FRANKFURT UNIVERSITY OF APPLIED SCIENCES

Master Thesis

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

Content

1 Use Cases

1.1 Use case-1: Basic Network Architecture

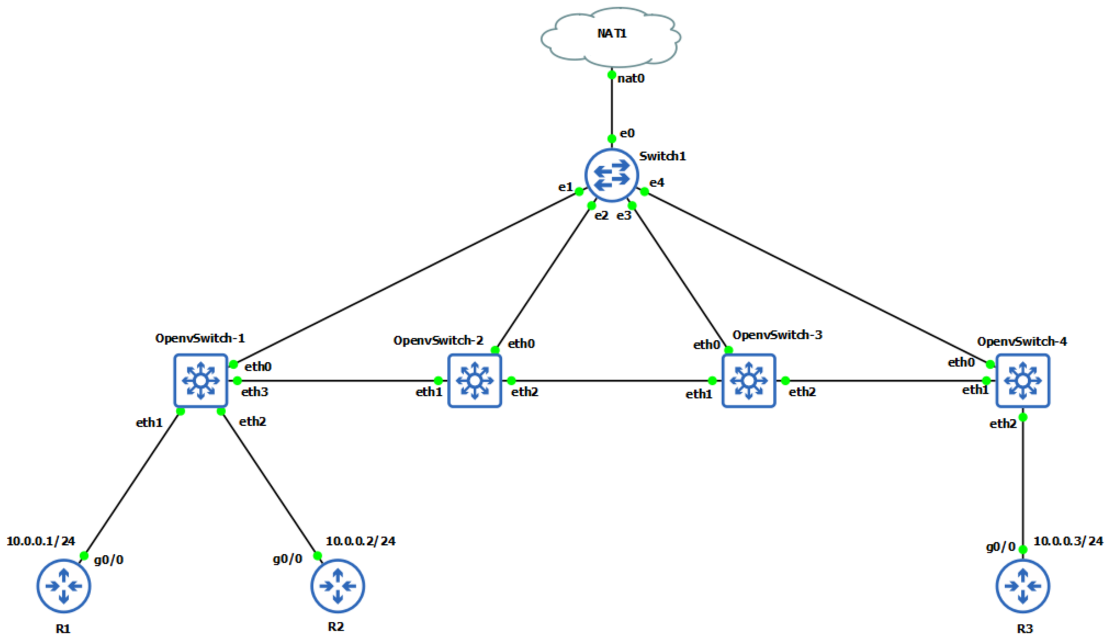
1.1.1 Introduction

A simple topology with four Open vSwitches and routers was created in the GNS3 application. For testing purpose of the routers are deployed at the far ends of the topology.

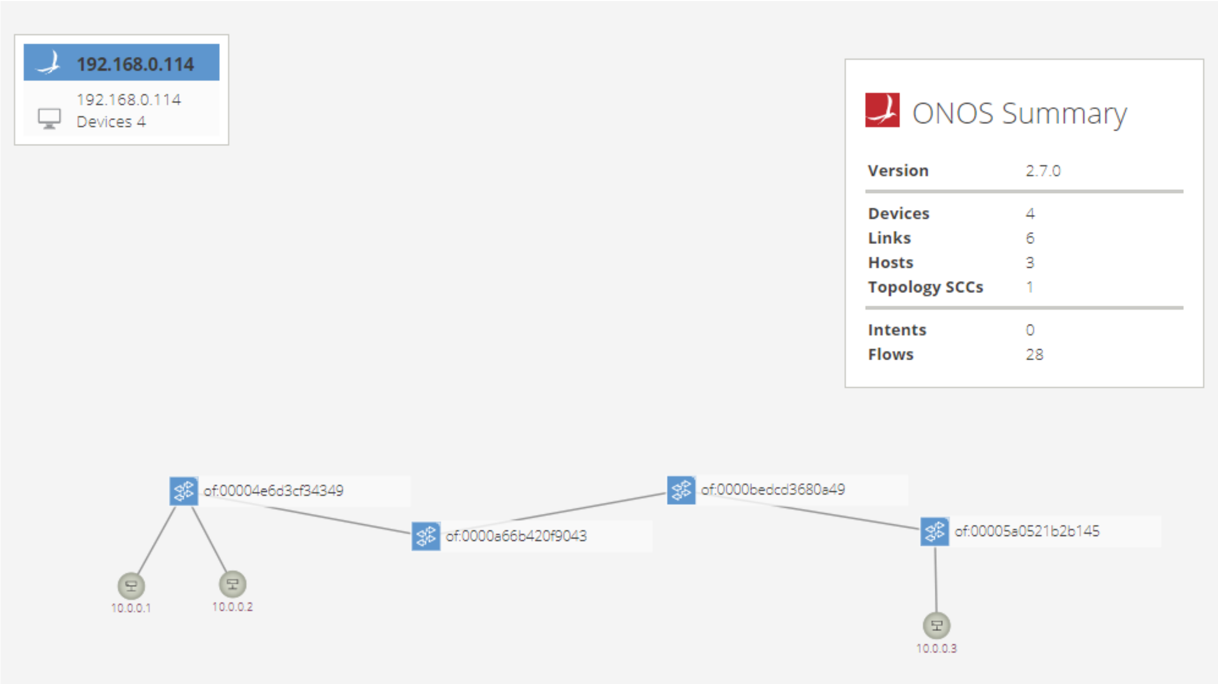
To connect any network device running in the GNS3 software with the device outside the GNS3 software, i.e. in local machine or in Internet, a NAT interface is used. In this scenario, a SDN controller is installed on a virtual machine outside the GNS3 application and hence a NAT interface is used to connect the Open vSwitches with the SDN controller. A L2 Switch or Hub can be used so that all devices can be directly connected to the NAT interface with just single interface. SDN controller used here is ONOS controller.

