COSC 465 Capstone Project

Alex Greene, John-Michael Chin, Lizeth Mora Guerrero, Luis Ramos

Milestone 2 Reflection

Progress:

Since milestone 1, we have incorporated new functionalities due to some issues and have refined existing components to align with our project goals. Specifically, we have made some modifications to packet.py to accommodate various packet types effectively. Our router.py file has been modified to enhance functionality. More specifically, the routing table structure has been redefined to optimize routing decisions. We have also implemented a method ‘add_links’ to dynamically update routing tables based on direct links. In the simulation.py file, the Simulation class now initializes routers and clients based on the provided JSON configuration, ensuring a seamless integration of network entities. After some time trying to fix our implementation of Switchyard, we have decided to instead use network.py where we will create our simulated “network” where, given a source IP and destination IP, it will send the packet to the correct Client or Router object. This is still in progress.

Struggles:

We’ve had issues with trying to run switchyard in order to test if our scenarios would work. Because of the issues with running switchyard, we’ve had to shift our focus towards creating our own Network class in place of switchyard. This still means that testing how our routers interact with each other is still a work in progress.

To-Do:

Our priority has been to address the issues encountered with switchyard integration. Ideally, our aim is to resolve the existing challenges and ensure compatibility with our project, but we have shifted our focus to implementing a network.py file where we will create our simulated “network” where given a source IP and destination IP, it will send the packet to the correct Client or Router object. Once core functionalities are implemented, we will shift focus to developing a user interface component. The UI will provide users with a visual representation of network topologies and simulation outcomes, enhancing usability and accessibility.

Timeline:

The main focus of our discussions has been dedicated to resolving our Switchyard integration issues. If this does not end up being possible, we have a plan of using network.py, which we have pseudocoded during our discussions, to aid in completing the functions of our program. We are also focusing on ARP handling and IP packet forwarding. UI development is our final component, which will commence once our code has been refined.