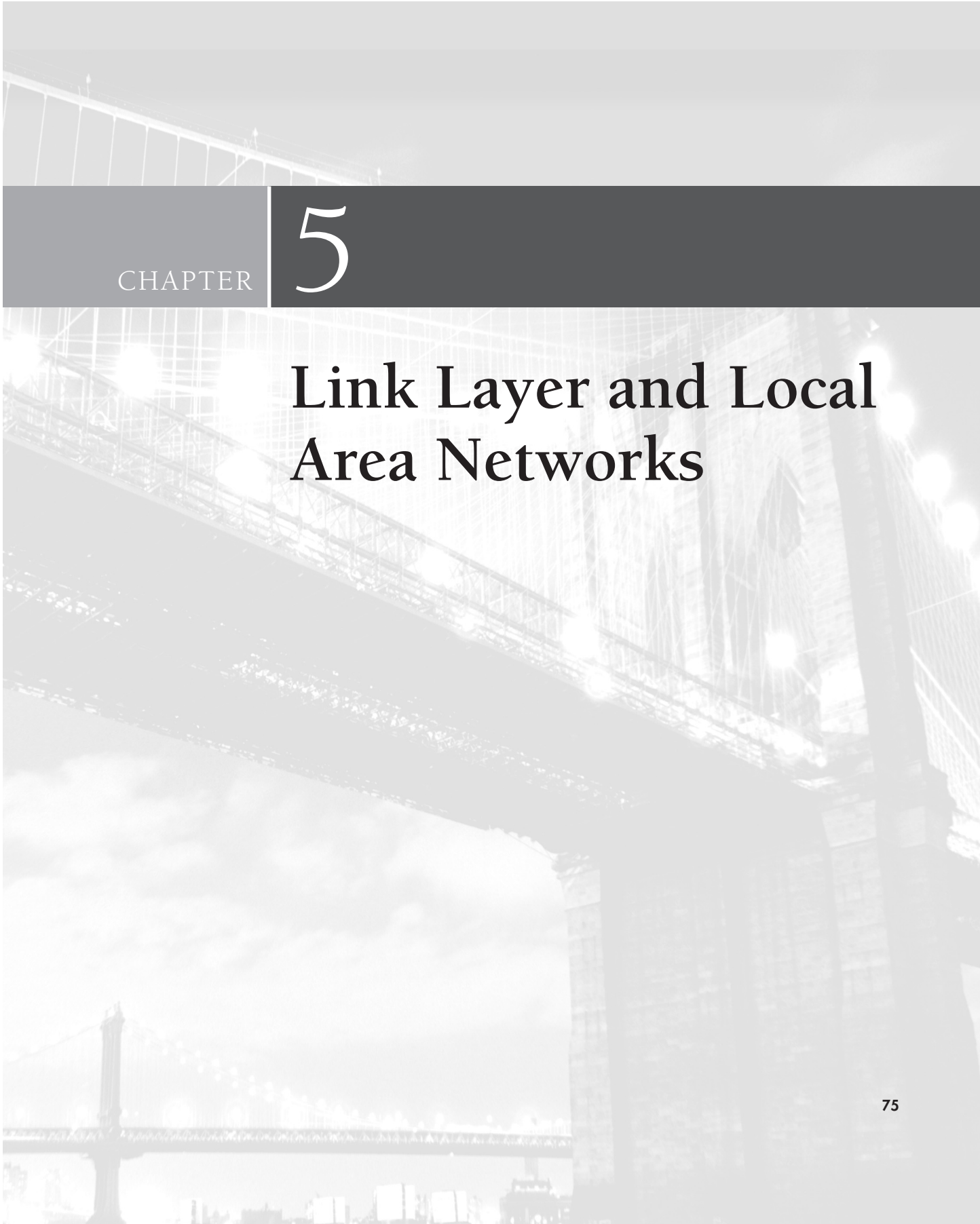


CHAPTER 5

Link Layer and Local Area Networks

Most Important Ideas and Concepts from Chapter 5

- **Link layer services.** One of the key facets of a layered architecture is that a protocol in one layer provides services to the protocols in the layer above. In particular, a protocol in the link layer (layer 2) provides services to the protocols in the network layer (layer 3). The most basic service of any link-layer protocol is to move a datagram from one node to an adjacent node over a single communication link. As discussed on page 420 of the textbook, a useful analogy here is that of a tourist, a travel agent, transportation segments, and transportation modes (bus, plane, train, and so on). A transportation segment (say, between two cities) is analogous to a link, a transportation mode is analogous to a link-layer protocol, a tourist is analogous to a packet, and a travel agent—who plans the trip from end-to-end—is analogous to a routing protocol.

