.

14.8 Flat fading, or nonselective fading, is that type of fading in which all frequency components of the received signal fluctuate in the same proportions simultaneously. Selective fading affects unequally the different spectral components of a radio signal.

14.9 Digital traffic channels: The most notable difference between the two generations is that first generation systems are almost purely analog, where as second generation systems are digital. In particular, the first generation systems are designed to support voice channels using FM; digital traffic is supported only by the use of a modem that converts the digital data into analog form. Second generation systems provide digital traffic channels. These readily support digital data; voice traffic is first encoded in digital form before transmitting. Of course, for second generation systems, the user traffic (data or digitized voice) must be converted to an analog signal for transmission between the mobile unit and the base station. Encryption: Because all of the user traffic, as well as control traffic, is digitized in second generation systems, it is a relatively simple matter to encrypt all of the traffic to prevent eavesdropping. All second generation systems provide this capability, whereas first generation systems send user traffic in the clear, providing no security. Error detection and correction: The digital traffic stream of second generation systems also lends itself to the use of error detection and correction techniques. The result can be very clear voice reception. Channel access: In first generation systems, each cell supports a number of channels. At any given time a channel is allocated to only one user. Second generation systems also provide multiple channels per cell, but each channel is dynamically shared by a number of users using time division multiple access (TDMA) or code division multiple access (CDMA).

14.10 Frequency diversity: Because the transmission is spread out over a larger bandwidth, frequency-dependent transmission impairments, such as noise bursts and selective fading, have less effect on the signal. Multipath resistance: The chipping codes used for CDMA not only exhibit low cross-correlation but also low autocorrelation. Therefore, a version of the signal that is delayed by more than one chip interval does not interfere with the dominant signal as much as in other multipath environments. Privacy: Because spread spectrum is obtained by the use of noise-like signals, where each user has a unique code, privacy is inherent. Graceful degradation: With FDMA or TDMA, a fixed number of users can simultaneously access the system. However, with CDMA, as more users simultaneously access the system, the noise level and hence the error rate increases; only gradually does the system degrade to the point of an unacceptable error rate.

14.11 Self-jamming: Unless all of the mobile users are perfectly synchronized, the arriving transmissions from multiple users will not be perfectly aligned on chip boundaries. Thus the spreading sequences of the different users are not orthogonal and there is some level of cross-correlation. This is distinct from either TDMA or FDMA, in which for reasonable time or frequency guardbands, respectively, the received signals are orthogonal or nearly so. Near-far problem: Signals closer to the receiver are received with less attenuation than signals farther away. Given the lack of complete orthogonality, the transmissions from the more remote mobile units may be more difficult to recover. Thus, power control techniques are very important in a CDMA system.

14.12 Voice quality comparable to the public switched telephone network; 144 kbps data rate available to users in high-speed motor vehicles over large areas; 384 kbps available to pedestrians standing or moving slowly over small areas; Support (to be phased in) for 2.048 Mbps for office use; Symmetrical and asymmetrical data transmission rates; Support for both packet switched and circuit switched data services; An adaptive interface to the Internet to reflect efficiently the common asymmetry between inbound and outbound traffic; More efficient use of the available spectrum in general; Support for a wide variety of mobile equipment; Flexibility to allow the introduction of new services and technologies

Answers to Problems

14.1 a. We have the number of clusters M = 16; bandwidth assigned to cluster B_{CL} = 40 MHz; bandwidth required for each two-way channel b_{ch} = 60 kHz. The total number of simultaneous calls that can be supported by the system is $k_{SYS} = MB_{CL}/b_{ch}$ = 10,666 channels

b. Total number of channels available is K = B_{CL}/b_{ch} = 666. For a frequency reuse factor N, each cell can use $k_{CE} = K/N$ channels.

For N = 4, k_{CE} = 166 channels

For N = 7, k_{CE} = 95 channels

For N = 12, k_{CE} = 55 channels

For N = 19, k_{CE} = 35 channels

c. For N = 4, area = 64 cells; For N = 7, area = 112 cells; For N = 12, area = 192 cells; For N = 19, area = 304 cells.

d. From part b, we know the number of channels that can be carried per cell for each system. The total number of channels available is just 100 times that number, for a result of 16600, 9500, 5500, 3500, respectively. Source: [CARN99]

14.2 a. Steps a and b are the same. The next step is placing the call over the ordinary public switched telephone network (PSTN) to the called subscriber. Steps d, e, and f are the same except that only the mobile unit can be involved in a handoff.

b. Instead of steps a, b, and c, the process starts with a call coming in from the PSTN to an MTSO. From there, steps c, d, e, and f are the same except that only the mobile unit can be involved in a handoff.

14.3 a. The total number of available channels is K = 33000 / 50 = 660. For a frequency reuse factor N, each cell can use $k_{CE} = K / N$ channels.

For $N = 4$, $k_{CE} = 165$ channels

For $N = 7$, $k_{CE} = 94$ channels

For $N = 12$, $k_{CE} = 55$ channels

b. 32 MHz is available for voice channels for a total of 640 channels.
For $N = 4$, we can have 160 voice channels and one control channel per cell
For $N = 7$, we can have 4 cells with 91 voice channels and 3 cells with 92 voice channels, and one control channel per cell.
For $N = 12$, we can have 8 cells with 53 voice channels and 4 cells with 54 voice channels, and one control channel per cell. Source: [RAPP96]

14.4 $(12.5 \times 10^6 - 2(10 \text{ x } 10^3) / (30 \times 10^3) = 416$

14.5 The amount of bandwidth allocated to voice channels (B_cN_t) must be no greater than the total bandwidth (B_w). Therefore $\eta_a \leq 1$.

CHAPTER 15
LOCAL AREA NETWORK OVERVIEW

ANSWERS TO QUESTIONS

15.1 Computer room networks require very high data rates and usually are concerned with transfer of large blocks of data.

15.2 Backend LAN: Backend networks are used to interconnect large systems such as mainframes, supercomputers, and mass storage devices. The key requirement here is for bulk data transfer among a limited number of devices in a small area. High reliability is generally also a requirement. SAN: A SAN is a separate network to handle storage needs. The SAN detaches storage tasks from specific servers and creates a shared storage facility across a high-speed network. Backbone LAN: A backbone LAN is a high-capacity LAN used to interconnect a number of lower-capacity LANs.

15.3 Network topology refers to the way in which the end parts or stations attached to the network are interconnected.

15.4 Bus: all stations attach, through appropriate hardware interfacing known as a tap, directly to a linear transmission medium, or bus. Full-duplex operation between the station and the tap allows data to be transmitted onto the bus and received from the bus. A transmission from any station propagates the length of the medium in both directions and can be received by all other stations. At each end of the bus is a terminator, which absorbs any signal, removing it from the bus. Tree: a generalization of the bus topology. The transmission medium is a branching cable with no closed loops. The tree layout begins at a point known as the *headend*. One or more cables start at the headend, and each of these may have branches. The branches in turn may have additional branches to allow quite complex layouts. Again, a transmission from any station propagates throughout the medium and can be received by all other stations. Ring: the network consists of a set of *repeaters* joined by point-to-point links in a closed loop. Each station attaches to the network at a repeater and can transmit data onto the network through the repeater. Star: each station is directly connected to a common central node. Typically, each station attaches to a central node via two point-to-point links, one for transmission and one for reception.

15.5 To develop LAN standards.

15.6 No single technical approach will satisfy all requirements. Requirements with respect to cost, data rate, and range dictate a variety of technical alternatives.

15.7 Unacknowledged connectionless service: This service is a datagram-style service. It is a very simple service that does not involve any of the flow- and error-control mechanisms. Thus, the delivery of data is not guaranteed. Connection-mode service: This service is similar to that offered by HDLC. A logical connection is set up between two users exchanging data, and flow control and error control are provided. Acknowledged connectionless service: This is a cross between the previous two services. It provides that datagrams are to be acknowledged, but no prior logical connection is set up.

15.8 Type 1 operation supports unacknowledged connectionless service. There is no acknowledgment, flow control, or error control. Type 2 operation supports connection-mode service, using mechanisms similar to HDLC. Type 3 operation supports acknowledged connectionless service. Each transmitted PDU is acknowledged using a stop-and-wait technique.

15.9 (1) On transmission, assemble data into a frame with address and error-detection fields. (2) On reception, disassemble frame, and perform address recognition and error detection. (3) Govern access to the LAN transmission medium.

15.10 For a bridge that connects LANs A and B: (1) Read all frames transmitted on A and accept those addressed to any station on B. (2) Using the medium access control protocol for B, retransmit each frame on B. (3) Do the same for B-to-A traffic.

15.11 For any connected graph, consisting of nodes and edges connecting pairs of nodes, there is a spanning tree of edges that maintains the connectivity of the graph but contains no closed loops.

15.12 With a hub, only one attached station may transmit at a time. A switch can accommodate multiple simultaneous transmissions.

15.13 Store-and-forward switch: The layer 2 switch accepts a frame on an input line, buffers it briefly, and then routes it to the appropriate output line. Cut-through switch: The layer 2 switch takes advantage of the fact that the destination address appears at the beginning of the MAC (medium access control) frame. The layer 2 switch begins repeating the incoming frame onto the appropriate output line as soon as the layer 2 switch recognizes the destination address.

Answers to Problems

15.1 HDLC has only one address field. In a LAN, any station may transmit to any other station. The receiving station needs to see its own address in order to know that the data is intended for itself. It also needs to see the sending address in order to reply.

15.2 Each individual character could be sent out as a separate packet, resulting in tremendous overhead. This problem could be overcome by buffering characters and only sending out blocks of characters.

15.3 a. $T = (8 \times 10^6 \text{ bits}) / (64 \times 10^3 \text{ bps}) = 125$ seconds

b. Transfer consists of a sequence of cycles. One cycle consists of:

Data Packet = Data Packet Transmission Time + Propagation Time
ACK Packet = ACK Packet Transmission Time + Propagation Time

Define:

C = Cycle time
Q = data bits per packet
T = total time required
T_d = Data Packet Transmit Time
T_a = ACK Packet Transmit Time
T_p = Propagation Time

Then

$T_p = (D) / (200 \times 10^6 \text{m} / \text{sec})$
$T_a = (88 \text{ bits}) / (B \text{ bps})$
$T_d = (P \text{ bits}) / (B \text{ bps})$
$T = (8 \times 10^6 \text{ bits} \times C \text{ sec} / \text{cycle}) / (Q \text{ bits} / \text{cycle})$
$C = T_a + T_d + 2T_p$
$Q = P - 80$

b1 $C = (88) / (10^6) + (256) / (10^6) + (2 \times 10^3) / (200 \times 10^6) = 354 \times 10^{-6}$
$Q = 176$
$T = (8 \times 10^6 \times 354 \times 10^{-6}) / (176) = 16$ sec

b2 $C = (88) / (10 \times 10^6) + (256) / (10 \times 10^6) + (2 \times 10^3) / (200 \times 10^6) = 44.4 \times 10^{-6}$
$Q = 176$
$T = (8 \times 10^6 \times 44.4 \times 10^{-6}) / (176) = 2$ sec

b3 $C = (88) / (10^6) + (256) / (10^6) + (2 \times 10^4) / (200 \times 10^6) = 444 \times 10^{-6}$
$Q = 176$
$T = (8 \times 10^6 \times 444 \times 10^{-6}) / (176) = 20$ sec

b4 $C = (80) / (50 \times 10^6) + (10^4) / (50 \times 10^6) + (2 \times 10^3) / (200 \times 10^6) = 211.6 \times 10^{-6}$
$Q = 9{,}920$
$T = (8 \times 10^6 \times 211.6 \times 10^{-6}) / (9920) = 0.17$ seconds

c. Define

Tr = Total repeater Delay = N/B

Then

$C = Td + 2Tp + Tr$

$Q = P - 80$

$T = (8 \times 10^6 \times C)/(Q)$

We show results for N = 100

c1 $C = (256)/(10^6) + (2 \times 10^3)/(200 \times 10^6) + (100)/(10^6) = 366 \times 10^{-6}$

$T = (8 \times 10^6 \times 366 \times 10^{-6})/(176) = 16.6$ sec

c2 $C = (256)/(10 \times 10^6) + (2 \times 10^3)/(200 \times 10^6) + (100)/(10 \times 10^6) = 45.6 \times 10^{-6}$

$T = (8 \times 10^6 \times 45.6 \times 10^{-6})/(176) = 2.1$ sec

c3 $C = (256)/(10^6) + (2 \times 10^4)/(200 \times 10^{-6}) + (100)/(10^6) = 456 \times 10^{-6}$

