

SDN based Network Management in Emulated Environment

Submitted by: Harshal Rajan Vaze

Matriculation no.: 1269879

First examiner: Prof. Dr. Ulrich Trick

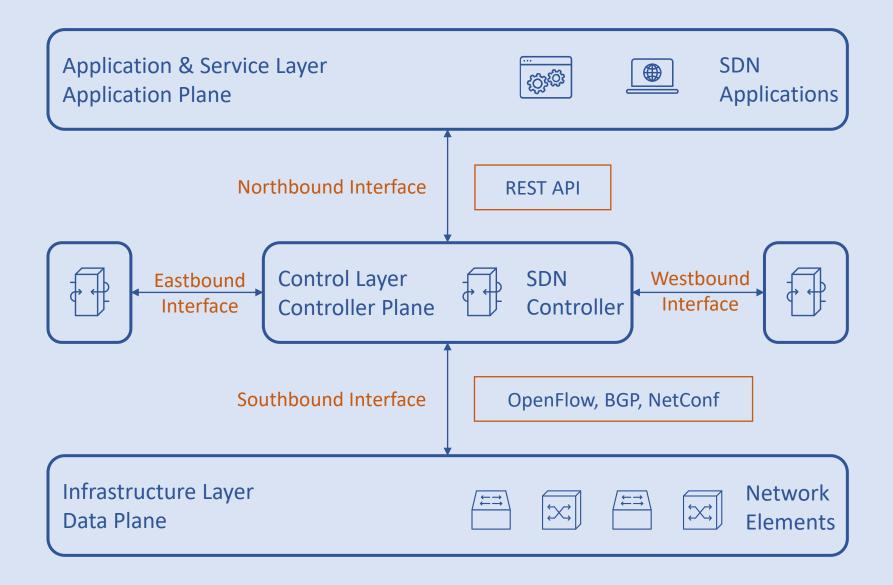
Second examiner: Prof. Dr. Armin Lehmann

External supervisor: Dr. Peter Gröschke

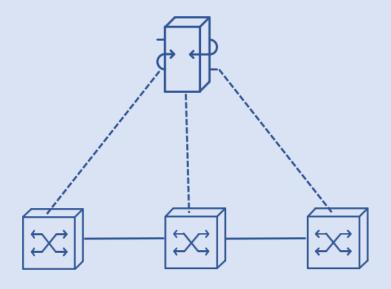
Agenda

- 1. SDN Architecture
- 2. Different SDN Architectures
- 3. Open-source SDN Controllers
- 4. Open vSwitch
- 5. OpenFlow
- 6. Creation of Flow rules
- 7. Creations of Intents
- 8. Use-cases

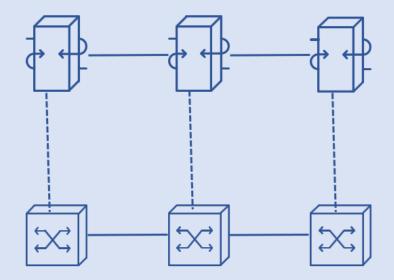
SDN Architecture



Different SDN Architectures (1/3)



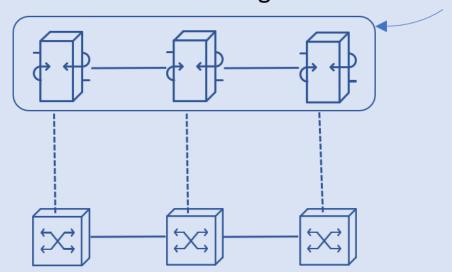
a. Physically-centralised architecture



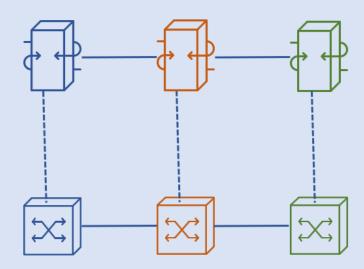
b. Physically-distributed architecture

Different SDN Architectures (2/3)

