

DECEMBER 7
2023

NETWORK FAILURE RECOVERY - TEAM 8

ARIF AWASAF
AQUIB

UTHMAN
FALOLA

NIKHIL SEBASTIAN
JOHN

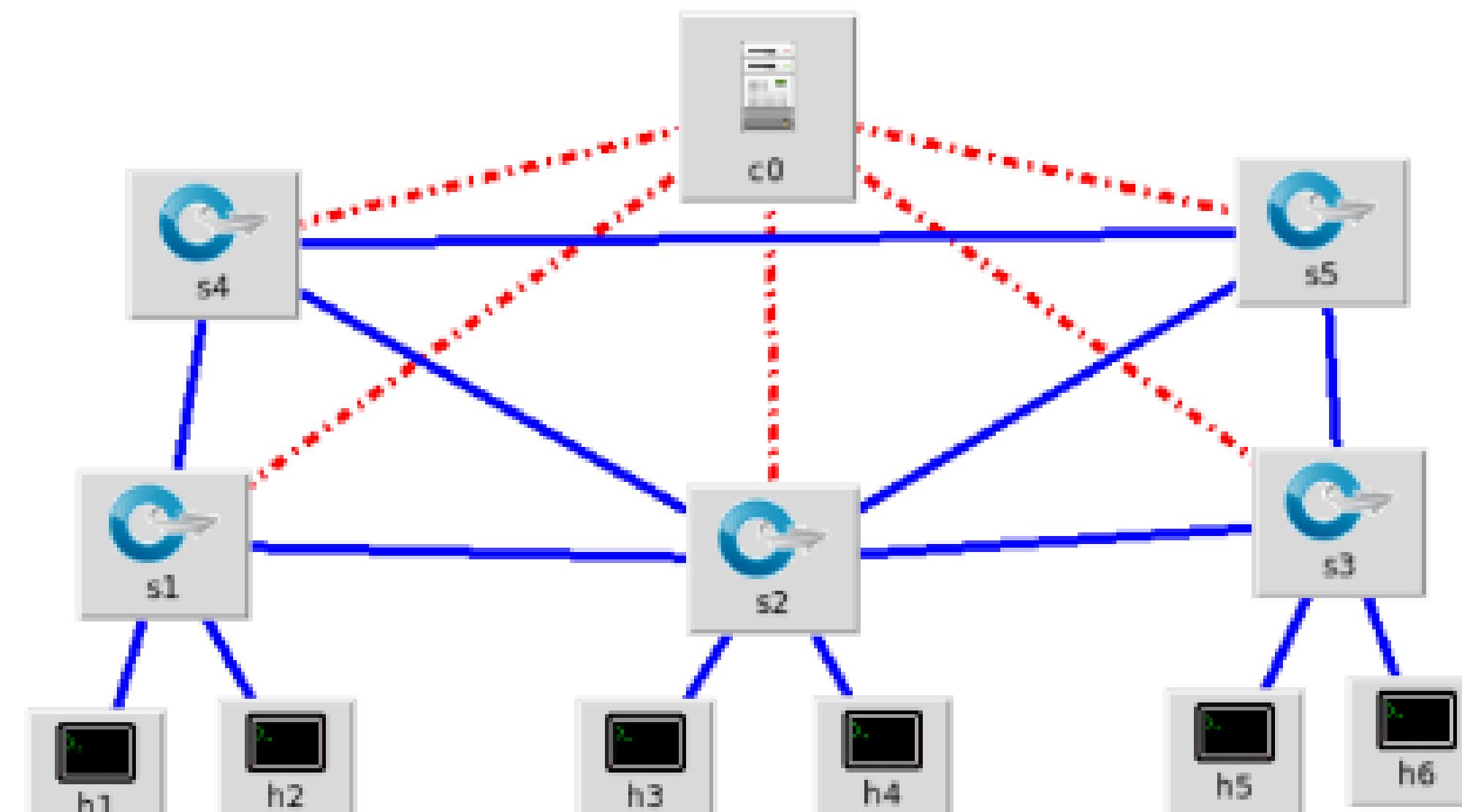
PROJECT OBJECTIVE

- Enables the recreation of complex network topologies for testing and simulation.
- Implement proactive measures like flow rerouting for rapid response to network failures.
- Utilize dynamic path computation to optimize network performance and minimize downtime.

Introduction

Network Topology

- 1 Controller
- 5 Switches
- 6 Hosts
- 13 Physical Links
- 5 Logical channels



Components Utilized

- Putty
 - Openflow
 - POX Controller
 - WIRESHARK
-

POX Controller

- provide an efficient and easy environment to test sdn networks
- Developed using python
- Uses openflow 1.0 version
- Designed primarily for the OpenFlow protocol
- Uses Event Driven Model

Different POX Modules used

L2 LEARNING SWITCH MODULE
(FORWARDING.L2_LEARNING)

SPANNING TREE PROTOCOL
(OPENFLOW.SPANNING_TREE --NO FLOOD)

OPENFLOW DISCOVERY AND HOST TRACKING
(OPENFLOW.DISCOVERY_HOST_TRACKER)

PACKET DUMPING
(INFO.PACKET_DUMP)

Demo



Part 1
Topology Discovery



Part 2
Failure Detection & Recovery

Test plan

- Initial Setup and Baseline Verification.
- Failure Simulation and Detection Tests.
- Multiple Failure Scenario Tests.
- Fail-back Tests

What we would have Done Differently !

- **Fine-Tune Logging and Debugging**
- **Performance Monitoring**
- **Error Handling**
- **Different algorithm**

Conclusion

References

- "MININET WALKTHROUGH"

[HTTP://MININET.ORG/WALKTHROUGH/](http://mininet.org/walkthrough/)

- "INSTALLING POX — POX MANUAL CURRENT DOCUMENTATION."

[HTTPS://NOXREPO.GITHUB.IO/POX-DOC/HTML/#STATISTICS-EVENTS](https://noxrepo.github.io/POX-DOC/HTML/#STATISTICS-EVENTS)