  At the sender side, a link-layer protocol encapsulates the datagram in a link-layer frame, which is then passed to the physical layer for transmission over the link. At the receiving side, after receiving a frame, the link-layer protocol extracts the datagram and passes it to the network layer. Other possible services a link-layer protocol can provide include: medium access control, reliable delivery, flow control, error detection, and correction. Many of these services can also be provided in other layers. For example, we learned in Chapter 3 that the TCP transport-layer protocol provides end-to-end flow control, so that the sending side of a TCP connection does not overwhelm the receiving side of the connection. Flow-control in the link layer has a similar objective, but is no longer from end-to-end but instead from node to adjacent node. In this chapter we cover in detail many of the classic link-layer services. We also highlight many of the key services in the paragraphs below.
- **Error detection and correction.** Error detection is another example of a service that can be provided in different layers. The checksum in UDP and TCP are examples of an error detection service at the transport layer. Similarly, IP's header checksum is an example of error detection at the network layer. Typically, the error detection in the network and transport layers is rather crude, only detecting single-bit errors and certain combinations of multiple-bit errors. Typically more sophisticated, a link-layer error detection scheme can detect (over a single link) single-bit and a wide-range of common multiple-bit errors. The main idea behind error detection is for the sender to create, as a function of the bits in the frame, a block of bits and include the block in a header field in the frame. When the receiving side of the link receives the frame, it runs the frame through the same function and compares the result with the block in the frame. If there is a match, the frame is considered error-free; if there is an inconsistency, the frame is considered corrupted. Section 5.2 discusses three error detection schemes, each scheme being more complicated and more powerful. The third scheme, cyclic redundancy check (CRC), is used in many link-layer protocols, including Ethernet and Wi-Fi. Schemes can also be designed so that not only they determine whether there is an error, but also they determine exactly which bits in the

frame are in error. In this case, the receiver can simply flip the erroneous bits, thereby correcting the errors. When a scheme detects and corrects erroneous bits, it is said to provide an error correction service.

- **Multiple access protocols.** There are two types of network links: point-to-point links and broadcast links. A point-to-point link consists of a single sender at one end of the link and single receiver at the other end of the link; an example is a fiber-optic link between two routers. Broadcast links have multiple sending and receiving nodes, all connected to the same shared broadcast channel. For a broadcast link, when any one node transmits a frame, each of the other nodes receives a copy of the frame. Ethernet and wireless LANs (for example, Wi-Fi) are examples of broadcast links. Without any coordination among the nodes, there is the possibility that two or more nodes transmit simultaneously, causing their frames to collide at the receiving nodes. Typically, when a collision occurs, it is difficult, if not impossible, for a receiver to disentangle the colliding frames. The purpose of a multiple access protocol is to coordinate the transmissions of the senders to reduce the probability of-or even entirely eliminate-the collisions at the receivers. Three of the most desirable characteristics of a multiple access protocol for a channel of rate R follow:

  (a) When M nodes have data to send, each of the M nodes gets an average throughput of R/M. Thus, the protocol is fair and makes use of all the transmission capacity, R.

  (b) The protocol is decentralized.

  (c) The protocol is simple.

  Multiple access protocols can be classified into three categories: channel partitioning protocols, taking-turns protocols, and random access protocols. Time division multiplexing (TDM) is an excellent example of a channel partitioning protocol. To describe TDM, suppose the channel supports N nodes. TDM divides time into time frames and further divides each frame into N slots. Each slot in is then assigned to one of the N nodes. Specifically, whenever a node has a packet to send, it transmits the packet's bits during its assigned slot in the revolving TDM frame.

  TDM eliminates collisions and is perfectly fair: each node gets R/N bps on average when it has something to send. But TDM has two major drawbacks. First, if only a few nodes are transmitting, then most of the channel transmission capacity is wasted. Second, when a node has something to transmit, it has to wait for its slot in the revolving frame to circle around. Token passing is the classic example of a taking turns protocol.

  In token passing, a small frame, known as the token, is passed among the nodes in some fixed order, for example, from node 1 to node 2, from node 2 to node 3, and so on. A node is permitted to transmit only when it has the token. As soon as a node is finished transmitting, it passes the token to the subsequent node. Token passing is decentralized, fair, and highly efficient. However, to handle the possibility of node failures, token passing protocols are necessarily complex.

- **Random access protocols and ALOHA.** Random access protocols are so pervasive and important in computer networks, that they deserve a special top-ten listing for themselves. In a random access protocol, nodes independently transmit, resulting in the possibility of collisions. A node that transmits receives some feedback (to be discussed later), so that it learns whether or not its transmission was successful without a collision. If nodes have colliding frames, then these nodes retransmit their frames. Of course, if all the colliding nodes were to retransmit at the same time, there would again be collisions, making a bad situation even worse. The key idea behind a random access protocol is that after a transmitting node experiences a collision, it waits a random period of time before retransmitting. In this manner, the colliding nodes will hopefully retransmit at different times, thereby getting their frames to the receivers without collisions.

  One of the simplest and well-known random access protocols is ALOHA, as described on page 435 of the textbook. In ALOHA, each node has a biased coin with the probability of a head occurring equal to p. If a node's frame collides, the node flips the coin. If the result is a head, the node retransmits the frame; otherwise the node waits a frame time and then flips the coin again. A simple probabilistic analysis shows that when there are many nodes and all nodes have many frames to transmit, the fraction of the time the channel transmits without collisions is only 37 percent. Thus, ALOHA is very inefficient when many nodes have data to send.