$T = (8 \times 10^6 \times 456 \times 10^{-6})/(176) = 20.7$ sec

c4 $C = (10^4)/(50 \times 10^6) + (2 \times 10^3)/(200 \times 10^6) + (100)/(50 \times 10^6) = 212 \times 10^{-6}$

$T = (8 \times 10^6 \times 212 \times 10^{-6})/(9920) = 0.17$ sec

15.4 a. Assume a mean distance between stations of 0.375 km. This is an approximation based on the following observation. For a station on one end, the average distance to any other station is 0.5 km. For a station in the center, the average distance is 0.25 km. With this assumption, the time to send equals transmission time plus propagation time.

$$T = \frac{10^3 \text{ bits}}{10^7 \text{ bps}} + \frac{375 \text{ m}}{200 \times 10^6 \text{ m/sec}} = 102\mu\text{sec}$$

b.

$$T_{\text{interfere}} = \frac{375 \text{ m}}{200 \times 10^6 \text{ m/sec}} = 1.875\mu\text{sec}$$

$$T_{\text{interfere(bit-times)}} = 10^7 \times 1.875 \times 10^{-6} = 18.75 \text{ bit} - \text{times}$$

15.5 a. Again, assume a mean distance between stations of 0.375 km.

$$T = \frac{10^3 \text{ bits}}{10^8 \text{ bps}} + \frac{375 \text{ m}}{200 \times 10^6 \text{ m/sec}} = 12\mu\text{sec}$$

b.

$$T_{\text{interfere}} = \frac{375\text{ m}}{200 \times 10^6 \text{ m/sec}} = 1.875\mu\text{sec}$$

$$T_{\text{interfere(bit-times)}} = 10^8 \times 1.875 \times 10^{-6} = 187.5\text{ bit} - \text{times}$$

15.6 a. $\frac{1\text{ bit}}{1\text{ Mbps}} = 1\mu\text{sec, equivalent to } 200\text{ m}$

b. $\frac{1\text{ bit}}{40\text{ Mbps}} = 0.025\mu\text{sec, equivalent to } 5\text{ m}$

15.7 The communication stations perform the function of a bridge.

15.8 a. The event [A does not fail] is the event [no link fails and no repeater fails]

Pr[A does not fail] = Pr[no link fails] × Pr[no repeater fails]
Pr[A fails] = 1 – (1 – P1)300 × (1 – Pr)300

b. For B to fail completely, all three rings must fail:
Pr[A fails] = [1 – (1 – P1)100 × (1 – Pr)100]3

c. A station will find network A unavailable if A has failed:
Pr[A found unavailable] = 1 – (1 – P1)300 × (1 – Pr)300
A station will find B unavailable if its ring has failed.
Pr[B found unavailable] = 1 – (1 – P1)100 × (1 – Pr)100

d. For network A, two stations will be unable to communicate if A fails:
Pr[no communication] = 1 – (1 – P1)300 × (1 – Pr)300

For network B, first compute the probability that the 2 stations, x and y, are not on the same ring (call the subrings, B1, B2, B3)

P1 = Pr[x, y on same ring]
= Pr[x on B1, y on B1] + Pr[x on B2, y on B2] + Pr[x on B2, y on B2]
= Pr[x on B_1]Pr[y on B_1] + Pr[x on B_2]Pr[y on B_2] + Pr[x on B_3]Pr[y on B_3]
= 1/3 × 1/3 + 1/3 × 1/3 + 1/3 × 1/3 = 1/3

P_2 = Pr[x, y not on same ring] = 2/3

P_3 = Pr[no communications, given on same ring]
= $1 - (1 - P_1)^{100} \times (1 - P_r)^{100}$

P_4 = Pr[no communications, given on different rings]

1- P_4 = Pr[communications, given on different rings]
= Pr[x's ring does not fail]Pr[y's ring does not fail]Pr[bridge does not fail]
= $(1 - P_1)^{100} \times (1 - P_r)^{100} \times (1 - P_1)^{100} \times (1 - P_r)^{100} \times (1 - P_b)$

$P_4 = 1 - [(1 - P_1)^{200} \times (1 - P_r)^{200} \times (1 - P_b)]$

Pr[no communications] = $P_1P_3 + P_2P_4$

e. (a) Pr[A fails] = $1 - (0.99)^{600} = 0.998$

(b) Pr[B fails] = $[1 - (0.99)^{200}]^3 = (0.87)^3 = 0.66$

(c) Pr[A found unavailable] = 0.998

Pr[B found unavailable] = $1 - (0.99)^{200} = 0.87$

(d) Pr[no communications on A] = 0.998

$P_1 = 0.33$

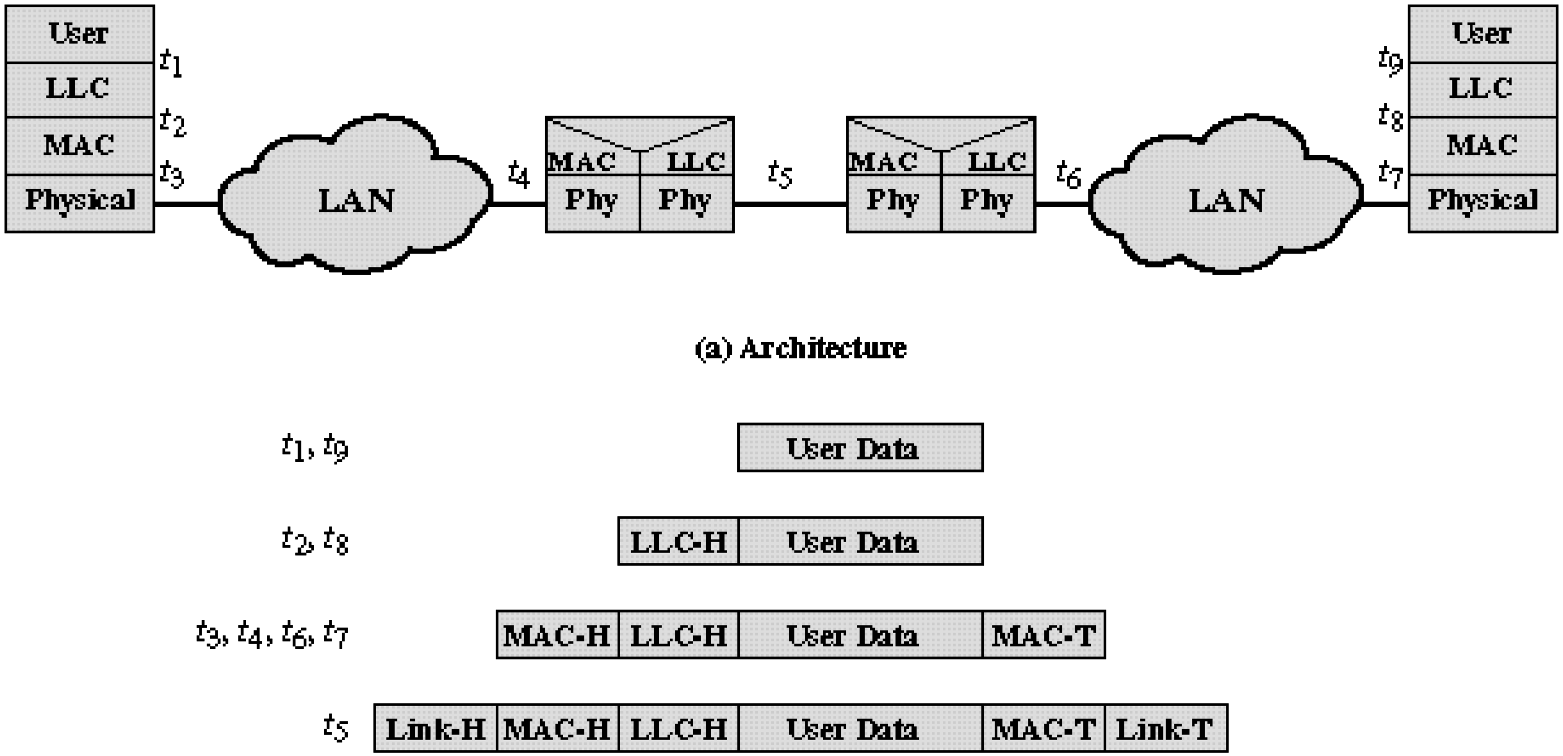
$P_2 = 0.67$

$P_3 = 1 - (0.99)^{200} = 0.87$

$P_4 = 1 - (0.99)^{401} = 0.98$

Pr[no communications on B] = (0.33)(0.87) + (0.67)(0.98) = 0.94

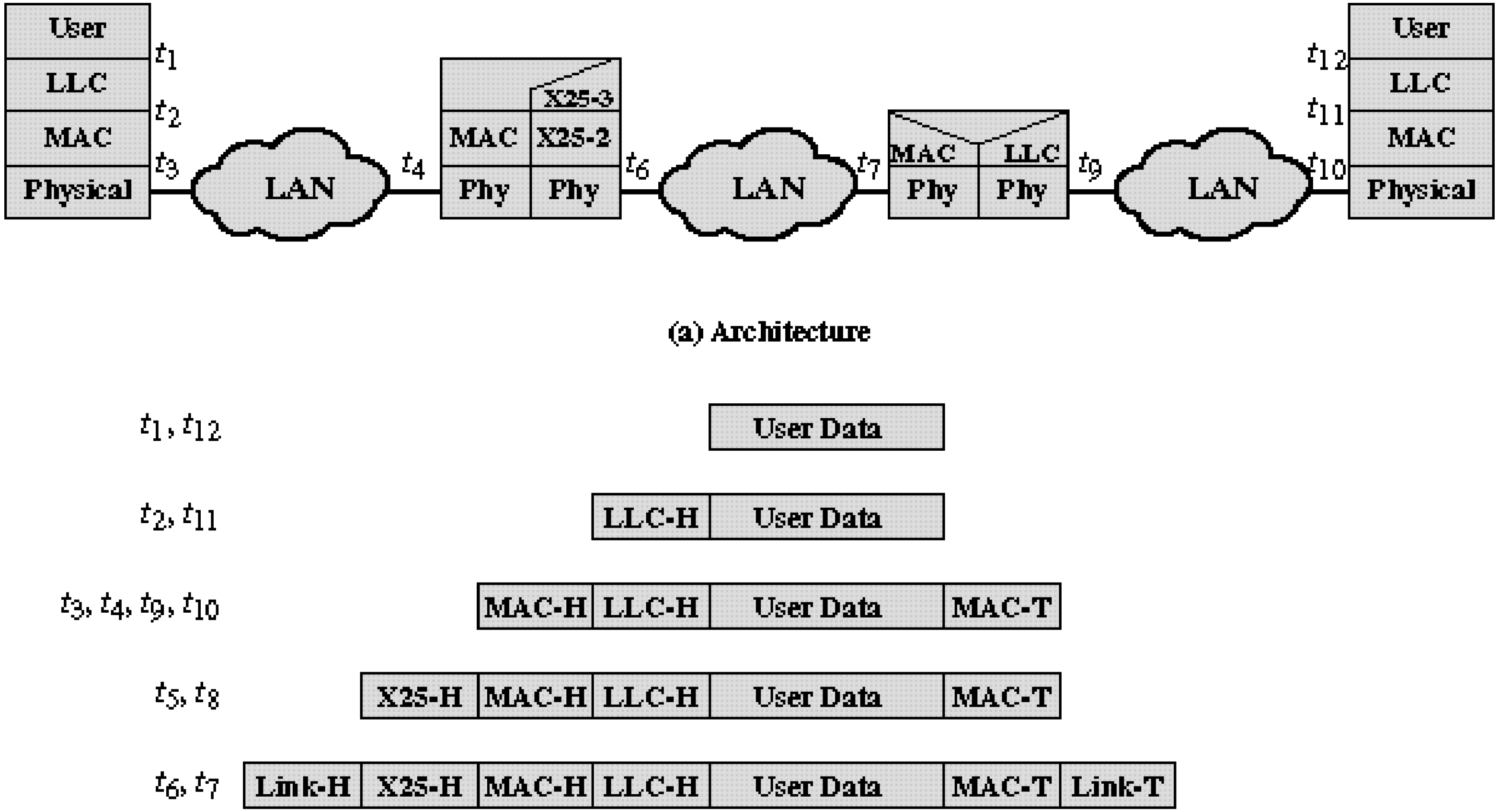
15.9a



(a) Architecture

(b) Operation

b



(a) Architecture

(b) Operation

15.10

CENTRAL ROUTING DIRECTORY

		Source LAN						
		A	B	C	D	E	F	G
	A	—	101	102	103	107	105	106
	B	101	—	102	103	104	105	106
Destination	C	102	101	—	103	107	105	106
LAN	D	101	103	102	—	104	105	106
	E	107	104	102	103	—	105	106
	F	102	101	105	103	107	—	106
	G	102	101	106	103	107	105	—

