Lab 5 Report

Kyle Cornelison, Spencer Wood

4/18/2017

1 Routing Algorithm Details

Our routing algorithm is based on the distance-vector routing algorithm which uses the Bellman-Ford algorithm used to find the shortest path from each node to all of the other nodes within a graph. Our implementation relies on nodes broadcasting their distance-vector to each of their neighbors, defined to be one hop away. Upon receiving a broadcasted distane-vector packet the node will compare its distance vector to the one it received and update any costs, and forwarding entries accordingly.

2 Experiment 1 - Line of 5 Nodes

2.1 Network Configuration

```
#
#  n1 -- n2 -- n3 -- n4 -- n5
#
n1 n2
n2 n1 n3
n3 n2 n4
n4 n3 n5
n5 n4
```

2.2 Experiment Description

The following experiment utilizes our routing algorithm to determine routes between the nodes in the network displayed above. We then send a packet from node 1 to node 5 and trace the path it takes. The path taken should be in sequence from node 1 to node 5 (1, 2, 3, 4, 5).

2.3 Experiment Results

2.3.1 Distance Vectors

```
To: n1, Packet #20
{
  "n1": {
    "cost": 0,
    "address": null
  },
  "n2": {
    "cost": 1,
    "address": 2
  },
  "n3": {
    "cost": 2,
    "address": 4
  },
  "n4": {
    "cost": 3,
    "address": 6
  },
  "n5": {
    "cost": 4,
    "address": 8
  }
}

To: n2, Packet #24
{
  "n1": {
    "cost": 1,
    "address": 1
  },
  "n2": {
    "cost": 0,
    "address": null
  },
  "n3": {
    "cost": 1,
    "address": 4
  },
  "n4": {
    "cost": 2,
    "address": 6
  },
  "n5": {
    "cost": 3,
    "address": 8
  }
}

To: n3, Packet #20
{
  "n1": {
    "cost": 2,
    "address": 1
  },
  "n2": {
    "cost": 1,
    "address": 3
  },
  "n3": {
    "cost": 0,
    "address": null
  },
  "n4": {
    "cost": 1,
    "address": 6
  },
  "n5": {
    "cost": 2,
    "address": 8
  }
}

To: n4, Packet #25
{
  "n1": {
    "cost": 3,
    "address": 1
  },
  "n2": {
    "cost": 2,
    "address": 3
  },
  "n3": {
    "cost": 1,
    "address": 5
  },
  "n4": {
    "cost": 0,
    "address": null
  },
  "n5": {
    "cost": 1,
    "address": 8
  }
}

To: n5, Packet #21
{
  "n1": {
    "cost": 4,
    "address": 1
  },
  "n2": {
    "cost": 3,
    "address": 3
  },
  "n3": {
    "cost": 2,
    "address": 5
  },
    "n4": {
      "cost": 1,
      "address": 7
    },
    "n5": {
      "cost": 0,
      "address": null
    }
  }
```

2.3.2 Trace

```
1000.0 n1 forwarding packet to 8
1000.0018 n2 forwarding packet to 8
1000.0036 n3 forwarding packet to 8
1000.0054 n4 forwarding packet to 8
1000.0072 n5 received packet
```

3 Experiment 2 - Ring of 5 Nodes

3.1 Network Configuration

```
#
#     n3
#    /  \
#  n2    n4
#  |      |
#  n1 -- n5
#
n1 n2 n5
n2 n1 n3
n3 n2 n4
n4 n3 n5
n5 n1 n4
```

3.2 Experiment Description

The following experiment utilizes our routing algorithm to determine routes between the nodes in the network displayed above. We then send a packet from node 1 to node 5 and trace the path it takes. The path taken should be in sequence from node 1 to node 5 (1, 5). We then take down a link (the link between 1 and 5) and trace the new path taken from node 1 to node 5 (1, 2, 3, 4, 5).

3.3 Experiment Results - No Failure

3.3.1 Distance Vectors

```
To: n1, Packet #25
{
  "n1": {
    "cost": 0,
    "address": null
  },
  "n2": {
    "cost": 1,
    "address": 3
  },
  "n3": {
    "cost": 2,
    "address": 5
  },
  "n4": {
    "cost": 2,
    "address": 8
  },
  "n5": {
    "cost": 1,
    "address": 9
  }
}

To: n2, Packet #24
{
  "n1": {
    "cost": 1,
    "address": 1
  },
  "n2": {
    "cost": 0,
    "address": null
  },
  "n3": {
    "cost": 1,
    "address": 5
  },
  "n4": {
    "cost": 2,
    "address": 7
  },
  "n5": {
    "cost": 2,
    "address": 9
  }
}

To: n3, Packet #23
{
  "n1": {
    "cost": 2,
    "address": 1
  },
  "n2": {
    "cost": 1,
    "address": 4
  },
  "n3": {
    "cost": 0,
    "address": null
  },
  "n4": {
    "cost": 1,
    "address": 7
  },
  "n5": {
    "cost": 2,
    "address": 10
  }
}

To: n4, Packet #25
{
  "n1": {
    "cost": 2,
    "address": 2
  },
  "n2": {
    "cost": 2,
    "address": 4
  },
  "n3": {
    "cost": 1,
    "address": 6
  },
  "n4": {
    "cost": 0,
    "address": null
  },
  "n5": {
    "cost": 1,
    "address": 10
  }
}

To: n5, Packet #23
{
  "n1": {
    "cost": 1,
    "address": 2
  },
  "n2": {
    "cost": 2,
    "address": 3
  },
  "n3": {
    "cost": 2,
    "address": 6
  },
  "n4": {
    "cost": 1,
    "address": 8
  },
  "n5": {
    "cost": 0,
    "address": null
  }
}
```

3.3.2 Trace

```
1000.0 n1 forwarding packet to 9
1000.0018 n5 received packet
```

3.4 Experiment Results - With Failure

3.4.1 Distance Vectors

3.4.2 Trace

4 Experiment 3 - Mesh Network

4.1 Network Configuration

```
#
#       --- n7 --- n8
#      /    |       |
#    n9     |       |
#      \--- n6     n2 -- n14 -- n15
#            \    /  \   /
#             \  /    \ /
#      n10 --- n1     n3
#               |      |
#               |      |
#              n4 --- n5 -- n13
#               |      |
#               |      |
#              n11    n12
#
```

```
n1 n2 n4 n6 n10
n2 n1 n3 n8 n14
n3 n2 n5 n14
n4 n1 n5 n11
n5 n3 n4 n12 n13
n6 n1 n7 n9
n7 n6 n8 n9
n8 n2 n7
n9 n6 n7
n10 n1
n11 n4
n12 n5
n13 n5
n14 n2 n3 n15
n15 n14
```

4.2 Experiment Description

The following experiment utilizes our routing algorithm to determine routes between the nodes in the network displayed above. We then send a packet from node 1 to node 5 and trace the path it takes. The path taken should be in sequence from node 1 to node 5 (1, 4, 5). We then take down a link (the link between 1 and 4) and trace the new path taken from node 1 to node 5 (1, 2, 3, 5).

4.3 Experiment Results - No Failure

4.3.1 Distance Vectors

4.3.2 Trace

4.4 Experiment Results - With Failure

4.4.1 Distance Vectors

4.4.2 Trace

4.5 Analysis

In conclusion the distance-vector routing algorithm is simple to implement and performs a good job at managing failures amongst nodes. The main drawback to this approach is the time it takes for the nodes' distance vectors to converge on the best routes to take.