



Knight Foundation
School of Computing
and Information Sciences



Knight Foundation School of Computing and Information Sciences

Spring 2023 Senior Design Project

AmLight-Learning OpenFlow/SDN network Research and Experimentation

Students: Gabriela Ramirez

Mentor: Vinicius Arcanjo, AmLight

Instructor/Faculty: Seyedmasoud Sadjadi, Florida International University

PROBLEM

Software-defined networking (SDN) has benefitted from OpenFlow protocol in allowing dynamic access to forwarding planes. The AmLight-Exp network utilizes OpenFlow in high-latency data transfers. However, there is a lack of visibility concerning multiple network domains as well as issues behind automated network operations.

CURRENT SYSTEM

Kytos-ng was developed to become AmLight's next controller by providing generic OpenFlow deployment. It operates on a Python 3 framework that permits users to develop network applications (Napps). Additionally, it also allows developers to share SDN paradigms with other users.

REQUIREMENTS

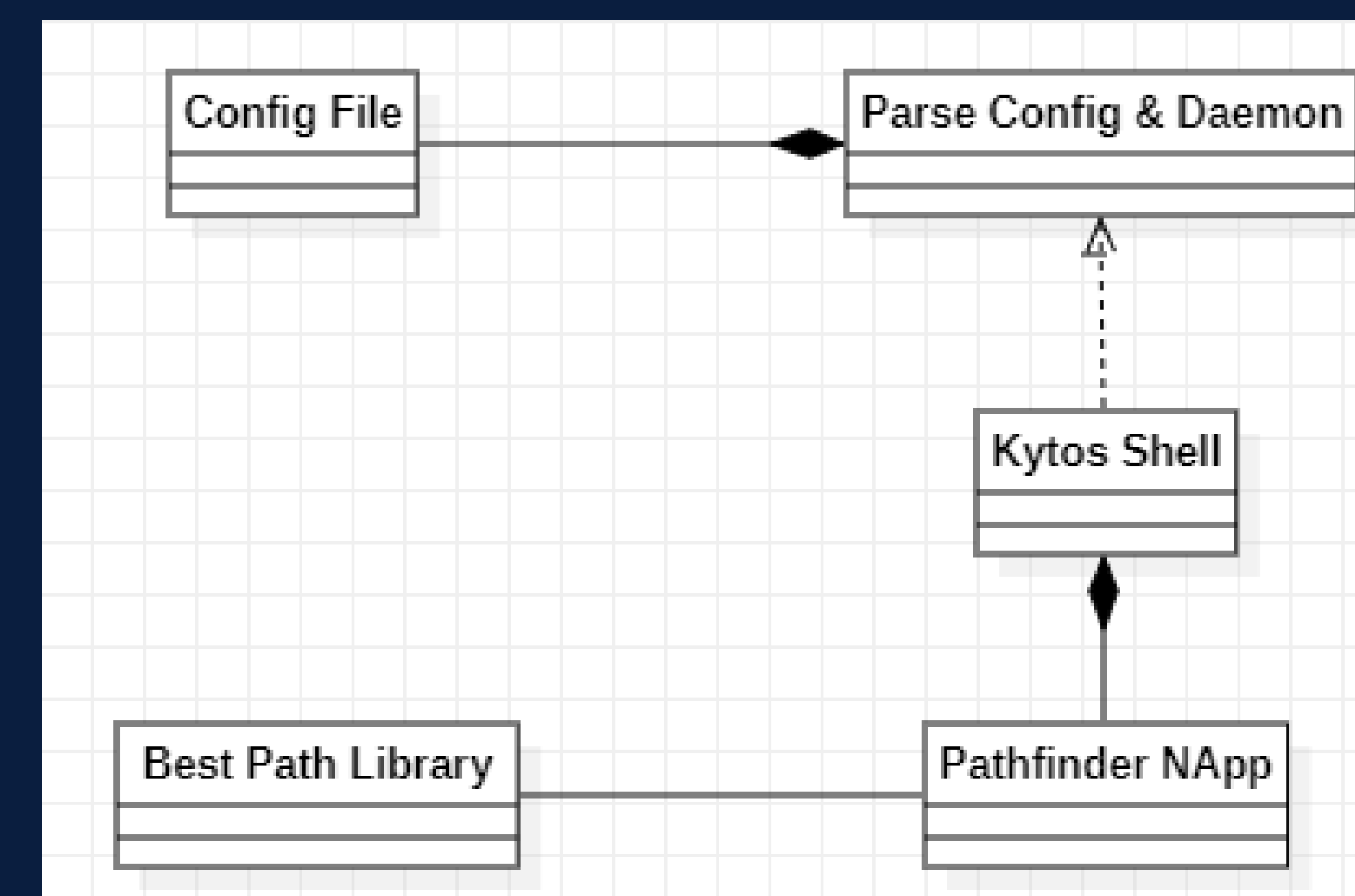
- Virtual environment (recommended)
- Git 2.40
- Docker 2.10
- Docker Compose 2.17
- Mininet 2.3.0
- Python 3.10

SYSTEM DESIGN

Three core components make the Kytos architecture:

- OpenFlow Library: Builds data structures from binary OpenFlow messages. Objects can then be translated to binary and sent to switches.
- Controller: Asynchronously defines queues for incoming messages, NApp events, and outgoing messages.
- NApps: Provides communication between nodes, statistics, and topology finding.

