ECE 438/638: Computer Communications **Spring 2013**
Problem Set 8 Solutions

All problems carry equal weight.

1. **Distance-vector routing**.
   (a) Node A's first update to its routing table just contains the information from node B, plus the cost of the link to B. Notice that the costs for a link can be different in different directions.

Routing table for A		
Destination	Next	Distance
A	A	0
B	B	7
C	B	14
D	B	15
E	B	12
F	B	14

   (b) Node A's updates the entries for which the cost is less through node B.

Routing table for A		
Destination	Next	Distance
A	A	0
B	B	7
C	C	9
D	B	15
E	C	11
F	C	12

2. **Distance-vector routing**.
   (a) The distance vectors that C sends depend on the destination for the message. Here are the two messages:

C to A	
Dest	cost
A	∞
B	1
D	11
E	5
F	4

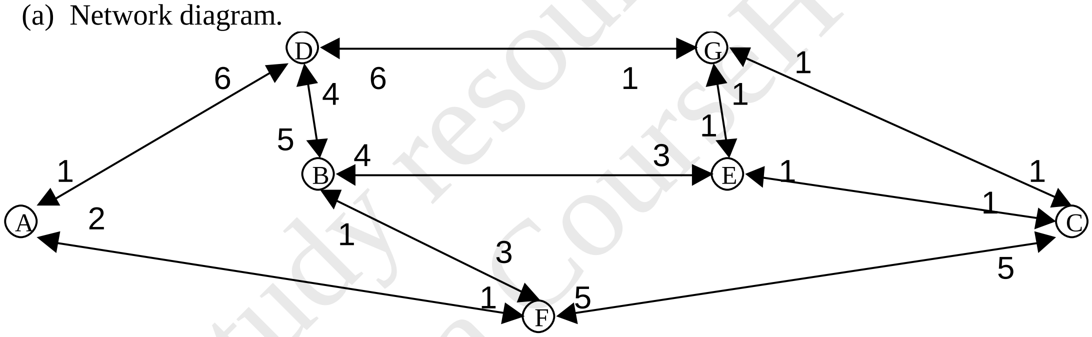
C to B	
Dest	cost
A	2
B	∞
D	∞
E	∞
F	4

   (b) After this one round of updates the loop is not fixed. For destination D, A will update its cost to 13. But B does not change its information to destination D, and B still forwards packets to destination D to A as the next hop. We see that the split horizon with poison reverse does not help with loops of 3 or more hops, and the count-to-infinity problem still occurs. When node B collects other distance vectors from its other neighbors, it will eventually correct its route to D, but this may take a number of iterations.

3. **Link-state routing.** The entries in the table have the form: (Destination, cost, next hop). In the even steps in the table the tentative list is updated. In the odd steps in the table the lowest cost entry from the tentative list is moved to the confirmed list.

D	Confirmed	Tentative
1.	(A,0,-)	
2.	(A,0,-)	**(B,2,B) (C,8,C)**
3.	(A,0,-) **(B,2,B)**	(C,8,C)
4.	(A,0,-) (B,2,B)	(C, **7**, B) **(D,4,D) (E, 5, B) (G, 9, B)**
5.	(A,0,-) (B,2,B) **(D, 4, B)**	(C, 7, B) (E, 5, B) (G, 9, B)
6.	(A,0,-) (B,2,B) (D, 4, B)	(C,7,B) (E,5,B) (G, **8**, B)
7.	(A,0,-) (B,2,B) (D, 4, B) **(E,5,B)**	(C, 7, B) (G, 8, B)
8.	(A,0,-) (B,2,B) (D, 4, B) (E,5,B)	(C, **6**, B) **(F, 12, B)** (G, 8, B)
9.	previous + **(C, 6, B)**	(F, 12, B) (G, 8, B)
10.	Previous	(F, **11**, B) (G, 8, B)
11.	Previous + **(G, 8, B)**	(F, 11, B)
12.	Previous	(F, 11, B)
13.	Previous + **(F, 11, B)**	

4. **Link-state routing**. Part (a)
   (a) Network diagram.