Before connecting the Open vSwitches to the SDN controller, OpenFlow protocol version (here OpenFlow version 1.4) needs to be specified.

Bridge br0 is the management interface on the Open vSwitch which accepts the configuration commands from the SDN controller. To connect this interface to the SDN controller, external IP address of the SDN controller (here, 192.168.0.114) and port number of OpenFlow protocol needs to be specified (here, 6653).



Topology created in the GNS3 with different Network devices



Topology view on the ONOS controller GUI

In this sub-chapter further explanation about Devices, Hosts, Ports of devices, Links, Packet processors and Applications required by ONOS to process the packet will be discussed.

1.1.2 Flows

In this sub-chapter following points will be discussed,

- Creating Flows for traffic
- Different Flow configuration methods
  - Through configuration API
  - Open vSwitch CLI
  - ONOS CLI

1.1.3 Intents

The Intent Framework is a subsystem that allows applications to specify their network control desires in form of policy rather than mechanism and these policy-based directives are *Intents*.

In this sub-chapter following points will be discussed,

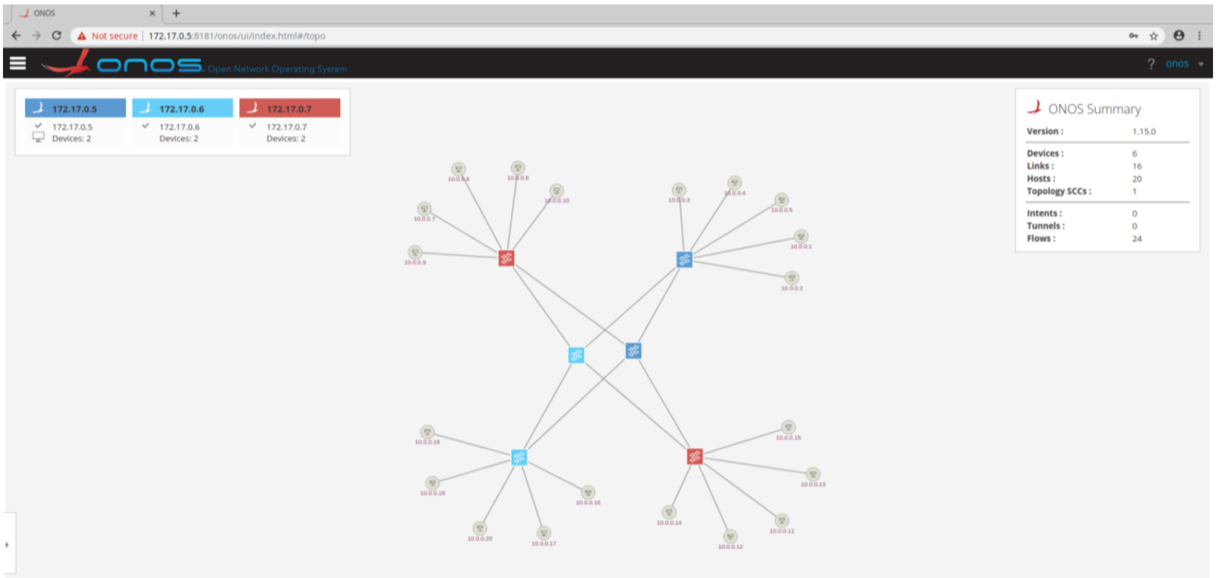
- Creating Intents for specific traffic between devices
- Different Intent configuration methods
  - Through configuration API
  - ONOS CLI
  - Through ONOS GUI

