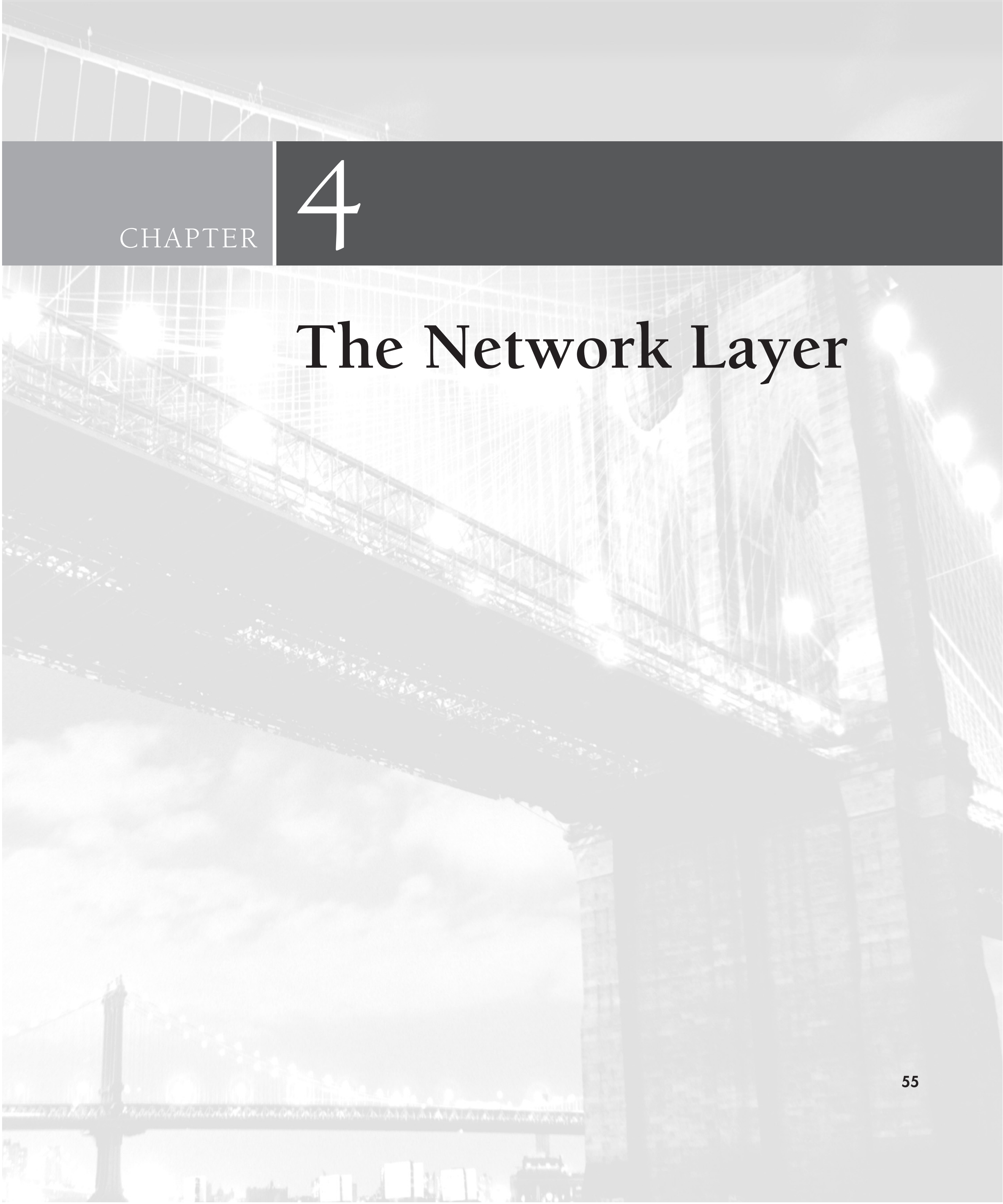


CHAPTER 4

The Network Layer

Most Important Ideas and Concepts from Chapter 4

- **Routing protocols: link-state and distance vector.** It's often said that the network layer is all about routing, and there's a lot of truth in that statement. In Chapter 4 we cover two basic approaches to routing: *link-state* and *distance vector*. In the link-state approach, all nodes *broadcast* their link-state information (the existence of links to neighbors and the links' costs) to all nodes in the network, giving all nodes a common global view of the network's topology. Each node then runs a shortest path algorithm (such as Dijkstra's algorithm) using this network topology to determine the least-cost path from itself to all other nodes in the network. Packets are then routed along these least-cost paths. Distance vector algorithms are distributed in nature—each node communicates only with its directly-connected neighbors, exchanging its estimates of its least cost to reach each node in the network. For routing purposes, a node also maintains the identity of the first-hop (directly-connected) neighbor along the least cost path to each destination. Through an iterative process of (i) pairwise exchange of distance vectors; (ii) recomputation of least cost paths given new distance vector information received from neighbors; and (iii) transmission of new distance vector information to neighbors if a node's estimates of its least cost to a destination has changed, the distance vector algorithm converges to a set of distance vectors for which each node has its own least cost to each destination, and the next-hop neighbor along that least cost path. Note that with distance vector algorithms, a node does not know the entire least cost path from source to destination; however, each node along the way knows the identity of the next node along this least cost path. Note also that neither link-state nor distance vector algorithms necessarily route along least congested paths (unless link costs reflect the current congestion state of a link).
- **Inter-AS versus Intra-AS routing.** Link-state and distance vector algorithms conceptually consider a flat (non-hierarchical) network topology. Internet routing, however, has a distinct two-level hierarchy based on the notion of an autonomous system—a network that is under the control of a single organization. When routing *within* an autonomous system (intra-AS routing), the organization can choose its own routing algorithm; RIP, OSPF, or IS-IS are popular choices. When routing *among* autonomous systems (inter-AS routing), each AS can be considered as a single node in an AS graph, with the inter-AS routing algorithm determining how packets are routed among ASs. BGP is the (only) Inter-AS routing algorithm in use in the Internet. For this reason, BGP is sometimes referred to as the "glue" that binds the Internet.
- **Service models for the network layer.** Students often find the discussion of the network-layer service model (or service models in general) to be boring. But the idea of the network service model, which defines the end-end packet delivery service that will be provided to the transport layer, is nonetheless extremely important. We have seen that the Internet adopts a best-effort model, in which no

guarantees are made at the network layer regarding how long it will take for a datagram to reach its destination, or whether the datagram will even make it to its destination in the first place.

