**Computer Communications and Networks**

**Project-3**

**Implementing Distance Vector Routing Protocol**

**Arun Kunnumpuram Thomas – 801027386**

**Sanjay Tiwari – 800968058**

**Implementation –**

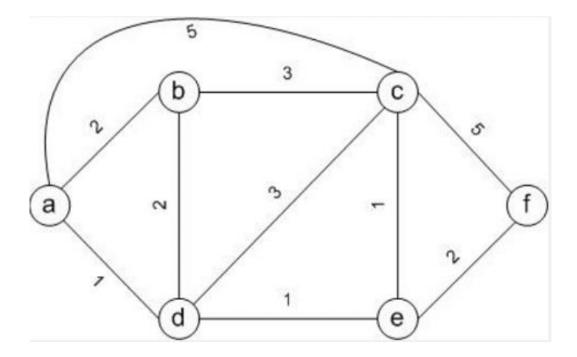
We implemented Distance Vector Routing protocol using Java programming language. There is only class called Router.java. We implemented the protocol on single as well as multiple machines (we tested on six machines). The output from both scenario gave us the correct result.

**Working –**

1. On executing, local variables are being initialized.
2. Two threads are there for each router.
3. One thread is for reading the distance vector information from the network.
4. Another thread is for calculating the shortest path based on the distance vector algorithm and writing the new information to neighbors.

**Instructions for Execution on Single Machine –**

We are using the network topology shown below –



1. Compile the java files for all routers manually –

C:\Users\sanja\Desktop\MS in CS\CCN\Project3\a>javac Router.java

C:\Users\sanja\Desktop\MS in CS\CCN\Project3\b>javac Router.java

C:\Users\sanja\Desktop\MS in CS\CCN\Project3\c>javac Router.java

C:\Users\sanja\Desktop\MS in CS\CCN\Project3\d>javac Router.java

C:\Users\sanja\Desktop\MS in CS\CCN\Project3\e>javac Router.java

C:\Users\sanja\Desktop\MS in CS\CCN\Project3\f>javac Router.java

1. Open six command prompts and run Router.java on each command line – We have given IP of 127.0.0.1 for all routers and port numbers as – router **a** has 9001 port number, router **b** has 9002, router **c** has 9003, router **d** has 9004, router **e** has 9005, router **f** has 9006 (you can give any port numbers you want for each router).

While running, we have pass multiple arguments for each router. The details of arguments are given below –

C:\Users\sanja\Desktop\MS in CS\CCN\Project3\a>java Router 6 router1.dat 9001 1 127.0.0.1:9002 127.0.0.1:9003 127.0.0.1:9004

In above,

**6** is the total number of routers in the topology.

**router1.dat** is the input file for router a (similarly, router2.dat is for router b and so on).

**9001** is the port number of router a.

**1** is ID for router a.

**127.0.0.1:9002** is the IP address and port number of its directly connected neighbors (same with other two arguments).

**For Router a –**

C:\Users\sanja\Desktop\MS in CS\CCN\Project3\a>java Router 6 router1.dat 9001 1 127.0.0.1:9002 127.0.0.1:9003 127.0.0.1:9004

**For Router b –**

C:\Users\sanja\Desktop\MS in CS\CCN\Project3\b>java Router 6 router2.dat 9002 2 127.0.0.1:9001 127.0.0.1:9003 127.0.0.1:9004

**For Router c –**

C:\Users\sanja\Desktop\MS in CS\CCN\Project3\c>java Router 6 router3.dat 9003 3 127.0.0.1:9001 127.0.0.1:9002 127.0.0.1:9004 127.0.0.1:9005 127.0.0.1:9006

**For Router d –**

C:\Users\sanja\Desktop\MS in CS\CCN\Project3\d>java Router 6 router4.dat 9004 4 127.0.0.1:9001 127.0.0.1:9002 127.0.0.1:9003 127.0.0.1:9005

