Software-Defined Networking

Lab 5
Midterm Lab

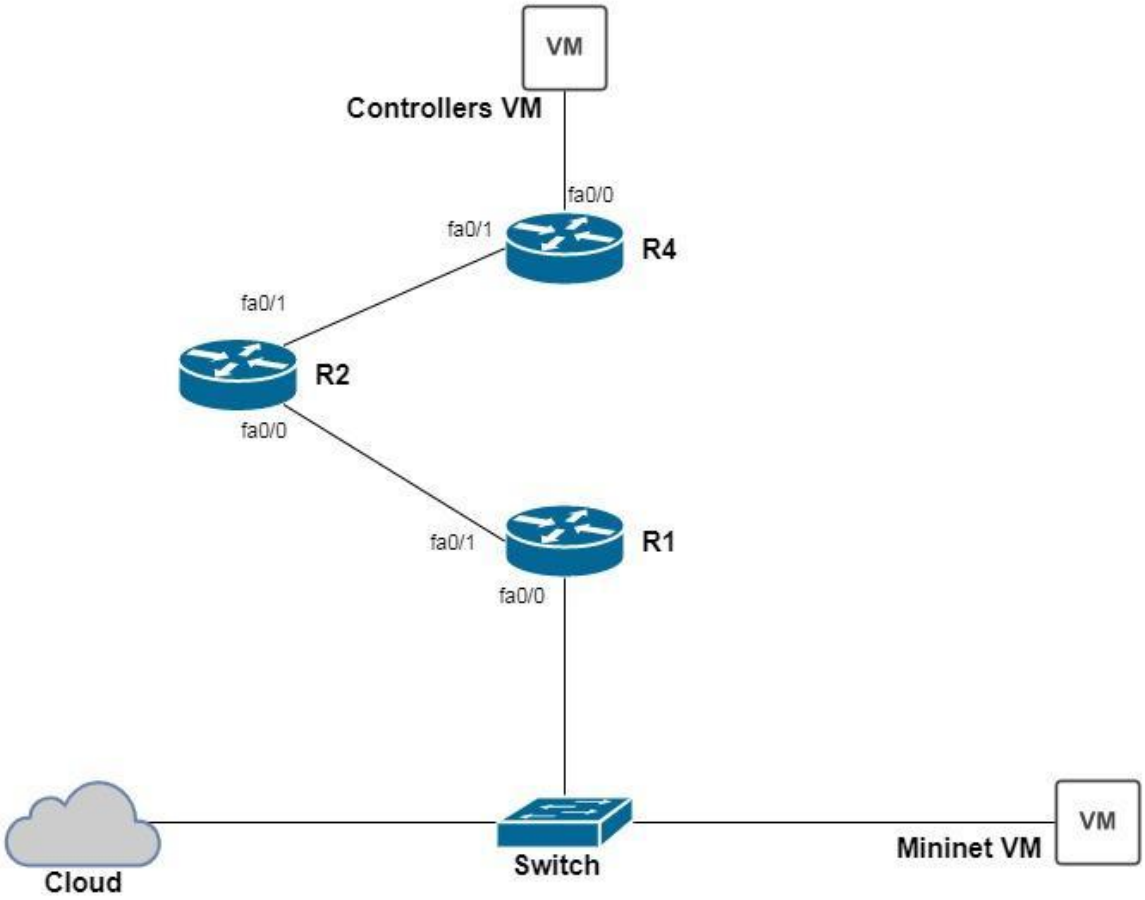
University of Colorado Boulder
Department of Computer Science

Professor Levi Perigo, Ph.D.

Summary

The objective of this lab is to recall and apply all the knowledge you have gained so far in this course. You will utilize the knowledge of traditional networking, software-defined networking, virtual switches, controllers, OpenFlow, packet capturing, GNS3 and Python. Students are encouraged to expand on the topics for additional learning and experiments.

