Data Center Networking Technology

Project 4

2022.04.26

Yu-Ho Chiang, Chieh-Ju Yu

chiang90102@gmail.com, lucy4116411.cs09@nycu.edu.tw

Outline

- Project Info
- Project Content
- Multi-tenancy
- Project Proposal
- Step-by-Step Instruction
- Project Report
- Demo

Project Info

Goal:

- In this project, student will learn what Network Virtualization is, how to achieve Network Virtualization and practice how to use SDN and Mininet

Project 4 Assigned: 2022.04.26
Project 4 Proposal deadline: 2022.05.10
Project 4 Report deadline: 2022.05.24
Project 4 Demo: 2022.05.24

Project Content

- In this project, you have to create a virtual Data Center network by Mininet and make every tenant has their own isolated network

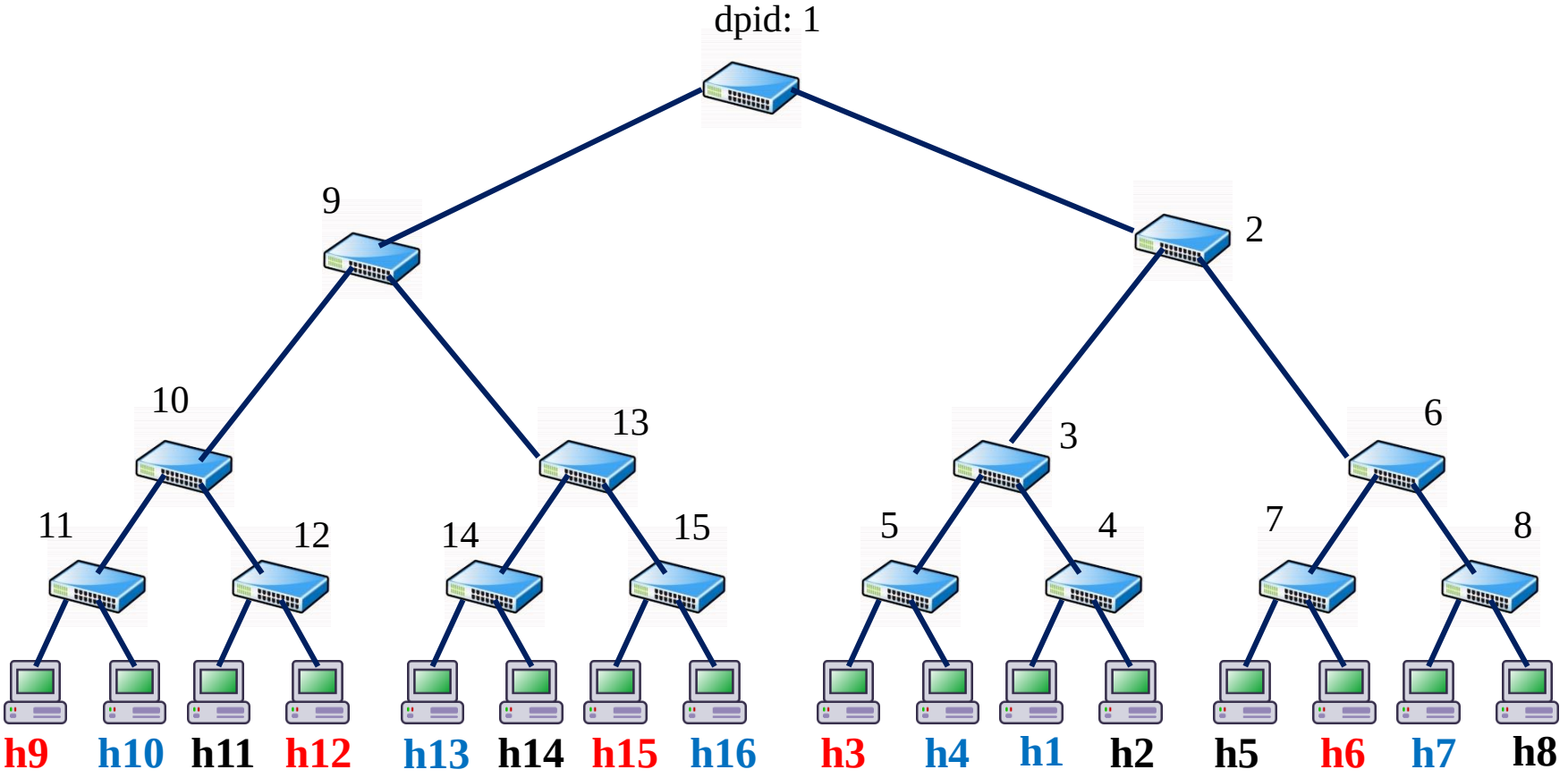
- This project has **two parts**, the first part requires every student to **propose your design** and explain how it works (2 weeks)