- **Datagrams and virtual circuits.** Datagrams and virtual circuits represent the two major approaches to network layer service that have been taken over the years. The Internet is a datagram network, in which each network-layer datagram carries the IP address of the final destination of the datagram. This address is used by a router in forwarding the datagram toward its final destination. In a virtual circuit (VC) network, each packet of data (called a "cell" in ATM VC networks) carries a VC number, which is used by a switch in forwarding the packet of data toward its destination. In a VC network, when a call is made between a source and destination, a call setup procedure is needed to create state in each switch on the end-to-end path that matches the VC number for this call with the outgoing switch port to which the VC's packets of data will be forwarded. Similarly, a call-teardown procedure is needed in VC networks.
- **Forwarding versus routing.** Forwarding and routing are two of the main functions of a network layer. *Forwarding* refers to the per-router action of moving a packet arriving at an input port to the appropriate output port. In a datagram network, the output port is determined by looking up the arriving datagram's destination in a forwarding table, and using a longest-prefix matching algorithm to find the entry that contains the identity of the output port to which the arriving datagram should be forwarded; in a VC network, this is done by looking up the VC of the arriving unit of data in the VC lookup table. *Routing* refers to the process of determining the end-to-end path that a packet will take through the network. We studied both link-state and distance vector routing algorithms.
- **Addressing.** Perhaps the only network-layer topic that students think is more boring than service models is addressing. But understanding Internet addressing is absolutely crucial to understanding how the network layer works! Everyone knows that the IP address is 32-bits long, and is often represented in dotted-decimal notation. More importantly, the addresses assigned to Internet devices (consisting of a network part and a host part) deeply reflect the structure of the network. Figure 4.16 on page 333 of the textbook shows that all hosts on the same subnet must have the same network part of their address; in Chapter 5 we will see that this ability to determine whether or not a destination is on the same network (in an IP addressing sense) will be used to determine whether a datagram can be sent directly to its final destination over the single network (in an IP addressing sense) that directly connects the two nodes, or whether the datagram must be forwarded to an intervening router. As we saw in Figure 4.18 on page 336 of the textbook, the structure of the addresses within an AS also allows for route aggregation, allowing a BGP router to advertise an address prefix that is common to nodes in the AS.
- **Delay and loss within a router.** In Chapters 1 through 3, we made many references to datagrams being lost or delayed within the network. In Section 4.3 (specifically,

Section 4.3.4), where we explore the inner structure of a router, we see that queuing delay and loss occur at the router's input and output ports. At the output port, queuing occurs because packets arrive to the output port at a rate that is faster than the outgoing link rate; hence, a queue of packets awaiting transmission begins to grow, and packet delay increases. If the memory used to hold queued packets becomes full, arriving packets will be dropped, or a packet will be dropped from the queue to make room for an arriving packet. At an input port, a queue will form when the rate at which the switching fabric is able to forward packets to the output ports is less than the rate of incoming packets to that input port.

