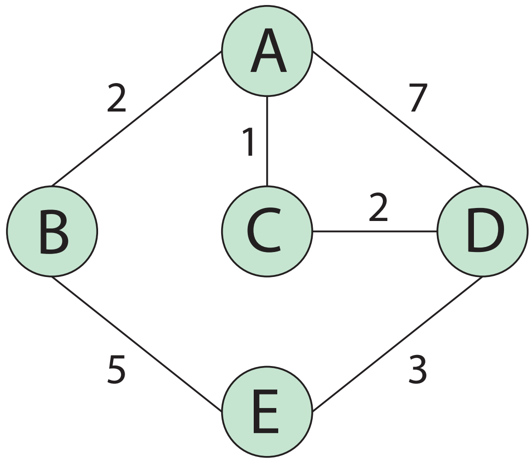
CS 168 Fall 2016 Section 4 – Distance Vector Routing



Problem 1: Distance-Vector Routing

The nodes in the above network communicate with each other using Distance-Vector routing. Below are the initial routing tables for each node, and a table showing the costs for each of their neighboring links.
In the routing tables, each row represents a neighbor and a column represents a destination. Each cell entry is of the format (`shortest known distance to dest, next hop`).

Node A

Nbr	*Cost*
A	0
B	2
C	1
D	7

To / From	A	B	C	D
A	0, A	2, B	1, C	7, D
B	-	0	-	-
C	-	-	0	-
D	-	-	-	0

Node B

Nbr	*Cost*
A	2
B	0
E	5

To / From	A	B	E
A	0	-	-
B	2, A	0, B	5,E
E	-	-	0

Node C

Nbr	*Cost*
A	1
C	0
D	2

To / From	A	C	D
A	0	-	-
C	1, A	0, C	2, D
D	-	-	0

Node D

Nbr	*Cost*
A	7
C	2
D	0
E	3

To / From	A	C	D	E
A	0	-	-	-
C	-	0	-	-
D	7, A	2, C	0, D	3, E
E	-	-	-	0

Node E

Nbr	*Cost*
B	5
D	3
E	0

To / From	B	D	E
B	0	-	-
D	-	0	-
E	5, B	3, D	0, E

The following questions indicate events that happen consecutively. You can assume that there are no other packet exchanges than the ones specified.

A. *C sends its update to A and D.*
  A.1. What information is contained in C's update?

  A.2. What do the routing tables for A and D look like after receiving C's update? (You may not need to fill in all columns)

Node A

Nbr	*Cost*
A	0
B	2
C	1
D	7

To From					
A					
B					
C					
D					

Node D

Nbr	*Cost*
A	7
C	2
D	0
E	3

To From					
A					
C					
D					
E					

  A.3. Which nodes among A and D are expected to send routing updates after receiving C's update?

B. *A sends its update to B, C, and D.*
  B.1. What information is contained in A's update?

  B.2. What do the routing tables for B, C, and D look like after receiving A's update (You may not need to fill in all columns)?

Node B

Nbr	*Cost*
A	2
B	0
E	5

A					
B					
E					

Node C

Nbr	*Cost*
A	1
C	0
D	2

A					
C					
D					

Node D

Nbr	*Cost*
A	7
C	2
D	0
E	3

B.3. At this point, what route does D use to reach B? It knows that it can route to A via C with total distance 3 and that A can reach B with distance 2. Should it use this information to optimize the route to B or should it wait for an update from C?

B.4. Which nodes among B, C, and D are expected to send routing updates after receiving A's update?

C. **Skip this part in section, and use it as review for the exams.**
D sends its update to A, C, and E.
C.1. What information is contained in D's update?

C.2. What do the routing tables for A, C, and E look like after receiving D's update? (You may not need to fill in all columns)

Node A

Nbr	*Cost*
A	2
B	7
C	0
D	5

A					
B					
C					
D					

Node C

Nbr	*Cost*
A	1
C	0
D	2

A					
C					
D					

Node E

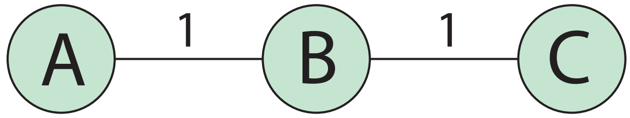
Nbr	*Cost*
B	5
D	3
E	0

C.3. Which nodes among A, C, and E are expected to send routing updates after receiving D's update?

D. Have the routing tables converged? Why or why not?

Problem 2: Count-To-Infinity Problem

Consider this simple topology:



A. What values will be in the routing tables when the system has stabilized (after many rounds)? For this question, assume each node advertises all its real distances to all its neighbors.

Node A

Nbr	*Cost*
A	0
B	1

A			
B			

Node B

Nbr	*Cost*
A	1
B	0
C	1

A			
B			
C			

Node C

Nbr	*Cost*
B	1
C	0

B. Now suppose the link from A to B goes down, such that A is no longer reachable:

B.1. B notices the link outage and updates its routing table. What does B's updated routing table look like?

Node B

Nbr	*Cost*
A	1
B	0
C	1

A			
B			
C			

B.2. According to its routing table, what is the cost of B's minimum-cost path to A?