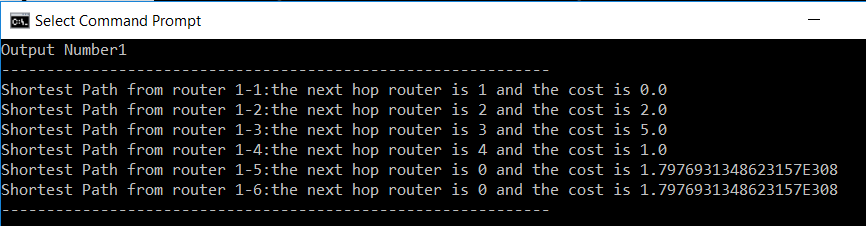
**For Router e –**

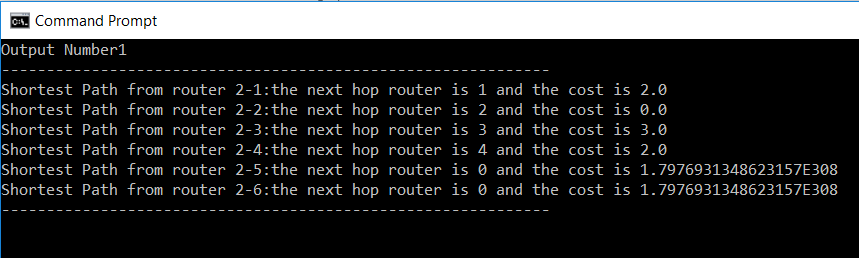
C:\Users\sanja\Desktop\MS in CS\CCN\Project3\e>java Router 6 router5.dat 9005 5 127.0.0.1:9003 127.0.0.1:9004 127.0.0.1:9006

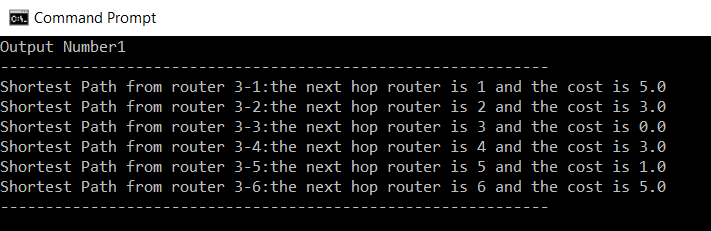
**For Router f –**

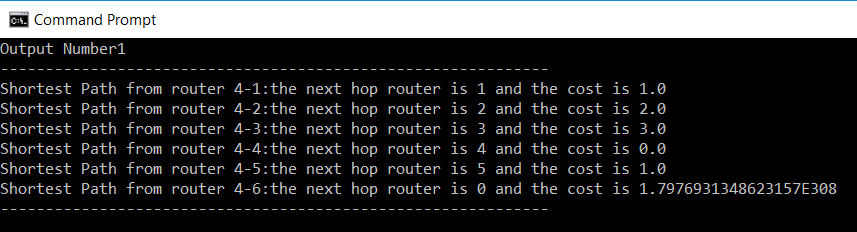
C:\Users\sanja\Desktop\MS in CS\CCN\Project3\f>java Router 6 router6.dat 9006 6 127.0.0.1:9003 127.0.0.1:9005

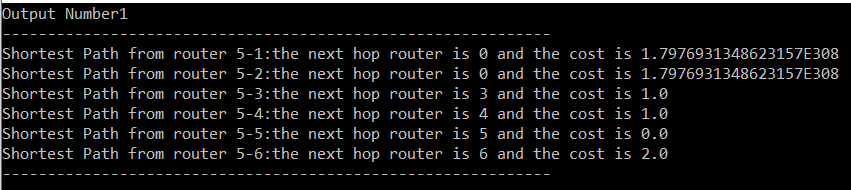
1. Initial terminal output for each router is given below –