- **Border Gateway Protocol (BGP).** We noted above that BGP is the *only* protocol used to route datagrams among autonomous systems, and thus it is the "glue" that binds the Internet together. We saw that BGP allows an autonomous system to choose which path it uses to reach a destination. Indeed, that is a local policy decision that is left to the network manager. More importantly, an autonomous system can also control which routes it advertises to its neighbors; this too is a local policy decision. For example, if an AS advertised no routes, then none of its neighbors would route any traffic to that AS. Of course, an AS that receives no incoming traffic isn't very useful!
- **Tunneling.** In Figure 4.24 on page 350 in the textbook, we see that tunneling can be used to connect two routers logically over a path that contains multiple routers. This allows two Ipv6 routers to exchange IPv6 datagrams with each other, via routers that only "speak" IPv4. This is done by having the router at the source end of the tunnel encapsulate the IPv6 datagram within an IPv4 datagram, and address the IPv4 datagram to the IPv4 address for the destination side of the tunnel; the upper-layer-protocol field is set to type 41 to indicate to the destination side of the tunnel that the IPv4 datagram contains an IPv6 datagram. When the destination end of the tunnel receives an IP datagram addressed to itself, with a protocol number of 41, it recognizes that this is an IPV6-within-IPv4 datagram, extracts the IPv6 datagram, and forwards the IPv6 datagram on as needed.
- **The IP datagram.** The message here is plain and simple: all networking students really should know what an IPv4 datagram looks like (see Figure 4.13 on page 326 of the textbook). While it may not be exciting, like broccoli, it is "good for you." Actually, there are not *that* many fields, and the meaning/use of each of the fields is fairly straightforward. Students are seldom asked to memorize all of the fields, but a favorite exam question is to ask students to describe five or six fields.

Review Questions

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

1. **Virtual circuit and datagram networks.** Identify three important differences between a virtual circuit network (for example, ATM) and a datagram network (for example, Internet).
2. **Virtual circuits.** Consider Figure 4.3 on page 308 of the textbook and the virtual circuit (VC) table for router R1 shown above the figure. Write the set of VC table entries in router R2 in Figure 4.3 that are needed to ensure that the VC tables in R1 and R2 are consistent, that is, that the VCs entering/leaving interface 2 in router R1 are consistent with the VCs leaving/entering interface 1 in router R2.
3. **IP addressing.**
   a. Write the IP address 129.17.129.97 in its binary form.
   b. Consider an IP subnet with prefix 129.17.129.97/27. Provide the range of IP addresses (of form xxx.xxx.xxx.xxx to yyy.yyy.yyy.yyy) that can be assigned to this subnet.
   c. Suppose an organization owns the block of addresses of the form 129.17.129.97/27. Suppose it wants to create four IP subnets from this block, with each block having the same number of IP addresses. What are the prefixes (of form xxx.xxx.xxx/y) for the four IP subnets?
4. **IP datagram.** Suppose a host has a file consisting of 2 million bytes. The host is going to send this file over a link with an MTU of 1,500 bytes. How many datagrams are required to send this file?
5. **IP fragmentation.** Consider sending a 2,000-byte datagram into a link with a MTU of 980 bytes. Suppose the original datagram has the identification number 227. How many fragments are generated? For each fragment, what is its size, what is the value of its identification, fragment offset, and fragment flag?
6. **Longest prefix matching.** Consider a datagram network using 32-bit host addresses. Suppose that a router has three interfaces, numbered 0 through 2, and that packets are to be forwarded to these link interfaces as follows. Any address not within the ranges in the table below should not be forwarded to an outgoing link interface. Create a forwarding table using longest prefix matching.

