

Report 109550027 紀竺均

- Describe each step and how to run your program

### Task1: Environment Setup

First, I install Wireshark and Oracle VM VirtualBox on my computer. Then, I import the .ova file into my oracle vm virtualbox. Finally, I use 'sudo mn' to run mininet for testing.

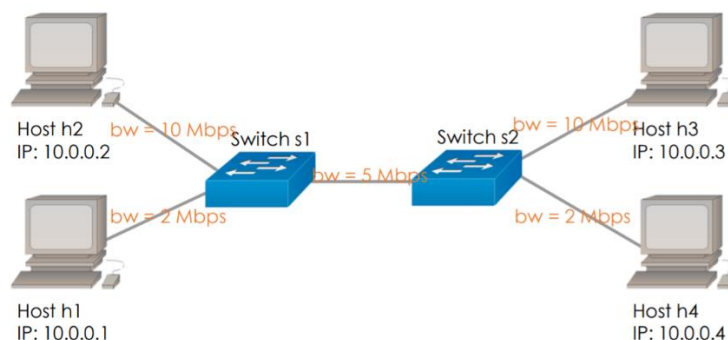


### Task2: Create a Topology

I refer the file 'topo.py' and create a new network topology.

```
class MininetTopo(Topo):
    def build(self):
        # Add hosts to a topology
        self.addHost("h1")
        self.addHost("h2")
        self.addHost("h3")
        self.addHost("h4")
        # Add switches to a topology
        self.addSwitch("s1")
        self.addSwitch("s2")
        # Add bidirectional links to a topology, and set bandwidth(Mbps)
        self.addLink("h1", "s1", bw=2)
        self.addLink("h2", "s1", bw=10)
        self.addLink("s1", "s2", bw=5)
        self.addLink("h3", "s2", bw=10)
        self.addLink("h4", "s2", bw=2)
```

This construct the topology as following:



### Task3: Generate Flows via iPerf

Similarly, I refer 'topo.py' and generate four flows in total. ({h1->h4, h2->h3} \*

{TCP, UDP}) The detailed iPerf command will be listed below.

#### Task4: Compute Throughput

I write a file 'computeRate.py' to parse the pcap file and compute the throughput of each flow.

Step one: get path.

Instead of using sys.argv[1] as the parser.py file did, I simply wrote down all the path of pcap file such that I don't have to call the four paths every time I want to run 'computeRate.py'.

```
# get path of pcap file
I1 = "../out/TCP_h4.pcap"
I2 = "../out/TCP_h3.pcap"
U1 = "../out/UDP_h4.pcap"
U2 = "../out/UDP_h3.pcap"
```

Step two: read pcap.

Using function rdpcap(FILE\_PATH)

Step three: compute the rate

First, I catch every packet in [TCP] layer, and calculate the total length using len(packet). Then, since the time the flow finish running is almost equal to five second. I use length(byte) / 5(sec) \* (8bits/byte) / 1000000(1/M) to compute Flow rate.

#### Task5: Check Your Answer

I open the .pcap file on my computer and use Wireshark to filter the flows. I use the src and dst IP address to filter, listed below in the question section.

How to run the program?

```
student@student-VirtualBox:~/Desktop/lab1-chuchunchi/src$ sudo python3 computeRate.py
```

- Show the results of computeRate.py and statistics of Wireshark / Describe your observations from the results in this lab

#### ■ Results of computeRate.py

```
student@student-VirtualBox:~/Desktop/lab1-chuchunchi/src$ sudo python3 computeRate.py
[sudo] password for student:
--- TCP ---
Flow1(h1->h4):2.0115103999999997Mbps
Flow3(h2->h3):3.0411552Mbps
--- UDP ---
Flow1(h1->h4):2.050464Mbps
Flow3(h2->h3):3.3181247999999997Mbps
```

#### ■ statistics of Wireshark

##### ■ TCP Flow1(h1->h4)

Statistics			
Measurement	Captured	Displayed	Marked
Packets	902	446 (49.4%)	—
Time span, s	15.745	5.069	—
Average pps	57.3	88.0	—
Average packet size, B	1401	2761	—
Bytes	1264054	1231284 (97.4%)	0
Average bytes/s	80k	242k	—
Average bits/s	642k	1943k	—

##### ■ TCP Flow3(h2->h3)

Statistics			
Measurement	Captured	Displayed	Marked
Packets	1327	647 (48.8%)	—
Time span, s	15.895	5.070	—
Average pps	83.5	127.6	—
Average packet size, B	1437	2875	—
Bytes	1907540	1859950 (97.5%)	0
Average bytes/s	120k	366k	—
Average bits/s	960k	2934k	—

