COSC 364: Asg1

[Company name] | [Company address]

RIP Routing protocol implemetation

2021

# Project Contribution:

## Bach Vu (25082165) (60%)

- Creation of Router Class

- Creating of socket, Sending and Receiving

- Creation of Timer.py/Garbage collection

- Printing routing table

## Charlie Hunter (27380476) (40%)

- Reading Configuration files

- Creating and processing packets

- Update routing table

- Tests: Basic Functionality, Split horizon with poisoned reversed, Larger Network.

# Questions

### Which aspects of your overall program (design or implementation) do you consider particularly well done?

The *daemon,py* ,*daemon\_sup.py*, *router.py* and *timer.py* all keep functionality of the code separate so debugging and reading code is easier. The *daemon.py* file is the main file where the code runs, while *daemon\_sup.py* has functions to assist the daemon file (reading config, creating/processing packets, etc). The *router.py* file has the Router Class where updates to the routing table, checking for expired routes and printing of the routing table and keeping all information about each route (Outputs, inputs, and timeouts) occurs. The ***timer.py*** file checks for a timeouts and garbage timeouts, also where times for each router is stored.

### Which aspects of your overall program (design or implementation) could be improved?

* Timeout thing in select
* More efficient way of doing the config files instead of one for each router

### How have you ensured atomicity of event processing?

The best we could find to do to ensure some level of atomicity was to create a short timeout on the select, and a guard statement to check the timeout. This was the best option but is not concurrency, just a way to minimize computer resources while waiting for an input to be sent to the socket. This can be seen in the *Daemon.py* in the receive function on line 53, where timeout is set to 0.013 and sending a packet is only done if the update table function is returned true.

### Have you identified any weaknesses of the RIP routing protocol?

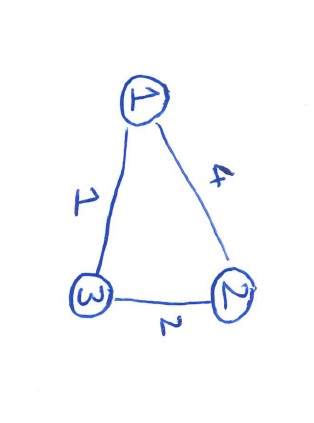
Two weakness of the RIP routing protocol are the maximum hop count of 15 and lots of network traffic. The RIP protocol can only reach routers in the network that are 15 hops away, this means in large networks where hops between routers are more than 15 hops away the RIP protocol is not able to be used. Another problem is the high level of network traffic due to updates being sent between routers periodically.

**Testing**

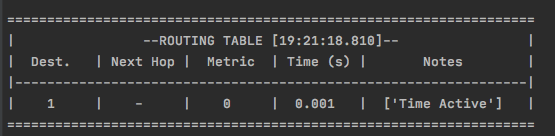
Three tests networks where made for testing each with a specific functionality to determine the code ran correctly. Test one, was a basic functionally test on a small network, test two was a split horizon with poisoned reverse test, and test three was to test convergence on a large network. Trivial test where also carried out on the demo network provided to us.

**Test 1 – Basic Functionality test.**

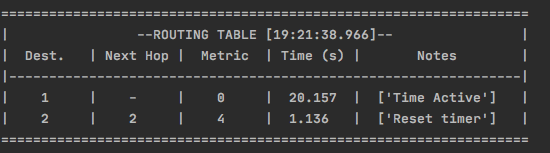
The first test conducted can be seen on the network below in *figure 1.* This test was to test the basic functionality of code. Firstly router one and router two where ran and connected to one another. After this router three was ran, and router one and two both found a shorter path to each other via router three. The process of router one can be seen below in *figures 2 3, 4* and *5*.



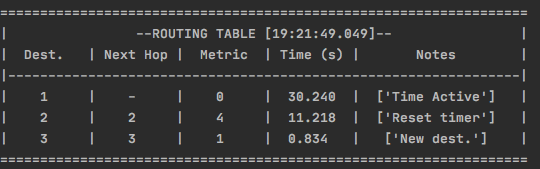
*Figure 1 – Basic functionality test network*



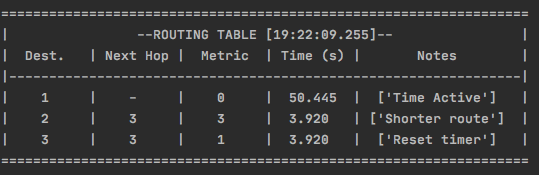
*Figure 2 – router one starting*



*Figure 3 – router one converging to router two*

**

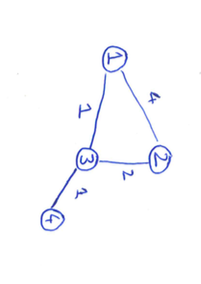
*Figure 4 – Router three turning on and converging to the network*



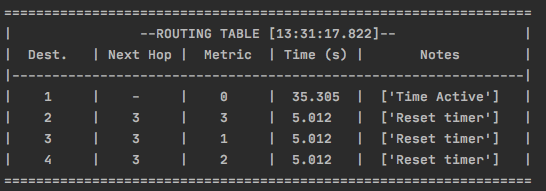
*Figure 5 – Router one connecting to router two via router three for a smaller cost*

