TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA MẠNG MÁY TÍNH VÀ TRUYỀN THÔNG

BÁO CÁO THỰC HÀNH

Bài thực hành số 01: Tìm hiểu về Mininet

Môn học: Công nghệ mạng khả lập trình

Lớp: NT541.P21.1

THÀNH VIÊN THỰC HIỆN:

STT	Họ và tên	MSSV
1	Phạm Thiều Gia Khang	21520967

Điểm tự đánh giá
9/10

ĐÁNH GIÁ KHÁC:

Tổng thời gian thực hiện	6 ngày
Phân chia công việc	
Ý kiến *(nếu có)* + Khó khăn + Đề xuất, kiến nghị	

Phần bên dưới của báo cáo này là báo cáo chi tiết của nhóm thực hiện

MỤC LỤC

A. BÁO CÁO CHI TIẾT

1. Cài đặt Mininet và Ryu Controller:

Khởi tạo môi trường bằng conda:

```
$ conda create -n ryuNet  python=3.9
$ conda activate ryuNet
```

a. Cài đặt mininet:

```
$ git clone https://github.com/mininet/mininet
$ cd mininet
$ ./util/install.sh
```

Kiểm tra cài đặt:

```
(ryuNet) khang@khang-computer:~/weak 1/mininet$ sudo mn --test pingall
[sudo] password for khang:
*** Creating network
*** 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 ...
*** Waiting for switches to connect
s1
*** Ping: testing ping reachability
h1 -> h2
h2 -> h1
*** Results: 0% dropped (2/2 received)
*** Stopping 1 controllers
c0
*** Stopping 2 links
..
*** Stopping 1 switches
s1
*** Stopping 2 hosts
h1 h2
*** Done
completed in 5.534 seconds
```

b. Cài đặt Ryu Controller:

Clone source code của Ryu Controller từ Github:

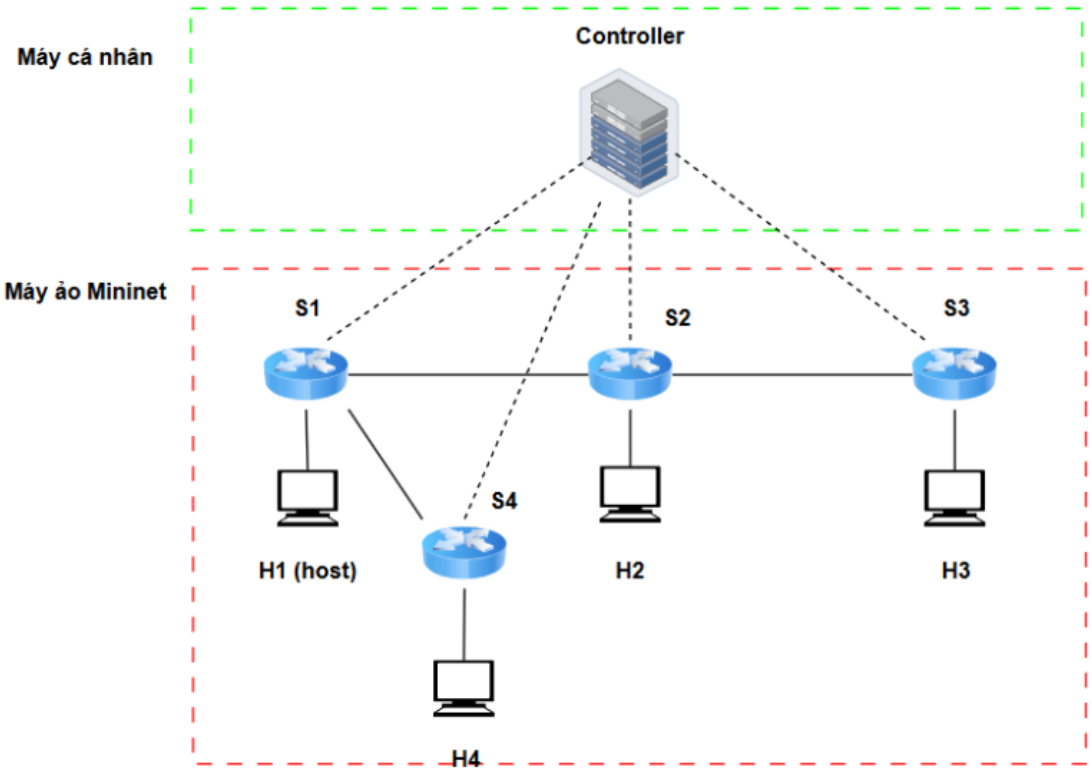
```
(base) khang@khang-computer:~/week_1$ git clone https://github.com/osrg/ryu.git
Cloning into 'ryu'...
remote: Enumerating objects: 26506, done.
remote: Counting objects: 100% (7/7), done.
remote: Compressing objects: 100% (5/5), done.
remote: Total 26506 (delta 1), reused 4 (delta 1), pack-reused 26499 (from 1)
Receiving objects: 100% (26506/26506), 13.95 MiB | 7.18 MiB/s, done.
Resolving deltas: 100% (19160/19160), done.
```

