

Telecommunication II

Assignment 2 Report

Introduction

The group name is The 5 and group members are Wessam Gholam, Zhuoyu Jin, Shahraiz Rizwan, Lingfeng Yue and Mariah Felicci Durias. The main idea of this assignment is sending messages from one smartphone to another smartphone or laptop directly by developing two routing approaches. Those two approaches are based on Link State Routing and Distance Vector Routing respectively. The smartphones used in the communication can only communicate with the closest laptop and each of the laptop acts as a router and smartphone acts as ending points in the whole scenario. Firstly, the smartphones get connected with a laptop and laptops then create their own individual routing tables in following steps. Each router establish its routing table once and after establishing routing tables, communication would happen. Moreover, only simple messages should be sent between two ending points and message should be sent through the shortest path in the network of the all routers.

Link-State Routing:

Each node has the way to reach its directly connected neighbors, and the totality of this knowledge is disseminated to each of them, so each node has enough knowledge of their network, which then they are able to decide the correct routes to any of the destinations. One process makes sure that all nodes participate is called reliable flooding, it gets a copy of the link-state information from all the other nodes. The idea is for a node to send its link-state information out to all of its directly connected links, in addition, each node that receives this information would forward it out to all of its links until the information has reached all nodes in the network.

For example:

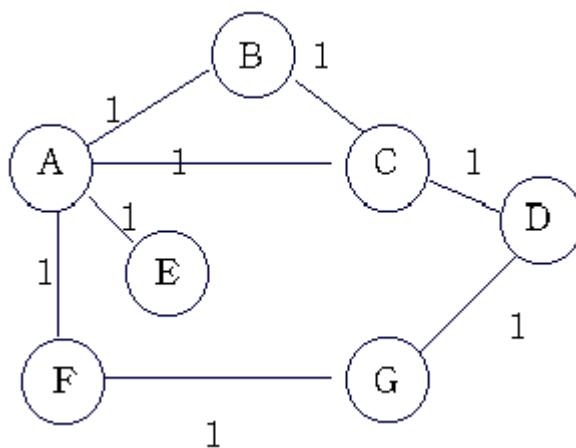
When a node A receives a copy of a link-state packet that originated at some other node

B, it would check if it has already stored a copy of a link-state packet from B. If not, it stores the link-state packet; else it compares the sequence numbers. If the new link – state packet has a larger sequence number, then it is supposed to be more recent, then the new link-state packet is stored by replacing the old one, the new one is then forward on to all neighbors of A except the one from which the link-state packet was just received.

Distance-Vector Routing:

Each node constructs a one-D array which contains the distances or costs to all the other nodes and distributes that vector to its immediate neighbors. Each node knows the cost of the link to each of its directly connected neighbors.

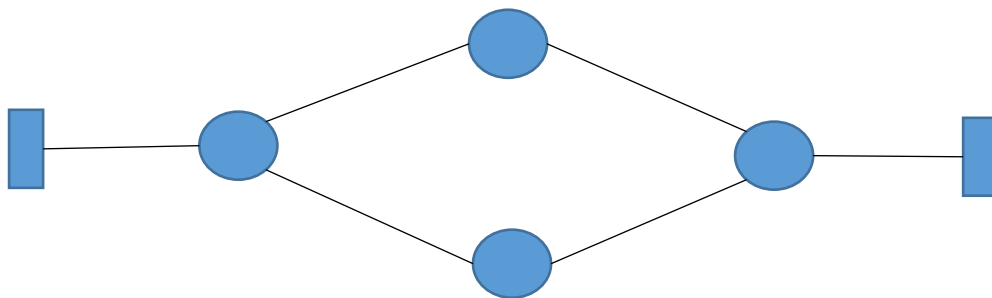
For example:



In this network, first of all, every node sends a message to its directly connected neighbors which contains the list of distance between the node itself and the neighbors. However, if any recipients for the information from A finds that A is the path shorter than the one they know about, they would update their lists for the new distance of the path and send packets to that destination through A. For example, if B knows from A that E can be reached at the cost of 1, and to get A is the cost of 1 from B, so it adds these to get the cost of reaching E by means of A. In record, B can reach E by cost of 2 through A). After each node updates their list of cost with its directly

connected neighbors, all nodes would know the least cost path to all the other nodes. When nodes update their lists, they need to track which node gives them the path they used, so that the forwarding table can be created.

