COSC 364: Asg1

[Company name] | [Company address]

RIP Routing protocol implemetation

Charlie hunter (27380476)

Charlie hunter (27380476)

2021

# Project Contribution:

## Bach Vu (60%)

- Creation of Router Class

- Creating of socket, Sending and Receiving

- Creation of Timer.py/Garbage collection

- Printing routing table

- Debugging and refining code

## Charlie Hunter (40%)

- Read 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 *deamon,py* ,*deamon\_sup.py*, *router.py* and *timer.py* all keep functionality of the code separate so debugging and reading code is easier. The *deamon.py* file is the main file where the code runs, while *deamon\_sup.py* has functions to assist the deamon 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.

Select.select(\_,\_,\_,\_) & concurrency

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

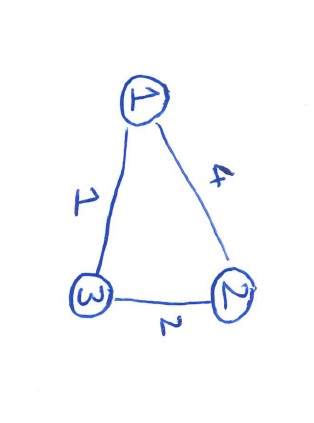
* Could be more efficient

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

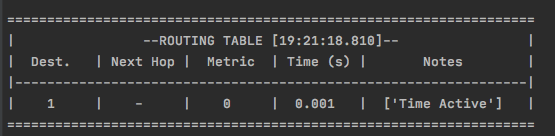
### Have you identified any weaknesses of the RIP routing protocol?

**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*

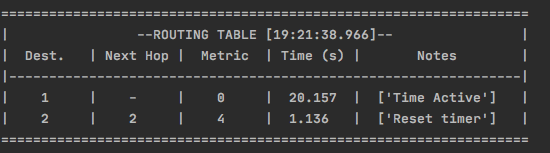


Figure 3 – router one converging to router two

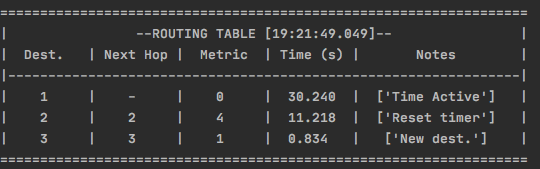


Figure 4 – Router three turning on and converging to the network

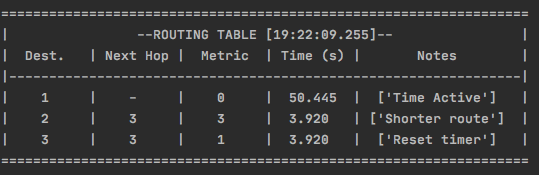
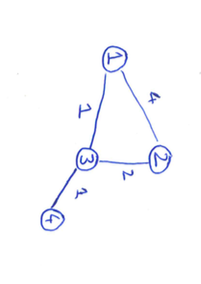


Figure 5 – Router one connecting to router two via router three for a smaller cost

**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 adversting 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 x to y***



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

**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 X***. The correct routing tables after convergence can also be seen for each router in **figures X-X.**

