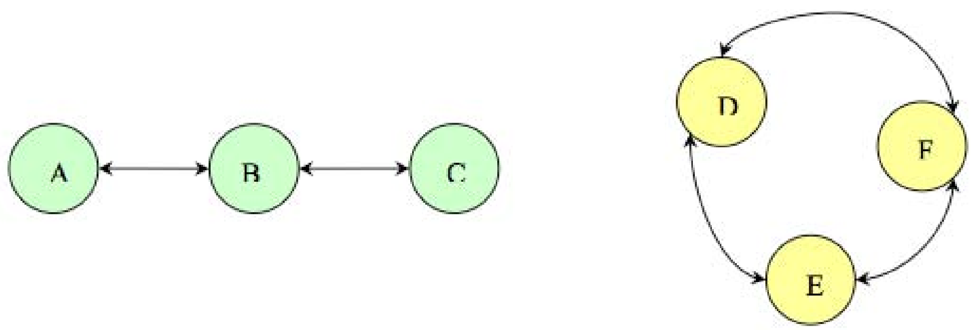
Show All Answers Hide All Answers

6.02 Practice Problems: Routing

IMPORTANT: IN ADDITION TO THESE PROBLEMS, PLEASE SOLVE THE PROBLEMS AT THE END OF CHAPTERS 17 AND 18.

Problem 1. Consider the following networks: network I (containing nodes A, B, C) and network II (containing nodes D, E, F).



A. The Distance Vector Protocol described in class is used in both networks. Assume advertisements are sent every 5 time steps, all links are fully functional and there is no delay in the links. Nodes take zero time to process advertisements once they receive them. The HELLO protocol runs in the background every time step in a way that any changes in link connectivity are reflected in the next DV advertisement. We count time steps from t=0 time steps.

   Please fill in the following table:

Event	Number of time steps
A's routing table has an entry for B	
A's routing table has an entry for C	
D's routing table has an entry for E	
F's routing table has an entry for D	

   Hide Answer

   A's routing table has an entry for B: 5 time steps
   A's routing table has an entry for C: 10 time steps
   D's routing table has an entry for E: 5 time steps
   F's routing table has an entry for D: 5 time steps

B. Now assume the link B-C fails at t = 51 and link D-E fails at t = 71 time steps. Please fill in this table:

Event	Number of time steps
B's *advertisements* reflect that C is unreachable	
A's routing table reflects C is unreachable	
D's routing table reflects a new route for E	

Hide Answer

B's advertisements reflect that C is unreachable: 55 time steps
A's routing table reflects C is unreachable: 55 time steps
D's routing table reflects a new route for E: 75 time steps

Problem 2. Alyssa P. Hacker manages MIT's internal network that runs link-state routing. She wants to experiment with a few possible routing strategies. Of all possible paths available to a particular destination at a node, a routing strategy specifies the path that must be picked to create a routing table entry. Below is the name Alyssa has for each strategy and a brief description of how it works.

MinCost: Every node picks the path that has the smallest sum of link costs along the path. (This is the minimum cost routing you implemented in the lab).

MinHop: Every node picks the path with the smallest number of hops (irrespective of what the cost on the links is).

SecondMinCost: Every node picks the path with the second lowest sum of link costs. That is, every node picks the second best path with respect to path costs.

MinCostSquared: Every node picks the path that has the smallest sum of squares of link costs along the path.

Assume that sufficient information (e.g., costs, delays, bandwidths, and loss probabilities of the various links) is exchanged in the link state advertisements, so that every node has complete information about the entire network and can correctly implement the strategies above. You can also assume that a link's properties don't change, e.g., it doesn't fail.

A. Help Alyssa figure out which of these strategies will work correctly, and which will result in routing with loops. In case of strategies that do result in routing loops, come up with an example network topology with a routing loop to convince Alyssa.

   Hide Answer

   Answer: All except SecondMinCost will work fine.

   To see why SecondMinCost will not work: consider the triangle topology with 3 nodes A, B, D, and equal cost on all the links. The second route at A to D is via B, and the second best route at B to D is via A, resulting in a routing loop.

B. How would you implement MinCostSquared in a distance-vector protocol?

   Hide Answer

   To implement MinCostSquared, your integrate_announcement routine would add the square of the link cost (instead of just the link cost) to any route costs advertised over that link.

Problem 3. Which of the following tasks does a router R in a packet-switched network perform when it gets a packet with destination address D? Indicate True or False for each choice.

A. R looks up D in its routing table to determine the outgoing link.

   Hide Answer

   True.

B. R sends out a HELLO packet or a routing protocol advertisement to its neighbors.

Hide Answer

False.

C. R calculates the minimum-cost route to destination D.

Hide Answer

False.

D. R may discard the packet.

Hide Answer

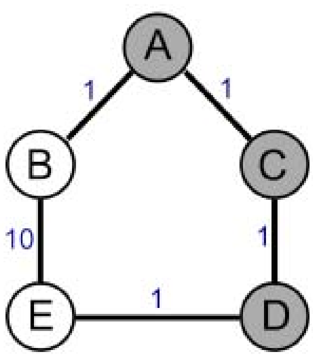
True.

