電腦網路實驗實驗報告 < 軟體定義網路 >

姓名: 翁佳煌　　學號: 409430030

1. 實驗名稱

Mininet & SDN Controller

2. 實驗目的

課堂實驗:

課堂上實現 IPv4 Multicasting，讓封包可以送到相同的 Multicast group 使用者。其中，Multicast groups 被定義為 10.0.0.0/24 和 10.0.1.0/24。通過在 A1 和 B1 終端機上執行 multicast_script.py，可以將封包發送到 Multicast group 中，並使用記錄檔記錄發送和接收的封包。最終，通過將記錄檔的內容擷圖表示，評估實驗結果的有效性。

作業:

作業為使用 OpenVSwitch 指令，對三條連線加入路由規則，同時使用 Meter 限定流量。在路由規則的設定上，我們要求使用 IPv4 位址和 UDP Port ID 進行路由。

3. 實驗設備

Linux 作業系統之電腦。
Mininet。
Testbed from: http://github.com/Hsun111/MininetTopology

4. 實驗步驟

課堂實驗:

步驟 1.

首先些必須安裝相關的套件，如以下指令。

Install from Mininet:

sudo apt install wireshark xterm ifconfig mininet

Install from APT:

sudo apt-get install openvswitch-switch

Download the testbed:

git clone http://github.com/Hsun111/MininetTopology

步驟 2.

開啟 Terminal，並 cd 到 MininetTopology 的資料夾中，然後輸入 sudo python3 ./LAB2.py，如下圖。

步驟 3.

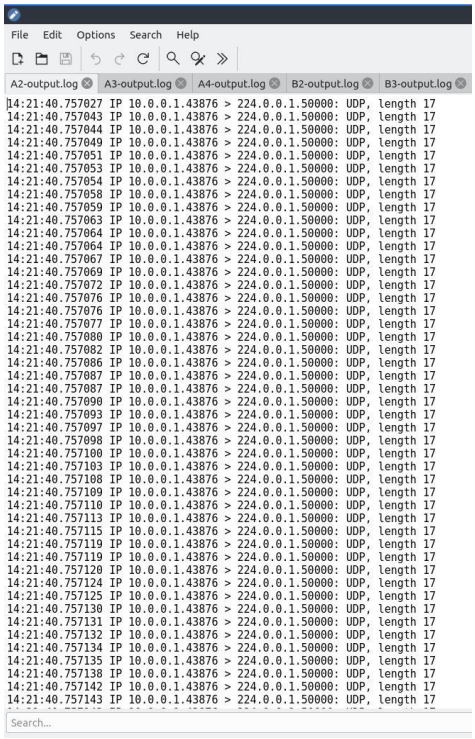
輸入以下指令:

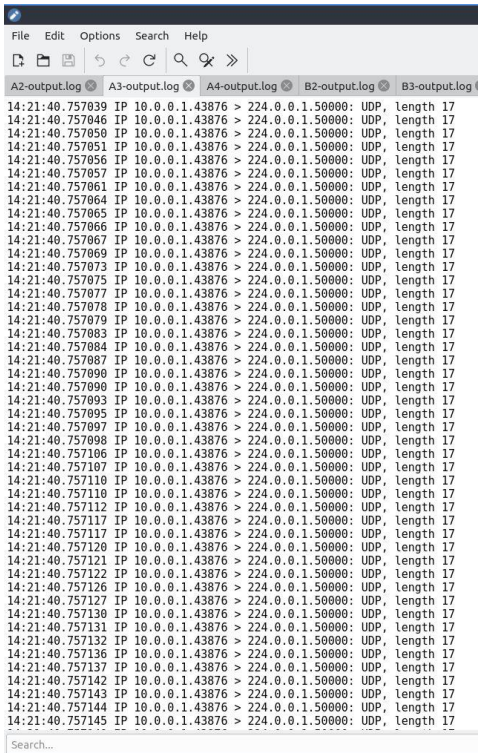
```
mininet> sh ovs-ofctl add-group s1 group_id=1,type=all,bucket=output:1,bucket=output:2,bucket=output:4
mininet> sh ovs-ofctl add-group s2 group_id=1,type=all,bucket=output:3
mininet> sh ovs-ofctl add-group s3 group_id=1,type=all,bucket=output:3
mininet> sh ovs-ofctl add-group s1 group_id=2,type=all,bucket=output:1,bucket=output:2
mininet> sh ovs-ofctl add-group s2 group_id=2,type=all,bucket=output:4
mininet> sh ovs-ofctl add-group s3 group_id=2,type=all,bucket=output:4,bucket=output:5
mininet> sh ovs-ofctl add-flow s1 dl_type=0x0800,nw_dst=224.0.0.1,actions=group:1
mininet> sh ovs-ofctl add-flow s1 dl_type=0x0800,nw_dst=224.0.0.2,actions=group:2
mininet> sh ovs-ofctl add-flow s2 dl_type=0x0800,nw_dst=224.0.0.1,actions=group:1
mininet> sh ovs-ofctl add-flow s2 dl_type=0x0800,nw_dst=224.0.0.2,actions=group:2
mininet> sh ovs-ofctl add-flow s3 dl_type=0x0800,nw_dst=224.0.0.1,actions=group:1
mininet> sh ovs-ofctl add-flow s3 dl_type=0x0800,nw_dst=224.0.0.2,actions=group:2
mininet> xterm A1 B1
```