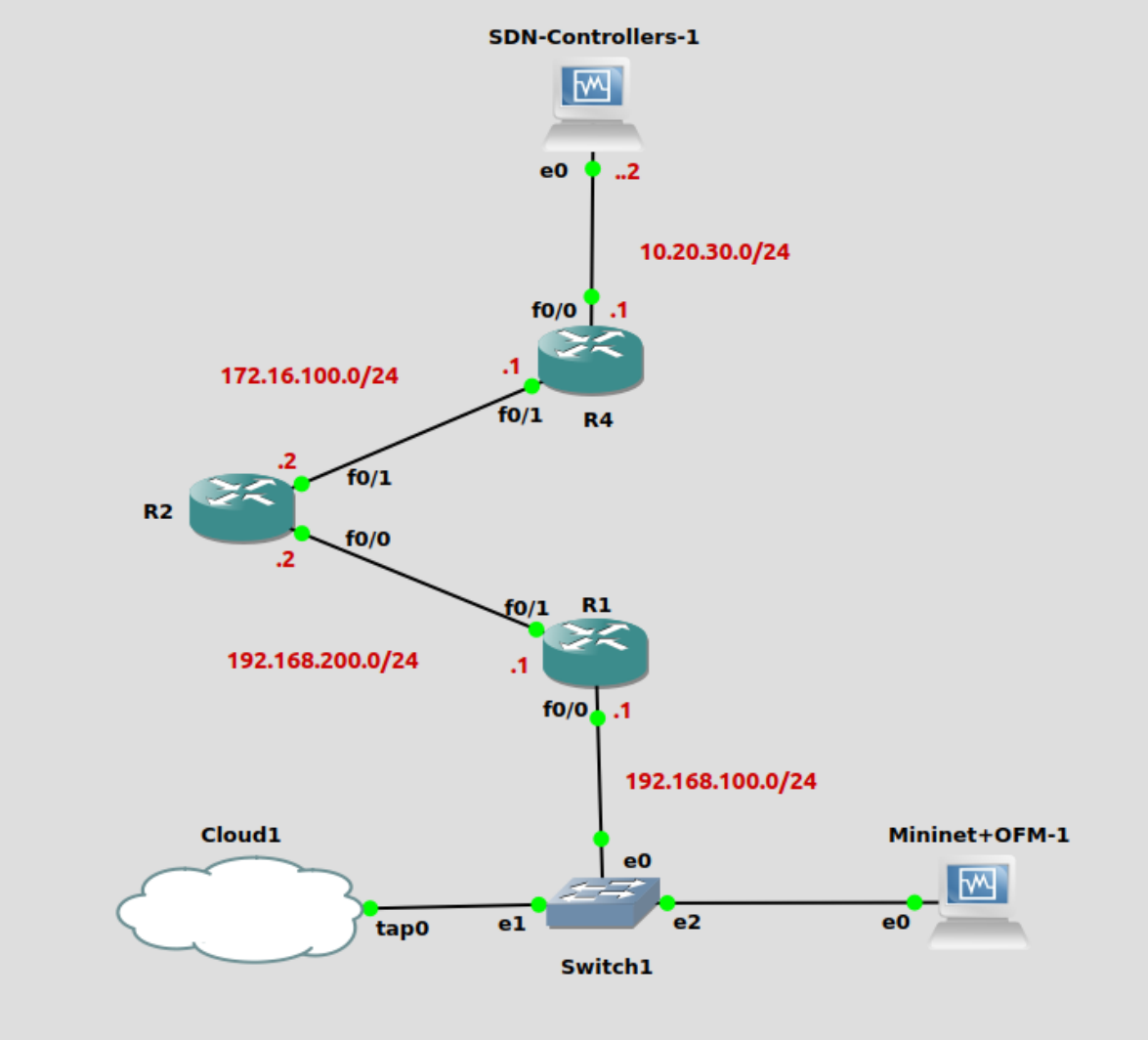
Objective 1 – Set up topology in GNS3



Device	Interface	IP address
R1	fa0/0	192.168.100.1/24
	fa1/0	192.168.200.1/24
R2	fa0/0	192.168.200.2/24

	fa1/0	172.16.100.2/24
R4	fa0/0	10.20.30.1/24
	fa1/0	172.16.100.1/24
Controllers VM		10.20.30.2/24

