Implementing Distance Vector Routing Protocol

By: Hrishikesh Kasar (801053462) and Avadhut Naik (801045233)

The Distance Vector Routing Algorithm is iterative as the process continues until no more information is to be exchanged between the neighbors, asynchronous as it does not require all nodes to operate on lockstep with each other and distributed as a node receives information (routing table) from one or more of its neighbors, performs its calculations (to compute the least cost) and distributes the results of that very calculations to its neighbors. Considering *dx*(*y*) to be the the cost of the least cost path from node x to node y, they are related by the Bellman Ford equation as follows

$$dx(y) = minv\{c(x,v) + dv(y)\}$$

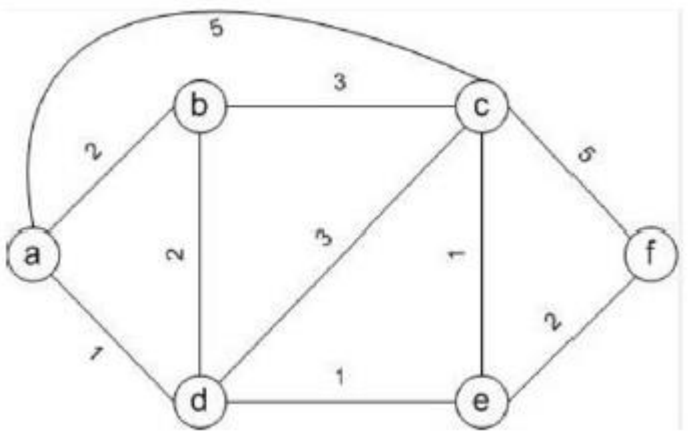
where *minv* is taken over all of neighbors of x, *c(x,v)* is the least cost path from node x to some intermediate node v and *dv(y)* is the cost of routing from node v to node y. Thus, the least cost path from node x to node y is the minimum of *c(x,v)* + *dv(y)*.

With the only information with a node being the least cost path to its neighbors and the information it receives from its neighbors, each node waits from an update in link costs from its neighbors, calculates its distance vector on receiving an update and distributes its new distance vectors to its neighbors. The Distance Vector algorithm thus, ensures routing of packets is undertaken at minimal costs.

In this project, we have implemented a slight variation of distance vector routing algorithm in which a node sends its least costs to all nodes within the network to its neighbors every 15 seconds, irrespective of changes in costs of links. With the input to the program at each node being only its routing table, which is passed off as a command line argument, the node initially prints the same, updating its distance vector for all nodes within the network as its receives information of least costs to all other nodes in the network from its neighbors every 15 seconds, irrespective of link cost changes. In the event of link cost changes occurring through the routing tables of nodes, the same are detected and the least costs are again calculated for all nodes in the network and sent to the node's neighbors. The diagram below indicates the network topology we have considered as a reference for the program.

Implementing Distance Vector Routing Protocol

By: Hrishikesh Kasar (801053462) and Avadhut Naik (801045233)



We assume, for the sake of simplicity, that nodes will have ID's as letters ranging from a to z.

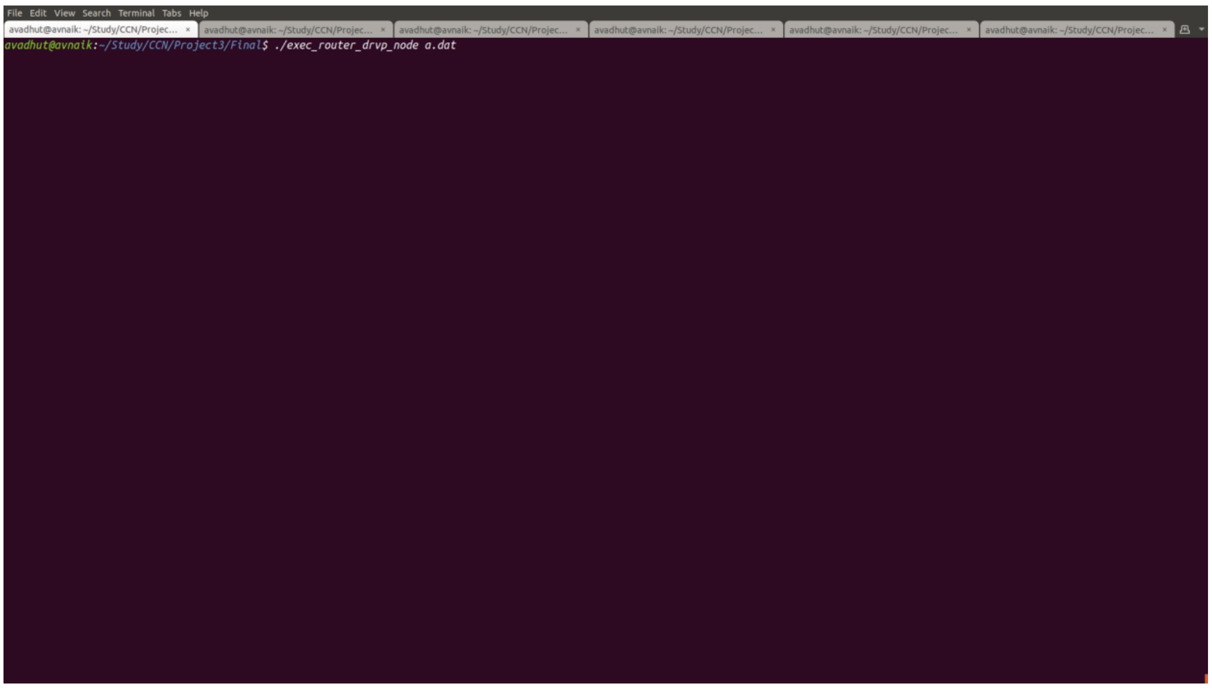
All nodes calculate their port number through the *'int calc_port_no (char node_id)'* function and are assumed to start from 10000 i.e. Node a has the listening port of 10000, Node b has the listening port of 10001 and so on. Utilizing the function to calculate their port numbers, nodes also calculate the port numbers of their neighbors to whom they have to send routing tables after reading the input file.

