

## 檢視 Ubuntu 版本

```
lsb_release -a
```

```
pinu@pinu-virtual-machine:~$ lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description:    Ubuntu 18.04.3 LTS
Release:        18.04
Codename:       bionic
```

## Part1:

### 1. 安裝 mininet

```
apt-get install mininet
```

```
root@pinu-virtual-machine:/home/pinu
檔案(F) 編輯(E) 檢視(V) 搜尋(S) 終端機(T) 助手(H)
pinu@pinu-virtual-machine:~$ sudo su
[sudo] password for pinu:
root@pinu-virtual-machine:/home/pinu# apt-get install mininet
正在讀取套件清單... 完成
正在建立相依關係...
正在讀取狀態資料... 完成
正在識別將要被安裝的套件...
下列的額外套件將會被安裝：
  cgroup-bin cgroup-tools iperf libcgroup1 openvswitch-common
  openvswitch-switch socat
建議套件：
  ethtool openvswitch-doc
下列「新」套件將會被安裝：
  cgroup-bin cgroup-tools iperf libcgroup1 mininet openvswitch-common
  openvswitch-switch socat
升級 0 個，新安裝 8 個，移除 0 個，有 68 個未被升級。
需要下載 3,011 kB 的套件檔。
此操作完成之後，會多佔用 12.6 MB 的磁碟空間。
是否繼續進行 [Y/n] ? [Y/n] Y
下載:1 http://tw.archive.ubuntu.com/ubuntu bionic/universe amd64 libcgroup1 amd64 0.41-8ubuntu2 [42.0 kB]
下載:2 http://tw.archive.ubuntu.com/ubuntu bionic/universe amd64 cgroup-tools amd64 0.41-8ubuntu2 [66.2 kB]
下載:3 http://tw.archive.ubuntu.com/ubuntu bionic-updates/universe amd64 iperf amd64 2.0.10+dfsg1-1ubuntu0.18.04.2 [60.5 kB]
下載:4 http://tw.archive.ubuntu.com/ubuntu bionic/universe amd64 cgroup-bin all 0.41-8ubuntu2 [2,576 B]
下載:5 http://tw.archive.ubuntu.com/ubuntu bionic/main amd64 socat amd64 1.7.3.2-2ubuntu2 [342 kB]
下載:6 http://tw.archive.ubuntu.com/ubuntu bionic/universe amd64 mininet amd64 2.2.2-2ubuntu1 [178 kB]
下載:7 http://tw.archive.ubuntu.com/ubuntu bionic-updates/main amd64 openvswitch-common amd64 2.9.2-0ubuntu0.18.04.3 [814 kB]
下載:8 http://tw.archive.ubuntu.com/ubuntu bionic-updates/main amd64 openvswitch-switch amd64 2.9.2-0ubuntu0.18.04.3 [1,506 kB]
取得 3,011 kB 用了 1秒 (2,887 kB/s)
選取了原先未選的套件 libcgroup1:amd64。
(讀取資料庫... 目前共安裝了 169251 個檔案和目錄。)
準備解開 .../0-libcgroup1:0.41-8ubuntu2_amd64.deb ...
解開 libcgroup1:amd64 (0.41-8ubuntu2) 中...
選取了原先未選的套件 cgroup-tools。
準備解開 .../1-cgroup-tools_0.41-8ubuntu2_amd64.deb ...
解開 cgroup-tools (0.41-8ubuntu2) 中...
選取了原先未選的套件 iperf。
準備解開 .../2-iperf_2.0.10+dfsg1-1ubuntu0.18.04.2_amd64.deb ...
解開 iperf (2.0.10+dfsg1-1ubuntu0.18.04.2) 中...
選取了原先未選的套件 cgroup-bin。
準備解開 .../3-cgroup-bin_0.41-8ubuntu2_all.deb ...
解開 cgroup-bin (0.41-8ubuntu2) 中...
選取了原先未選的套件 socat。
準備解開 .../4-socat_1.7.3.2-2ubuntu2_amd64.deb ...
解開 socat (1.7.3.2-2ubuntu2) 中...
選取了原先未選的套件 mininet。
準備解開 .../5-mininet_2.2.2-2ubuntu1_amd64.deb ...
```

2. 建立基本的虛擬拓樸

mn



The screenshot shows a terminal window titled "root@pinu-virtual-machine: /home/pinu". The window contains the following text output from the "mn" command:

```
檔案(F) 編輯(E) 檢視(V) 搜尋(S) 終端機(T) 求助(H)
root@pinu-virtual-machine:/home/pinu# mn
*** No default OpenFlow controller found for default switch!
*** Falling back to OVS Bridge
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1)
*** Configuring hosts
h1 h2
*** Starting controller

*** Starting 1 switches
s1 ...
*** Starting CLI:
mininet> 
```

3. 顯示現在的虛擬拓樸中節點資訊或鏈結的訊息

help

```
mininet> help

Documented commands (type help <topic>):
=====
EOF      gterm    iperfudp   nodes      pingpair      py      switch
dpctl    help     link       noecho     pingpairfull  quit     time
dump     intfs    links      pingall     ports       sh      x
exit     iperf    net       pingallfull px      source   xterm

You may also send a command to a node using:
  <node> command {args}
For example:
  mininet> h1 ifconfig

The interpreter automatically substitutes IP addresses
for node names when a node is the first arg, so commands
like
  mininet> h2 ping h3
should work.

Some character-oriented interactive commands require
noecho:
  mininet> noecho h2 vi foo.py
However, starting up an xterm/gterm is generally better:
  mininet> xterm h2
```

help nodes

```
mininet> help nodes
List all nodes.
```

help net

```
mininet> help net
List network connections.
```

help dump

```
mininet> help dump
Dump node info.
```

下載 wireshark

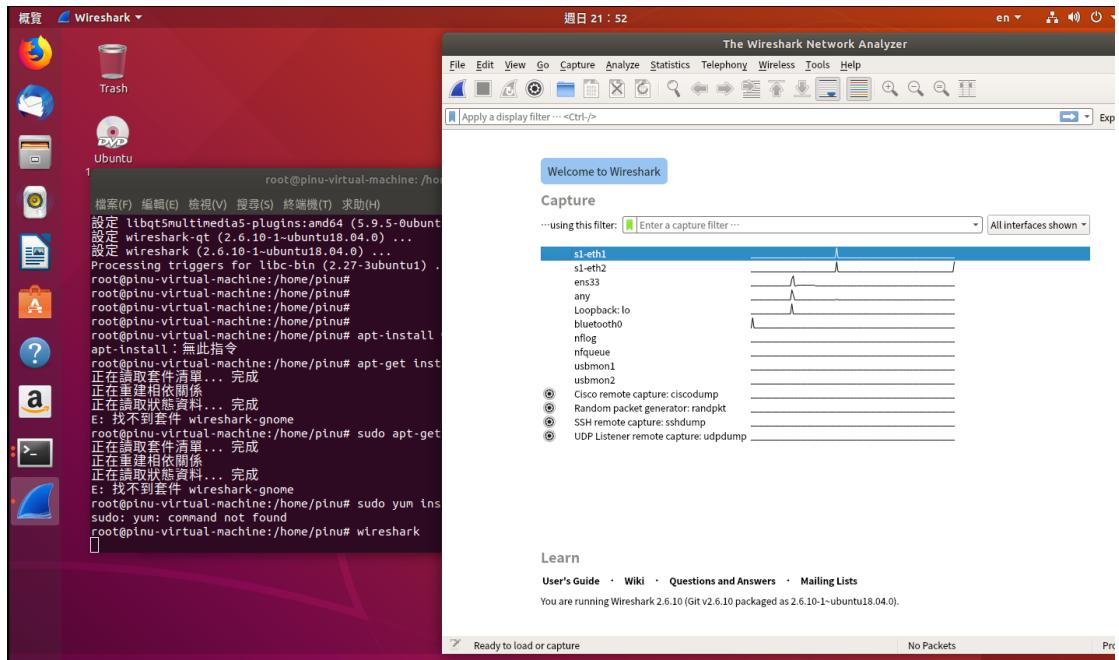
apt-get install wireshark

抓包基本功能，無法使用 wireshark 命令和圖形介面

```
pinu@pinu-virtual-machine:~$ sudo su
[sudo] password for pinu:
root@pinu-virtual-machine:/home/pinu# apt-get install wireshark
正在讀取套件清單... 完成
正在重建相依關係
正在讀取狀態資料... 完成
下列的額外套件將被安裝：
  libc-ares2 libdouble-conversion1 libmaxminddb0 libnl-route-3-200 libqgsttools-
p1 libqt5core5a libqt5dbus5 libqt5gui5 libqt5multimedia5
  libqt5multimedia5-plugins libqt5multimediawidgets5 libqt5network5 libqt5opengl
5 libqt5printsupport5 libqt5svg5 libqt5widgets5 libsmi2ldbl
  libspandsp2 libwireshark-data libwireshark11 libwiretap8 libwscodecs2 libwsuti
l9 libxcb-xinerama0 qt5-gtk-platformtheme qttranslations5-l10n
  wireshark-common wireshark-qt
建議套件：
  mmbdb-bin qt5-image-formats-plugins qtwayland5 snmp-mibs-downloader wireshark-d
oc
下列【新】套件將會被安裝：
  libc-ares2 libdouble-conversion1 libmaxminddb0 libnl-route-3-200 libqgsttools-
p1 libqt5core5a libqt5dbus5 libqt5gui5 libqt5multimedia5
```

開啟 wireshark

wireshark



#### 4. 摄取兩個虛擬 host 的網卡

h1 : 32:36:d4:6b:86:49

Frame 434: 100 bytes on wire (800 bits), 100 bytes captured (800 bits) on interface 0

Linux cooked capture

Packet type: Unicast to another host (3)

Link-layer address type: 1

Link-layer address length: 6

Source: 32:36:d4:6b:86:49 (32:36:d4:6b:86:49)

Unused: 0000

Protocol: IPv4 (0x0800)

Internet Protocol Version 4, Src: 10.0.0.1, Dst: 10.0.0.2

0100 .... = Version: 4

.... 0101 = Header Length: 20 bytes (5)

Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)

0000 00.. = Differentiated Services Codepoint: Default (0)

.... ..00 = Explicit Congestion Notification: Not ECN-Capable Transport (0)

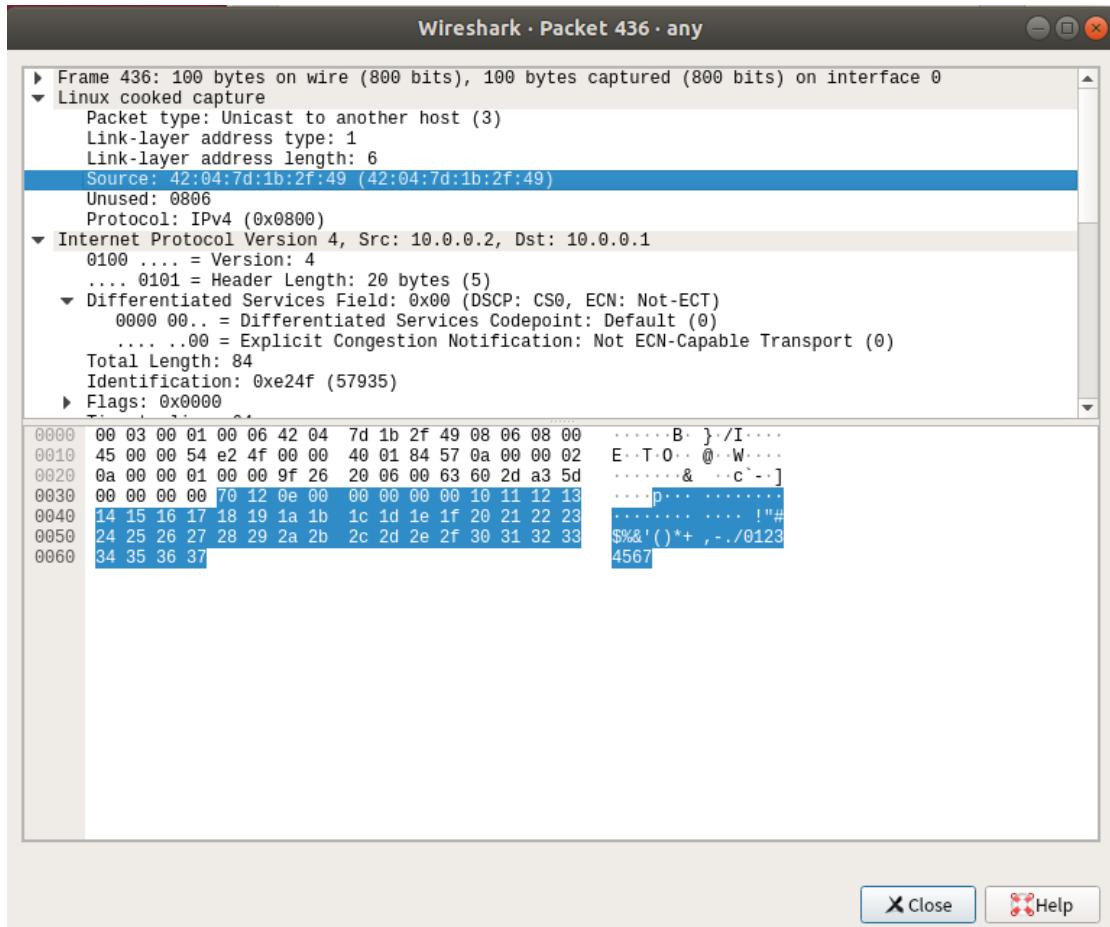
Total Length: 84

Identification: 0xd934 (55604)

