

# Distance Vector Routing Assignment

---

Data Communications I Author: Brian Reid

## Getting started

In order to run this program, you will need to install the Java SDK. Adoptium is a common choice. You can download it [here](#).

Compile .java files.

Once you are set up with the Java SDK, you will be able to compile the Java files in this repository. The command `javac *.java` will compile all \*.java files.

## Run the Project

There should now be a corresponding .class file for each .java file in the repository. To run the program and reproduce the results seen in this submission, simply run `java Project`.

## Design

### How it works

The project is a distributed distance vector routing algorithm. In this scenario, each node in the network calculates the minimum distance to each of its neighboring nodes, and then broadcasts that information to those same neighboring nodes. That way, if node A is connected to B and node B is connected to node C but nodes A and C are not connected, node A will learn some information about node C when node B broadcasts its distance table.

Each node is initialized such that each entry in the distance table is set to 999 (which we're considering to be infinity). Once this is complete, the known distances are manually set (read: hard coded) into the distance table. With more time, this would be one of the first things that I would change. It would be much more scalable to keep a representation of the network in a place that is publicly accessible, read that data into the node during initialization, and determine the shortest path to each neighboring node. Obviously we would need to be careful not to give the node more information from the onset of the program, but this design is much more flexible. This would allow us to much more easily change the layout of the network graph from a central location and test the program in a much more robust way. This design would also allow each node to dynamically determine a list of its neighbor nodes during initialization. That value is currently hardcoded into each node as an Array.

The distance table computed above is then read to determine the least-cost path to each neighboring node. In other words, each row of the table is scanned and we store the minimum value of that row in a new integer array that is declared to store the minimum distances. This knowledge of minimum distances is then broadcast out to all the neighbor nodes using the `Packet` class and `NetworkSimulator.toLayer2()` method. Simply, for each node, that node sends information to each of its surrounding nodes saying "this is my node index, here are the node indexes that neighbor me, and these are the shortest distances I have discovered to each of them."

When a node receives a minimum distance packet from a neighbor, it must process that information, update its own distance table accordingly, and broadcast any changes back out to its neighbors. This is handled in the `update()` method. We first declare a boolean variable at the beginning method that defaults to `false` so that we can track whether or not a change has been made to the distance table. We only broadcast packets to neighbors if this boolean becomes true. This prevents cluttering the network with the unnecessary traffic of packets that do not represent a change in the network. We then re-compute the minimum distance array. This is a wasted computation that I would like to change. The minimum distance array should be maintained on the class so we only need to recompute it when necessary and can access it more easily. With the minimum distances now recomputed, we check to see whether the cost of getting to any node via the source node that sent us the packet is cheaper than the cost we know about. For example, if the known cost to node C from A is 999, suppose node B tells node A that it can get to C with a cost of 10. We now take the known cost to B (suppose this is 5), add it to the cost from B to C, and get 15. Since 15 is less than 999, A will update its distance table and min distance array to reflect the cost of 15. Because this change was made, the aforementioned boolean is flipped to `true` (if it isn't already), and we'll then broadcast the new min distances out to the neighbors.

The overarching theme for what I would like to change here from a design perspective is that I would like to extract out a lot of the reused code and make this design more scalable. In the essence of time, I have a lot of reused code between each of the entities that should really be factored out and made more commonly available. This is ok in its current state since the code is only spread across four nodes, but it's certainly not ideal. This project will not be able to scale until that happens.

## Testing and Defects

The bottom line here is that this solution "works" in the sense that it finds the right answer for minimum distances. If you go through the logging and check the distance tables near program termination, the minimum distances in each row are correct. However, there are a couple spots where the data in the table is incorrect. For example, at the bottom of the first trace (no link changes), D2 seems to think that it can route to D3 via D0 for a cost of 6. The minimum acyclic route cost in that satisfies this requirement is 9 (D2 -> D1 -> D0 -> D3), but there is a cyclic route that costs 7 (D2 -> D0 -> D1 -> D2 -> D3). Obviously 6 is neither of these numbers. I have an off-by-one error somewhere that is causing this, but I haven't found it yet. This also brings up another interesting design question of whether nodes should be considering routes that involve cycles. The obvious answer is no, especially if we have a scenario where there is a cycle with a negative link cost in the network. This could send the algorithm into an infinite loop trying to find the lowest cost route.