**Test 2 – split horizon and poisoned reverse**

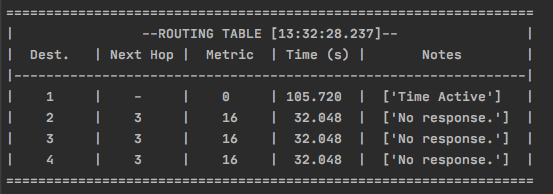
The second test conducted can be seen below on *figure 6*. The purpose of this test was to test split horizon with poisoned reverse. Firstly all routers where ran so that they could all converge. Once convergence has occurred router three was turned off. Routers one and two both receive a timeout from router three they understand that router three is dead and stop advertising that router three and four and set both the metric to router three and four to 16. After a garbage timeout both routers one and two will remove routers three and four from there routing table. Router ones process of this can be seen below in *figures 7 to 10*



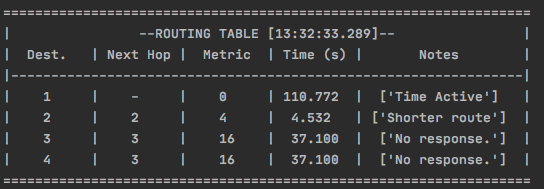
*Figure 6 – Split horizon with poisoned reverse test network*

**

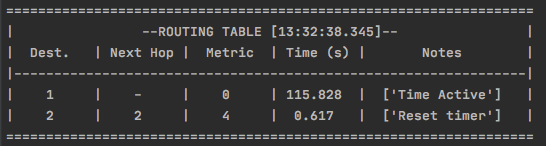
*Figure 7 – Routing table of route one after convergence of all routers*

**

*Figure 8 – routing table of route one after route three is turned off*

**

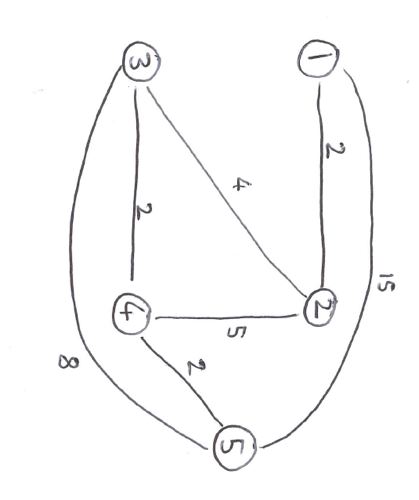
*Figure 9 – routing table of route one re-routing to route 2*

**

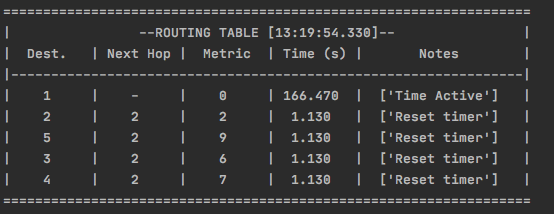
*Figure 10 – routing table of route one, after garbage time where both route four and route three are removed from the routing table*

**Test 3 – A larger network test with five routers.**

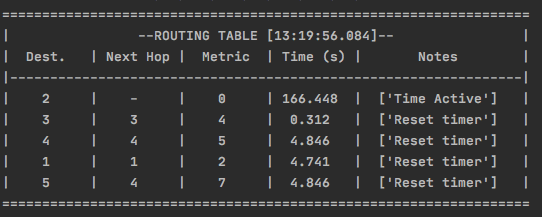
The third test was to test how the network would go with a larger network of 5 routers. This test was to make sure in a larger network different from the provided demo network the code would still run correctly. The network can be seen below in *figure 11*. The correct routing tables after convergence can also be seen for each router in *figures 12-16*.



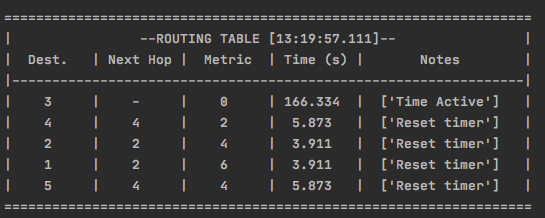
*Figure 11 - five routers test network*



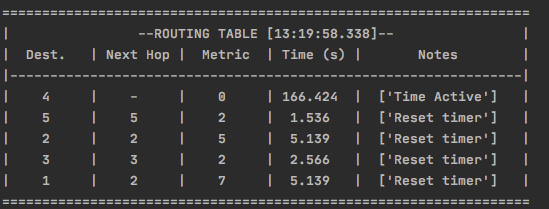
*Figure 12 – Routing table for route one*



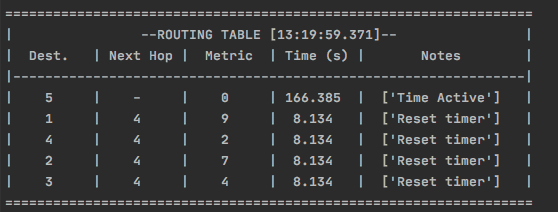
*Figure 13 – Routing table for route two*



*Figure 14 – Routing table for route three*



*Figure 15 – Routing table for route four*

**

*Figure 16 – Routing table for route five*

**Example configuration file**

Below an example configuration file for a router can be seen, the example shows router one.

router-id: 1

input-ports: 51002, 51003, 51004, 51005, 51006, 51007

outputs: 51007-57001-8-7, 51006-56001-5-6, 51002-52001-1-2

Input-ports is identified by 5X00Y, where X is equal to the router Id of the sending router and Y is the router Id of the receiving router

Outputs is defined by X-Y-Z-A, where X is the input port being sent from, Y is the Output Port being sent too, Z is the metric value, and A is the router Id being sent too.

**Source code**