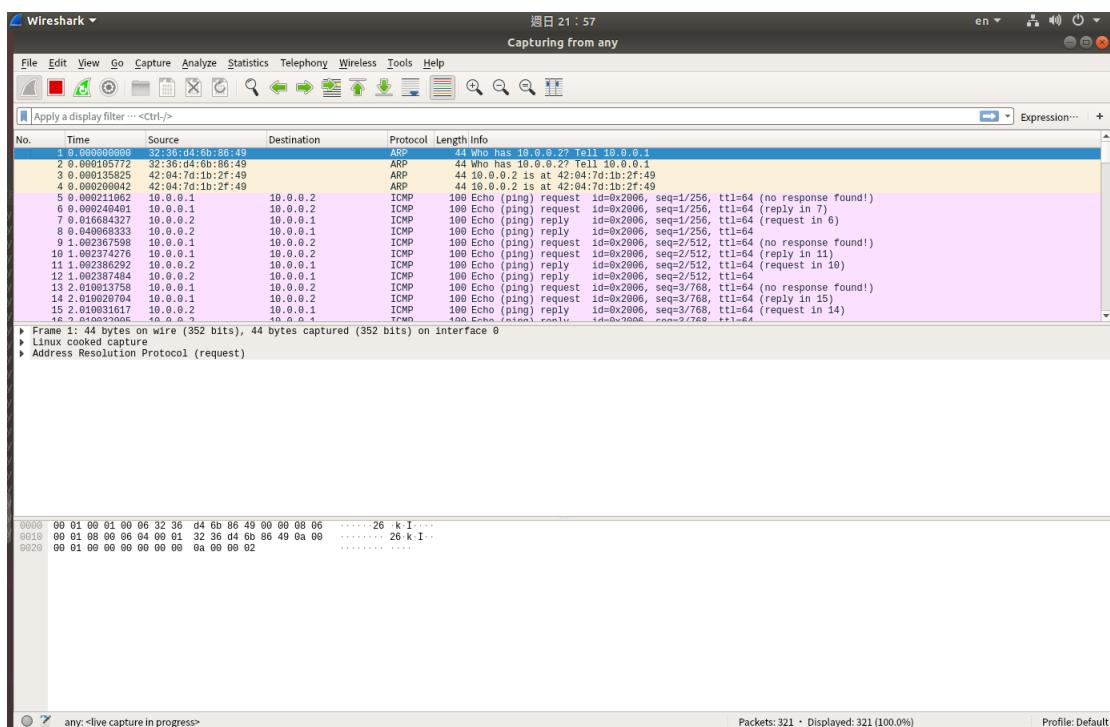
Flags: 0x4000, Don't fragment

Hex	Dec	ASCII
0000	00 03 00 01 00 06 32 36	d4 6b 86 49 00 00 08 00
0010	45 00 00 54 d9 34 40 00	40 01 4d 72 0a 00 00 01
0020	0a 00 00 02 08 00 97 26	20 06 00 63 60 2d a3 5d
0030	00 00 00 00 70 12 0e 00	00 00 00 00 10 11 12 13
0040	14 15 16 17 18 19 1a 1b	1c 1d 1e 1f 20 21 22 23
0050	24 25 26 27 28 29 2a 2b	2c 2d 2e 2f 30 31 32 33
0060	34 35 36 37	\$%&!'()*+, -./0123 4567

h2: 42:04:7d:1b:2f:49



h1 ping h2



Part2:

下載 tshark

```
sudo apt install tshark
```

```
pinu@pinu-virtual-machine:~$ sudo apt install tshark
[sudo] password for pinu:
正在讀取套件清單... 完成
正在重建相依關係
正在讀取狀態資訊... 完成
下列【新】套件將會被安裝：
  tshark
升級 0 個，新安裝 1 個，移除 0 個，有 68 個未被升級。
需要下載 134 kB 的套件檔。
此操作完成之後，會多佔用 389 kB 的磁碟空間。
下載:1 http://tw.archive.ubuntu.com/ubuntu bionic-updates/universe amd64 tshark amd64 2.6.10-1~ubuntu18.04.0 [134 kB]
取得 134 kB 用了 1秒 (171 kB/s)
選取了原先未選的套件 tshark。
(讀取資料庫 ... 目前共安裝了 170468 個檔案和目錄。)
準備解開 .../tshark_2.6.10-1~ubuntu18.04.0_amd64.deb ...
解開 tshark (2.6.10-1~ubuntu18.04.0) 中...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
設定 tshark (2.6.10-1~ubuntu18.04.0) ...
```

切換為 root，並且切換到 跟目錄位置。

執行 tshark 指令，才會有權限

```
pinu@pinu-virtual-machine:~$ sudo su
[sudo] password for pinu:
root@pinu-virtual-machine:/home/pinu# cd /
root@pinu-virtual-machine:/#
```

1. a. 抓取 icmp 封包，來源或目的是 8.8.8.8

```
root@pinu-virtual-machine: /
```

檔案(F) 編輯(E) 檢視(V) 搜尋(S) 終端機(T) 求助(H)

```
pinu@pinu-virtual-machine:~$ sudo su
[sudo] password for pinu:
root@pinu-virtual-machine:/home/pinu# cd /
root@pinu-virtual-machine:/# tshark -Y "icmp && ip.addr==8.8.8.8"
Running as user "root" and group "root". This could be dangerous.
Capturing on 'ens33'
```

```
root@pinu-virtual-machine: /
```

檔案(F) 編輯(E) 檢視(V) 搜尋(S) 終端機(T) 求助(H)

```
root@pinu-virtual-machine:/# tshark -i any -w packet01.csv
Running as user "root" and group "root". This could be dangerous.
Capturing on 'any'
635 [ ]
```

```
root@pinu-virtual-machine: /
```

檔案(F) 編輯(E) 檢視(V) 搜尋(S) 終端機(T) 求助(H)

```
d=0x1cb9, seq=66/15360, ttl=128 (request in 429) 433 617.020864366 192.168.79.131 →8.8.8.8 ICMP 98 Echo (ping) request i
d=0x1cb9, seq=61/15616, ttl=64 434 617.027220869 8.8.8.8 →192.168.79.131 ICMP 98 Echo (ping) reply i
d=0x1cb9, seq=61/15616, ttl=128 (request in 433) 437 618.022466187 192.168.79.131 →8.8.8.8 ICMP 98 Echo (ping) request i
d=0x1cb9, seq=62/15872, ttl=64 438 618.029210159 8.8.8.8 →192.168.79.131 ICMP 98 Echo (ping) reply i
d=0x1cb9, seq=62/15872, ttl=128 (request in 437) 441 619.025733540 192.168.79.131 →8.8.8.8 ICMP 98 Echo (ping) request i
d=0x1cb9, seq=63/16128, ttl=64 442 619.032497098 8.8.8.8 →192.168.79.131 ICMP 98 Echo (ping) reply i
d=0x1cb9, seq=63/16128, ttl=128 (request in 441) 445 620.027570799 192.168.79.131 →8.8.8.8 ICMP 98 Echo (ping) request i
d=0x1cb9, seq=64/16384, ttl=64 446 620.034317237 8.8.8.8 →192.168.79.131 ICMP 98 Echo (ping) reply i
d=0x1cb9, seq=64/16384, ttl=128 (request in 445) 453 621.030602716 192.168.79.131 →8.8.8.8 ICMP 98 Echo (ping) request i
d=0x1cb9, seq=65/16640, ttl=64 454 621.036518087 8.8.8.8 →192.168.79.131 ICMP 98 Echo (ping) reply i
d=0x1cb9, seq=65/16640, ttl=128 (request in 453) 467 622.032241317 192.168.79.131 →8.8.8.8 ICMP 98 Echo (ping) request i
d=0x1cb9, seq=66/16896, ttl=64 468 622.038755907 8.8.8.8 →192.168.79.131 ICMP 98 Echo (ping) reply i
```

```
tshark -Y "icmp && ip.addr==8.8.8.8"
```

-Y <display filter>

b. 將擷取封包儲存為一個檔案，名稱為 packet01

```
root@pinu-virtual-machine:/# tshark -i any -w packet01.csv
Running as user "root" and group "root". This could be dangerous.
Capturing on 'any'
18 [
```

tshark -I any -w packet01.csv

-I 指定網卡

-w 寫入的檔案名 (packet01.csv)

### c. ping 8.8.8.8 及 ping 208.67.220.220

The screenshot shows two terminal windows side-by-side. Both windows have the title 'pinu@pinu-virtual-machine: ~' and a menu bar with '檔案(F)', '編輯(E)', '檢視(V)', '搜尋(S)', '終端機(T)', and '求助(H)'. The left terminal window displays the command 'tshark -I any -w packet01.csv' and its output, which consists of numerous ICMP echo requests (ICMP seq=27 to 49) sent from 8.8.8.8 to 208.67.220.220. The right terminal window also displays the same command and output, showing ICMP echo requests from 208.67.220.220 to 8.8.8.8.

```
pinu@pinu-virtual-machine: ~
檔案(F) 編輯(E) 檢視(V) 搜尋(S) 終端機(T) 求助(H)
64 bytes from 8.8.8.8: icmp_seq=27 ttl=128 time=6.45 ms
64 bytes from 8.8.8.8: icmp_seq=28 ttl=128 time=6.27 ms
64 bytes from 8.8.8.8: icmp_seq=29 ttl=128 time=6.46 ms
64 bytes from 8.8.8.8: icmp_seq=30 ttl=128 time=16.6 ms
64 bytes from 8.8.8.8: icmp_seq=31 ttl=128 time=6.46 ms
64 bytes from 8.8.8.8: icmp_seq=32 ttl=128 time=6.30 ms
64 bytes from 8.8.8.8: icmp_seq=33 ttl=128 time=6.32 ms
64 bytes from 8.8.8.8: icmp_seq=34 ttl=128 time=6.48 ms
64 bytes from 8.8.8.8: icmp_seq=35 ttl=128 time=6.41 ms
64 bytes from 8.8.8.8: icmp_seq=36 ttl=128 time=6.69 ms
64 bytes from 8.8.8.8: icmp_seq=37 ttl=128 time=6.52 ms
64 bytes from 8.8.8.8: icmp_seq=38 ttl=128 time=6.23 ms
64 bytes from 8.8.8.8: icmp_seq=39 ttl=128 time=6.53 ms
64 bytes from 8.8.8.8: icmp_seq=40 ttl=128 time=6.58 ms
64 bytes from 8.8.8.8: icmp_seq=41 ttl=128 time=6.57 ms
64 bytes from 8.8.8.8: icmp_seq=42 ttl=128 time=8.01 ms
64 bytes from 8.8.8.8: icmp_seq=43 ttl=128 time=6.26 ms
64 bytes from 8.8.8.8: icmp_seq=44 ttl=128 time=6.16 ms
64 bytes from 8.8.8.8: icmp_seq=45 ttl=128 time=6.25 ms
64 bytes from 8.8.8.8: icmp_seq=46 ttl=128 time=6.47 ms
64 bytes from 8.8.8.8: icmp_seq=47 ttl=128 time=6.26 ms
64 bytes from 8.8.8.8: icmp_seq=48 ttl=128 time=6.57 ms
64 bytes from 8.8.8.8: icmp_seq=49 ttl=128 time=6.79 ms
64 bytes from 208.67.220.220: icmp_seq=7 ttl=128 time=28.4 ms
64 bytes from 208.67.220.220: icmp_seq=8 ttl=128 time=27.9 ms
64 bytes from 208.67.220.220: icmp_seq=9 ttl=128 time=32.9 ms
64 bytes from 208.67.220.220: icmp_seq=10 ttl=128 time=70.5 ms
64 bytes from 208.67.220.220: icmp_seq=11 ttl=128 time=30.4 ms
64 bytes from 208.67.220.220: icmp_seq=12 ttl=128 time=27.3 ms
64 bytes from 208.67.220.220: icmp_seq=13 ttl=128 time=36.0 ms
64 bytes from 208.67.220.220: icmp_seq=14 ttl=128 time=27.3 ms
64 bytes from 208.67.220.220: icmp_seq=15 ttl=128 time=27.8 ms
64 bytes from 208.67.220.220: icmp_seq=16 ttl=128 time=27.6 ms
64 bytes from 208.67.220.220: icmp_seq=17 ttl=128 time=31.7 ms
64 bytes from 208.67.220.220: icmp_seq=18 ttl=128 time=27.4 ms
64 bytes from 208.67.220.220: icmp_seq=19 ttl=128 time=31.8 ms
64 bytes from 208.67.220.220: icmp_seq=20 ttl=128 time=27.9 ms
64 bytes from 208.67.220.220: icmp_seq=21 ttl=128 time=29.5 ms
64 bytes from 208.67.220.220: icmp_seq=22 ttl=128 time=27.8 ms
64 bytes from 208.67.220.220: icmp_seq=23 ttl=128 time=28.6 ms
64 bytes from 208.67.220.220: icmp_seq=24 ttl=128 time=32.9 ms
64 bytes from 208.67.220.220: icmp_seq=25 ttl=128 time=32.9 ms
64 bytes from 208.67.220.220: icmp_seq=26 ttl=128 time=27.6 ms
64 bytes from 208.67.220.220: icmp_seq=27 ttl=128 time=34.8 ms
64 bytes from 208.67.220.220: icmp_seq=28 ttl=128 time=27.7 ms
64 bytes from 208.67.220.220: icmp_seq=29 ttl=128 time=41.5 ms
```

#### d. 使用 tshark 查看擷取的檔案

tshark -r packet01.csv

[ -r <infile> ]