Running the program in the setting where link costs do change results in a similar plight of correct overall solutions with some anomalies in the data. I have also noticed that in certain cases the program does not correctly converge when the links change. The result here is the program terminates with some 999 values in the distance tables. This leads me to believe that my `LinkChangeCostHandler()` methods are imperfect. This result is not consistently reproducible, but may manifest itself if you use a random seed.

## Sample output

Trace level 2, weights do not change:

```

brianreid@brians-air distance_vector_routing % java Project
Network Simulator v1.0
Enter trace level (>= 0): [0] 2
Will the link change (1 = Yes, 0 = No): [0] 0
Enter random seed: [random]
Entity0 Initialized.. Distance Table is:

```

	via		
D0	1	2	3
1	1	999	999
2	999	2	999
3	999	999	7

```

Entity1 Initialized at.. Distance Table is:

```

	via	
D1	0	2
0	1	999
2	999	1
3	999	999

```

Entity2 Initialized.. Distance Table is:

```

	via		
D2	0	1	3
0	3	999	999
1	999	1	999
3	999	999	2

```

Entity3 Initialized .. Distance Table is:

```

	via	
D3	0	2
0	7	999
1	999	999
2	999	2

```

main(): event received.  t=1.0019298928945441, node=1

```

```

    src=2, dest=1, contents=[3, 1, 999, 2]

```

```

Detected a min distance change at Entity1... broadcasting to other nodes.

```

```

Entity1 Updated.. Distance Table is:

```

	via	
D1	0	2
0	1	4
2	999	1

```
3| 999 3
```

```
main(): event received. t=1.473746997348084, node=0
src=1, dest=0, contents=[1, 999, 1, 999]
```

```
Detected a min distance change at Entity0... broadcasting to other nodes.
```

```
Entity0 Updated.. Distance Table is:
```

	via		
D0	1	2	3
-----+-----			
1	1	999	999
2	2	2	999
3	999	999	7

```
main(): event received. t=2.1321941473964374, node=1
src=0, dest=1, contents=[999, 1, 2, 7]
```

```
Entity1 Updated.. Distance Table is:
```

	via	
D1	0	2
-----+-----		
0	1	4
2	3	1
3	8	3

```
main(): event received. t=2.998715114618939, node=0
src=3, dest=0, contents=[7, 999, 2, 999]
```

```
Entity0 Updated.. Distance Table is:
```

	via		
D0	1	2	3
-----+-----			
1	1	999	999
2	2	2	9
3	999	999	7

```
main(): event received. t=4.030662158031729, node=3
src=2, dest=3, contents=[3, 1, 999, 2]
```

```
Detected a min distance change at Entity3... broadcasting to other nodes.
```

```
Entity3 Updated.. Distance Table is:
```

	via	
D3	0	2
-----+-----		
0	7	5
1	999	3
2	999	2

```
main(): event received. t=4.420649226788617, node=1
```

```
src=0, dest=1, contents=[2, 1, 2, 7]
Entity1 Updated.. Distance Table is:
```

	via	
D1	0	2
-----+-----		
0	1	4
2	3	1
3	8	3

```
main(): event received. t=4.9368685480657994, node=2
src=0, dest=2, contents=[999, 1, 2, 7]
Detected a min distance change at Entity2... broadcasting to other nodes.
```

```
Entity2 Updated.. Distance Table is:
```

	via		
D2	0	1	3
-----+-----			
0	3	999	999
1	4	1	999
3	10	999	2

```
main(): event received. t=6.350543684833641, node=3
src=2, dest=3, contents=[3, 1, 5, 2]
Entity3 Updated.. Distance Table is:
```

	via	
D3	0	2
-----+-----		
0	7	5
1	999	3
2	999	2

```
main(): event received. t=6.80953910077873, node=1
src=2, dest=1, contents=[3, 1, 5, 2]
Entity1 Updated.. Distance Table is:
```

	via	
D1	0	2
-----+-----		
0	1	4
2	3	1
3	8	3

```
main(): event received. t=7.554514883892335, node=3
src=0, dest=3, contents=[999, 1, 2, 7]
Entity3 Updated.. Distance Table is:
```

	via	
D3	0	2

-----+-----		
0	7	5
1	6	3
2	7	2

main(): event received. t=7.7395229378372195, node=2

src=3, dest=2, contents=[7, 999, 2, 999]