Destination address range	Outgoing link interface
00000000 00000000 00000000 00000000 through 00000001 11111111 11111111 11111111	0
01010101 00000000 00000000 00000000 through 01010101 11111111 11111111 11111111	1
01010110 00000000 00000000 00000000 through 01010111 11111111 11111111 11111111	2

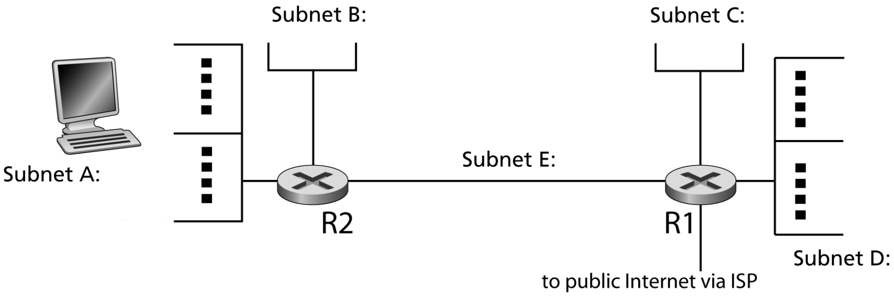
7. **Longest prefix matching.** Consider the same datagram network using 32-bit host addresses, and a router that has three interfaces, numbered 0 through 2 (see Question 6). Packets are to be forwarded to these link interfaces as follows. The address ranges for the first, third, and fourth entries in the table below are the same as in Question 6; the second entry below is new. Any address not within the ranges in the table below should not be forwarded to an outgoing link interface.

Destination address range	Outgoing link interface
00000000 00000000 00000000 00000000 through 00000001 11111111 11111111 11111111	0
00000000 00000000 10000000 00000000 through 00000000 00000000 11111111 11111111	2
01010101 00000000 00000000 00000000 through 01010101 11111111 11111111 11111111	1
01010110 00000000 00000000 00000000 through 01010111 11111111 11111111 11111111	2

8. **Longest prefix matching.** Consider the same datagram network using 32-bit host addresses, and a router that has three interfaces, numbered 0 through 2 (see Question 7). Packets are to be forwarded to these link interfaces as follows. The address ranges for the second and third entries in the table below are the same as in the earlier problem; the first entry below has an upper-end of the address range that is smaller than before. Any address not within the ranges in the table below should not be forwarded to an outgoing link interface.

Destination address range	Outgoing link interface
00000000 00000000 00000000 00000000 through 00000001 10000000 00000000 00000000	0
01010101 00000000 00000000 00000000 through 01010101 11111111 11111111 11111111	1
01010110 00000000 00000000 00000000 through 01010111 11111111 11111111 11111111	2

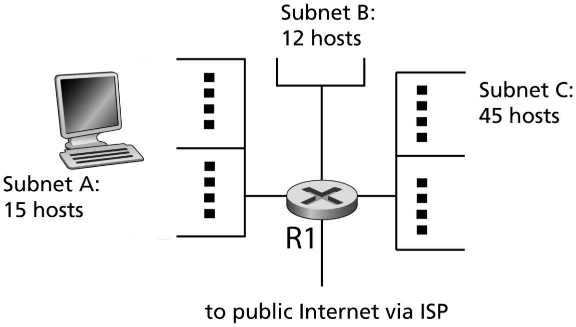
9. **IP addressing.** Consider the network shown below. Each of the subnets A-D contains at most 30 hosts; subnet E connects routers R1 and R2.



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

b. Suppose that there are 17 hosts in A–D. Does your answer to Question 9a) change? If so why or why not?

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



10. **IP addressing.**

    a. Consider an Internet address of the form 129.19.40.0/23. What does the /23 signify?

    b. 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 so that only a single aggregated address needs to be advertised by R1 to the public Internet, and that the size of the advertised aggregated address range is minimized. In a sentence or two, explain how you arrived at your answer.

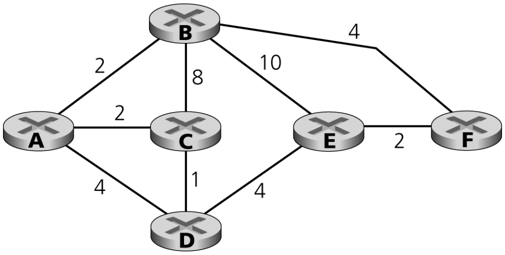
11. **NAT.** Consider the scenario shown in Figure 4.20 on page 340 of the textbook. Suppose that host 10.0.0.2 initiates a connection, using source port 5500 to a Web server listening at port 80 at 128.119.40.186.

    a. Complete the NAT translation table for this TCP connection.

    b. What are the source and destination IP addresses and port numbers on the IP datagram arriving to the WAN side of the router with interface address 138.76.29.7?