- As for the second part of project, students have to **implement their design** with SDN controller RYU and Mininet (2 weeks)

Multi-tenancy

- **Multi-tenancy** is an architecture in which a single instance of a software application serves multiple customers
- Each customer is called a tenant, and a tenant is a group of users who share a common access with specific privileges to the software instance
- Multi-tenancy is the fundamental technology that clouds use to share IT resources cost-efficiently and securely, just like in an apartment building-in which many tenants cost-efficiently share the common infrastructure of the building but have walls and doors that give them privacy from other tenants

Topology



Tenant 1: h9, h12, h15, h3, h6

Tenant 2: h10, h13, h16, h4, h1, h7

Tenant 3: h11, h14, h2, h5, h8

Proposal

Propose your design and explain how your design work to implement the multi-tenancy network in pervious page
Upload your proposal to new e3 platform by 05/10 13:20

The proposal should include:

1. Your design and explanation of how your design work
   E.g.: 1.Use flow table to isolate different tenant traffic by source address
   2. Use flow table to implement the VLAN technique
2. What are the benefits of Multi-tenancy in Data Center?

File name should be “Project-4_ Proposal_X.pdf ”
(E.g.: Project-4_ Proposal_310123456.pdf)

Step-by-Step Instruction

Implement your design with SDN controller RYU and Mininet

Step 1: Create the topology in Mininet

Use the following command to build the topology:

sudo mn --controller=remote,ip=127.0.0.1 --topo tree,depth=4 --switch default,protocols=OpenFlow13 --mac --arp

With the "--mac" command, host mac address will be fixed. This will make your design easier

EX: The mac address of h1 will be 00:00:00:00:00:01
The mac address of h11 will be 00:00:00:00:00:0b
The mac address of h16 will be 00:00:00:00:00:10 (HEX)

With the "--arp" command, switch will not broadcast ARP to get corresponding IP, and MAC address pair

Step-by-Step Instruction

Step 2: Implement your multi-tenancy in controller

1. You can modify the code based on previous project

2. Add the flows when the switch connect to the controller or when controller receive packet-in

3. Use the function “add_flow( )” in the “simple_switch_13.py” to add the flows in flow table

Step-by-Step Instruction

Step 2: Implement your multi-tenancy in controller

How to add the flows?

There are 4 parameters you have to send to the add_flow( ) function

self.add_flow(datapath, priority, match, actions)

1. "Datapath" represents which switch you want to send the message
   You can get the switch's datapath in the variable self.datapaths

2. Priority is the flows priority

Step-by-Step Instruction

Step 2: Implement your multi-tenancy in controller

How to add the flows?

3. For “match” parameter, you should use the structure as below:
   ev.datapath.ofproto_parser.OFPMatch(eth_src='00:00:00:00:00:0c')

you can use other variables in the match field
Ex: in_port , eth_dst , vlan_vid ……

You can reference the simple_switch_13.py or the link:
http://ryu.readthedocs.org/en/latest/ofproto_v1_3_ref.html#flow-match-structure

Step-by-Step Instruction

Step 2: Implement your multi-tenancy in controller

How to add the flows?

