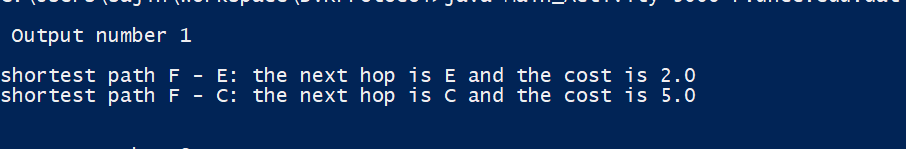
Implementation of Distance Vector Routing Protocol

The route is as below-

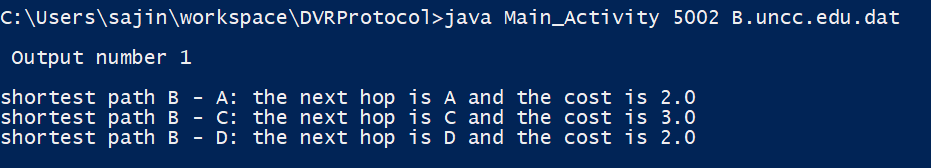


Every node has only its neighbours initially.

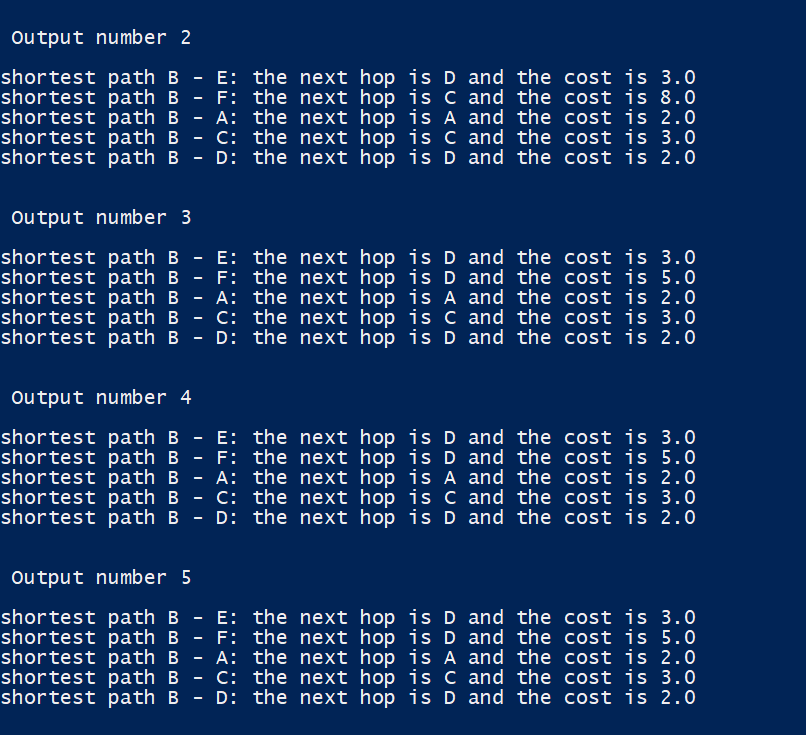
Node F

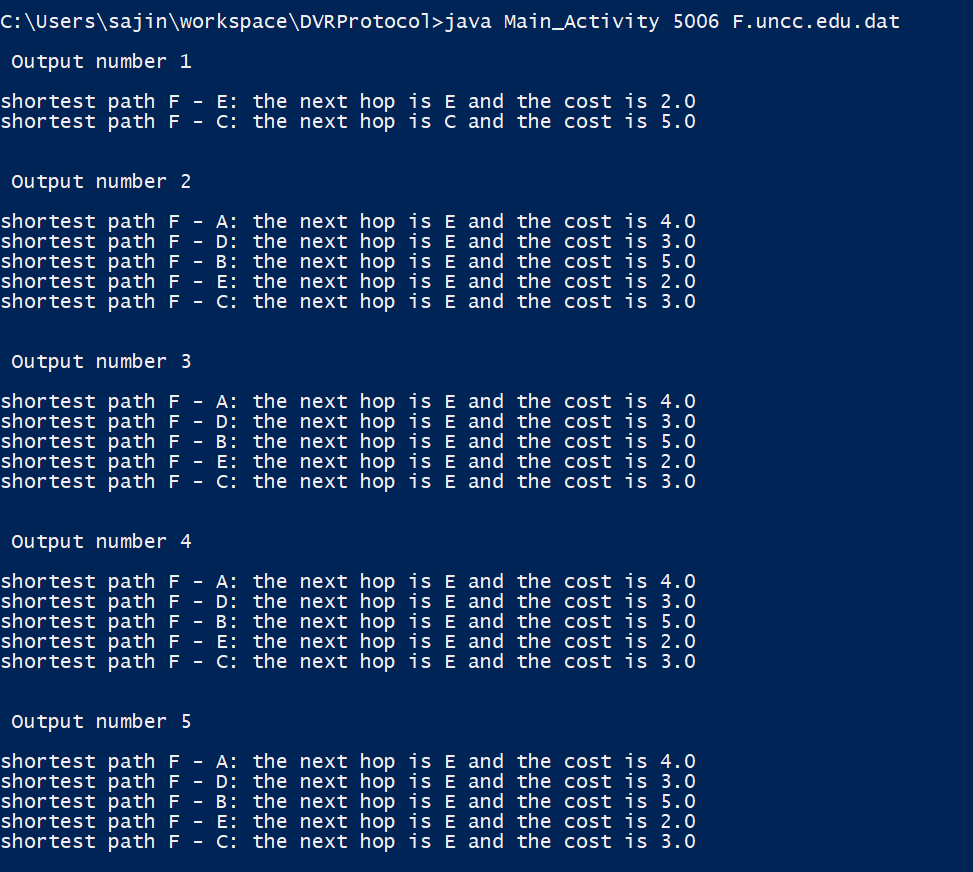


Node B



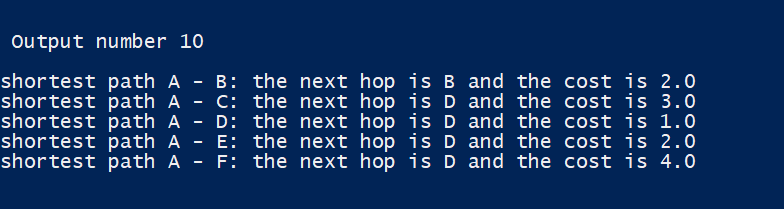
As each node broadcasts the routing information, each router learns the shortest paths, and the screenshots below show the routing information for the first few iterations.

  
The slow starting neighbour will eventually get the information. Below is the information for node F.