- **Link-layer addressing.** Just as human beings have many identifiers (names, social security numbers, street addresses, and so on), so do hosts in a network. We have already learned about one such identifier, namely, the host's network-layer address, which is called the IP address in the Internet. Another important identifier is the host's link-layer address or—more commonly called—the host's MAC address. The MAC address is 48 bits and is typically written in hexadecimal notation. For example, 1A-23-F9-CD-06-9B could be a MAC address for some host. Similar to a social security number, a MAC address has a flat structure, with all bits (higher- and lower-order) bits having equal importance. A MAC address does not change no matter where the host moves, again analogous to a person's social security number, which does not change even when a person changes residences. In contrast, a host's IP address is hierarchical—with a network and a host part—and does change when the host moves from one access network to another. Just as a person may find it useful to have both a postal address and a social security number, it is useful for a host to have both a network-layer address and a MAC address. Routers also have multiple MAC addresses—one for each interface.

  In LANs, when a node (host or router) wants to send a frame to another node in the LAN, the node inserts the destination node's MAC address into the frame. For a broadcast LAN—such as Ethernet and Wi-Fi—the frame will be received by all nodes on the LAN. Each node that receives the frame checks to see if the destination MAC address matches its own MAC address; if so, the node passes the payload of the frame to the network layer; if not, the node simply drops the frame. There

is an exception to this rule, however. If the sending node uses the broadcast address (FF-FF-FF-FF-FF-FF), then all of the receiving nodes will pass the payload to their network layers.

- **ARP: translating between link-layer and IP addresses.** Suppose you are a host on a LAN and you want to send an IP datagram to another host on the same LAN; further suppose you know (perhaps from DNS) the destination host's IP address, but not its MAC address. Being a lazy guy and following standard practice, you give your datagram to the link layer and ask it to deliver the datagram to the destination host on your behalf. The link layer, of course, creates a link-layer frame, and inserts the IP datagram into the data field of the frame. But to deliver the frame to the destination host, the link layer must also insert the destination MAC address into the frame. So now we come to an interesting question: Knowing only the destination IP address, how is the sending host going to determine the destination MAC address? This task is the job of ARP, which stands for Address Resolution Protocol. Each node maintains an ARP table, providing the mappings from IP addresses to MAC addresses for nodes on the same LAN. Importantly, this ARP table is not configured by a network administrator. Instead, it is self-learning, that is, it learns about the mappings as it needs them. Specifically, if a sending node needs to translate an IP address to a MAC address, and the mapping is currently not in the table, the sending node sends an ARP query message, as part of a broadcast frame, into the LAN. The destination node—which of course knows its own MAC and IP addresses—answers with an ARP response message, providing the desired mapping.

  ARP is localized to a subnet—in particular, a host does not use ARP to determine the MAC address of a host on a different subnet. Instead, the host uses ARP to determine the MAC address of the router that is in the same subnet and that is along the path to the destination host. When the frame arrives at this router, the router extracts the IP datagram and then inserts the datagram in a new link-layer frame for forwarding to the next node (either another router or the destination host).

- **Ethernet.** Ethernet is an immensely popular LAN technology. The Ethernet frame itself provides significant insight into Ethernet and link-layer protocols in general, and should be memorized. It has exactly six fields: a preamble field used by the receiving host to synchronize its clock to the sending clock and to determine when the frame begins; source and destination MAC address fields; a data field, which carries the IP datagram; a CRC field for error checking; and a type field, which indicates the type of payload (IP datagram, ARP packet, and so on). Ethernet uses the CSMA/CD protocol, which is a random access protocol along the lines of unslotted ALOHA, but with some features that exploit the local area setting. In particular, in CSMA/CD, when a host knows that another host is transmitting, it refrains, thereby averting a collision; and when a host starts to transmit but learns shortly afterwards that another host has just started to transmit, it aborts its transmission and begins a random access procedure. Ethernet has many seen many technological variations over the years, starting at 10 Mbps and moving up to 10

Gbps. However, the Ethernet frame structure and CSMA/CD access have always been part of Ethernet. In the not-so-distant past, Ethernet used a bus topology. However, today Ethernet nodes are connected in a star topology.