12. **Tunneling.** How does the router at the destination end of a tunnel (see Figure 4.24 on page 350 of the textbook) know that the IPv4 datagram contains an IPv6 datagram that it should extract from the IPv4 packet?

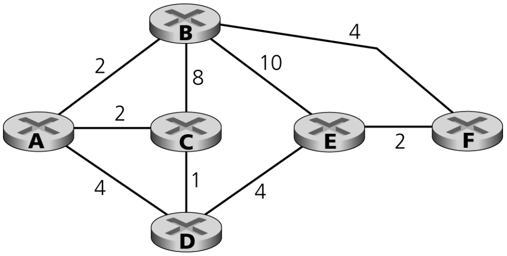
13. **Dijkstra's (link-state) algorithm.** Consider the network shown below. Show the operation of Dijkstra's (link-state) algorithm for computing the least cost path from D to all destinations. What is the shortest path from D to B, and what is the cost of this path?



14. **Dijkstra's (link-state) algorithm (more).** Consider the network shown in Question 13. Show the operation of Dijkstra's (link-state) algorithm for computing the least cost path from E to all destinations. What is the shortest path from E to B, and what is the cost of this path?

15. **Dijkstra's (link-state) algorithm (even more).** Consider the network shown in Question 14. Show the operation of Dijkstra's (link-state) algorithm for computing the least cost path from B to all destinations. What is the shortest path from B to D, and what is the cost of this path?

16. **Distance vector algorithm.** Consider the network below.



a. What are A, B, C, D, E, and F's distance vectors? Note: you do not have to run the distance vector algorithm; you should be able to compute distance vectors by inspection. Recall that a node's distance vector is the vector of the least cost paths from itself to each of the other nodes in the network.

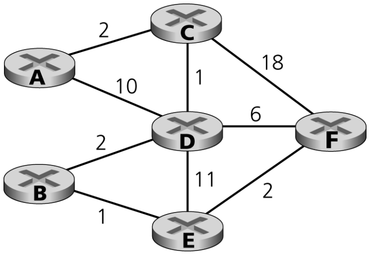
b. Now consider node C. From which other nodes does C receive distance vectors?

c. Consider node C again. Through which neighbor will C route its packets destined to E? Explain how you arrived at your answer, given the distance vectors that C has received from its neighbors.

d. Consider node E. From which other nodes does E receive distance vectors?

e. Consider node E again. Through which neighbor will E route its packets destined to B. Explain how you arrived at your answer, given the distance vectors that E has received from its neighbors.

17. **Distance vector algorithm (more).** Consider the network below.



a. What are A, B, C, D, E, and F's distance vectors? Note: you do not have to run the distance vector algorithm; you should be able to compute distance vectors by inspection. Recall that a node's distance vector is the vector of the least cost paths from itself to each of the other nodes in the network.

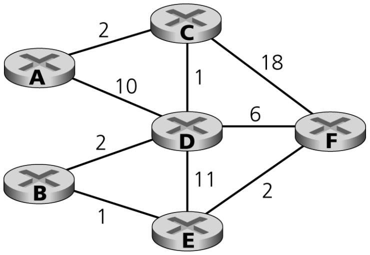
b. Consider node C. From which other nodes does C receive distance vectors?

c. Consider node C again. Through which neighbor will C route its packets destined to F? Explain how you arrived at your answer, given the distance vectors that C has received from its neighbors.

d. Consider node B. From which other nodes does E receive distance vectors?

e. Consider node B again. Through which neighbor will B route its packets destined to C? Explain how you arrived at your answer, given the distance vectors that B has received from its neighbors.

18. **Distance vector algorithm (even more).** Consider the network below.



a. What are the initial distance vectors in A, C, D and F, before the distance vector algorithm begins executing?

b. Suppose that node A sends its distance vector to C (and that no other distance vectors are exchanged). What are the distance vectors in A, C, D, and F?

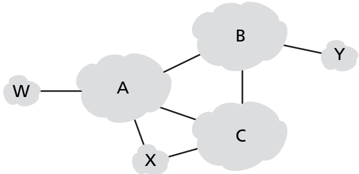
c. Suppose that node D sends its distance vector to C (and that no other distance vectors are exchanged). What are the distance vectors in A, C, D, and F?