Act like a single controller for network

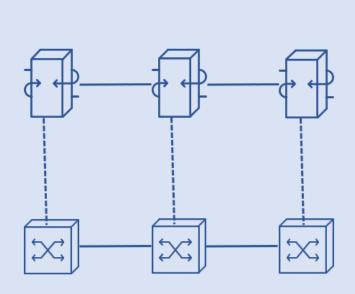


c. Physically-distributed & Logically-centralised architecture

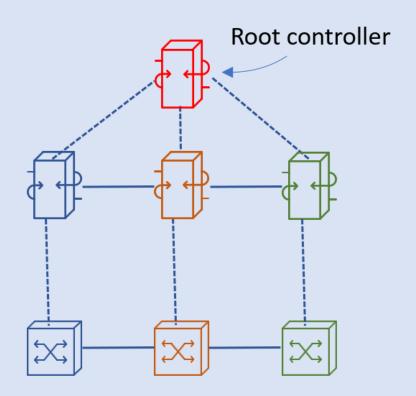


d. Physically-distributed & Logically-distributed architecture

Different SDN Architectures (3/3)



e. Flat architecture



f. Hierarchical architecture

SDN Network Components

- Different components by different communities
- SDN Controllers
 - 1. ONOS
 - 2. OpenDaylight
 - 3. Ryu
- Open vSwitch
- OpenFlow protocol
- Emulated environments

ONOS Controller



Applications
North-bound (Consumer) API
Core (Modules, Subsystems)
South-bound (Provider) API
Providers (Link, Flow, Device, Host)
Protocols
Network Elements

OpenDaylight Controller



Applications, Orchestration

North-bound API

OpenDaylight

Platform Services

- AAA
- Data Exchange
- Infrastructure utilities

Network Services

- LISP service
- Group-based policy
- Application layer traffic optimization

Built-in Applications

- Layer 2 Switch
- Virtual Tenant Network

Platform (MD-SAL, YANG tools)

