Here is a brief description of the functions and variables in each script:

Tester_traceroute.py

- traceroute(): This function performs the traceroute operation and returns a list of hops.
- traceroute_with_threads(): This function uses a thread pool to execute the traceroute() function in parallel.
- router_iterator(): This function generates a list of IP addresses to traceroute, based on a start and end IP address.
- createJSON(): This function creates a JSON object from the traceroute results.
- pushToMongoDB(): This function inserts the traceroute results into a MongoDB database.

Pymongo_get_database.py

- get_database(): This function connects to a MongoDB database and returns a reference to the database.

Network Topology of Howard University Campus Network

Introduction

The Howard University campus network is a complex and dynamic system that supports a variety of academic and administrative services. The network topology is a critical component of the overall network infrastructure, and it plays a vital role in determining the performance and reliability of the network.

This document describes the approaches that were used to construct the network topology of the Howard University campus network. The following methods were used:

Traceroute: Traceroute is a network diagnostic tool that can be used to trace the path that packets take from a source host to a destination host. It does this by sending ICMP packets to the destination host with increasing TTL (Time to Live) values. The TTL value indicates how many times a packet can be forwarded by routers before it is discarded.
Traceroute was used to identify the routers and other network devices that are located on the Howard University campus network. It was also used to identify the paths that packets take between different parts of the network.

IP address analysis: IP address analysis can be used to identify the subnets and networks that are present on a network. It can also be used to identify the relationships between different networks.
IP address analysis was used to identify the different subnets and networks that are present on the Howard University campus network. It was also used to identify the routers and other network devices that connect the different networks together.

Multiprocessing: Multiprocessing is a programming technique that allows multiple processors to execute code simultaneously. This can be used to improve the performance of computationally intensive tasks.
Multiprocessing was used to speed up the traceroute and network topology mapping processes. By running these processes in parallel, we were able to complete the network topology construction process in a shorter amount of time.

Multithreading: Multithreading is a programming technique that allows multiple threads to execute code simultaneously within a single process. This can be used to improve the performance of tasks that involve a lot of input/output (I/O).

Multithreading was used to improve the performance of the IP address analysis process. By running multiple threads in parallel, we were able to analyze the IP addresses on the network more quickly.

Network topology mapping tools: Network topology mapping tools can be used to create a visual representation of a network topology. These tools can be used to identify the devices that are on the network, the connections between the devices, and the types of devices that are on the network.

Network topology mapping tools were used to create a visual representation of the Howard University campus network topology. This visual representation was used to identify the different devices and networks that are on the network, as well as the connections between them.

In addition to the methods listed above, we also used the following tools and technologies:

Python: Python is a general-purpose programming language that is well-suited for network programming. We used Python to develop the scripts that were used to perform the traceroute, IP address analysis, and network topology mapping tasks.

PyMongo: PyMongo is a Python library that can be used to interact with MongoDB databases. We used PyMongo to store the network topology data in a MongoDB database.

NetworkX: NetworkX is a Python library that can be used to create and manipulate network graphs. We used NetworkX to create a visual representation of the Howard University campus network topology.

Results

The results of the network topology construction process are as follows:

A list of all of the devices that are on the Howard University campus network.

A list of all of the networks that are present on the Howard University campus network.

A list of all of the routers and other network devices that connect the different networks together.

A visual representation of the Howard University campus network topology.

This information can be used to improve the performance and reliability of the Howard University campus network by identifying potential bottlenecks and other problems in the network topology.

Conclusion

The methods that were used to construct the network topology of the Howard University campus network were traceroute, IP address analysis, multiprocessing, multithreading, Python, PyMongo, and NetworkX and network topology mapping tools. These methods were used to identify the devices, networks, and connections that are present on the network. These methods and technologies were used to speed up the network topology construction process and to create a comprehensive and accurate representation of the network topology. This information can be used to improve the performance and reliability of the network.

Note:

Once the database and traceroute was established, it was decided that these group members would explore these ranges:

(10.0.0.1 —--10.64.0.1) ~ Myles

(10.64.0.2 —- 10.128.0.1) ~ Darnell

(10.128.0.2 —- 10.192.0.1) ~ Cammy