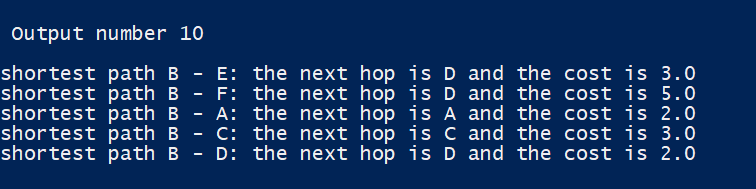


For the above test input, below is the shortest path for the nodes –

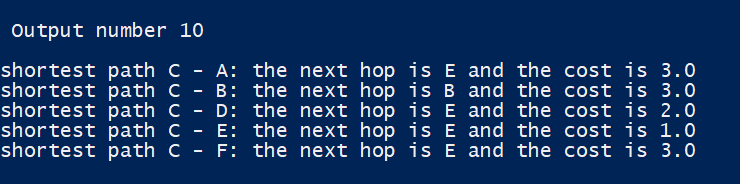
At node A



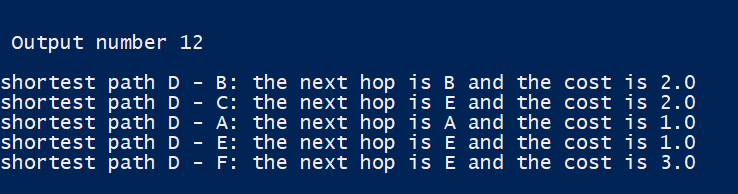
At node B



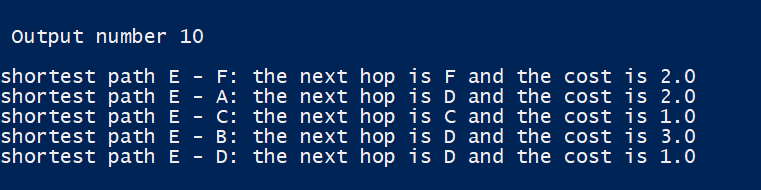
At node C



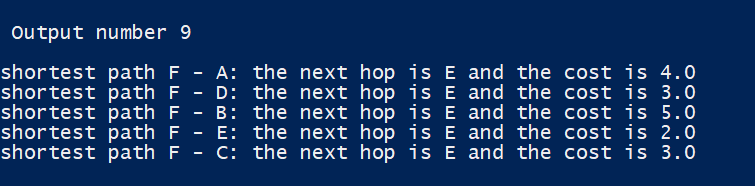
At node D



At node E



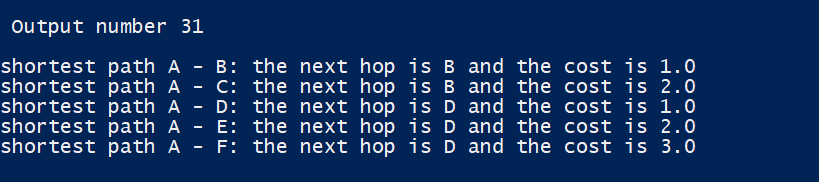
At node F



**Link Cost Change Case**

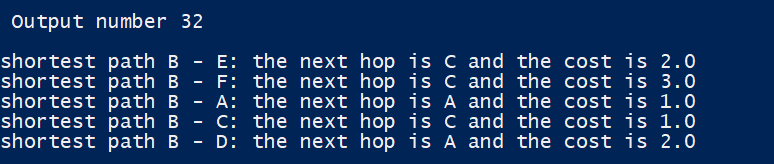
If we change the link cost at node A to B as 1, B to C as 1, and E to F as 1, the following results can be observed at each of the nodes.

