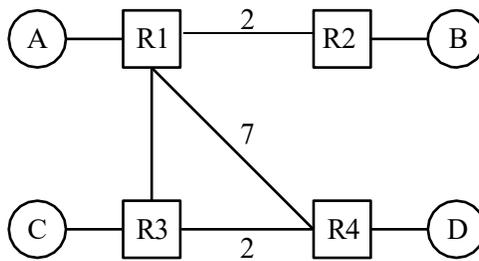


# Lecture 5.2 - Distance-Vector Exercises ANS

## 1 Distance Vector

Consider running the distance-vector protocol on the topology below. Unlabeled links have cost 1.



The routing tables start out initially with direct routes only:

R1's table

Dest.	Hop, Dist.
A	Direct, 1

R2's table

Dest.	Hop, Dist.
B	Direct, 1

R3's table

Dest.	Hop, Dist.
C	Direct, 1

R4's table

Dest.	Hop, Dist.
D	Direct, 1

Assumptions for this question:

- Each subpart continues on from the previous subparts. After finishing each subpart, we suggest first copying your answer to the next subpart before solving the next subpart.
- No other events occur other than the ones specified.
- We use *triggered updates*: a router sends out advertisements immediately after its table updates.
- We do not use *incremental updates*: when a router sends out advertisements, it advertises all entries in its table.
- You may not need to fill in all the rows.

1.1 EVENT: R3 advertises its routes to R1 and R4.

What do the routing tables look like after receiving R3's routes? Entries in **bolded red** denote changes caused by new advertisements made in this subpart.

R1's table

Dest.	Hop, Dist.
A	Direct, 1
<b>C</b>	<b>R3, 2</b>

R2's table

Dest.	Hop, Dist.
B	Direct, 1

R3's table

Dest.	Hop, Dist.
C	Direct, 1

R4's table

Dest.	Hop, Dist.
D	Direct, 1
<b>C</b>	<b>R3, 3</b>

1.2 Which routers will advertise their routes after receiving R3's routes?

Since both R1 and R4 updated their tables, they will both advertise their routing tables.

1.3 EVENT: R1 advertises its routes to R2, R3, and R4.

What do the routing tables look like after receiving R1's routes?

R1's table		R2's table		R3's table		R4's table	
Dest.	Hop, Dist.						
A	Direct, 1	B	Direct, 1	C	Direct, 1	D	Direct, 1
C	R3, 2	A	R1, 3	A	R1, 2	C	R3, 3
		C	R1, 4			A	R1, 8

1.4 EVENT: R4 advertises its routes to R1 and R3.

What do the routing tables look like after receiving R4's routes?

R1's table		R2's table		R3's table		R4's table	
Dest.	Hop, Dist.						
A	Direct, 1	B	Direct, 1	C	Direct, 1	D	Direct, 1
C	R3, 2	A	R1, 3	A	R1, 2	C	R3, 3
D	R4, 8	C	R1, 4	D	R4, 3	A	R1, 8

1.5 EVENT: R1 advertises its routes to R2, R3, and R4.

What do the routing tables look like after receiving R1's routes?

R1's table		R2's table		R3's table		R4's table	
Dest.	Hop, Dist.						
A	Direct, 1	B	Direct, 1	C	Direct, 1	D	Direct, 1
C	R3, 2	A	R1, 3	A	R1, 2	C	R3, 3
D	R4, 8	C	R1, 4	D	R4, 3	A	R1, 8
		D	R1, 10				

1.6 At this point, what path does R2 use to reach D, and what is the cost?

R2 → R1 → R4 with a cost of 10.

R1 has only heard about a route to D from R4. R1 in turn advertises this route to R2.

1.7 EVENT: R3 advertises its routes to R1 and R4.

What do the routing tables look like now?

R1's table                      R2's table                      R3's table                      R4's table

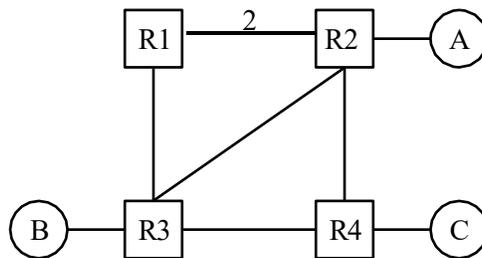
Dest.	Hop, Dist.						
A	Direct, 1	B	Direct, 1	C	Direct, 1	D	Direct, 1
C	R3, 2	A	R1, 3	A	R1, 2	C	R3, 3
D	R3, 4	C	R1, 4	D	R4, 3	A	R3, 4
		D	R1, 10				