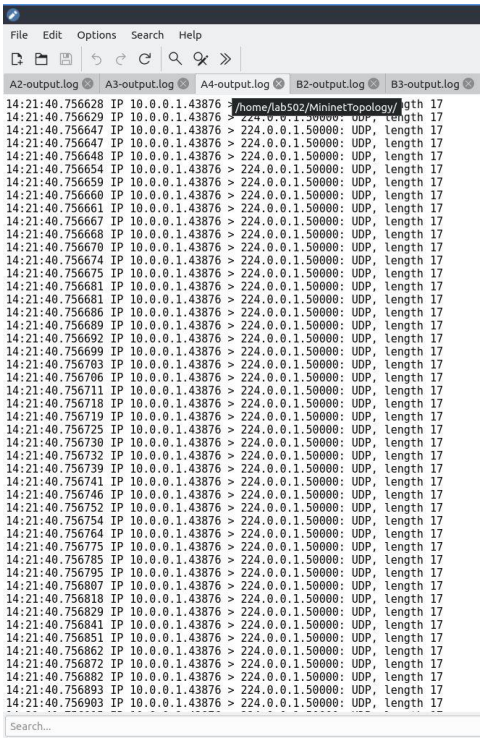
步驟 4.

觀察 log 檔:

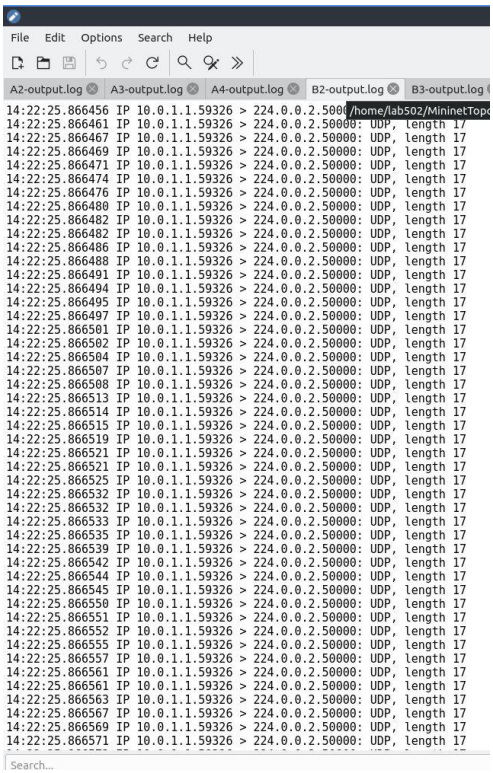
下圖由左至右分別為 A2-output.log、A3-output.log、A4-output.log

