

# CPSC 441

## Assignment 4

### Distance Vector Routing

1. Explain how distance vector routing works.

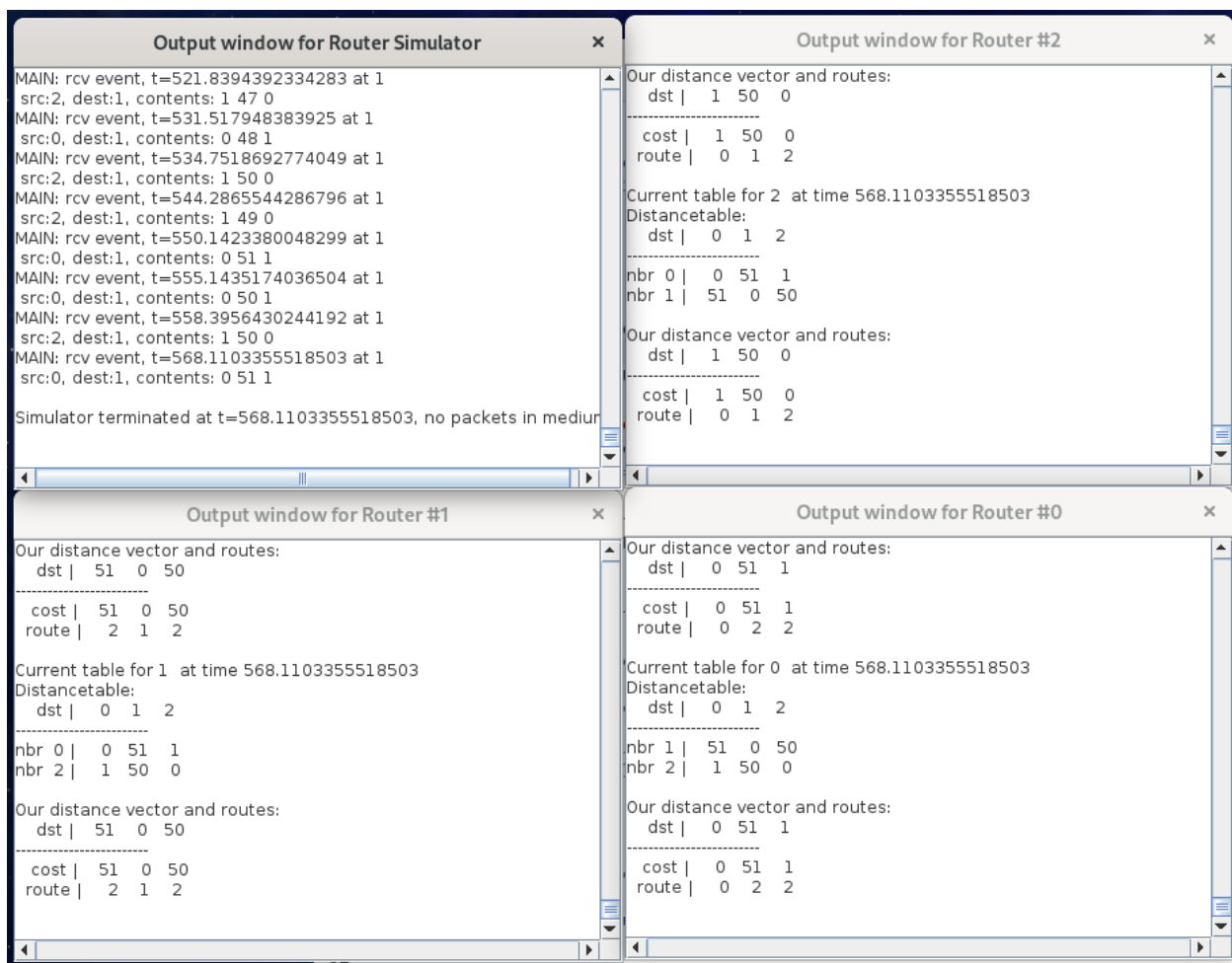
Distance vector routing is a simple protocol used in packet switching networks that utilizes distance to decide the best packet forwarding path. Distance is typically represented by hop count. The distance vector routing protocol is a shortest path from one router to another across a network.