1. Configure the above topology in GNS3 interconnecting traditional routers, virtual machines, cloud and host laptop.
2. Configure IP addresses on the routers and the Controllers VM as given. Do not configure any IP address on the Mininet VM.
3. Configure DHCP server on R1 to provide an IP address to the Mininet VM.
4. Do not configure any routing commands manually.
5. Paste screenshot of the topology created in GNS3. [**15 points**]



6. Start the Ryu app simple_switch_13.py on the controllers VM.

Objective 2 – Python script

1. Write a script in Python which runs on your laptop to achieve the following objectives (Please read all the objectives before you begin to write your script)-
    a. SSH into R1 and find the IP address leased out to the Mininet VM. Paste relevant screenshots. [**20 points**]

```
kiran@kiran:~/sdn-midterm$ /usr/bin/python3.8 /home/kiran/SDNcodes/code1.py
Starting DHCP server on R1
Waiting for Mininet VM to fetch the IP fro dhcp
Bindings from all pools not associated with VRF:
IP address          Client-ID/              Lease expiration        Type
                    Hardware address/
                    User name
192.168.100.2       0800.2786.a66d          Mar 02 2002 12:02 AM    Automatic
Mininet IP --> 192.168.100.2
```

    b. SSH into the Mininet VM using the IP address found in the previous step and initialize the default Mininet topology (sudo mn). Paste relevant screenshots. [**20 points**]

```
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1)
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Starting CLI:
mininet>
Waiting for the switch to connect to the controller
OVS Bridge <---------- CONNECTED ------> controller
```

c. Configure the OvS on the Mininet VM to connect to the controller. Paste relevant screenshots. [**20 points**]

```
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1)
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Starting CLI:
mininet>
Waiting for the switch to connect to the controller
OVS Bridge <---------- CONNECTED ------> controller
```