Problem 4. Alice and Bob are responsible for implementing Dijkstra's algorithm at the nodes in a network running a link-state protocol. On her nodes, Alice implements a minimum-cost algorithm. On his nodes, Bob implements a "shortest number of hops" algorithm. Give an example of a network topology with 4 or more nodes in which a routing loop occurs with Alice and Bob's implementations running simultaneously in the same network. Assume that there are no failures.

(Note: A routing loop occurs when a group of k ≥ 1 distinct nodes, n_0 , n_1 , n_2 , …, n_(k-1) have routes such that n_i's next-hop (route) to a destination is n_(i+1 mod k).

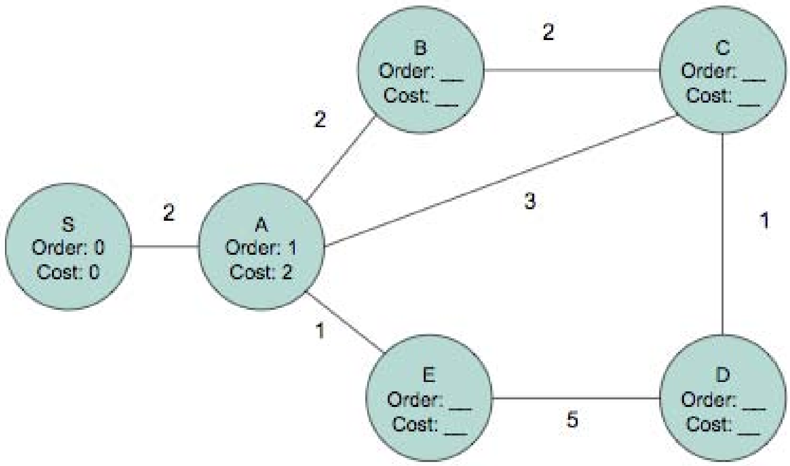
Hide Answer

In the picture below, the grey nodes (A in particular) run Bob's algorithm (shortest number of hops), while the white nodes (B in particular) run Alice's (minimum-cost).



Suppose the destination is E. A will pick B as its next hop because ABE is the shortest path. B will pick A as its next hop because BACDE is the minimum-cost path (cost of 4, compared to 11 for the ABE path). The result is a routing loop ABABAB...

Problem 5. Consider the network shown below. The number near each link is its cost.



We're interested in finding the shortest paths (taking costs into account) from S to every other node in the network. What is the result of running Dijkstra's shortest path algorithm on this network? To answer this question, near each node, list a pair of numbers: The first element of the pair should be the order, or the iteration of the algorithm in which the node is picked. The second element of each pair should be the shortest path cost from S to that node.

To help you get started, we've labeled the first couple of nodes: S has a label (Order: 0, Cost: 0) and A has the label (Order: 1, Cost: 2).

Hide Answer

A: order = 1, cost = 2
E: order = 2, cost = 3
B: order = 3, cost = 4
C: order = 4, cost = 5
D: order = 5, cost = 6

Problem 6. Consider any two graphs(networks) G and G' that are identical except for the costs of the links. Please answer these questions.

A. The cost of link l in graph G is c_l > 0, and the cost of the same link l in Graph G' is `k*c_l`, where k > 0 is a constant and the same scaling relationship holds for all the links. Are the shortest paths between any two nodes in the two graphs identical? Justify your answer.

   Hide Answer

   Yes, they're identical. Scaling all the costs by a constant factor doesn't change their relative size.

B. Now suppose that the cost of a link l in G' is `k*c_l + h`, where k > 0 and h > 0 are constants. Are the shortest paths between any two nodes in the two graphs identical? Justify your answer.

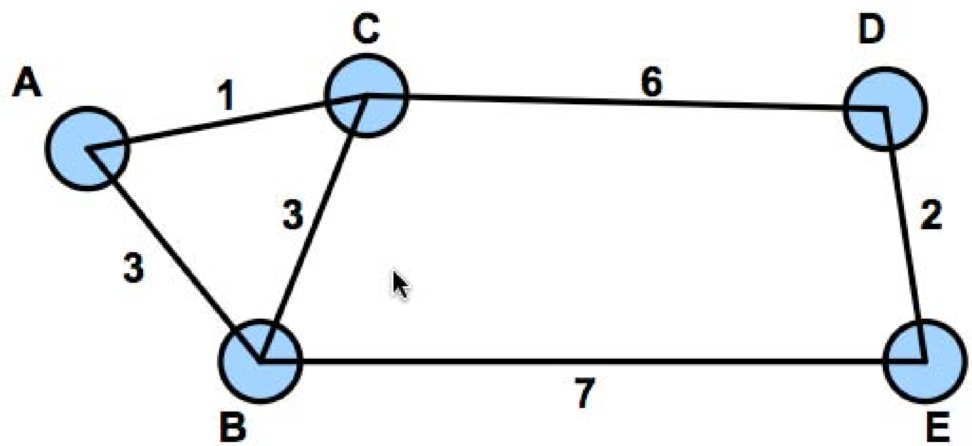
   Hide Answer

   No, they're not necessarily identical. Consider two paths between nodes A and B in graph G. One path takes 3 hops, each of cost 1, for a total cost of 3. The other path takes 1 hop, with a cost of 4. In this case, the shortest path between nodes A and B is the first one.