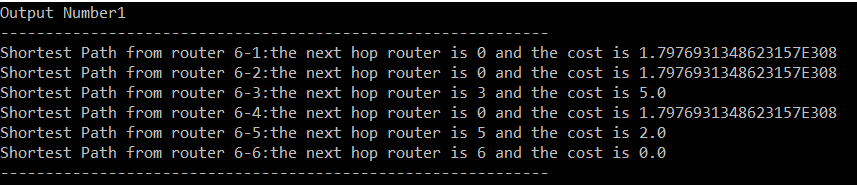






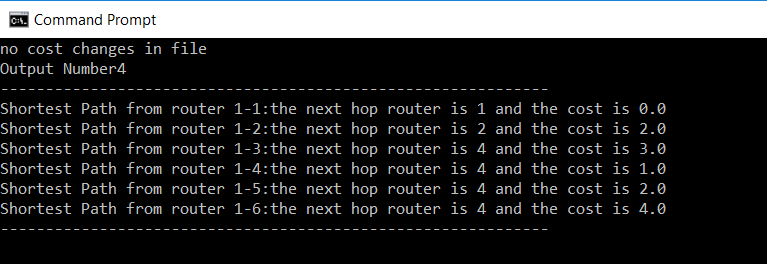




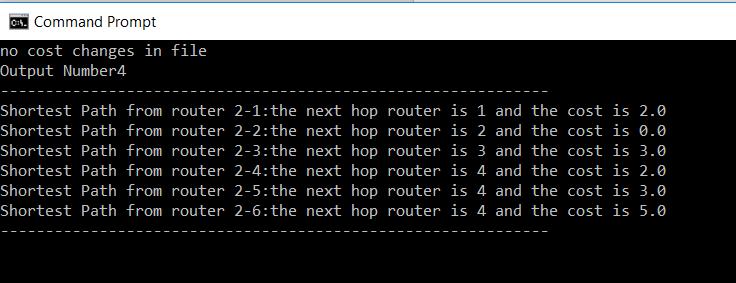


1. Shortest path on each router after some iterations –

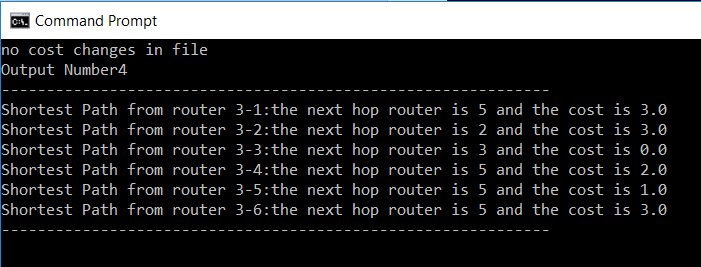
For Router a –



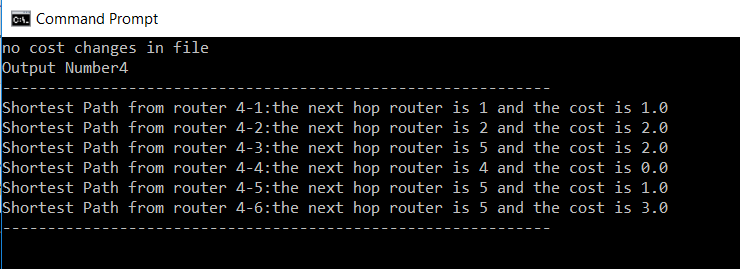
For Router b –



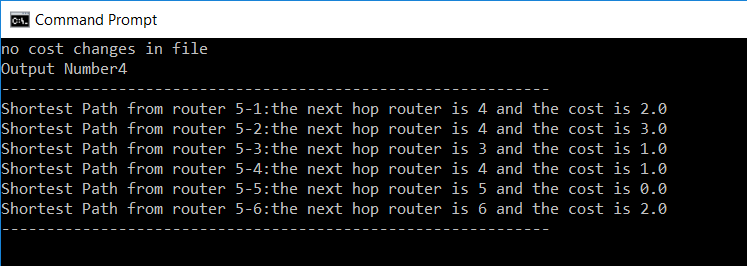
For Router c –



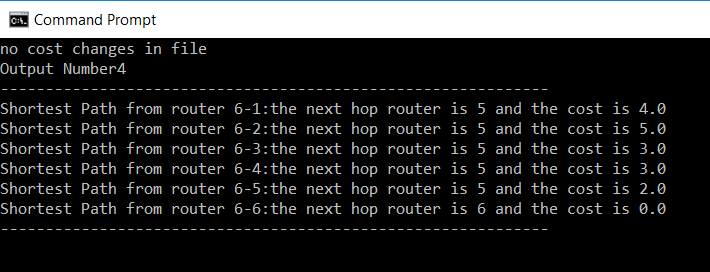
For Router d –