[ -R <Read filter> ]

```
root@pinu-virtual-machine:/# tshark -r packet01.csv
Running as user "root" and group "root". This could be dangerous.
  1. 0.000000000 127.0.0.1 →127.0.0.53 DNS 102 Standard query 0x147d A connectivity-check.ubuntu.com OPT
  2. 0.000023765 127.0.0.1 →127.0.0.53 DNS 102 Standard query 0x540f AAAA connectivity-check.ubuntu.com OPT
  3. 0.000434595 192.168.79.131 →192.168.79.2 DNS 102 Standard query 0x5d41 A connectivity-check.ubuntu.com OPT
  4. 0.000634094 192.168.79.131 →192.168.79.2 DNS 102 Standard query 0xc7a8 AAAA connectivity-check.ubuntu.com OPT
  5. 0.001481792 192.168.79.2 →192.168.79.131 DNS 134 Standard query response 0x5d41 A connectivity-check.ubuntu.com A 35.222.85.5 A 35.224.99.156 OPT
T
  6. 0.001548571 192.168.79.2 →192.168.79.131 DNS 163 Standard query response 0xc7a8 AAAA connectivity-check.ubuntu.com SOA ns1.canonical.com OPT
  7. 0.001647721 127.0.0.53 →127.0.0.1 DNS 134 Standard query response 0x147d A connectivity-check.ubuntu.com A 35.222.85.5 A 35.224.99.156 OPT
  8. 0.001717027 127.0.0.53 →127.0.0.1 DNS 102 Standard query response 0x540f AAAA connectivity-check.ubuntu.com OPT
  9. 1.001811141 192.168.79.131 →35.222.85.5 TCP 76 49422 →80 [SYN] Seq=0 Win=42340 Len=0 MSS=1460 SACK_PERM=1 TSeq=0 TSecr=0 WS=512
  10. 1.001811141 Vmware_e2:79:f1 → ARP 62 Who has 192.168.79.131? Tell 192.168.79.2
  11. 1.165034663 Vmware_d8:b4:e9 → ARP 44 192.168.79.131 is at 00:0c:29:d8:b4:e9
  12. 1.165191359 35.222.85.5 →192.168.79.131 TCP 56 80 →49422 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
  13. 1.165221099 192.168.79.131 →35.222.85.5 TCP 56 49422 →80 [ACK] Seq=1 Ack=1 Win=42340 Len=0
  14. 1.165333111 192.168.79.131 →35.222.85.5 HTTP 143 GET / HTTP/1.1
  15. 1.165609935 35.222.85.5 →192.168.79.131 TCP 56 80 →49422 [ACK] Seq=1 Ack=88 Win=64240 Len=0
  16. 1.328595604 35.222.85.5 →192.168.79.131 HTTP 204 HTTP/1.1 204 No Content
  17. 1.328781757 192.168.79.131 →35.222.85.5 TCP 56 49422 →80 [FIN, ACK] Seq=88 Ack=150 Win=42191 Len=0
  18. 1.328953254 35.222.85.5 →192.168.79.131 TCP 56 80 →49422 [ACK] Seq=150 Ack=89 Win=64239 Len=0
  19. 39.712915867 Vmware_c0:00:08 → ARP 62 Who has 192.168.79.2? Tell 192.168.79.1
  20. 40.771195629 Vmware_c0:00:08 → ARP 62 Who has 192.168.79.2? Tell 192.168.79.1
  21. 41.712876127 Vmware_c0:00:08 → ARP 62 Who has 192.168.79.2? Tell 192.168.79.1
  22. 42.713073580 Vmware_c0:00:08 → ARP 62 Who has 192.168.79.2? Tell 192.168.79.1
  23. 43.765963840 192.168.79.1 →239.255.255.250 SSDP 218 M-SEARCH * HTTP/1.1
  24. 43.773237310 Vmware_c0:00:08 → ARP 62 Who has 192.168.79.2? Tell 192.168.79.1
  25. 44.71331514 Vmware_c0:00:08 → ARP 62 Who has 192.168.79.2? Tell 192.168.79.1
  26. 44.767750599 192.168.79.1 →239.255.255.250 SSDP 218 M-SEARCH * HTTP/1.1
  27. 45.712976231 Vmware_c0:00:08 → ARP 62 Who has 192.168.79.2? Tell 192.168.79.1
  28. 45.768421084 192.168.79.1 →239.255.255.250 SSDP 218 M-SEARCH * HTTP/1.1
  29. 46.708417219 192.168.79.1 →239.255.255.250 SSDP 218 M-SEARCH * HTTP/1.1
  688 376.680942902 192.168.79.131 →208.67.220.220 ICMP 100 Echo (ping) request id=0x1cce, seq=121/30976, ttl=64
  689 376.709052610 208.67.220.220 →192.168.79.131 ICMP 100 Echo (ping) reply id=0x1cce, seq=121/30976, ttl=128 (request in 688)
  690 376.891609541 192.168.79.131 →8.8.8.8 ICMP 100 Echo (ping) request id=0x1ccb9, seq=141/36096, ttl=64
  691 376.897950363 8.8.8.8 →192.168.79.131 ICMP 100 Echo (ping) reply id=0x1ccb9, seq=141/36096, ttl=128 (request in 690)
  692 377.682654400 192.168.79.131 →208.67.220.220 ICMP 100 Echo (ping) request id=0x1cce, seq=122/31232, ttl=64
  693 377.710308666 208.67.220.220 →192.168.79.131 ICMP 100 Echo (ping) reply id=0x1cce, seq=122/31232, ttl=128 (request in 692)
  694 377.893226527 192.168.79.131 →8.8.8.8 ICMP 100 Echo (ping) request id=0x1ccb9, seq=142/36352, ttl=64
  695 377.899749701 8.8.8.8 →192.168.79.131 ICMP 100 Echo (ping) reply id=0x1ccb9, seq=142/36352, ttl=128 (request in 694)
  696 378.684449263 192.168.79.131 →208.67.220.220 ICMP 100 Echo (ping) request id=0x1cce, seq=123/31488, ttl=64
  697 378.712018604 208.67.220.220 →192.168.79.131 ICMP 100 Echo (ping) reply id=0x1cce, seq=123/31488, ttl=128 (request in 696)
  698 378.895112238 192.168.79.131 →8.8.8.8 ICMP 100 Echo (ping) request id=0x1ccb9, seq=143/36608, ttl=64
  699 378.901440928 8.8.8.8 →192.168.79.131 ICMP 100 Echo (ping) reply id=0x1ccb9, seq=143/36608, ttl=128 (request in 698)
  700 379.686898425 192.168.79.131 →208.67.220.220 ICMP 100 Echo (ping) request id=0x1cce, seq=124/31744, ttl=64
  701 379.714236250 208.67.220.220 →192.168.79.131 ICMP 100 Echo (ping) reply id=0x1cce, seq=124/31744, ttl=128 (request in 700)
  702 379.896594880 192.168.79.131 →8.8.8.8 ICMP 100 Echo (ping) request id=0x1ccb9, seq=144/36864, ttl=64
  703 379.902893921 8.8.8.8 →192.168.79.131 ICMP 100 Echo (ping) reply id=0x1ccb9, seq=144/36864, ttl=128 (request in 702)
  704 380.688974960 192.168.79.131 →208.67.220.220 ICMP 100 Echo (ping) request id=0x1cce, seq=125/32080, ttl=64
  705 380.720378561 208.67.220.220 →192.168.79.131 ICMP 100 Echo (ping) reply id=0x1cce, seq=125/32080, ttl=128 (request in 704)
  706 380.899058605 192.168.79.131 →8.8.8.8 ICMP 100 Echo (ping) request id=0x1ccb9, seq=145/37120, ttl=64
  707 380.905420255 8.8.8.8 →192.168.79.131 ICMP 100 Echo (ping) reply id=0x1ccb9, seq=145/37120, ttl=128 (request in 706)
  708 380.915771085 Vmware_d8:b4:e9 → ARP 44 Who has 192.168.79.2? Tell 192.168.79.131
  709 380.915933216 Vmware_e2:79:f1 → ARP 62 192.168.79.2 ls at 00:50:56:e2:79:f1
  710 381.691192079 192.168.79.131 →208.67.220.220 ICMP 100 Echo (ping) request id=0x1cce, seq=126/32256, ttl=64
  711 381.718531271 208.67.220.220 →192.168.79.131 ICMP 100 Echo (ping) reply id=0x1cce, seq=126/32256, ttl=128 (request in 710)
  712 381.900735720 192.168.79.131 →8.8.8.8 ICMP 100 Echo (ping) request id=0x1ccb9, seq=146/37376, ttl=64
  713 381.907118809 8.8.8.8 →192.168.79.131 ICMP 100 Echo (ping) reply id=0x1ccb9, seq=146/37376, ttl=128 (request in 712)
  714 382.269934793 Fe80::7e69:36eb:b6:dd9 →ff02::fb MDNS 109 Standard query 0x0000 PTR _ipp._tcp.local, "Q" question PTR _ipp._tcp.local, "Q" question
  715 382.693406779 192.168.79.131 →208.67.220.220 ICMP 100 Echo (ping) request id=0x1cce, seq=127/32512, ttl=64
  716 382.721017982 208.67.220.220 →192.168.79.131 ICMP 100 Echo (ping) reply id=0x1cce, seq=127/32512, ttl=128 (request in 715)
  717 382.902709609 192.168.79.131 →8.8.8.8 ICMP 100 Echo (ping) request id=0x1ccb9, seq=147/37632, ttl=64
  718 382.909356788 8.8.8.8 →192.168.79.131 ICMP 100 Echo (ping) reply id=0x1ccb9, seq=147/37632, ttl=128 (request in 717)
  719 383.507529963 192.168.79.131 →224.0.0.251 MDNS 89 Standard query 0x0000 PTR _ipp._tcp.local, "Q" question PTR _ipp._tcp.local, "Q" question
  720 383.695339583 192.168.79.131 →208.67.220.220 ICMP 100 Echo (ping) request id=0x1cce, seq=128/32768, ttl=64
  721 383.723219765 208.67.220.220 →192.168.79.131 ICMP 100 Echo (ping) reply id=0x1cce, seq=128/32768, ttl=128 (request in 720)
  722 383.905075310 192.168.79.131 →8.8.8.8 ICMP 100 Echo (ping) request id=0x1ccb9, seq=148/37888, ttl=64
  723 383.911533552 8.8.8.8 →192.168.79.131 ICMP 100 Echo (ping) reply id=0x1ccb9, seq=148/37888, ttl=128 (request in 722)
  724 384.697832528 192.168.79.131 →208.67.220.220 ICMP 100 Echo (ping) request id=0x1cce, seq=129/33024, ttl=64
  725 384.726885434 208.67.220.220 →192.168.79.131 ICMP 100 Echo (ping) reply id=0x1cce, seq=129/33024, ttl=128 (request in 724)
  726 384.907711176 192.168.79.131 →8.8.8.8 ICMP 100 Echo (ping) request id=0x1ccb9, seq=149/38144, ttl=64
  727 384.914076775 8.8.8.8 →192.168.79.131 ICMP 100 Echo (ping) reply id=0x1ccb9, seq=149/38144, ttl=128 (request in 726)
  728 385.699760879 192.168.79.131 →208.67.220.220 ICMP 100 Echo (ping) request id=0x1cce, seq=130/33280, ttl=64
  729 385.734677956 208.67.220.220 →192.168.79.131 ICMP 100 Echo (ping) reply id=0x1cce, seq=130/33280, ttl=128 (request in 728)
  730 385.916654298 192.168.79.131 →8.8.8.8 ICMP 100 Echo (ping) request id=0x1ccb9, seq=150/38400, ttl=64
  731 385.917074767 8.8.8.8 →192.168.79.131 ICMP 100 Echo (ping) reply id=0x1ccb9, seq=150/38400, ttl=128 (request in 730)
```

## 2. tcpstat

下載 tcpstat

```
apt install tcpstat

root@pinu-virtual-machine:/# apt install tcpstat
正在讀取套件清單... 完成
正在重建相依關係
正在讀取狀態資料... 完成
下列【新】套件將會被安裝：
  tcpstat
升級 0 個，新安裝 1 個，移除 0 個，有 113 個未被升級。
需要下載 33.1 kB 的套件檔。
此操作完成之後，會多佔用 91.1 kB 的磁碟空間。
下載:1 http://tw.archive.ubuntu.com/ubuntu bionic/universe amd64 tcpstat amd64
.5-8build1 [33.1 kB]
取得 33.1 kB 用了 0秒 (106 kB/s)
選取了原先未選的套件 tcpstat。
(讀取資料庫 ... 目前共安裝了 170475 個檔案和目錄。)
準備解開 .../tcpstat_1.5-8build1_amd64.deb ...
解開 tcpstat (1.5-8build1) 中...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
設定 tcpstat (1.5-8build1) ...
root@pinu-virtual-machine:/#
```

tcpstat 參數

### **tcpstat(1) - Linux man page**

#### **Name**

**tcpstat** - report network interface statistics

#### **Synopsis**

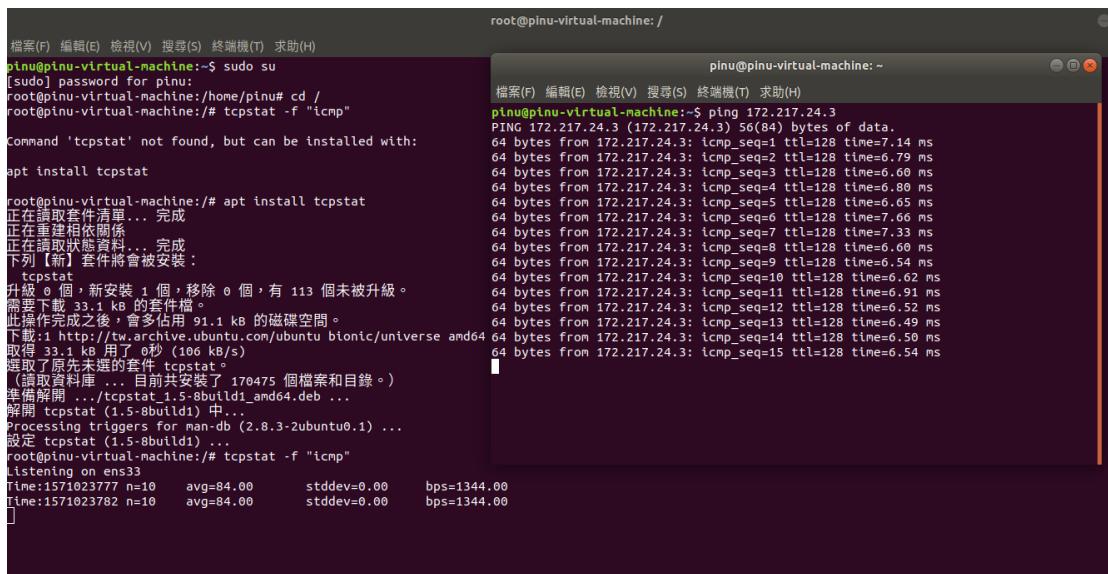
```
tcpstat [-?haeFlp] [-B bps] [-b bps] [-f filter expr] [-i interface] [-o output] [-R seconds] [-r filename] [-s seconds] [interval]
```

a. 抓取 “icmp” 的封包

```
root@pinu-virtual-machine:/# tcpstat -f "icmp"
Listening on ens33
```

b.ping google.com.tw

```
C:\Users\品瑜>nslookup  
預設伺服器: dns.nsysu.edu.tw  
Address: 140.117.11.1  
  
> www.hp.com  
伺服器: dns.nsysu.edu.tw  
Address: 140.117.11.1  
  
未經授權的回答:  
名稱: www-hpcom.g1b1.hp.com  
Addresses: 15.73.200.26  
          15.72.238.28  
Aliases: www.hp.com
```