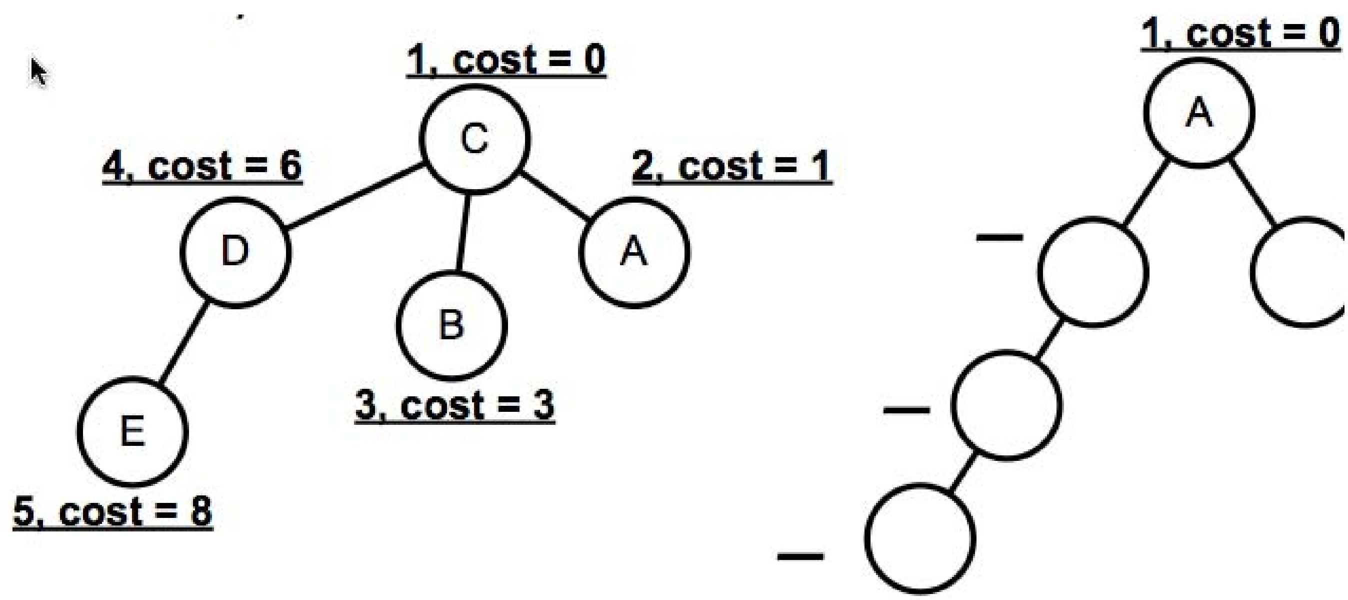
   Consider k=1 and h=1 and compute the costs and shortest paths in G'. Now the 3-hop path has cost 6 and the 1-hop path has cost 5. In G' the shortest path is the second path.

