EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Network Softwarization
SS 2022
Prof. Dr. M. Menth
M. Häberle, S. Lindner, D. Merling, B. Steinert

Exercise 1

May 02, 2022

Deadline: May 29, 2022, 23:59

Problem 1.1: *Introduction to Network Softwarization*

Each participant will be asked **5 questions** from the following pool of questions **on May 16, 2022 in the exercise lesson**. Giving answers to all questions is a prerequisite for getting a mark for the assignment!

Chapter 0 - Organization and Introduction

1. Explain the differences between legacy network devices (e.g., a switch or router) and the concepts of open networking or software-defined networking!

Chapter 1 - Introduction to Network Softwarization

1. What is SNMP used for? What is the purpose of an SNMP agent?
2. What is the purpose of policy-based routing?
3. Name the entities of a ForCES network!
4. What is the purpose of the Route Control Server?
5. Name the 4 layers of 4D!
6. Name an example use case for PCE!

Chapter 2 - OpenFlow: Basics

1. Explain the OpenFlow architecture! Describe the three main components and explain their relations!
2. Describe the two different paradigms (flow-based vs. aggregated and proactive vs. reactive) in OpenFlow and discuss the tradeoffs! Think about an example for each tradeoff!
3. Briefly explain the main components of an OpenFlow 1.0 and OpenFlow 1.5.1 switch!
4. Explain the differences between OpenFlow 1.0 and OpenFlow 1.5.1 regarding the concept of Flow Table(s)! Discuss advantages, disadvantages and consequences for the network operators and network hardware manufacturers!

Chapter 3 - OpenFlow Switches + Controller

1. Briefly explain the tradeoff between programmability and processing performance between general purpose processors and custom ASICs!
2. What is the difference between CAM and TCAM?
3. What is the difference between a Whitebox switch, an OpenFlow-only switch and a hybrid switch?
4. Name and briefly explain the different functions of an SDN controller!
5. Explain the purpose of an east-/westbound interface!
6. What is a northbound interface? Explain the purpose of it using the IDS example from the lecture!
7. Briefly explain the anatomy of an OpenFlow controller!

Assignment 1

1. Explain the fundamental functionality of the implementation of a MAC learning switch `simple_switch_13.py` in Ryu! You should be able to explain single lines / parts of the sourcecode!
2. How does the SDN controller differentiate between multiple OpenFlow switches?
3. Which kind of communication channel is used for OpenFlow communication? Out-of-band or in-band?
4. How is packet forwarding achieved between routers?
5. When/where can different IP-prefixes be aggregated on routers?
6. Why is packet-based anycast not suitable for TCP flows?

Problem 1.2: *Simple IP routing*

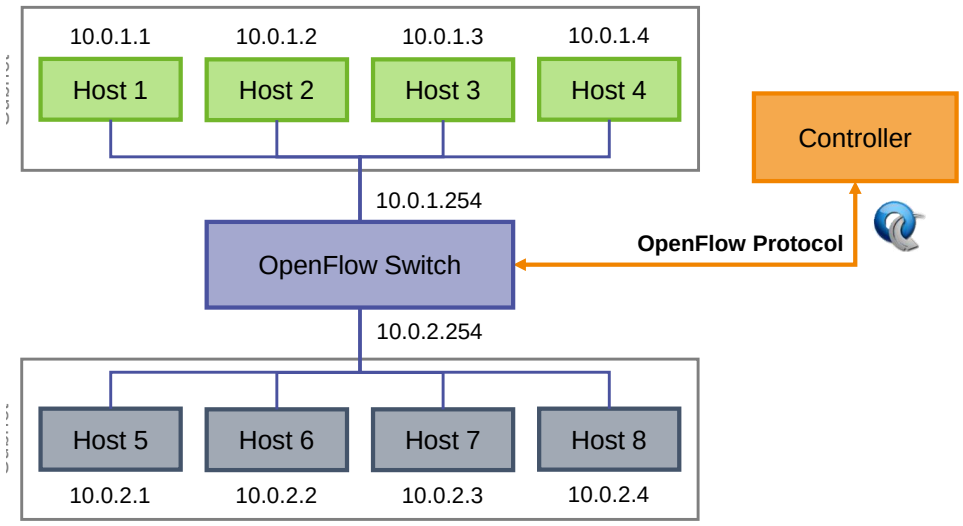

Figure 1: OpenFlow SDN setup including an OpenFlow controller, OpenFlow switch, and 8 hosts residing in two different subnets

