CHAPTER 20

PRACTICE SET

Questions

Q20-1. According to the principle we mentioned in the text, the shortest path is the inverse of the original one. The shortest path is G → E → B → A.

Q20-2. Each AS is independent, which means that it can run one of the two common intradomain routing protocols (RIP or OSPF). On the other hand, the whole Internet is considered as one entity, which means that we must run only one interdomain routing protocol (the common one is BGP).

Q20-3. The three ASs described in the text are *stub*, *multihomed*, and *transient.* The first two do not allow transient traffic; the third does. The stub and multi-homed ASs are similar in that they are either the sink or source of traffic; the first is connected to only one other AS, but the second is connected to more than one ASs.

Q20-4. If the AS is small, it is normally recommended to consider it as only one area (the backbone area) to reduce the overhead of information exchange between areas.

Q20-5. The source and destination IP addresses in datagrams carrying payloads between the hosts are the IP addresses of the hosts; the IP addresses carrying routing update packets between routers are IP addresses of the routing interfaces from which the packets are sent or received. This shows that a router needs as many IP addresses as it has interfaces.

Q20-6. RIP messages are short with clear message boundaries. It is not efficient to use the service of TCP with all of the connection establishment and connection teardown overhead.

Q20-7. Although RIP is running as a process using the service of the UDP, the process is called a *daemon* because it is running all the time in the background. Each router acts both as a client and a server; it acts as a client when there is a message to send; it acts as a server when a message arrives.

Q20-8. Each datagram has a different source IP address: the IP address of the interface from which it is sent out (a router can have only one immediate neighbor on each interface). Each datagram also has a different destination IP address: the IP address of the router interface at which it arrives.

Q20-9. BGP is designed to create semi-permanent communication between two BGP speakers; this requires the service of TCP. A connection is made between the two speakers and remains open, while the messages are exchanged between them. UDP cannot provide such a service.

Q20-10. The path-vector routing algorithm is actually distance-vector routing using the best path instead of the shortest distance as the metric. Each node first creates a forwarding table, assuming it can only reach immediate neighbors. The forwarding table is gradually improved as path vectors arrive from the immediate neighbors.

Q20-11. It cannot. A link needs to be advertised in a router link LSP; a network needs to be advertised in a network link LSP.

Q20-12. We can say that a number of hops in RIP is the number of networks a packet travels to reach its final destination. The first network, in which the original host is located, is normally not counted in this calculation because the source host does not take part in routing. To reduce the traffic of exchanging routing updates, the hosts in the Internet do not take part in this process. This is done because the number of hosts in the Internet is much larger than the number of routers. Including hosts in this process makes the routing-update traffic unbearable.

Q20-13. In RIP, each router just needs to share its distance vector with its neighbor. Since each router has one type of distance vector, we need only one update message. In OSPF, each router needs to share the state of its links with every other router. Since a router can have several types of links (a router link, a network link, …), we need several update messages.

Q20-14. We need to have OSPF processes that run all the time because we never know when an OSPF message will arrive. These processes are running at the network layer, not at the application layer. They are normally referred to as daemons.

Q20-15. Link-state routing uses Dijkstra's algorithm to first create the shortest-path tree before creating the forwarding table. The algorithm needs to have the complete LSDB to start.

Q20-16. The intradomain routing routes the packet inside an autonomous system that is totally in the control of the organization. On the other hand, the interdomain routing routes the packet through an autonomous system that is out of the control of the organization; the organization needs to apply a policy to decide through which AS the packet should pass.

Q20-17. OSPF divides an AS into areas, in which routing in each area is independent from the others; the areas only exchange a summary of routing information between them. RIP, on the other hand, considers the whole AS as one single entity.

Q20-18.

a. router link **b.** router link **c.** network link

Q20-19. The type of payload can be determined from the value of the protocol field. The protocol field value for ICMP is 01; for OSPF, it is 89.