   (b) Path from A to G: A – F – B – E – G (cost = 6). Either calculate by inspection or by Dijkstra's algorithm.
   (c) Path from A to G after change at B: A – F – B – D – G. Note that A and F have no new information so they calculates the same path as found in part (b). That is, A determines that F is the next hop, and F determines that B is the next hop. However, B has new information in its link-state packet (LSP) database. Before the change, B calculated the shortest path to G as B – E – G with a cost of 4. However, now that B has a link cost of 1 for the link B to D, B's new calculation of the shortest path to G is B – D – G with a cost of 2. So B forwards the packet (with destination G) to D rather than E.
   (d) Node D has a new cost for the link D to G (cost = 8). This information is sent to nodes A, F, and G, so nodes A, F, G, and D run Dijkstra's algorithm to calculation new forwarding tables. Notice that the routes calculated at A and F (for destination G) do not change. Also, B has not changed its forwarding table from part (c). However, node D calculates a new route to G. By inspection it is easy to see that the new route to G is D – A – F – B – E – G with a cost of 7. So, D inserts into its forwarding table that all packets for destination G should use node A as the next hop. The result is that packets from node

A to destination G will follow the path A – F – B – D – A, thus traveling in a loop. The only way these packets are removed from the network is when the TTL field in the IP packet header reaches zero (when the packet will be dropped). As can be seen in this problem, it is critical that the flooding protocol used to distribute LSP's must be reliable. Furthermore, if there are even significant delays in flooding the LSP's (or running Dijkstra's algorithm after updating the LSP database), there can be (temporary) loops in the network.

5. Distance-Vector Routing

First, consider the problem in which the cost increases on the link between A and B from 1 to 10

Event number	Action	Response by neighbor
1.	link cost increase from 1 to 10	Node B had these DVs for destination D *before* change Via A (dest=D, cost = 1) Via C (dest=D, cost = 3) *After* increase in link cost: Node B's best route to D: next=C, cost =4 (cost increases from 2 to 4) Note that loop has formed between B and C
2.	Node B generates triggered update with DV (dest=D, cost=4)	Node C had these DVs for D *before* receiving triggered update Via A (dest=D, cost = 1) Via B (dest=D, cost = 2) *After* change to C's DV table: Via B (dest=D, cost = 4) Node C's best route to D: next=B, cost=5 (cost increase from 3 to 5)
3	Node C generates triggered update with DV (dest=D, cost=5)	Node B had these DVs for D *before* receiving triggered update Via A (dest=D, cost = 1) Via C (dest=D, cost = 3) *After* change to B's DV table: Via C (dest=D, cost = 5). Node B's best route to D: next=C, cost =6 (cost increase from 4 to 6)
4	Node B generates triggered update with DV (dest=D, cost=6)	Node C had these DVs for D *before* receiving triggered update Via A (dest=D, cost = 1) Via B (dest=D, cost = 4) *After* change to C's DV table: Via B (dest=D, cost = 6) Node C's best route to D: next=B, cost=7 (cost increase from 5 to 7)
5	C generates DV (dest=D, cost=7)	B's DV before A: (D, 1) and C: (D, 5) (note difference from step 3) *After* change C: (D, 7) B's best route to D: next C, cost 8 (inc 6 to 8)
6	B generates DV (dest=D, cost=8)	C's DV before A: (D, 1) and B: (D, 6) *After* change B: (D, 8) C's best route to D: next A, cost 8 (inc 7 to 8) Note that the loop is finally broken.
7	C generates DV (dest-D, cost=8)	B's DV before A: (D, 1) and C: (D, 7) *After* change C: (D, 8) B's best route to D: next C, cost 9 (inc 8 to 9)
8	B generates DV	C's DV before

	(dest D, cost=9)	A: (D, 1) and B: (D, 8) *After* change B: (D, 9) However, C's route does not change so no DV is generated.