1.2 Use case-2: Testing the network with multiple Controllers

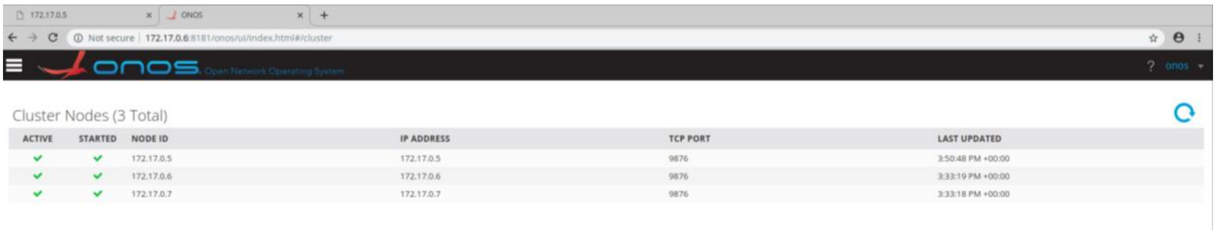
1.2.1 Introduction

In this Use case multiple SDN controllers were deployed to control the devices in the same network. A Spine-Leaf topology was created with three controllers, six devices and multiple hosts. In the following figure, three controllers can be observed in three different colours (navy blue, blue & red) having three different IP addresses (172.17.0.5, 172.17.0.6 & 172.17.0.7). Cluster of these three SDN controllers was created to act together as a unified and coherent distributed system.

The topology observed in the following figure was created in the Mininet environment and the SDN controllers were deployed as the Docker containers.



Topology view on the ONOS controller GUI

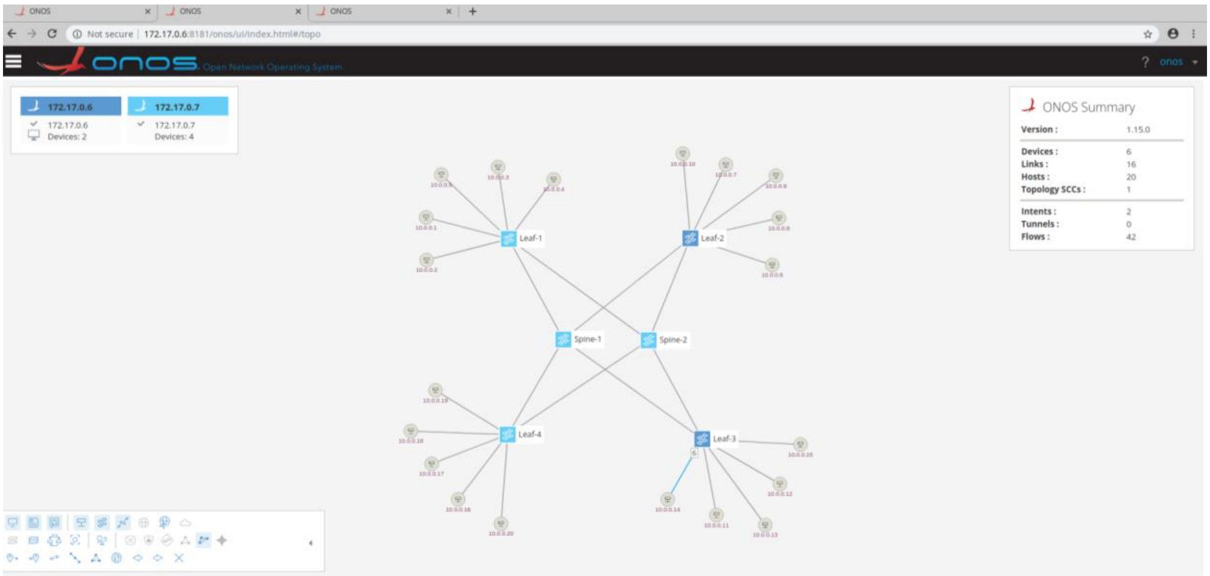


Cluster Nodes with three ONOS controllers

1.2.2 Different Controllers with their Devices

In this sub-chapter following points will be discussed,

- Distribution of devices amongst the Controllers
- Selection of Master Controller amongst the Controllers
- Failure of one Controller