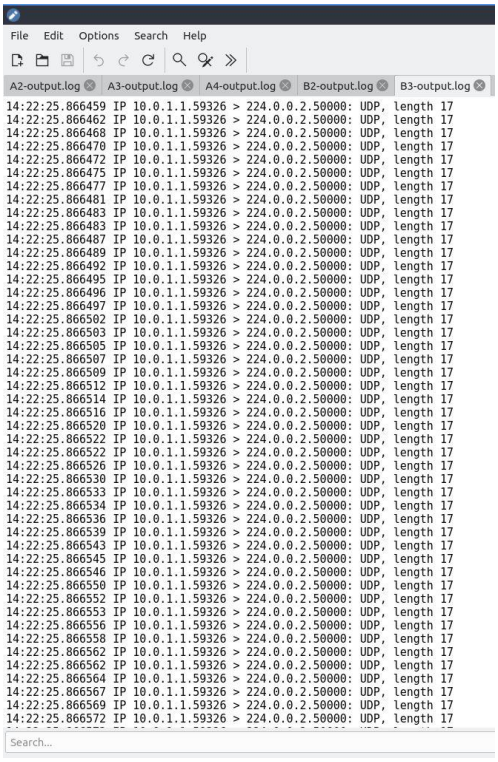


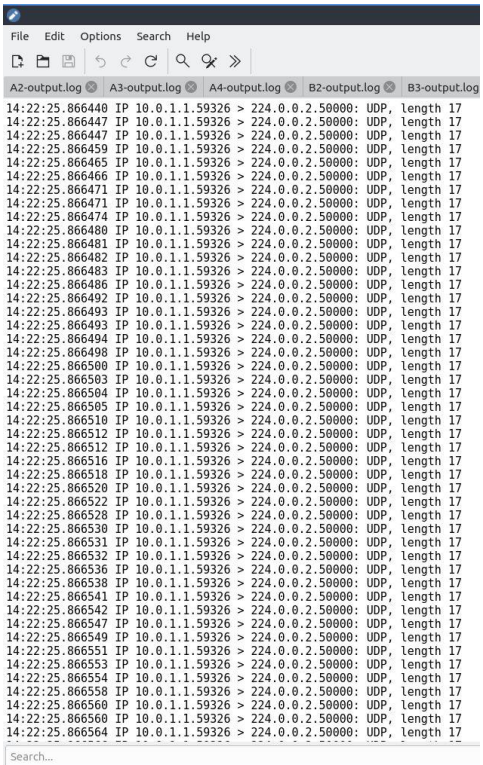




下圖由左至右分別為 B2-output.log、B3-output.log 、B4-output.log







作業:

步驟 1.

開啟 Terminal，並 cd 到 MininetTopology 的資料夾中，然後輸入 sudo python3 ./Homework.py，如下圖 1。

```
alan@alan-VirtualBox:~/MininetTopology$ sudo python3 ./Homework.py
(1000.00Mbit) (1000.00Mbit) (1000.00Mbit) (1000.00Mbit) (100.00Mbit) (100.00Mbit
) (100.00Mbit) (100.00Mbit) (100.00Mbit) (100.00Mbit) (100.00Mbit) (100.00Mbit)
(100.00Mbit) (100.00Mbit) (100.00Mbit) (100.00Mbit) *** Configuring hosts
h1 h2
*** Starting controller

*** Starting 5 switches
s1 (1000.00Mbit) (100.00Mbit) (100.00Mbit) (100.00Mbit) s2 (100.00Mbit) (100.00M
bit) s3 (100.00Mbit) (100.00Mbit) s4 (100.00Mbit) (100.00Mbit) s5 (1000.00Mbit)
(100.00Mbit) (100.00Mbit) (100.00Mbit) ...(1000.00Mbit) (100.00Mbit) (100.00Mbit
) (100.00Mbit) (100.00Mbit) (100.00Mbit) (100.00Mbit) (100.00Mbit) (100.00Mbit)
(100.00Mbit) (1000.00Mbit) (100.00Mbit) (100.00Mbit) (100.00Mbit)
*** h2 : ('iperf -s -u -p 50001 > 50001-s.log &',)
*** h2 : ('iperf -s -u -p 50002 > 50002-s.log &',)
*** h2 : ('iperf -s -u -p 50003 > 50003-s.log &',)
*** h1 : ('arp -s 10.0.0.2 00:05:00:00:00:02',)
*** h2 : ('arp -s 10.0.0.1 00:04:00:00:00:02',)
*** Starting CLI:
mininet>
```