Cài đặt:

```
$ pip3 install setuptools==57.5.0
$ python3 ./setup.py install
```

```
byte-compiling /home/khang/miniconda3/envs/ryuNet/lib/python3.9/site-packages/ryu/contrib/__init__.py to __init__.cpython-39.pyc
running install_data
creating /home/khang/miniconda3/envs/ryuNet/etc/ryu
copying etc/ryu/ryu.conf -> /home/khang/miniconda3/envs/ryuNet/etc/ryu
running install_egg_info
Copying ryu.egg-info to /home/khang/miniconda3/envs/ryuNet/lib/python3.9/site-packages/ryu-4.34-py3.9.egg-info
running install_scripts
/home/khang/miniconda3/envs/ryuNet/lib/python3.9/site-packages/setuptools/command/easy_install.py:2085: EasyInstallDeprecationWarning: Use get_args
  warnings.warn("Use get_args", EasyInstallDeprecationWarning)
/home/khang/miniconda3/envs/ryuNet/lib/python3.9/site-packages/setuptools/command/easy_install.py:2087: EasyInstallDeprecationWarning: Use get_header
  header = cls.get_script_header("", executable, wininst)
Installing ryu script to /home/khang/miniconda3/envs/ryuNet/bin
Installing ryu-manager script to /home/khang/miniconda3/envs/ryuNet/bin
```

2. Tạo mạng SDN theo topology sau:



Cấu hình hệ thống mạng thông qua code Python:

```
from mininet.net import Mininet
from mininet.node import RemoteController
from mininet.cli import CLI
from mininet.log import setLogLevel
from mininet.topo import Topo


class Lab1_Topo( Topo ):
   def build( self ):
       # Add hosts
       h1 = self.addHost( 'h1' )
       h2 = self.addHost( 'h2' )
       h3 = self.addHost( 'h3' )
       h4 = self.addHost( 'h4' )
       #Add switch
       s1 = self.addSwitch( 's1' )
       s2 = self.addSwitch( 's2' )
       s3 = self.addSwitch( 's3' )
       s4 = self.addSwitch( 's4' )
       # Add links switch with switch
       self.addLink( s1, s2 )
       self.addLink( s2, s3 )
       self.addLink( s1, s4 )
       #Add link switch with host
       self.addLink( s1, h1 )
       self.addLink( s2, h2 )
       self.addLink( s3, h3 )
       self.addLink( s4, h4 )
def run():
   topo = Lab1_Topo()
   net = Mininet(topo=topo, controller=None)
```

```
        c0      =      net.addController('c0',      controller=RemoteController,
ip="127.0.0.1", port=6633)

        net.start()

        CLI(net)

        net.stop()

    if __name__ == '__main__':

        setLogLevel('info')

        run()
```

Trong đó:

- Sử dụng hàm addHost() để thêm 1 host vào mạng
- Sử dụng hàm addSwitch() để thêm 1 switch vào mạng
- Sử dụng hàm addLink() để liên kết các thành phần với nhau, ở đây là host và switch.
- Sử dụng hàm addController() để thêm controller vào mạng với ip là 127.0.0.1 (loalhost) và port 6633.