At node A

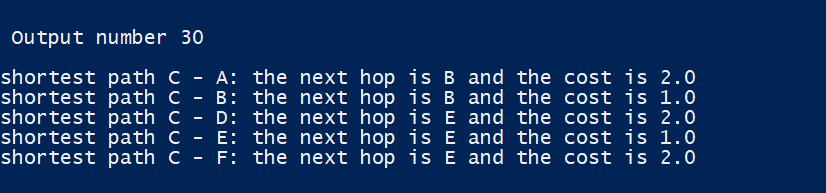


As you can see cost to F is updated to 3 because of these changes. The costs to all the nodes in the system is updated similarly.

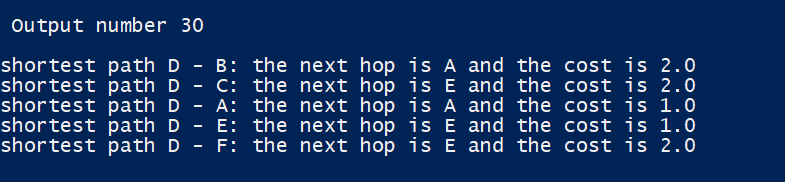
At node B



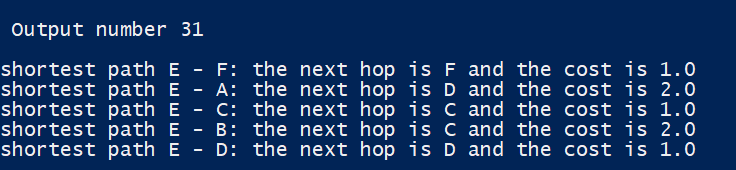
At node C



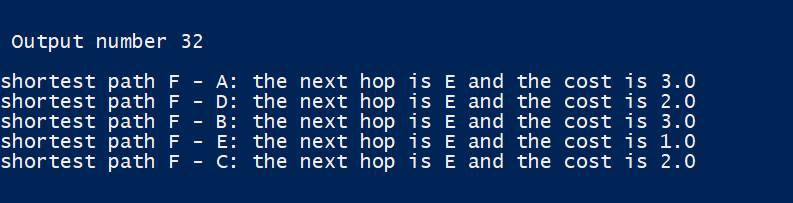
At node D



At node E

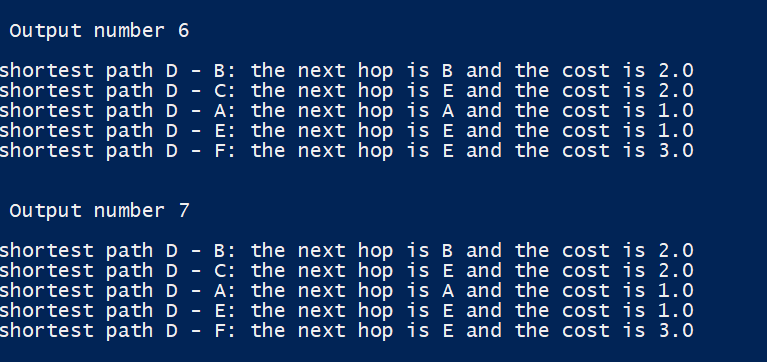


At node F



**Recursive update problem**

Using reverse poisoning method to send the advertisement with infinite or very high value. When a node does not respond, this takes place.



**Implementation details**

* The sender and receiver run on two separate threads.
* Each thread run in an infinite loop and send routing information.
* Receiver reads the port number from the command line argument and creates a listening socket at that port.
* It then extracts the node and cost information and stores in the list of neighbours and ncpair hashtable
* Whenever new routing information in received, it is run against the Distance vector algorithm and new updated cost and corresponding node pair is stored
* Sender sends the packet to the port mentioned in the file corresponding to the node
* At the sender, it first reads the input file, recognizes the host/node name, and adds to the neighbours array list.
* It records the cost in a Hashtable hcpair as a pair of hostname and cost
* For each host in the neighbours list, it will send its neighbouring cost details stored in hcpair in formatted manner.
* The sender thread displays the data after the algorithm is finished.
* The Sender sends information every 15 seconds and sleeps after that.