Split horizon with poisoned reverse. The same scenario starts from the beginning but now with the new routing protocol. In this scenario the loop between nodes B and C is avoided.

Event number	Action	Response by neighbor
1.	link cost increase from 1 to 10	Node B had these DVs for destination D *before* change Via A (dest=D, cost = 1) Via C (dest=D, cost = infinity ←*Hint: what goes here?* *After* increase in link cost: Node B's best route to D: next= A, cost = 11 The key idea is that node C has B as next hop in its route to D (before any changes). So when C reports a distance vector to B it "poisons" the entry for dest D because B is the next hop. So, C reports the DV (dest=D, cost=infinity) to node B. Because of this, node B has to stick with the route through A, and thus the loop is avoided.
2 (notice C changes route to use A as next hop)	B generates DV (dest=D, cost=11)	C's DV before change A: (D, 1) and B: (D, 2) *After* change B: (D, 11) C's best route to D: next=A, cost=8 (increase in cost from 3 to 8 and also a new next hop)
3 (notice B changes route to use C as next hop)	C generates DV (dest=D, cost=8)	B's DV before A: (D, 1) and C: (D, infinity) *After* change C: (D, 8) (no longer poisoned because C has new route through A not B) B's best route to D: next C, cost = 9 (decrease in cost from 11 to 9 and new next hop to C)
4	B generates DV (dest D, infinity) for nbr C by poisoned reverse	C's DV before A: (D, 1) and B: (D, 11) *After* change B: (D, infinity) C's best route to D is not changed so no triggered update is generated

6. **Mobile IP**.

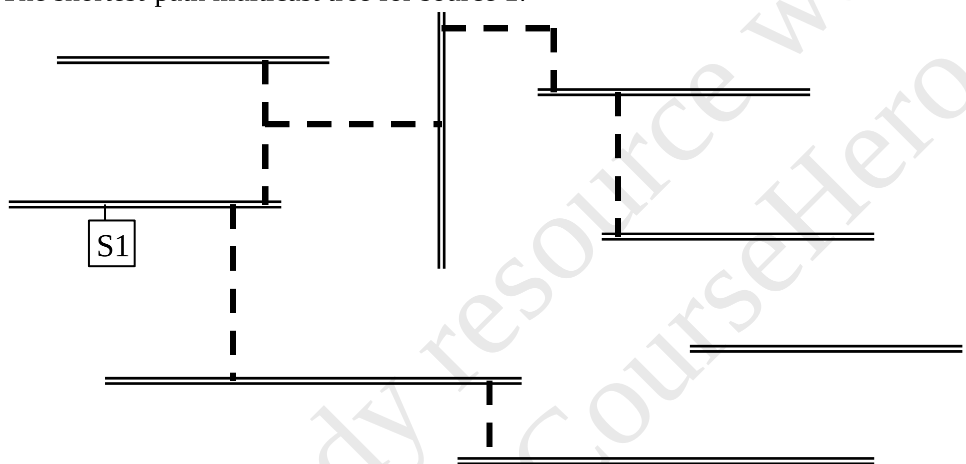
(a) The key idea here is that proxy-ARP is used by the access point. When R1 has an IP packet with an IP address assigned to one of the mobiles, it will broadcast and ARP query on LAN B for that IP address. The access point will respond with the ARP reply but use its own MAC address. R1 will be fooled into thinking it is forwarding the frame directly to the mobile, when in fact the access point will get the frame and then use the wireless link to forward it to the mobile.

(b) The key idea for this part is the use of a tunnel. Once the mobile registers with the foreign agent, the foreign agent sends a care-of address to the home agent. When a packet is received for the mobile, the home agent wraps the packet inside an IP packet for the foreign agent, and all nodes forward the IP packet to the foreign agent. The foreign agent un-wraps the packet and recovers the original packet for the mobile. See also the textbook for description of the triangle routing problem and how it can also be solved with a tunnel.