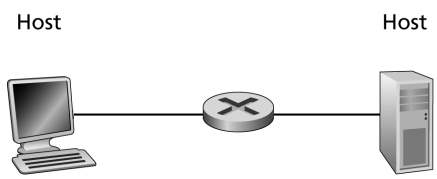
- **LAN interconnection.** LANs can be interconnected with hubs, bridges, or switches. Hubs are a physical-layer interconnection device and simply repeat bits coming from a link to all other links connected to the hub. Switches are link-layer interconnection devices, processing link-layer frames. Routers, as we saw in Chapter 4, are network-layer devices, acting on link-layer and network-layer header fields. Let's give some special attention to switches, as they illustrate some new networking concepts. Like routers, switches can forward a packet to the appropriate outbound link. This is done with the aid of a switch table, in which an entry is a mapping from a destination MAC address to a switch interface. As you recall from Chapter 4, the entries in a router's forwarding table are either configured manually or are configured via a routing protocol, such as OSPF. The entries in a switching table are acquired through a *self-learning mechanism:* when a frame arrives on one of its interfaces, the switch examines the frame's source MAC address and creates an entry for that MAC address in its table. Entries that are not refreshed with new packets are purged from the table after a (typically configurable) timeout period. Finally, if a frame arrives to the switch and there isn't an entry for the frame's destination address, the frame is broadcast into all the other links connected to the switch.
- **Link-layer protocols for a point-to-point link.** There are two types of links: broadcast links and point-to-point links. Protocols for point-to-point links are naturally simpler and more straightforward. Section 5.7 examines the issues surrounding point-to-point protocols using a specific illustrative example, namely, the Point-to-Point Protocol (PPP), which is the protocol of choice for a dial-up link for residential access. PPP provides packet framing, transparency (that is, it doesn't place constraints on the bits that are carried in PPP's payload), network-layer protocol independence, link-type independence, error detection, connection liveliness, and network-layer address negotiation. To achieve transparency, PPP uses byte-stuffing, which is used to distinguish control flag bit patterns with identical data bit patterns. Byte-stuffing is an important networking concept that appears in many protocols in many layers.
- **Asynchronous Transfer Mode: link virtualization.** ATM provides an alternative suite of protocols to the TCP/IP protocol suite. The original designers of ATM had hoped that global networks would be built with the ATM protocol suite; but much to their chagrin, the Internet and TCP/IP became dominant and pervasive, leaving little room for ATM. However, ATM merits discussion for two reasons. First, it provides an alternative protocol suite—with an alternative service model—to TCP/IP. Second, it is actually deployed at the link layer, as we discuss in Section 5.8.

Review Questions

This section provides additional study questions. Answers to each question are provided in the next section.

1. **Two-dimensional parity scheme.** Suppose the information content of a packet is the bit pattern 1110101010101111 and an even parity scheme is being used. What would the value of the checksum field be for the case of a two-dimensional parity scheme? Your answer should be such that a minimum-length checksum field is used.
2. **Cyclic Redundancy Check (CRC).** Consider the 4-bit generator G = 1001, and suppose that *D* has the value 111010. What is the value of *R*?
3. **Efficiency of slotted ALOHA.** Section 5.3 outlines a derivation of the efficiency of slotted ALOHA. In this problem, we will examine a special case. Suppose there are exactly three nodes, all with an infinite number of packets to transmit. Let p be the probability that a node transmits in any slot.
   a. As a function of p, find the probability that there is a successful transmission in any given slot.
   b. Find the value of p that maximizes this expression.
   c. What is the maximum efficiency for N = 3?
4. **Polling.** Consider a broadcast channel with *N* nodes and a transmission rate of *R* bps. Suppose the broadcast channel uses polling (with an additional polling node) for its multiple access. Suppose the amount of time from when a node completes transmission until the subsequent node is permitted to transmit (that is, the polling delay) is t_{poll}. Suppose that within a polling round, a given node is allowed to transmit at most *Q* bits. Further suppose node 1, initially with no bits to send, receives Q bits to send. What is the maximum time from when node 1 receives the bits until it can begin to send them?
5. **CSMA/CD.** In CSMA/CD, after the fourth collision, what is the probability that the node chooses K = 3? The result K = 3 corresponds to a delay of how many microseconds on a 10 Mbps Ethernet?
6. **Carrier sense and collision detection.** Suppose nodes A and B are on the same 10 Mbps Ethernet segment, and the propagation delay between the two nodes is 225 bit times. Suppose at time t = 0, B starts to transmit a frame. Suppose A also transmits at some t = x, but before completing its transmission it receives bits from B (hence, a collision occurs at A). Assuming node A follows the CSMA/CD protocol, what is the maximum value of x?

7. **Carrier sense and collision detection.** Consider two nodes A and B on the same Ethernet segment, and suppose the propagation delay between the two nodes is 225 bit times. Suppose at time t = 0, both nodes A and B begin to transmit a frame. At what time do they detect the collision? Assuming both nodes transmit a 48-bit jam signal after detecting a collision, at what time (in bit times) do nodes A and B sense an idle channel? How many seconds is this for a 10 Mbps Ethernet?
8. **Ethernet efficiency.** Consider a 100 Mbps 100BaseT Ethernet. Suppose the maximum propagation delay between any two nodes on the Ethernet is .512 microseconds. What is the efficiency of this LAN? Assume a frame length of 64 bytes and that there are no repeaters.
9. **Link-layer services and Ethernet.** Section 5.1.1 lists a number of different services that a link layer can potentially provide to the network layer. These services include: a) framing, b) medium access, c) reliable delivery, d) flow control, e) error detection, f) error correction, g) full-duplex and half-duplex. For each of these services, discuss how or how not Ethernet provides the service.
10. **Ethernet broadcast packets.** List two protocols that require Ethernet to use broadcast frames. Explain.
11. **ARP delays.** Consider transmitting a packet from host A to host B via a router, as shown below:



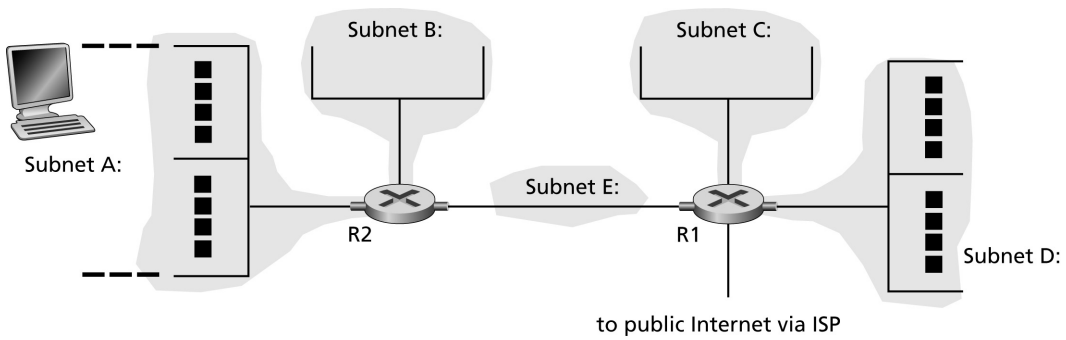
Suppose that before sending the packet, all the ARP tables (in the two hosts and in the router) are empty. Let x denote the time to transmit a packet. Let y denote the delay from beginning the transmission of an ARP query until receiving and processing an ARP response. Ignore propagation delays. Assuming host A knows the IP address of host B, what is the total delay in moving the packet from Host A to Host B?

12. **Self-learning switch.** Consider an Ethernet LAN consisting of N nodes interconnected with a switch. Suppose the switch's forwarding table is initially empty. Suppose node A wants to TCP three-way handshake with node B, where both nodes are on the LAN. Assuming this is the only traffic on the network, and there are no packet errors or loss, how many frames will be transmitted in the process of establishing the TCP connection? Assume node A knows the IP address of node B, and ARP tables have all the necessary mappings.

13. **Self-learning switch and ARP.** Repeat the above problem, but now assume that the ARP tables are also initially empty.

14. **Self-learning switch, ARP, and DNS.** Repeat the above problem, but now also suppose that Host A only knows the hostname of Host B (and not its IP address). Assume the DNS server is Host C in the LAN, and Host A knows the IP address of the DNS server.

15. **Addressing at the network and link layers, routing versus switching.** (This question concerns material from Chapters 4 and 5.) Consider the network shown below. Each of the subnets A-D contains at most 31 hosts; subnet E connects routers R1 and R2.



a. Assign network addresses to the five subnets shown above (that is, write down the addresses you have assigned).

b. Assign (write down) a full (32-bit) IP address for each the two hosts shown in subnets A and D.

c. Assign (write down) a full IP address to the router interface on subnet E.

d. What is the network prefix advertised by router R1 to the public Internet?

e. Assign (write down) a MAC address to D.

f. Does the host in A ever need to know the MAC address of the R1's interface in subnet E in order to send an IP packet to the host in D? Explain your answer in one or two sentences.

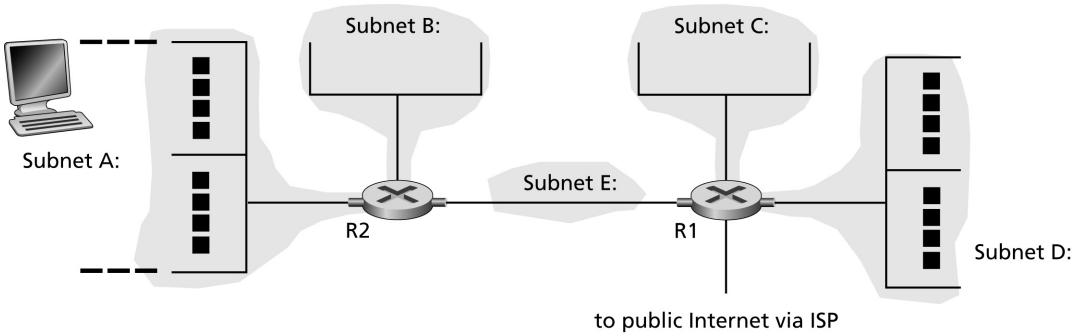
Now suppose that router R2 above is replaced by an Ethernet *switch*, S2 (Router R1 remains a router).

g. Are the interfaces that previously were in subnets A, B, and E still in the same separate three IP subnets now that R2 is replaced by S2? Explain your answer in a few sentences.

h. In order to send an IP packet to the host in D, does the host in A ever need to know the MAC address of the R1's left interface now that R2 is

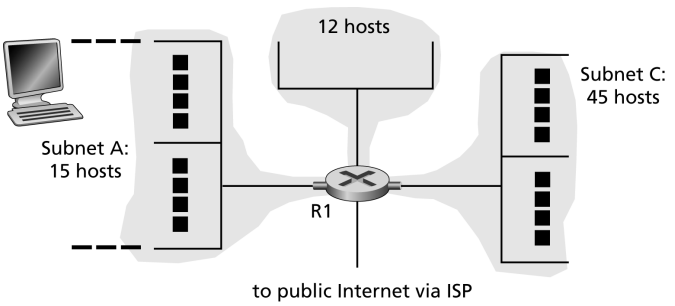
replaced by S2? If so, how does it get the MAC address of R1's left interface? Explain your answer in one or two sentences.