Failure of one controller

1.2.3 Path Identifier

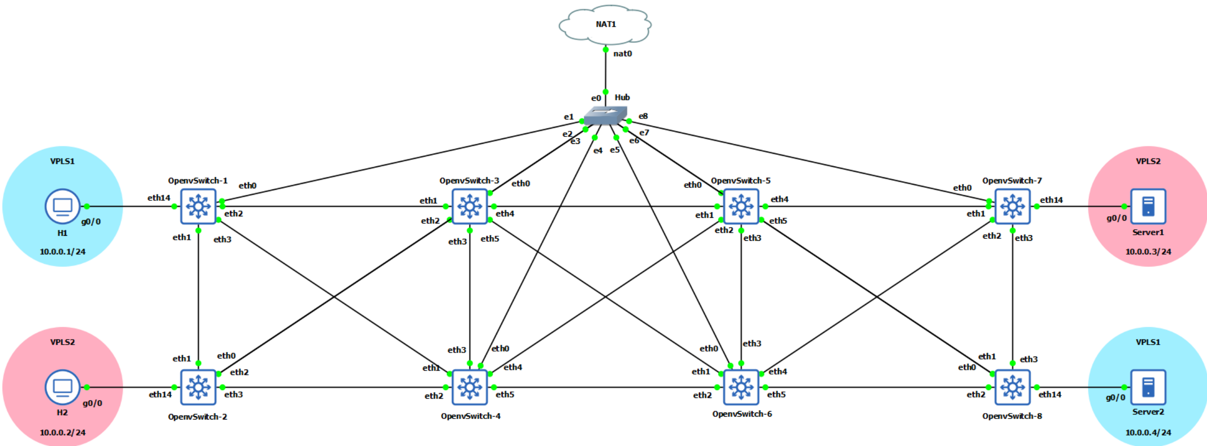
In this sub-chapter following points will be discussed,

- Links between the devices/hosts
- Path identification for traffic between the devices/hosts
- Proof and validation of functioning failover mechanisms of link between the devices

1.3 Use case-3: Testing network with isolated L2 overlay networks

1.3.1 Introduction

In this Use case virtually separate paths were created to isolate the traffic flow between the end points. For this Virtual Private LAN Service (VPLS) was implemented in this Use case. The goal was to connect multiple end-points in an OpenFlow network, creating isolated L2 broadcast overlay networks. While legacy technologies require the manual configuration of multiple devices in the network, VPLS tries to make the process easier for network operators. Hosts that get connected together can send in either untagged or VLAN tagged traffic, using either the same or different VLAN IDs. Two different VPLS were created, first, blue VPLS connecting Host H1 with Server 1 and second, red VPLS connecting Host H2 with Server 2 as observed in following figure.



Topology created in the GNS3 with different Network devices

1.3.2 Configuration

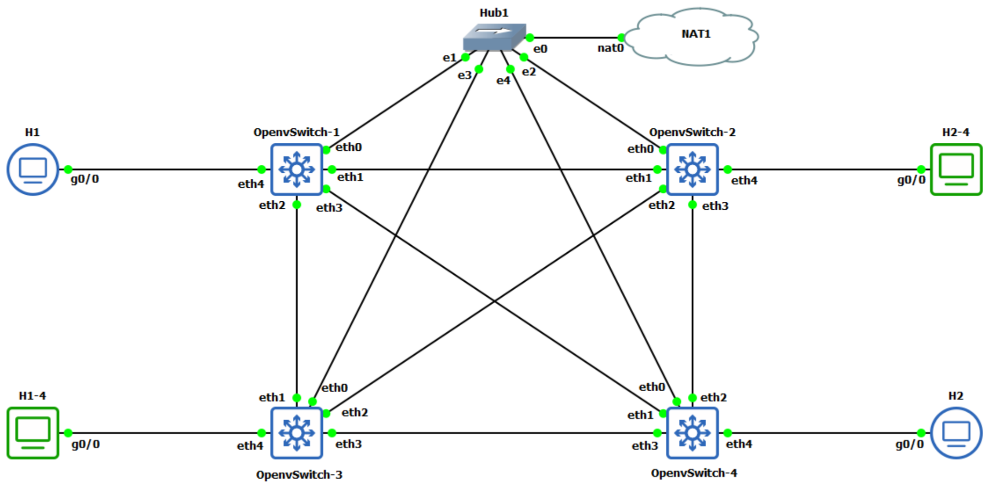
In this sub-chapter the following points will be discussed,