Q20-20. According to the principle we mention in the text, the shortest path from A to N can be found in two steps. We first use the shortest path from A to H to move to node H. We then use the shortest path from node H to N. The result is shown below:

$$A \rightarrow B \rightarrow H \rightarrow G \rightarrow N$$

Q20-21. The following shows the use of each attribute:

a. The LOCAL-PREF is used to implement the organization policy.

b. The AS-PATH defines the list of autonomous systems through which the destination can be reached.

c. The NEXT-HOP defines the next router to which the data packet should be forwarded.

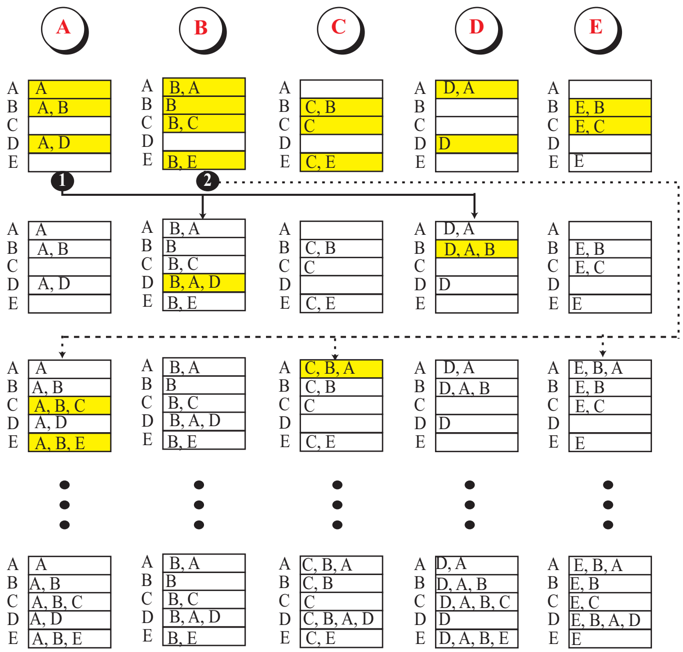
Problems

P20-1. We have

$$D_{xy} = \min \{(c_{xa} + D_{ay}), (c_{xb} + D_{by}), (c_{xc} + D_{cy}), (c_{xd} + D_{dy})\}$$

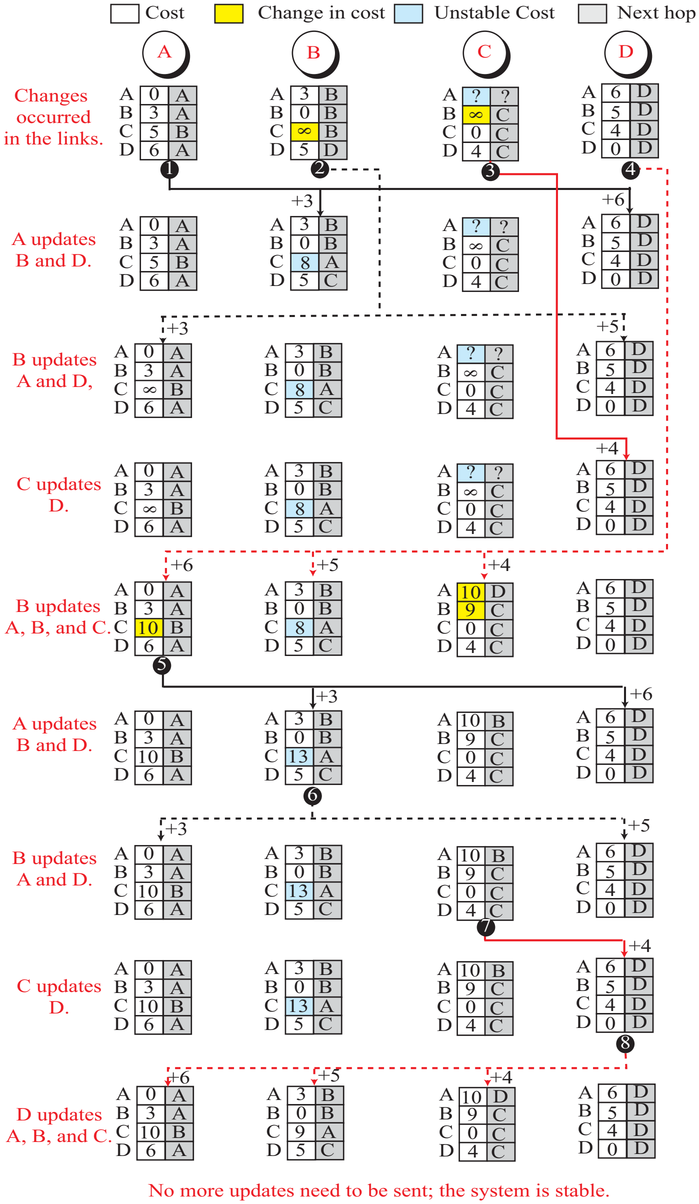
$$D_{xy} = \min \{(2 + 5), (1 + 6), (3 + 4), (1 + 3)\} = \min \{7, 7, 7, 4\} = 4$$