Tạo mạng SDN theo topology trên:

Khởi động ryu controller, thực hiện lắng nghe từ mininet:

```
$ ryu-manager ryu.app.simple_switch
```

```
(ryuNet) (base) khang@khang-computer:~/week_1/ryu$ ryu-manager ryu.app.simple_switch
loading app ryu.app.simple_switch
loading app ryu.controller.ofp_handler
instantiating app ryu.app.simple_switch of SimpleSwitch
instantiating app ryu.controller.ofp_handler of OFPHandler
```

Khởi động mininet theo file custom_1.py đã tạo từ trước:

```
(ryuNet) (base) khang@khang-computer:~/week_1$ sudo /home/khang/miniconda3/envs/ryuNet/bin/python /home/khang/week_1/mininet/custom/custom_1.py
*** Creating network
*** Adding hosts:
h1 h2 h3 h4
*** Adding switches:
s1 s2 s3 s4
*** Adding links:
(s1, h1) (s1, s2) (s1, s4) (s2, h2) (s2, s3) (s3, h3) (s4, h4)
*** Configuring hosts
h1 h2 h3 h4
*** Starting controller
c0
*** Starting 4 switches
s1 s2 s3 s4 ...
*** Starting CLI:
mininet>
```

Khi ryu controller kết nối thành công với mininet sẽ hiển thị thông tin về các packet trong mạng:

```
(ryuNet) (base) khang@khang-computer:~/week_1/ryu$ ryu-manager ryu.app.simple_switch
loading app ryu.app.simple_switch
loading app ryu.controller.ofp_handler
instantiating app ryu.app.simple_switch of SimpleSwitch
instantiating app ryu.controller.ofp_handler of OFPHandler
packet in 4 ea:2a:fa:5a:eb:8c 33:33:00:00:00:02 2
packet in 1 ea:2a:fa:5a:eb:8c 33:33:00:00:00:02 2
packet in 2 ea:2a:fa:5a:eb:8c 33:33:00:00:00:02 1
packet in 3 ea:2a:fa:5a:eb:8c 33:33:00:00:00:02 1
packet in 3 f2:e5:5a:fd:46:38 33:33:00:00:00:02 1
packet in 2 16:c5:06:b1:26:3b 33:33:00:00:00:02 2
packet in 1 16:c5:06:b1:26:3b 33:33:00:00:00:02 1
packet in 4 16:c5:06:b1:26:3b 33:33:00:00:00:02 1
packet in 4 e2:d3:cb:f9:a9:0d 33:33:00:00:00:fb 1
packet in 1 06:ea:dc:0d:a3:6d 33:33:00:00:00:fb 2
packet in 2 06:ea:dc:0d:a3:6d 33:33:00:00:00:fb 1
packet in 3 06:ea:dc:0d:a3:6d 33:33:00:00:00:fb 1
packet in 3 f2:e5:5a:fd:46:38 33:33:00:00:00:fb 1
packet in 2 26:83:2e:1e:70:38 33:33:00:00:00:fb 1
packet in 3 26:83:2e:1e:70:38 33:33:00:00:00:fb 1
packet in 1 16:85:86:43:e5:72 33:33:00:00:00:fb 1
packet in 4 16:85:86:43:e5:72 33:33:00:00:00:fb 1
packet in 2 16:c5:06:b1:26:3b 33:33:00:00:00:fb 2
packet in 1 16:c5:06:b1:26:3b 33:33:00:00:00:fb 1
packet in 4 16:c5:06:b1:26:3b 33:33:00:00:00:fb 1
packet in 1 06:ea:dc:0d:a3:6d 33:33:00:00:00:02 2
packet in 1 16:85:86:43:e5:72 33:33:00:00:00:02 1
```