The screenshot shows two terminal windows side-by-side. The left window is a root shell on a virtual machine named 'pinu'. It displays the output of several commands: 'sudo su' to become root, 'tcpstat -f "icmp"' which shows the command was not found but can be installed via apt, 'apt install tcpstat' which installs the package, and 'tcpstat' which shows system statistics for ICMP traffic. The right window is also a root shell on the same machine, showing the output of 'ping 172.217.24.3' which sends 56(84) bytes of data and receives multiple replies from the target IP address.

```
檔案(F) 編輯(E) 檢視(V) 搜尋(S) 終端機(T) 求助(H)  
pinu@pinu-virtual-machine:~$ sudo su  
[sudo] password for pinu:  
root@pinu-virtual-machine:/home/pinu# cd /  
root@pinu-virtual-machine:# tcpstat -f "icmp"  
Command 'tcpstat' not found, but can be installed with:  
apt install tcpstat  
root@pinu-virtual-machine:# apt install tcpstat  
正在讀取套件清單... 完成  
正在重建相依關係  
正在讀取狀態資料... 完成  
下列【新】套件將會被安裝：  
tcpstat  
升級 0 個，新安裝 1 個，移除 0 個，有 113 個未被升級。  
需要下載 33.1 kB 的套件檔。  
此操作完成之後，會多佔用 91.1 kB 的磁碟空間。  
下載:1 http://tw.archive.ubuntu.com/ubuntu bionic/universe amd64  
取得: 33.1 kB 用了: 0秒 (106 kB/s)  
選取了原先未選的套件: tcpstat。  
(選取資料庫 ... 目前共安裝了 170475 個檔案和目錄。)  
準備解開 .../tcpstat_1.5-8build1_amd64.deb ...  
解開 tcpstat_1.5-8build1 中。  
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...  
設定 tcpstat (1.5-8build1) ...  
root@pinu-virtual-machine:# tcpstat -f "icmp"  
Listening on ens33  
Time:1571023777 n=10      avg=84.00      stddev=0.00      bps=1344.00  
Time:1571023782 n=10      avg=84.00      stddev=0.00      bps=1344.00  
[]  
  
檔案(F) 編輯(E) 檢視(V) 搜尋(S) 終端機(T) 求助(H)  
pinu@pinu-virtual-machine:~$ ping 172.217.24.3  
PING 172.217.24.3 (172.217.24.3) 56(84) bytes of data.  
64 bytes from 172.217.24.3: icmp_seq=1 ttl=128 time=7.14 ms  
64 bytes from 172.217.24.3: icmp_seq=2 ttl=128 time=6.79 ms  
64 bytes from 172.217.24.3: icmp_seq=3 ttl=128 time=6.60 ms  
64 bytes from 172.217.24.3: icmp_seq=4 ttl=128 time=6.80 ms  
64 bytes from 172.217.24.3: icmp_seq=5 ttl=128 time=6.65 ms  
64 bytes from 172.217.24.3: icmp_seq=6 ttl=128 time=7.06 ms  
64 bytes from 172.217.24.3: icmp_seq=7 ttl=128 time=7.33 ms  
64 bytes from 172.217.24.3: icmp_seq=8 ttl=128 time=6.66 ms  
64 bytes from 172.217.24.3: icmp_seq=9 ttl=128 time=6.54 ms  
64 bytes from 172.217.24.3: icmp_seq=10 ttl=128 time=6.62 ms  
64 bytes from 172.217.24.3: icmp_seq=11 ttl=128 time=6.91 ms  
64 bytes from 172.217.24.3: icmp_seq=12 ttl=128 time=6.52 ms  
64 bytes from 172.217.24.3: icmp_seq=13 ttl=128 time=6.49 ms  
64 bytes from 172.217.24.3: icmp_seq=14 ttl=128 time=6.50 ms  
64 bytes from 172.217.24.3: icmp_seq=15 ttl=128 time=6.54 ms
```

Ctrl + Z 中斷 tcpstat

```
root@pinu-virtual-machine:/# tcpstat -f "icmp"
Listening on ens33
Time:1571023777 n=10    avg=84.00      stddev=0.00    bps=1344.00
Time:1571023782 n=10    avg=84.00      stddev=0.00    bps=1344.00
Time:1571023787 n=10    avg=84.00      stddev=0.00    bps=1344.00
Time:1571023792 n=10    avg=84.00      stddev=0.00    bps=1344.00
Time:1571023797 n=10    avg=84.00      stddev=0.00    bps=1344.00
Time:1571023802 n=10    avg=84.00      stddev=0.00    bps=1344.00
Time:1571023807 n=10    avg=84.00      stddev=0.00    bps=1344.00
Time:1571023812 n=10    avg=84.00      stddev=0.00    bps=1344.00
Time:1571023817 n=10    avg=84.00      stddev=0.00    bps=1344.00
Time:1571023822 n=10    avg=84.00      stddev=0.00    bps=1344.00
Time:1571023827 n=10    avg=84.00      stddev=0.00    bps=1344.00
Time:1571023832 n=10    avg=84.00      stddev=0.00    bps=1344.00
Time:1571023837 n=10    avg=84.00      stddev=0.00    bps=1344.00
Time:1571023842 n=10    avg=84.00      stddev=0.00    bps=1344.00
Time:1571023847 n=10    avg=84.00      stddev=0.00    bps=1344.00
Time:1571023852 n=10    avg=84.00      stddev=0.00    bps=1344.00
^C
^C^C^Z
[1]+  已停止                  tcpstat -f "icmp"
```

### 3. tcpdump & tcpstat & gnuplot

查看網卡 :ens33

```
root@pinu-virtual-machine:/# ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 192.168.79.131 netmask 255.255.255.0 broadcast 192.168.79.255
        inet6 fe80::7969:36eb:bc6f:dde9 prefixlen 64 scopeid 0x20<link>
          ether 00:0c:29:d8:b4:e9 txqueuelen 1000 (Ethernet)
            RX packets 1008 bytes 507260 (507.2 KB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 478 bytes 56864 (56.8 KB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
        inet 127.0.0.1 netmask 255.0.0.0
        inet6 ::1 prefixlen 128 scopeid 0x10<host>
          loop txqueuelen 1000 (Local Loopback)
            RX packets 179 bytes 14464 (14.4 KB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 179 bytes 14464 (14.4 KB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

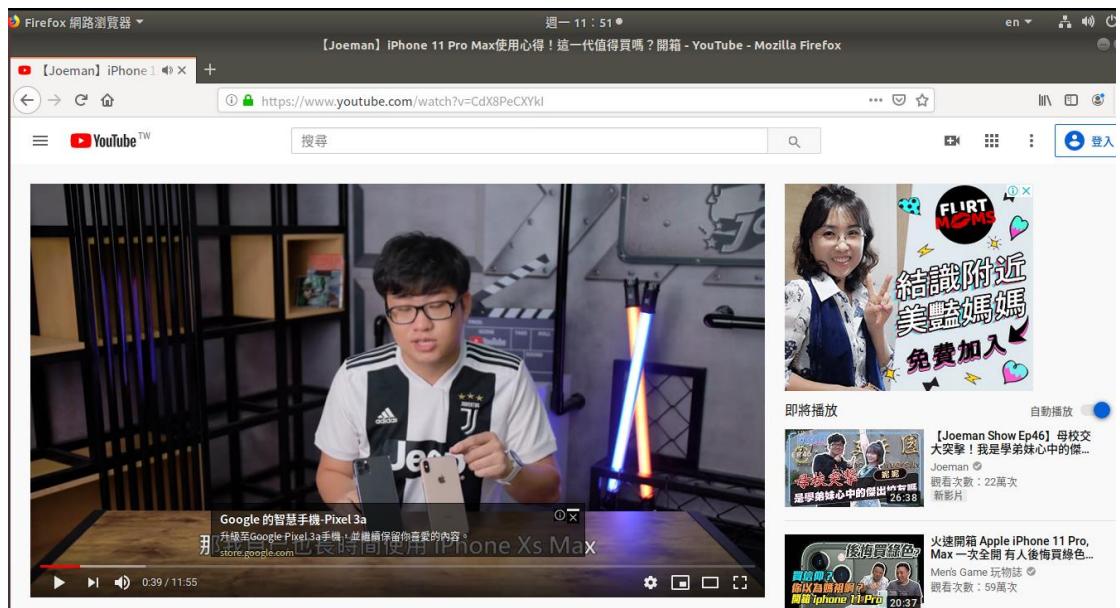
a. tcpdump -l ens33 -w rawdata.dmp

```
root@pinu-virtual-machine:/# sudo tcpdump -i ens33 -w rawdata.dmp -Z root
dropped privs to root
tcpdump: rawdata.dmp: Permission denied
root@pinu-virtual-machine:/# cd /tmp
root@pinu-virtual-machine:/tmp# sudo tcpdump -i ens33 -w rawdata.dmp -Z root
dropped privs to root
tcpdump: listening on ens33, link-type EN10MB (Ethernet), capture size 262144 bytes
```

如果在根目錄執行會有 tcpdump 自動降權限的問題  
所以切到 /tmp (writable directory)

b. 開啟瀏覽器瀏覽網頁一分鐘





c. 中斷 tcpdump

```
root@pinu-virtual-machine:/tmp# sudo tcpdump -i ens33 -w rawdata.dmp -Z root
dropped privs to root
tcpdump: listening on ens33, link-type EN10MB (Ethernet), capture size 262144 bytes
^C13425 packets captured
13425 packets received by filter
0 packets dropped by kernel
```

d. tcpstat 將擷取的檔案格式化

```
root@pinu-virtual-machine:/tmp# tcpstat -r rawdata.dmp -o "%r %A %T %U %I %b\n" > tcpstat.log
```

vim 寫一個 script

```
root@pinu-virtual-machine:/tmp# vi script1
```

```
set terminal png
set style data lines
set xlabel "Time (seconds)"
set ylabel "packets/s"
plot [00:60] "tcpstat.log" using 1:2 title "ARP", \
            "tcpstat.log" using 1:3 title "TCP", \
            "tcpstat.log" using 1:4 title "UDP"
```

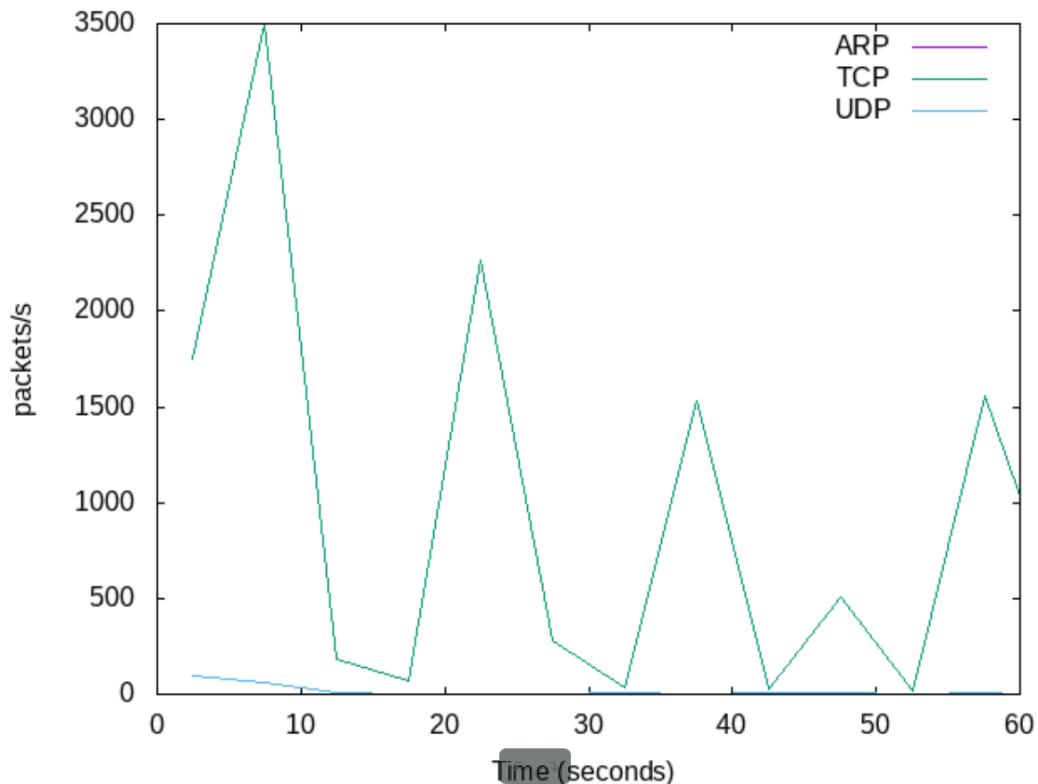
```
apt install gnuplot-nox
apt install gnuplot-qt
apt install gnuplot-x11
```

e. gnuplot 繪圖