19. **BGP.** Consider the network below in which network W is a customer of ISP A, network Y is a customer of ISP B, and network X is a customer of both ISPs A and C.

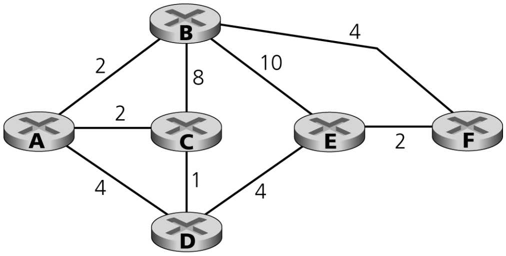
a. What BGP routes will A advertise to X?

b. What routes will X advertise to A?

c. What routes will A advertise to C? For each answer provide a one-sentence explanation.



20. **Minimum spanning tree.** Consider the network shown below and find the minimum spanning tree that connects all nodes.



21. **Minimum spanning tree.** Consider the network from Question 20. Find the set of shortest paths from all nodes to A (and indicate these paths in the graph using thicker shaded lines). Then, using arrows like those shown in Figure 4.41 on page 388 of the textbook, indicate the links over which packets will be forwarded using reverse path forwarding, and the links over which packets will not be forwarded, given that node A is the source.

Answers to Review Questions

1. a. A virtual circuit requires call setup.

   b. A virtual circuit has call teardown.

   c. In a VC network, a packet carries a VC ID rather than a destination address.

   d. Resources can be allocated to a call/connection in a VC network (typically during call setup).

2. A partial VC table for R2 is shown below. Note that for incoming interface 1, there must be entries for VC numbers 22 and 17, since the VC table for R1 (shown on page 308 of the textbook) has entries for outgoing interface 2, with VC numbers 22 and 17 (and interface 2 in R1 is connected to interface 1 in R2). Similarly, since interface 2 in R1 has an incoming VC number 63, R2 must have a VC with outgoing VC number 63 on R2 interface 1.

Incoming interface	Incoming VC number	Outgoing interface	Outgoing VC number
1	22		
1	17		
		1	63

3. a. 10000001 00010001 10000001 01100001

   b. 10000001 00010001 10000001 01100000 to 10000001 00010001 10000001 01111111 or equivalently, 129.17.129.96 to 129.17.129.127

   c. There are 32 addresses in the range; we give 8 addresses to each block; thus, 129.17.129.96 to 129.17.129.103, 129.17.129.104 to 129.17.129.111, 129.17.129.112 to 129.17.129.119, and 129.17.129.120 to 129.17.129.127.

4. Assume the data is carried in TCP segments, with each TCP segment also having 20 bytes of header. Then each datagram can carry $1{,}500 - 40 = 1460$ bytes of the file as follows:

$$\text{Number of datagrams required} = \left\lceil \frac{2 \times 10^6}{1460} \right\rceil = 1370.$$

   All but the last datagram will be 1,500 bytes; the last datagram will be $1{,}260 + 40 = 1{,}300$ bytes. Note that the host creates datagrams and not fragments.

5. The maximum size of data field in each fragment $= 960$ (20 bytes IP header). Thus, the number of required fragments

$$= \left\lceil \frac{2000 - 20}{960} \right\rceil = 3$$

Each fragment will have identification number 227. Each fragment except the last one will be of size 980 bytes (including IP header). The last datagram will be of size 80 bytes (including IP header). The offsets of the three fragments will be 0,120, 240. The first two fragments will have flag = 1; the last fragment will have flag = 0.

6.

Prefix	Outgoing link interface
0000000	0
01010101	1
0101011	2

Since all addresses beginning with 0000000 are routed to interface 0, all addresses beginning with 0101010101 are forwarded to interface 1, and all addresses beginning 0101011 are forwarded to interface 2, we only need a single table entry for these interfaces.

7.

Prefix	Outgoing link interface
0000000	0
00000000 00000000 1	2
01010101	1
0101011	2

Because of the longest prefix matching rule, all addresses beginning with 0000000 00000000 1 are now forwarded to interface 2, while all other addresses beginning with 00000000 are forwarded to interface 0. The forwarding for prefixes 01010101 and 0101011 are the same as before.

8.

Prefix	Outgoing link interface
00000000	0
00000001 0	0
00000001 10000000 00000000 00000000	0
01010101	1
0101011	2

The tricky part of this question is to make sure that no packets with addresses higher than 00000001 10000000 00000000 00000000 are routed to interface 0.