3. Kiểm tra mạng SDN vừa tạo:

Sử dụng lệnh net để hiển thị các thành phần trong mạng:

```
*** Starting CLI:
mininet> net
h1 h1-eth0:s1-eth3
h2 h2-eth0:s2-eth3
h3 h3-eth0:s3-eth2
h4 h4-eth0:s4-eth2
s1 lo:  s1-eth1:s2-eth1 s1-eth2:s4-eth1 s1-eth3:h1-eth0
s2 lo:  s2-eth1:s1-eth1 s2-eth2:s3-eth1 s2-eth3:h2-eth0
s3 lo:  s3-eth1:s2-eth2 s3-eth2:h3-eth0
s4 lo:  s4-eth1:s1-eth2 s4-eth2:h4-eth0
c0
mininet>
```

a. ping h1 đến h3:

```
mininet> h1 ping -c 10 h3
PING 10.0.0.3 (10.0.0.3) 56(84) bytes of data.
64 bytes from 10.0.0.3: icmp_seq=1 ttl=64 time=11.1 ms
64 bytes from 10.0.0.3: icmp_seq=2 ttl=64 time=0.769 ms
64 bytes from 10.0.0.3: icmp_seq=3 ttl=64 time=0.067 ms
64 bytes from 10.0.0.3: icmp_seq=4 ttl=64 time=0.100 ms
64 bytes from 10.0.0.3: icmp_seq=5 ttl=64 time=0.066 ms
64 bytes from 10.0.0.3: icmp_seq=6 ttl=64 time=0.083 ms
64 bytes from 10.0.0.3: icmp_seq=7 ttl=64 time=0.065 ms
64 bytes from 10.0.0.3: icmp_seq=8 ttl=64 time=0.066 ms
64 bytes from 10.0.0.3: icmp_seq=9 ttl=64 time=0.062 ms
64 bytes from 10.0.0.3: icmp_seq=10 ttl=64 time=0.107 ms

--- 10.0.0.3 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9172ms
rtt min/avg/max/mdev = 0.062/1.252/11.137/3.301 ms
```

b. ping từ h1 đến h4:

```
mininet> h1 ping -c 10 h4
PING 10.0.0.4 (10.0.0.4) 56(84) bytes of data.
64 bytes from 10.0.0.4: icmp_seq=1 ttl=64 time=6.73 ms
64 bytes from 10.0.0.4: icmp_seq=2 ttl=64 time=0.271 ms
64 bytes from 10.0.0.4: icmp_seq=3 ttl=64 time=0.065 ms
64 bytes from 10.0.0.4: icmp_seq=4 ttl=64 time=0.067 ms
64 bytes from 10.0.0.4: icmp_seq=5 ttl=64 time=0.077 ms
64 bytes from 10.0.0.4: icmp_seq=6 ttl=64 time=0.074 ms
64 bytes from 10.0.0.4: icmp_seq=7 ttl=64 time=0.062 ms
64 bytes from 10.0.0.4: icmp_seq=8 ttl=64 time=0.095 ms
64 bytes from 10.0.0.4: icmp_seq=9 ttl=64 time=0.080 ms
64 bytes from 10.0.0.4: icmp_seq=10 ttl=64 time=0.071 ms

--- 10.0.0.4 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9180ms
rtt min/avg/max/mdev = 0.062/0.759/6.730/1.991 ms
mininet>
```