Bridge 101 Table			
from LAN A		from LAN B	
Dest	Next	Dest	Next
B	B	A	A
C	—	C	A
D	B	D	—
E	—	E	—
F	—	F	A
G	—	G	A

Bridge 102 Table			
from LAN A		from LAN C	
Dest	Next	Dest	Next
B	—	A	A
C	C	B	A
D	—	D	A
E	—	E	A
F	C	F	—
G	C	G	—

Bridge 103 Table			
from LAN B		from LAN D	
Dest	Next	Dest	Next
A	—	A	B
C	—	B	B
D	D	C	B
E	—	E	B
F	—	F	B
G	—	G	B

Bridge 104 Table			
from LAN B		from LAN E	
Dest	Next	Dest	Next
A	—	A	—
C	—	B	B
D	—	C	—
E	E	D	B
F	—	F	—
G	—	G	—

Bridge 105 Table			
from LAN C		from LAN F	
Dest	Next	Dest	Next
A	—	A	C
B	—	B	C
D	—	C	C
E	—	D	C
F	F	E	C
G	—	G	C

Bridge 106 Table			
from LAN C		from LAN G	
Dest	Next	Dest	Next
A	—	A	C
B	—	B	C
D	—	C	C
E	—	D	C
F	—	E	C
G	G	F	C

Bridge 107 Table			
from LAN A		from LAN E	
Dest	Next	Dest	Next
B	—	A	A
C	—	B	—
D	—	C	A
E	E	D	—
F	—	F	A
G	—	G	A

CHAPTER 16
HIGH-SPEED LANs

ANSWERS TO QUESTIONS

16.1 A collection of multiple, centralized servers.

16.2 Nonpersistent: If the medium is idle, transmit; if the medium is busy, wait an amount of time drawn from a probability distribution and then check the medium. 1-persistent: If the medium is idle, transmit; if the medium is busy continue to listen until the channel is sensed idle; then transmit immediately. *p*-persistent: If the medium is idle, transmit with probability *p*, and delay one time unit with probability (1 – *p*); if the medium is busy, continue to listen until the channel is idle and repeat.

16.3 Carrier sense multiple access with collision detection (CSMA/CD) is a form of medium access control in which a station listens to the medium to try to see if another transmission is in progress. If the medium appears idle, the station transmits. If a collision occurs, the workstations wait a random amount of time before trying again.

16.4 A station will attempt to transmit repeatedly in the face of repeated collisions. For the first 10 retransmission attempts, the mean value of the random delay is doubled. This mean value then remains the same for 6 additional attempts. After 16 unsuccessful attempts, the station gives up and reports an error.

16.5 Shielded twisted pair, high quality unshielded twisted pair (Category 5), relatively low quality unshielded twisted pair (Category 3) and optical fiber are transmission medium options for Fast Ethernet.

16.6 Fast Ethernet differs from 10BASE-T in two important ways. Two physical links are used between nodes -- one for transmitting, one for receiving (four links are used for Fast Ethernet using lower quality unshielded twisted pair (Category 3)). A different coding scheme 4B/5B-NRZI is used rather than Manchester Coding.

16.7 With full-duplex operation, a station can transmit and receive simultaneously.

16.8 Data insertion, data reception, and data removal.

16.9 With dedicated token ring, the token is not necessary because a central layer-2 switch is used with full-duplex operation.

16.10 The Fibre Channel standard is organized into five levels:

FC-0: The Physical Interface and Media Level allows a variety of physical media and data rates.

FC-1: The Transmission Protocol Level define the signal encoding technique used for transmission and for synchronization across the point-to-point link.

FC-2: The Framing Protocol Level deals with the transmission of data between ports in the form of frames, similar to other data link control protocols.

FC-3: The Common Services Level provides a set of services that are common across multiple ports of a node.

FC-4: The Mapping Level defines the mapping of various channel and network protocols to FC-0 through FC-2.

16.11 Fabric topology, point-to-point topology, and arbitrated loop topology.

16.12 Under heavy loads, CSMA/CD's performance declines because of the number of collisions. The token ring functions more efficiently under a heavy load.

ANSWERS TO PROBLEMS

16.1 The fraction of slots wasted due to multiple transmission attempts is equal to the probability that there will be 2 or more transmission attempts in a slot.

Pr[2 or more attempts] = 1 – Pr[no attempts] – Pr[exactly 1 attempt]

$$= 1 - (1 - p)^N - Np(1 - p)^{N-1}$$

16.2 $$E\left[\text{number of iterations}\right] = \sum_{i=1}^{\infty}\left(i \times \Pr\left[\text{exactly } i \text{ iterations of step } 1\right]\right)$$

$$P_i = \left(1 - PF_i\right) \times \prod_{j=1}^{i-1} PF_j$$

Pr [exactly 1 iteration of step 1] = p

Pr [exactly 2 iterations of step 1] = (1 – p)p

Pr [exactly 3 iterations of step 1] = $(1 - p)^2p$

Pr [exactly i iterations of step 1] = $(1 - p)^{i-1}p$

$$E\left[\text{number of iterations}\right] = p \times \sum_{i=1}^{\infty}\left(i \times \left[1 - p\right]^{i-1}\right)$$

$$= \frac{p}{\left(1 - (1 - p)\right)^2} = \frac{1}{p}$$

If a stations transmits, on average on iteration number (1/p), then it must wait for the first $\left(\frac{1}{p}-1\right)$ iterations, which is an amount of time $T\times\left(\frac{1}{p}-1\right)$.

16.3 Define

PF_i = Pr[fail on attempt i]

P_i = Pr[fail on first (i – 1) attempts, succeed on ith]

Then, the mean number of retransmission attempts before one station successfully retransmits is given by:

$$\mathrm{E}\left[\text{retransmissions}\right]=\sum_{i=1}^{\infty} iP_i$$

For two stations:

PF_i = 1/2^K where K = MIN[i, 10]

$$P_i=\left(1-PF_i\right)\times\prod_{j=1}^{i-1}PF_j$$

PF_1 = 0.5

P_1 = 0.5

PF_2 = 0.25

P_2 = 0.375

PF_3 = 0.125

P_3 = 0.109

PF_4 = 0.0625

P_4 = 0.015

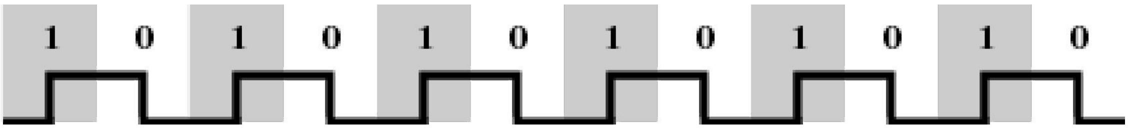
The remaining terms are negligible

E[retransmissions] = 1.637

16.4 The preamble pattern is

10101010 10101010 10101010 10101010 10101010 10101010 10101010

transmitted in order from left to right. The Manchester pattern produced looks, in part, like so:



The pattern appears as a periodic waveform on the medium, which enables bit synchronization.

16.5 It may actually take *longer* with acknowledgment frames. If the source station removes the frame, then the time for one cycle of activity is:

$$T_t = T_f + T_{360}$$

where

T_t = total time for one transmission plus acknowledgment

T_f = time for sender to transmit one frame

T_{360} = propagation time completely around the ring

If the destination station removes the frame, then the time for one cycle of activity is:

$$\begin{aligned} T_t &= T_f + T_x + T_{ack} + T_{360-x} \\ &= T_f + T_{360} + T_{ack} \end{aligned}$$

where

T_{ack} = time for receiver to transmit an acknowledgment frame.

T_x = propagation time from sender to receiver

T_{360-x} = propagation time from receiver back to sender.

The difference is that, in the first case, the receiver acknowledges a frame by setting one or more bits while the frame goes by. In the second case, the receiver absorbs the entire frame, and then sends an acknowledgment frame.

16.6 With slotted ring, a station may transmit when an empty slot goes by (absence of data), and multiple stations may transmit at the same time. With token ring, a station may transmit when a token goes by (presence of data), and only one station may transmit at a time. Also, with token ring, you are not constrained to a fixed-size frame.

16.7 First compute the propagation time around the ring.

$$\begin{aligned} T_{prop} &= 500 \text{ repeaters} \times 10^{-7} \text{ sec/repeater} + (10^4\text{m})/(200 \times 10^6\text{m/sec}) \\ &= 100 \times 10^{-6} \text{ sec} \end{aligned}$$

Next compute the "bit length" of the ring:

$$L_b = 100 \times 10^{-6}\text{sec} \times 10^7\text{bits/sec} = 1000 \text{ bits}$$

We have 1000 bits/(37 bits/slot) = 27.03

For an integral number, there are 27 slots on the ring.

16.8 With pure ternary signaling, each signal element can take on one of three states, so that the information-carrying capacity of a signal element is

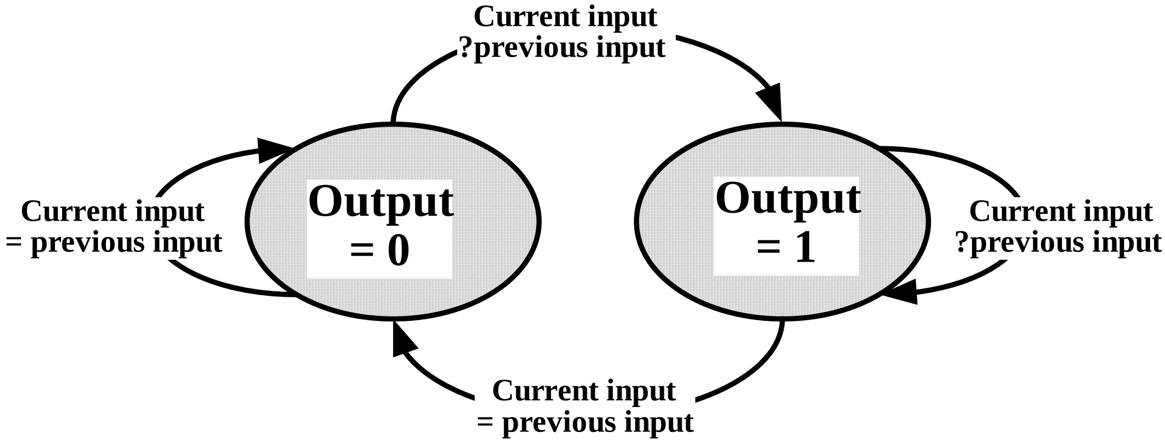
$$\log_2 (3) = 1.585 \text{ bits/baud}$$