With the routing tables sent out over UDP protocol as is the case with Distance Vector algorithm, a new thread is created for sending these very packets to neighbors and exits upon its completion in the aftermath of which reception is undertaken in the main process through creation of the listening socket. To avert the scenario of the program getting stuck in case a certain packet is not received, we have implemented a timeout for receiver.

The '*void UpdateMyRoutingTable (void)*' function undertakes the tasks of updating routing tables which is implemented in the event of new cost being less than the older cost. The aforesaid function also averts the scenario of looping. In lieu of poisoned reverse which would require significant changes in sender and receiver code, the receiver is provided with the responsibility of checking the next hop entry. In the event of the next hop entry being itself, the entry is skipped. Testing the same over the terminal, the looping issue has been resolved by the implementation.

The function '*void UpdateNeighborDetails(void)*' handles the file reading the task of updating routing tables in the event of link cost changes. Testing the same over the terminal, the implementation was found to work correctly.

The code for implementing Distance Vector routing protocol has been undertaken in C language, written, implemented and tested over the Desktop version of Linux Operating Systems' 64-bit Ubuntu 17.10.1. Getting through with developing the code for the algorithm as per the requirements of the project and tips provided for the same, the program was run over six different terminal windows of Linux, each representing a node in the network as follows:

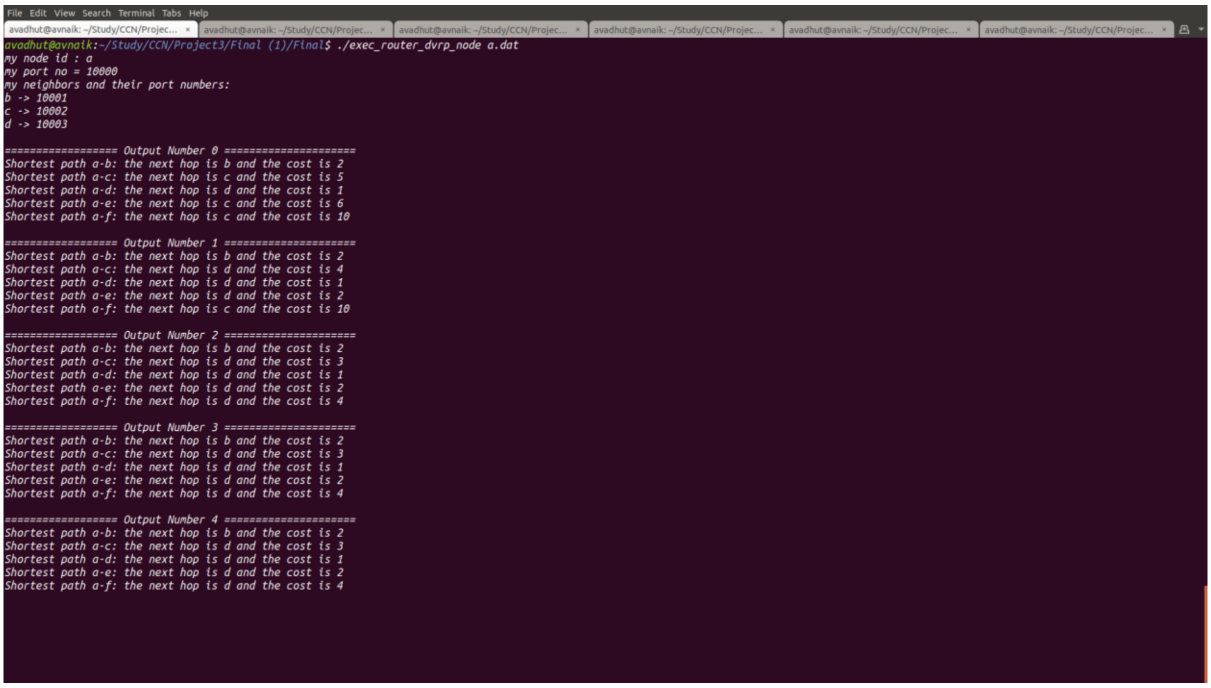

File Edit View Search Terminal Tabs Help
avadhut@avnaik: ~/Study/CCN/Projec...
avadhut@avnaik: ~/Study/CCN/Projec...
avadhut@avnaik: ~/Study/CCN/Projec...
avadhut@avnaik: ~/Study/CCN/Projec...
avadhut@avnaik: ~/Study/CCN/Projec...
avadhut@avnaik: ~/Study/CCN/Projec...

```
avadhut@avnaik:~/Study/CCN/Project3/Final$ ./exec_router_drvp_node a.dat
```