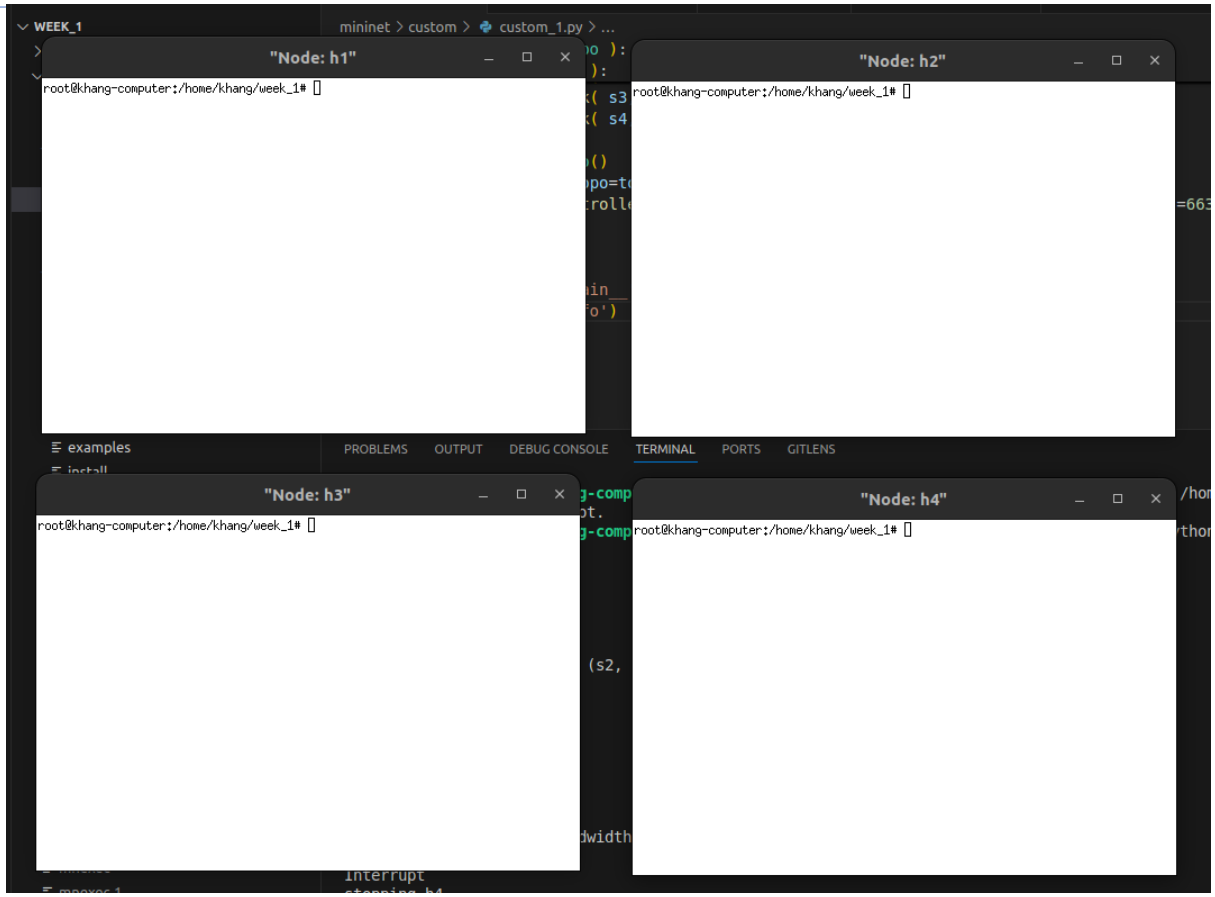
c. ping tới tất cả các host:

```
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 h3 h4
h2 -> h1 h3 h4
h3 -> h1 h2 h4
h4 -> h1 h2 h3
*** Results: 0% dropped (12/12 received)
mininet>
```

d. Dùng iperf để test through output của mạng:

Mở terminal của h1, h2, h3, h4:

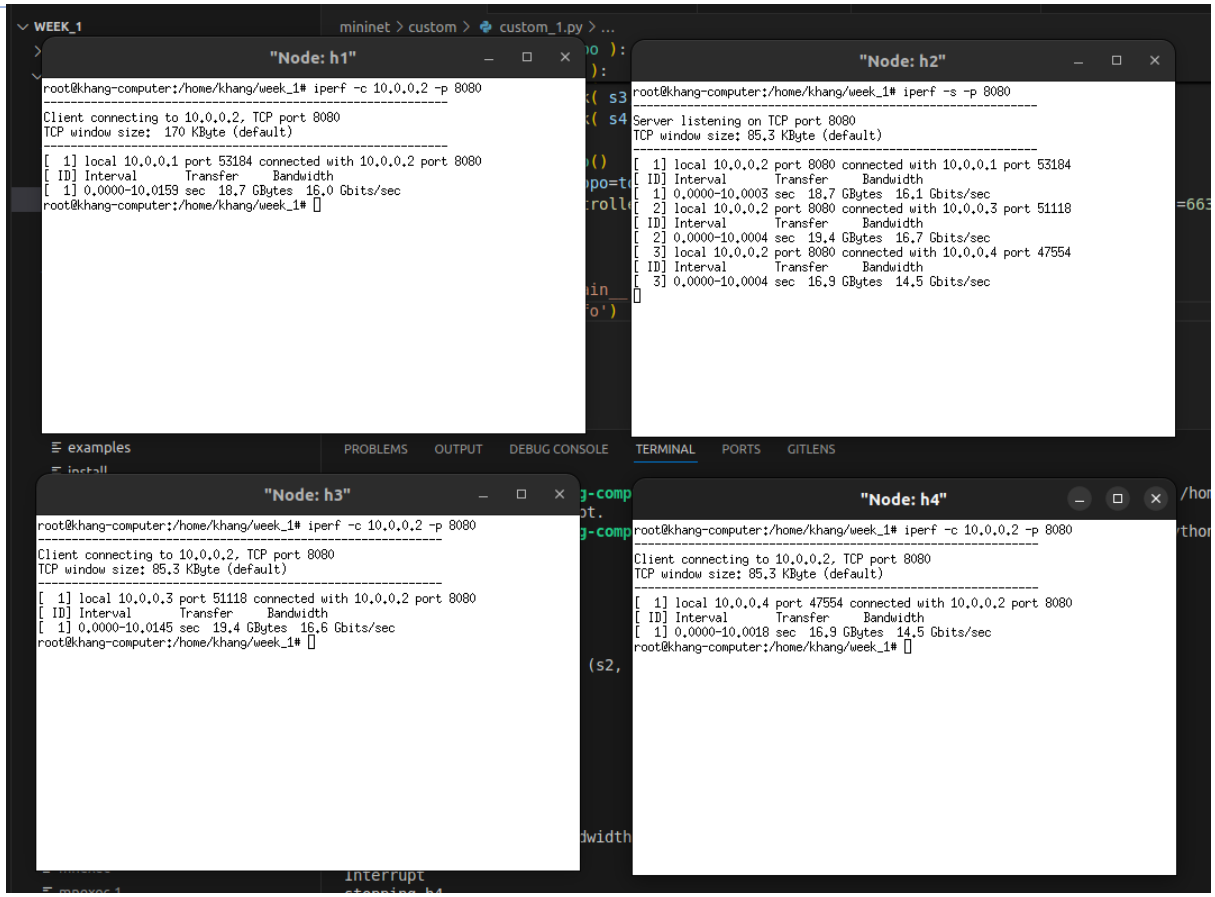
```
mininet> xterm h1 h2 h3 h4
```



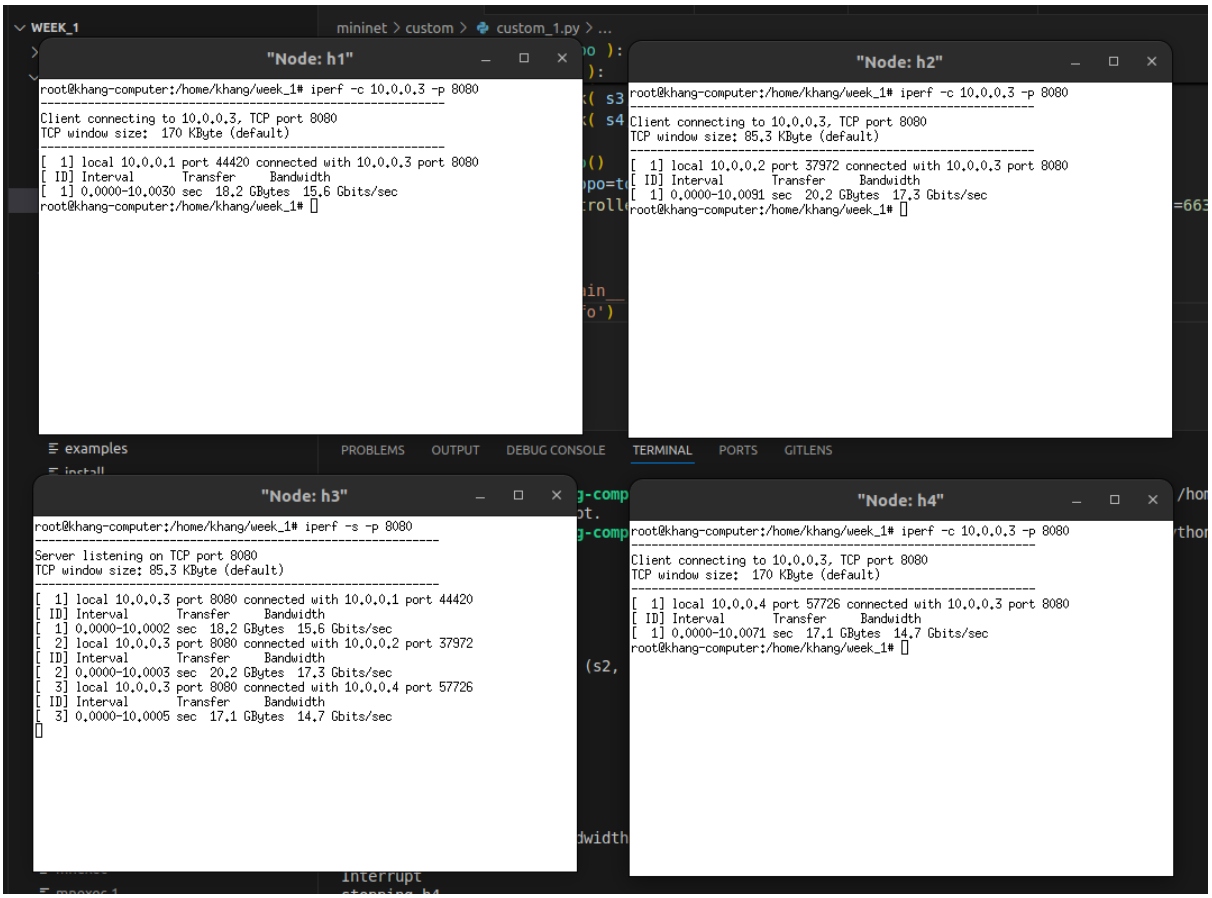
Ở đây chúng ta sẽ chạy server TCP trên từng host để kiểm tra tại port 8080:

Host 1 (10.0.0.1):