OBJECT DESIGN

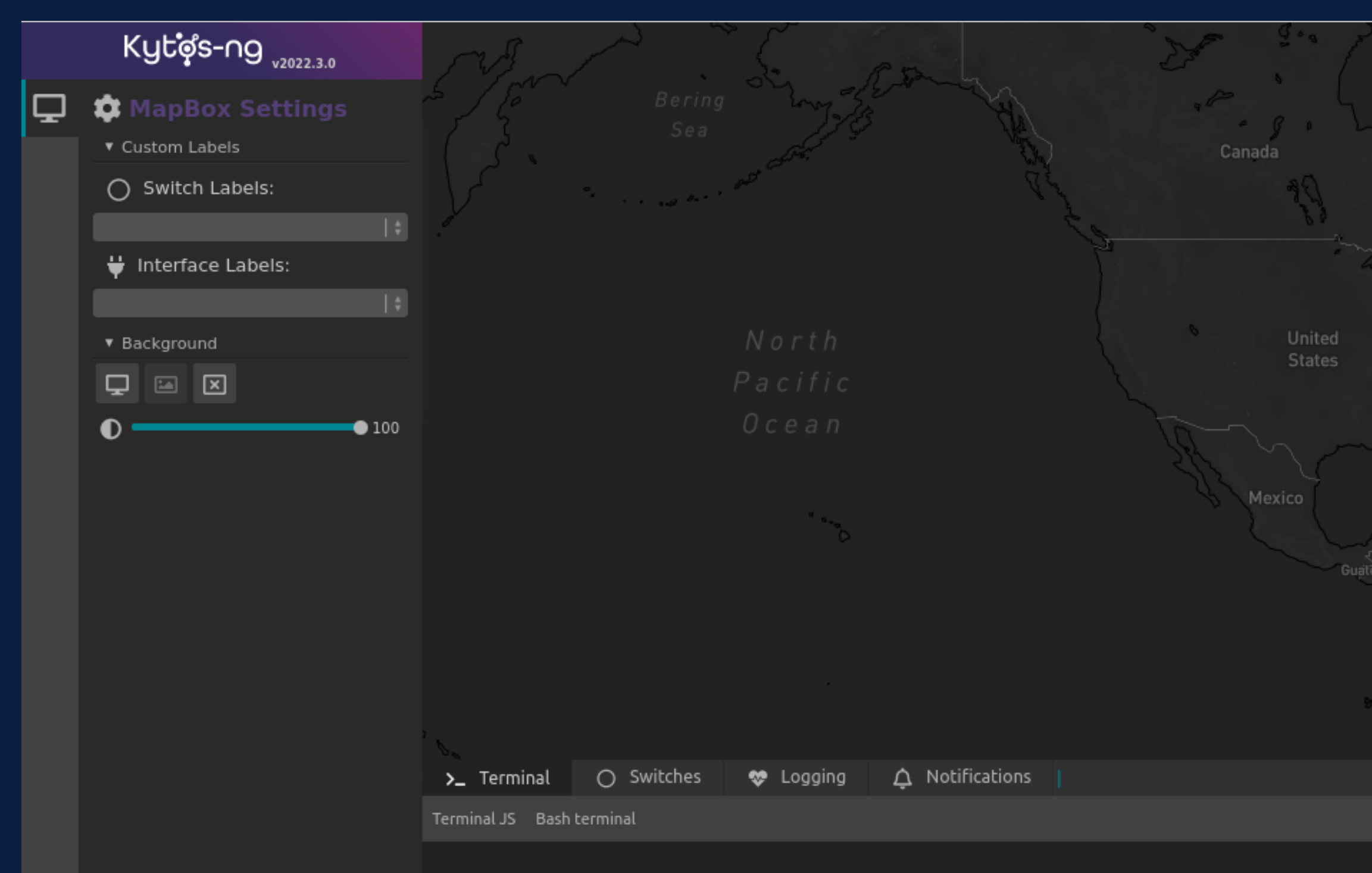


IMPLEMENTATION

Kytos is notable for having a simplistic user interface based on Vue.js:

- Menu Bar: Triggers respective toolbars.
- Toolbar: Used for input controls. Can be either compressed or expanded.
- Map: Renders the network topology to display hosts and switches.
- Info Panel: Displays generic information.
- Tabs: Displays a terminal, switch information, instance logs, and UI alerts.
- Actions Menu: Displays a list of actions.

SCREENSHOTS



SUMMARY

Kytos-ng is currently in development – seeking faster ways to deploy SDN networks. A fork of the now-shutdown Kytos framework, it builds upon its predecessor by granting users the ability to customize their own OpenFlow controllers. With its shareability, developers can connect with one another to propose solutions.

REFERENCES

- <https://docs.kytos.io/>
- <https://github.com/kytos-ng/kytos>

ACKNOWLEDGEMENT

The material presented in this poster is based upon the work supported by Vasilka Chergarova and Vinicius Arcanjo. We thank Vasilka Chergarova and Vinicius Arcanjo for their assistance and mentorship that we received throughout the senior design project.



Engineering
& Computing

FLORIDA INTERNATIONAL UNIVERSITY