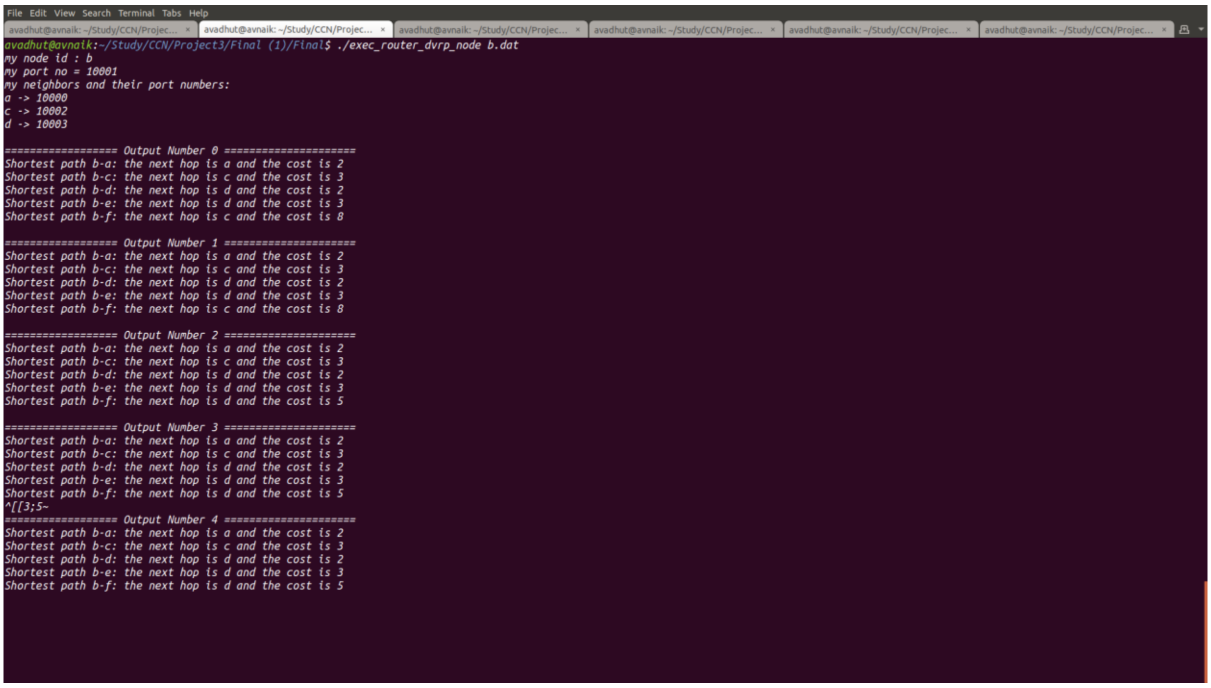
Giving the required input for execution, i.e. the routing table of the respective node, we attempted to start the program, eventually starting all nodes in the network to initiate transfer of routing information to all nodes in the network as per the requirements of the project.

Implementing Distance Vector Routing Protocol

By: Hrishikesh Kasar (801053462) and Avadhut Naik (801045233)