P20-2. The following shows the initialization and updates.



P20-3. Router R1, using its OSPF forwarding table, knows how to forward a packet destined for N4. R1 announces this reachability to R5 using an eBGP session. R5 adds an entry to its RIP forwarding table that shows R1 as the next router for any packet destined for N4.

P20-4. The following shows how the forwarding tables will be changed.



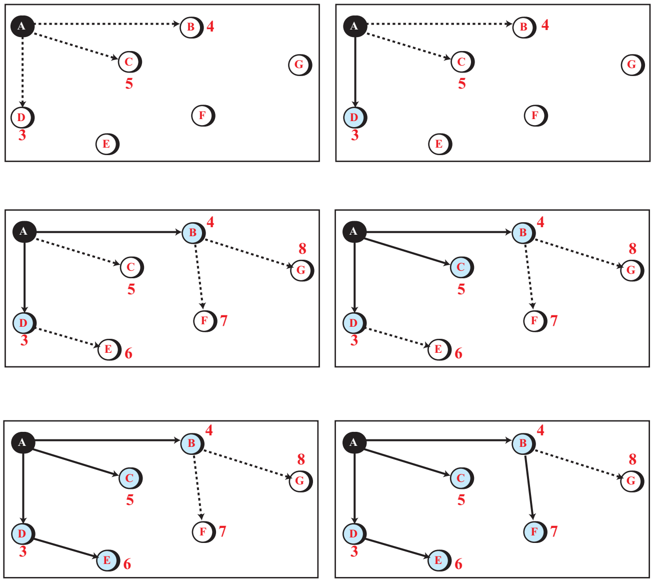
Note that there are some unstable cost values that are not finalized. These wrong pieces of information may create looping in the system; the packet may bound back and forth until the system becomes stable. Eight updates are needed to stabilize the system.

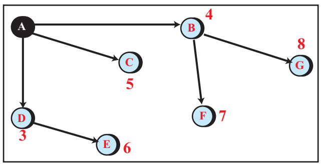
P20-5. The number of searches in each iteration of Dijkstra's algorithm is different. In the first iteration, we need n number of searches, in the second iteration, we need $(n - 1)$, and finally in the last iteration, we need only one. In other words, the total number of searches for each node to find its own shortest-path tree is

$$\text{Number of searches} = n + n - 1 + n - 2 + n - 3 + \ldots + 3 + 2 + 1 = n\,(n + 1) / 2$$

The series can be calculated if it is written twice: once in order and once in the reverse order. We then have n items, each of value $(n + 1)$, which results in $n(n + 1)$. However, we need to divide the result by 2. In computer science, this complexity is written as $O(n^2)$ and is referred to as Big-O notation.

P20-6. The following shows the shortest-path tree and the forwarding table for node A.