Problem 7. Dijkstra's algorithm

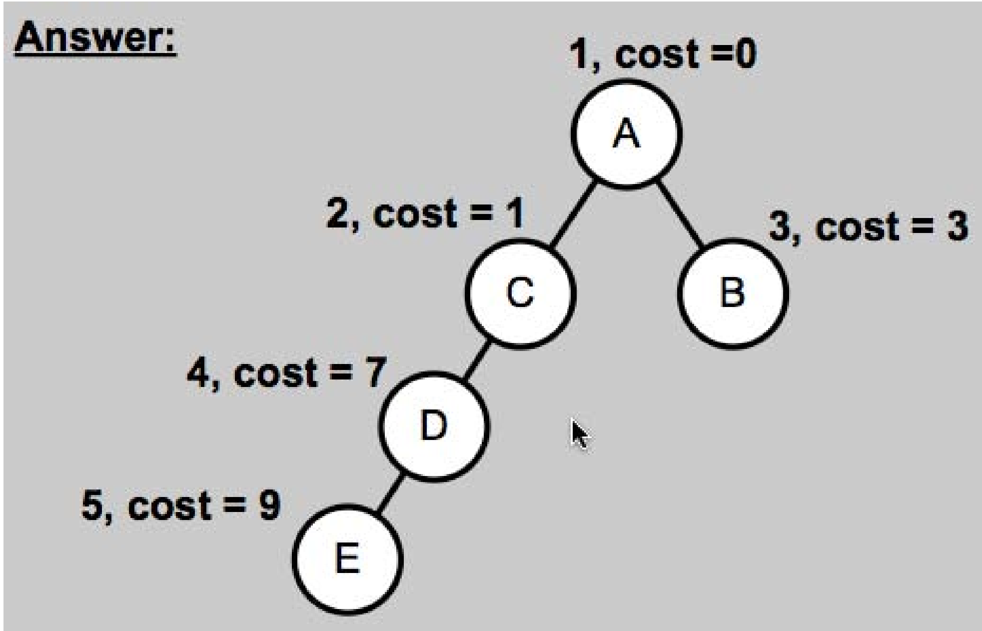
A. For the following network



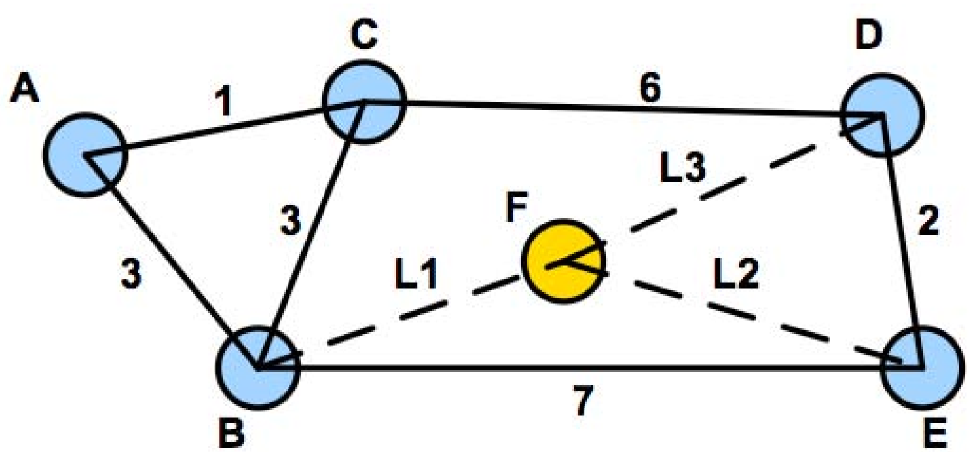
an empty routing tree generated by Dijkstra's algorithm for node A (to every other node) is shown below. Fill in the missing nodes and indicate the order that each node was added and its associated cost. For reference, node C's completed routing tree is shown as well.



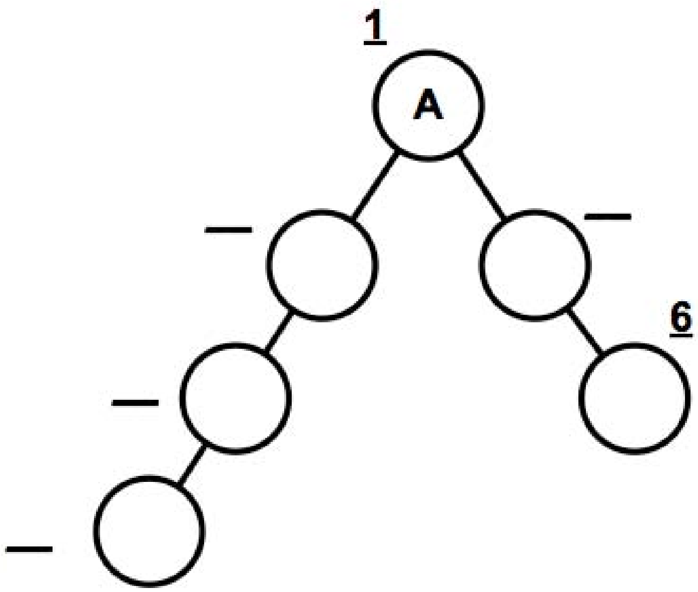
Hide Answer



B. Now assume that node F has been added to the network along with links L1, L2 and L3.



What are the constraints on L1, L2 and L3 such that node A's routing tree must match the topology shown below (regardless of how ties are broken in the algorithm), and it is known that node F is not the last node added when using Dijkstra's algorithm? All costs are positive integers.



Hide Answer

First note that there are several possibilities for the tree, but only one really works as we will show.

Case 1: The left could be ACDF and the right could be ABE.
This case is impossible because ABE is not a min-cost path; ACDE has cost 9, which is less than the cost of ABE (10).

Case 2: The left is ACDE and the right is ABF.
This is not possible because F cannot be the last node added, and the last node added is on the bottom right.