South-bound API

Network Elements

Ryu Controller



Applications, Operator, OpenStack cloud orchestration

North-bound API

Ryu Framework

Built-in Applications: tenant isolation, layer2 switch

Libraries: topology discovery, OpenFlow REST

OpenFlow protocols parser/ serializer

OpenFlow, OF-config

Non-OpenFlow protocols parser/ serializer

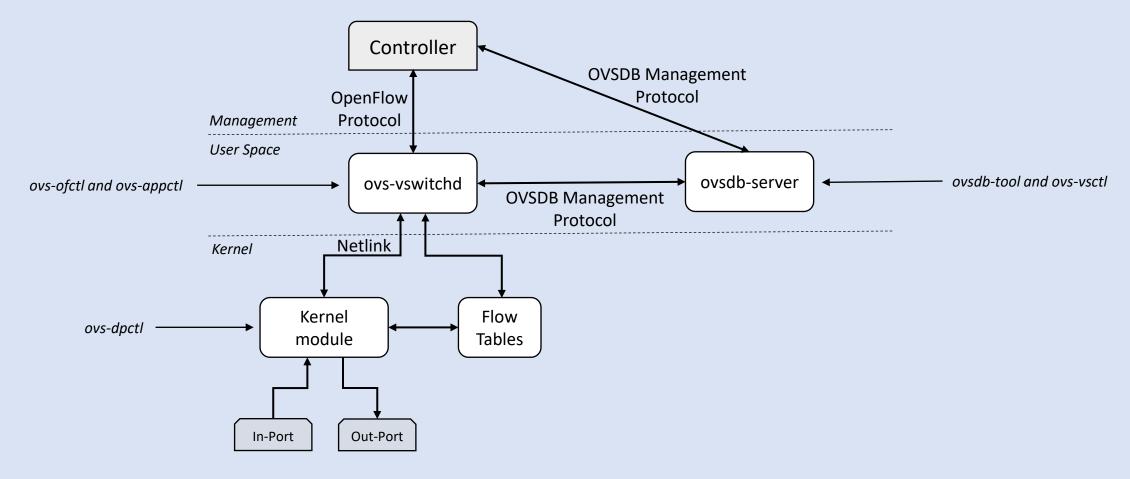
NetConf, NetFlow, VRRP

South-bound API

Network Elements

Open vSwitch

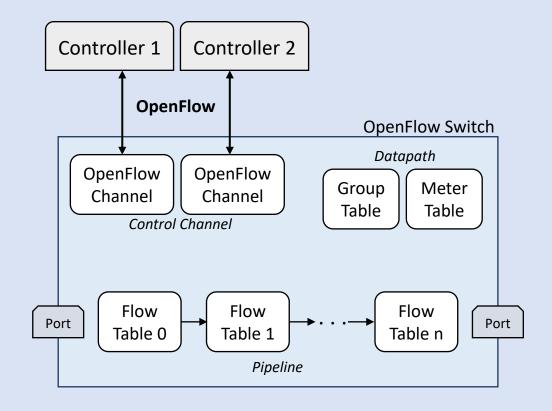




OpenFlow Protocol



- SBI protocol
- 6 major releases
- Main components
 - Flow table
 - Group table
 - Meter table
 - Control channel
- Three Message types:
 - 1. Controller-to-switch
 - 2. Asynchronous
 - 3. Symmetric



SDN network emulators

❖ GNS3

- 1. Client part: The GNS3-all-in-one software (GUI)
- 2. Server part: The GNS3 virtual machine (GNS3 VM)
- ➤ Advantages of GNS3:
- Implements easy to access network topology.
- Pre-configured and optimised appliances available to simplify deployment.
- Availability of in-built software packages such as Wireshark and Putty.
- Supports sharing of implemented network including the device configuration.
- ➤ Disadvantages of GNS3:
- Limitation of CPU and memory resources provided by the host machine.
- The software images of the network devices are required to be purchased and downloaded by the user.

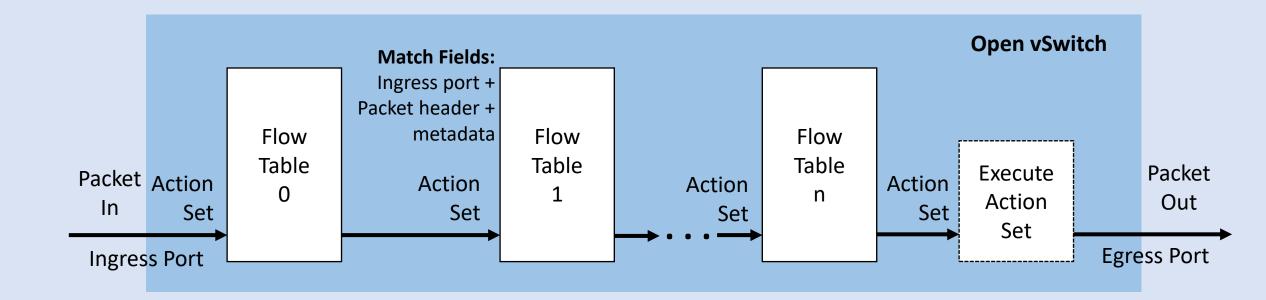
Mininet

- > Advantages of Mininet:
- It is very fast and takes very little time for booting.
- It is easy to install and use.
- It saves money because the emulators are cost-effective instead of testing with hardware devices.
- It is also very easy to connect with other network devices.
- > Disadvantages of Mininet:
- Deployed network elements cannot exceed the CPU and bandwidth available on host machine.
- Single server implementation gave rise to the performance problems.
- All Mininet hosts share the host file system and PID space, a careful assessment of running daemons is required for process of debugging.