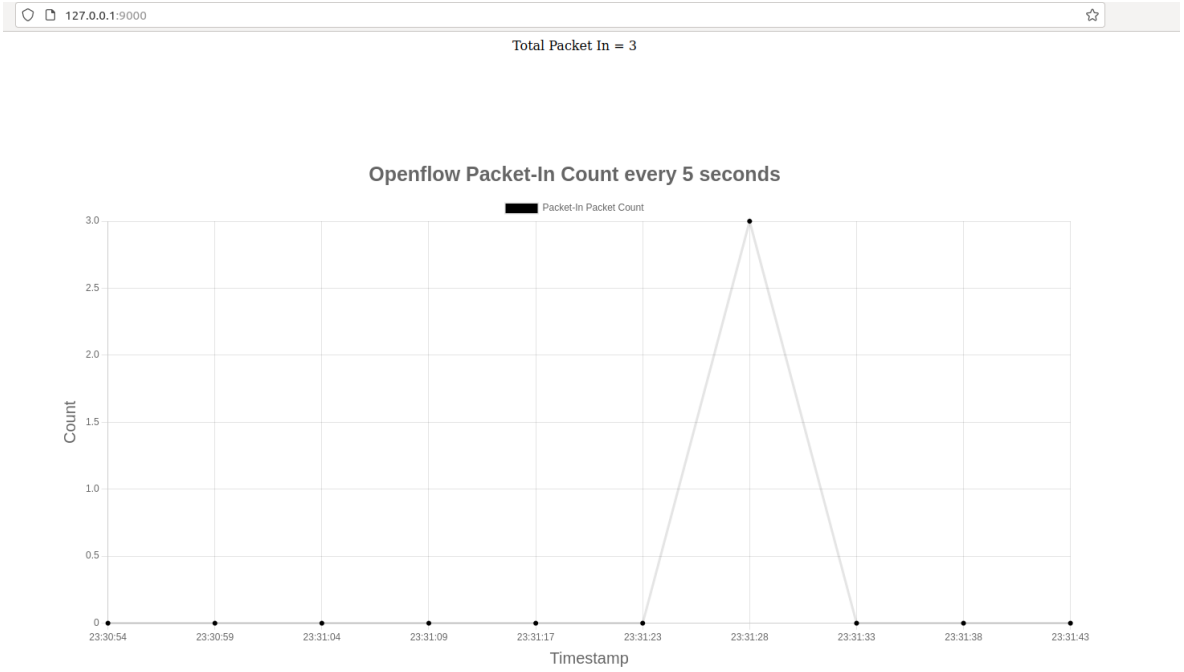
d. SSH into the traditional routers R1, R2 and R3 to configure routing to establish the OpenFlow connectivity between the OvS and the controller. Paste relevant screenshots. [**20 points**]

```
Routes learnt via OSPF
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

     172.16.0.0/24 is subnetted, 1 subnets
O       172.16.100.0 [110/20] via 192.168.200.2, 00:00:26, FastEthernet0/1
C    192.168.200.0/24 is directly connected, FastEthernet0/1
     10.0.0.0/24 is subnetted, 1 subnets
O       10.20.30.0 [110/30] via 192.168.200.2, 00:00:06, FastEthernet0/1
C    192.168.100.0/24 is directly connected, FastEthernet0/0
Waiting for  Ryu to warm up
[1] 746
Url - http://10.20.30.2:8080
```

e. Verify and display that the successful OpenFlow connectivity between the OvS and the controller. Paste relevant screenshots. [**10 points**]

```
OVS Bridge <---------- CONNECTED ------> controller
```