So that the effective data rate for a signaling rate of 25 Mbaud is

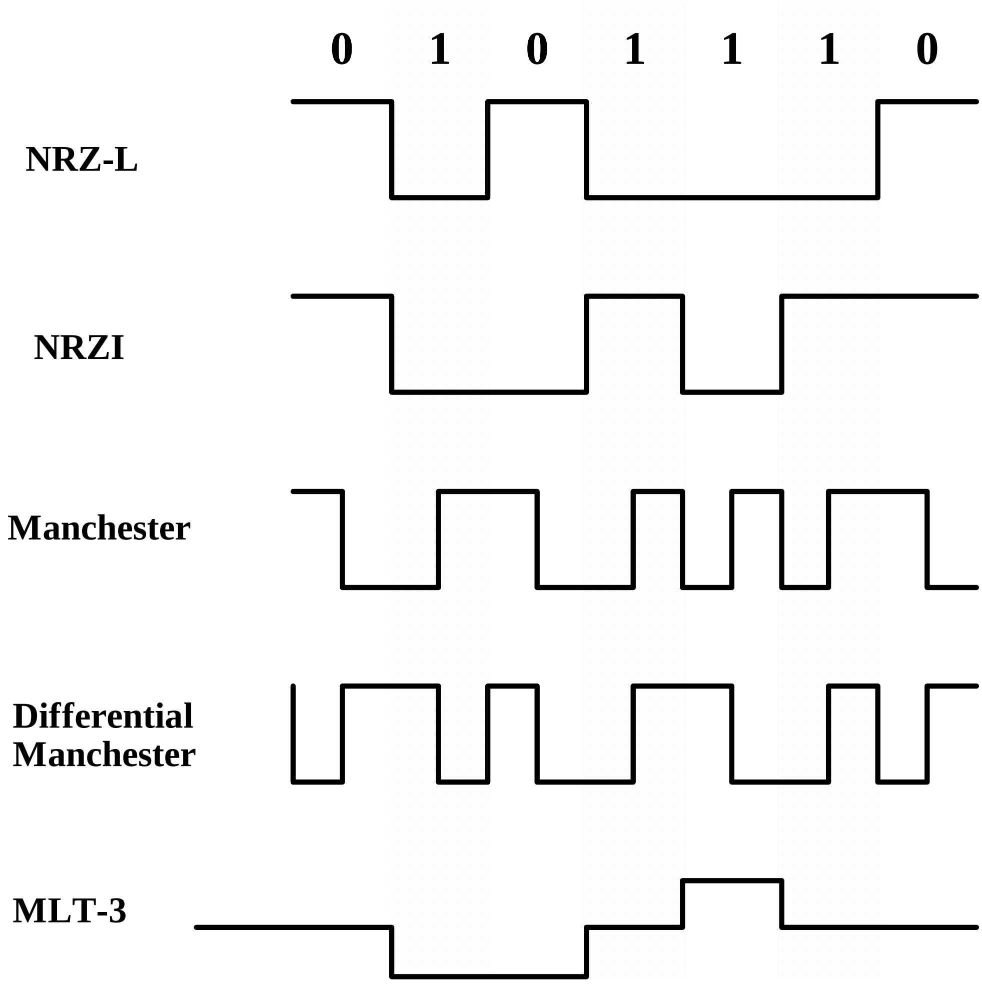
$$1.585 \times 25 = 39.62 \text{ Mbps}$$

16.9 The receiver performs the same cumulative weight algorithm as the transmitter and reverses the actions of the transmitter in this regard.

16.10


Current input
?previous input
Current input
= previous input
Output
= 0
Output
= 1
Current input
?previous input
Current input
= previous input

16.11


0
1
0
1
1
1
0
NRZ-L
NRZI
Manchester
Differential
Manchester
MLT-3

CHAPTER 17
WIRELESS LANs

ANSWERS TO QUESTIONS

17.1 LAN extension: a wireless LAN integrated with a wired LAN to extend the coverage area of the LAN complex; cross-building interconnect: wireless point-to-point link two LANs; nomadic access: provides a wireless link between a LAN hub and a mobile data terminal equipped with an antenna, such as a laptop computer or notepad computer; ad hoc network: a peer-to-peer network (no centralized server) set up temporarily to meet some immediate need.

17.2 Throughput: The medium access control protocol should make as efficient use as possible of the wireless medium to maximize capacity. Number of nodes: Wireless LANs may need to support hundreds of nodes across multiple cells. Connection to backbone LAN: In most cases, interconnection with stations on a wired backbone LAN is required. For infrastructure wireless LANs, this is easily accomplished through the use of control modules that connect to both types of LANs. There may also need to be accommodation for mobile users and ad hoc wireless networks. Service area: A typical coverage area for a wireless LAN has a diameter of 100 to 300 m. Battery power consumption: Mobile workers use battery-powered workstations that need to have a long battery life when used with wireless adapters. This suggests that a MAC protocol that requires mobile nodes to monitor access points constantly or engage in frequent handshakes with a base station is inappropriate. Typical wireless LAN implementations have features to reduce power consumption while not using the network, such as a sleep mode. Transmission robustness and security: Unless properly designed, a wireless LAN may be interference prone and easily eavesdropped. The design of a wireless LAN must permit reliable transmission even in a noisy environment and should provide some level of security from eavesdropping. Collocated network operation: As wireless LANs become more popular, it is quite likely for two or more wireless LANs to operate in the same area or in some area where interference between the LANs is possible. Such interference may thwart the normal operation of a MAC algorithm and may allow unauthorized access to a particular LAN. License-free operation: Users would prefer to buy and operate wireless LAN products without having to secure a license for the frequency band used by the LAN. Handoff/roaming: The MAC protocol used in the wireless LAN should enable mobile stations to move from one cell to another. Dynamic configuration: The MAC addressing and network management aspects of the LAN should permit dynamic and automated addition, deletion, and relocation of end systems without disruption to other users.

17.3 Single-cell wireless LAN: all of the wireless end systems are within range of a single control module. Multiple-cell wireless LAN: there are multiple control modules interconnected by a wired LAN; each control module supports a number of wireless end systems within its transmission range.

17.4 (1) The spectrum for infrared is virtually unlimited, which presents the possibility of achieving extremely high data rates. (2) The infrared spectrum is unregulated worldwide, which is not true of some portions of the microwave spectrum. (3) Infrared light is diffusely reflected by light-colored objects; thus it is possible to use ceiling reflection to achieve coverage of an entire room. (4) Infrared light does not penetrate walls or other opaque objects. This has two advantages: First, infrared communications can be more easily secured against eavesdropping than microwave; and second, a separate infrared installation can be operated in every room in a building without interference, enabling the construction of very large infrared LANs. (5) Another strength of infrared is that the equipment is relatively inexpensive and simple.

17.5 (1) Many indoor environments experience rather intense infrared background radiation, from sunlight and indoor lighting. This ambient radiation appears as noise in an infrared receiver, requiring the use of transmitters of higher power than would otherwise be required and also limiting the range. (2) Increases in transmitter power are limited by concerns of eye safety and excessive power consumption.

17.6 The transmitted signal can be focused and aimed (as in a remote TV control); it can be radiated omnidirectionally; or it can be reflected from a light-colored ceiling.

17.7 An access point functions as a bridge to enable the linking of multiple separate 802.11 wireless LANs. A portal provides an interconnection point between an 802.11 wireless LAN and a wired LAN.

17.8 It may or may not be.

17.9 Association: Establishes an initial association between a station and an AP. Authentication: Used to establish the identity of stations to each other. Deauthentication: This service is invoked whenever an existing authentication is to be terminated. Disassociation: A notification from either a station or an AP that an existing association is terminated. A station should give this notification before leaving an ESS or shutting down. Distribution: used by stations to exchange MAC frames when the frame must traverse the DS to get from a station in one BSS to a station in another BSS. Integration: enables transfer of data between a station on an IEEE 802.11 LAN and a station on an integrated IEEE 802.x LAN. MSDU delivery: delivery of MAC service data units. Privacy: Used to prevent the contents of messages from being read by other than the intended recipient. Reassocation: Enables an established association to be transferred from one AP to another, allowing a mobile station to move from one BSS to another.

17.10 Mobility refers to the types of physical transitions that can be made by a mobile node within an 802.11 environment (no transition, movement from one BSS to another within an ESS, movement from one ESS to another). Association is a service that allows a mobile node that has made a transition to identify itself to the AP within a BSS so that the node can participate in data exchanges with other mobile nodes.

CHAPTER 18
INTERNET PROTOCOLS

ANSWERS TO QUESTIONS

18.1 (1) The communications network may only accept blocks of data up to a certain size. (2) Error control may be more efficient with a smaller PDU size. With smaller PDUs, fewer bits need to be retransmitted when a PDU suffers an error. (3) More equitable access to shared transmission facilities, with shorter delay, can be provided. (4) A smaller PDU size may mean that receiving entities can allocate smaller buffers. (5) An entity may require that data transfer comes to some sort of "closure" from time to time, for checkpoint and restart/recovery operations.

18.2 (1.) Provide a link between networks. At minimum, a physical and link control connection is needed. (2) Provide for the routing and delivery of data between processes on different networks. (3) Provide an accounting service that keeps track of the use of the various networks and routers and maintains status information. (4) Provide the services just listed in such a way as not to require modifications to the networking architecture of any of the constituent networks. This means that the internetworking facility must accommodate a number of differences among networks.

18.3 If intermediate reassembly is not allowed, the datagram must eventually be fragmented to the smallest allowable size along the route. Once the datagram has passed through the network that imposes the smallest-size restriction, the fragments may be unnecessarily small for later networks, degrading performance.

On the other hand, intermediate reassembly requires compute and buffer resources at the intermediate routers. Furthermore, all fragments of a given original datagram would have to pass through the same intermediate node for reassembly, which would prohibit dynamic routing.

18.4 The More bit is used for fragmentation and reassembly. If this bit is 0, then either there has been no fragmentation of this packet or this is the last fragment. If this bit is 1, then this packet has been fragmented and this is not the last fragment. The Don't Fragment bit prohibits fragmentation when set.

18.5 The checksum is formed by taking the ones complement of the 16-bit ones complement addition of all 16-bit words in the header. For purposes of computation, the checksum field is itself initialized to a value of zero.

18.6 Traffic Class (8 bits): Available for use by originating nodes and/or forwarding routers to identify and distinguish between different classes or priorities of IPv6 packets. The first six bits of the traffic class field are referred to as the DS (differentiated services) field, discussed in Chapter 19. The remaining 2 bits are reserved for an ECN (explicit congestion notification) field, currently in the process of standardization. Flow Label (20 bits): May be used by a host to label those packets for which it is requesting special handling by routers within a network. All packets that are to be part of the same flow are assigned the same flow label by the source.

18.7 Unicast: An identifier for a single interface. A packet sent to a unicast address is delivered to the interface identified by that address. Anycast: An identifier for a set of interfaces (typically belonging to different nodes). A packet sent to an anycast address is delivered to one of the interfaces identified by that address (the "nearest" one, according to the routing protocols' measure of distance). Multicast: An identifier for a set of interfaces (typically belonging to different nodes). A packet sent to a multicast address is delivered to all interfaces identified by that address.

18.8 Hop-by-Hop Options header: Defines special options that require hop-by-hop processing. Routing header: Provides extended routing, similar to IPv4 source routing. Fragment header: Contains fragmentation and reassembly information. Authentication header: Provides packet integrity and authentication. Encapsulating Security Payload header: Provides privacy. Destination Options header: Contains optional information to be examined by the destination node.

Answers to Problems

18.1 The IP entity in the source may need the ID and don't-fragment parameters. If the IP source entity needs to fragment, these two parameters are essential. Ideally, the IP source entity should not need to bother looking at the TTL parameter, since it should have been set to some positive value by the source IP user. It can be examined as a reality check. Intermediate systems clearly need to examine the TTL parameter and will need to examine the ID and don't-fragment parameters if fragmentation is desired. The destination IP entity needs to examine the ID parameter if reassembly is to be done and, also the TTL parameter if that is used to place a time limit on reassembly. The destination IP entity should not need to look at the don't-fragment parameter.

18.2 The header is a minimum of 20 octets.

18.3 Possible reasons for strict source routing: (1) to test some characteristics of a particular path, such as transit delay or whether or not the path even exists; (2) the source wishes to avoid certain unprotected networks for security reasons; (3) the source does not trust that the routers are routing properly.

Possible reasons for loose source routing: (1) Allows the source to control some aspects of the route, similar to choosing a long-distance carrier in the telephone network; (2) it may be that not all of the routers recognize all addresses and that for a particular remote destination, the datagram needs to be routed through a "smart" router.

18.4 A buffer is set aside big enough to hold the arriving datagram, which may arrive in fragments. The difficulty is to know when the entire datagram has arrived. Note that we cannot simply count the number of octets (bytes) received so far because duplicate fragments may arrive.

a. Each hole is described by a descriptor consisting of hole.first, the number of the first octet in the hole, relative to the beginning of the buffer, and hole.last, the number of the last octet. Initially, there is a single hole descriptor with hole.first = 1 and hole.last = some maximum value. Each arriving fragment is characterized by fragment.first and fragment.last, which can be calculated from the Length and Offset fields.