▲圖 1

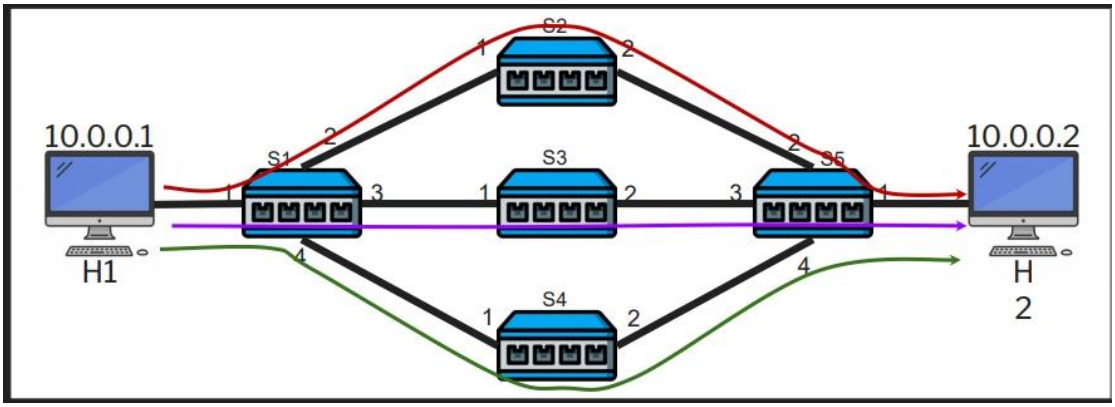
步驟 2.

根據下圖 3 和下圖 4 的拓譜圖和規則，來輸入指令編寫 flow(如下圖 2)。

```
lab502@lab502:~/MininetTopology$ sudo python3 ./Homework.py
(1000.00Mbit) (1000.00Mbit) (1000.00Mbit) (1000.00Mbit) (100.00Mbit) (100.00Mbit) (100.00Mbit) (100.00Mbit) (100.00Mbit) (100.00Mbi
) (100.00Mbit) (100.00Mbit) (100.00Mbit) (100.00Mbit) (100.00Mbit) *** Configuring hosts
h1 h2
*** Starting controller

*** Starting 5 switches
s1 (1000.00Mbit) (100.00Mbit) (100.00Mbit) (100.00Mbit) s2 (100.00Mbit) (100.00Mbit) s3 (100.00Mbit) (100.00Mbit) s4 (100.00Mbit) (100.00Mbit) s
 (1000.00Mbit) (100.00Mbit) (100.00Mbit) (100.00Mbit) ...(1000.00Mbit) (100.00Mbit) (100.00Mbit) (100.00Mbit) (100.00Mbit) (100.00Mbit) (100.00M
it) (100.00Mbit) (100.00Mbit) (100.00Mbit) (1000.00Mbit) (100.00Mbit) (100.00Mbit) (100.00Mbit)
*** h2 : ('iperf -s -u -p 50001 > 50001-s.log &',)
[1] 45450
*** h2 : ('iperf -s -u -p 50002 > 50002-s.log &',)
[2] 45451
*** h2 : ('iperf -s -u -p 50003 > 50003-s.log &',)
[3] 45452
*** h1 : ('arp -s 10.0.0.2 00:05:00:00:00:02',)
*** h2 : ('arp -s 10.0.0.1 00:04:00:00:00:02',)
*** Starting CLI:
mininet> sh ovs-ofctl add-meter -O OpenFlow13 s1 meter=2,kbps,band=type=drop,rate=20000
mininet> sh ovs-ofctl add-meter -O OpenFlow13 s1 meter=1,kbps,band=type=drop,rate=50000
mininet> sh ovs-ofctl add-meter -O OpenFlow13 s1 meter=3,kbps,band=type=drop,rate=30000
mininet> sh ovs-ofctl add-flow s1 -O OpenFlow13 dl_type=0x0800,nw_dst=10.0.0.2,nw_proto=17,tp_dst=50001,actions=meter:1,output:2
mininet> sh ovs-ofctl add-flow s1 -O OpenFlow13 dl_type=0x0800,nw_dst=10.0.0.2,nw_proto=17,tp_dst=50002,actions=meter:2,output:3
mininet> sh ovs-ofctl add-flow s1 -O OpenFlow13 dl_type=0x0800,nw_dst=10.0.0.2,nw_proto=17,tp_dst=50003,actions=meter:3,output:4
mininet> sh ovs-ofctl add-flow s2 dl_type=0x0800,nw_dst=10.0.0.2,nw_proto=17,tp_dst=50001,actions=output:2
mininet> sh ovs-ofctl add-flow s5 dl_type=0x0800,nw_dst=10.0.0.2,nw_proto=17,tp_dst=50001,actions=output:1
mininet> sh ovs-ofctl add-flow s3 dl_type=0x0800,nw_dst=10.0.0.2,nw_proto=17,tp_dst=50002,actions=output:2
mininet> sh ovs-ofctl add-flow s4 dl_type=0x0800,nw_dst=10.0.0.2,nw_proto=17,tp_dst=50003,actions=output:2
mininet> sh ovs-ofctl add-flow s5 dl_type=0x0800,nw_dst=10.0.0.2,nw_proto=17,tp_dst=50001,actions=output:1
mininet> sh ovs-ofctl add-flow s5 dl_type=0x0800,nw_dst=10.0.0.2,nw_proto=17,tp_dst=50002,actions=output:1
mininet> sh ovs-ofctl add-flow s5 dl_type=0x0800,nw_dst=10.0.0.2,nw_proto=17,tp_dst=50003,actions=output:1
mininet> xterm h1
mininet>
```