2. Describe how you tested your solution.

I tested my solution against the three given test cases and observed the output was as expected. The screenshots at question #3 contain information regarding how I executed the tests.

3. Screenshots of your outputs for three test cases.

The following commands have were ran to get the output in the screenshots. (Note: poisoned reverse was set to *false*).



- make install 3
- make test

Output window for Router Simulator

```

MAIN: rcv event, t=122.20657671372095 at 1
src:0, dest:1, contents: 0 1 2 4
MAIN: rcv event, t=123.42102041713242 at 2
src:0, dest:2, contents: 0 1 3 5
MAIN: rcv event, t=128.0844280645295 at 2
src:1, dest:2, contents: 1 0 1 3
MAIN: rcv event, t=133.38759705990208 at 2
src:0, dest:2, contents: 0 1 3 7
MAIN: rcv event, t=141.8451188328788 at 2
src:3, dest:2, contents: 4 3 2 0
MAIN: rcv event, t=145.85521874722264 at 2
src:0, dest:2, contents: 0 1 2 4
MAIN: rcv event, t=153.34837892290938 at 2
src:0, dest:2, contents: 0 1 2 5
MAIN: rcv event, t=161.5412284907392 at 2
src:0, dest:2, contents: 0 1 2 4

Simulator terminated at t=161.5412284907392, no packets in medium

```

Output window for Router #0

dst	0	1	2	4
cost	0	1	2	4
route	0	1	1	1

Current table for 0 at time 161.5412284907392

Distanctetable:

dst	0	1	2	3
nbr 1	1	0	1	3
nbr 2	2	1	0	2
nbr 3	4	3	2	0

Our distance vector and routes:

dst	0	1	2	4
cost	0	1	2	4
route	0	1	1	1

Output window for Router #3

dst	4	3	2	0
cost	4	3	2	0
route	2	2	2	3

Current table for 3 at time 161.5412284907392

Distanctetable:

dst	0	1	2	3
nbr 0	0	1	2	4
nbr 1	999	999	999	999
nbr 2	2	1	0	2

Our distance vector and routes:

dst	4	3	2	0
cost	4	3	2	0
route	2	2	2	3

Output window for Router #1

dst	1	0	1	3
cost	1	0	1	3
route	0	1	2	2

Current table for 1 at time 161.5412284907392

Distanctetable:

dst	0	1	2	3
nbr 0	0	1	2	4
nbr 2	2	1	0	2
nbr 3	999	999	999	999

Our distance vector and routes:

dst	1	0	1	3
cost	1	0	1	3
route	0	1	2	2

Output window for Router #2

dst	2	1	0	2
cost	2	1	0	2
route	1	1	2	3

Current table for 2 at time 161.5412284907392

Distanctetable:

dst	0	1	2	3
nbr 0	0	1	2	4
nbr 1	1	0	1	3
nbr 3	4	3	2	0

Our distance vector and routes:

dst	2	1	0	2
cost	2	1	0	2
route	1	1	2	3

- make install4
- make test

**Output window for Router Simulator**

```

MAIN: rcv event, t=248.7830584220748 at 2
src:0, dest:2, contents: 0 1 2 4
MAIN: rcv event, t=250.20540797854832 at 4
src:0, dest:4, contents: 0 1 2 4
MAIN: rcv event, t=254.94686912042292 at 1
src:0, dest:1, contents: 0 1 2 4
MAIN: rcv event, t=256.1836613828707 at 0
src:4, dest:0, contents: 1 1 2 6
MAIN: rcv event, t=257.0876678979504 at 3
src:0, dest:3, contents: 0 1 2 4
MAIN: rcv event, t=264.3889994415473 at 0
src:2, dest:0, contents: 2 1 0 2
MAIN: rcv event, t=266.96660437467375 at 0
src:3, dest:0, contents: 4 3 2 0
MAIN: rcv event, t=271.16550027082724 at 0
src:4, dest:0, contents: 1 1 2 4
Simulator terminated at t=271.16550027082724, no packets in medium