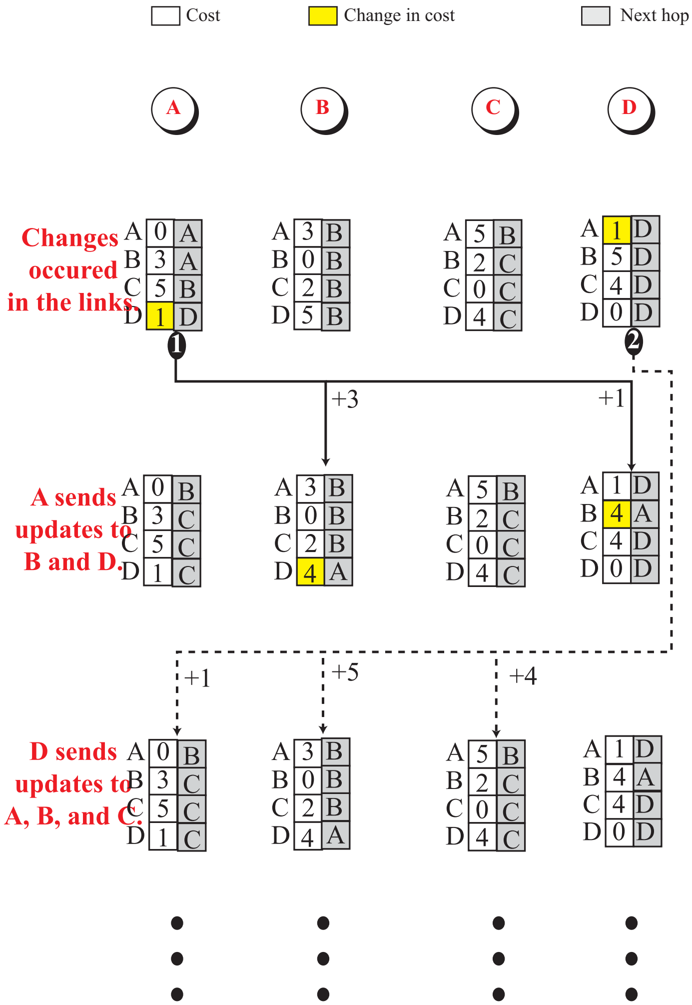




Destination	Cost	Next hop
A	0	—
B	4	—
C	5	—
D	3	—
E	6	D
F	7	B
G	8	B

Forwarding table for node A

P20-7. Two nodes, A and D, see the changes. These two nodes update their vectors immediately. We assume that changes in each round are fired in the order A, B, C, D. The following shows that the internet is actually stable after two rounds of updates, but more updates are fired to assure the system is stable. We have shown only three columns for each forwarding table, but RIP usually uses more than columns. Also note that we have used the yellow color to show the changed in cost field, which triggers updates. The cost and the next hop fields participate in updating.



P20-8. The following shows the advertisement in each case (a triplet defines the destination, cost, and the next hop):

a. From A to B: (A, 0, A), (B, ∞, A), (C, 4, A), (D, ∞, B).

b. From C to D: (A, 4, C), (B, ∞, D), (C, 0, C), (D, ∞, C).

c. From D to B: (A, ∞, B), (B, ∞, D), (C, 6, D), (D, 0, D).

d. From C to A: (A, ∞, C), (B, 8, D), (C, 0, C), (D, 6, C).

P20-9. The forwarding table for node A can be made using the least-cost tree, as shown below:

Forwarding table for node A

Destination	Cost	Next hop
A	0	——
B	2	——
C	7	B
D	3	——
E	6	B
F	8	B
G	9	B

P20-10. Router R9 knows how to reach N13 through its RIP forwarding table. R9 advertises this reachability to R4 using an eBGP session. R4 in turn advertises this reachability to R2 using an iBGP session. R2 then advertises this reachability to R6 using an eBGP session. R6 advertises its reachability to R8. R8 adds an entry to its RIP forwarding table to show that any packet destined for N13 should be forwarded to R6.

P20-11.

a. The *hello message* (type 1) is used by a router to introduce itself to neighboring routers and to introduce already-known neighboring routers to other neighbors.

b. The *data description message* (type 2) is sent in response to a hello message. A router sends its full LSDB to the newly joined router.