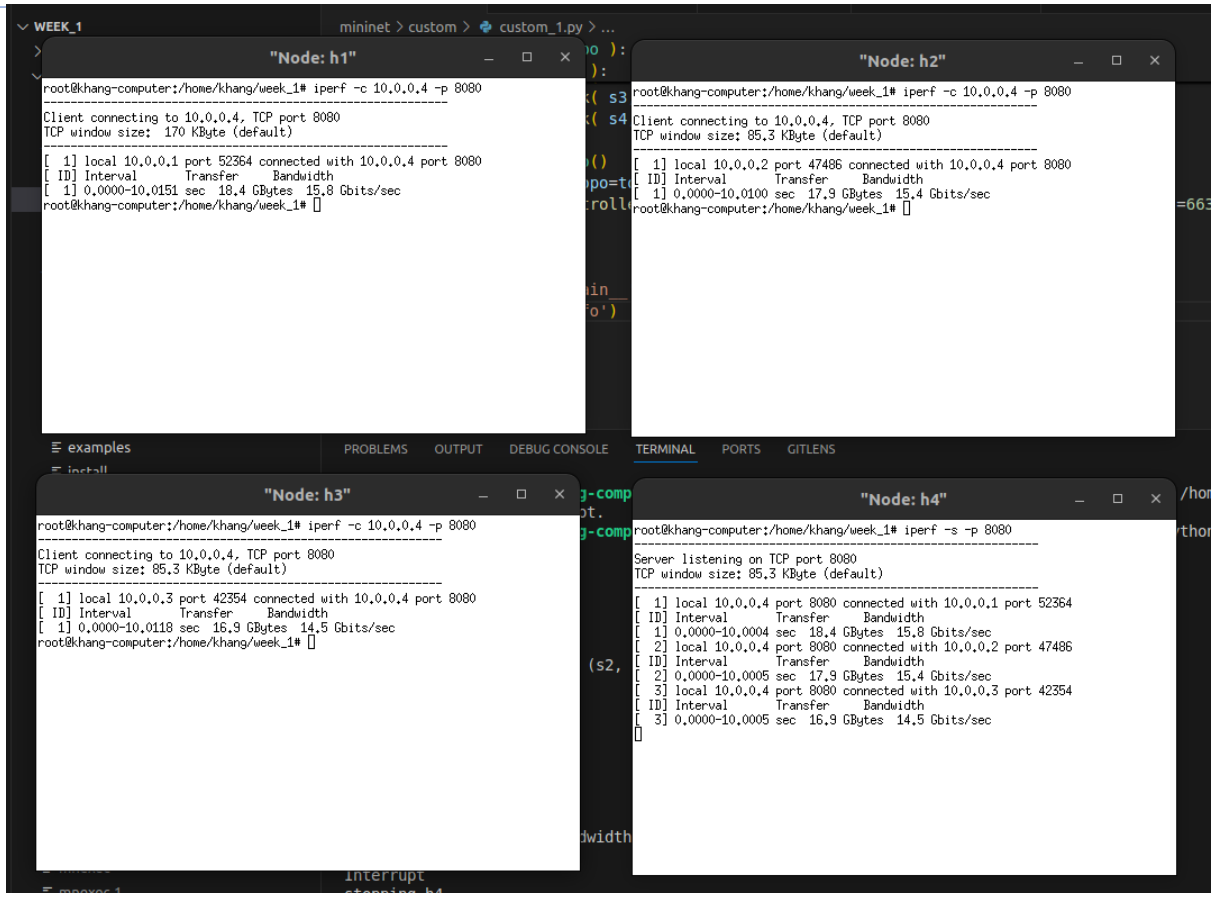
Host 2(10.0.0.2):



Host 3 (10.0.0.3):



Host 4 (10.0.0.4):



B. TÀI LIỆU THAM KHẢO

[1] Ryu SDN Framework, "Ryu SDN Framework," [Online]. Available: https://ryu-sdn.org/. [Accessed: 18-Mar-2025].

[2] Mininet, "Mininet Walkthrough," [Online]. Available: https://mininet.org/walkthrough/. [Accessed: 18-Mar-2025].