- Configuration of VPLS on SDN controller
- Working of VPLS
- Studying VPLS application of ONOS controller

1.4 Use case-4: Testing network with IPv6 addressing

1.4.1 Introduction

For testing the SDN controller functionality with the IPv6 addressing this Use case was implemented. In this Use case, the network was created consisting of IPv6 addresses and IPv4 addresses. The network was simultaneously tested for both the addresses. In the below figure two IPv6 Hosts are represented as H1& H2 and two IPv4 Hosts are represented as H1-4 & H2-4.

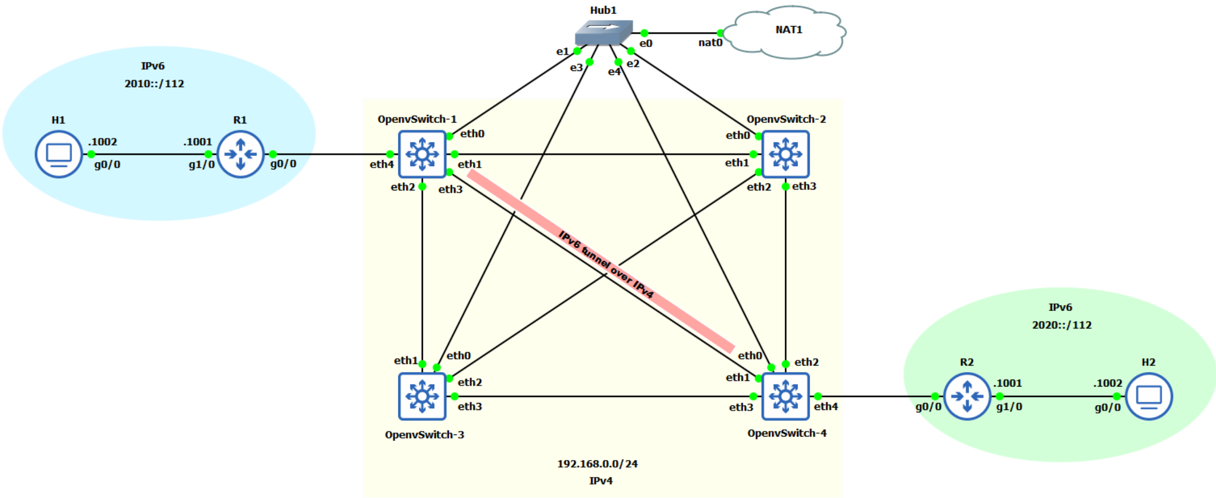


Topology created in the GNS3 with different Network devices

1.4.2 IPv6 tunnelling over IPv4

In this sub-chapter following points will be discussed,

- Creating three networks; two IPv6 networks and one IPv4 network
- Configuration of all these networks on SDN controller
- Creating IPv6 tunnel through IPv4 network
- Evaluating how packets are forwarded through OpenFlow protocol