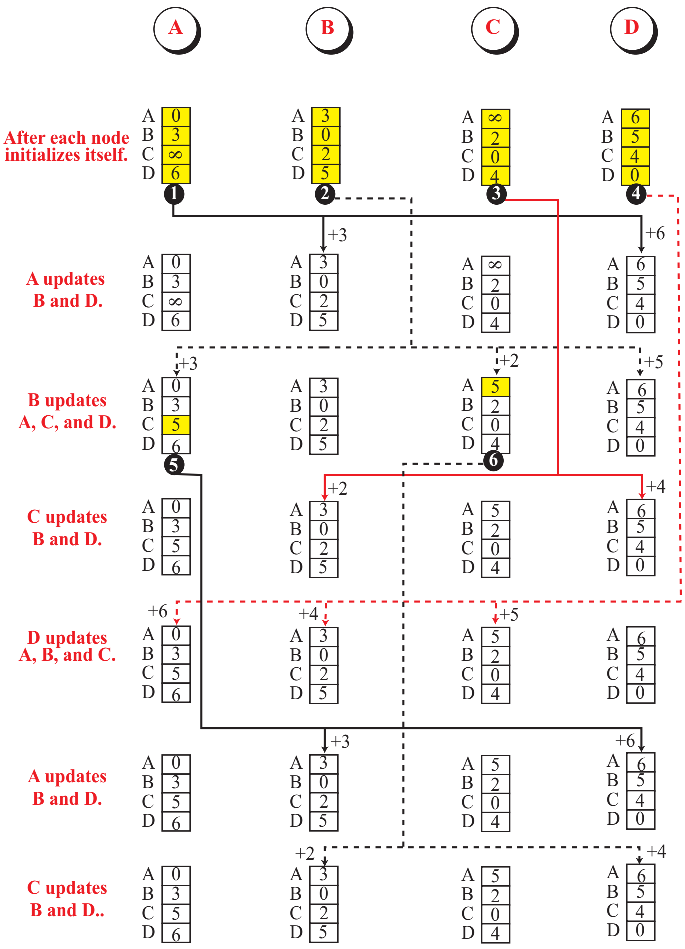
c. The *link-state request message* (type 3) is sent by a router that needs information about a specific LS.

d. The *link-state update message* (type 4) is sent by a router to other routers for building the LSDB. There are five different versions of this message to announce different link states.

e. The *link-state acknowledge message* (type 5) is sent by a router to announce the receiving of a link-state update message. This message is used to provide reliability for the main message used in OSFP.

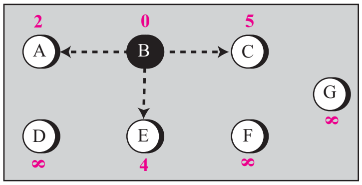
P20-12. The following shows the initialization and two rounds of updates. Although the process is asynchronous, which means that a node can initialize itself and fire updates to its neighbors at the any time, we have assumed the updates takes place in an orderly way (A, B, C, D). After all nodes has sent their updates, a new round starts and those nodes that have seen any change, will fire updates again. The result should be the same using any other order. The process should stop after there is no change in any node.

Note: Colored boxes show the changes in the metric.

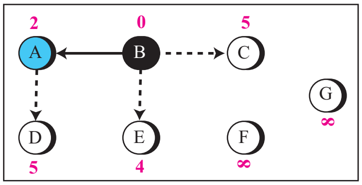


No more updates need to be sent; the system is stable.

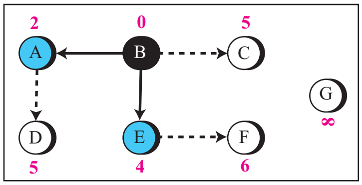
P20-13. The following shows the steps to create the shortest path tree for node B and the forwarding table for this node.