9. a. Each subnet needs to address up to 30 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, and x.y.z.100/27, where we have shown the first three binary digits of the last byte of the addresses explicitly. More properly, these addresses are x.y.z.0/27, x.y.z.32/27, x.y.z.64/27, x.y.z.96/27, and x.y.z.128/27 in dotted decimal notation. Other answers with different bit values in bits 25, 26, and 27 are also possible, as long as the five three-bit patterns used are unique.

   b. The answer stays unchanged. In order to address 17 hosts, 5 bits are still needed, and so the network part of the address will be 27 bits long again.

   c. x.y.z./24

10. a. The /23 signifies that the network part of the host address is the leftmost 23 bits.

    b. 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 are in the range X.Y.Z.00CCCCCC, where the last byte of the address begins with two 0's 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 subnets A and B, this bit will be a 1.
    - The address for hosts in subnet B are in the range X.Y.Z.010BBBBB. 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 are in the range X.Y.Z.011AAAAA. 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.

11. a.

NAT translation table	
WAN side	LAN side
138.76.29.7, 5002	10.0.0.2, 5500

    Note that the use of 5002 on the WAN side is arbitrary. The NAT box will simply use an unused port number.

    b. For the datagram returning from the Web sever: source IP: 128.119.40.186, source port: 80, dest. IP: 138.76.29.7, dest. Port: 5002.

12. First, since the IPv4 datagram is addressed to the router at the destination end of the tunnel, that router knows it must do something with that IPv4 datagram when it arrives (after all, it is the destination of the IPv4 datagram!). The upper-layer protocol field value in the IPv4 datagram tells the router that the IPv4 datagram contains an encapsulated IPv6 datagram (just as this upper-layer protocol field value is used to indicate that the IP datagram contains a TCP or UDP segment).

13.

N	D(A),p(A)	D(B),p(B)	D(C),p(C)	D(E),p(E)	D(F),p(F)
D	4,D	infty	1,D	4,D	infty
DC	3,C	9,C		4,D	infty
DCA		5,A		4,D	infty
DCAE		5,A			6,E
DCAEB					6,E

The shortest path from D to B is D C A B. The cost of this path is 5.

14.

N	D(A),p(A)	D(B),p(B)	D(C),p(C)	D(D),p(D)	D(F),p(F)
E	infty	10,E	infty	4,E	2,E
EF	infty	6,F	infty	4,E	
EFD	8,D	6,F	5,D		
EFDC	7,C	6,F			
EFDCB	7C				

The shortest path from E to B is E F B. The cost of this path is 6.

15.

N	D(A),p(A)	D(C),p(C)	D(D),p(D)	D(E),p(E)	D(F),p(F)
B	2,B	8,B	infty	10,B	4,B
BA		4,A	8,A	10,B	4,B
BAC			5,C	10,E	4,B
BACF			5,C	6,E	
BACFD				6,E	

The shortest path from B to D is B A C D. The cost of this path is 5.

16. a.

	Destination					
node	A	B	C	D	E	F
A	0	2	2	3	7	6
B	2	0	4	5	6	4
C	2	4	0	1	5	7
D	3	5	1	0	4	6
E	7	6	5	4	0	2
F	6	4	7	6	2	0

b. From its neighbors, nodes A, B, and D. Note that C does not receive distance vectors from nodes E and F, since they are not direct neighbors.

c. See page 358 in the textbook for notation.

C's cost to E via B is $c(C,B) + D_B(E) = 8 + 6 = 14$

C's cost to E via A is $c(C,A) + D_A(E) = 2 + 7 = 9$ (note that A's shortest path to E is through C!)

C's cost to E via D is $c(C,D) + D_D(E) = 1 + 4 = 5$

Thus, C will route to E via D, since that path through D has minimum cost.

d. From its neighbors, nodes B, D, and F. Note that E does not receive distance vectors from nodes A and C, since they are not direct neighbors.

e. See page 358 in the textbook for notation.

E's cost to B via B is $c(E,B) + D_B(B) = 10 + 0 = 10$

E's cost to B via D is $c(E,D) + D_D(B) = 4 + 5 = 9$

E's cost to B via F is $c(E,F) + D_F(B) = 2 + 4 = 6$

Thus, E will route to B via F, since that path through F has minimum cost.

17. a.