In the first task, we introduce simple IP routing techniques to use an OpenFlow-capable switch to route between two subnets as depicted in Figure 1. Host $1-4$ are in subnet $10.0.1.0/24$, Hosts $5-8$ are in subnet $10.0.2.0/24$. The OpenFlow-capable switch acts as a simple router between both subnets, i.e., it routes packets according to their IP destination address. 15 Points

1. Open Miniedit in the NetSoft-VM and set up the topology! Ensure that you have activated the following properties!
   - Edit/Preferences: Start CLI and OpenFlow 1.3
   - Controller Properties: Controller Type - Remote Controller
2. Set the default gateway to $10.0.1.254$ for Host $1-4$ and $10.0.2.254$ for Host $5-8$!
3. Expand the `example.py` application by introducing simple IP routing! Provide means to forward packets between the two subnets according to IP destination addresses! Ensure connectivity by using the `ping` command!

Upload your final implementation to Moodle. Include your controller implemenetation and your Mininet network (don't export it, use "save" instead).

Problem 1.3: *IP Router for IPv4*

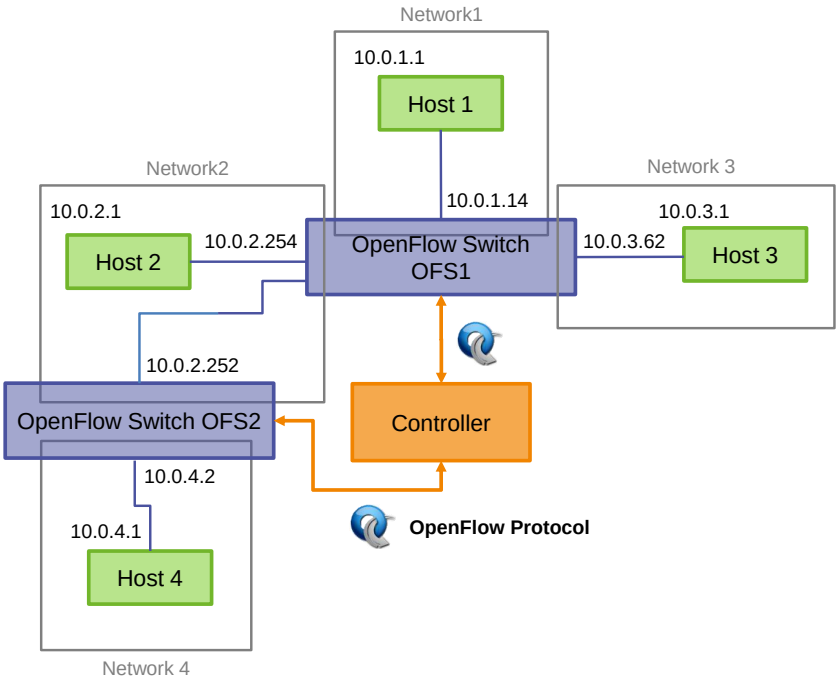

Figure 2: OpenFlow SDN setup including an OpenFlow controller, two OpenFlow switches and four hosts.

In the following, we implement a simple network topology as depicted in Figure 2. It consists of four IP networks that are connected via OpenFlow switches that are controlled by one central SDN controller. We implement an SDN controller application to introduces IP routing to the network topology.
Use the controller from exercise 1.2 as a starting point. You will have to extend it in a way so that it can control 2 switches. 10 Points

1. Open Miniedit in the NetSoft-VM and set up the topology! Ensure that you have activated the following properties!
   - Edit/Preferences: Start CLI and OpenFlow 1.3
   - Controller Properties: Controller Type - Remote Controller
2. The IPs for the different networks are specifed in the following file:

```
[Network1]
IPv4: 10.0.1.0/28

[Network2]
IPv4: 10.0.2.0/24
```

```
[Network3]
IPv4: 10.0.3.0/26

[Network4]
IPv4: 10.0.4.0/30
```

3. The SDN switches act as gateways and therefore hold the highest IP address of the networks connected to them. The gateways should answer ICMP request from both – external hosts and hosts in their corresponding network.