```
root@pinu-virtual-machine:/tmp# gnuplot script1 > graph1.png
```

開啟圖檔

```
root@pinu-virtual-machine:/tmp# eog graph1.png
```



#### 4. mininet & iperf & gnuplot

4-1 iperf 的用途？什麼情況下會使用它？

iperf 是一個網路頻寬測試指令工具，支援 IPv4 與 IPv6 網路位址與 TCP、UDP、SCTP 傳輸協定，可在 Windows ,Mac OS X,Linux,FreeBSD 與手機等各種平台使用。

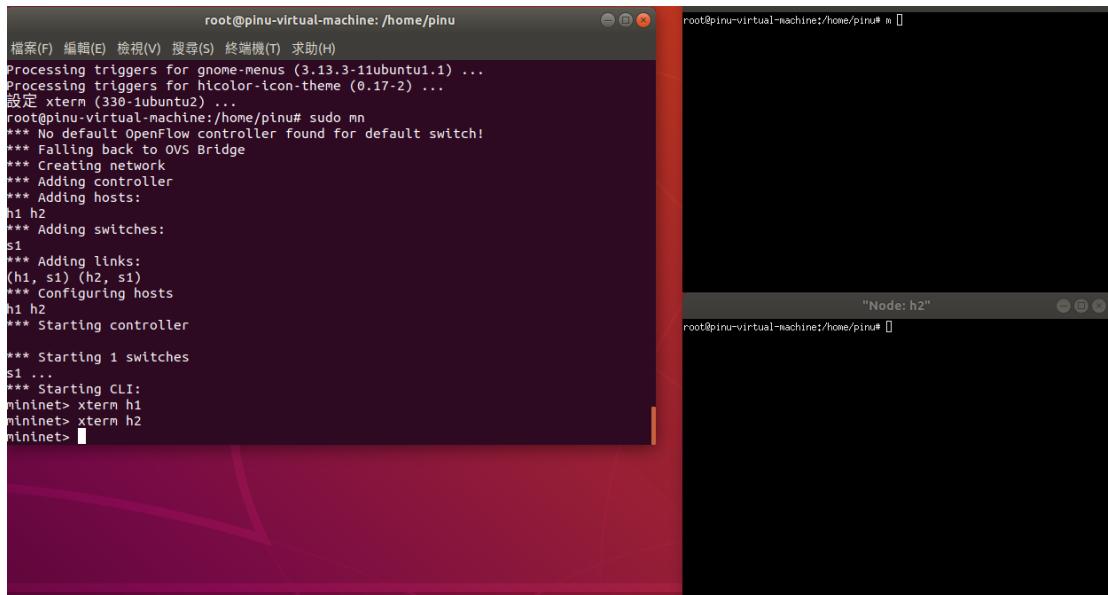
TCP/IP 和 UDP/IP 的性能測量工具，提供網路吞吐率信息，以及震動,丟包率，最大段和最大傳輸單元大小等統計信息;從而能夠幫助測試網路性能，定位網路瓶頸。

版本 18.04 需安裝 xterm (否則 mininet 模式下無法運作)

```
root@pinu-virtual-machine:/home/pinu# sudo apt install xterm
正在讀取套件清單... 完成
正在重建相依關係
正在讀取狀態資料... 完成
下列的額外套件將被安裝：
  libutempter0
建議套件：
  xfonts-cyrillic
下列【新】套件將會被安裝：
  libutempter0 xterm
升級 0 個，新安裝 2 個，移除 0 個，有 113 個未被升級。
需要下載 669 kB 的套件檔。
此操作完成之後，會多佔用 2,023 kB 的磁碟空間。
是否繼續進行 [Y/n] ? [Y/n] Y
下載:1 http://tw.archive.ubuntu.com/ubuntu bionic/main amd64 libutempter0 amd64
1.1.6-3 [7,898 B]
下載:2 http://tw.archive.ubuntu.com/ubuntu bionic/universe amd64 xterm amd64 330
-1ubuntu2 [661 kB]
取得 669 kB 用了 0秒 (1,967 kB/s)
選取了原先未選的套件 libutempter0:amd64。
(讀取資料庫 ... 目前共安裝了 171313 個檔案和目錄。)
準備解開 .../libutempter0_1.1.6-3_amd64.deb ...
```

假設 h1 為 server , h2 為 client

預設情況下 hosts 之間有 1 個節點



先在 root 模式下 安裝 iperf3 , xterm h1 / xterm h2 要測 TCP/UDP 頻寬時才能測

```
root@pinu-virtual-machine:/home/pinu# apt install iperf3
正在讀取套件清單... 完成
正在重建相依關係
正在讀取狀態資料... 完成
下列的額外套件將被安裝：
  libiperf0
下列【新】套件將會被安裝：
  iperf3 libiperf0
升級 0 個，新安裝 2 個，移除 0 個，有 113 個未被升級。
需要下載 64.0 kB 的套件檔。
此操作完成之後，會多佔用 205 kB 的磁碟空間。
是否繼續進行 [Y/n] ? [Y/n] Y
下載:1 http://tw.archive.ubuntu.com/ubuntu bionic/universe amd64 libiperf0 amd64
  3.1.3-1 [55.2 kB]
下載:2 http://tw.archive.ubuntu.com/ubuntu bionic/universe amd64 iperf3 amd64 3.
  1.3-1 [8,802 B]
取得 64.0 kB 用了 6秒 (136 kB/s)
選取了原先未選的套件 libiperf0:amd64。
(讀取資料庫 ... 目前共安裝了 171364 個檔案和目錄。)
準備解開 .../libiperf0_3.1.3-1_amd64.deb ...
解開 libiperf0:amd64 (3.1.3-1) 中...
選取了原先未選的套件 iperf3。
準備解開 .../iperf3_3.1.3-1_amd64.deb ...
解開 iperf3 (3.1.3-1) 中...
```

Ifconfig 查看 h1 位址

```
root@pinu-virtual-machine:/home/pinu# ifconfig
h1-eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
      inet 10.0.0.1 netmask 255.0.0.0 broadcast 10.255.255.255
        inet6 fe80::286e:1dff:fe7d:8190 prefixlen 64 scopeid 0x20<link>
          ether 2a:6e:1d:7d:81:90 txqueuelen 1000 (Ethernet)
            RX packets 31 bytes 4100 (4.1 KB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 9 bytes 726 (726.0 B)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
      inet 127.0.0.1 netmask 255.0.0.0
        inet6 ::1 prefixlen 128 scopeid 0x10<host>
          loop txqueuelen 1000 (Local Loopback)
            RX packets 0 bytes 0 (0.0 B)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 0 bytes 0 (0.0 B)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

開啟 server (h1)

```
root@pinu-virtual-machine:/home/pinu# iperf3 -s
-----
Server listening on 5201
-----
```

## Client 端(h2) : iperf3 -c 10.0.0.1

The image shows three terminal windows on a Linux system. The top window is titled "Node: h1" and displays the output of an iperf3 server listening on port 5201. It shows a connection from node h2 at 10.0.0.2, port 53662. The bandwidth achieved is approximately 14.2 Gbit/sec. The middle window is titled "Node: h2" and shows the iperf3 client command being run: "root@pinu-virtual-machine:/home/pinu# iperf3 -c 10.0.0.1". The bottom window is also titled "Node: h2" and shows the client's perspective of the connection, indicating a bandwidth of 18.8 Gbit/sec.

```
root@pinu-virtual-machine:/home/pinu# iperf3 -c 10.0.0.1
[ 13] local 10.0.0.2 port 53662 connected to 10.0.0.1 port 5201
[ 13] 0.00-1.00  sec  1.65 GBytes 14.2 Gbit/sec
[ 13] 1.00-2.00  sec  2.40 GBytes 20.6 Gbit/sec
[ 13] 2.00-3.00  sec  2.38 GBytes 20.5 Gbit/sec
[ 13] 3.00-4.00  sec  2.45 GBytes 21.0 Gbit/sec
[ 13] 4.00-5.00  sec  2.39 GBytes 20.6 Gbit/sec
[ 13] 5.00-6.00  sec  2.37 GBytes 20.3 Gbit/sec
[ 13] 6.00-7.00  sec  2.20 GBytes 18.9 Gbit/sec
[ 13] 7.00-8.00  sec  2.20 GBytes 18.9 Gbit/sec
[ 13] 8.00-9.00  sec  1.83 GBytes 15.7 Gbit/sec
[ 13] 9.00-10.00 sec  2.06 GBytes 17.6 Gbit/sec
[ 13] 10.00-10.04 sec  67.9 MBytes 13.0 Gbit/sec
[ 13] Interval      Transfer     Bandwidth
[ 13] 0.00-10.04 sec  0.00 Bytes  0.00 bits/sec      sender
[ 13] 0.00-10.04 sec  22.0 GBytes 18.8 Gbit/sec      receiver
Server listening on 5201
[ 12] local 10.0.0.2 port 53664 connected to 10.0.0.1 port 5201
[ 12] 0.00-1.00  sec  1.75 GBytes 15.0 Gbit/sec  0  8.27 MBbytes
[ 12] 1.00-2.00  sec  2.39 GBytes 20.5 Gbit/sec  0  8.27 MBbytes
[ 12] 2.00-3.00  sec  2.37 GBytes 20.4 Gbit/sec  0  8.27 MBbytes
[ 12] 3.00-4.00  sec  2.46 GBytes 21.1 Gbit/sec  0  8.27 MBbytes
[ 12] 4.00-5.00  sec  2.39 GBytes 20.6 Gbit/sec  0  8.27 MBbytes
[ 12] 5.00-6.00  sec  2.37 GBytes 20.3 Gbit/sec  0  8.27 MBbytes
[ 12] 6.00-7.00  sec  2.20 GBytes 18.9 Gbit/sec  0  8.27 MBbytes
[ 12] 7.00-8.00  sec  2.20 GBytes 18.9 Gbit/sec  0  8.27 MBbytes
[ 12] 8.00-9.00  sec  1.80 GBytes 15.4 Gbit/sec  0  8.27 MBbytes
[ 12] 9.00-10.00 sec  2.07 GBytes 17.8 Gbit/sec  0  8.27 MBbytes
[ 12] Interval      Transfer     Bandwidth      Retr
[ 12] 0.00-10.00 sec  22.0 GBytes 18.9 Gbit/sec  0      sender
[ 12] 0.00-10.00 sec  22.0 GBytes 18.9 Gbit/sec  0      receiver
iperf Done.
```

建立 Custom topology Code

## 建立 Custom topology 並測試

```
root@pinu-virtual-machine:/home/pinu# sudo mn --custom topo-3sw-2host.py --topo mytopo --test pingall
*** No default OpenFlow controller found for default switch!
*** Falling back to OVS Bridge
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1 s2 s3
*** Adding links:
(h1, s1) (s1, s2) (s2, s3) (s3, h2)
*** Configuring hosts
h1 h2
*** Starting controller

*** Starting 3 switches
s1 s2 s3 ...
*** Waiting for switches to connect
s1 s2 s3
*** Ping: testing ping reachability
h1 -> h2
h2 -> h1
*** Results: 0% dropped (2/2 received)
*** Stopping 0 controllers

*** Stopping 4 links
....
*** Stopping 3 switches
s1 s2 s3
*** Stopping 2 hosts
h1 h2
*** Done
completed in 1.026 seconds
```

## 開啟 Custom topology 的 mininet 環境

```
root@pinu-virtual-machine:/home/pinu# sudo mn --custom topo-3sw-2host.py --topo mytopo
*** No default OpenFlow controller found for default switch!
*** Falling back to OVS Bridge
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1 s2 s3
*** Adding links:
(h1, s1) (s1, s2) (s2, s3) (s3, h2)
*** Configuring hosts
h1 h2
*** Starting controller

*** Starting 3 switches
s1 s2 s3 ...
*** Starting CLI:
mininet> █
```

```
檔案(F) 編輯(E) 檢視(V) 搜尋(S) 終端機(T) 求助(H) root@pinu-virtual-machine:/home/pinu

*** Adding links:
(h1, s1) (s1, s2) (s2, s3) (s3, h2)
*** Configuring hosts
h1 h2
*** Starting controller

*** Starting 3 switches
s1 s2 s3 ...
*** Waiting for switches to connect
s1 s2 s3
*** Ping: testing ping reachability
h1 -> h2
h2 -> h1
*** Results: 0% dropped (2/2 received)
*** Stopping 0 controllers

*** Stopping 4 links
...
*** Stopping 3 switches
s1 s2 s3
*** Stopping 2 hosts
h1 h2
*** Done
completed in 1.026 seconds
root@pinu-virtual-machine:/home/pinu# sudo mn --custom topo-3sw-2host.py --topo mytopo
*** No default OpenFlow controller found for default switch!
*** Falling back to OVS Bridge
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1 s2 s3
*** Adding links:
(h1, s1) (s1, s2) (s2, s3) (s3, h2)
*** Configuring hosts
h1 h2
*** Starting controller

*** Starting 3 switches
s1 s2 s3 ...
*** Starting CLI:
mininet> xterm h1
mininet> xterm h2
mininet> 
```

