ENTS 749E: Network Automation

Spring 2024

Student: Diego Guzman UID: 117294071

Professor: Zoltan Safar

1. Introduction

The objective of this project is to demonstrate the knowledge acquired in class by creating a Python application that gathers network connectivity information from Juniper Networks devices and displays it on the screen in a graphical representation.

Requirements

The specifications for a successful completion of this assignment are that the python application should:

A. *Enable and run LLDP (Link-Layer Discovery Protocol) on the given devices*
B. *Gather LLDP connectivity information from a set of pre-configured routers*
C. *Display the network graph on the screen including the routers and their connections. The routers should be represented by graphical icons, and the connections should be represented by lines connecting the icons. The interface numbers and IP addresses should be displayed nearby the router and the corresponding connection line.*

Background and motivation

The Link Layer Discovery Protocol is a standardized layer 2 protocol defined by the IEEE 802.1AB-2005 standard [5]. LLDP is designed to enable devices to advertise their identity, capabilities, and neighbors on a local area network. In this way, LLDP facilitates automatic discovery of neighboring devices and provides essential information about their capabilities and configurations. For this reason, it is primarily used to help network administrators to understand

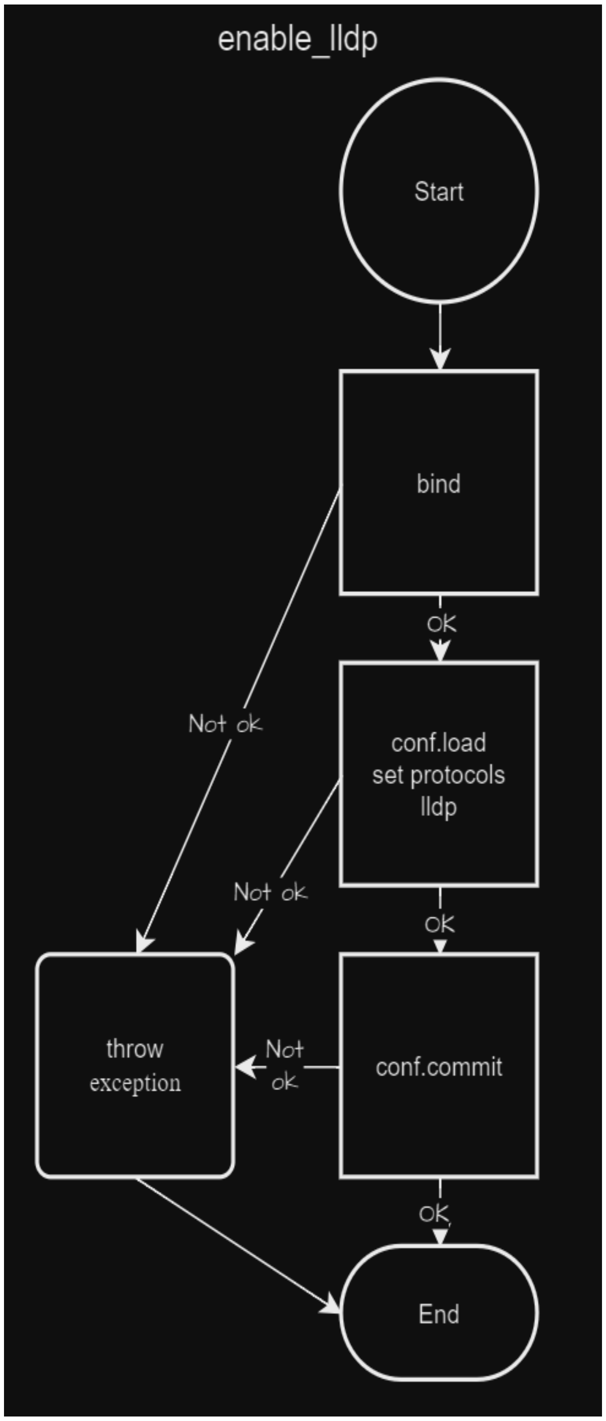
and manage network topologies in a more programmatical way which allows for better network optimization.