Flow Rules

- Forwarding flow rules
- Application commands >> Flow rules
- Creation & Installation
 - 1. Open vSwitch CLI
 - 2. ONOS REST API
 - 3. ONOS CLI

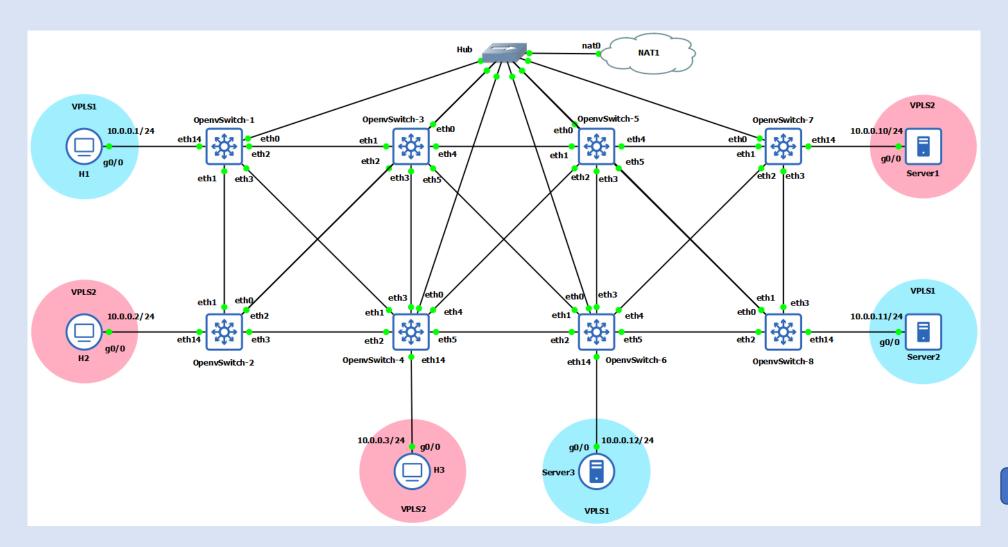
Packet transmission through flow tables within an Open vSwitch



Creation of Intents

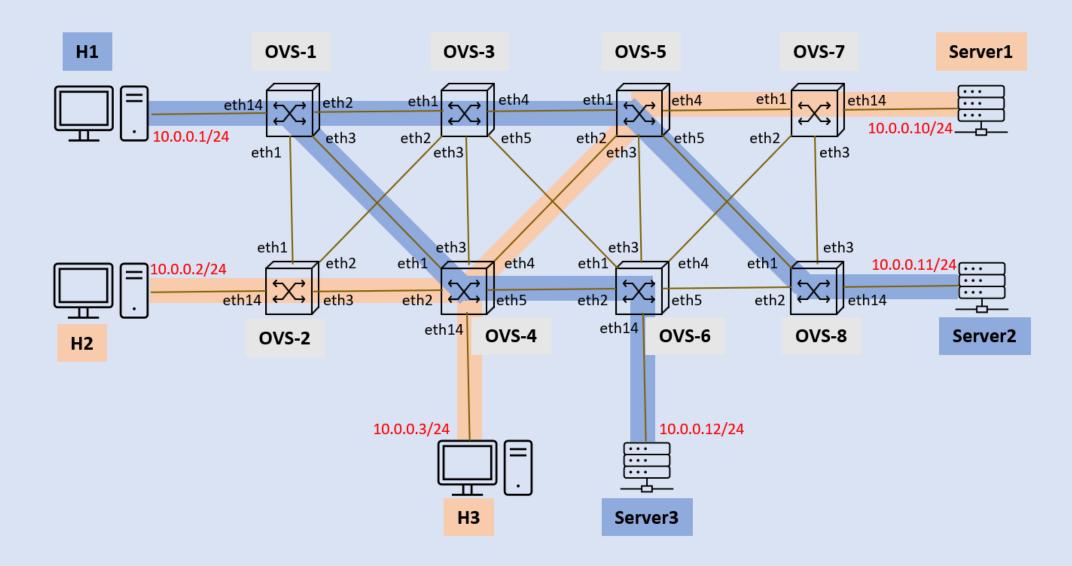
- Policy-based directives
- Intent Framework is a subsystem
- Application commands >> Flow rules
- Creation & Installation
 - 1. ONOS CLI
 - 2. ONOS GUI
 - 3. ONOS REST API