16. **Addressing at the network and link layers, routing versus switching.**



(This question concerns material from Chapters 4 and 5.)

Consider the network shown above Each of the subnets A-D contains at most



31 hosts; subnet E connects routers R1 and R2.

a. Consider the network shown above, consisting of a single router, R1, with three subnets A, B and C, with 15, 12, and 45 hosts respectively on these subnets. Assign an address range to the hosts in subnets A, B, and C such that only a single aggregated address need be advertised by R1 to the public Internet, and that the size of the aggregated address ranges that is advertised is minimized. In a sentence or two, explain how you arrived at your answer.

b. Assign a specific Internet address and MAC address to the host shown in subnet A, an Internet address and MAC address to the host shown in subnet C, and a MAC address and IP address to each router interface. These addresses (both Internet and MAC) can be of your own choosing. Consider an IP datagram sent from the host in Subnet A that is destined to the

host in subnet C. This IP datagram is contained in an Ethernet frame sent from the host in subnet C to router R1. What are the source and destination MAC addresses of this Ethernet frame? What are the source and destination address in the IP datagram contained in this Ethernet frame?

c. Now consider the Ethernet frame carrying this IP datagram from the router to the host in subnet C. What are the source and destination MAC addresses of this Ethernet frame? What are the source and destination addresses in the IP datagram contained in this Ethernet frame?

17. **Multiple access protocols: voice-over-IP and data.** In this chapter, we studied a number of multiple access protocols, including TDMA, CSMA, slotted Aloha, and token passing.

a. Suppose you were charged with putting together a large LAN to support IP telephony (only) and that multiple users may want to carry on a phone call at the same time. Recall that IP telephony digitizes and packetizes voice at a constant bit rate when a user is making an IP phone call. How well suited are these four protocols for this scenario? Provide a brief (one sentence) explanation of each answer.

b. Now suppose you were charged with putting together a LAN to support the occasional exchange of data between nodes (in this part of this question, there is no voice traffic). That is, any individual node does not have data to send very often. How well suited are these four protocols for this scenario? Provide a brief (one sentence) explanation of each answer.

c. Now suppose the LAN must support both voice and data and you must choose one of these multiple access strategies in order to support both applications on the same network, with the understanding that voice calls are more important than data. Which would you choose and why? How would voice and data be sent in this scenario? That is, which access protocol would you use, or adapt/modify, and why?

Answers to Review Questions

1. The minimum-length checksum is obtained by arranging the 16 bits in four rows and four columns. We then add an additional row and column for the parity bits. Below, the rightmost column and bottom row are for parity bits:

   1 1 1 0 1
   1 0 1 0 0
   1 0 1 0 0
   1 1 1 1 0
   0 0 0 1 1

2. R is the remainder of $D \cdot 2^r/G$, where $r = 3$ since G has 4 bits. Thus, we divide 1001 into 111010000 to get 111101 with remainder $R = 101$. So we send (data + CRC) 111010101. To check this result we multiply G times 111101, which gives = 111010101. To this we add R, which gives 111010000, which is exactly equal to 2^rD.

3. a. Consider one of the nodes. It has a success if and only if it transmits and the other two nodes do not transmit. The probability that it transmits is p; the probability that the second node does not transmit is $(1 - p)$; and the probability that the third node doesn't transmit is $(1 - p)$. Since each of these events are independent, the probability that only the first node transmits is $p(1 - p)(1 - p) = p(1 - p)^2$. Now, a success occurs if any of three nodes have a success. Thus the overall probability of success is probability that the first node has a success plus the probability the second node has a success plus the probability that the third node has a success, which is $3p(1 - p)^2$.

   b. To find the p that maximizes the probability of success, we differentiate $f(p) = p(1 - p)^2$, set the result to zero, and solve for p. The derivative of f(p) is $f(p) = (1 - p)^2 + 2p(1 - p) = (3p - 1)(p - 1)$. The value $p = 1$ minimizes the probability; the value $p = 1/3$ maximizes the probability.

   c. With $p = 1/3$, the probability of success (equivalently, the efficiency) is $3(1/3)(1 - 1/3)^2 = 4/9$.

4. Node 1 must wait to be polled. In the worst case, all other nodes have Q bits to send, and the Q bits arrive to node 1 just after node 1 completes a transmission. Before node 1 gets polled again, $N - 1$ other nodes transmit Q bits at rate R, giving a delay of $(N - 1)Q/R$. In addition to this, there are N polling delays. So the total wait is $(N - 1)Q/R + N\, t_{poll}$.