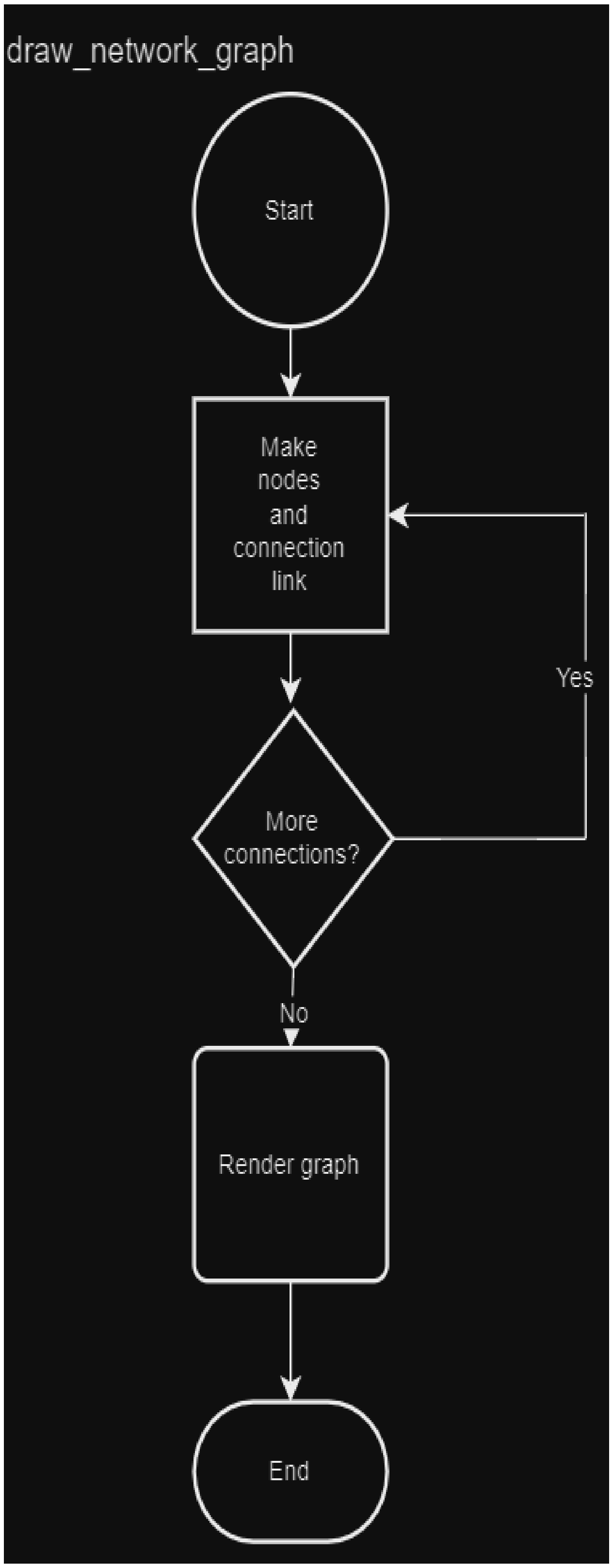
However, sometimes it is not enough to have the connection information in XML or TEXT format to get a good grasp of a complex architecture. To solve this problem, this program allows for automating the graphical visualization of a network with LLDP which can drastically improve the time taken to adapt to network changes in a dynamic network.