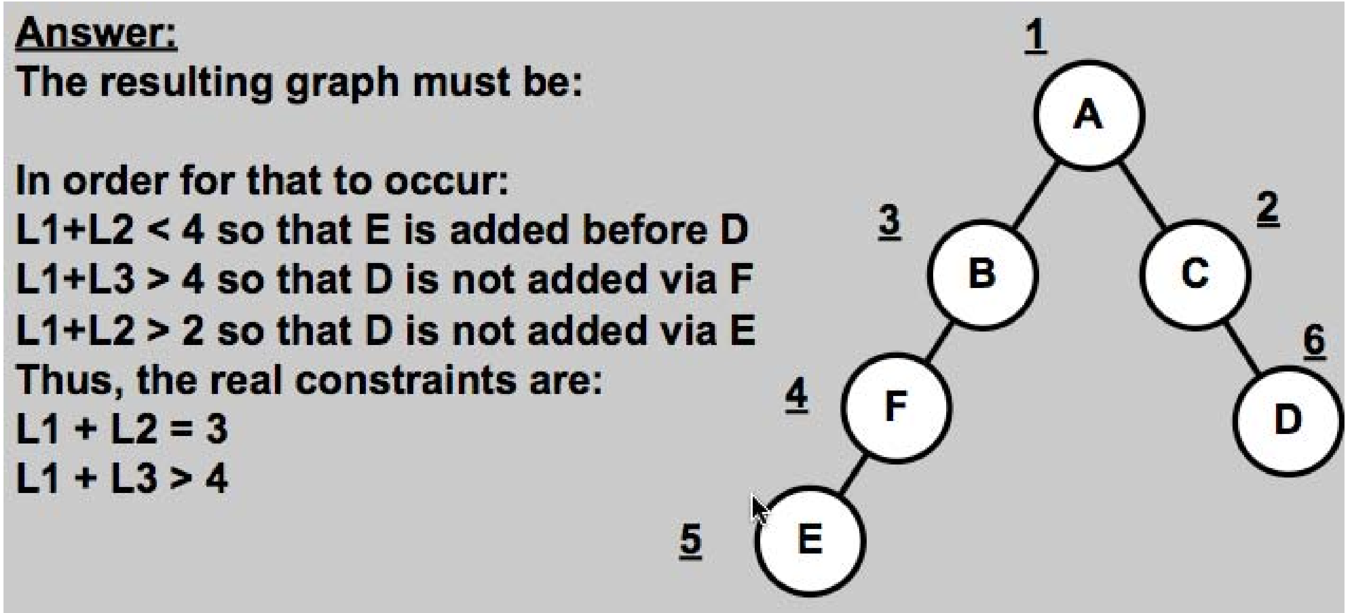
Case 3 is the only remaining possibility: The left is ABFE and the right is ACD. We need to analyze this case to come up with the required constraints.

1) Node E is added before D, so cost(ABFE) < cost (ACD), so $3 + L1 + L2 < 1 + 6 \implies L1 + L2 < 4$.

2) Node D is not added via F, so cost(ABFD) > cost(ACD), so $3 + L1 + L3 > 1 + 6 \implies L1 + L3 > 4$

3) Node D is not added via E, so cost(ABFED) > cost(ACD), so $3 + L1 + L2 + 2 > 1 + 6 \implies L1 + L2 > 2$.

1) and 3) imply $L1 + L2 = 3$. So the two constraints are: $L1 + L2 = 3$ AND $L1 + L3 > 4$.



Problem 8. Under some conditions, a distance vector protocol finding minimum cost paths suffers from the "count-to-infinity" problem. Indicate True or False for each choice.

A. The count-to-infinity problem may arise in a distance vector protocol when the network gets disconnected.

Hide Answer

True.

B. The count-to-infinity problem may arise in a distance vector protocol even when the network never gets disconnected.

Hide Answer

False.

C. The "path vector" enhancement to a distance vector protocol always enables the protocol to converge without counting to infinity.

Hide Answer

True.

Problem 9. Ben Bitdiddle has set up a multi-hop wireless network in which he would like to find paths with high probability of packet delivery between any two nodes. His network runs a distance vector protocol similar to what you developed in the pset. In Ben's distance vector (BDV) protocol, each node maintains a metric to every destination that it knows about in the network. The metric is the nodeâ€™s estimate of the packet success probability along the path between the node and the destination.

The packet success proba bility along a link or path is defined as 1 minus the packet loss probability along the corresponding link or path. Each node uses the periodic HELLO messages sent by each of its neighbors to estimate the packet loss probability of the link from each neighbor. You may assume that the link loss probabilities are symmetric; i.e., the loss probability of the link from node A to node B is the same as from B to A. Each link L maintains its loss probability in the variable `L.lossprob` and `0 < L.lossprob < 1`.