For Router e –

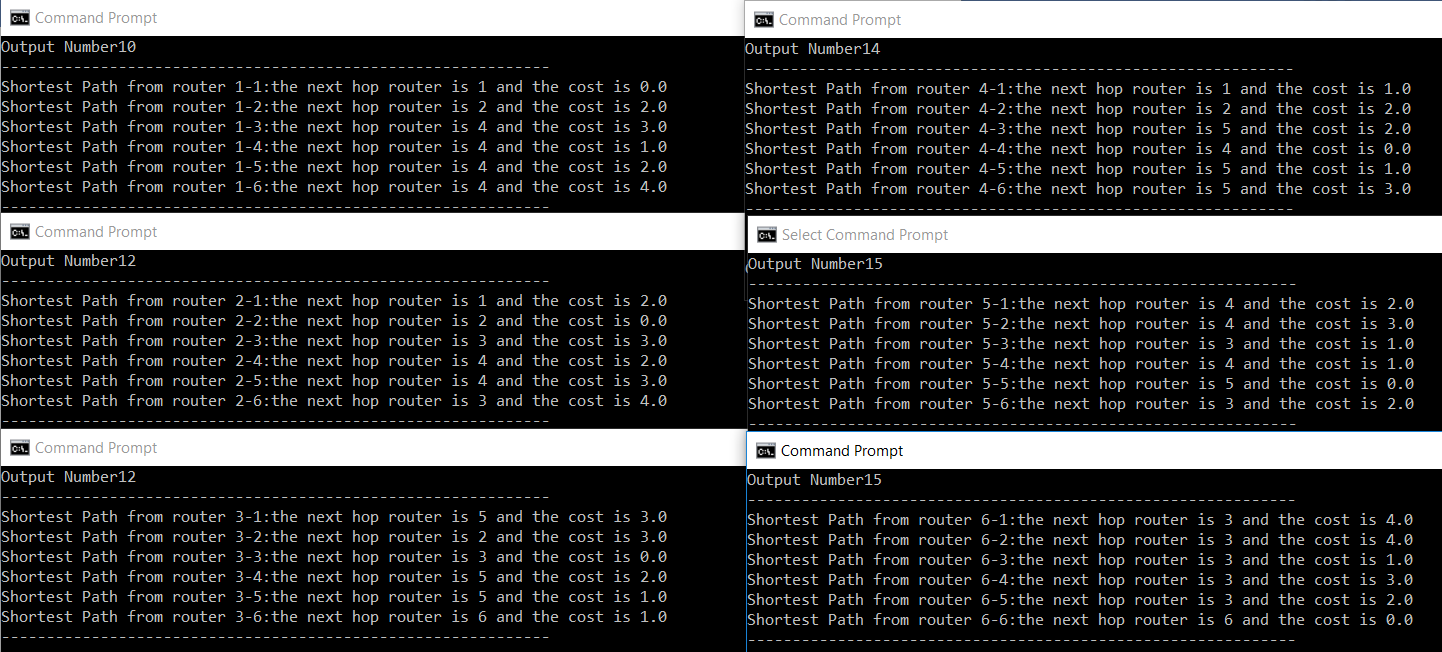


For Router f –

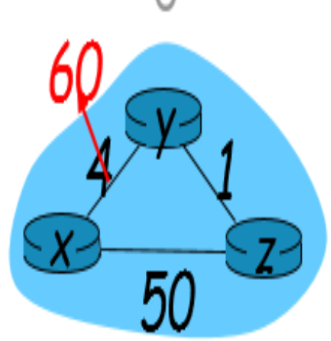


1. Link Cost Handling –

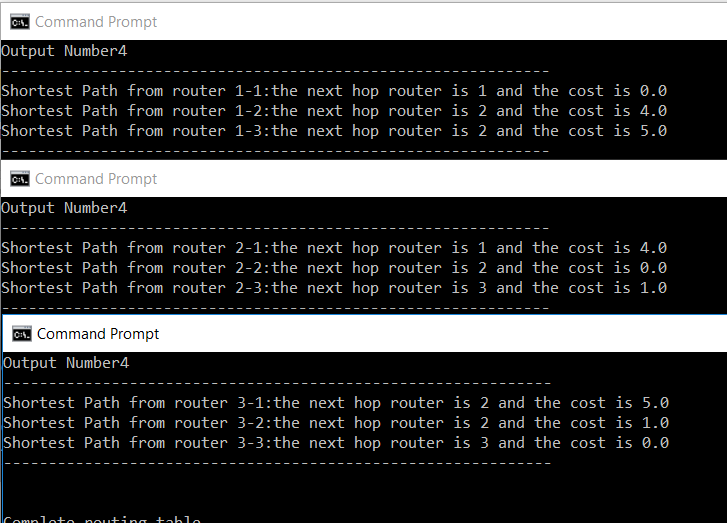
Decreasing link cost of C-F (router3 to router6) from 5 to 1.



1. Handling Recursive Update Problem – For this, consider a new network topology,



Initially, before increasing the cost –



After link cost change to large value 60 –