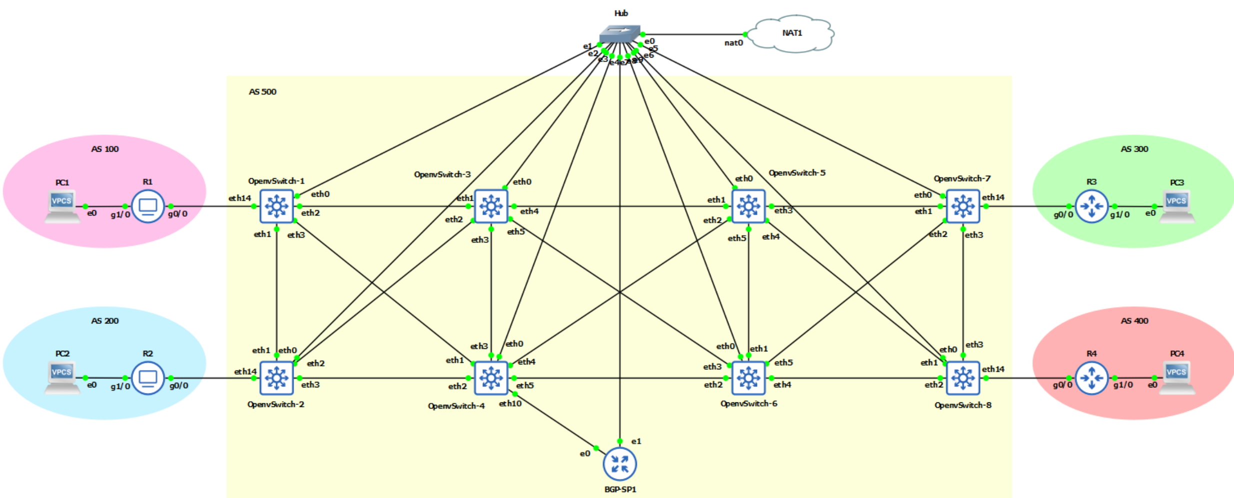


Topology created in the GNS3 with different Network devices

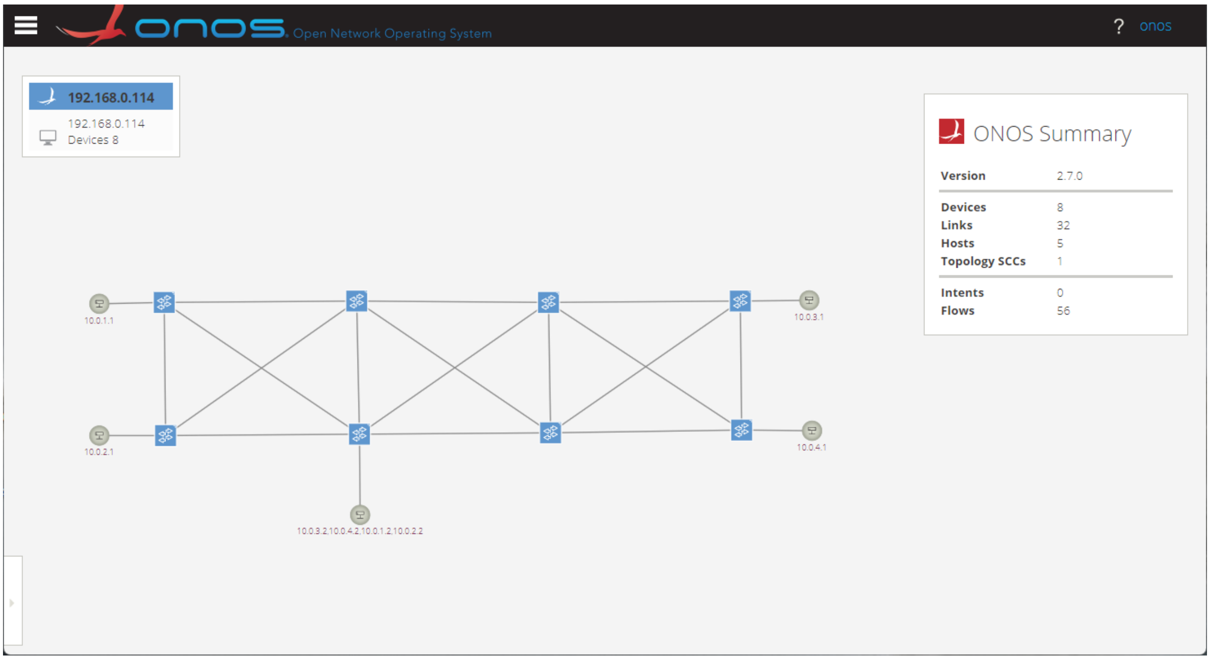
1.5 Use case-5: Integrating SDN with legacy network

1.5.1 Introduction

In this Use case SDN network was integrated into the legacy networks. Four different legacy networks were created with Border routers R1, R2, R3 & R4 in different Autonomous Systems (AS) as shown in the following figure. These Border routers were configured with BGP protocol. The SDN network was implemented in different AS than these Border routers. The goal was to integrate the SDN network into these legacy networks and study the functionality of SDN controller with the BGP protocol.



Topology created in the GNS3 with different Network devices



Topology view on the ONOS controller GUI

1.5.2 Configuration

In this sub-chapter following points will be discussed,

- Creating and configuring all these networks
- Configuration of all these networks on SDN controller
- Studying SDN-IP application of ONOS controller