4. The connectivity between the different networks is configured in the following file:

```
[Network1]
Network2: direct
Network3: direct
Network4: Network2

[Network2]
Network1: direct
Network3: direct
Network4: OFS2

[Network3]
Network1: direct
Network2: direct
Network4: Network2

[Network4]
Network1: Network2
Network2: OFS1
Network3: Network2
```

5. Implement routing for IPv4 based longest prefix matching (LPF)! Ensure that all hosts can reach each other!

Upload your final implementation to Moodle. Include your controller implemenetation and your Mininet network (don't export it, use "save" instead).

Problem 1.4: *IP Anycast*

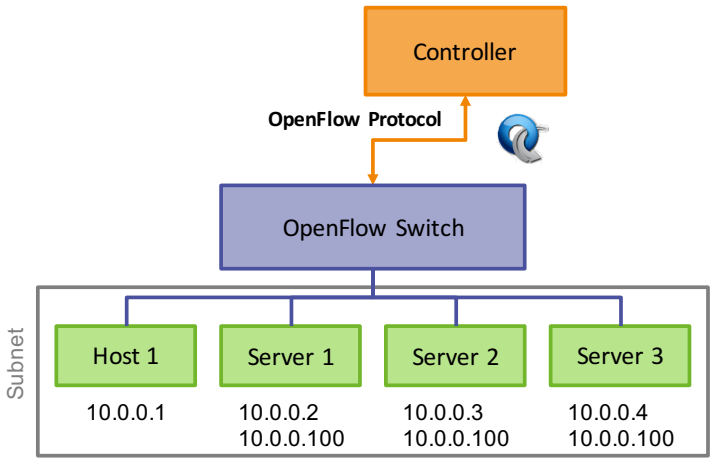

Figure 3: OpenFlow SDN setup including an OpenFlow controller, OpenFlow switch, and 4 hosts residing in one subnet.

In the next tasks, we introduce IP anycast techniques for an OpenFlow switch. IP anycast makes a group of different hosts reachable under one IP adress where one member of the group answers. This mechanism is, e.g., useful for load-balancing or resilience purposes.

10 Points

Packet-based IP Anycast
In packet-based anycast, the router or switch performs anycast forwarding per packet individually. Each packet is either forwarded in a random or round-robin way to one of the servers in the anycast group.

1. Set up the new topology depicted in Figure 3 in Miniedit! Ensure that you have activated the following properties:
   - Edit/Preferences: Start CLI and OpenFlow 1.3
   - Controller Properties: Controller Type - Remote Controller
2. Implement packet-based IP anycast with a round-robin mechanism iterating over the different servers. You can test your anycast implementation like following:
   - Start a UDP listening process on all servers: *nc -l ANYCAST-IP -p 12345 -u -k*.
   - Send packets from the client to the anycast group: *nc -u ANYCAST-IP 12345*.
   - Incoming traffic should be visible on the different servers.

Flow-based IP Anycast 15 Points

Packet-based anycast is not feasible for connection-oriented transport protocols like TCP as the connection state is not shared between the servers in the anycast group. With flow-based anycast, only the first packet of each flow to an anycast IP is forwarded to a random server in the anycast group. All following packets of that flow are then forwarded to the same server. To ensure this behavior, the router performs a hash over the 5-tupel (Src-IP, Src-Port, Dst-IP, Dst-Port, Protocol-ID) identifying the flow and forwards all packets with the same hash to the same server. That allows a TCP connection to be established with one of the servers in the anycast group.

1. Set up the new topology depicted in Figure 3 in Miniedit! Ensure that you have activated the following properties:
   - Edit/Preferences: Start CLI and OpenFlow 1.3
   - Controller Properties: Controller Type - Remote Controller
2. Implement flow-based IP anycast! You can test your anycast implementation like the following:
   - Start a TCP listening process on all servers: *nc -l ANYCAST-IP -p 12345 -k*.
   - Send packets from the client to the anycast group: *nc ANYCAST-IP 12345*.
   - Flow-based forwarding to different servers should be visible.

Upload your final implementation to Moodle. Include your controller implemenetation and your Mininet network (don't export it, use "save" instead).

Total: 50 Points