5. The node chooses K from the elements in the set {0, 1, 2, … ,15} with equal probability. The probability that it chooses $K = 3$ is thus 1/16. With $K = 3$,

the node waits $3 \cdot 512 = 1{,}536$ bit times. The corresponding delay over a 10 Mbps Ethernet link is (1536 bits)/(10^7 bits/sec) = 153.6 microseconds.

6. Node A senses an empty channel from time $t = 0$ to time $t = 225$. Node A can transmit at any time in this interval. At time $t = 225$, Node A senses a busy channel and will refrain from transmitting. So the maximum value of x is $x = 224$.
7. Both nodes A and B detect the collision at time $t = 225$. At time $t = 225 + 48 = 273$ both nodes stop transmitting their jam signals. The last bit of the jam signal from B arrives at A at time $t = 273 + 225 = 498$ bit times. Similarly, the last bit of the jam signal from B arrives at A at time $t = 273 + 225 = 498$ bit times. For a 10 Mbps Ethernet, this corresponds to (498 bits)/(10^7 bits/sec) = 49.8 microseconds.
8. The efficiency is $1/(1 + 5a)$ where $a = t_{prop}/t_{trans}$. We have $t_{prop} = .512$ microseconds and t_{trans} = (512 bits)/(10^8bits/sec) = 5.12 microseconds. Thus, the efficiency is $1/(1 + 5/10) = .6667$.
9. a. Framing: Ethernet encapsulates the payload (such as an IP datagram) in an Ethernet frame. Included in this encapsulation is the preamble, which helps the receiving node determine where the frame begins and helps the receiving node synchronize its clock to the frame.
   b. Ethernet provides CSMA/CD medium access.
   c. Reliable delivery: Ethernet does not provide reliable delivery. Receivers do not send ACKS or NACKS to senders; senders do not maintain timers for transmitted frames. Thus, if a receiver determines that a frame has errors, it simply discards the frame. Higher-layer protocols may eventually retransmit the frame.
   d. Ethernet does not provide flow control. Thus, if the network layer in the receiving node does not read data out of the adapter fast enough, the sender can overflow the link-layer receive buffer in the adapter.
   e. Ethernet does perform error-detection using the CRC field in the Ethernet frame. If an error is detected, it discards the frame.
   f. Ethernet does not correct bit errors.
   g. Generally, CSMA/CD is half-duplex, as packets collide if transmitted at the same time. However, if all nodes are connected through a full-duplex switch, then Ethernet is full-duplex.
10. ARP: An ARP query is encapsulated in an Ethernet broadcast frame; however, the response is sent in a unicast frame. DHCP: The DHCP discover message is also sent within an Ethernet broadcast frame (after encapsulation in an IP broadcast datagram!).

11. First Host A does an ARP query-response exchange with the router, taking y. Then it sends the packet to the router, taking x. The router does an ARP query-response exchange with Host B, taking y. Then it sends the packet to Host B, taking y. So the total time is $2y + 2x$.
12. Node A creates a TCP SYN packet, which (after encapsulation in an IP datagram) gets encapsulated into an Ethernet frame. This Ethernet frame will have B's MAC address for its destination address. Node A transmits the frame. When the frame arrives at the switch, the switch will take note of A's location and then transmit the frame onto the other $N - 1$ links, giving a total of N transmissions so far. When B receives the frame, it will send a SYNACK, encapsulated in an Ethernet frame with A's MAC address for the destination address. Thus, there are $N + 1$ frames so far. When the switch receives the frame, it will take note of B's location; it will already have an entry in its table for A and thus will only transmit the frame onto one link. Thus, there are $N + 2$ frames so far. When A receives the SYNACK it will send an ACK. Two more transmissions are required for this ACK, giving a total of $N + 4$ transmitted frames.
13. Because the ARP tables are empty, first host A must send out an ARP query within an Ethernet broadcast frame. This will generate 1 transmission at A and $N - 1$ transmissions at the switch. Then host B will respond with an ARP response, which will generate 2 transmissions, giving a total of $N + 2$ so far. In this process, host B will update its ARP table with an entry for host A. Also, during this ARP exchange, the switch will learn about the locations of hosts A and B. Thus, when host A sends a SYN, the switch can send the SYN packet directly to B. The TCP handshake will therefore generate an additional 6 Ethernet frames, giving a total of $N + 8$ frames.
14. First Host A needs to do an ARP exchange with node C to get node C's MAC address. This generates $N + 2$ Ethernet frames. This also generates entries for A and C in the switch table. Then node A must do a DNS exchange with C. This will generate 4 Ethernet frames, giving $N + 6$ frames thus far. Node A will now have B's IP address, but not B's MAC address. So A will have to do an ARP exchange with B. Since B is not yet in the switch table, the ARP exchange will generate another $N + 2$ Ethernet frames, giving a total of $2N + 8$ frames thus far. The TCP exchange will then generate another 6 frames, giving a total of $2N + 14$ frames.
15. a. Each subnet needs to address up to 31 hosts, using the rightmost 5 bits of the address. The five subnet addresses are thus x.y.z.000_/27, x.y.z.001_/27, x.y.z.010_/27, x.y.z.011_/27, x.y.z.100_/27, where the notation x.y.z.000_ means that the leftmost three bits of the fourth address byte are 000. Other answers with different bit values in bits 25, 26, 27 are also possible, as long as the five three-bit patterns used are unique.
    b. If you chose an address range x.y.z.000_/27 for network A, then the address you choose here must have these 27 leading bits, and can have any