1. Select the next hole descriptor from the hole descriptor list. If there are no more entries, go to step eight.
2. If fragment.first is greater than hole.last, go to step one.
3. If fragment.last is less than hole.first, go to step one.
   (If either step two or step three is true, then the newly arrived fragment does not overlap with the hole in any way, so we need pay no further attention to this hole. We return to the beginning of the algorithm where we select the next hole for examination.)
4. Delete the current entry from the hole descriptor list.
   (Since neither step two nor step three was true, the newly arrived fragment does interact with this hole in some way. Therefore, the current descriptor will no longer be valid. We will destroy it, and in the next two steps we will determine whether or not it is necessary to create any new hole descriptors.)
5. If fragment.first is greater than hole.first, then create a new hole descriptor "new-hole" with new-hole.first equal to hole.first, and new-hole.last equal to fragment.first minus one.
   (If the test in step five is true, then the first part of the original hole is not filled by this fragment. We create a new descriptor for this smaller hole.)
6. If fragment.last is less than hole.last and fragment.more fragments is true, then create a new hole descriptor "new-hole", with new-hole.first equal to fragment.last plus one and new-hole.last equal to hole.last.
   (This test is the mirror of step five with one additional feature. Initially, we did not know how long the resembled datagram would be, and therefore we created a hole reaching from zero to (effectively) infinity. Eventually, we will receive the last fragment of the datagram. At this point, that hole descriptor which reaches from the last octet of the buffer to infinity can be discarded. The fragment which contains the last fragment indicates this fact by a flag in the internet header called "more fragments". The test of this bit creating in this statement prevents us from creating a descriptor for the unneeded hole which describes the space from the end of the datagram to infinity.)
7. Go to step one.

8. If the hole descriptor list is now empty, the datagram is now complete. Pass it on to the higher level protocol processor for further handling. Otherwise, return.

b. The hole descriptor list could be managed as a separate list. A simpler technique is to put each hole descriptor in the first octets of the hole itself. The descriptor must contain hole.last plus pointer to the predecessor and successor holes.

18.5 The original datagram includes a 20-octet header and a data field of 4460 octets. The Ethernet frame can take a payload of 1500 octets, so each frame can carry an IP datagram with a 20-octet header and 1480 data octets. Note the 1480 is divisible by 8, so we can use the maximum size frame for each fragment except the last. To fit 4460 data octets into frames that carry 1480 data octets we need:

3 datagrams × 1480 octets = 4440 octets, plus
1 datagram that carries 20 data octets (plus 20 IP header octets)
The relevant fields in each IP fragment:

Total Length = 1500 More Flag = 1 Offset = 0	Total Length = 1500 More Flag = 1 Offset = 1480	Total Length = 1500 More Flag = 1 Offset = 2960	Total Length = 1500 More Flag = 0 Offset = 4440

18.6 For computing the IP checksum, the header is treated as a sequence of 16-bit words. and the 16-bit checksum field is set to all zeros. The checksum is then formed by performing the one's complement addition of all words in the header, and then taking the ones complement of the result. We can express this as follows:

A = W(1) +' W(2) +' . . . +' W(M)
C = ~A

where

+'	=	one's complement addition
C	=	16-bit checksum
A	=	ones complement summation made with the checksum value set to 0
~A	=	ones complement of A
M	=	length of block in 16-bit words
W(i)	=	value of ith word

Suppose that the value in word k is changed by Z = new_value – old_value. Let A" be the value of A after the change and C" be the value of C after the change. Then

A" = A +' Z
C" = ~(A +' Z) = C +' ~Z

18.7 In general, those parameters that influence the progress of the fragments through the internet must be included with each fragment. RFC 791 lists the following options that must be included in each fragment: Security (each fragment must be treated with the same security policy); Loose Source Routing (each fragment must follow the same loose route); Strict Source Routing (each fragment must follow the same strict route).

RFC 791 lists the following options that should be included only in the first fragment: Record Route (unnecessary, and too much overhead to do in all fragments); Internet Timestamp (unnecessary, and too much overhead to do in all fragments).

18.8 Data plus transport header plus internet header equals 1820 bits. This data is delivered in a sequence of packets, each of which contains 24 bits of network header and up to 776 bits of higher-layer headers and/or data. Three network packets are needed. Total bits delivered = 1820 + 3 × 24 = 1892 bits.

18.9 The router-host IP over a local net could be expanded to include some specific form of flow control. The simplest form would be that a local network host would wait for a positive acknowledgment to each packet before transmitting the next. Alternatively, a sliding-window protocol could be used.

The router could send control packets to each local network host advising it of the degree of congestion in the next network.

18.10 No. That would violate the "separation of layers" concept. There is no need for IP to know how data is routed within a network. For the next hop, IP specifies another system on the same network and hands that address down to the network-layer protocol which then initiates the network-layer routing function.

18.11
- Version: Same field appears in IPv6 header; value changes from 4 to 6.
- IHL: Eliminated; not needed since IPv6 header is fixed length.
- Type of Service: This field is eliminated. Three bits of this field define an 8-level precedence value; this is replaced with the priority field, which defines an 8-level precedence value for non-congestion-control traffic. The remaining bits of the type-of-service field deal with reliability, delay, and throughput; equivalent functionality may be supplied with the flow label filed.
- Total Length: Replaced by Payload Length field.
- Identification: Same field, with more bits, appears in Fragment header.
- Flags: The More bit appears in the Fragment header. In IPv6, only source fragmenting is allowed; therefore, the Don't Fragment bit is eliminated.
- Fragment Offset: Same field appears in Fragment header
- Time to Live: Replaced by Hop Limit field in IPv6 header, with in practice the same interpretation.
- Protocol: Replaced by Next Header field in IPv6 header, which either specifies the next IPv6 extension header, or specifies the protocol that uses IPv6.
- Header Checksum: This field is eliminated. It was felt that the value of this checksum was outweighed by the performance penalty.
- Source Address: The 32-bit IPv4 source address field is replaced by the 128-bit source address in the IPv6 header.

- Destination Address: The 32-bit IPv4 destination address field is replaced by the 128-bit source address in the IPv6 header.

18.12 The recommended order, with comments:

1. IPv6 header: This is the only mandatory header, and will indicate if one or more additional headers follow.
2. Hop-by-Hop Options header: After the IPv6 header is processed, a router will need to examine this header immediately afterwards to determine if there are any options to be exercised on this hop.
3. Destination Options header: For options to be processed by the first destination that appears in the IPv6 Destination Address field plus subsequent destinations listed in the Routing header. This header must precede the routing header, because it contains instructions to the node that receives this datagram and that will subsequently forward this datagram using the Routing header information.
4. Routing header: This header must precede the Fragment header, which is processed at the final destination, whereas the routing header must be processed by an intermediate node.
5. Fragment header: This header must precede the Authentication header, because authentication is performed based on a calculation on the original datagram; therefore the datagram must be reassembled prior to authentication.
6. Authentication header: The relative order of this header and the Encapsulating Security Payload header depends on the security configuration. Generally, authentication is performed before encryption so that the authentication information can be conveniently stored with the message at the destination for later reference. It is more convenient to do this if the authentication information applies to the unencrypted message; otherwise the message would have to be re-encrypted to verify the authentication information.
7. Encapsulating Security Payload header: see preceding comment.
8. Destination Options header: For options to be processed only by the final destination of the packet. These options are only to be processed after the datagram has been reassembled (if necessary) and authenticated (if necessary). If the Encapsulating Security Payload header is used, then this header is encrypted and can only be processed after processing that header. There is no advantage to placing this header before the Encapsulating Security Payload header; one might as well encrypt as much of the datagram as possible for security.

18.13 These answers are taken from RFC 1809, *Using the Flow Label Field in IPv6* (June 1995).

a. The IPv6 specification allows routers to ignore Flow Labels and also allows for the possibility that IPv6 datagrams may carry flow setup information in their options. Unknown Flow Labels may also occur if a router crashes and loses its state. During a recovery period, the router will receive datagrams with Flow Labels it does not know, but this is arguably not an error, but rather a part of the recovery period. Finally, if the controversial suggestion that each TCP connection be assigned a separate Flow Label is adopted, it may be necessary to manage Flow Labels using an LRU cache (to avoid Flow Label cache overflow in routers), in which case an active but infrequently used flow's state may have been intentionally discarded. In any case, it is clear that treating this situation as an error and, say dropping the datagram and sending an ICMP message, is inappropriate. Indeed, it seems likely that in most cases, simply forwarding the datagram as one would a datagram with a zero Flow Label would give better service to the flow than dropping the datagram.

b. An example is a router which has two paths to the datagram's destination, one via a high-bandwidth satellite link and the other via a low-bandwidth terrestrial link. A high bandwidth flow obviously should be routed via the high-bandwidth link, but if the router loses the flow state, the router may route the traffic via the low-bandwidth link, with the potential for the flow's traffic to swamp the low-bandwidth link. It seems likely, however, these situations will be exceptions rather than the rule.

18.14 These answers are taken from RFC 1809.

a. An internet may have partitioned since the flow was created. Or the deletion message may be lost before reaching all routers. Furthermore, the source may crash before it can send out a Flow Label deletion message.

b. The obvious mechanism is to use a timer. Routers should discard Flow Labels whose state has not been refreshed within some period of time. At the same time, a source that crashes must observe a quiet time, during which it creates no flows, until it knows that all Flow Labels from its previous life must have expired. (Sources can avoid quiet time restrictions by keeping information about active Flow Labels in stable storage that survives crashes). This is precisely how TCP initial sequence numbers are managed and it seems the same mechanism should work well for Flow Labels.

18.15 Again, RFC 1809:

The argument in favor of using Flow Labels on individual TCP connections is that even if the source does not request special service, a network provider's routers may be able to recognize a large amount of traffic and use the Flow Label field to establish a special route that gives the TCP connection better service (e.g., lower delay or bigger bandwidth). Another argument is to assist in efficient demux at the receiver (i.e., IP and TCP demuxing could be done once).

An argument against using Flow Labels in individual TCP connections is that it changes how we handle route caches in routers. Currently one can cache a route for a destination host, regardless of how many different sources are sending to that destination host. Thus, if five sources each have two TCP connections sending data to a server, one cache entry containing the route to the server handles all ten TCPs' traffic. Putting Flow Labels in each datagram changes the cache into a Flow Label cache, in which there is a cache entry for every TCP connection. So there's a potential for cache explosion. There are ways to alleviate this problem, such as managing the Flow Label cache as an LRU cache, in which infrequently used Flow Labels get discarded (and then recovered later). It is not clear, however, whether this will cause cache thrashing.

Observe that there is no easy compromise between these positions. One cannot, for instance, let the application decide whether to use a Flow Label. Those who want different Flow Labels for every TCP connection assume that they may optimize a route without the application's knowledge. Forcing all applications to use Flow Labels will force routing vendors to deal with the cache explosion issue, even if we later discover that we don't want to optimize individual TCP connections.

18.16 From RFC 1809:

During its discussions, the End-to-End group realized this meant that if a router forwarded a datagram with an unknown Flow Label, it had to ignore the Priority field, because the priority values might have been redefined. (For instance, the priorities might have been inverted). The IPv6 community concluded this behavior was undesirable. Indeed, it seems likely that when the Flow Label is unknown, the router will be able to give much better service if it uses the Priority field to make a more informed routing decision.

18.17 From RFC 1883:

A Routing header is not examined or processed until it reaches the node identified in the Destination Address field of the IPv6 header. In that node, dispatching on the Next Header field of the immediately preceding header causes the Routing module to be invoked, which, in the case of Routing Type 0, performs the following algorithm:

- If Next Addr < Num Addrs, swap the IPv6 Destination Address and Address[Next Addr], then increment Next Addr by one. If bit n of the Strict/Loose Bit Mask has value 0, where n equals the new value of Next Addr, or if the new Destination Address identifies a neighbor of the processing node, re-submit the packet to the IPv6 module for forwarding to the new destination. Otherwise, send an ICMP Destination Unreachable-Not a Neighbor message to the Source Address and discard the packet.
- If Next Addr = Num Addrs, dispatch to the next header processing module, as identified by the Next Header field in the Routing header.
- If Next Addr > Num Addrs, send an ICMP Parameter Problem, Code 0, message to the Source Address, pointing to the Num Addrs field, and discard the packet.

CHAPTER 19
INTERNETWORK OPERATION

ANSWERS TO QUESTIONS

19.1 Multimedia: A number of users "tune in" to a video or audio transmission from a multimedia source station. Teleconferencing: A group of workstations form a multicast group such that a transmission from any member is received by all other group members. Database: All copies of a replicated file or database are updated at the same time. Distributed computation: Intermediate results are sent to all participants. Real-time workgroup: Files, graphics, and messages are exchanged among active group members in real time.

19.2 Unicast: An identifier for a single interface. A packet sent to a unicast address is delivered to the interface identified by that address. Multicast: An identifier for a set of interfaces (typically belonging to different nodes). A packet sent to a multicast address is delivered to all interfaces identified by that address. Broadcast: An identifier for a all interfaces on a network or multiple networks. A packet sent to a broadcast address is delivered to all interfaces identified by that address.