A. The key pieces of the Python code for each node's `integrate()` function in BDV is given below. It has three missing blanks. Please fill them in so that the protocol will eventually converge without routing loops to the correct metric at each node. The variables are the same as in the pset: `self.routes` is the dictionary of routing entries (mapping destinations to links), `self.getlink(fromnode)` returns the link connecting the node self to the node fromnode, and the `integrate` procedure runs whenever the node receives an advertisement (`adv`) from node fromnode. As in the pset, `adv` is a list of (destination, metric) tuples. In the code below, `self.metric` is a dictionary storing the nodeâ€™s current estimate of the routing metric (i.e., the packet success probability) for each known destination. Please fill in the missing code.

```
# Process an advertisement from a neighboring node in BDV

def integrate(self, fromnode, adv):
  L = self.getlink(fromnode)
  for (dest, metric) in adv:
    my_metric = __________________________________
    if (dest not in self.routes
        or self.metric[dest] _____ my_metric
        or ___________________________________________):
      self.routes[dest] = L
      self.metric[dest] = my_metric

  # rest of integrate() not shown
```

Hide Answer

First blank: `(1 - L.lossprob)*metric`

Second blank: < (<= is also fine since we said that lossprob is strictly > 0)

Third blank: `self.routes[dest] == L`

Ben wants to try out a link-state protocol now. During the flooding step, each node sends out a link-state advertisement comprising its address, an incrementing sequence number, and a list of tuples of the form `(neighbor,lossprob)`, where the lossprob is the estimated loss probability to the `neighbor`.

B. Why does the link-state advertisement include a sequence number?

Hide Answer

To enable a node to determine whether the advertisement is new or not; only new information should be integrated into the routing table. (This information is also used to decide whether to rebroadcast the advertisement, since we want to rebroadcast an advertisement only once per link.)

Ben would like to reuse, without modification, his implementation of Dijkstra's shortest paths algorithm from the pset, which takes a map in which the links have non-negative costs and produces a path that minimizes the sum of the costs of the links on the path to each destination.

C. Ben has to transform the lossprob information from the LSA to produce link costs so that he can use his Dijkstra implementation without any changes. Which of these transformations will accomplish this goal? Choose the BEST answer.

   a. Use `lossprob` as the link cost.
   b. Use `-1/log(1-lossprob)` as the link cost.
   c. Use `log(1/(1-lossprob))` as the link cost.
   d. Use `log(1 - lossprob)` as the link cost.

Hide Answer

The correct choice is C. The reason is that maximizing the product of link success probabilities is the same as maximizing the sum of the logs of these probabilities, and that is the same as minimizing the sum of the logs of the reciprocals of these probabilities. A is correct only when lossprob << 1, which isn't always the case. D is plausible because of the log term, but is negative, so Dijkstra's doesn't work on a network with negative costs with negative-cost loops. B is plausible for the same reason, but is a decreasing function of lossprob, and so can't be right.

Problem 10. We studied a few principles for designing networks in 6.02.

A. State one significant difference between a circuit-switched and a packet-switched network.

Hide Answer

In a packet-switched network, packets carry information in the header that tells the switches about the destination. Circuit-switched networks donâ€™t carry any destination information in the data frames.

The abstraction provided by a circuit-switched network is that of a dedicated link of a fixed rate; a packet-switched network provides no such guarantee to the communicating end points.

B. Why does topological addressing enable large networks to be built?

Hide Answer

It reduces the size of the routing tables and the amount of information that must be exchanged in the routing protocol.

C. Give one difference between what a switch does in a packet-switched network and a circuit-switched network.

Hide Answer

Switches in a circuit-switched network participate in a connection set-up/teardown protocol, but not in a packet-switched network.

Switches in a packet-switched network look-up the destination address of a packet during forwarding, but not in a circuit-switched network.

Problem 11. Eager B. Eaver implements distance vector routing in his network in which the links all have arbitrary positive costs. In addition, there are at least two paths between any two nodes in the network. One node, u, has an erroneous implementation of the integration step: it takes the advertised costs from each neighbor and picks the route corresponding to the minimum advertised cost to each destination as its route to that destination, *without adding the link cost to the neighbor*. It breaks any ties arbitrarily. All the other nodes are implemented correctly.

Let's use the term "correct route" to mean the route that corresponds to the minimum-cost path. Which of the following statements are true of Eager's network?

a. Only u may have incorrect routes to any other node.
b. Only u and u's neighbors may have incorrect routes to any other node.
c. In some topologies, all nodes may have correct routes.
d. Even if no HELLO or advertisements packets are lost and no link or node failures occur, a routing loop may occur.

Hide Answer

A and B are false, C is true, and D is false.

A is false because u could propagate an incorrect cost to its neighbors causing the neighbor to have an incorrect route. In fact, u's neighbors could do the same.

C is correct; a simple example is where the network is a tree, where there is exactly one path between any two nodes.

D is false; no routing loops can occur under the stated condition. We can reason by contradiction. Consider the shortest path from any node s to any other node t running the flawed routing protocol. If the path does not traverse u, no node on that path can have a loop because distance vector routing without any packet loss or failures is loop-free. Now consider the nodes for which the computed paths go through u; all these nodes are correctly implemented except for u, which means the paths between u and each of them is loop-free. Moreover, the path to u is itself loop-free because u picks one of its neighbors with smaller cost, and there is no possibility of a loop.

Problem 12. Consider a network running the link-state routing protocol as described in lecture and on the pset. How many copies of any given LSA are received by a given node in the network?