Link-State Routing

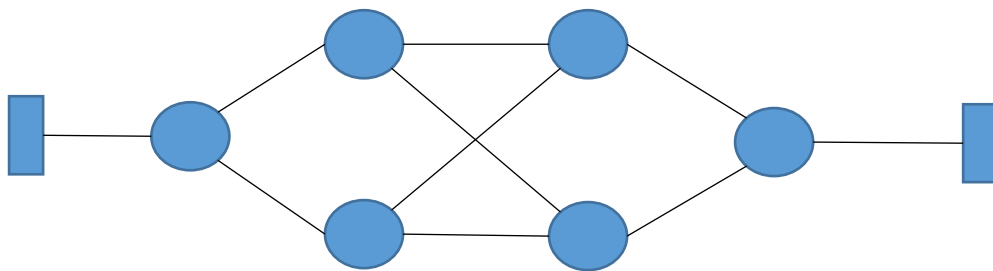


Five classes are created in this approach: Connection (main class), Node, Router, RoutingTable and Smartphone. For the Smartphone class, it can be added to a few numbers of smartphones used as ending points by the function addSmartphone by finding its address. DatagramSocket is a class to create the ending points. Once the sign that “String to send” is out on the terminal, the user then could input the message that they would like to send another ending point of the network.

While routers are used to communicate in the network each by each, and they can be created using the new DatagramSocket and also can be added numbers of listeners. To fetch the routing table, finding the shorter path by comparing the distance between two routers. Routing tables are built up for the first router at the start then the knowledge would be sent out to the neighbors and the routing table is updated time by time, since that the new routing table can be received by the neighbors always, which the totality of the information can always been updated through router by router and

tables will also record the cost or the distance between neighbors. Connections can be implemented by giving different distances between routers and when the program is run it will automatically find the first router and create the routing table for it and pass it to the next neighbor with updated information.

Distance-Vector Routing



There are six classes in this approach: Connection (main class), ForwardTable, Node, Router, RoutingTable and Smartphone, each one of them plays a different role in this routing approach.

As the ending point of the whole network, Smartphone class is using DatagramSocket class, which implements a UDP socket for sending and receiving DatagramPacket. A DatagramSocket object can be used for both endpoints of a connection for a packet delivery service. First of all, creating a smart phone using terminal and give it a name, then every time when adding a smart phone, there is a function by giving the address of the smartphone and also a function to get the name of the smartphone wanted to add. By starting the network, a function 'start' gets the data by byte and the data will get into the terminal when "String to send" is coming out from the terminal and user input gets into the terminal.

Router class is all the laptops that used to communicate between two ending points of smartphones. Firstly, setting all the routers used in the network by creating new DatagramSocket., while addListener is allowed to add a number of listeners. Function setRouterManuel is constructing to add forward tables to each of the router, and also same as the smartphone class, there is a function to get the names of routers as well. For the forward table, comparing the distance between two routers and find the shorter path of two routers and then update the table. By updating the table, it is important to find the neighbors and distance between neighbors and pass the information to the next router. While the routing table and messages are required to be serialized to get it fit in. To pass the whole complete routing table to other routers, tracking and updating the tables when sending them to each one of the router. Moreover, to create the routing table, a 2-D array is needed to have its lines and columns.

To the main Connection class, difference distances among routers decide the way they connect to each other and different paths that can be chosen to reach the smartphone. Once it starts, it will automatically find the first router and create a routing table for it and its neighbors and send the routing table to its neighbors. Then the neighbors find the way to reach other routers by the table they receive, also the port will be recorded for the final path. By this way, it is easy to find the shortest path of two ending points in the end.

Conclusion

All in all, the difference between two approaches is that the routing tables sent out are in various ways. For link-state routing, the routing table that it needs is created from the first router at the start and it gives the table for the neighbors and once they find the shortest path, the table get reach to the last router, so that the last router know everything about the network, so it can update the table and send it back to each one of the router. However, for distance-vector routing, it gets the table form the start as

well, the difference is that each one of the neighbor knows the list of the cost from itself and all the other routers and also record the shortest way to get to the router which is not the neighbor of itself, since that it will find out the shortest way to get to the last router.