C. B sends an update to C. What is C's routing table after receiving the update?

Node C

Nbr	*Cost*
B	1
C	0

D. After updating its table, C sends an update to B. What is B's routing table after receiving the update?

Node B

Nbr	*Cost*
A	1
B	0
C	1

A			
B			
C			

E. How many updates are exchanged before the tables converge?

Problem 3: Poison Reverse

One solution to the count-to-infinity problem is "poison-reverse": if you are currently routing through a neighbor, tell that neighbor that your path to the destination has infinite cost.

A. Consider the network topology in Problem 2. Assuming that poison reverse was used when exchanging route information, what does B's routing table look like before the link from A to B goes down?

Node B

Nbr	*Cost*
A	1
B	0
C	1

A			
B			
C			

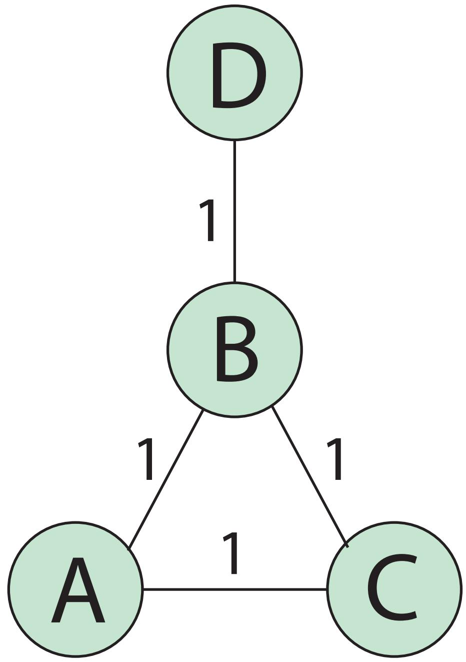
B. *B detects the link outage and sends an update to C.*
B.1. What information is contained in B's update?

B.2. What does C's routing table look like after receiving the update?

Node C

Nbr	*Cost*
B	1
C	0

C. Now consider a more complex topology, with stabilized routing tables for A, B, and C:



Node A

Nbr	*Cost*
A	0
B	1
C	1

	A	B	C	D
A	0, A	1, B	1, C	2, B
B	1	0	1	1
C	1	1	0	2

Node B

Nbr	*Cost*
A	1
B	0
C	1
D	1

	A	B	C	D
A	0	1	1	∞
B	1, A	0	1, C	1, D
C	1	1	0	∞
D	∞	1	∞	0

Node C

Nbr	*Cost*
A	1
B	1
C	0

	A	B	C	D
A	0	1	1	2
B	1	0	1	1
C	1, A	1, B	0, C	2, B

Suppose the link between B and D goes down. B notices this change and sends an update to A.

C.1. What is A's routing table after processing B's update?

Node A

Nbr	*Cost*
A	0
B	1
C	1

A				
B				
C				

C.2. A then sends an update back to B. What is B's routing table after processing A's update?

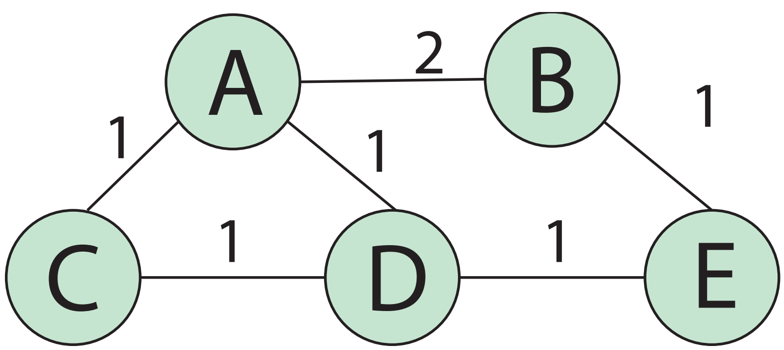
Node B

Nbr	*Cost*
A	1
B	0
C	1
D	1

A				
B				
C				
D				

C.3. How might you avoid the count-to-infinity problem here altogether?

Problem 4: Split Poison? Poisoned horizon?



A. Assume that the routers use **split horizon**. Say that E sends its initial update (B: 1, D: 1) to D. Assuming that D has received no other updates, what does D now tell E about D's path to B?

B. Assume that the routers use **poisoned reverse**. Furthermore, assume that the routing tables haven't converged, and D believes its shortest path to B is D-A-B (length 3). D sends this update to E. Now, E sends its first update (D: 1, B: 1) to D. After recomputing its routes, D sends an update to E. In this update, what is the advertised distance to B?

C. Now assume that the routers use **split horizon *and* poisoned reverse**. After the same scenario as in 2), what distance to B does D advertise to E?

D. Consider the simple topology (A-B-C) from problem 2. After the routing tables have converged, link A-B goes down. When B sends C an update containing (A: ∞), is this an act of **poisoning a route** or **poisoned reverse**?

E. **Poisoning a route** and **poisoned reverse** might sound similar, but actually we can think of one of them as being "honest" while the other one is "lying." Which one tells the truth, and which one tells a white lie to keep the network functioning?