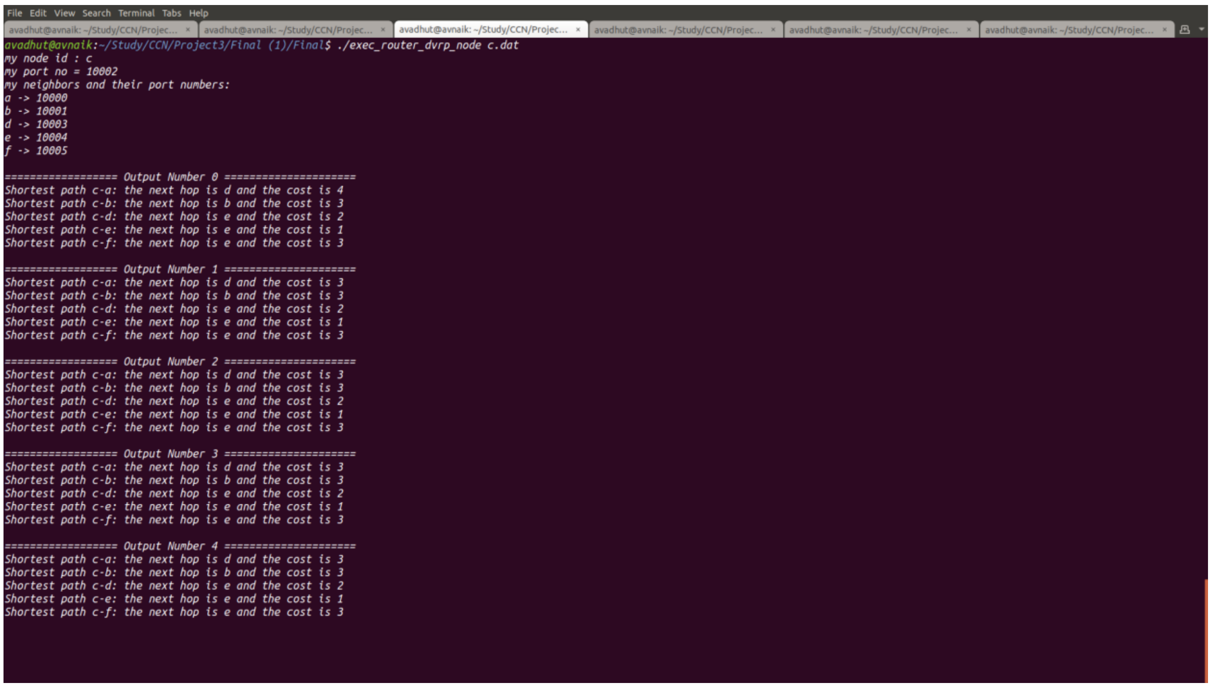
```
avadhut@avnaik:~/Study/CCN/Project3/Final (1)/Final$ ./exec_router_dvrp_node a.dat
my node id : a
my port no = 10000
my neighbors and their port numbers:
b -> 10001
c -> 10002
d -> 10003

================= Output Number 0 ====================
Shortest path a-b: the next hop is b and the cost is 2
Shortest path a-c: the next hop is c and the cost is 5
Shortest path a-d: the next hop is d and the cost is 1
Shortest path a-e: the next hop is c and the cost is 6
Shortest path a-f: the next hop is c and the cost is 10

================= Output Number 1 ====================
Shortest path a-b: the next hop is b and the cost is 2
Shortest path a-c: the next hop is d and the cost is 4
Shortest path a-d: the next hop is d and the cost is 1
Shortest path a-e: the next hop is d and the cost is 2
Shortest path a-f: the next hop is c and the cost is 10

================= Output Number 2 ====================
Shortest path a-b: the next hop is b and the cost is 2
Shortest path a-c: the next hop is d and the cost is 3
Shortest path a-d: the next hop is d and the cost is 1
Shortest path a-e: the next hop is d and the cost is 2
Shortest path a-f: the next hop is d and the cost is 4

================= Output Number 3 ====================
Shortest path a-b: the next hop is b and the cost is 2
Shortest path a-c: the next hop is d and the cost is 3
Shortest path a-d: the next hop is d and the cost is 1
Shortest path a-e: the next hop is d and the cost is 2
Shortest path a-f: the next hop is d and the cost is 4

================= Output Number 4 ====================
Shortest path a-b: the next hop is b and the cost is 2
Shortest path a-c: the next hop is d and the cost is 3
Shortest path a-d: the next hop is d and the cost is 1
Shortest path a-e: the next hop is d and the cost is 2
Shortest path a-f: the next hop is d and the cost is 4
```



```
avadhut@avnaik:~/Study/CCN/Project3/Final (1)/Final$ ./exec_router_dvrp_node b.dat
my node id : b
my port no = 10001
my neighbors and their port numbers:
a -> 10000
c -> 10002
d -> 10003

================= Output Number 0 ====================
Shortest path b-a: the next hop is a and the cost is 2
Shortest path b-c: the next hop is c and the cost is 3
Shortest path b-d: the next hop is d and the cost is 2
Shortest path b-e: the next hop is d and the cost is 3
Shortest path b-f: the next hop is c and the cost is 8

================= Output Number 1 ====================
Shortest path b-a: the next hop is a and the cost is 2
Shortest path b-c: the next hop is c and the cost is 3
Shortest path b-d: the next hop is d and the cost is 2
Shortest path b-e: the next hop is d and the cost is 3
Shortest path b-f: the next hop is c and the cost is 8

================= Output Number 2 ====================
Shortest path b-a: the next hop is a and the cost is 2
Shortest path b-c: the next hop is c and the cost is 3
Shortest path b-d: the next hop is d and the cost is 2
Shortest path b-e: the next hop is d and the cost is 3
Shortest path b-f: the next hop is d and the cost is 5

================= Output Number 3 ====================
Shortest path b-a: the next hop is a and the cost is 2
Shortest path b-c: the next hop is c and the cost is 3
Shortest path b-d: the next hop is d and the cost is 2
Shortest path b-e: the next hop is d and the cost is 3
Shortest path b-f: the next hop is d and the cost is 5
^[[3;5~
================= Output Number 4 ====================
Shortest path b-a: the next hop is a and the cost is 2
Shortest path b-c: the next hop is c and the cost is 3
Shortest path b-d: the next hop is d and the cost is 2
Shortest path b-e: the next hop is d and the cost is 3
Shortest path b-f: the next hop is d and the cost is 5
```