4. For “actions” parameter, you should use the structure as below:
[ev.datapath.ofproto_parser.OFPActionOutput(1)]

It means output port is port 1

You can add more than one actions and use “ , ” to separate actions

If the action sets have no output actions, then the packet should be drop

You can reference the simple_switch_13.py or the link:
http://ryu.readthedocs.org/en/latest/ofproto_v1_3_ref.html#action-structures

You can use “pingall” to check your multi-tenancy
The result should be like this:

```
mininet> pingall
*** Ping: testing ping reachability
h1 -> X X h4 X X h7 X X h10 X X h13 X X h16
h2 -> X X X h5 X X h8 X X h11 X X h14 X X
h3 -> X X X X h6 X X h9 X X h12 X X h15 X
h4 -> h1 X X X X h7 X X h10 X X h13 X X h16
h5 -> X h2 X X X X h8 X X h11 X X h14 X X
h6 -> X X h3 X X X X h9 X X h12 X X h15 X
h7 -> h1 X X h4 X X X X h10 X X h13 X X h16
h8 -> X h2 X X h5 X X X X h11 X X h14 X X
h9 -> X X h3 X X h6 X X X X h12 X X h15 X
h10 -> h1 X X h4 X X h7 X X X X h13 X X h16
h11 -> X h2 X X h5 X X h8 X X X X h14 X X
h12 -> X X h3 X X h6 X X h9 X X X X h15 X
h13 -> h1 X X h4 X X h7 X X h10 X X X X h16
h14 -> X h2 X X h5 X X h8 X X h11 X X X X
h15 -> X X h3 X X h6 X X h9 X X h12 X X X
h16 -> h1 X X h4 X X h7 X X h10 X X h13 X X
*** Results: 70% dropped (70/240 received)
mininet>
```

Step-by-Step Instruction

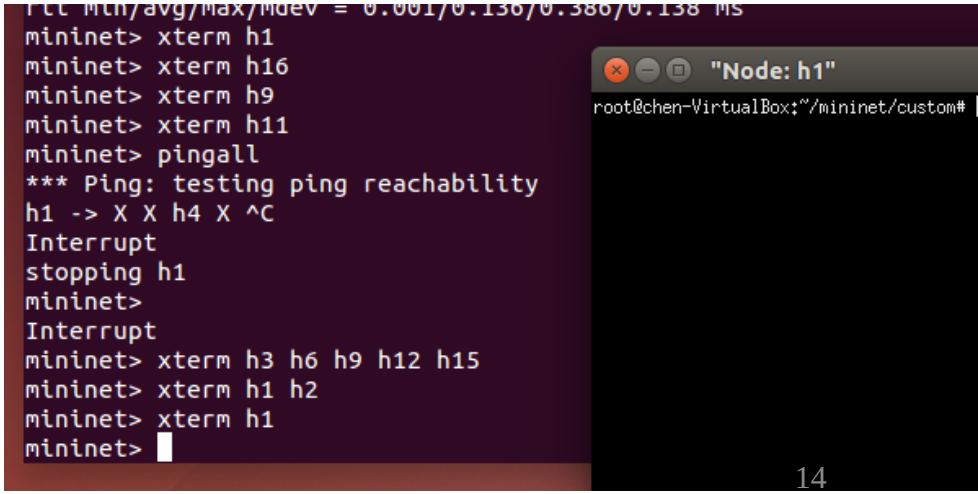
Step 3: Run test program to verify multi-tenancy policy

The host in the same tenant should not receive the broadcast traffic from other tenant

