Donald Cochran - Diego Martinez

Jason Zhao – Yara Contijoch

Data Communication and Networking

4/10/2019

**Project-1 Lab Report (BONUS)**

Learning Objectives

-Set up a routing network using 4 nodes

-Establish communication using the 4 nodes

-Create and manipulate configuration files for references

-Capture total route cost to calculate next hop

-Use a representation of Bellman-Ford’s algorithm to handle distance vector changes

Synopsis

Our team updated the Update Route Cost command to collect the total route cost to the selected nodes direct neighbors and store it in a newly added node structure variable called totalCost. This variable will be used to compare these paths to other nodes. The “next hop” or where to send the message will be calculated by which total cost path is the least costing.

Our next task is to utilize the Update Route Cost to have the user select the destination node to be changed. (table.node.name) and that would select the corelating cost to be changed. (table.node.cost). Then the new table will be printed with the updated cost and the table will be sent into a buffer to the corresponding node.config file. In addition, the message needs to be routed through the SendTable command Since A is not connected directly to D and C is not directly connected to B.

In the listening stage located in a while loop, the program will receive the sent table as (routing\_table recv) and the goal is to compare that table to the current table to check if node cost needs to be changed. We created different handling functions that check to make sure we received the correct receiving table, to update the cost of a changed node by comparing total cost paths to extended paths, and finally to calculate the cost of the other paths to establish if the nextHop needs to be changed for a more efficient path.

User-Story

The user can perform the following:

-Send a node table to other neighbors

- Update route cost of certain nodes to manipulate a path

-Route a table through multiple multiple tables

Difficulties

The SendTable command would send the same table twice to D since it is being delivered by B and C. To fix this, every recv table is stored in another table and they are both being compared to make sure they are not the same table so the funcions do not do the same calculations twice. We could not get this application fully working because the calculations are too extensive with the methods we used for this project.