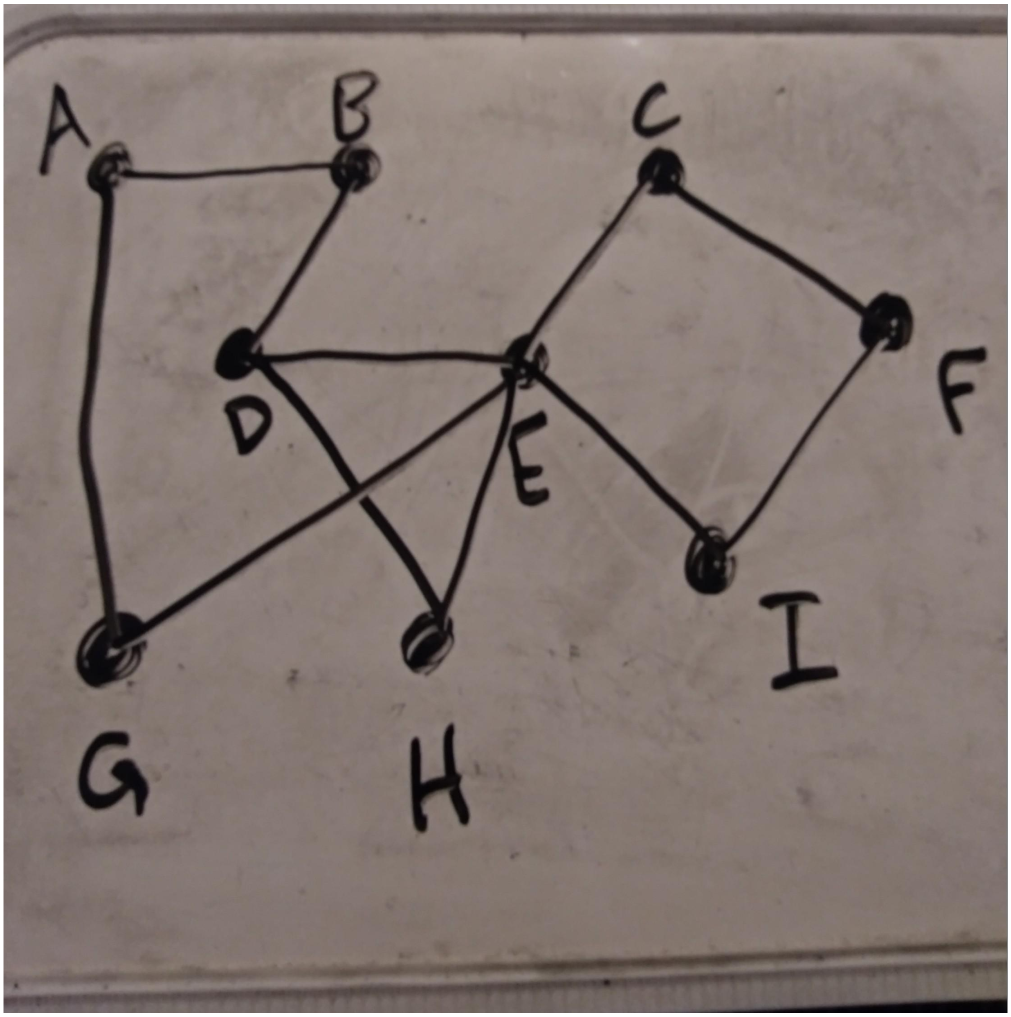
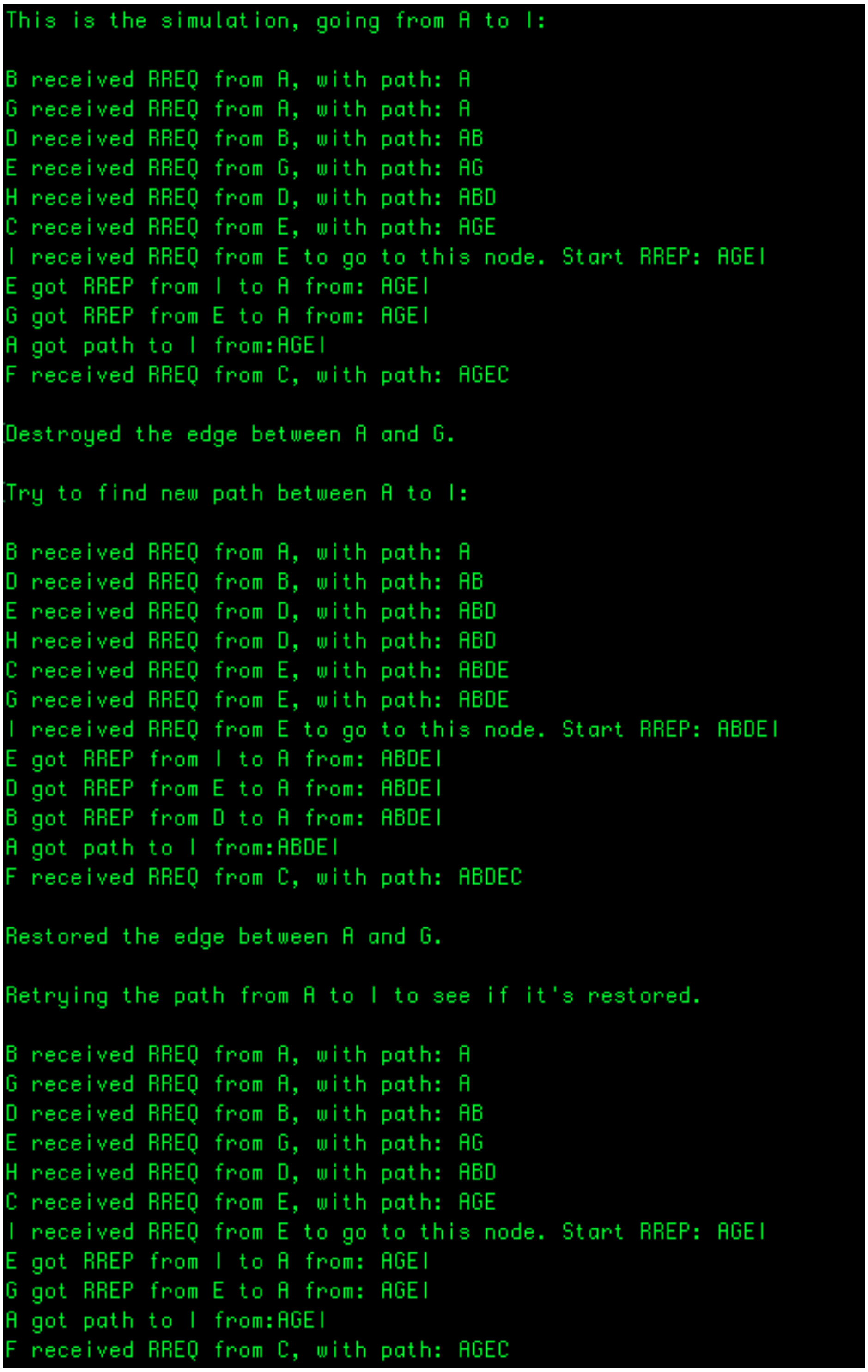
CPE 400 Course Project