Detected a min distance change at Entity2... broadcasting to other nodes.

Entity2 Updated.. Distance Table is:

via		
D2	0	1 3
-----+-----		
0	3 999	9
1	4 1 999	
3	10 999	2

main(): event received. t=8.03571903690814, node=0

src=2, dest=0, contents=[3, 1, 999, 2]

Detected a min distance change at Entity0... broadcasting to other nodes.

Entity0 Updated.. Distance Table is:

via		
D0	1	2 3
-----+-----		
1	1 3 999	
2	2 2 9	
3	999 4 7	

main(): event received. t=8.921408301865505, node=0

src=1, dest=0, contents=[1, 2, 1, 3]

Entity0 Updated.. Distance Table is:

via		
D0	1	2 3
-----+-----		
1	1 3 999	
2	2 2 9	
3	4 4 7	

main(): event received. t=9.603319465464619, node=1

src=2, dest=1, contents=[3, 1, 4, 2]

Entity1 Updated.. Distance Table is:

via		
D1	0	2
-----+-----		
0	1	4

```

2| 3 1
3| 8 3

```

```

main(): event received.  t=9.699966550379274, node=2
    src=1, dest=2, contents=[1, 999, 1, 999]
Detected a min distance change at Entity2... broadcasting to other nodes.

```

Entity2 Updated.. Distance Table is:

```

      via
D2 |  0  1  3
----+-----
0|  3  2  9
1|  4  1 999
3| 10 999  2

```

```

main(): event received.  t=10.117428674511878, node=2
    src=3, dest=2, contents=[5, 3, 2, 4]
Entity2 Updated.. Distance Table is:

```

```

      via
D2 |  0  1  3
----+-----
0|  3  2  7
1|  4  1  5
3| 10 999  2

```

```

main(): event received.  t=12.253662416317548, node=1
    src=0, dest=1, contents=[2, 1, 2, 4]
Entity1 Updated.. Distance Table is:

```

```

      via
D1 |  0  2
----+-----
0|  1  4
2|  3  1
3|  5  3

```

```

main(): event received.  t=12.2784030661314, node=1
    src=2, dest=1, contents=[2, 1, 2, 2]
Entity1 Updated.. Distance Table is:

```

```

      via
D1 |  0  2
----+-----
0|  1  3
2|  3  1
3|  5  3

```

```

main(): event received.  t=13.402534191576391, node=2

```

```
src=1, dest=2, contents=[1, 2, 1, 3]
Entity2 Updated.. Distance Table is:
```

		via		
D2		0	1	3
-----+-----				
0		3	2	7
1		4	1	5
3		10	4	2

```
main(): event received. t=13.736148329254739, node=0
src=3, dest=0, contents=[5, 3, 2, 4]
Entity0 Updated.. Distance Table is:
```

		via		
D0		1	2	3
-----+-----				
1		1	3	7
2		2	2	6
3		4	4	7

```
main(): event received. t=14.678400320556575, node=2
src=0, dest=2, contents=[2, 1, 2, 7]
Entity2 Updated.. Distance Table is:
```

		via		
D2		0	1	3
-----+-----				
0		3	2	7
1		3	1	5
3		9	4	2

```
main(): event received. t=15.027178083716338, node=3
src=2, dest=3, contents=[3, 1, 4, 2]
Entity3 Updated.. Distance Table is:
```

		via	
D3		0	2
-----+-----			
0		7	5
1		6	3
2		7	2

```
main(): event received. t=16.033601483489534, node=0
src=2, dest=0, contents=[3, 1, 5, 2]
Entity0 Updated.. Distance Table is:
```

		via		
D0		1	2	3
-----+-----				



```

1| 1 3 7
2| 2 2 6
3| 4 4 7

```

```

main(): event received. t=16.055055107673823, node=3
src=0, dest=3, contents=[2, 1, 2, 7]
Entity3 Updated.. Distance Table is:

```

```

      via
D3 | 0 2
----+-----
0| 7 5
1| 6 3
2| 7 2

```

```

main(): event received. t=20.846785681193545, node=3
src=2, dest=3, contents=[2, 1, 2, 2]
Detected a min distance change at Entity3... broadcasting to other nodes.

```

```

Entity3 Updated.. Distance Table is:

```

```

      via
D3 | 0 2
----+-----
0| 7 4
1| 6 3
2| 7 2

```

```

main(): event received. t=21.037716028423937, node=0
src=2, dest=0, contents=[3, 1, 4, 2]
Entity0 Updated.. Distance Table is:

```

```

      via
D0 | 1 2 3
----+-----
1| 1 3 7
2| 2 2 6
3| 4 4 7

```

```

main(): event received. t=22.62442560443737, node=0
src=3, dest=0, contents=[4, 3, 2, 4]
Entity0 Updated.. Distance Table is:

```

```

      via
D0 | 1 2 3
----+-----
1| 1 3 7
2| 2 2 6
3| 4 4 7

```

```

main(): event received. t=23.860747603672866, node=2
src=0, dest=2, contents=[2, 1, 2, 4]

```

Entity2 Updated.. Distance Table is:

D2	via		
	0	1	3
-----+-----			
0	3	2	7
1	3	1	5
3	6	4	2

main(): event received. t=25.042094070469233, node=3

src=0, dest=3, contents=[2, 1, 2, 4]

Entity3 Updated.. Distance Table is:

D3	via	
	0	2
-----+-----		
0	6	4
1	5	3
2	6	2

main(): event received. t=27.525296311625873, node=0

src=2, dest=0, contents=[2, 1, 2, 2]

Entity0 Updated.. Distance Table is:

D0	via		
	1	2	3
-----+-----			
1	1	3	7
2	2	2	6
3	4	4	7

main(): event received. t=30.82119695746792, node=2

src=3, dest=2, contents=[4, 3, 2, 4]

Entity2 Updated.. Distance Table is:

D2	via		
	0	1	3
-----+-----			
0	3	2	6
1	3	1	5
3	6	4	2

Simulator terminated at t=30.82119695746792, no packets in medium.

brianreid@brians-air distance\_vector\_routing %

Trace level, weights do change:

```
brianreid@Brians-Air distance_vector_routing % java Project
Network Simulator v1.0
Enter trace level (>= 0): [0] 2
```

Will the link change (1 = Yes, 0 = No): [0] 1

Enter random seed: [random]

Entity0 Initialized.. Distance Table is:

	via		
D0	1	2	3
-----+-----			
1	1	999	999
2	999	2	999
3	999	999	7

Entity1 Initialized at.. Distance Table is:

	via	
D1	0	2
-----+-----		
0	1	999
2	999	1
3	999	999

Entity2 Initialized.. Distance Table is:

	via		
D2	0	1	3
-----+-----			
0	3	999	999
1	999	1	999
3	999	999	2

Entity3 Initialized .. Distance Table is:

	via	
D3	0	2
-----+-----		
0	7	999
1	999	999
2	999	2

main(): event received. t=1.063280270080512, node=3

src=0, dest=3, contents=[999, 1, 2, 7]

Detected a min distance change at Entity3... broadcasting to other nodes.

Entity3 Updated.. Distance Table is:

	via	
D3	0	2
-----+-----		
0	7	999
1	8	999
2	9	2

main(): event received. t=2.215634737197012, node=2

src=0, dest=2, contents=[999, 1, 2, 7]

Detected a min distance change at Entity2... broadcasting to other nodes.