##### ■ UDP Flow1(h1->h4)

Statistics			
Measurement	Captured	Displayed	Marked
Packets	931	847 (91.0%)	—
Time span, s	32.065	5.382	—
Average pps	29.0	157.4	—
Average packet size, B	1388	1499	—
Bytes	1292436	1269600 (98.2%)	0
Average bytes/s	40k	235k	—
Average bits/s	322k	1887k	—

##### ■ UDP Flow3(h2->h3)

Statistics			
Measurement	Captured	Displayed	Marked
Packets	1453	1371 (94.4%)	—
Time span, s	32.409	5.261	—
Average pps	44.8	260.6	—
Average packet size, B	1435	1504	—
Bytes	2084636	2061888 (98.9%)	0
Average bytes/s	64k	391k	—
Average bits/s	514k	3135k	—

From the comparison of the two results, the average rates are approximately the same. Also, the rate of TCP flows and UDP flows are roughly equal, too.

- Answer the following question in short:
- What does each iPerf command you used mean?

Take one UDP flow for example:

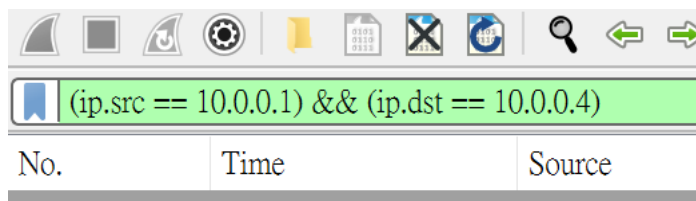
```
# UDP flow
h3.cmd("iperf -s -i 1 -t 5 -p 7777 -b 10M -u > ../out/UDP_s_h3.txt &")
h2.cmd("iperf -c " + "10.0.0.3" + " -i 1 -t 5 -p 7777 -b 10M -u > ../out/UDP_c_h2.txt &")
```

- -s: server mode
- -i 1: set the interval time to one second.
- -t 5: set transmit time for five second.
- -p 7777: the server port to listen on = port 7777.
- -b 10M: Set target bandwidth to 10M bits/sec. (Required to be larger than 5M bits/sec.)
- -u: use UDP
- -c: client mode, connecting to a server.

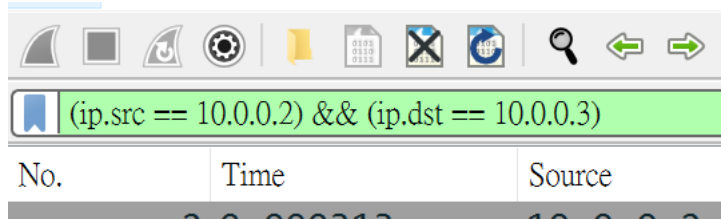
- What is your command to filter each flow in Wireshark?

I use the src and dst IP address to filter.

For Flow1(h1->h4), either in TCP or UDP, I use



For Flow3(h2->h3), either in TCP or UDP, I use



- Does the throughput match the bottleneck throughput of the path?

Yes,

from the result above, either in TCP or UDP, the bandwidth of flow1 = 2 and the bandwidth of flow3 = 3.

$2 + 3 = 5$  = the bottleneck throughput of the path.

- Do you observe the same throughput from TCP and UDP? Can both flows equally share the bandwidth?

Yes, the throughput from TCP and UDP are roughly the same, but the flows DIDN'T equally share the bandwidth. (Flow1 = 2 , Flow3 = 3)

- Bonus

- What you have learned from this lab? • What difficulty you have met in this lab?

這次 LAB 是我第一次操作虛擬機，我覺得蠻酷的，在過程中遇到一個問題，就是我原本想要使用 `udpdump` 去把封包記錄在 `pcap` 檔上，可是他一直 `dump` 不了，後來我去請教我的大神同學，在他幫我 `debug` 的過程中，我真的覺得她好厲害，從一個終端機中又開出另一個終端機(好像叫 `Xterm`)，還輸入一堆我看不懂的指令，跟駭客一樣，從此以後成為我學習的標竿。

對了，關於那個問題，我後來直接使用 `tcpdump`，關於兩者的差異我會再看看 `document`。

還有，對於計算時間的部分，我本來想要去計算系統 `run` 多久的時間，後來經過諮商才發現可以直接用 5 秒就好...

總而言之，這次的 LAB 對我而言是一個很新的嘗試，原本不太熟悉的網路、`bandwidth` 計算，經過實作以後好像變得不再那麼抽象了。