Drew Morri & Louis Doherty

To start off we need to understand nodes in a mobile ad hoc network. The nodes are communicating with each other by forwarding data between the other connected nodes. Nodes can be connected directly but nodes can also be considered connected if they share a mutual directly connected node. The data is moving from node to node which is what is making up the network. It is the nodes job to figure out how to transport this data to other nodes. The nodes can be seen as devices and they all need to be able to transfer data to each other which makes up the network. A router-centric network is different. End-hosts are connected to routers and these control the data passing. Wireless networking wireless hosts are connected to a network through a base station and with mobile ad hoc networks are known as rapidly changing network topology. We accommodate for this with a reactive routing protocol called Dynamic Source Routing Protocol. Dynamic Source Routing Protocol, or DSR, is used to facilitate data exchanges between nodes in a mobile ad hoc network. Essentially what this does is it allows a device to come into a new device through mutual devices. This might sound repetitive but it is like getting a person's phone number from a mutual friend. If that mutual friend doesn’t happen to have their phone number they ask another mutual phone number, and so on and so on until you make that connection. A path will eventually be established because if this process is repeated eventually the two people, or devices, will eventually connect. This is a recursive process because each node will continue to ask their neighboring nodes if they are connected to the requested node. Once a connection is made between two nodes all the connecting nodes are not needed to make a connection anymore because a connection has been established. To implement DSR you simulate how paths are determined. A route request must propagate throughout a network and when the requested node receives a request a route reply will travel back along that same traversed path. Nodes also keep track of requests to prevent redundant requests. In the program we have a fixed network. This demonstrates the basic functions of DSR. In this example a request is sent between two nodes and the propagation of the request gets outputted to the command line once a node receives that request. If a path is found the reply is also outputted. A simple time out occurs if no path is found.

This is the graph used for testing purposes:





```
This is the simulation, going from A to I:

B received RREQ from A, with path: A
G received RREQ from A, with path: A
D received RREQ from B, with path: AB
E received RREQ from G, with path: AG
H received RREQ from D, with path: ABD
C received RREQ from E, with path: AGE
I received RREQ from E to go to this node. Start RREP: AGEI
E got RREP from I to A from: AGEI
G got RREP from E to A from: AGEI
A got path to I from:AGEI
F received RREQ from C, with path: AGEC

Destroyed the edge between A and G.

Try to find new path between A to I:

B received RREQ from A, with path: A
D received RREQ from B, with path: AB
E received RREQ from D, with path: ABD
H received RREQ from D, with path: ABD
C received RREQ from E, with path: ABDE
G received RREQ from E, with path: ABDE
I received RREQ from E to go to this node. Start RREP: ABDEI
E got RREP from I to A from: ABDEI
D got RREP from E to A from: ABDEI
B got RREP from D to A from: ABDEI
A got path to I from:ABDEI
F received RREQ from C, with path: ABDEC

Restored the edge between A and G.

Retrying the path from A to I to see if it's restored.

B received RREQ from A, with path: A
G received RREQ from A, with path: A
D received RREQ from B, with path: AB
E received RREQ from G, with path: AG
H received RREQ from D, with path: ABD
C received RREQ from E, with path: AGE
I received RREQ from E to go to this node. Start RREP: AGEI
E got RREP from I to A from: AGEI
G got RREP from E to A from: AGEI
A got path to I from:AGEI
F received RREQ from C, with path: AGEC
```

The simulation demonstrates that even after removing a node (router), by using dynamic source routing protocols new paths can be found. For example, at first the program attempts to send an

RREQ from node A to node I, and it is successful. It returns the path of AGEC after sending back an RREP. The next step destroys the connection or edge between A and G, then attempts to find a new path from A to I. It does so successfully, and returns the new path of ABDEC. Finally, the connection between A and G is reestablished, so the final simulation shows that it once again found the most efficient path, AGEC. To test the code, simply use the makefile by writing "make dsr" into the terminal.

DSR protocol is a great way to establish connections or paths between nodes that are connected to other nodes in order to communicate. Communication paths can be determined through frequently changing network topologies.