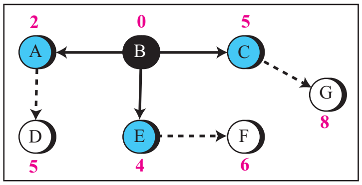
Initialization



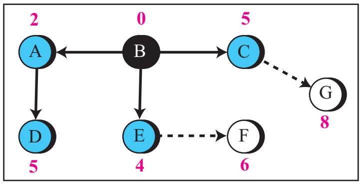
Iteration 1



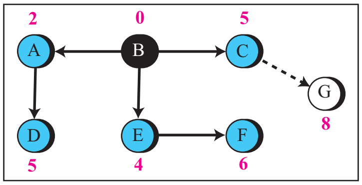
Iteration 2



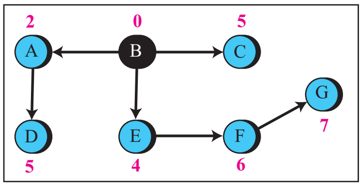
Iteration 3



Iteration 4



Iteration 5

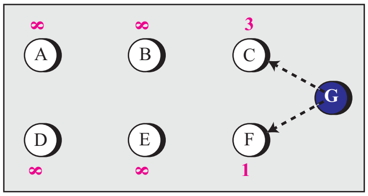


Iteration 6

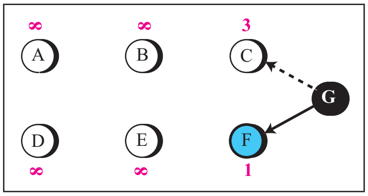
Destination	Cost	Next hop
A	2	———
B	0	———
C	5	———
D	5	A
E	4	———
F	6	E
G	8	E

Forwarding table for node B

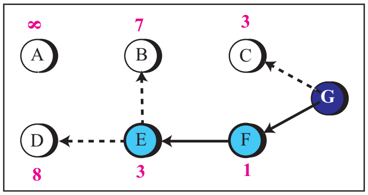
P20-14. The following shows the steps to create the shortest-path tree for node G and the forwarding table for this node.



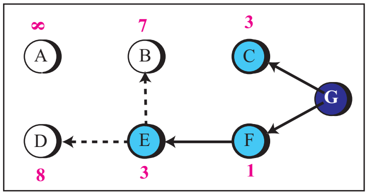
Initialization



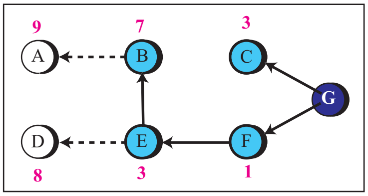
Iteration 1



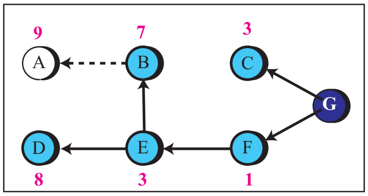
Iteration 2



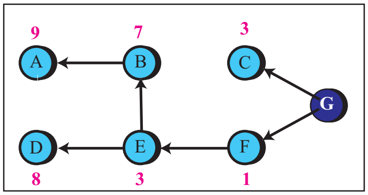
Iteration 3



Iteration 4



Iteration 5



Iteration 6

Destination	Cost	Next hop
A	9	F
B	7	F
C	3	——
D	8	F
E	3	F
F	1	——
G	0	——

Forwarding table for node G

P20-15. The following shows the advertisement in each case (a triplet defines the destination, cost, and the next hop):

a. From A to B: (A, 0, A), (C, 4, A).

b. From C to D: (A, 4, C), (C, 0, C).

c. From D to B: (C, 6, D), (D, 0, D).

d. From C to A: (B, 8, D), (C, 0, C), (D, 6, C).

P20-16. At time t_1, we have one periodic timer, ten expiration timers, and no garbage collection timer. An expiration timer becomes invalid after 180 seconds. This means, at time t_2, we have one periodic timer, nine expiration timers, and one garbage collection timer (for the one which has become invalid).

P20-17. The number of operations in each iteration of the algorithm is n, in which n is the number of nodes in the network. In computer science, this complexity is written as $O(n)$ and is referred to as Big-O notation.

P20-18. We can guess the new routing table because the only way each node can reach node E is via D. The following shows the new network and the forwarding tables. Note that we only add an entry to each forwarding table for nodes A, B, C, and D. The forwarding table for node E is totally new.