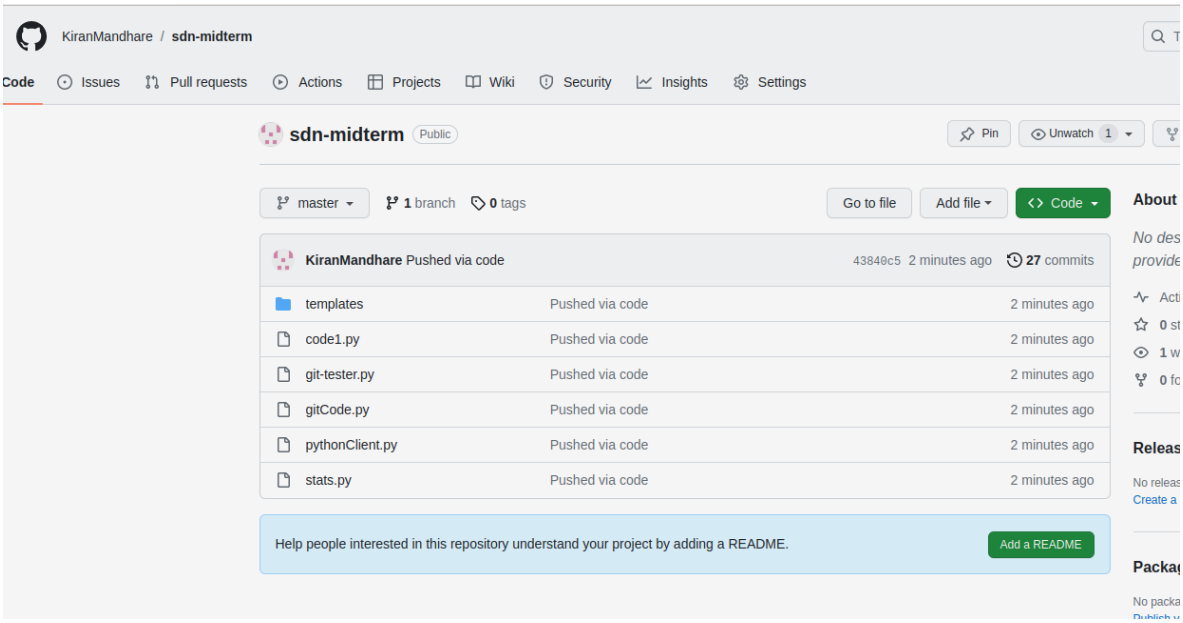
f. Capture the number of OpenFlow Packet_In messages sent from the switch to the controller and visualize this through an interactive graph on a webpage. You can use your favorite Python web-framework like Flask, Django, etc. to set up the webpage. The graphs should be displayed in real-time i.e. they should get refreshed automatically after a periodic interval of time (say 5 seconds) without manually reloading the webpage. Paste relevant screenshots. [**30 points**]




```
10.20.30.2 - - [15/Oct/2023 23:54:58] "POST /updatePacketInCount HTTP/1.1" 200 -
10.20.30.2 - - [15/Oct/2023 23:55:03] "POST /updatePacketInCount HTTP/1.1" 200 -
10.20.30.2 - - [15/Oct/2023 23:55:08] "POST /updatePacketInCount HTTP/1.1" 200 -
10.20.30.2 - - [15/Oct/2023 23:55:13] "POST /updatePacketInCount HTTP/1.1" 200 -
10.20.30.2 - - [15/Oct/2023 23:55:18] "POST /updatePacketInCount HTTP/1.1" 200 -
10.20.30.2 - - [15/Oct/2023 23:55:23] "POST /updatePacketInCount HTTP/1.1" 200 -
10.20.30.2 - - [15/Oct/2023 23:55:28] "POST /updatePacketInCount HTTP/1.1" 200 -
10.20.30.2 - - [15/Oct/2023 23:55:34] "POST /updatePacketInCount HTTP/1.1" 200 -
10.20.30.2 - - [15/Oct/2023 23:55:39] "POST /updatePacketInCount HTTP/1.1" 200 -
10.20.30.2 - - [15/Oct/2023 23:55:44] "POST /updatePacketInCount HTTP/1.1" 200 -
10.20.30.2 - - [15/Oct/2023 23:55:49] "POST /updatePacketInCount HTTP/1.1" 200 -
10.20.30.2 - - [15/Oct/2023 23:55:54] "POST /updatePacketInCount HTTP/1.1" 200 -
10.20.30.2 - - [15/Oct/2023 23:55:59] "POST /updatePacketInCount HTTP/1.1" 200 -
10.20.30.2 - - [15/Oct/2023 23:56:04] "POST /updatePacketInCount HTTP/1.1" 200 -
```

g. At the end, push your Python script to a new private repo ‘SDN-Midterm' in your GitHub account. Paste relevant screenshots. [**5 points**]

```
Untracked files:
  (use "git add <file>..." to include in what will be committed)

        code1.py
        git-tester.py
        gitCode.py
        pythonClient.py
        stats.py
        templates/

nothing added to commit but untracked files present (use "git add" to track)
0
[master 43840c5] Pushed via code
 7 files changed, 389 insertions(+)
 create mode 100644 code1.py
 create mode 100644 git-tester.py
 create mode 100644 gitCode.py
 create mode 100644 pythonClient.py
 create mode 100644 stats.py
 create mode 100644 templates/favicon.ico
 create mode 100644 templates/index.html
Counting objects: 10, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (9/9), done.
Writing objects: 100% (10/10), 4.65 KiB | 1.55 MiB/s, done.
Total 10 (delta 0), reused 0 (delta 0)
To https://github.com/KiranMandhare/sdn-midterm.git
   29577f2..43840c5  master -> master
```



Total Score = _____ / 140