```

**Output window for Router #4**

```

cost | 1 1 2 4 0
route | 0 1 1 1 4

Current table for 4 at time 271.16550027082724
Distancetable:
dst | 0 1 2 3 4

nbr 0 | 0 1 2 4 1
nbr 1 | 1 0 1 3 1
nbr 2 | 2 1 0 2 2
nbr 3 | 999 999 999 999 999

Our distance vector and routes:
dst | 1 1 2 4 0

cost | 1 1 2 4 0
route | 0 1 1 1 4

```

**Output window for Router #2**

```

cost | 2 1 0 2 2
route | 1 1 2 3 1

Current table for 2 at time 271.16550027082724
Distancetable:
dst | 0 1 2 3 4

nbr 0 | 0 1 2 4 1
nbr 1 | 1 0 1 3 1
nbr 3 | 4 3 2 0 4
nbr 4 | 1 1 2 4 0

Our distance vector and routes:
dst | 2 1 0 2 2

cost | 2 1 0 2 2
route | 1 1 2 3 1

```

**Output window for Router #3**

```

cost | 4 3 2 0 4
route | 2 2 2 3 2

Current table for 3 at time 271.16550027082724
Distancetable:
dst | 0 1 2 3 4

nbr 0 | 0 1 2 4 1
nbr 1 | 999 999 999 999 999
nbr 2 | 2 1 0 2 2
nbr 4 | 999 999 999 999 999

Our distance vector and routes:
dst | 4 3 2 0 4

cost | 4 3 2 0 4
route | 2 2 2 3 2

```

**Output window for Router #0**

```

cost | 0 1 2 4 1
route | 0 1 1 1 4

Current table for 0 at time 271.16550027082724
Distancetable:
dst | 0 1 2 3 4

nbr 1 | 1 0 1 3 1
nbr 2 | 2 1 0 2 2
nbr 3 | 4 3 2 0 4
nbr 4 | 1 1 2 4 0

Our distance vector and routes:
dst | 0 1 2 4 1

cost | 0 1 2 4 1
route | 0 1 1 1 4