Run the UDP socket program in the host of Mininet

You can use "xterm" to open the host's terminal in Mininet

Ex: xterm h1 h4 h7 h8 h9



Step-by-Step Instruction

Step 3: Run test program to verify multi-tenacy policy

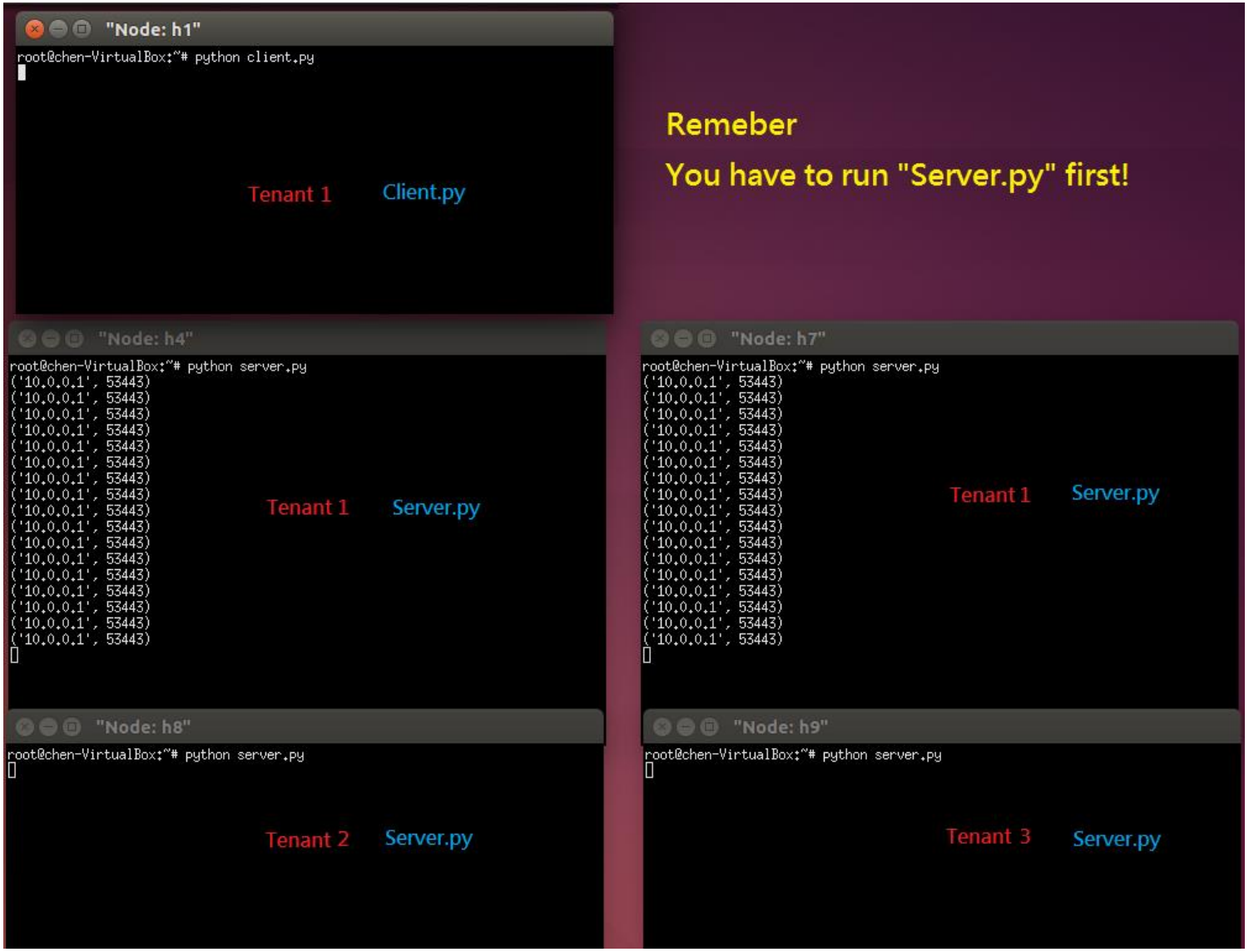
Server.py

```
# Receive UDP packets transmitted by a
broadcasting service

MYPORT = 50000

import sys
from socket import *

s = socket(AF_INET, SOCK_DGRAM)
s.bind(('', MYPORT))

while 1:
    data, wherefrom = s.recvfrom(1500, 0)
    sys.stderr.write(repr(wherefrom) + '\n')
    sys.stdout.write(data)
```

Client.py

```
# Send UDP broadcast packets

MYPORT = 50000

import sys, time
from socket import *

s = socket(AF_INET, SOCK_DGRAM)
s.bind(('', 0))
s.setsockopt(SOL_SOCKET, SO_BROADCAST, 1)

while 1:
    data = repr(0)
    s.sendto(data, ('10.255.255.255', MYPORT))
    time.sleep(2)
```

Step-by-Step Instruction



Report

Upload your report to new e3 before 05/24 13:20

Report should include:

1. Explanation of your design implementation
   a) Block diagram, flowchart to explain your implementation
   b) Screenshot of your design working
      (like the one in pp.13 and pp.16)
   c) Optional: Flow Table information & Wireshark monitoring
2. What difficulties/bottleneck do you encounter in this project ?
3. What's the advantage and disadvantage of your design?

Report name should be "Project-4_ Report_X.pdf"
(EX: Project-4_Report_310123456.pdf)

DEMO

- We will have DEMO on 05/24
- Please go to this link and fill demo time you prefer
- Email TAs, if you have any questions