Implementing Distance Vector Routing Protocol

By: Hrishikesh Kasar (801053462) and Avadhut Naik (801045233)

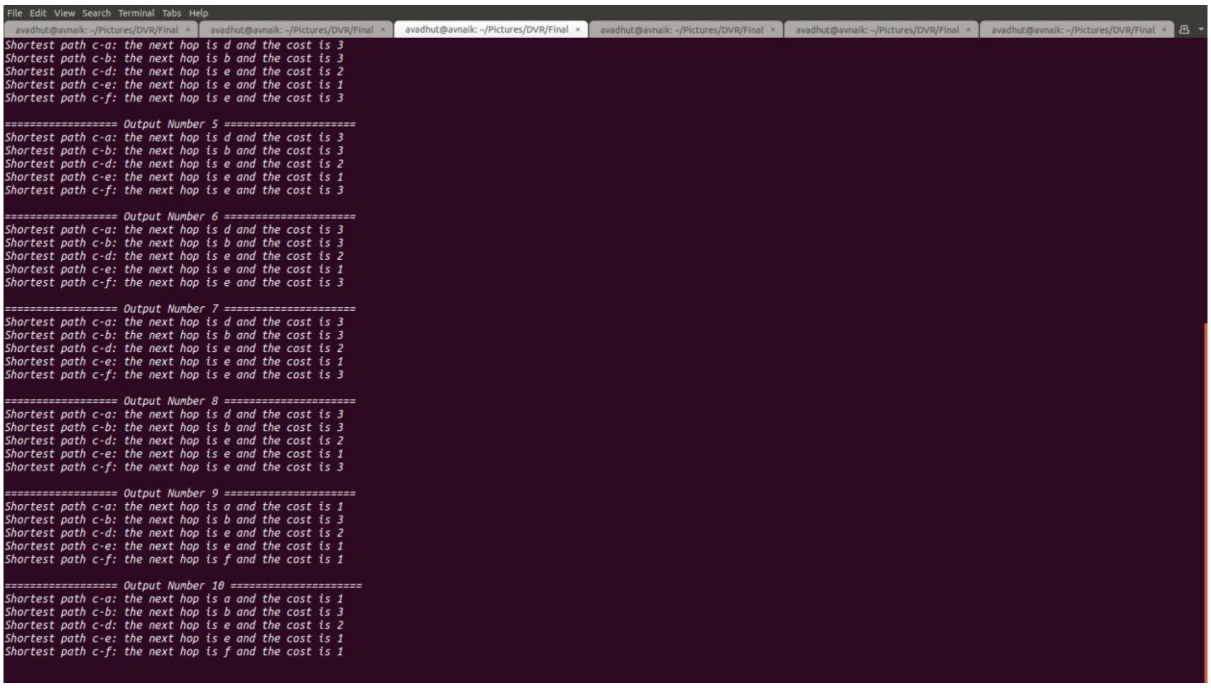

File Edit View Search Terminal Tabs Help
avadhut@avnaik: ~/Study/CCN/Projec...

```
avadhut@avnaik:~/Study/CCN/Project3/Final (1)/Final$ ./exec_router_dvrp_node c.dat
my node id : c
my port no = 10002
my neighbors and their port numbers:
a -> 10000
b -> 10001
d -> 10003
e -> 10004
f -> 10005

================== Output Number 0 =====================
Shortest path c-a: the next hop is d and the cost is 4
Shortest path c-b: the next hop is b and the cost is 3
Shortest path c-d: the next hop is e and the cost is 2
Shortest path c-e: the next hop is e and the cost is 1
Shortest path c-f: the next hop is e and the cost is 3

================== Output Number 1 =====================
Shortest path c-a: the next hop is d and the cost is 3
Shortest path c-b: the next hop is b and the cost is 3
Shortest path c-d: the next hop is e and the cost is 2
Shortest path c-e: the next hop is e and the cost is 1
Shortest path c-f: the next hop is e and the cost is 3

================== Output Number 2 =====================
Shortest path c-a: the next hop is d and the cost is 3
Shortest path c-b: the next hop is b and the cost is 3
Shortest path c-d: the next hop is e and the cost is 2
Shortest path c-e: the next hop is e and the cost is 1
Shortest path c-f: the next hop is e and the cost is 3

================== Output Number 3 =====================
Shortest path c-a: the next hop is d and the cost is 3
Shortest path c-b: the next hop is b and the cost is 3
Shortest path c-d: the next hop is e and the cost is 2
Shortest path c-e: the next hop is e and the cost is 1
Shortest path c-f: the next hop is e and the cost is 3

================== Output Number 4 =====================
Shortest path c-a: the next hop is d and the cost is 3
Shortest path c-b: the next hop is b and the cost is 3
Shortest path c-d: the next hop is e and the cost is 2
Shortest path c-e: the next hop is e and the cost is 1
Shortest path c-f: the next hop is e and the cost is 3
```