19.3 (1) A convention is needed for identifying a multicast address. (2) Each node (router or source participating in the routing algorithm) must translate between an IP multicast address and a list of networks that contain members of this group. This information allows the node to construct a shortest-path spanning tree to all of the networks containing group members. (3) A router must translate between an IP multicast address and a network multicast address in order to deliver a multicast IP datagram on the destination network. (4) Although some multicast addresses may be assigned permanently, the more usual case is that multicast addresses are generated dynamically and that individual hosts may join and leave multicast groups dynamically. Thus, a mechanism is needed by which an individual host informs routers attached to the same network as itself of its inclusion in and exclusion from a multicast group. (5) Routers must exchange two sorts of information. First, routers need to know which networks include members of a given multicast group. Second, routers need sufficient information to calculate the shortest path to each network containing group members. These requirements imply the need for a multicast routing protocol. (6) A routing algorithm is needed to calculate shortest paths to all group members. (7) Each router must determine multicast routing paths on the basis of both source and destination addresses.

19.4 IGMP supports two principle operations: (1) Hosts send messages to routers to subscribe to and unsubscribe from a multicast group defined by a given multicast address. (2) Routers periodically check which multicast groups are of interest to which hosts.

19.5 An autonomous system (AS) exhibits the following characteristics: (1) An AS is a set of routers and networks managed by a single organization. (2) An AS consists of a group of routers exchanging information via a common routing protocol. (3) Except in times of failure, an AS is connected (in a graph-theoretic sense); that is, there is a path between any pair of nodes.

19.6 An interior router protocol passes routing information between routers within an AS. An exterior router protocol passes routing information between routers in different ASs.

19.7 Distance-vector routing: requires that each node (router or host that implements the routing protocol) exchange information with its neighboring nodes. For this purpose, each node maintains a vector of link costs for each directly attached network and distance and next-hop vectors for each destination. Link-state routing: When a router is initialized, it determines the link cost on each of its network interfaces. The router then advertises this set of link costs to all other routers in the internet topology, not just neighboring routers. From then on, the router monitors its link costs. Whenever there is a significant change (a link cost increases or decreases substantially, a new link is created, an existing link becomes unavailable), the router again advertises its set of link costs to all other routers in the configuration. Path-vector routing: dispenses with routing metrics and simply provide information about which networks can be reached by a given router and the ASs that must be crossed to get there. The approach differs from a distance-vector algorithm in two respects: First, the path-vector approach does not include a distance or cost estimate. Second, each block of routing information lists all of the ASs visited in order to reach the destination network by this route.

19.8 Neighbor acquisition: neighbor acquisition occurs when two neighboring routers in different autonomous systems agree to exchange routing information regularly. One router sends a request message to the other, which may either accept or refuse the offer. Neighbor reachability: Each partner needs to be assured that the other partner still exists and is still engaged in the neighbor relationship. For this purpose, the two routers periodically issue Keepalive messages to each other. Network reachability: Each router maintains a database of the networks that it can reach and the preferred route for reaching each network. Whenever a change is made to this database, the router issues an Update message that is broadcast to all other routers implementing BGP. Because the Update message is broadcast, all BGP routers can build up and maintain their routing information.

19.9 ISA is an architecture intended to provide QoS services over IP-based internets.

19.10 Elastic traffic can adjust, over wide ranges, to changes in delay and throughput across an internet and still meet the needs of its applications. Inelastic traffic does not easily adapt, if at all, to changes in delay and throughput across an internet. The prime example is real-time traffic, such as voice and video.

19.11 Admission control: For QoS transport (other than default best-effort transport), ISA requires that a reservation be made for a new flow. If the routers collectively determine that there are insufficient resources to guarantee the requested QoS, then the flow is not admitted. The protocol RSVP is used to make reservations. Routing algorithm: The routing decision may be based on a variety of QoS parameters, not just minimum delay. Queuing discipline: A vital element of the ISA is an effective queuing policy that takes into account the differing requirements of different flows. Discard policy: A queuing policy determines which packet to transmit next if a number of packets are queued for the same output port. A separate issue is the choice and timing of packet discards. A discard policy can be an important element in managing congestion and meeting QoS guarantees.

19.12 Guaranteed: With this service, an application provides a characterization of its expected traffic profile, and the service determines the end-to-end delay that it can guarantee. Controlled load: The service tightly approximates the behavior visible to applications receiving best-effort service under unloaded conditions. There is no specified upper bound on the queuing delay through the network. However, the service ensures that a very high percentage of the packets do not experience delays that greatly exceed the minimum transit delay (i.e., the delay due to propagation time plus router processing time with no queuing delays). Best effort: The normal processing provided over an IP-based internet.

19.13 First-in-first-out (FIFO) queuing: A single queue is maintained at each output port. When a new packet arrives and is routed to an output port, it is placed at the end of the queue. As long as the queue is not empty, the router transmits packets from the queue, taking the oldest remaining packet next. Weighted fair queuing (WFQ): takes into account the amount of traffic through each queue and gives busier queues more capacity without completely shutting out less busy queues. In addition WFQ can take into account the amount of service requested by each traffic flow and adjust the queuing discipline accordingly.

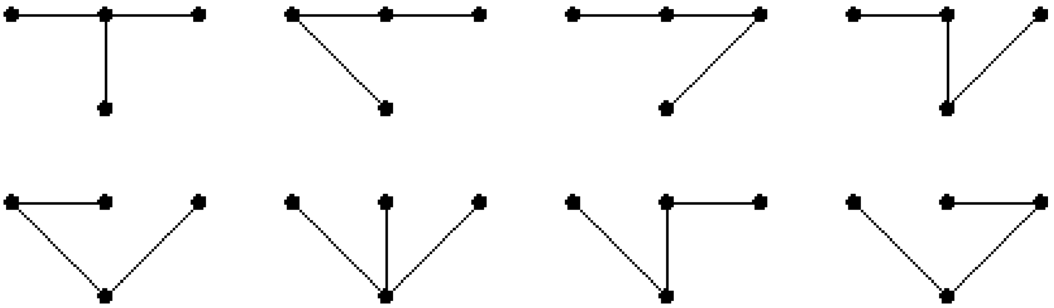
19.14 The DS codepoint is the DS label used to classify packets for differentiated services.

19.15 Classifier: Separates submitted packets into different classes. Meter: Measures submitted traffic for conformance to a profile. The meter determines whether a given packet stream class is within or exceeds the service level guaranteed for that class. Marker: Re-marks packets with a different codepoint as needed. Shaper: Delays packets as necessary so that the packet stream in a given class does not exceed the traffic rate specified in the profile for that class. Dropper: Drops packets when the rate of packets of a given class exceeds that specified in the profile for that class.

19.16 The DS specifications refer to the forwarding treatment provided at a router as per-hop behavior (PHB).

ANSWERS TO PROBLEMS

19.1

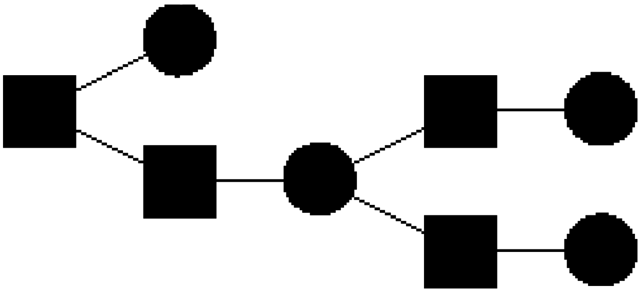


19.2 For convenience, flip the table on its side:

N1	N2	N3	N4	N5	N6	L1	L2	L3	L4	L5	Total
1	1	1	2	1	1	1	2	2	1	2	15

This is the least efficient method, but it is very robust: a packet will get through if there is at least one path from source to destination. Also, no prior routing information needs to be exchanged.

19.3 Root to the left:



19.4 So that the queries are not propagated outside of the local network.

19.5 Load balancing increases the chance of packets being delivered out of order, and possibly distorts the round-trip times calculated by the transport layer. Source: [STEV94]

19.6 a. $\lambda = \lambda 1 + \lambda 2 = 0.5$; Using the M / M / 1 equations in Table 8.6:
$T_r = T_s / (1 - \lambda T_s) = 1 / (1 - 0.5) = 2$; Then
$V = (4 - 2 \times 2) + (4 - 2) = 2$

b. Using the M / M / 1 equations from Table 8.9:
$\rho_1 = \lambda_1 T_{s1} = 0.25 = \rho_2$; $\rho = \rho_1 + \rho_2 = 0.5$

$T_{r1} = T_{s1} + (\rho_1 T_{s1} + \rho_2 T_{s2}) / (1 - \rho_1) = 1.67$
$T_{r2} = T_{s2} + (T_{q1} - T_{s1}) / (1 - \rho) = 2.33$
$V = (4 - 2\times1.67) + (4 - 2.33) = 2.33$

Therefore, the strict priority service is more efficient in the sense that it delivers a higher utility for the same throughput.

19.7 a. During a bust of S seconds, a total of MS octets are transmitted. A burst empties the bucket (b octets) and, during the burst, tokens for an additional rS octets are generated, for a total burst size of (b + rS). Thus,
$b + rS = MS$
$S = b / (M - r)$

b. $S = (250 \times 10^3) / (23 \times 10^6) \approx 11$ msec

19.8 a. Problem: IPv6 inserts a variable number of variable-length Internet-layer headers before the transport header, increasing the difficulty and cost of packet classification for QoS. Solution: Efficient classification of IPv6 data packets could be obtained using the Flow Label field of the iPv6 header.

b. Problem: IP-level security, under either IPv4 or IPv6, may encrypt the entire transport header, hiding the port numbers of data packets from intermediate routers. Solution: There must be some means of identifying source and destination IP users (equivalent to the TCP or UDP port numbers). With IP-level security, there is a parameter, called the Security Parameter Index (SPI) carried in the security header. While SPIs are allocated based on destination address, they will typically be associated with a particular sender. As a result, two senders to the same unicast destination will usually have different SPIs. In order to support the control of multiple independent flows between source and destination IP addresses, the SPI will be included as part of the FILTER_SPEC.

CHAPTER 20
TRANSPORT PROTOCOLS

ANSWERS TO QUESTIONS

20.1 User identification. Transport entity identification. Host address. Network number.

20.2 (1) The TS user knows the address it wishes to use ahead of time. This is basically a system configuration function. For example, a process may be running that is only of concern to a limited number of TS users, such as a process that collects statistics on performance. From time to time, a central network management routine connects to the process to obtain the statistics. These processes generally are not, and should not be, well known and accessible to all. (2) Some commonly used services are assigned "well-known addresses. (3) A name server is provided. The TS user requests a service by some generic or global name. The request is sent to the name server, which does a directory lookup and returns an address. The transport entity then proceeds with the connection. This service is useful for commonly used applications that change location from time to time. (4) In some cases, the target user is to be a process that is spawned at request time. The initiating user can send a process request to a well-known address. The user at that address is a privileged system process that will spawn the new process and return an address.

20.3 With respect to the interface between the transport protocol and higher-level protocols, the transport protocol performs a multiplexing/demultiplexing function. That is, multiple users employ the same transport protocol and are distinguished by port numbers or service access points. The transport entity may also perform a downward multiplexing function with respect to the network services that it uses.

20.4 The credit scheme decouples acknowledgment from flow control. In a credit scheme, a segment may be acknowledged without granting new credit, and vice versa. For the credit scheme, each individual octet of data that is transmitted is considered to have a unique sequence number. In addition to data, each transmitted segment includes in its header three fields related to flow control: sequence number (*SN*), acknowledgment number (*AN*), and window (*W*).

20.5 In a sliding-window scheme, acknowledgment and flow control are bound together. An acknowledgment results in a fixed additional credit being granted.

20.6 Two-way: A connection establishment calls for the exchange of SYNs, a procedure sometimes referred to as a two-way handshake. Suppose that A issues a SYN to B.

It expects to get a SYN back, confirming the connection. Three-way: As part of connection establishment, each side acknowledges explicitly the other's SYN and sequence number.

20.7 It solves the duplicate SYN problem, in which an obsolete SYN arrives after the close of a connection.

20.8 Data stream push: Ordinarily, TCP decides when sufficient data have accumulated to form a segment for transmission. The TCP user can require TCP to transmit all outstanding data up to and including that labeled with a push flag. On the receiving end, TCP will deliver these data to the user in the same manner. A user might request this if it has come to a logical break in the data. Urgent data signaling: This provides a means of informing the destination TCP user that significant or "urgent" data is in the upcoming data stream. It is up to the destination user to determine appropriate action.