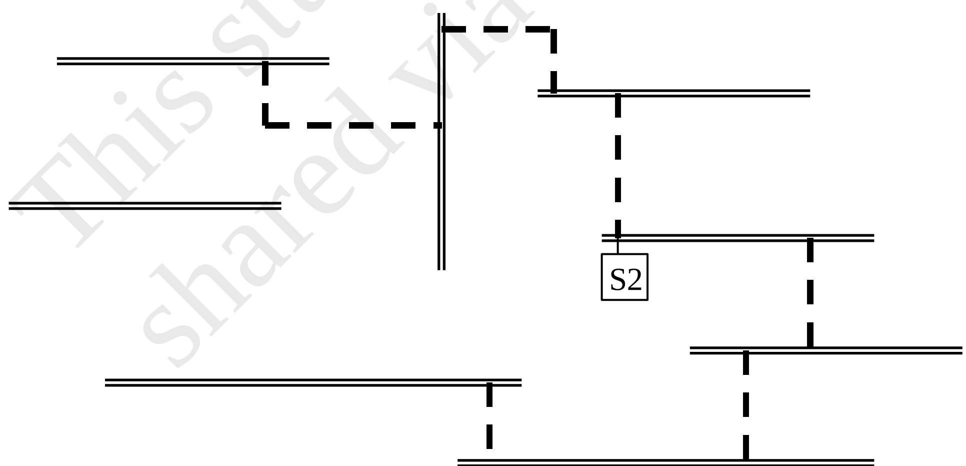
Students enrolled in 638 will be responsible for material in Sections 4.2 and 4.3 and should solve the next two problems.

7. **Multicast**. Chapter 4, number 16

The shortest-path multicast tree for source 1:



The shortest-path multicast tree for source 2:



8. **Multicast**.

(a) Heard on A, B, D, G. Router R2 receives a packet from R4 with source address R4 (or α) and destination address Z. R2 computes (or has precomputed) a tree of shortest paths from α (or R4) to the nodes of Z using Dijkstra's algorithm. It then sees that its only responsibility is to forward the packet to β via LAN D.

(b) Heard on all LAN's. Note parent routers ensure that only one copy of the packet is broadcast on each LAN.

(c) Heard on A, B, D, G (as in part (a)). Router R2 receives a packet from R4 with the source address R4 and the destination address Z. It relays it only on LAN D (addressed to β). R2 does not forward the packet on C because it has cached information that tells it no multicast group members are in that direction. (or possibly LAN C if you used a different approach for assigning parent routers. However, then you needed to specify how the packet gets from host α to R3)

P&D imply that for reverse-path multicast, the multicast tree is calculated from the host instead of from a router. This approach could be implemented but it makes the calculations more difficult as the initial routers have a more complicated decision. I stated in the problem that the host sends the packet as a unicast transmission to the router, so that the router serves as the root of the tree. If you assigned parent routers that reflected distance to the host α instead of router R4, the parent for LAN C is R3. However, the multicast is more complicated because host α should not use a unicast transmission to R4, but instead should make a multicast transmission so that both routers R3 and R4 pick up the packet. One problem with this is that the host α needs to know what kind of multicast algorithm the routers implement. If instead host α simply defaults by sending the packet to its default router, the routers can coordinate among themselves on the multicast algorithm they execute and the hosts need not be updated as new multicast algorithms are deployed.

9. **Extra credit**. Computer program for minimum distance calculation.

The least cost, 2768, is achieved by four paths:

```
0   4  10  19  30  44  60  78  99
0   4  10  19  30  43  59  78  99
0   4  10  18  29  43  59  78  99
0   3   9  18  29  43  59  78  99.
```

The particular path that is found depends on how you break ties, and finding any one path is all that is required.