5 remaining bits you want. If you chose an address range x.y.z.011_/27 for network D, then the address you choose here must have these 27 leading bits, and can have any 5 remaining bits you want.

c. If you chose an address range x.y.z.100_/27 for network E, then the address you choose here must have these 27 leading bits, and can have any 5 remaining bits you want.

d. x.y.x./24

e. Any 48 bit number is OK.

f. No. The host in subnet A needs to address a link-layer frame (containing the IP packet addresses to the host in D) to the R2 interface in subnet A only.

g. No. They are now all in the same subnet from an IP addressing point of view, since there is no longer any intervening router.

h. Yes. Now the host in A now needs to address its link-layer frame to the left interface of R1. The host in A gets the MAC address of the left interface of R1 using ARP. The host in A knows that in order to route its packet to the host in D, it must first send that packet (over Ethernet) to router R1, whose IP address is in the hosts routing table. Thus, it uses ARP to get the MAC address associated with the IP address of R1's left interface.

16. a. Given the stated number of hosts, subnet A requires at least 4 bits of addressing, subnet B requires at least 4 bits of addressing, and subnet C requires at least 6 bits of addressing. Let the first 3 bytes of the address for all of the hosts be X.Y.Z.

- The address for hosts in subnet C is in the range X.Y.Z.00_, where the last byte of the address begins with two zeros and the rest of the 6 bits are used to address hosts in C. Note that the second bit in the last byte is a 0. For subsets A and B, this bit will be a 1.
- The address for hosts in subnet B is in the range X.Y.Z.010_. The last byte of the address begins with 010 and the final 5 bits can be used to address the hosts in B.
- The address for hosts in subnet A is in the range X.Y.Z.011_. The last byte of the address begins with 011 (which differs from the leading 010 for subnet B, and the leading 00 for subnet C) and the final 5 bits can be used to address the hosts in A.

The size of the single aggregated network that is advertised is thus X.Y.Z.0/25-the last seven bits are used to address hosts in subnets A, B, and C.

b. • Let the host in A have IP address 128.119.40.011000001 (where we have abused notation and shown the last byte in binary format) and MAC address aa:aa:aa:aa:aa:aa.

- Let the router interface into subnet A have IP address 128.119.40.01100010, and MAC address bb:bb:bb:bb:bb:bb.
- Let the host in C have IP address 128.119.40.000000001 (where we have abused notation and shown the last byte in binary format) and MAC address cc:cc:cc:cc:cc:cc:cc.
- Let the router interface into subnet C have IP address 128.119.40.00000010, and MAC address dd:dd:dd:dd:dd:dd:dd.

The IP datagram from the host in subset A to the router interface in subnet A has IP source 128.119.40.011000001 and IP destination 128.119.40.000000001. The source MAC address is aa:aa:aa:aa:aa:aa, and the destination MAC address is bb:bb:bb:bb:bb:bb.

c. The IP datagram from the router interface in subnet C to the destination host in subnet C has IP source 128.119.40.011000001 and IP destination 128.119.40.000000001-the same answer as in (b). The source MAC address is cc:cc:cc:cc:cc:cc:cc, and the destination MAC address is dd:dd:dd:dd:dd:dd:dd.

17. a. TDMA works well here since it provides a constant bit rate service of 1 slot per frame. CSMA will not work as work well here (unless the channel utilization is low) due to collisions and variable amount of time needed to access the channel (for example, channel access delays can be unbounded) and the need for voice packets to be played out synchronously and with low delay at the receiver. Slotted Aloha has the same answer as CSMA. Token passing works well here since each station gets a turn to transmit once per token round, yielding an essentially constant bit rate service.

b. TDMA would not work well here as if there is only one station with something to send, it can only send once per frame. Hence, the access delays are long (one half frame time on average), and the throughput over a long period of time is only 1/N of the channel capacity. CSMA would work well since at low utilization, a node will get to use the channel as soon as it need to. Slotted Aloha has the same answer as CSMA Token passing would work better than TDMA but slightly less well than CSMA and Slotted Aloha, since it must wait for the token to be passed to the other stations (who likely wouldn't use it) before sending again.

c. Here are two possible answers. One approach would be to divide the channel into two "pieces"-one for data packets and one for voice. This can be accomplished by assigning some number of TDMA slots for voice calls (for example, one slot to each user). Also, add some additional slots and allow the stations with data to send to perform random access (for example, slotted aloha or CSMA) within those data slots only. A second approach would be to use token passing with priorities, and give priority to voice packets.