20.9 The TCP standard provides a precise specification of the protocol to be used between TCP entities. However, certain aspects of the protocol admit several possible implementation options. These options are defined in the TCP standard. Although two implementations that choose alternative options will be interoperable, there may be performance implications.

20.10 The TCP flow control mechanism can be used to recognize the onset of congestion (by recognizing increased delay times and dropped segments) and to react by reducing the flow of data. If many of the TCP entities operating across a network exercise this sort of restraint, internet congestion is relieved.

20.11 UDP provides the source and destination port addresses and a checksum that covers the data field. These functions would not normally be performed by protocols above the transport layer. Thus UDP provides a useful, though limited, service.

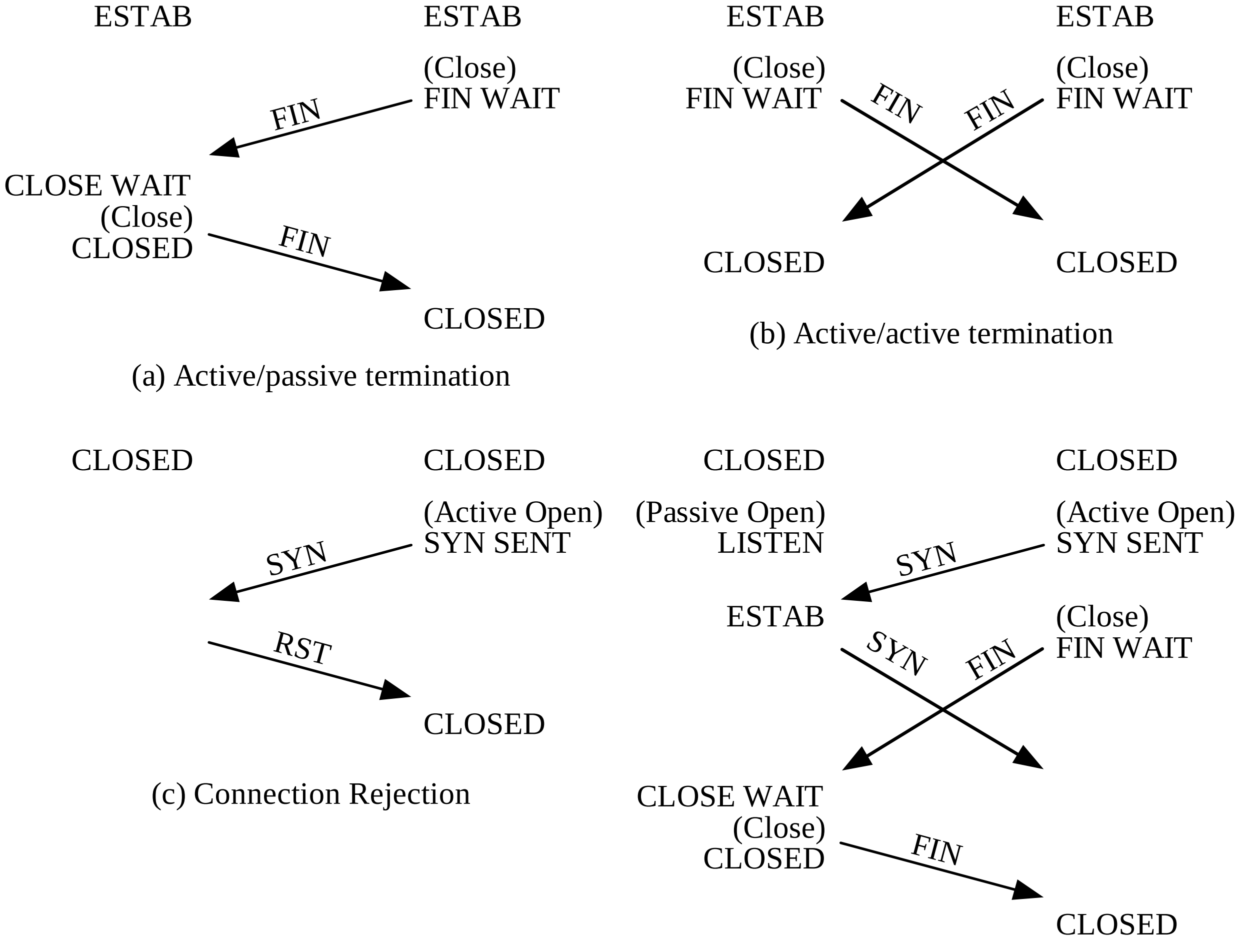
Answers to Problems

20.1 A single control channel implies a single control entity that can manage all the resources associated with connections to a particular remote station. This may allow more powerful resource control mechanisms. On the other hand, this strategy requires a substantial number of permanent connections, which may lead to buffers or state table overhead.

20.2 It is relatively sluggish and may unnecessarily stress the network layer.

20.3 The number of unacknowledged segments in the "pipeline" at any time is 5. Thus, once steady state is reached, the maximum achievable throughput is equal to the normalized theoretical maximum of 1.

20.4



(a) Active/passive termination

(b) Active/active termination

(c) Connection Rejection

(d) Connection abortion

20.5 No. They do make it easier to implement flow control in a manner that is extensible to unreliable and / or nonsequencing networks.

20.6 When a reset occurs, the transport entity may have a number of outstanding segments that have not been acknowledged. The entity does not know if they were received by the other side before the network connection went down. This uncertainty must be resolved during the resynchronization procedure.

20.7 There is no good solution if the delay experienced has high variance. One approach is to use an exponential decay smoothing algorithm discussed in this chapter.

20.8 This will depend on whether multiplexing or splitting occurs. If there is a one-to-one relationship between network connections and transport connections, then it will do no good to grant credit at the transport level in excess of the window size at the network level. If one transport connection is split among multiple network connections (each one dedicated to that single transport connection), then a practical upper bound on the transport credit is the sum of the network window sizes. If multiple transport connections are multiplexed on a single network connection, their aggregate credit should not exceed the network window size. Furthermore, the relative amount of credit will result in a form of priority mechanism.

20.9 A sender may not send more than 256 packets; that is, 256 × 128 × 8 = 262,144 bits in 30 sec. The data rate is thus no more than 8738 bps.

20.10 Deadlocks are possible. For example, an old RFC arrives at A and A acknowledges it. The acknowledgement is lost, but A is now open. Now the same thing happens to B, and both are open, but expecting different sequence numbers. Source: [TANE03]

20.11 a. A letter addressed to a friend.
b. An advertising circular sent to all of the boxes at a post office. The address of the post office is well-known; each box number is akin to an SAP address.
c. A letter addressed to a company officer by title (e.g., personnel manager, accounts payable manager) is received by the company mail room. Someone in the mail room determines the location of the individual and routes the letter appropriately.
d. A company hires a mass mailing firm to send a letter out to the latest version of a mailing list data base. The mailing firm addresses the letters at the last minute from the address list in the database.

20.12 In TCP, no provision is made. A later segment can provide a new credit allocation. Provision is made for misordered and lost credit allocations in the ISO transport protocol (TP) standard. In ISO TP, ACK/Credit messages (AK) are in separate PDUs, not part of a data PDU. Each AK TPDU contains a YR-TU-NR field, which is the sequence number of the next expected data TPDU, a CDT field, which grants credit, and a "subsequence number", which is used to assure that the credit grants are processed in the correct sequence. Further, each AK contains a "flow control confirmation" value which echoes the parameter values in the last AK received (YR-TU-NR, CDT, subsequence number). This can be used to deal with lost AKs.

20.13 The transport entity could interrupt the user to notify it of a pending request. The user could then move into the LISTEN state. An alternative would be to implement an Accept command, which would allow the user to move to ESTAB directly. The transport entity could also queue the request or discard it.

20.14 The connection is held in limbo to allow all connection messages (e.g., late duplicates) that may still exits out in the network to arrive or be discarded. If any messages arrive, TCP will know that they belong to a defunct connection and will discard them.

20.15 The upper limit ensures that the maximum difference between sender and receiver can be no greater than 2^{31}. Without such a limit, TCP might not be able to tell when the 32-bit sequence number had rolled over from $2^{31} - 1$ back to 0.

20.16 a. SRTT(n) $= \alpha \times \text{SRTT}(0) + (1 - \alpha) \times \text{RTT} \times (\alpha^{n-1} + \alpha^{n-2} + \ldots \alpha + 1)$

$= \alpha \times \text{SRTT}(0) + (1 - \alpha) \times \text{RTT} \times (1 - \alpha^{n}) / (1- \alpha)$

SRTT(19) = 1.1 sec

b. SRTT(19) = 2.9 sec; in both cases, the convergence speed is slow, because in both cases, the initial SRTT(0) is improperly chosen.

20.17 When the 50-octet segment arrives at the recipient, it returns a credit of 1000 octets. However, the sender will now compute that there are 950 octets in transit in the network, so that the usable window is now only 50 octets. Thus, the sender will once again send a 50-octet segment, even though there is no longer a natural boundary to force it.

In general, whenever the acknowledgment of a small segment comes back, the usable window associated with that acknowledgment will cause another segment of the same small size to be sent, until some abnormality breaks the pattern. Once the condition occurs, there is no natural way for those credit allocations to be recombined; thus the breaking up of the usable window into small pieces will persist.

20.18 a. As segments arrive at the receiver, the amount of available buffer space contracts. As data from the buffer is consumed (passed on to an application), the amount of available buffer space expands. If SWS is not taken into account, the following procedure is followed: When a segment is received, the recipient should respond with an acknowledgment that provides credit equal to the available buffer space. The SWS avoidance algorithm introduces the following rule: When a segment is received, the recipient should not provide additional credit unless the following condition is met:

$$\text{available buffer space} \geq \text{MIN}\left(\frac{\text{buffer size}}{2}, \text{maximum segment size}\right)$$

The second term is easily explained: if the available buffer space is greater than the largest possible segment, then clearly SWS cannot occur. The first term is a reasonable guideline that states that if at least half of the buffer is free, the sender should be provided the available of credit.

b. The suggested strategy is referred to as the Nagle algorithm and can be stated as follows: If there is unacknowledged data, then the sender buffers all data until the outstanding data have been acknowledged or until a maximum-sized segment can be sent. Thus, the sender accumulates data locally to avoid SWS.

20.19 SRTT(K + 1) = (1 - g)SRTT(K) + gRTT(K + 1)

SERR(K + 1) = RTT(K + 1) - SRTT(K)

Substituting for SRTT(K) in the first equation from the second equation:

SRTT(K + 1) = RTT(K + 1) - (1 - g)SERR(K + 1)

Think of RTT(K + 1) as a prediction of the next measurement and SERR(K + 1) as the error in the last prediction. The above expression says we make a new prediction based on the old prediction plus some fraction of the prediction error.

20.20 TCP initializes the congestion window to 1, sends an initial segment, and waits. When the ACK arrives, it increases the congestion window to 2, sends 2 segments, and waits. When the 2 ACKs arrives, they each increase the congestion window by one, so that it can send 4 segments. In general, it takes $\log_2 N$ round trips before TCP can send N segments.

20.21 a. $W = (10^9 \times 0.06) / (576 \times 8) \approx 13{,}000$ segments
If the window size grows linearly from 1, it will take about 13,000 round trips, or about 13 minutes to get the correct window size.

b. $W = (10^9 \times 0.06) / (16{,}000 \times 8) \approx 460$ segments
In this case, it takes about 460 round trips, which is less than 30 seconds.

CHAPTER 21
NETWORK SECURITY

ANSWERS TO QUESTIONS

21.1 Passive attacks have to do with eavesdropping on, or monitoring, transmissions. Electronic mail, file transfers, and client/server exchanges are examples of transmissions that can be monitored. Active attacks include the modification of transmitted data and attempts to gain unauthorized access to computer systems.

21.2 Passive attacks: release of message contents and traffic analysis. Active attacks: masquerade, replay, modification of messages, and denial of service.

21.3 Data Encryption Standard (DES) is a conventional encryption scheme standardized by the National Bureau of Standards (now the National Institute of Standards and Technology) as a federal standard in 1977. It is the most widely used encryption scheme, but as the years have passed the capabilities of computers have increased to the point that the 56 bit key is becoming insufficient. To increase the effective key length, triple DES is used, which involves using DES three times in sequence with either two or three distinct keys.