Use Case-1: Testing the ONOS Controller with Isolated Layer 2 Overlay Networks

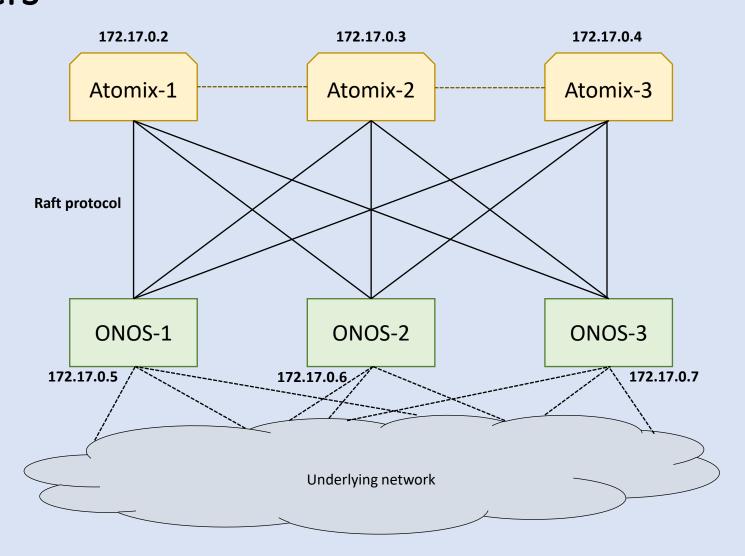




Created Layer 2 Overlay Networks

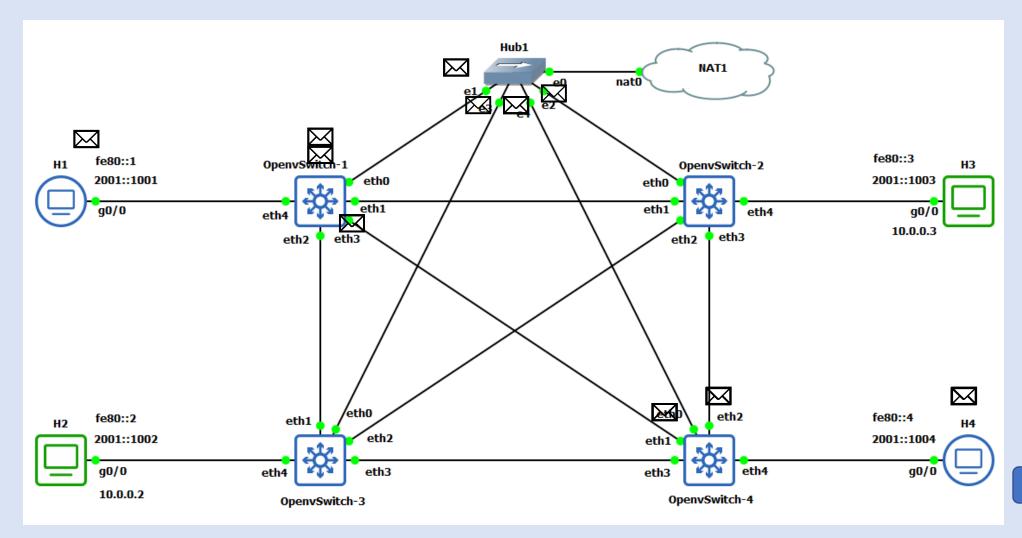


Use Case-2: Testing the Network with Multiple ONOS Controllers