"Node: h1"

```
root@pinu-virtual-machine:/home/pinu# 
```

"Node: h2"

```
root@pinu-virtual-machine:/home/pinu# 
```

## Server 端: Iperf -s

Client 端: iperf -c 10.0.0.1

```
*** Adding hosts:  
h1 h2  
*** Adding switches:  
s1  
*** Adding links:  
(h1, s1) (h2, s1)  
*** Configuring hosts  
h1 h2  
*** Starting controller  
  
*** Starting 1 switches  
s1 ...  
*** Starting CLI:  
mininet> exit  
*** Stopping 0 controllers  
  
*** Stopping 2 links  
...  
*** Stopping 1 switches  
s1  
*** Stopping 2 hosts  
h1 h2  
*** Done  
completed in 3.542 seconds  
root@pinu-virtual-machine:/home/pinu# sudo mn --custom topo-3sw-2host.py --topo mytopo  
*** No default OpenFlow controller found for default switch!  
*** Falling back to OVS Bridge  
*** Creating network  
*** Adding controller  
*** Adding hosts:  
h1 h2  
*** Adding switches:  
s1 s2 s3  
*** Adding links:  
(h1, s1) (s1, s2) (s2, s3) (s3, h2)  
*** Configuring hosts  
h1 h2  
*** Starting controller  
  
*** Starting 3 switches  
s1 s2 s3 ...  
*** Starting CLI:  
mininet> xterm h1  
mininet> xterm h2  
mininet> [ ]  
  
root@pinu-virtual-machine:/home/pinu# ping -c 10 0.0.0.1  
Connecting to host 10.0.0.1 port 5201  
[ 16] local 10.0.0.2 port 53706 connected to 10.0.0.1 port 5201  
[ ID] Interval Transfer Bandwidth Retr Cwnd  
[ 16] 0.00-1.03 sec 451.0BBytes 3.69 Gbytes/sec 0 2.99 Mbytes  
[ 16] 1.03-2.06 sec 399.0BBytes 3.38 Gbytes/sec 0 2.99 Mbytes  
[ 16] 2.03-3.09 sec 399.0BBytes 3.17 Gbytes/sec 0 2.94 Mbytes  
[ 16] 3.09-4.08 sec 1.90 Gbytes 17.3 Gbytes/sec 0 8.38 Mbytes  
[ 16] 4.00-5.00 sec 1.62 Gbytes 15.3 Gbytes/sec 0 8.38 Mbytes  
[ 16] 5.00-6.00 sec 1.62 Gbytes 15.3 Gbytes/sec 0 8.38 Mbytes  
[ 16] 6.00-7.00 sec 1.37 Gbytes 16.1 Gbytes/sec 0 8.38 Mbytes  
[ 16] 7.00-8.00 sec 2.12 Gbytes 18.2 Gbytes/sec 0 8.38 Mbytes  
[ 16] 8.00-9.00 sec 2.24 Gbytes 19.2 Gbytes/sec 0 8.38 Mbytes  
[ 16] 9.00-10.00 sec 2.22 Gbytes 19.0 Gbytes/sec 0 8.38 Mbytes  
  
[ ID] Interval Transfer Bandwidth Retr sender  
[ 16] 0.00-10.00 sec 15.1 Gbytes 13.0 Gbytes/sec 0 sender  
[ 16] 0.00-10.00 sec 15.1 Gbytes 13.0 Gbytes/sec 0 receiver  
  
iperf Done.  
root@pinu-virtual-machine:/home/pinu# [ ]  
  
"Node: h2"  
[ 17] local 10.0.0.1 port 5201 connected to 10.0.0.2 port 53704  
[ ID] Interval Transfer Bandwidth Retr Cwnd  
[ 17] 0.00-0.06 sec 445.0BBytes 3.49 Gbytes/sec  
[ 17] 1.06-2.02 sec 415.0BBytes 3.65 Gbytes/sec  
[ 17] 2.02-3.06 sec 399.0BBytes 3.20 Gbytes/sec  
[ 17] 3.06-4.00 sec 399.0BBytes 3.03 Gbytes/sec  
[ 17] 4.00-5.00 sec 1.61 Gbytes 13.8 Gbytes/sec  
[ 17] 5.00-6.00 sec 1.94 Gbytes 16.6 Gbytes/sec  
[ 17] 6.00-7.00 sec 1.88 Gbytes 15.0 Gbytes/sec  
[ 17] 7.00-8.00 sec 2.23 Gbytes 19.2 Gbytes/sec  
[ 17] 8.00-9.00 sec 2.23 Gbytes 19.2 Gbytes/sec  
[ 17] 9.00-10.00 sec 2.22 Gbytes 19.1 Gbytes/sec  
[ 17] 10.00-10.04 sec 78.9 Mbytes 16.1 Gbytes/sec  
  
[ ID] Interval Transfer Bandwidth Retr sender  
[ 17] 0.00-10.04 sec 0.00 Bytes 0.00 bits/sec  
[ 17] 0.00-10.04 sec 15.1 Gbytes 12.9 Gbytes/sec  
sender receiver  
  
Server listening on 5201  
[ ]  
  
"Node: h1"  
[ 17] local 10.0.0.2 port 5201 connected to 10.0.0.1 port 53706  
[ ID] Interval Transfer Bandwidth Retr Cwnd  
[ 17] 0.00-0.06 sec 445.0BBytes 3.49 Gbytes/sec  
[ 17] 1.06-2.02 sec 415.0BBytes 3.65 Gbytes/sec  
[ 17] 2.02-3.06 sec 399.0BBytes 3.20 Gbytes/sec  
[ 17] 3.06-4.00 sec 399.0BBytes 3.03 Gbytes/sec  
[ 17] 4.00-5.00 sec 1.61 Gbytes 13.8 Gbytes/sec  
[ 17] 5.00-6.00 sec 1.94 Gbytes 16.6 Gbytes/sec  
[ 17] 6.00-7.00 sec 1.88 Gbytes 15.0 Gbytes/sec  
[ 17] 7.00-8.00 sec 2.23 Gbytes 19.2 Gbytes/sec  
[ 17] 8.00-9.00 sec 2.23 Gbytes 19.2 Gbytes/sec  
[ 17] 9.00-10.00 sec 2.22 Gbytes 19.1 Gbytes/sec  
[ 17] 10.00-10.04 sec 78.9 Mbytes 16.1 Gbytes/sec  
  
[ ID] Interval Transfer Bandwidth Retr sender  
[ 17] 0.00-10.04 sec 0.00 Bytes 0.00 bits/sec  
[ 17] 0.00-10.04 sec 15.1 Gbytes 12.9 Gbytes/sec  
sender receiver  
  
Server listening on 5201  
[ ]
```

5 個 switch , 2 個 host

```
檔案(F) 編輯(E) 檢視(V) 搜尋(S) 終端機(T) 求助(H)
from mininet.topo import Topo

class MyTopo( Topo ):
    "Simple topology example."

    def __init__( self ):
        "Create custom topo."

        # Initialize topology
        Topo.__init__( self )

        # Add hosts and switches
        leftHost = self.addHost( 'h1' )
        rightHost = self.addHost( 'h2' )
        Switch_1 = self.addSwitch( 's1' )
        Switch_2 = self.addSwitch( 's2' )
        Switch_3 = self.addSwitch('s3')
        Switch_4 = self.addSwitch('s4')
        Switch_5 = self.addSwitch('s5')

        # Add links
        self.addLink( leftHost, Switch_1 )
        self.addLink( Switch_1, Switch_2 )
        self.addLink( Switch_2, Switch_3 )
        self.addLink( Switch_3, Switch_4)
        self.addLink( Switch_4, Switch_5)
        self.addLink( Switch_5, rightHost)

topos = { 'mytopo': ( lambda: MyTopo() ) }
```

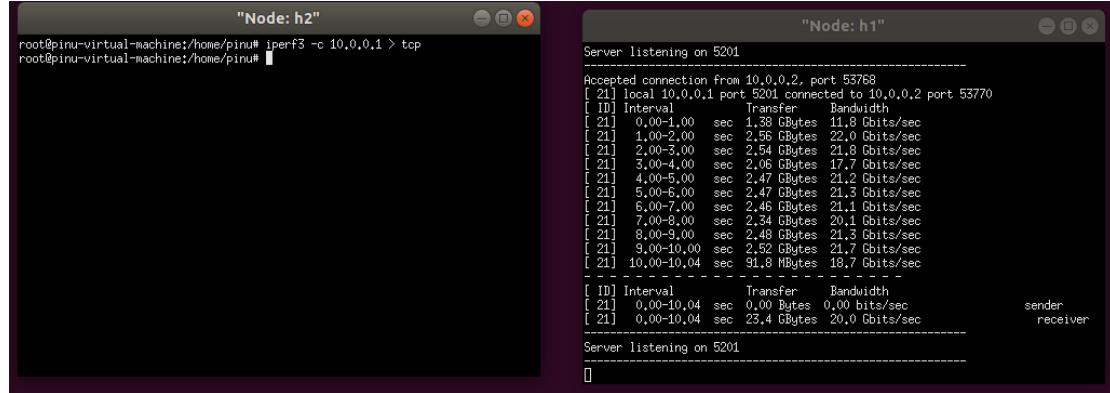
~  
~  
~  
~  
~  
~  
~  
~  
"topo-5sw-2host.py" 35L, 848C

```

root@pinu-virtual-machine:/home/pinu# sudo mn --custom topo-5sw-2host.py --topo mytopo
*** No default OpenFlow controller found for default switch!
*** Falling back to OVS Bridge
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1 s2 s3 s4 s5
*** Adding links:
(h1, s1) (s1, s2) (s2, s3) (s3, s4) (s4, s5) (s5, h2)
*** Configuring hosts
h1 h2
*** Starting controller

*** Starting 5 switches
s1 s2 s3 s4 s5 ...
*** Starting CLI:
mininet> 

```



## Cat tcp 查看結果

```

檔案(F) 編輯(E) 檢視(V) 搜尋(S) 終端機(T) 求助(H)
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1)
*** Configuring hosts
h1 h2
*** Starting controller
*** Starting 1 switches
s1 ...
*** Starting CLI:
mininet> exit
*** Stopping 0 controllers
*** Stopping 2 links
...
*** Stopping 1 switches
s1
*** Stopping 2 hosts
h1 h2
*** Done
completed in 2.684 seconds
root@pinu-virtual-machine:/home/pinu# sudo mn --custom topo-5sw-2host.py --topo mytopo
*** No default OpenFlow controller found for default switch!
*** Falling back to OVS Bridge
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1 s2 s3 s4 s5
*** Adding links:
(h1, s1) (s1, s2) (s2, s3) (s3, s4) (s4, s5) (s5, h2)
*** Configuring hosts
h1 h2
*** Starting controller

*** Starting 5 switches
s1 s2 s3 s4 s5 ...
*** Starting CLI:
mininet> xterm h1
mininet> xterm h2
mininet> 

```

抓取要的 Bandwidth 和 時間，使用 grep sec 去篩

```
root@pinu-virtual-machine:/home/pinu# cat top | grep sec
[ 20] 0.00-1.00 sec 1.46 GBytes 12.6 Gbits/sec 0 8.31 MBytes
[ 20] 1.00-2.00 sec 2.56 GBytes 22.0 Gbits/sec 0 8.31 MBytes
[ 20] 2.00-3.00 sec 2.54 GBytes 21.8 Gbits/sec 0 8.31 MBytes
[ 20] 3.00-4.00 sec 2.06 GBytes 17.7 Gbits/sec 0 8.31 MBytes
[ 20] 4.00-5.00 sec 2.48 GBytes 21.3 Gbits/sec 0 8.31 MBytes
[ 20] 5.00-6.00 sec 2.46 GBytes 21.2 Gbits/sec 0 8.31 MBytes
[ 20] 6.00-7.00 sec 2.48 GBytes 21.3 Gbits/sec 0 8.31 MBytes
[ 20] 7.00-8.00 sec 2.32 GBytes 19.9 Gbits/sec 0 8.31 MBytes
[ 20] 8.00-9.00 sec 2.49 GBytes 21.4 Gbits/sec 0 8.31 MBytes
[ 20] 9.00-10.00 sec 2.51 GBytes 21.6 Gbits/sec 0 8.31 MBytes
[ 20] 0.00-10.00 sec 23.4 GBytes 20.1 Gbits/sec 0 sender
[ 20] 0.00-10.00 sec 23.4 GBytes 20.1 Gbits/sec 0 receiver
```

抓取 前 10 筆

```
root@pinu-virtual-machine:/home/pinu# cat top | grep sec | head -n 10
[ 20] 0.00-1.00 sec 1.46 GBytes 12.6 Gbits/sec 0 8.31 MBytes
[ 20] 1.00-2.00 sec 2.56 GBytes 22.0 Gbits/sec 0 8.31 MBytes
[ 20] 2.00-3.00 sec 2.54 GBytes 21.8 Gbits/sec 0 8.31 MBytes
[ 20] 3.00-4.00 sec 2.06 GBytes 17.7 Gbits/sec 0 8.31 MBytes
[ 20] 4.00-5.00 sec 2.48 GBytes 21.3 Gbits/sec 0 8.31 MBytes
[ 20] 5.00-6.00 sec 2.46 GBytes 21.2 Gbits/sec 0 8.31 MBytes
[ 20] 6.00-7.00 sec 2.48 GBytes 21.3 Gbits/sec 0 8.31 MBytes
[ 20] 7.00-8.00 sec 2.32 GBytes 19.9 Gbits/sec 0 8.31 MBytes
[ 20] 8.00-9.00 sec 2.49 GBytes 21.4 Gbits/sec 0 8.31 MBytes
[ 20] 9.00-10.00 sec 2.51 GBytes 21.6 Gbits/sec 0 8.31 MBytes
```

用空格取代- : tr - " "

```
root@pinu-virtual-machine:/home/pinu# cat top | grep sec | head -n 10
[ 20] 0.00-1.00 sec 1.46 GBytes 12.6 Gbits/sec 0 8.31 MBytes
[ 20] 1.00-2.00 sec 2.56 GBytes 22.0 Gbits/sec 0 8.31 MBytes
[ 20] 2.00-3.00 sec 2.54 GBytes 21.8 Gbits/sec 0 8.31 MBytes
[ 20] 3.00-4.00 sec 2.06 GBytes 17.7 Gbits/sec 0 8.31 MBytes
[ 20] 4.00-5.00 sec 2.48 GBytes 21.3 Gbits/sec 0 8.31 MBytes
[ 20] 5.00-6.00 sec 2.46 GBytes 21.2 Gbits/sec 0 8.31 MBytes
[ 20] 6.00-7.00 sec 2.48 GBytes 21.3 Gbits/sec 0 8.31 MBytes
[ 20] 7.00-8.00 sec 2.32 GBytes 19.9 Gbits/sec 0 8.31 MBytes
[ 20] 8.00-9.00 sec 2.49 GBytes 21.4 Gbits/sec 0 8.31 MBytes
[ 20] 9.00-10.00 sec 2.51 GBytes 21.6 Gbits/sec 0 8.31 MBytes
root@pinu-virtual-machine:/home/pinu# cat top | grep sec | head -n 10 | tr - " "
[ 20] 0.00 1.00 sec 1.46 GBytes 12.6 Gbits/sec 0 8.31 MBytes
[ 20] 1.00 2.00 sec 2.56 GBytes 22.0 Gbits/sec 0 8.31 MBytes
[ 20] 2.00 3.00 sec 2.54 GBytes 21.8 Gbits/sec 0 8.31 MBytes
[ 20] 3.00 4.00 sec 2.06 GBytes 17.7 Gbits/sec 0 8.31 MBytes
[ 20] 4.00 5.00 sec 2.48 GBytes 21.3 Gbits/sec 0 8.31 MBytes
[ 20] 5.00 6.00 sec 2.46 GBytes 21.2 Gbits/sec 0 8.31 MBytes
[ 20] 6.00 7.00 sec 2.48 GBytes 21.3 Gbits/sec 0 8.31 MBytes
[ 20] 7.00 8.00 sec 2.32 GBytes 19.9 Gbits/sec 0 8.31 MBytes
[ 20] 8.00 9.00 sec 2.49 GBytes 21.4 Gbits/sec 0 8.31 MBytes
[ 20] 9.00 10.00 sec 2.51 GBytes 21.6 Gbits/sec 0 8.31 MBytes
```