Entity2 Updated.. Distance Table is:

		via		
D2		0	1	3
-----+-----				
0		3	999	999
1		4	1	999
3		10	999	2

main(): event received. t=2.371120797552103, node=0

src=2, dest=0, contents=[3, 1, 999, 2]

Detected a min distance change at Entity0... broadcasting to other nodes.

Entity0 Updated.. Distance Table is:

		via		
D0		1	2	3
-----+-----				
1		1	3	999
2		999	2	999
3		999	4	7

main(): event received. t=3.900966482389725, node=0

src=3, dest=0, contents=[7, 999, 2, 999]

Entity0 Updated.. Distance Table is:

		via		
D0		1	2	3
-----+-----				
1		1	3	999
2		999	2	6
3		999	4	7

main(): event received. t=3.9687515844929484, node=1

src=0, dest=1, contents=[999, 1, 2, 7]

Detected a min distance change at Entity1... broadcasting to other nodes.

Entity1 Updated.. Distance Table is:

		via	
D1		0	2
-----+-----			
0		1	999
2		3	1
3		8	999

main(): event received. t=4.0282819230630444, node=3

src=2, dest=3, contents=[3, 1, 999, 2]

Detected a min distance change at Entity3... broadcasting to other nodes.

Entity3 Updated.. Distance Table is:

via		
D3	0	2
-----+-----		
0	7	5
1	8	3
2	9	2

main(): event received. t=6.352321547752828, node=0  
src=1, dest=0, contents=[1, 999, 1, 999]  
Detected a min distance change at Entity0... broadcasting to other nodes.

Entity0 Updated.. Distance Table is:

via			
D0	1	2	3
-----+-----			
1	1	3	999
2	2	2	6
3	999	4	7

main(): event received. t=7.045643920228855, node=3  
src=2, dest=3, contents=[3, 1, 5, 2]  
Entity3 Updated.. Distance Table is:

via		
D3	0	2
-----+-----		
0	7	5
1	8	3
2	9	2

main(): event received. t=7.1944635128447985, node=3  
src=0, dest=3, contents=[5, 1, 2, 4]  
Entity3 Updated.. Distance Table is:

via		
D3	0	2
-----+-----		
0	7	5
1	6	3
2	7	2

main(): event received. t=7.889298866575477, node=2  
src=1, dest=2, contents=[1, 999, 1, 999]  
Detected a min distance change at Entity2... broadcasting to other nodes.

Entity2 Updated.. Distance Table is:

via

D2	0	1	3
-----+-----			
0	3	2	999
1	4	1	999
3	10	999	2

main(): event received. t=8.420826231479639, node=1

src=2, dest=1, contents=[3, 1, 999, 2]

Detected a min distance change at Entity1... broadcasting to other nodes.

Entity1 Updated.. Distance Table is:

	via		
D1	0	2	
-----+-----			
0	1	4	
2	3	1	
3	8	3	

main(): event received. t=9.259534454697294, node=2

src=3, dest=2, contents=[7, 999, 2, 999]

Entity2 Updated.. Distance Table is:

	via		
D2	0	1	3
-----+-----			
0	3	2	9
1	4	1	999
3	10	999	2

main(): event received. t=9.338227770058095, node=0

src=3, dest=0, contents=[7, 8, 2, 14]

Entity0 Updated.. Distance Table is:

	via		
D0	1	2	3
-----+-----			
1	1	3	12
2	2	2	6
3	999	4	7

main(): event received. t=9.777321376939906, node=0

src=2, dest=0, contents=[3, 1, 5, 2]