- 1.8 Let us now reflect on the routing state after all the advertisements in the previous subparts. In theory, under the most optimal routing state that's attainable, what is the least-cost path that R2 could use to reach D? Do the current routing tables reflect this? If not, what additional advertisement(s) could be done to allow R2 to reach D optimally? Express them in this form: **Router X advertises its routes to Router Y,**

The optimal (least-cost) path theoretically attainable from R2 to D is  $R2 \rightarrow R1 \rightarrow R3 \rightarrow R4 \rightarrow D$ , with optimal cost 6. The current routing state does not reflect this, because in subpart 1.7, R2 still has not heard about R3's routing update to R1 about a lower cost route to D yet. If next, **R1 advertises its routes to R2 about its new routes attained after subpart 1.7**, then the new routing state will finally allow R2 to reach D optimally!

This is just one possible sequence of router advertisements, and in reality, any sequence may be possible, since routers send advertisements periodically and asynchronously. (Note that R2 did not advertise in this sequence, so R1, R3, R4 do not have routes to B via R2 yet.)

## 2 Split Horizon and Poison



All unlabeled links have a cost of 1. The parts of the question do not build on each other.

- 2.1 Assume that the routers use **split horizon**. Say that R4 advertises (A: 2, C : 1) to R3. Assuming that R3 has received no other advertisements, what does R3 now tell R4 about R3's path to A?

Nothing. Split Horizon means that we never tell a neighbor about paths that go through that neighbor. So in this case, R3 doesn't tell R4 about its path to A.

- 2.2 Assume that the routers use **poisoned reverse**. Routing tables have not converged and R3 believes its shortest path to A is through R1 (this path is R3-R1-R2 of length 4). R3 advertises its routes to R4. Now, R4 advertises to R3. R4 bases this advertisement off of its routing table which has: (B: 2, A: 2, C : 1). After recomputing its routes, R3 advertises its routes to R4. What is the advertised distance to A?

R3 will tell R4 that its distance to A is infinity (a white lie), because R3's new shortest route to A goes through

R4 (this path is R3-R4-R2 of length 3), by poisoned reverse. (Note that R2 has not advertised in our sequence of events, so R3 does not yet know about the shortest path R3-R2 to A.)

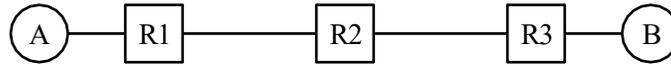
- 2.3 Consider the simple topology (A-R1-R2-R3). Assume that routing tables have converged, with R2 believing its shortest path to A is through R1 (this path is R2-R1-A of length 3). Then, suppose that link R1-R2 goes down. When R2 advertises to R3 ( $A: \infty$ ), is this an act of poisoning a route or poisoned reverse?

R2 is poisoning a route. Namely, it tells R3 that its distance is  $\infty$  because R2 actually has no route to A now.

- 2.4 Route Poisoning and Poisoned Reverse sound similar, but 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?

Poisoned reverse encourages routers to tell a white lie. With poisoned reverse, we tell a neighbor that we have no path to a certain destination if our path goes through that neighbor. Since we actually do have a path, our message is not strictly true. On the other hand, Route Poisoning happens when a link goes down, and we actually lose our path to some destination. Thus, we’re telling the truth when we advertise a distance of  $\infty$  to this destination (given that an infinitely long path is equivalent to no path).

### 3 Count to Infinity



For part 1 of this question there is **no split-horizon or poisoned reverse**, and advertisements are only sent periodically (aka when it is explicitly stated).

3.1 What do the routing tables look like once R1, R2 and R3 converge?

R1's table

Dest.	Hop, Dist.
A	Direct, 1
B	R2, 3

R2's table

Dest.	Hop, Dist.
A	R1, 2
B	R3, 2

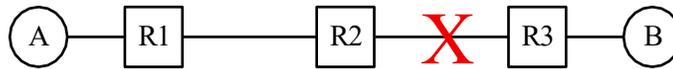
R3's table

Dest.	Hop, Dist.
B	Direct, 1
A	R2, 3

3.2 What periodic advertisement will R1 and R2 send to each other? (One such message is given as an example.0)

From	To	(Destination, Distance)
R1	R2	(A, 1)
R1	R2	(B, 3)
R2	R1	(A, 2)
R2	R1	(B, 2)

3.3 EVENT: The link between R2 and R3 goes down.



What will R1 and R2 send to each other?