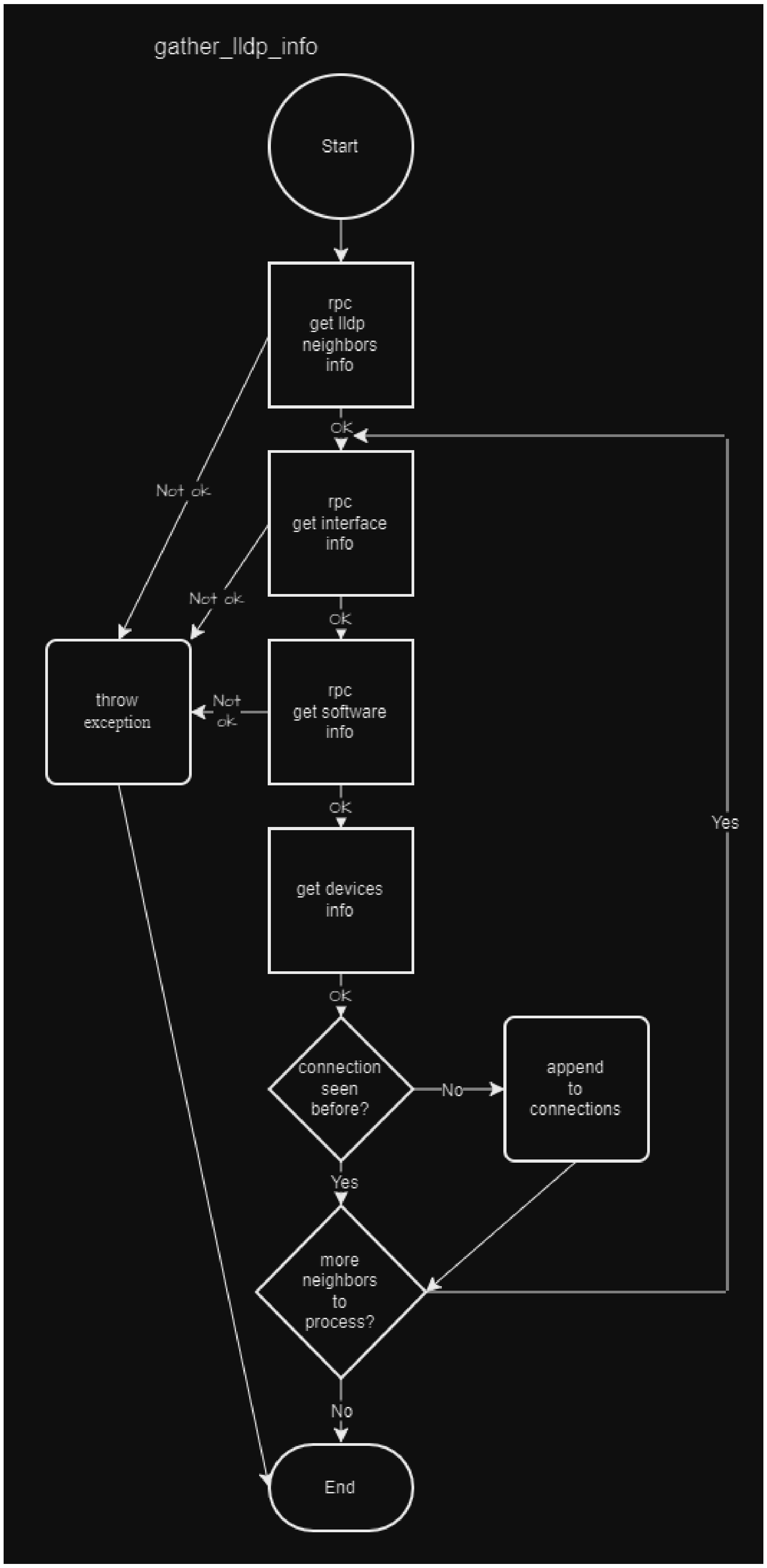
2. Program design

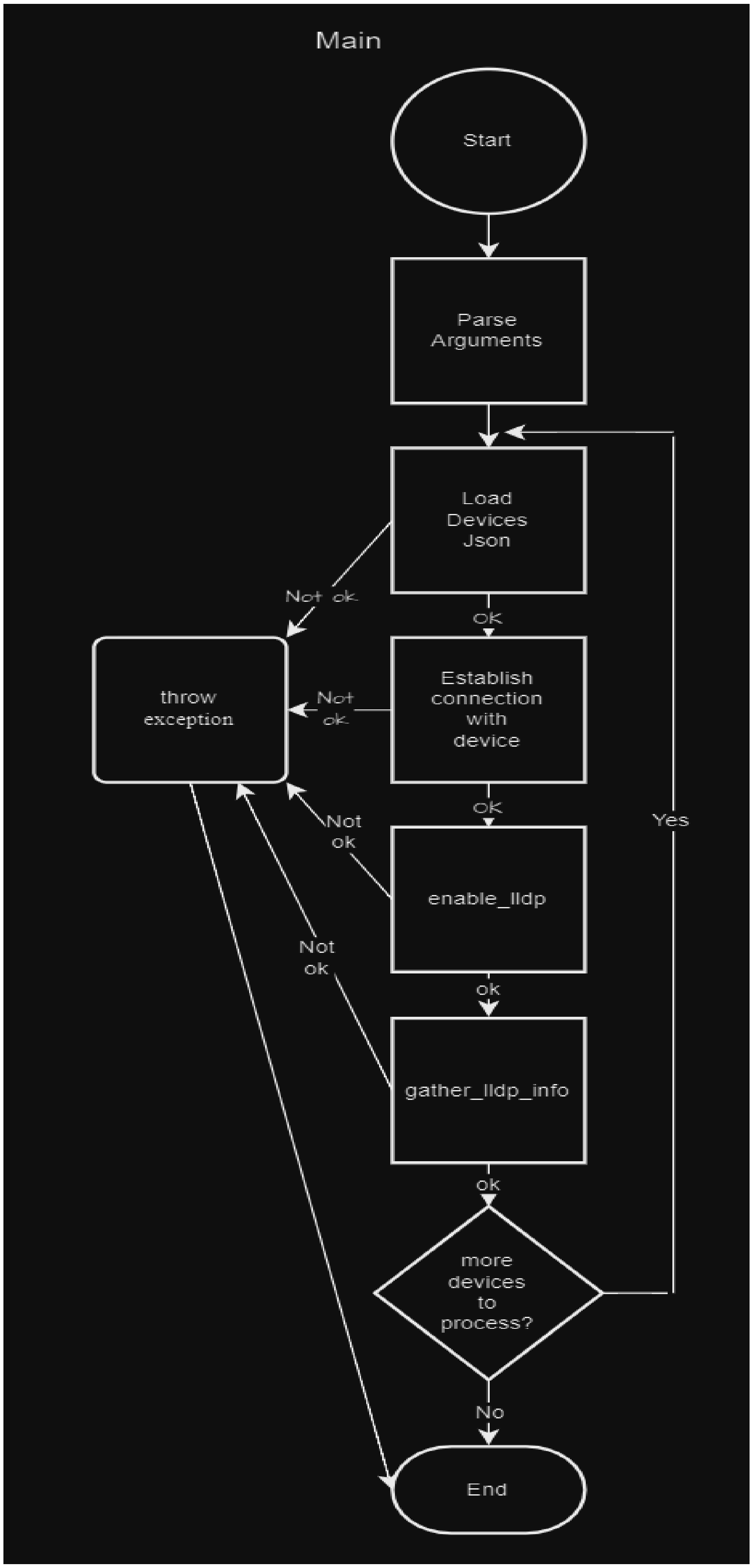
With the objective of maintaining a simple and logical design, the application was divided into 3 main blocks which helps to abstract the technicalities of the program. The actual design is pretty straightforward. Each block makes a sequence of processes either rpc requests or data gathering, and then all comes together in the main function to draw the final graph output. There is however an optimization to reduce the lookup time to know whether a connection has been seen before by keeping track of previous connections in a set to make the lookup time constant time (O(1)) instead of linear for lists (O(n)). The actual code for all functions in the program are included in the lldp_info.py file.

The flowcharts that show the methods functionality are the following:


enable_lldp
Start
bind
ok
conf.load
set protocols
lldp
Not ok
Not ok
ok
throw
exception
Not
ok
conf.commit
ok
End


draw_network_graph
Start
Make
nodes
and
connection
link
More
connections?
Yes
No
Render graph
End