▲圖 2


S2
1
2
10.0.0.1
S1
2
2
S5
10.0.0.2
1
3
1
S3
2
3
1
H1
4
4
S4
1
2
H
2

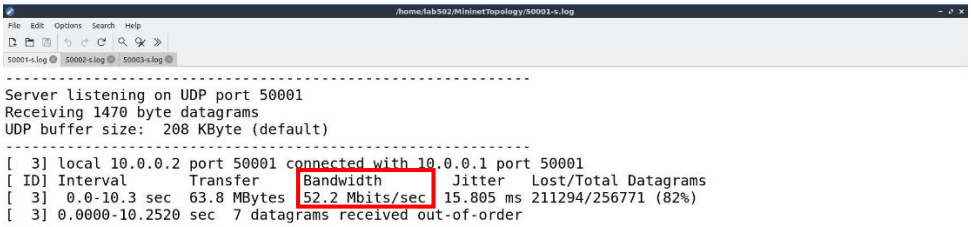
▲圖 3

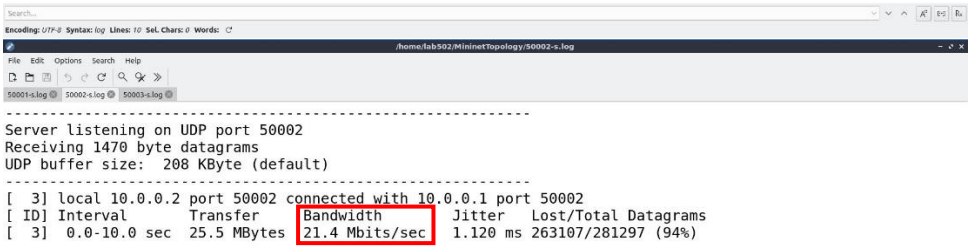

紅色： UDP Port ID 50001
藍色： UDP Port ID 50002
綠色： UDP Port ID 50003

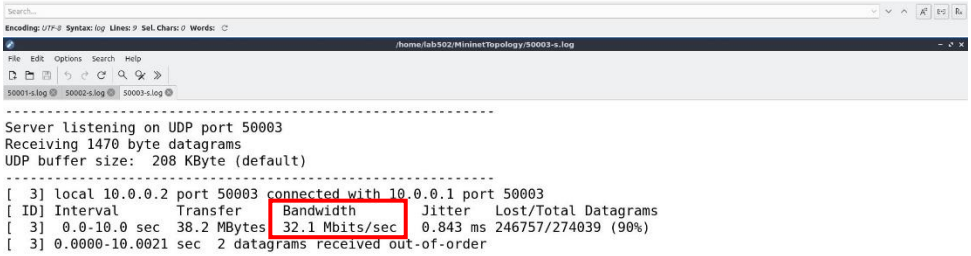
▲圖 4

步驟 3.