From	To	(Destination, Distance)
R1	R2	(A, 1)
R1	R2	(B, 3)
R2	R1	(A, 2)
R2	R1	(B, 2)

The tables did not change because no routes have expired yet.

3.4 EVENT: R2's route to B finally expires, so its table entry B / R3, 2 is deleted.

After R1 and R2 exchange advertisements again, what will their routing tables look like?  
R2's table gets a new entry B / R1, 4.

R1's table

Dest.	Hop, Dist.
A	Direct, 1
B	R2, 3

R2's table

Dest.	Hop, Dist.
A	R1, 2
B	R1, 4

R3's table

Dest.	Hop, Dist.
B	Direct, 1
A	R2, 3

3.5 After R1 and R2 exchange advertisements again, what will their routing tables look like?  
R1's table gets a new entry B / R2, 5. (based on Rule 2: Updates From Next-Hop)

R1's table

Dest.	Hop, Dist.
A	Direct, 1
B	R2, 5

R2's table

Dest.	Hop, Dist.
A	R1, 2
B	R1, 4

R3's table

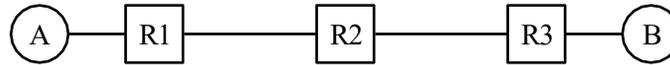
Dest.	Hop, Dist.
B	Direct, 1
A	R2, 3

3.6 Is this good?

(Question 3 continued...)

No! This is called count to infinity. Both routers think they have a path to B for a long time after the path ceases to exist.

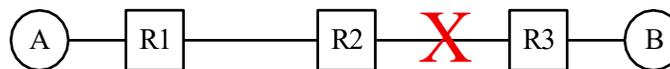
For the remainder of this question, **there is split-horizon, but no poisoned reverse**, and advertisements are only sent periodically (i.e., when it is explicitly stated). Also, all dropped links are back up, and the routing state starts out converged!



3.7 What will R1 and R2 send to each other after everything has converged? (R1 does not advertise to R2 route to B via R2; R2 does not advertise to R1 route to A via R1.)

From	To	(Destination, Distance)
R1	R2	(A, 1)
R2	R1	(B, 2)

3.8 EVENT: The link between R2 and R3 goes down.



What will R1 and R2 send to each other? (Same.)

From	To	(Destination, Distance)
R1	R2	(A, 1)
R2	R1	(B, 2)

3.9 EVENT: R2's route to B via R3 finally expires, so its table entry B / R3, 2 is deleted.

After R1 and R2 exchange advertisements again, what will their routing tables look like?

Dest.	Hop, Dist.
A	Direct, 1
B	R2, 3

R1's table

Dest.	Hop, Dist.
A	R1, 2

R2's table

Dest.	Hop, Dist.
B	Direct, 1
A	R2, 3

R3's table

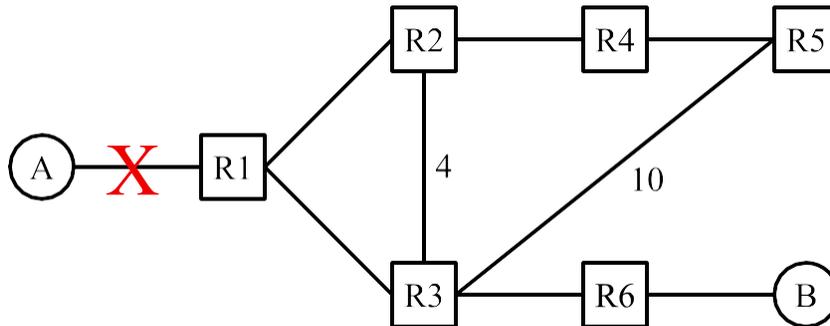
3.10 Will this end well?

Yes! R1's route to B via R2 will expire because it has not been updated in a while (R1 still hears from R2, but not about the route to B via R2). And R3's route to A via R2 will expire (R3 never hears from R2



(Question 3 continued...)

- No poison is ever sent.
- A long time later, the network has converged. After convergence, host A leaves the network.
- At time  $t = 100$ , R2 still has its (stale) entry for destination A (the same entry that it had at convergence).



All other routers have had their entries for destination A expired and deleted.

Q4.4 At time  $t = 103$ , what table entry does each router have for destination A?

Note:  $t = 103$  means you should consider three rounds of sending advertisements, receiving advertisements, and updating tables. The first round ( $t = 101$ ) starts with R2 advertising its entry for A.