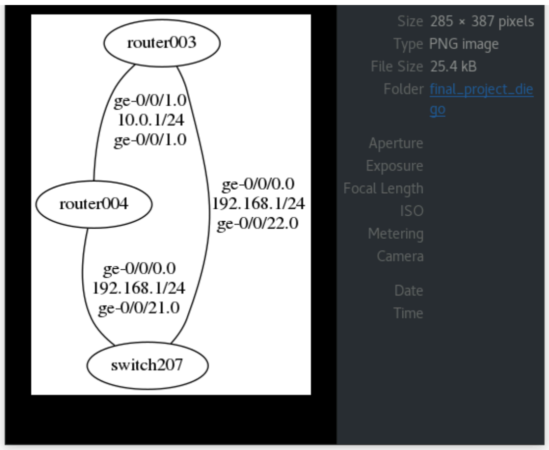

gather_lldp_info
Start
rpc get lldp neighbors info
ok
Not ok
rpc get interface info
ok
Not ok
throw exception
Not ok
rpc get software info
ok
get devices info
ok
connection seen before?
No
append to connections
Yes
more neighbors to process?
Yes
No
End


Main
Start
Parse Arguments
Load Devices Json
Not ok
ok
throw exception
Not ok
Establish connection with device
ok
Not ok
enable_lldp
Not ok
ok
gather_lldp_info
ok
more devices to process?
Yes
No
End

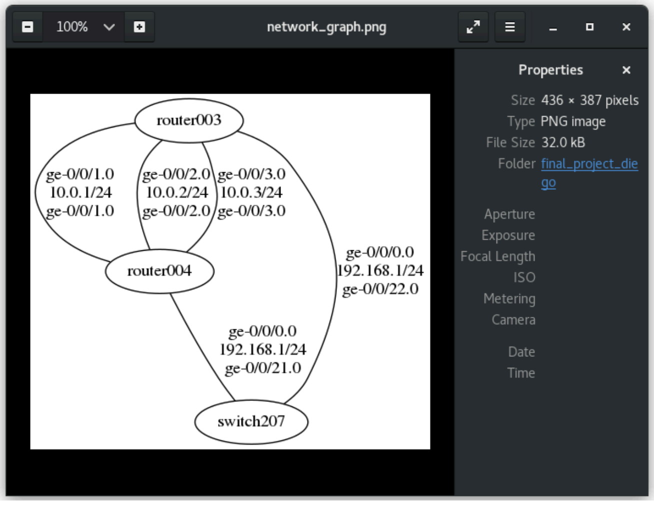
3. Program execution examples

In the README.md there is a thorough explanation of how to execute the program with all the options available, and the graph outputs generated. In this report, I include a screenshot of what actually shows up in your cli when you run the program. (Network is preconfigured before program execution).