	Destination					
node	A	B	C	D	E	F
A	0	5	2	3	6	8
B	5	0	3	2	1	3
C	2	3	0	1	4	6
D	3	3	1	0	3	5
E	6	1	4	3	0	2
F	8	3	6	5	2	0

b. From its neighbors, nodes A, D, and F. Note that C does not receive distance vectors from nodes B and E, since they are not direct neighbors.

c. See page 358 in the textbook for notation.

C's cost to F via A is $c(C,A) + D_A(F) = 2 + 8 = 10$ (note that A's shortest path to F is through C!)

C's cost to F via D is $c(C,D) + D_D(F) = 1 + 7 = 6$

C's cost to F via F is $c(C,F) + D_F(F) = 18 + 0 = 18$

Thus, C will route to F via D, since that path through D has minimum cost.

d. From its neighbors, nodes D and E. Note that E does not receive distance vectors from nodes A, C, and F, since they are not direct neighbors.

e. See page 358 in the textbook for notation.

B's cost to C via D is $c(B,D) + D_D(C) = 2 + 1 = 3$

B's cost to C via E is $c(B,E) + D_E(C) = 1 + 4 = 5$

Thus, B will route to C via D, since that path through D has minimum cost.

18. a.

	Destination					
node	A	B	C	D	E	F
A	0	infty	2	10	infty	infty
C	2	infty	0	1	infty	18
D	10	2	1	0	11	6
F	infty	infty	18	6	2	0

b. C’s distance table is unchanged. Since A’s distance table does not cause C to learn of any new shorter paths to any of the destinations.

	Destination					
node	A	B	C	D	E	F
A	0	infty	2	10	infty	infty
C	2	infty	0	1	infty	18
D	10	2	1	0	11	6
F	infty	infty	18	6	2	0

c.

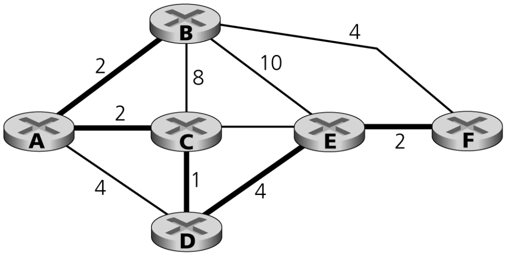
	Destination					
node	A	B	C	D	E	F
A	0	infty	2	10	infty	infty
C	2	3	0	1	12	7
D	10	2	1	0	11	6
F	infty	infty	18	6	2	0

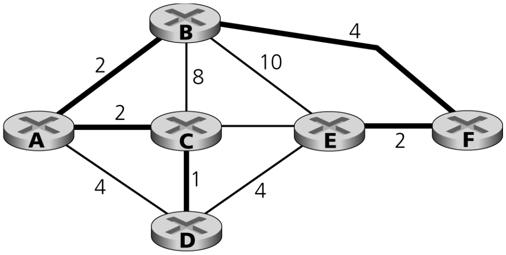
19. a. A will advertise that it can reach w and y, since x needs to know which networks its provider can reach. It may also advertise that it can reach B and C. However, if B and C are only transit networks (that is, only providing service to/from their customers networks), then A would not have to advertise B and C to x.

b. X will not advertise any routes to A, since otherwise A might try to route through x, and x is a customer network, not a transit network.

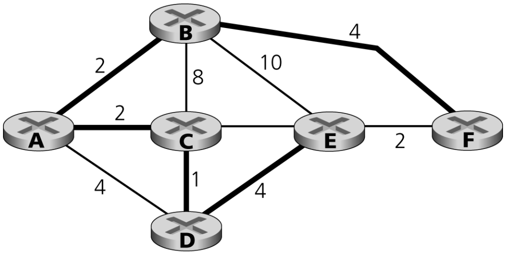
c. A will advertise that it can reach w and x. Note that since C is a peer network, A will only advertise its customers to X. In particular, A wouldn’t advertise y to C, since that might cause C to route to y via A.

20. The following spanning trees have the same minimal cost.

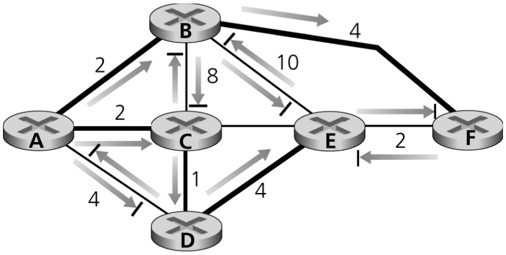




21. The set of shortest paths is:



The forwarding behavior of reverse path forwarding is:


A
B
C
D
E
F
2
2
8
10
4
4
1
4
2