Name - Aditya Singla
Email - a28@buffalo.edu

CSE – 489 – MODERN NETWORKING CONCEPTS
PROGRAMMING ASSIGNMENT 2
ROUTING PROTOCOLS

Implementation Details

The assignment emulates a distance routing vector protocol. For this servers were used for testing namely

1) Timberlake
2) Metallica
3) Beatles
4) Nickelback
5) Dragonforce

Each server is provided with a topology file at the start. The topology file contains the number of servers, number of neighbors, server ip-port list and routing information. Following are the execution of the program:

Step 1 - At the start, the program reads the topology file. The server-ip list is saved in the system with the routing information.

Step 2 - Data Structure for saving routing information

2 dimensional array (**Line 34**). Thus the data for routing information of server with id 1 is saved as:

	via1	via2	via3	via4	via5
Server 1	0	0	0	0	0
Server 2	0	3	3	4	9999
Server 3	0	4	4	5	9999
Server 4	0	10	4	6	9999
Server 5	0	5	6	7	9999

The array was created as an array of structures. **The cost 9999 is considered as infinity.** The above table shows that server 1 is not connected to server 5 as the cost via server 5 is 9999 in all cases.

Step 3 - Regular routing table updated are sent to neighbor servers via a specified message format. The message format was defined as:

<table>
<tr><th>Number of update fields(2 bytes)</th><th>Server port(2 bytes)</th></tr>
<tr><td colspan="2">Source server ip address (4 bytes)</td></tr>
<tr><td colspan="2">Routing update server ip (4 bytes)</td></tr>
<tr><td>Port (2 bytes)</td><td>Blank (2 bytes)</td></tr>
<tr><td>Server id (2 bytes)</td><td>Cost (2 bytes)</td></tr>
</table>

A function was used to create the above mentioned message format (**declared on line 564**).

Name - Aditya Singla
Email - a28@buffalo.edu

Step 4 - The bellman-ford distance vector algorithm was run after receiving a routing update. Any update into the routing table was sent to the neighbors in periodic intervals defined by the user.

Step 5 – The steps 3 and 4 are repeated after time interval defined by the user. The user can update the link cost between to servers using "UPDATE" command. This would run the bellman-ford algorithm again thereby recalculating the costs.

Following commands were implemented in the system:

1) DISPLAY – Display the routing information for a server.
2) UPDATE <src> <dest> <cost> – Update the link cost to a particular server.
3) STEP – Sends the routing information immediately to the neighbors.
4) PACKETS – Displays the packets received by the server since the last time the command was executed.
5) DISABLE <server-id> - Disable a neighbor. No packets will be sent to the neighbor or received by the server.
6) CRASH – Crashes the server. No command are executed on the server. No messages are send or received by the server.

To implement each of the following command, a function was defined in the proj2_a28.c file:

1) Display () – Line 363 – The function displays the minimum cost to each server.
2) Update () – Line 408 – The function updates the data structure that's stores link vectors. After updating the function runs the bellman-ford algorithm again, there by calculating new costs.
3) Disable () – Line 384 – The function disables a link from one server to other. After calling disable on a server, no messages are sent or received from the server.
4) runAlgorithm () – Implements bellman-ford algorithm.

On updating cost to infinity or disabling the server, the "count to infinity" scenario will be observed. The scenario was observed for the following topologies:

1) Ring Topology
2) Line Topology
3) Star Topology