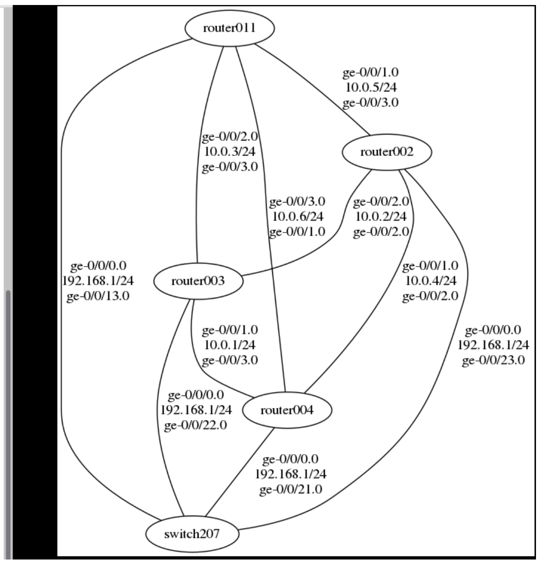
```
[labuser@localhost final_project_diego]$ python3 lldp_info.py
Initializing execution...


Establishing connection with ip 192.168.1.3...
Connection successful!
Enabling LLDP (Link-Layer Discovery Protocol)...
Successfully enabled LLDP!
Gathering LLDP connectivity information...
Successfully gathered LLDP connectivity information!


Establishing connection with ip 192.168.1.4...
Connection successful!
Enabling LLDP (Link-Layer Discovery Protocol)...
Successfully enabled LLDP!
Gathering LLDP connectivity information...
Successfully gathered LLDP connectivity information!

Displaying network graph...
Finalizing successful execution...
This tool has been deprecated, use 'gio open' instead.
See 'gio help open' for more info.
```



```
[labuser@localhost final_project_diego]$ python3 lldp_info.py
Initializing execution...


Establishing connection with ip 192.168.1.3...
Connection successful!
Enabling LLDP (Link-Layer Discovery Protocol)...
Successfully enabled LLDP!
Gathering LLDP connectivity information...
Successfully gathered LLDP connectivity information!


Establishing connection with ip 192.168.1.4...
Connection successful!
Enabling LLDP (Link-Layer Discovery Protocol)...
Successfully enabled LLDP!
Gathering LLDP connectivity information...
Successfully gathered LLDP connectivity information!

Displaying network graph...
Finalizing successful execution...
This tool has been deprecated, use 'gio open' instead.
See 'gio help open' for more info.

[labuser@localhost final_project_diego]$ 
```



```
[labuser@localhost final_project_diego]$ python3 lldp_info.py -f network_config.json
Initializing execution...


Establishing connection with ip 192.168.1.11...
Connection successful!
Enabling LLDP (Link-Layer Discovery Protocol)...
Successfully enabled LLDP!
Gathering LLDP connectivity information...
Successfully gathered LLDP connectivity information!


Establishing connection with ip 192.168.1.2...
Connection successful!
Enabling LLDP (Link-Layer Discovery Protocol)...
Successfully enabled LLDP!
Gathering LLDP connectivity information...
Successfully gathered LLDP connectivity information!


Establishing connection with ip 192.168.1.3...
Connection successful!
Enabling LLDP (Link-Layer Discovery Protocol)...
Successfully enabled LLDP!
Gathering LLDP connectivity information...
Successfully gathered LLDP connectivity information!


Establishing connection with ip 192.168.1.4...
Connection successful!
Enabling LLDP (Link-Layer Discovery Protocol)...
Successfully enabled LLDP!
Gathering LLDP connectivity information...
Successfully gathered LLDP connectivity information!

Displaying network graph...
Finalizing successful execution...
This tool has been deprecated, use 'gio open' instead.
See 'gio help open' for more info.

[labuser@localhost final_project_diego]$ 
```



In order to understand the output graph better there are the following considerations:

Ovals: devices' host-name are indicated inside the ovals.

Edges: the connections are represented by edges.

Edges_labels: located to the right of the connection, they indicate from top to bottom: the interface of the device on top, the subnet, and lastly the interface of the device on the bottom. An illustrative image of the format is shown below.

4. Conclusion

By developing this Python application, I have learned and demonstrated the effectiveness of using LLDP to gather network connectivity information to visualize a network topology. The application is complete with all 3 of the proposed requirements, with comprehensible documentation, with examples of execution and output, and with clear and understandable code that can be easily reused for other functionalities. Moving forward, a potential improvement would be to make the graph interactive to provide more customization.

From this project, I have understood the importance of visual capabilities in networks which has led me to consider another potentially interesting project that would take a graphical input and programmatically create the equivalent network.

5. References

[1] “Graphviz — graphviz 0.20.3 documentation.” https://graphviz.readthedocs.io/en/stable/ (Accessed May 5, 2024)

[2] Juniper Networks, “Junos PyEZ Documentation | Juniper Networks,” *Juniper Networks*. https://www.juniper.net/documentation/product/us/en/junos-pyez/#cat=developer_guides (Accessed April 15, 2024)

[3] P. Congdon, Andy Bierman, and Keith McCloghrie, “Link Layer Discovery Protocol and MIB.” [Online]. Available: https://www.ieee802.org/1/files/public/docs2002/lldp-protocol-00.pdf (Accessed April 15, 2024)

[4] “Features — jsons documentation.” https://jsons.readthedocs.io/en/latest/ (Accessed April 25, 2025)

[5]“IEEE Standard for Local and metropolitan area networks -- Station and Media Access Control Connectivity Discovery.” pp. 1–176, 2005.

[6] “argparse — Parser for command-line options, arguments and sub-commands,” *Python Documentation*. https://docs.python.org/3/library/argparse.html (Accessed May 8, 2024)