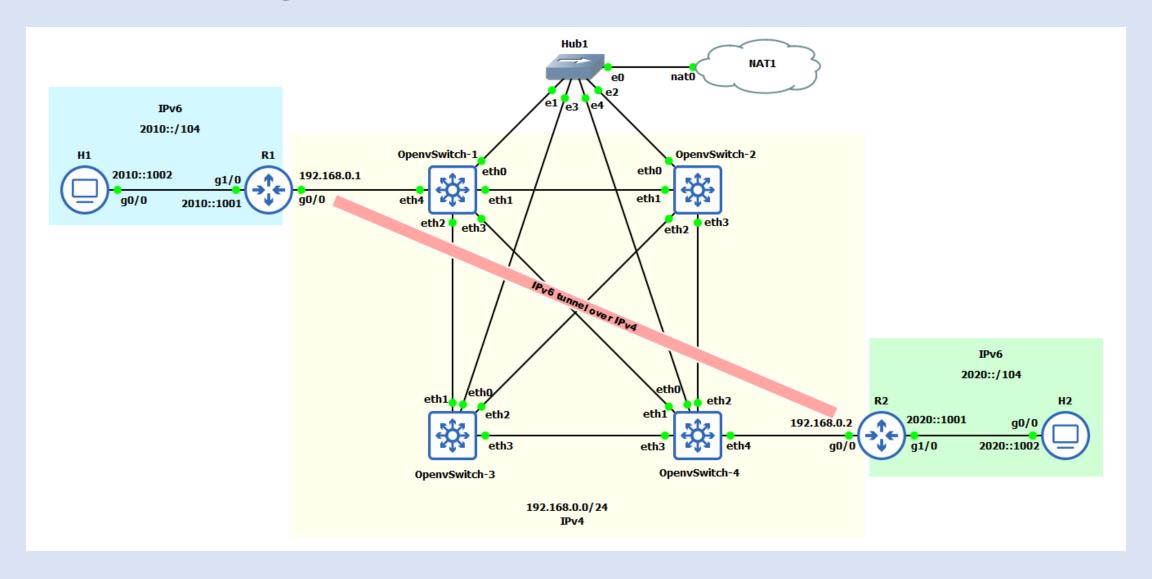


Use Case-3: Testing the ONOS Controller with IPv6 Addressing

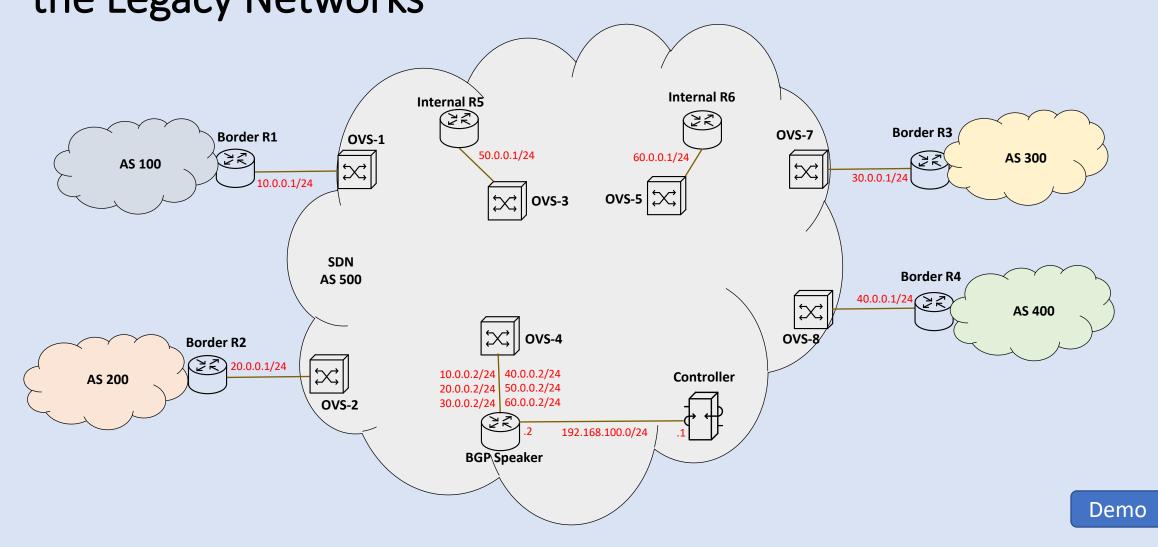


Demo

IPv6 tunneling over IPv4 network



Use Case-4: Integrating Software-defined Network with the Legacy Networks



Questions?

Thank you!