This program attempts to mimic how a network of routers build a routing table based on Distance Vector Routing (DVR). This means that for each router (node0, node1, node2, node3) in the network, it will compute the distance between itself and its immediate neighbors and share that information with those neighbors. The neighbors can then update its distance tables accordingly with new minimum costs to a given node.

In an actual implementation of the DVR, we would be making each node share information with its neighbors at regular intervals, but we are not doing that for this project.

Additionally, we assume that:

* All connections between nodes are bidirectional and not directed in any way
* There are no edges/connections with negative weights
* The problem size will be small so we don’t have to worry about the count-to-infinity problem

From the amount of packets being sent around in the network, we can see that it isn’t a huge amount. We can also see from our results that it doesn’t converge quickly, with up to 5 seconds of runtime on our machines for DVR to finish. These results are well within our expectations, as those are characteristics present within DVR.

In theory, if we implemented DVR correctly, we should obtain the following distance tables for each node:

|  |  |  |  |
| --- | --- | --- | --- |
| D0 | 1 | 2 | 3 |
| 1 | 1 | 4 | 10 |
| 2 | 2 | 3 | 9 |
| 3 | 4 | 5 | 4 |

|  |  |  |
| --- | --- | --- |
| D1 | 0 | 2 |
| 0 | 1 | 4 |
| 2 | 4 | 1 |
| 3 | 6 | 3 |

|  |  |  |  |
| --- | --- | --- | --- |
| D2 | 0 | 1 | 3 |
| 0 | 3 | 2 | 9 |
| 1 | 4 | 1 | 10 |
| 3 | 7 | 9 | 2 |

|  |  |  |
| --- | --- | --- |
| D3 | 0 | 2 |
| 0 | 7 | 5 |
| 1 | 8 | 3 |
| 2 | 9 | 2 |