Report to mininet lab2

result

```
wayne@wayne-VirtualBox:~/hw2 mininet$ sudo python3 custom_topology.py
Starting network
*** Configuring hosts
h1 h2 h3 h4 h5 h6 h7 h8 h9
Starting iperf session from h1 to h2 with bandwidth limit 5 Mbps
------------------------------------------------------------
Client connecting to 10.0.0.2, TCP port 5001
TCP window size: 85.3 KByte (default)
------------------------------------------------------------
[  1] local 10.0.0.1 port 59234 connected with 10.0.0.2 port 5001
[ ID] Interval       Transfer     Bandwidth
[  1] 0.0000-10.0117 sec  6.38 MBytes  5.34 Mbits/sec

iperf session completed
Starting iperf session from h1 to h3 with bandwidth limit 10 Mbps
------------------------------------------------------------
Client connecting to 10.0.0.3, TCP port 5001
TCP window size: 85.3 KByte (default)
------------------------------------------------------------
[  1] local 10.0.0.1 port 54680 connected with 10.0.0.3 port 5001
[ ID] Interval       Transfer     Bandwidth
[  1] 0.0000-10.0087 sec  12.6 MBytes  10.6 Mbits/sec

iperf session completed
Starting iperf session from h4 to h5 with bandwidth limit 15 Mbps
------------------------------------------------------------
Client connecting to 10.0.0.5, TCP port 5001
TCP window size: 85.3 KByte (default)
------------------------------------------------------------
[  1] local 10.0.0.4 port 38152 connected with 10.0.0.5 port 5001
[ ID] Interval       Transfer     Bandwidth
[  1] 0.0000-10.0092 sec  18.9 MBytes  15.8 Mbits/sec

iperf session completed
```

```
------------------------------------------------------------
Client connecting to 10.0.0.5, TCP port 5001
TCP window size: 85.3 KByte (default)
------------------------------------------------------------
[  1] local 10.0.0.4 port 38152 connected with 10.0.0.5 port 5001
[ ID] Interval       Transfer     Bandwidth
[  1] 0.0000-10.0092 sec  18.9 MBytes  15.8 Mbits/sec

iperf session completed
Starting iperf session from h6 to h8 with bandwidth limit 20 Mbps
------------------------------------------------------------
Client connecting to 10.0.0.8, TCP port 5001
TCP window size: 85.3 KByte (default)
------------------------------------------------------------
[  1] local 10.0.0.6 port 36624 connected with 10.0.0.8 port 5001
[ ID] Interval       Transfer     Bandwidth
[  1] 0.0000-10.0169 sec  25.1 MBytes  21.0 Mbits/sec

iperf session completed
Running CLI
*** Starting CLI:
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 h3 h4 h5 h6 h7 h8 h9
h2 -> h1 h3 h4 h5 h6 h7 h8 h9
h3 -> h1 h2 h4 h5 h6 h7 h8 h9
h4 -> h1 h2 h3 h5 h6 h7 h8 h9
h5 -> h1 h2 h3 h4 h6 h7 h8 h9
h6 -> h1 h2 h3 h4 h5 h7 h8 h9
h7 -> h1 h2 h3 h4 h5 h6 h8 h9
h8 -> h1 h2 h3 h4 h5 h6 h7 h9
h9 -> h1 h2 h3 h4 h5 h6 h7 h8
*** Results: 0% dropped (72/72 received)
```

```
# 從Mininet中導入必要的模塊
from mininet.net import Mininet
from mininet.node import OVSSwitch, Controller, RemoteController
from mininet.cli import CLI
from mininet.log import setLogLevel
from mininet.link import TCLink
```

- 引入mininet所需要的module

```python
# 函數運行帶有帶寬限制的兩個主機之間的iperf會話
def runIperf(net, src, dst, bw_limit, duration=10):
    src_node = net.get(src)
    dst_node = net.get(dst)

    # 顯示iperf會話的信息
    print("從 {} 到 {} 開始iperf會話，帶寬限制為 {} Mbps".format(src, dst, b

    # 在目標主機上啟動iperf服務器
    iperf_server_cmd = 'iperf -s -t {} &'.format(duration)
    server_process = dst_node.popen(iperf_server_cmd)

    # 在源主機上構建並執行iperf客戶端命令
    iperf_client_cmd = 'iperf -c {} -t {} -b {}M'.format(dst_node.IP(),
    client_process = src_node.popen(iperf_client_cmd)
```

獲取源主機和目標主機的節點,
啟用iperf服務在目標主機,而源主機啟動客戶端

```python
# 等待iperf客戶端進程完成
client_exit_code = client_process.wait()

# 打印iperf客戶端輸出
print(client_process.stdout.read().decode('utf-8'))

# 檢查帶寬是否超過設定值
if client_exit_code == 0:
    client_output = client_process.communicate()[0].decode('utf-8')
    if "Mbits/sec" in client_output:
        measured_bandwidth = float(client_output.split()[-2])
        if measured_bandwidth > bw_limit:
            print("警告：帶寬超過設定限制。終止iperf進程。")

print("iperf會話完成")
```

印出所需顯示東西並檢查帶寬是否在設定值內

- 這個函數定義了Mininet拓撲
- 它創建交換機,主機和鏈接

```python
def createTopo():
    # 創建Mininet網絡
    net = Mininet(controller=None, switch=OVSSwitch, link=TCLink)

    # 添加Ryu控制器
    c1 = net.addController('c1', controller=RemoteController, ip='127.0

    # 添加交換機
    s1 = net.addSwitch('s1')
    s2 = net.addSwitch('s2')
    s3 = net.addSwitch('s3')
    s4 = net.addSwitch('s4')
    s5 = net.addSwitch('s5')
    s6 = net.addSwitch('s6')

    # 添加主機
    h1 = net.addHost('h1')
    h2 = net.addHost('h2')
    h3 = net.addHost('h3')
    h4 = net.addHost('h4')
    h5 = net.addHost('h5')
    h6 = net.addHost('h6')
    h7 = net.addHost('h7')
    h8 = net.addHost('h8')
    h9 = net.addHost('h9')
```

啟動網絡
設計網路接口ip,port
建立主機之間會話(eg.限制流量)
進入CLI
檢查腳本是否作為主程序運行

```
    print("啟動網絡")
    net.build()
    c1.start()
    s1.start([c1])
    s2.start([c1])
    s3.start([c1])
    s4.start([c1])
    s5.start([c1])
    s6.start([c1])

    # 为主机设置IP地址
    for i in range(1, 10):
        net.get('h{}'.format(i)).cmd('ifconfig h{}-eth0 10.0.0.{} netma

    # 设置指定主机之间的iperf会话
    runIperf(net, 'h1', 'h2', 5)
    runIperf(net, 'h1', 'h3', 10)
    runIperf(net, 'h4', 'h5', 15)
    runIperf(net, 'h6', 'h8', 20)

    # 啟動Mininet命令行界面（CLI）
    print("運行CLI")
    CLI(net)

    # 在CLI關閉時停止網絡
    print("停止網絡")
    net.stop()

# 如果是主程序，執行腳本
if __name__ == '__main__':
    # 將Mininet日誌級別設置為“info”
    setLogLevel('info')

    # 調用函數創建網絡拓撲
    createTopo()
```