觀察 50001-s.log, 50002-s.log, 50003-s.log 之檔案內容並擷圖:







▲圖 5

5. 問題與討論

如何使用 Meter 限制流量？

在這個實驗中，使用 Meter 限制流量，可以有效地避免過量流量的問題，從而提高網路效能。然而，如何確定 Meter 的設定值，以及如何動態調整 Meter 的設定值，是一個需要討論的問題，因為過高或過低的設定值都可能對網路效能產生負面影響。如果 Meter 的設定值太低，可能會導致網路頻繁地進入流量限制狀態，影響網路的實際傳輸速率。相反，如果 Meter 的設定值太高，可能會導致某些高流量應用程序佔據了大量帶寬，從而影響其他應用程序的網路性能。

如何處理網路中的故障和延遲？

在這個實驗中，沒有考慮網路中的故障和延遲等問題。當網路中出現故障或延遲時，需要及時處理，以維護網路的穩定性和可用性。在這個問題上，需要更進一步的研究和討論，以提高網路的可靠性。

路由規則（Flow entries）的設定和操作：編寫路由的規則是很重要的，因此查詢下圖 6 是很重要的步驟和參考工具。

Match Fields	Descriptions
in_port=port	OpenFlow port ID
dl_vlan=vlan	Virtual LAN tag ID
dl_src=xx:xx:xx:xx:xx:xx dl_dst=xx:xx:xx:xx:xx:xx	Ethernet source (or destination) address (MAC addresses)
dl_type=ethertype	Ethernet protocol type
nw_src=ip[/netmask] nw_dst=ip[/netmask]	When **dl_type=0x0800**, matches IPv4 source (or destination) address.
arp_spa=ip arp_tpa=ip	When **dl_type=0x0806** is specified, matches the arp_spa or arp_tpa field. (IPv4 Addresses)
nw_proto=proto	When **dl_type=0x0800** is specified, matches IP protocol type proto, which is specified as a decimal number between 0 and 255
tp_src=port tp_dst=port	When dl_type and nw_proto specify TCP(6) or UD(17), tp_src and tp_dst match the UDP or TCP source or destination port
arp_sha=xx:xx:xx:xx:xx:xx arp_tha=xx:xx:xx:xx:xx:xx	When dl_type=0x0806, arp_sha and arp_tha match the source and target hardware address, respectively

Actions	Descriptions
output:port	Outputs the packet to port
normal	Subjects the packet to the device's normal L2/L3 processing.
flood	Outputs the packet on all switch physical ports other than the port on which it was received and any ports on which flooding is disabled
all	Outputs the packet on all switch physical ports other than the port on which it was received.
in_port	Outputs the packet on the port from which it was received.
drop	Discards the packet, so no further processing or forwarding takes place.
set_field:value->dst	Writes the literal value into the field dst, which should be specified as a name used for matching. Example: set_field:fe80:0123:4567:890a:a6ba:dbff:fefe:59fa-\>ipv6_src

▲圖 6

6. 心得與感想

在課堂上的實驗中，我們學習了如何實現 IPv4 Multicasting，讓封包可以同時發送給多個主機，從而實現有效的資料傳輸。通過在 A1 和 B1 終端機上執行 multicast_script.py，我們可以將封包發送到 Multicast group 中，並使用記錄檔記錄發送和接收的封包。這讓我們可以方便地評估實驗的有效性，並進一步優化和調整 Multicast 的設置。

另外，透此作業的實驗，我了解到如何使用 OpenVSwitch 指令對連線加入路由規則，並且利用 Meter 限制流量，從而避免過量流量的問題，進而提高網路效能。這是一個很實用的技術，特別是當網路擁有大量使用者時，避免過量流量的問題可以有效地避免網路擁塞。同時，我也發現 Meter 的設定值很重要，必須要根據實際情況進行調整，才能發揮最大效益。總之，這個實驗讓我更深入地了解網路流量管理的相關知識，並且增強了我的技術能力。

7. 參考文獻

http://github.com/Hsun111/MininetTopology

https://www.vmware.com/tw/topics/glossary/content/software-defined-networking.html

https://ithelp.ithome.com.tw/articles/10235434

https://www.openedu.tw/course?id=1002