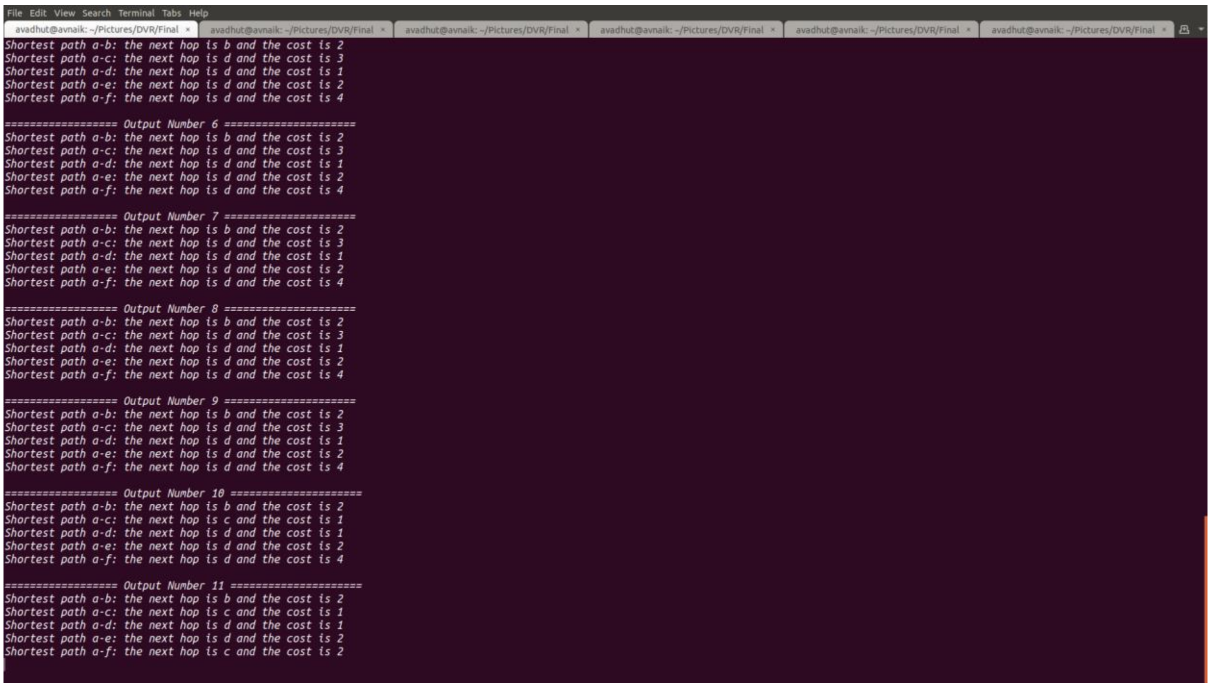
As is evident from the three images above, each node upon being started prints its ID, listening port number and the listening port numbers of its neighbors in the aftermath of which it prints its own routing table along with costs for other nodes in the network which are not its neighbors and costs for whom are obtained through its neighbors. Upon successive iterations however, as each node receives from each of its neighbors its least costs and next hop for all other nodes in the network, it computes its least cost for all nodes according the information it receives from its neighbors.

Implementing Distance Vector Routing Protocol

By: Hrishikesh Kasar (801053462) and Avadhut Naik (801045233)