Hide Answer

When there are no packet losses, it is equal to the number of neighbors of the node.

Problem 13. In implementing Dijkstra's algorithm in the link-state routing protocol at node u, Louis Reasoner first sets the route for each directly connected node v to be the link connecting u to v. Louis then implements the rest of the algorithm correctly, aiming to produce minimum-cost routes, but does not change the routes to the directly connected nodes. In this network, u has at least two directly connected nodes, and there is more than one path between any two nodes. Assume that all link costs are non-negative. Which of the following statements is true of u's routing table?

A. There are topologies and link costs where the *majority* of the routes to other nodes will be *incorrect*.

   Hide Answer

   True. For example, all the neighbors but one could have very high cost, and all the other links have low cost, so all the routes could in fact be just one link.

B. There are topologies and link costs where *no* routing table entry (other than from u to itself) will be *correct*.

   Hide Answer

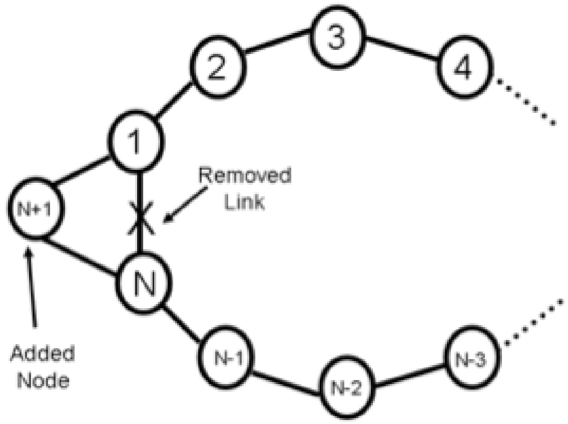
   False. The lowest-cost neighbor's route will be the direct link, of course!

C. There are topologies and link costs where *all* routing table entry (other than from u to itself) will be *correct*.

   Hide Answer

   True. A trivial example is when all the links have equal cost.

Problem 14. A network with N nodes and N bidirectional links is connected in a ring, and N is an even number.



The network runs a distance-vector protocol in which the advertisement step at each node runs when the local time is `T*i` seconds and the integration step runs when the local time is `T*i + T/2` seconds, (i=1,2,...). Each advertisement takes time δ to reach a neighbor. Each node has a separate clock and time is not synchronized between the different nodes.

Suppose that at some time t after the routing has converged, node N + 1 is inserted into the ring, as shown in the figure above. Assume that there are no other changes in the network topology and no packet losses. Also assume that nodes 1 and N update their routing tables at time t to include node N + 1, and then rely on their next scheduled advertisements to propagate this new information.

A. What is the minimum time before every node in the network has a route to node N + 1?

Hide Answer

The key insight to observe is that each introduces a delay of at least T/2 because it takes that long between the integration and advertisement steps. Given this fact, the answer would be

```
(N/2 - 1)*(T/2 + δ)
```

However, there is a small "fence-post error" in this argument. As stated in the problem, the nodes labeled 1 and N update their routing tables at time t to include node N + 1. In the best case, these two nodes could both immediately send out advertisements, and nodes 2 and N-1 could run their integration steps immediately after receiving these advertisements. Because of that, the delay of T/2 only starts applying to the other nodes in the network. Hence, the answer is

```
(N/2 - 2)*T/2 + (N/2 - 1)*δ
```

B. What is the maximum time before every node in the network has a route to node N + 1?

Hide Answer

The key insight is that it takes in the worst case T + T/2 seconds per hop because each node may get the information about the new node *just after* it completes the previous integration step. So it has to wait T for the next integration, and then another T/2 to advertise. The correct answer is

```
(N/2 -1)*(3T/2 + δ)
```

Problem 15. Louis Reasoner implements the link-state routing protocol discussed in 6.02 on a best-effort network with a non-zero packet loss rate. In an attempt to save bandwidth, instead of sending link-state advertisements periodically, each node sends an advertisement *only if* one of its links fails or when the cost of one of its links changes. The rest of the protocol remains unchanged. Will Louis' implementation always converge to produce correct routing tables on all the nodes?

Hide Answer

No, because the LSA could be lost on all of the links connected to some one node (or more than one node), causing that node to not necessarily have correct routes. In fact this protocol doesn't converge even if packets are not lost. Consider a network where the failure of one link disconnects the network into two connected components, each with multiple nodes and links. Suppose the cost of one or more of the links in some component changes. When the network heals because the failed link recovers, the previously discon- nected component will not have the correct link costs for one or more links in the other component. However, its routes will still be correct because all paths to the other component go via the failed link. However, if we were to add another link between the two components at this time, the routing would never converge correctly.