Some entries are filled in for you. For example, the bottom-right row says that R6's forwarding table has the entry "I can reach A with cost 7, via R3."

In the Cost column, write an integer, or  $\infty$ , or "N/A" (if the table has no entry for destination A).

In the Next-Hop column, write a router (e.g. "R1"), or "N/A" (if the entry has cost infinity, or if the table has no entry for destination A).

Router	Cost to A	Next-Hop to A	Router	Cost to A	Next-Hop to A
R1	Solution: 7	Solution: R3	R4	3	R2
R2	Solution: 8	Solution: R1	R5	Solution: 4	Solution: R4
R3	Solution: 6	Solution: R2	R6	7	R3

ANS: At  $t = 101$ , R2 has an entry "I can reach A via R1." This is advertised to R3 and R4 (but not R1 because of split horizon).

At  $t = 102$ , R3 advertises to R1, R6, and R5 (but not R2 because of split horizon). Also, at  $t = 102$ , R4 advertises to R5 (but not R2 because of split horizon).

At  $t = 103$ , R1 advertises to R2 (but not R3).

Q4.5 Many time steps later, what happens to the routers' table entries for destination A? (Reminder: All costs 16 or greater are infinite, and all costs 15 or less are finite.)

ANS: Every router's entry has infinite cost. We have encountered the count-to-infinity problem. A nonexistent path is being advertised in the R1-R2-R3 loop, causing their costs to increase to infinity. Those high costs are also advertised to R4, R5, and R6, causing their entries to increase to infinity as well. (The full table on the next page is not required for the exam.)

**Solution:**

The asterisk denotes an entry that updated at that time step. For example, at  $t = 101$ , R3 and R4 have their forwarding tables updated.

$t = 100$ :

R2 can reach with cost 2 via R1 \*

$t = 101$ :

R2 can reach with cost 2 via R1

R3 can reach with cost 6 via R2 \*

R4 can reach with cost 3 via R2 \*

$t = 102$ :

R1 can reach with cost 7 via R3 \*

R2 can reach with cost 2 via R1

R3 can reach with cost 6 via R2

R4 can reach with cost 3 via R2

R5 can reach with cost 4 via R4 \*

R6 can reach with cost 7 via R3 \*

$t = 103$ :

R1 can reach with cost 7 via R3

R2 can reach with cost 8 via R1 \*

R3 can reach with cost 6 via R2

R4 can reach with cost 3 via R2

R5 can reach with cost 4 via R4

R6 can reach with cost 7 via R3

$t = 104$ :

R1 can reach with cost 7 via R3

R2 can reach with cost 8 via R1

R3 can reach with cost 12 via R2 \*

R4 can reach with cost 9 via R2 \*

R5 can reach with cost 4 via R4

R6 can reach with cost 7 via R3

$t = 105$ :

R1 can reach with cost 13 via R3 \*

R2 can reach with cost 8 via R1

R3 can reach with cost 12 via R2

R4 can reach with cost 9 via R2

R5 can reach with cost 10 via R4 \*

R6 can reach with cost 13 via R3 \*

$t = 106$ :

R1 can reach with cost 13 via R3

R2 can reach with cost 14 via R1 \*

R3 can reach with cost 12 via R2

R4 can reach with cost 9 via R2

R5 can reach with cost 10 via R4

R6 can reach with cost 13 via R3

$t = 107$ :

R1 can reach with cost 13 via R3

R2 can reach with cost 14 via R1

R3 can reach with cost 18 via R2 \*

R4 can reach with cost 15 via R2 \*

R5 can reach with cost 10 via R4

R6 can reach with cost 13 via R3

$t = 108$ :

R1 can reach with cost 19 via R3 \*

R2 can reach with cost 14 via R1

R3 can reach with cost 18 via R2

R4 can reach with cost 15 via R2

R5 can reach with cost 16 via R4 \*

R6 can reach with cost 19 via R3 \*

$t = 109$ :

R1 can reach with cost 19 via R3

R2 can reach with cost 20 via R1 \*

R3 can reach with cost 18 via R2

R4 can reach with cost 15 via R2

R5 can reach with cost 16 via R4

R6 can reach with cost 19 via R3

$t = 110$ :

R1 can reach with cost 19 via R3

R2 can reach with cost 20 via R1

R3 can reach with cost 24 via R2 \*

R4 can reach with cost 19 via R2 \*

R5 can reach with cost 16 via R4

R6 can reach with cost 19 via R3