Entity0 Updated.. Distance Table is:

	via		
D0	1	2	3
-----+-----			
1	1	3	12
2	2	2	6

```
3| 999  4  7
```

```
main(): event received.  t=10.072839426709457, node=2
  src=0, dest=2, contents=[5, 1, 2, 4]
Entity2 Updated.. Distance Table is:
```

	via		
D2	0	1	3
0	3	2	9
1	3	1	999
3	6	999	2

```
main(): event received.  t=10.384684930244651, node=1
  src=2, dest=1, contents=[3, 1, 5, 2]
Entity1 Updated.. Distance Table is:
```

	via	
D1	0	2
0	1	4
2	3	1
3	8	3

```
main(): event received.  t=11.623174163807846, node=1
  src=0, dest=1, contents=[5, 1, 2, 4]
Entity1 Updated.. Distance Table is:
```

	via	
D1	0	2
0	1	4
2	3	1
3	5	3

```
main(): event received.  t=13.16810024608833, node=2
  src=1, dest=2, contents=[1, 2, 1, 8]
Entity2 Updated.. Distance Table is:
```

	via		
D2	0	1	3
0	3	2	9
1	3	1	999
3	6	9	2

```
main(): event received.  t=13.23662636754019, node=1
  src=0, dest=1, contents=[2, 1, 2, 4]
Entity1 Updated.. Distance Table is:
```

via		
D1	0	2
-----+-----		
0	1	4
2	3	1
3	5	3

main(): event received. t=13.387977182677478, node=3  
 src=2, dest=3, contents=[2, 1, 2, 2]  
 Detected a min distance change at Entity3... broadcasting to other nodes.

Entity3 Updated.. Distance Table is:

via		
D3	0	2
-----+-----		
0	7	4
1	6	3
2	7	2

main(): event received. t=13.624322669609167, node=0  
 src=1, dest=0, contents=[1, 2, 1, 8]  
 Entity0 Updated.. Distance Table is:

via			
D0	1	2	3
-----+-----			
1	1	3	12
2	2	2	6
3	9	4	7

main(): event received. t=14.507528762158485, node=1  
 src=2, dest=1, contents=[2, 1, 2, 2]  
 Entity1 Updated.. Distance Table is:

via		
D1	0	2
-----+-----		
0	1	3
2	3	1
3	5	3

main(): event received. t=14.529151943066875, node=0  
 src=2, dest=0, contents=[2, 1, 2, 2]  
 Entity0 Updated.. Distance Table is:

via			
D0	1	2	3
-----+-----			
1	1	3	12



```

2|  2  2  6
3|  9  4  7

```

```

main(): event received.  t=14.754944552313464, node=3
      src=0, dest=3, contents=[2, 1, 2, 4]
Entity3 Updated.. Distance Table is:

```

```

      via
D3 |  0  2
----+-----
0|  6  4
1|  5  3
2|  6  2

```

```

main(): event received.  t=14.79484888559142, node=2
      src=3, dest=2, contents=[7, 8, 2, 14]
Entity2 Updated.. Distance Table is:

```

```

      via
D2 |  0  1  3
----+-----
0|  3  2  9
1|  3  1 10
3|  6  9  2

```

```

main(): event received.  t=16.26206408285531, node=0
      src=3, dest=0, contents=[5, 3, 2, 4]
Entity0 Updated.. Distance Table is:

```

```

      via
D0 |  1  2  3
----+-----
1|  1  3  7
2|  2  2  6
3|  9  4  7

```

```

main(): event received.  t=18.660719020025354, node=2
      src=0, dest=2, contents=[2, 1, 2, 4]
Entity2 Updated.. Distance Table is:

```

```

      via
D2 |  0  1  3
----+-----
0|  3  2  9
1|  3  1 10
3|  6  9  2

```

```

main(): event received.  t=19.70805395441185, node=2
      src=3, dest=2, contents=[5, 3, 2, 4]
Entity2 Updated.. Distance Table is:

```

D2	via		
	0	1	3
-----+-----			
0	3	2	7
1	3	1	5
3	6	9	2

main(): event received. t=20.921414567758866, node=2  
 src=1, dest=2, contents=[1, 2, 1, 3]  
 Entity2 Updated.. Distance Table is:

D2	via		
	0	1	3
-----+-----			
0	3	2	7
1	3	1	5
3	6	4	2

main(): event received. t=22.085719181660835, node=2  
 src=3, dest=2, contents=[4, 3, 2, 4]  
 Entity2 Updated.. Distance Table is:

D2	via		
	0	1	3
-----+-----			
0	3	2	6
1	3	1	5
3	6	4	2

main(): event received. t=22.629835963451175, node=0  
 src=1, dest=0, contents=[1, 2, 1, 3]  
 Entity0 Updated.. Distance Table is:

D0	via		
	1	2	3
-----+-----			
1	1	3	7
2	2	2	6
3	4	4	7

main(): event received. t=25.529492557482143, node=0  
 src=3, dest=0, contents=[4, 3, 2, 4]  
 Entity0 Updated.. Distance Table is:

D0	via		
	1	2	3
-----+-----			
1	1	3	7
2	2	2	6

```
3| 4 4 7
```

```
main(): event received. t=10000.0, node=0
```

```
Link cost change.
```

```
Entity0 Link Change Cost Completed. Distance Table is:
```

```

      via
D0 | 1 2 3
----+-----
1| 20 999 999
2| 999 2 999
3| 999 999 7
```

```
Entity1 Link Change Cost Completed.. Distance Table is:
```

```

      via
D1 | 0 2
----+-----
0| 20 999
2| 999 1
3| 999 999
```

```
main(): event received. t=10001.012898296894, node=0
```

```
src=1, dest=0, contents=[20, 999, 1, 999]
```

```
Detected a min distance change at Entity0... broadcasting to other nodes.
```

```
Entity0 Updated.. Distance Table is:
```

```

      via
D0 | 1 2 3
----+-----
1| 20 999 999
2| 21 2 999
3| 999 999 7
```

```
main(): event received. t=10001.89787595896, node=3
```

```
src=0, dest=3, contents=[999, 20, 2, 7]
```

```
Entity3 Updated.. Distance Table is:
```

```

      via
D3 | 0 2
----+-----
0| 6 4
1| 5 3
2| 6 2
```

```
main(): event received. t=10002.194181945872, node=2
```

```
src=0, dest=2, contents=[999, 20, 2, 7]
```

```
Entity2 Updated.. Distance Table is:
```

```
via
```

D2		0	1	3
0		3	2	6
1		3	1	5
3		6	4	2

main(): event received. t=10002.342884113286, node=1  
src=0, dest=1, contents=[999, 20, 2, 7]  
Detected a min distance change at Entity1... broadcasting to other nodes.

Entity1 Updated.. Distance Table is:

		via
D1		0 2
0		20 999
2		22 1
3		27 999

main(): event received. t=10005.838226421421, node=1  
src=0, dest=1, contents=[40, 20, 2, 7]  
Entity1 Updated.. Distance Table is:

		via
D1		0 2
0		20 999
2		22 1
3		27 999

main(): event received. t=10007.357551407817, node=0  
src=1, dest=0, contents=[20, 40, 1, 27]  
Entity0 Updated.. Distance Table is:

		via
D0		1 2 3
1		20 999 999
2		21 2 999
3		47 999 7

main(): event received. t=10007.53376584747, node=2  
src=0, dest=2, contents=[40, 20, 2, 7]  
Entity2 Updated.. Distance Table is:

		via
D2		0 1 3
0		3 2 6
1		3 1 5

```
3| 6 4 2
```

```
main(): event received. t=10009.791323308293, node=2
src=1, dest=2, contents=[20, 999, 1, 999]
Entity2 Updated.. Distance Table is:
```

	via		
D2	0	1	3
0	3	2	6
1	3	1	5
3	6	4	2

```
main(): event received. t=10010.04196477642, node=3
src=0, dest=3, contents=[40, 20, 2, 7]
Entity3 Updated.. Distance Table is:
```

	via	
D3	0	2
0	6	4
1	5	3
2	6	2

```
main(): event received. t=10016.225128141854, node=2
src=1, dest=2, contents=[20, 40, 1, 27]
Entity2 Updated.. Distance Table is:
```

	via		
D2	0	1	3
0	3	2	6
1	3	1	5
3	6	4	2

```
Simulator terminated at t=10016.225128141854, no packets in medium.
brianreid@Brians-Air distance_vector_routing %
```