

COSC 465 Capstone Project

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

Milestone 2 Reflection

Progress:

Since milestone 1, we have made some key developments. The client.py file has been refined, with revised functionalities and improved packet handling capabilities. This includes introducing methods to send packets, receive packets, perform ARP requests, and handle ARP packets efficiently. This enhances communication capabilities within the network, allowing clients to send packets seamlessly to any directly connected router. A new network.py file has been introduced to manage network configuration effectively. This file parses a JSON configuration file ('network_config_test.json') and initializes routers and clients based on the specified network topology. This addition replaces our previous usage of Switchyard and streamlines network setup procedure and enhances configurability. The packet.py file has been refined to include additional parameters and functionalities to support various packet types. Notability, there are improvements to the checksum calculation, payload handling, and size estimation functionalities, ensuring packet processing capabilities. The router.py file has been enhanced to include the introduction of methods to handle ARP replies, process IP packets, perform ARP requests, and manage routing tables efficiently. Additionally, we have made some modifications to optimize routing decisions. Finally, the test.py file has been expanded to incorporate comprehensive testing scenarios. Specifically, a new class ('TestNetworkSimulation') has been introduced and test cases have been developed to verify packet transfer between clients within the simulated network.

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. Thus, the testing for whether our network worked as intended took longer than expected, giving us less time to work on other matters such as implementing dynamic routing protocols.

To-Do:

An aspect that still needs to be worked on is implementing dynamic routing protocols, which should be useful for dealing with networks with a large quantity of routers and clients. So we need to create a network topology that is large enough to warrant the use of dynamic routing protocols and testing it so that packets are sent correctly, routers and clients interact without issue, and that everything works as intended.

Timeline:

Our timeline remains mostly unchanged from our initial proposal and ideas. Dynamic routing protocol integration remains a priority, with completion anticipated within the upcoming milestone. We are also making efforts to enhance the testing framework and ensure comprehensive coverage of the project's functionality. Documentation preparation is in progress throughout each milestone as we work and that has remained consistent. It is worth noting that due to time constraints, plans to incorporate a user interface have been revised. We will instead implement print statements to mimic basic interaction capabilities for demonstration purposes. This approach aims to ensure accessibility and ease of understanding, especially for individuals learning about this topic, to facilitate a seamless learning experience.