

Python UDP

By: Jared Knutson, Clayton Franklin, Ian Grant

Technical Report:

Functionality:

This protocol is based on UDP and the RIP routing algorithm. We used distance vectors with a hop cost of one to each next node. When a node goes offline, it is added to a list of nodes that are currently down and removed from the routing table. A node going offline is detected by a failed response to a ping packet. This failed reception adds the node to a list of offline nodes that is propagated throughout the network so that each node removes the downed node from their respective routing table. Once the node comes back online, the nodes surrounding the downed node will re-add the node to their routing tables and broadcast the path to other nodes in the network. It is able to send messages over UDP to nodes anywhere in the network.

Novel contribution:

Our novel contribution to the project is our ping packet. We used out of the box thinking to design our ping packet that we used to detect if a node went down or not. The ping packet worked on a send and receive basis. If this ping packet were to not get any response, then the node would be set to a down state and removed from the routing table as it is no longer reachable by any other node. This allows for a node to be down, the outage be detected, and then to update each node accordingly.

Results:

The protocol was able to accomplish the task that we set out to solve. The routing tables are created and propagated for all of the nodes. Routing tables are dynamically updated when a node goes offline. The neighbors of the dead node inform their neighbors of the change, causing the updating of the tables to propagate throughout the network. While the protocol worked well, it occasionally had issues with removing routes from the routing table.

Screen Shots:

NID: 4

Link Table:

NODE	L1	L2	L3	L4
1	2	3	0	0
2	1	4	0	0
3	1	4	0	0
4	2	3	0	0

Address Data Table:

DEST	ADDR	PORT
1	127.0.0.1	50555
2	127.0.0.1	50556
3	127.0.0.1	50557
4	127.0.0.1	50558

Routing Table:

DEST	COST	NEXT
4	0	4
2	1	2
1	2	2
3	1	3

Press any key to continue...

```
3. bash
enter node to message: 4
enter the message you want to send: hi
Press any key to continue...█
```

```
4. bash
Enter 'info' to check network information
Enter 'send_udp' to message another node via UDP
Enter 'q' to end program
Enter Selection: hi
Press any key to continue...█
```

NID: 2

Link Table:

NODE	L1	L2	L3	L4
1	2	3	0	0
2	1	4	0	0
3	1	4	0	0
4	2	3	0	0

Address Data Table:

DEST	ADDR	PORT
1	127.0.0.1	50555
2	127.0.0.1	50556
3	127.0.0.1	50557
4	127.0.0.1	50558

Routing Table:

DEST	COST	NEXT
2	0	2
1	1	1
3	2	4
4	1	4

Press any key to continue...