File Edit View Search Terminal Tabs Help
avadhut@avnaik: ~/Pictures/DVR/Final

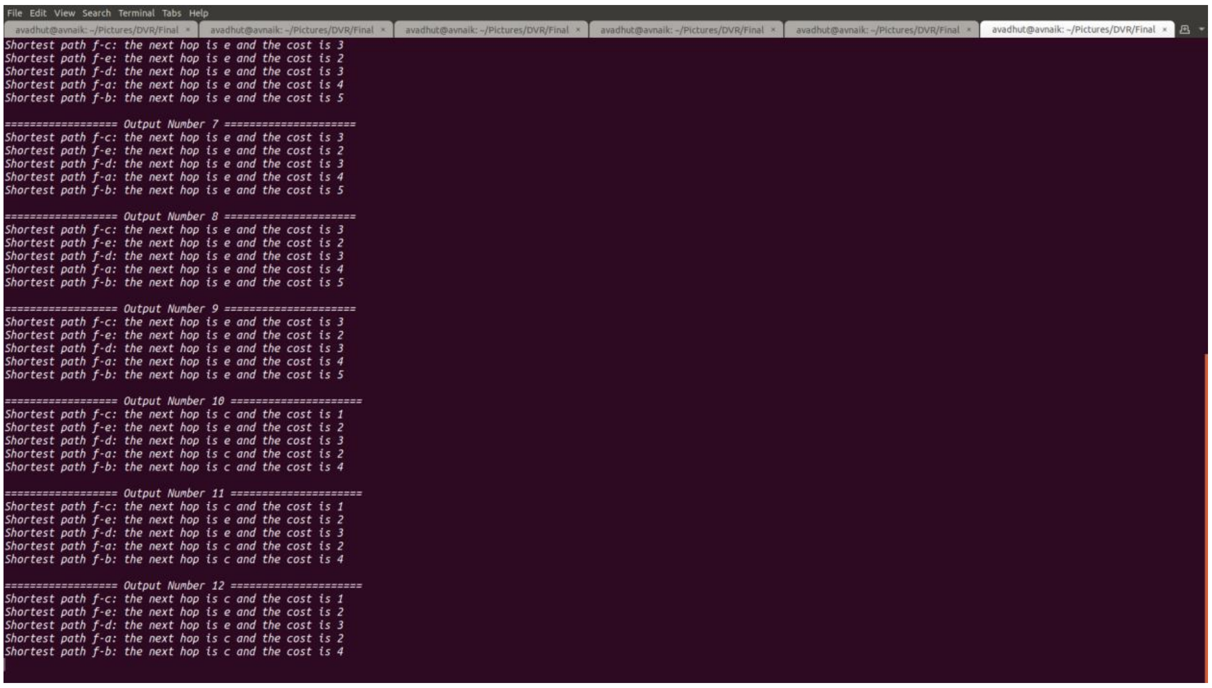
```
Shortest path c-a: the next hop is d and the cost is 3
Shortest path c-b: the next hop is b and the cost is 3
Shortest path c-d: the next hop is e and the cost is 2
Shortest path c-e: the next hop is e and the cost is 1
Shortest path c-f: the next hop is e and the cost is 3

================= Output Number 5 ====================
Shortest path c-a: the next hop is d and the cost is 3
Shortest path c-b: the next hop is b and the cost is 3
Shortest path c-d: the next hop is e and the cost is 2
Shortest path c-e: the next hop is e and the cost is 1
Shortest path c-f: the next hop is e and the cost is 3

================= Output Number 6 ====================
Shortest path c-a: the next hop is d and the cost is 3
Shortest path c-b: the next hop is b and the cost is 3
Shortest path c-d: the next hop is e and the cost is 2
Shortest path c-e: the next hop is e and the cost is 1
Shortest path c-f: the next hop is e and the cost is 3

================= Output Number 7 ====================
Shortest path c-a: the next hop is d and the cost is 3
Shortest path c-b: the next hop is b and the cost is 3
Shortest path c-d: the next hop is e and the cost is 2
Shortest path c-e: the next hop is e and the cost is 1
Shortest path c-f: the next hop is e and the cost is 3

================= Output Number 8 ====================
Shortest path c-a: the next hop is d and the cost is 3
Shortest path c-b: the next hop is b and the cost is 3
Shortest path c-d: the next hop is e and the cost is 2
Shortest path c-e: the next hop is e and the cost is 1
Shortest path c-f: the next hop is e and the cost is 3

================= Output Number 9 ====================
Shortest path c-a: the next hop is a and the cost is 1
Shortest path c-b: the next hop is b and the cost is 3
Shortest path c-d: the next hop is e and the cost is 2
Shortest path c-e: the next hop is e and the cost is 1
Shortest path c-f: the next hop is f and the cost is 1

================= Output Number 10 ====================
Shortest path c-a: the next hop is a and the cost is 1
Shortest path c-b: the next hop is b and the cost is 3
Shortest path c-d: the next hop is e and the cost is 2
Shortest path c-e: the next hop is e and the cost is 1
Shortest path c-f: the next hop is f and the cost is 1
```


File Edit View Search Terminal Tabs Help
avadhut@avnaik: ~/Pictures/DVR/Final

```
Shortest path a-b: the next hop is b and the cost is 2
Shortest path a-c: the next hop is d and the cost is 3
Shortest path a-d: the next hop is d and the cost is 1
Shortest path a-e: the next hop is d and the cost is 2
Shortest path a-f: the next hop is d and the cost is 4

================= Output Number 6 ====================
Shortest path a-b: the next hop is b and the cost is 2
Shortest path a-c: the next hop is d and the cost is 3
Shortest path a-d: the next hop is d and the cost is 1
Shortest path a-e: the next hop is d and the cost is 2
Shortest path a-f: the next hop is d and the cost is 4

================= Output Number 7 ====================
Shortest path a-b: the next hop is b and the cost is 2
Shortest path a-c: the next hop is d and the cost is 3
Shortest path a-d: the next hop is d and the cost is 1
Shortest path a-e: the next hop is d and the cost is 2
Shortest path a-f: the next hop is d and the cost is 4

================= Output Number 8 ====================
Shortest path a-b: the next hop is b and the cost is 2
Shortest path a-c: the next hop is d and the cost is 3
Shortest path a-d: the next hop is d and the cost is 1
Shortest path a-e: the next hop is d and the cost is 2
Shortest path a-f: the next hop is d and the cost is 4

================= Output Number 9 ====================
Shortest path a-b: the next hop is b and the cost is 2
Shortest path a-c: the next hop is d and the cost is 3
Shortest path a-d: the next hop is d and the cost is 1
Shortest path a-e: the next hop is d and the cost is 2
Shortest path a-f: the next hop is d and the cost is 4

================= Output Number 10 ====================
Shortest path a-b: the next hop is b and the cost is 2
Shortest path a-c: the next hop is c and the cost is 1
Shortest path a-d: the next hop is d and the cost is 1
Shortest path a-e: the next hop is d and the cost is 2
Shortest path a-f: the next hop is d and the cost is 4

================= Output Number 11 ====================
Shortest path a-b: the next hop is b and the cost is 2
Shortest path a-c: the next hop is c and the cost is 1
Shortest path a-d: the next hop is d and the cost is 1
Shortest path a-e: the next hop is d and the cost is 2
Shortest path a-f: the next hop is c and the cost is 2
```