```

**Output window for Router #1**

```

cost | 1 0 1 3 1
route | 0 1 2 2 4

Current table for 1 at time 271.16550027082724
Distancetable:
dst | 0 1 2 3 4

nbr 0 | 0 1 2 4 1
nbr 2 | 2 1 0 2 2
nbr 3 | 999 999 999 999 999
nbr 4 | 1 1 2 4 0

Our distance vector and routes:
dst | 1 0 1 3 1

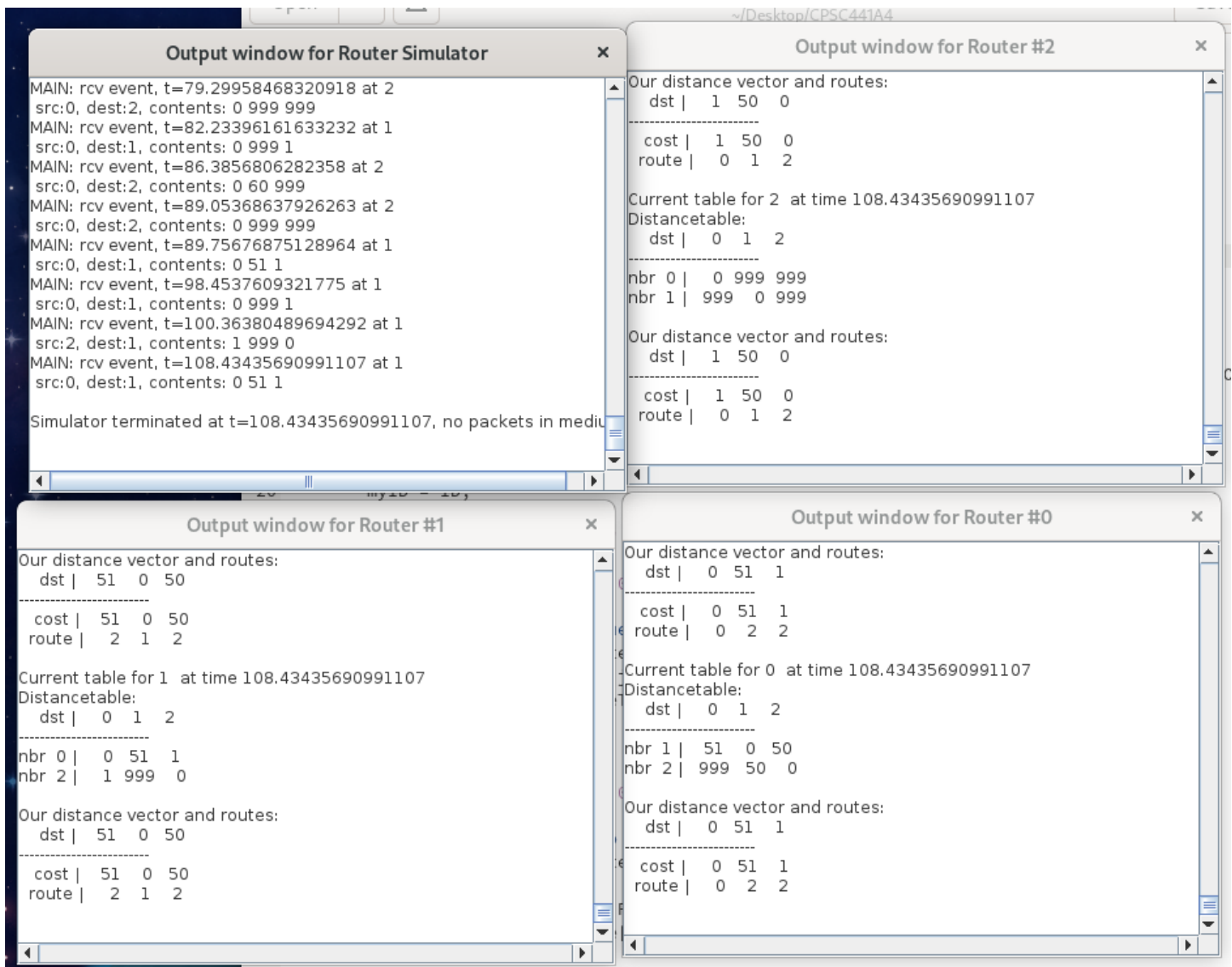
cost | 1 0 1 3 1
route | 0 1 2 2 4

```

- make install5
- make test

#### 4. Show in a test case why poisoned reverse is necessary.

This is RouterSimulator3.java being ran. In this case, Poisoned Reverse was set to True. The simulation terminated at time 108.43, whereas before (with poisoned reverse set to False), the simulator would take up to time 568.11 to finish. In this scenario, Poisoned Reverse was necessary, because RouterSimulator3 has a change in default Link Costs, and without poisoned reverse, the algorithm needs longer to find the shortest path.



##### 5. Explain how poisoned reverse works.

The idea of poison reverse is to make sure that a path does not turn back into the same node if a cost has changed within the network of routers, this would lead to a count-to-infinity problem. With poison reverse, two routers, A and B, that have no way to get rid of the loop in a timely manner using plain Bellman-Ford, announce an infinite metric default route to each other, so that the algorithm gets rid of the routing loop as soon as an update is successfully transmitted.