取第 4 個 column 和 第 8 個 column

```
root@pinu-virtual-machine:/home/pinu# cat top | grep sec | head -n 10 | tr - " " | awk '{print $4,$8}'
1.00 12.6
2.00 22.0
3.00 21.8
4.00 17.7
5.00 21.3
6.00 21.2
7.00 21.3
8.00 19.9
9.00 21.4
10.00 21.6
```

依照上面的格式 匯出 tcp 檔案

```
root@pinu-virtual-machine:/home/pinu# cat tcp | grep sec | head -n 10 | tr - " " | awk '{print $4,$8}' > mytcp
```

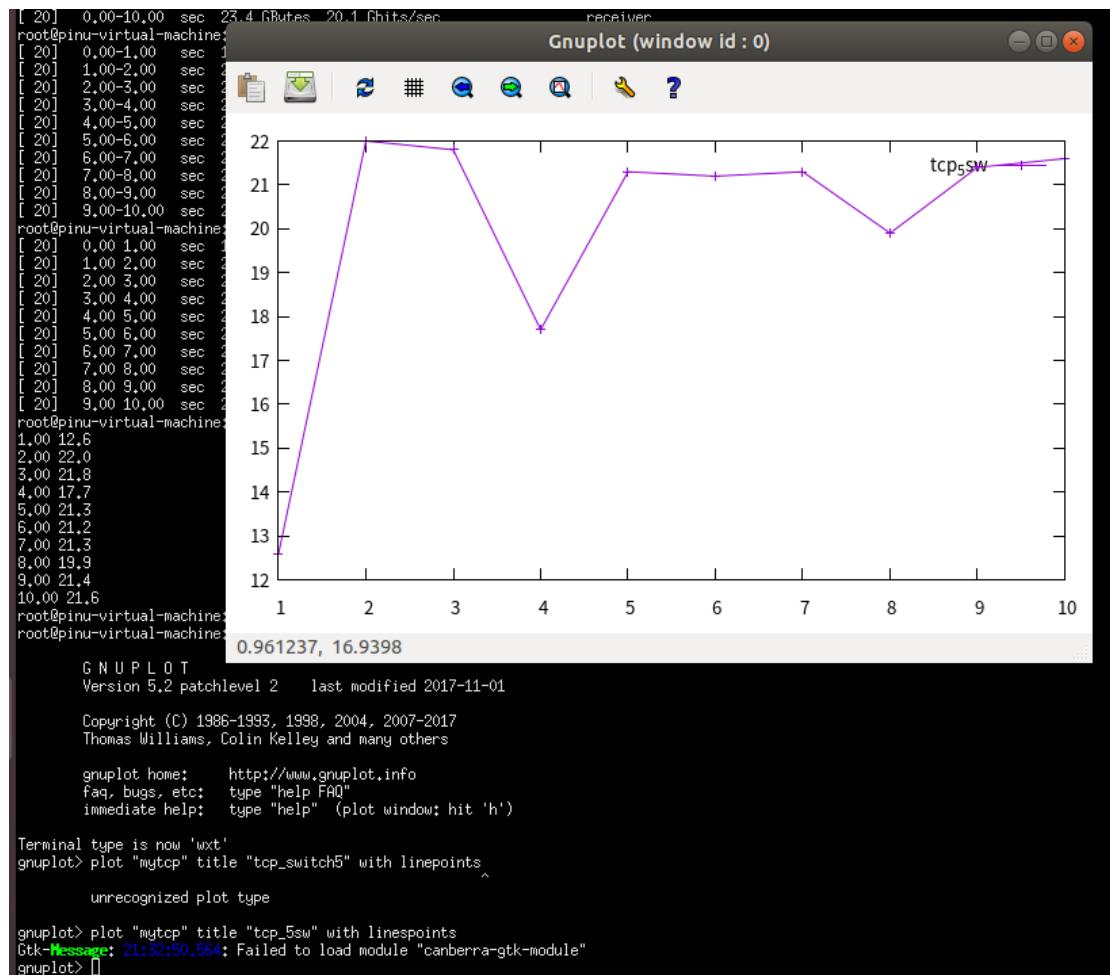
開啟 gnuplot 準備繪製 tcp 的圖

```
root@pinu-virtual-machine:/home/pinu# gnuplot
GNUPLOT
Version 5.2 patchlevel 2    last modified 2017-11-01
Copyright (C) 1986-1993, 1998, 2004, 2007-2017
Thomas Williams, Colin Kelley and many others

gnuplot home:      http://www.gnuplot.info
faq, bugs, etc:   type "help FAQ"
immediate help:   type "help" (plot window: hit 'h')

Terminal type is now 'wxt'
gnuplot>
```

繪製 2 個 host 5 個 switch 的圖



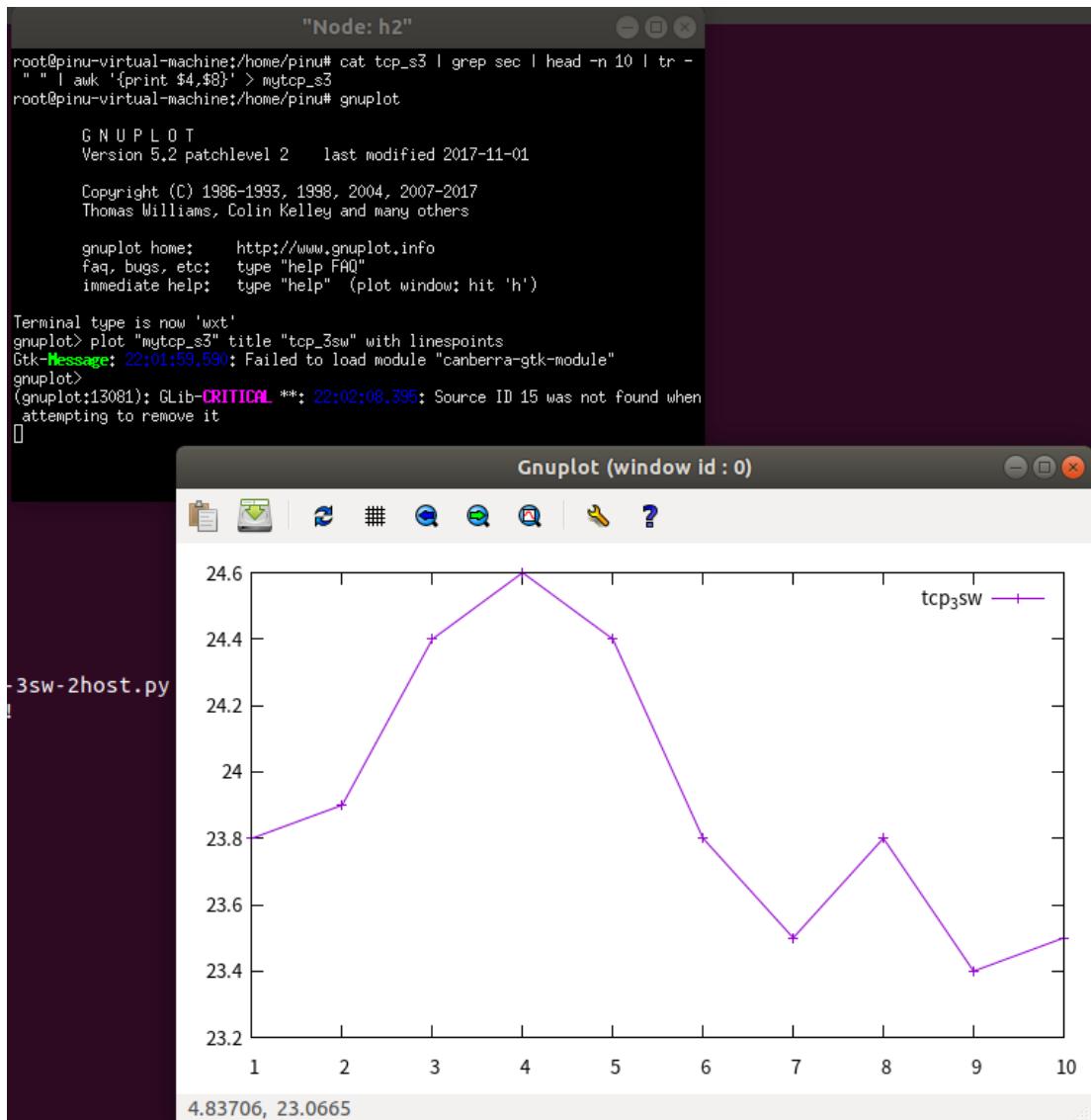
依題目要求 繪製多條 TCP (3 條) 於同一圖片中  
文字描述為 3 條 (1 個節點 + 3 條節點+5 條節點)  
但 sample 的圖 為 5 條 (TCP 0~4) ，所以我都會!

節點數為 3

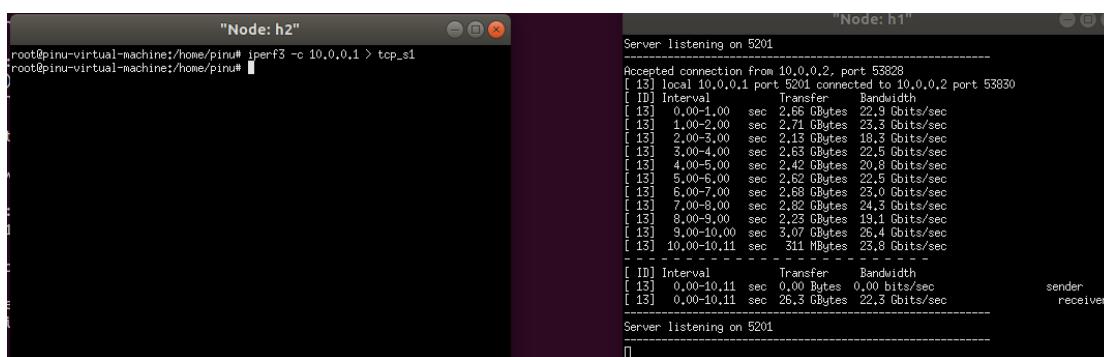
```
"Node: h2"
root@pinu-virtual-machine:/home/pinu# iperf -c 10.0.0.1 > tcp_s3
root@pinu-virtual-machine:/home/pinu# iperf3 -c 10.0.0.1 > tcp_s3
root@pinu-virtual-machine:/home/pinu# 

"Node: h1"
Server listening on 5201
-----
Accepted connection from 10.0.0.2, port 53812
[ 17] local 10.0.0.1 port 5201 connected to 10.0.0.2 port 53814
[ 17] Interval Transfer Bandwidth
[ 17] 0.00-1.00 sec 2.67 GBytes 22.3 Gbits/sec
[ 17] 1.00-2.00 sec 2.78 GBytes 23.9 Gbits/sec
[ 17] 2.00-3.00 sec 2.64 GBytes 24.4 Gbits/sec
[ 17] 3.00-4.00 sec 2.71 GBytes 24.6 Gbits/sec
[ 17] 4.00-5.00 sec 2.64 GBytes 24.0 Gbits/sec
[ 17] 5.00-6.00 sec 2.77 GBytes 23.8 Gbits/sec
[ 17] 6.00-7.00 sec 2.74 GBytes 23.5 Gbits/sec
[ 17] 7.00-8.00 sec 2.78 GBytes 23.8 Gbits/sec
[ 17] 8.00-9.00 sec 2.72 GBytes 23.4 Gbits/sec
[ 17] 9.00-10.00 sec 2.73 GBytes 23.5 Gbits/sec
[ 17] 10.00-10.04 sec 1.02 MBytes 20.3 Gbits/sec
[ 17] 0.00-10.04 sec 0.00 Bytes 0.00 bits/sec
[ 17] 0.00-10.04 sec 27.8 GBytes 23.8 Gbits/sec
----- sender receiver
Server listening on 5201
[]

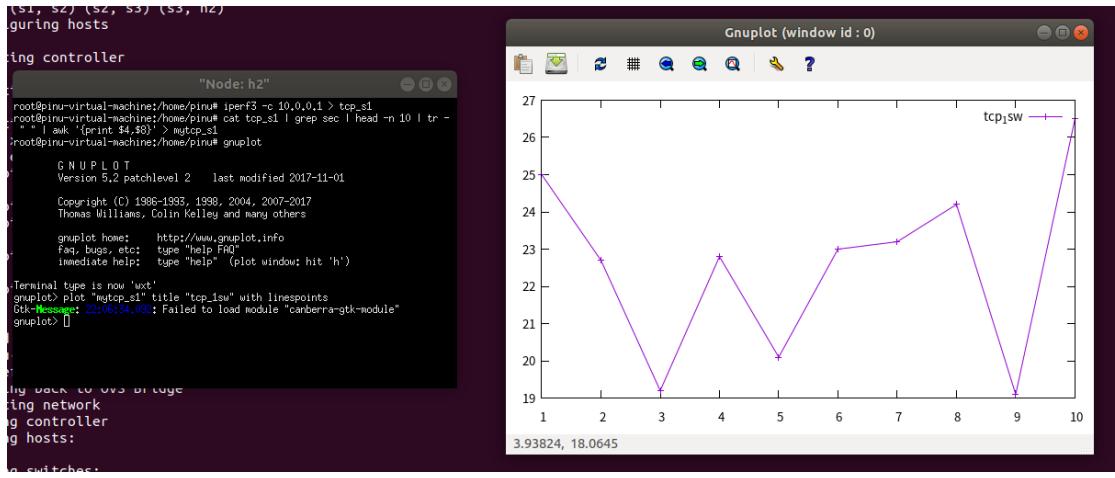
root@pinu-virtual-machine:/home/pinu# cat tcp_s3 | grep sec | head -n 10 | tr -
" " | awk '{print $4,$8}' > mytcp_s3
root@pinu-virtual-machine:/home/pinu# "
```