21.4 AES is more efficient in software than 3DES and does provide the option for a longer key length. Also AES uses a 128-bit block size, which is more efficient and may be more secure than the 64-bit block size used by 3DES.

21.5 Traffic padding produces ciphertext output continuously, even in the absence of plaintext. A continuous random data stream is generated. When plaintext is available, it is encrypted and transmitted. When input plaintext is not present, random data are encrypted and transmitted. This makes it impossible for an attacker to distinguish between true data flow and padding and therefore impossible to deduce the amount of traffic.

21.6 Message encryption: Encrypt entire message. This provides authentication if only sender and receiver share secret key. Message authentication code: Use of a secret key to generate a small block of data, known as a message authentication code, that is appended to the message. This provides authentication if only sender and receiver share secret key. Hash function: Calculate a hash code over the message plus a secret key, then send the message and the hash code, but not the secret key. The receiver uses the incoming message and the shared secret key to verify the hash code.

21.7 A secure hash function is a one-way function H that satisfies: 1. H can be applied to a block of data of any size. 2. H produces a fixed-length output. 3. H(*x*) is relatively easy to compute for any given *x*, making both hardware and software implementations practical. 4. For any given code *h*, it is computationally infeasible to find *x* such that H(*x*) = *h*. 5. For any given block *x*, it is computationally infeasible to find $y \neq x$ with H(*y*) = H(*x*). 6. It is computationally infeasible to find any pair (*x*, *y*) such that H(*x*) = H(*y*).

21.8 Symmetric encryption is based on a single key that is secretly shared between the parties involved in the information transfer. Public-key encryption systems are based on two keys, one of which is public, and the other kept secret.

21.9 The key used in symmetric encryption is typically referred to as a secret key. The two keys used for public-key encryption are referred to as the public key and the private key. Invariably, the private key is kept secret, but it is referred to as a private key rather than a secret key to avoid confusion with symmetric encryption.

21.10 To sign a document, Bob calculates a hash function of the text to be signed. He then encrypts it with his private key, which presumably only he knows. This is the digital signature.

21.11 A public-key certificate consists of a public key plus a User ID of the key owner, with the whole block signed by a trusted third party. Typically, the third party is a certificate authority (CA) that is trusted by the user community, such as a government agency or a financial institution.

21.12 The SSL Record Protocol provides basic security services to various higher-layer protocols. In particular, the Hypertext Transfer Protocol (HTTP), which provides the transfer service for Web client/server interaction, can operate on top of SSL. Three higher-layer protocols are defined as part of SSL: the Handshake Protocol, The Change Cipher Spec Protocol, and the Alert Protocol.

21.13 Connection: A connection is a transport (in the OSI layering model definition) that provides a suitable type of service. For SSL, such connections are peer-to-peer relationships. The connections are transient. Every connection is associated with one session. Session: An SSL session is an association between a client and a server. Sessions are created by the Handshake Protocol. Sessions define a set of cryptographic security parameters, which can be shared among multiple connections. Sessions are used to avoid the expensive negotiation of new security parameters for each connection.

21.14 Confidentiality: The Handshake Protocol defines a shared secret key that is used for conventional encryption of SSL payloads. Message Integrity: The Handshake Protocol also defines a shared secret key that is used to form a message authentication code (MAC).

21.15 Access control; connectionless integrity; data origin authentication; rejection of replayed packets (a form of partial sequence integrity); confidentiality (encryption); and limited traffic flow confidentiality

Answers to Problems

21.1 In a military environment, an increased level of traffic to and from a particular commander might serve as a predictor of troop movements. If the adversary can tap into the network at the point of connection of the commander's system the network, this traffic can be observed even with both end-to-end and link-by-link encryption.

21.2 The central points should be highly fault-tolerant, should be physically secured, and should use trusted hardware/software.

21.3 Yes. The eavesdropper is left with two strings, one sent in each direction, and their XOR is the secret key.

21.4

a. $M3 =$

5	2	1	4	5
1	4	3	2	2
3	1	2	5	3
4	3	4	1	4
2	5	5	3	1

b. Assume a plaintext message p is to be encrypted by Alice and sent to Bob. Bob makes use of M1 and M3, and Alice makes use of M2. Bob chooses a random number, k, as his private key, and maps k by M1 to get x, which he sends as his public key to Alice. Alice uses x to encrypt p with M2 to get z, the ciphertext, which she sends to Bob. Bob uses k to decrypt z by means of M3, yielding the plaintext message p.

c. If the numbers are large enough, and M1 and M2 are sufficiently random to make it impractical to work backwards, p cannot be found without knowing k.

21.5 a. $n = 33$; $\phi(n) = 20$; $e = 3$; C = 26.

b. $n = 55$; $\phi(n) = 40$; $d = 27$; C = 14.

c. $n = 77$; $\phi(n) = 60$; $d = 53$; C = 57.

d. $n = 143$; $\phi(n) = 120$; $d = 11$; C = 106.

e. $n = 527$; $\phi(n) = 480$; $d = 343$; C = 128. For decryption, we have

$$\begin{aligned} 128^{343} \bmod 527 &= 128^{256} \times 128^{64} \times 128^{16} \times 128^{4} \times 128^{2} \times 128^{1} \bmod 527 \\ &= 35 \times 256 \times 35 \times 101 \times 47 \times 128 = 2 \bmod 527 \\ &= 2 \bmod 257 \end{aligned}$$

21.6 M = 5

21.7 $d = 3031$

21.8 Yes. If a plaintext block has a common factor with n modulo n then the encoded block will also have a common factor with n modulo n. Because we encode blocks which are smaller than pq, the factor must be p or q and the plaintext block must be a multiple of p or q. We can test each block for primality. If prime, it is p or q. In this case we divide into n to find the other factor. If not prime, we factor it and try the factors as divisors of n.

21.9 Refer to Figure 21.10 The private key k is the pair {d, n}; the public key x is the pair {e, n}; the plaintext p is M; and the ciphertext z is C. M1 is formed by calculating $d = e^{-1} \bmod \phi(n)$. M2 consists of raising M to the power e (mod n). M2 consists of raising C to the power d (mod n).

21.10 Yes.

21.11 The opponent has the two-block message B1, B2 and its hash RSAH(B1, B2). The following attack will work. Choose an arbitrary C1 and choose C2 such that:

$$C2 = RSA(C1) \oplus RSA(B1) \oplus B2$$

then

$$\begin{aligned} RSA(C1) \oplus C2 &= RSA(C1) \oplus RSA(C1) \oplus RSA(B1) \oplus B2 \\ &= RSA(B1) \oplus B2 \end{aligned}$$

so

$$\begin{aligned} RSAH(C1, C2) &= RSA[RSA(C1) \oplus C2)] = RSA[RSA(B1) \oplus B2] \\ &= RSAH(B1, B2) \end{aligned}$$

21.12 The change cipher spec protocol exists to signal transitions in ciphering strategies, and can be sent independent of the complete handshake protocol exchange.

21.13 a. Immutable: Version, Internet Header Length, Total Length, Identification, Protocol (This should be the value for AH.), Source Address, Destination Address (without loose or strict source routing). None of these are changed by routers in transit.
Mutable but predictable: Destination Address (with loose or strict source routing). At each intermediate router designated in the source routing list, the Destination Address field is changed to indicate the next designated address. However, the source routing field contains the information needed for doing the MAC calculation.
Mutable (zeroed prior to ICV calculation): Type of Service (TOS), Flags, Fragment Offset, Time to Live (TTL), Header Checksum. TOS may be altered by a router to reflect a reduced service. Flags and Fragment offset are altered if an router performs fragmentation. TTL is decreased at each router. The Header Checksum changes if any of these other fields change.

b. Immutable: Version, Payload Length, Next Header (This should be the value for AH.), Source Address, Destination Address (without Routing Extension Header)

Mutable but predictable: Destination Address (with Routing Extension Header)

Mutable (zeroed prior to ICV calculation): Class, Flow Label, Hop Limit

c. IPv6 options in the Hop-by-Hop and Destination Extension Headers contain a bit that indicates whether the option might change (unpredictably) during transit.

Mutable but predictable: Routing

Not Applicable: Fragmentation occurs after outbound IPSec processing and reassembly occur before inbound IPSec processing , so the Fragmentation Extension Header, if it exists, is not seen by IPSec.

CHAPTER 22
DISTRIBUTED APPLICATIONS

ANSWERS TO QUESTIONS

22.1 RFC 821 defines SMTP which is the protocol for exchanging email messages. RFC 822 describes the format of those messages.

22.2 The Simple Mail Transfer Protocol (SMTP) is the standard protocol in the TCP/IP protocol suite for transferring mail between hosts. Multipurpose Internet Mail Extension (MIME) is an extension of SMTP to address some of its problems and limitations. MIME addresses SMTP's inability to directly transmit executables and other binary files and provides support for national language characters that SMTP does not directly support.

22.3 Content-Type: Describes the data contained in the body with sufficient detail that the receiving user agent can pick an appropriate agent or mechanism to present the data to the user or otherwise deal with the data in an appropriate manner. Transfer-Encoding: Indicates the type of transformation that has been used to represent the body of the message in a way that is acceptable for mail transport.

22.4 R64 converts a raw 8-bit binary stream to a stream of printable ASCII characters. Each group of three octets of binary data is mapped into four ASCII characters.

22.5 A proxy acts on behalf of other clients and presents requests from other clients to a server. The proxy acts as a server in interacting with a client and as a client in interacting with a server. A gateway is a server that appears to the client as if it were an origin server. It acts on behalf of other servers that may not be able to communicate directly with a client. Unlike the proxy and the gateway, the tunnel performs no operations on HTTP requests and responses. Instead, a tunnel is simply a relay point between two TCP connections, and the HTTP messages are passed unchanged as if there were a single HTTP connection between user agent and origin server.

22.6 A network management system is a collection of tools for network monitoring and control that is integrated in the following senses: (1) A single operator interface with a powerful but user-friendly set of commands for performing most or all network management tasks. (2) A minimal amount of additional equipment. That is, most of the hardware and software required for network management is incorporated into the existing user equipment.

22.7 Management station: typically a standalone device, but may be a capability implemented on a shared system. In either case, the management station serves as the interface for the human network manager into the network management system. Agent: Key platforms, such as hosts, bridges, routers, and hubs, may be equipped with agent software so that they may be managed from a management station. The agent responds to requests for information from a management station, responds to requests for actions from the management station, and may asynchronously provide the management station with important but unsolicited information. Management information base (MIB): To manage resources in the network, each resource is represented as an object. An object is, essentially, a data variable that represents one aspect of the managed agent. The collection of objects is referred to as a MIB. Network management protocol: used for the exchange of management commands, responses, and information between management stations and agents.

22.8 The Simple Network Management Protocol (SNMP) is the network management tool for TCP/IP networks. It provides the following three basic capabilities: (i) *Get*: enables the management station to retrieve the value of objects at the agent; (ii) *Set*: enables the management station to set the value of objects at the agent; (iii) *Trap*: enables an agent to notify the management station of significant events.

22.9 SNMPv2 extends the original version SNMPv1 by providing support for decentralized network management and efficient transfer of large blocks of data. SNMPv3 provides a security capability to be used with SNMPv1 or SNMPv2.

ANSWERS TO PROBLEMS

22.1 Mail-bagging economizes on data transmission time and costs. It also reduces the amount of temporary storage that each MTA must have available to buffer messages in its possession. These factors can be very significant in electronic mail systems that process a large number of messages. Routing decisions may keep mail-bagging in mind.

22.2 a. The value of a Gauge has its maximum value whenever the information being modeled is greater than or equal to that maximum value; if the information being modeled subsequently decreases below the maximum value, the Gauge remains at the maximum value. The gauge can only be released from this maximum value by subsequent management action.

b. The SNMPv2 interpretation provides a realistic representation of the underlying value at all times, subject to the limitation of the gauge. However, a manager may want to know that some maximum value has been reached or exceeded. By "sticking" the gauge at its maximum value until it is noticed and released by a manager, this information is preserved.

22.3 Six round trips: The HELO command, MAIL, RCPT, DATA, body of the message, and QUIT. Source: [STEV94]

22.4 Consider the first five network round trips from Problem 22.3. Each is a small command (probably a single segment) that places little load on the network. If all five make it through to the server without retransmission, the congestion window could be six segments when the body is sent. If the body is large , the client could send the first six segment at once, which the network might not be able to handle. Source: [STEV94]

22.5 If the same port were used for both traps and requests, separating the manager from the agent in the same system would be difficult. Source: [STEV94]