Implementing Distance Vector Routing Protocol

By: Hrishikesh Kasar (801053462) and Avadhut Naik (801045233)



File Edit View Search Terminal Tabs Help
avadhut@avnaik: ~/Pictures/DVR/Final
avadhut@avnaik: ~/Pictures/DVR/Final
avadhut@avnaik: ~/Pictures/DVR/Final
avadhut@avnaik: ~/Pictures/DVR/Final
avadhut@avnaik: ~/Pictures/DVR/Final
avadhut@avnaik: ~/Pictures/DVR/Final

```
Shortest path f-c: the next hop is e and the cost is 3
Shortest path f-e: the next hop is e and the cost is 2
Shortest path f-d: the next hop is e and the cost is 3
Shortest path f-a: the next hop is e and the cost is 4
Shortest path f-b: the next hop is e and the cost is 5

================= Output Number 7 ====================
Shortest path f-c: the next hop is e and the cost is 3
Shortest path f-e: the next hop is e and the cost is 2
Shortest path f-d: the next hop is e and the cost is 3
Shortest path f-a: the next hop is e and the cost is 4
Shortest path f-b: the next hop is e and the cost is 5

================= Output Number 8 ====================
Shortest path f-c: the next hop is e and the cost is 3
Shortest path f-e: the next hop is e and the cost is 2
Shortest path f-d: the next hop is e and the cost is 3
Shortest path f-a: the next hop is e and the cost is 4
Shortest path f-b: the next hop is e and the cost is 5

================= Output Number 9 ====================
Shortest path f-c: the next hop is e and the cost is 3
Shortest path f-e: the next hop is e and the cost is 2
Shortest path f-d: the next hop is e and the cost is 3
Shortest path f-a: the next hop is e and the cost is 4
Shortest path f-b: the next hop is e and the cost is 5

================= Output Number 10 ====================
Shortest path f-c: the next hop is c and the cost is 1
Shortest path f-e: the next hop is e and the cost is 2
Shortest path f-d: the next hop is e and the cost is 3
Shortest path f-a: the next hop is c and the cost is 2
Shortest path f-b: the next hop is c and the cost is 4

================= Output Number 11 ====================
Shortest path f-c: the next hop is c and the cost is 1
Shortest path f-e: the next hop is e and the cost is 2
Shortest path f-d: the next hop is e and the cost is 3
Shortest path f-a: the next hop is c and the cost is 2
Shortest path f-b: the next hop is c and the cost is 4

================= Output Number 12 ====================
Shortest path f-c: the next hop is c and the cost is 1
Shortest path f-e: the next hop is e and the cost is 2
Shortest path f-d: the next hop is e and the cost is 3
Shortest path f-a: the next hop is c and the cost is 2
Shortest path f-b: the next hop is c and the cost is 4
```

Upon a link change being initiated through the routing table of c, in which costs of links from node c to node a and node c to node f were changed from 5 to 1, the same was reflected in the information sent by node c to its neighbors and printed over the terminal window. The cost change also reflected in the information sent out by nodes a and f and printed over their respective terminal windows. The program, thus robustly supports link cost changes occurring through routing table of any node in the network.

Implementing Distance Vector Routing Protocol

By: Hrishikesh Kasar (801053462) and Avadhut Naik (801045233)

Instructions for Execution of Program:

1) Heading to the directory where program is stored from the terminal of Linux, execute the make file for the program, to compile the program and open six terminal windows, each for a node in the network.

```
$ make
```

2) Execute the exec_router_drvp_node file and provide the appropriate routing table file (.dat) to start the node in a terminal window.

```
$ ./exec_router_drvp_node a.dat OR $ ./exec_router_drvp_node b.dat and so on.
```

3) To initiate a link cost change, head into the routing table of the node and make the desired changes through gedit, save the file and close it.

```
$ gedit c.dat OR gedit f.dat and so on.
```

4) Observe the effects of link cost changes in terms of changes in least costs of node over the appropriate terminal window.

Implementing Distance Vector Routing Protocol

By: Hrishikesh Kasar (801053462) and Avadhut Naik (801045233)

References: -

1) James F. Kurose and Keith W. Ross COMPUTER NETWORKING: A Top-Down Approach