CS460 Lab 4 Report: Routing

Michael K. Eagar

April 18th, 2014

1 Routing Protocol Implementation

I did not fully complete this lab. However, I did get routing to work partially. My implementation of distance vector routing only works successfully if every node in the network is connected directly to every other node in the network—if each node is a neighbor to every other node. But the network can consist any number of nodes. My implementation is saved in the file *test_routingApp.py*.

In my implementation, I use the file *five-nodes-mesh.txt* saved in the **networks** directory to create a network consisting of five nodes, in which every node is linked directly to every other node. I then demonstrate that every node is reachable by every other node by sending one packet from each node to all of its neighbors. At the end of the implementation I also print out the distance vector table for each node to show that it is correct for the network. I tested it on completely connected mesh networks of sizes 2, 3, 4, 5, and 6, and it worked successfully for those as well.

My implementation of distance vector routing really takes place in the **RoutingApp** class within the *test_routingApp.py* file. The methods *updateDVTable()*, *addDVToTable()*, and *sendDVTable()* make up my implementation. I also modified *src/packet.py* and changed the **Packet** class to include an **orig_host** variable, in order to get my implementation to work. Here are the distance vector routing methods that make up my implementation:

Listing 1: Distance Vector Routing implmentation

```
1         def updateDVTable(self, packet):
2                 print "Node %s is updating its DVT, packet from: %s, packet.body
                      : %s" % (self.node.hostname, packet.orig_host, packet.body)
3                 newRow = dict(packet.body)
4
5                 if packet.orig_host in self.distanceVectorTable:
6                         oldRow = self.distanceVectorTable[packet.orig_host]
7                         for key in newRow.keys():
8                                 if key in oldRow:
9                                         if (oldRow[key]) > (newRow[key] + 1):
10                                                oldRow[key] = newRow[key]
11                                                self.sendDVTable()
12                                else:
13                                        oldRow[key] = newRow[key]
14                                        self.sendDVTable()
15
16                else:
17                        self.addDVToTable(packet.orig_host, newRow)
18                        linkToMe = self.net.get_node(packet.orig_host).
                              get_address(self.node.hostname)
19                        linkToHim = self.node.get_link(packet.orig_host)
20                        self.node.add_forwarding_entry(linkToMe, linkToHim)
21                        self.sendDVTable()
22
```

```
23
24          def addDVToTable(self, nodeName, distanceVector):
25                  print "nodeName:",nodeName
26                  self.distanceVectorTable[nodeName] = distanceVector
27                  for key in distanceVector.keys():
28                          if nodeName == self.node.hostname:
29                                  self.distanceVectorTable[self.node.hostname][key
                                        ] = distanceVector[key]
30                          else:
31                                  self.distanceVectorTable[self.node.hostname][key
                                        ] = distanceVector[key] + 1
32
33          def sendDVTable(self):
34                  p = Packet(source_address=0, destination_address=0, ttl=1,
                        protocol='dvrouting', body=self.distanceVectorTable[self.node
                        .hostname], orig_host=self.node.hostname)
35                  Sim.scheduler.add(delay=0, event=p, handler=self.node.
                        send_packet)
```

I feel that if I had been able to get started on the lab earlier (before the last week of class) I would have been able to finish it completely. I do think that I would need to redo my implementation completely in order to get routing to work correctly on any network arrangement, as it breaks down fairly quickly when each node is not a neighbor to the others.