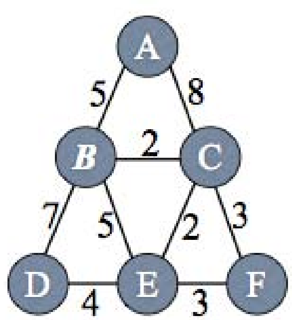
Problem 16. Consider a network implementing minimum-cost routing using the distance-vector protocol. A node, S, has k neighbors, numbered 1 through k, with link cost c_i to neighbor i (all links have symmetric costs). Initially, S has no route for destination D. Then, S hears advertisements for D from each neighbor, with neighbor i advertising a cost of p_i. The node integrates these k advertisements. What is the cost for destination D in S's routing table after the integration?

Hide Answer

This question asks for the update rule in the Bellman-Ford integration step. The cost in S's routing table for D should be set to

$\min_i\{c\_i + p\_i\}$

Problem 17. Consider the network shown in the picture below. Each node implements Dijkstra's shortest path algorithm using the link costs shown in the picture.



A. Initially, node B's routing table contains only one entry, for itself. When B runs Dijkstra's algorithm, in what order are nodes added to the routing table? List all possible answers.

   Hide Answer

   B,C,E,A,F,D and B,C,E,F,A,D.

B. Now suppose the link cost for one of the links changes but all costs remain non-negative. For each change in link cost listed below, state whether it is possible for the route at node B (i.e., the link used by B) for any destination to change, and if so, name the destination(s) whose routes may change.

   a. The cost of link(A, C) increases.

      Hide Answer

      No effect. The edge AC is not in any shortest path.

b. The cost of link(A, C) decreases.

Hide Answer

Can affect route to A. If cost_AC ≤ 3, then we can start using this edge to go to A instead of the edge BA.

c. The cost of link(B, C) increases.

Hide Answer

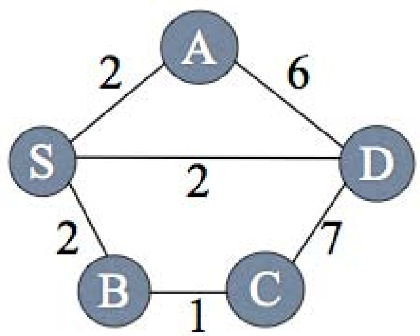
Can affect route to C,F,E. If cost_BC ≥ 7, then we can use BE-EC to go to C instead of BC. If cost BC > 5, then we can use BE-EF to go to F. If cost_BC ≥ 3, can use BE to go to B.

d. The cost of link(B, C) decreases.

Hide Answer

Can affect route to D. If cost_BC ≤ 1, then we can use BC-CE-ED to go to D instead of BD.

Problem 18. Alyssa P. Hacker implements the 6.02 distance-vector protocol on the network shown below. Each node has its own local clock, which may not be synchronized with any other node's clock. Each node sends its distance-vector advertisement every 100 seconds. When a node receives an advertisement, it immediately integrates it. The time to send a message on a link and to integrate advertisements is negligible. No advertisements are lost. There is no HELLO protocol in this network.



A. At time 0, all the nodes except D are up and running. At time 10 seconds, node D turns on and immediately sends a route advertisement for itself to all its neighbors. What is the minimum time at which each of the other nodes is guaranteed to have a correct routing table entry corresponding to a minimum-cost path to reach D? Justify your answers.

Hide Answer

Node S: 10 seconds.
Node A: 110 seconds.
Node B: 110 seconds.
Node C: 210 seconds.

At time t = 10, D advertises to S, A, and C. They integrate this advertisement into their routing tables, so that cost(S,D) = 2, cost(A,D) = 2, cost(C,D) = 7. Note that only Sss route is correct. In the worst case, we wait 100s for the next round of advertisements. So at time t = 110, S, A, and C all advertise about D, and everyone integrates. Now cost(A, D) = 4 (via S), cost(B, D) = 4 (via S), and cost(C, D) = 7 still. A and B's routes are correct; C's is not. Finally, after 100 more seconds, another round of advertisements is sent. In particular, C hears about B's route to D, and updates cost(C, D) = 5 (via B).

B. If every node sends packets to destination D, and to no other destination, which link would carry the most traffic?

Hide Answer

S→D. Every node's best route to D is via S.

Alyssa is unhappy that one of the links in the network carries a large amount of traffic when all the nodes are sending packets to D. She decides to overcome this limitation with Alyssa's Vector Protocol (AVP). In AVP, *S lies*, advertising a "path cost" for destination D that is different from the sum of the link costs along the path used to reach D. All the other nodes implement the standard distance-vector protocol, not AVP.

C. What is the smallest numerical value of the cost that S should advertise for D along each of its links, to *guarantee* that only its own traffic for D uses its direct link to D? Assume that all advertised costs are integers; if two path costs are equal, one can't be sure which path will be taken.

Hide Answer

7. S needs to advertise a high enough cost such that everyone's path to D via S will no longer be the best path. In particular, since B's cost to D without going through S is the highest (8), S must advertise a cost so that linkcost(B, S) + advertisedcost(S, D) > 8. Hence, S advertises a cost of 7.