Project Members:

Sharif Ahmad : 139743 Asad Nawaz : 122323

As soon as the first router gets up and gets its neighbor data then it continuously checks if its neighbors have got on the network if they have then it sends them its data else it does not but as soon as they get on the network it sets a variable saying the router is up and does not check anymore. If a router goes down it also notes it saying the router has gone down.

For distance vector sharing between the sockets I use python pickle module for serializing them into a byte stream and then de-serialize them.

Because UDP is connectionless protocol therefore we don’t really know if a port has gone down therefore I use some connection check packets every 2 to 3 seconds to know if a port has gone down if it does not reply to the packets. As soon as one know a port has gone down it share this information with its neighbors which they share further away.

Interface for link cost change will work as follows:

It will save the initial file and every 30 sec to 1 min will compare the file to initial file and if there is a change will write that change to the object (neighbors data). If there is no change the neighbors data will remain as it is. After file change and saving neighbors data the initial file will be rewritten to be the new file. The initial file will always remain in memory.

Working of interface\_thread:

The function waits until the file is loaded initially and then it start comparing the respective file of the socket with the initial copy with 30 sec between each comparison if the file is changed then it finds the changed location and loads the new data

We receive either message or distance vector information but the main issue now is that messages come out of order that is you are waiting for response to “is\_alive” you may receive distance vector which we need to handle

If all of our neighbors goes down we should stop our threads and gracefully show a message saying “I’m alone”

We need to handle “is\_alive”, “yes” and “distance\_vec” messages and also look for timeout or OSerrors.