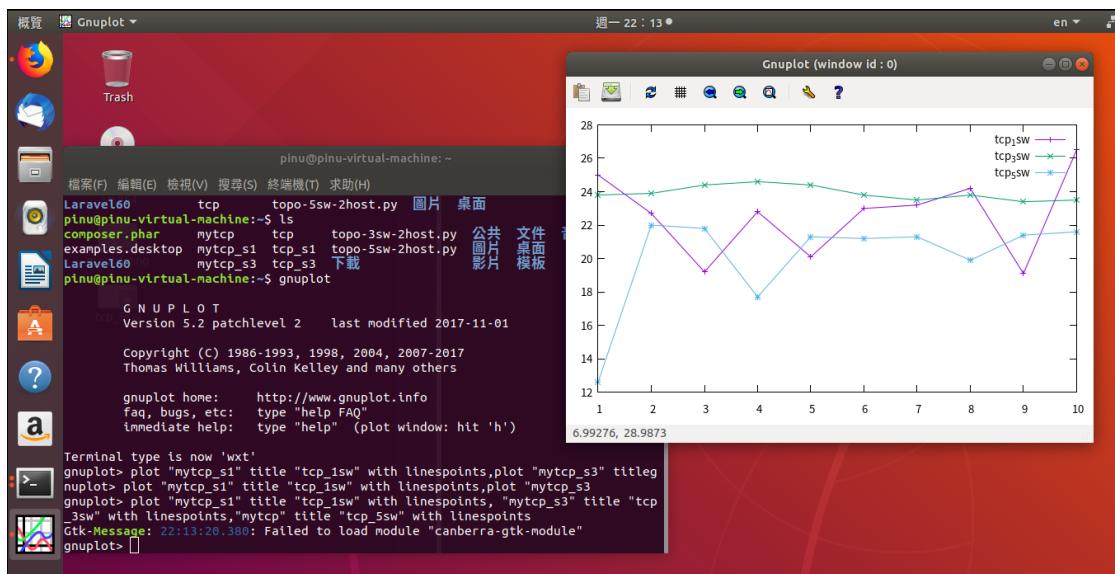
## 節點數為 1



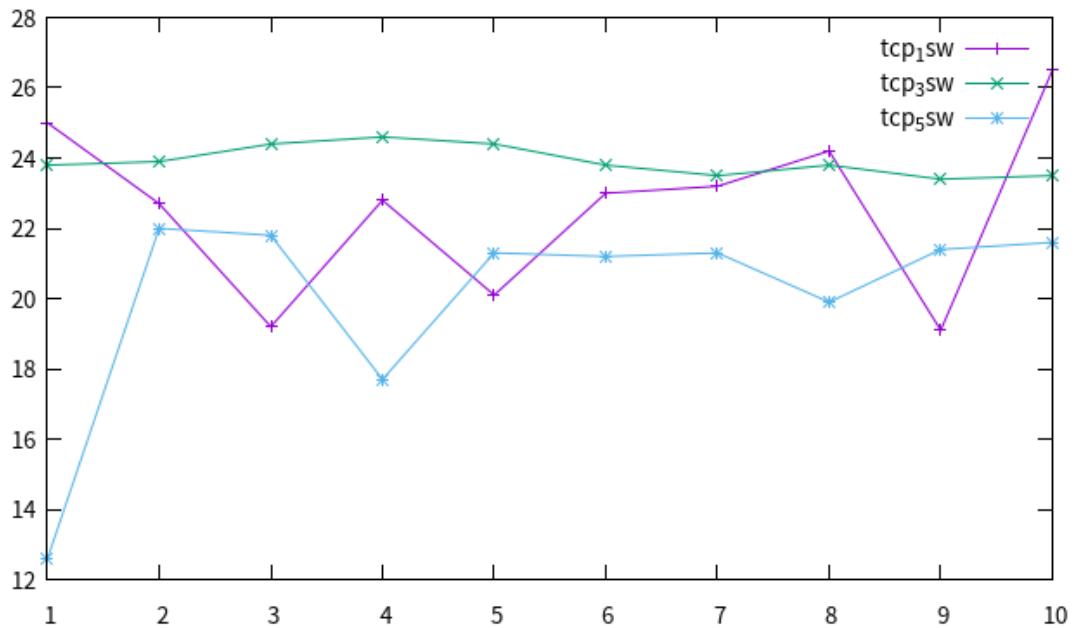
```
root@pinu-virtual-machine:/home/pinu# iperf3 -c 10.0.0.1 > tcp_s1
root@pinu-virtual-machine:/home/pinu# cat tcp_s1 | grep sec | head -n 10 | tr -
" " | awk '{print $4,$8}' > mytcp_s1
```



以 mytcp(五個 node) + mytcp\_s1(1 個節點) + mytcp\_s3 (3 個節點) 來畫圖



```
gnuplot> plot "mytcp_s1" title "tcp_1sw" with linespoints, "mytcp_s3" title "tcp_3sw" with linespoints, "mytcp" title "tcp_5sw" with linespoints
Gtk-Message: 22:13:20.380: Failed to load module "canberra-gtk-module"
gnuplot> 
```



以 1,2,3,4,5 個中間節點來畫圖

2 nodes

```
from mininet.topo import Topo

class MyTopo( Topo ):
    "Simple topology example."

    def __init__( self ):
        "Create custom topo."
        # Initialize topology
        Topo.__init__( self )

        # Add hosts and switches
        leftHost = self.addHost( 'h1' )
        rightHost = self.addHost( 'h2' )
        Switch_1 = self.addSwitch( 's1' )
        Switch_2 = self.addSwitch( 's2' )

        # Add links
        self.addLink( leftHost, Switch_1 )
        self.addLink( Switch_1, Switch_2 )
        self.addLink( Switch_2, rightHost )■

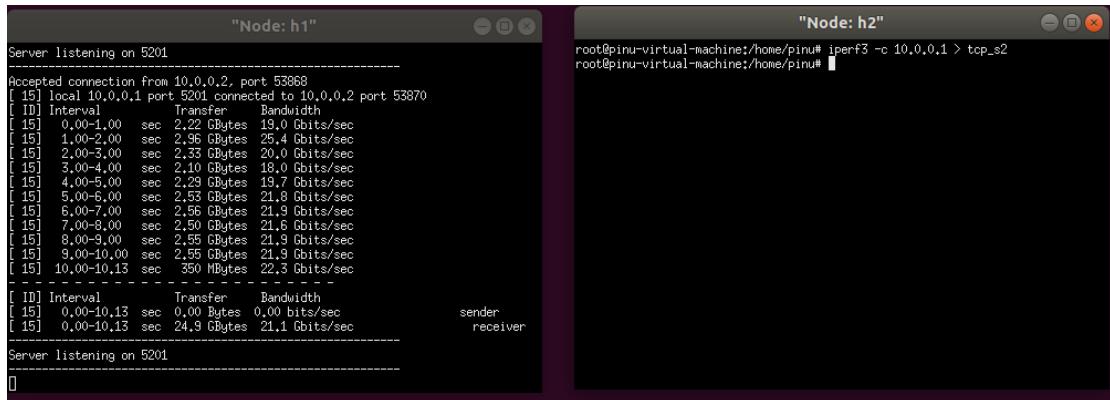
topos = { 'mytopo': ( lambda: MyTopo() ) }
```

```

root@pinu-virtual-machine:/home/pinu# sudo mn --custom topo-2sw-2host.py --topo mytopo
*** No default OpenFlow controller found for default switch!
*** Falling back to OVS Bridge
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1 s2
*** Adding links:
(h1, s1) (s1, s2) (s2, h2)
*** Configuring hosts
h1 h2
*** Starting controller

*** Starting 2 switches
s1 s2 ...
*** Starting CLI:
mininet> 

```



```

root@pinu-virtual-machine:/home/pinu# cat tcp_s2 | grep sec | head -n 10 | tr -
" " | awk '{print $4,$8}' > mytcp_s2

```

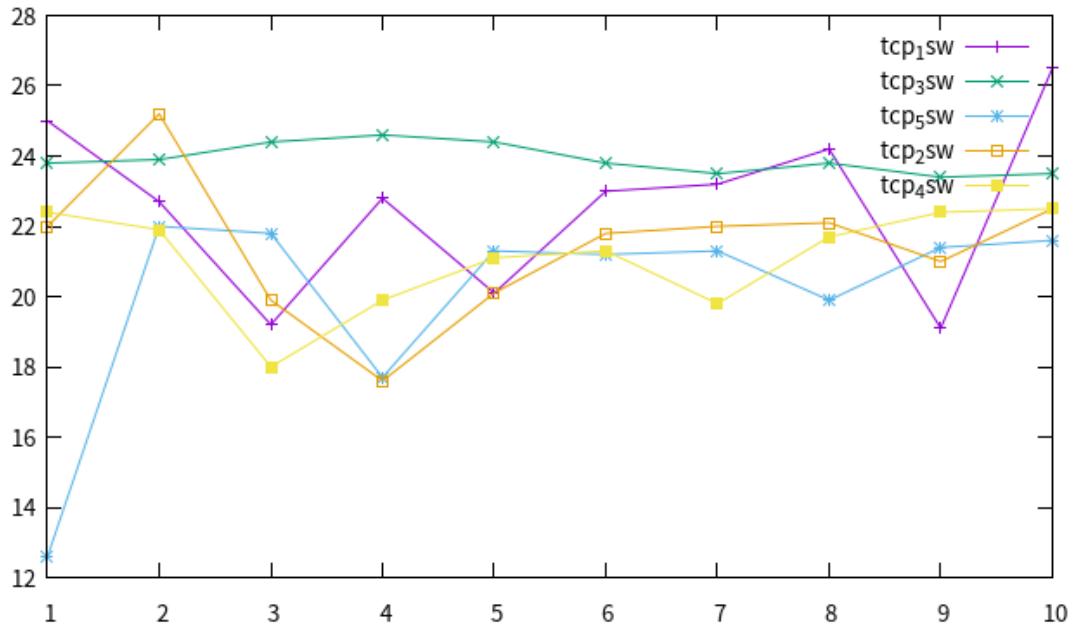
## 4 nodes

```

"Node: h2"
root@pinu-virtual-machine:/home/pinu# iperf3 -c 10.0.0.1 > tcp_s4
root@pinu-virtual-machine:/home/pinu# cat tcp_s4 | grep sec | head -n 10 | tr -
" " | awk '{print $4,$8}' > mytcp_s4
root@pinu-virtual-machine:/home/pinu# 

"Node: h1"
Server listening on 5201
-----
Accepted connection from 10.0.0.2, port 53906
[ 13] local 10.0.0.1 port 5201 connected to 10.0.0.2 port 53906
[ ID] Interval Transfer Bandwidth
[ 13] 0.00-1.00 sec 2.52 GBytes 21.7 Gbits/sec
[ 13] 1.00-2.00 sec 2.55 GBytes 22.0 Gbits/sec
[ 13] 2.00-3.00 sec 2.08 GBytes 17.9 Gbits/sec
[ 13] 3.00-4.00 sec 2.31 GBytes 19.8 Gbits/sec
[ 13] 4.00-5.00 sec 2.46 GBytes 21.1 Gbits/sec
[ 13] 5.00-6.00 sec 2.48 GBytes 21.3 Gbits/sec
[ 13] 6.00-7.00 sec 2.33 GBytes 20.0 Gbits/sec
[ 13] 7.00-8.00 sec 2.50 GBytes 21.5 Gbits/sec
[ 13] 8.00-9.00 sec 2.62 GBytes 22.5 Gbits/sec
[ 13] 9.00-10.00 sec 2.62 GBytes 22.5 Gbits/sec
[ 13] 10.00-10.04 sec 65.7 MBytes 16.9 Gbits/sec
[ 13] 0.00-10.04 sec 0.00 Bytes 0.00 bits/sec
[ 13] 0.00-10.04 sec 24.6 GBytes 21.0 Gbits/sec
sender receiver
-----
Server listening on 5201
[ 13] Interval Transfer Bandwidth
[ 13] 0.00-10.04 sec 0.00 Bytes 0.00 bits/sec
[ 13] 0.00-10.04 sec 24.6 GBytes 21.0 Gbits/sec
-----
```

gnuplot> plot "mytcp\_s1" title "tcp\_1sw" with linespoints, "mytcp\_s3" title "tcp\_3sw" with linespoints, "mytcp" title "tcp\_5sw" with linespoints, "mytcp\_s2" title "tcp\_2sw" with linespoints, "mytcp\_s4" title "tcp\_4sw" with linespoints  
Gtk::Message: 22:41:44.769: Failed to load module "canberra-gtk-module"  
gnuplot>  
(gnuplot:14399): GLib-CRITICAL \*\*: 22:42:00.341: Source ID 22 was not found when attempting to remove it



X 軸為 時間 ,y 軸為 頻寬 Gbits/sec

### Udp iperf

注意 UDP 頻寬有預設的最大值，需調整頻寬最大值才能看出差異

-u 使用udp協議

Client specific:

-b, --bandwidth #[KM] for UDP, bandwidth to send at in bits/sec  
(default 1 Mbit/sec, implies -u)

## Udp - 1 node

```
pinu@pinu-virtual-machine:~$ sudo mn
[sudo] password for pinu:
*** No default OpenFlow controller found for default switch!
*** Falling back to OVS Bridge
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1)
*** Configuring hosts
h1 h2
*** Starting controller
*** Starting 1 switches
s1 ...
*** Starting CLI:
mininet> xterm h1 h2
mininet> [ ]
```

"Node: h2"

```
root@pinu-virtual-machine:~# iperf -c 10.0.0.1 -u -b 10M > udp_s1
read failed: Connection refused
[ 11] WARNING: did not receive ack of last datagram after 5 tries.
root@pinu-virtual-machine:~# iperf3 -c 10.0.0.1 -u -b 10M > udp_s1
root@pinu-virtual-machine:~# [ ]
```

"Node: h1"

```
Accepted connection from 10.0.0.2, port 53946
[ 13] local 10.0.0.1 port 5201 connected to 10.0.0.2 port 37842
[ 10] Interval Transfer Bandwidth Jitter Lost/Total Datag
rame
[ 13] 0.00-1.00 sec 1.08 MBytes 9.04 Mbits/sec 0.013 ms 0/138 (0%)
[ 13] 1.00-2.00 sec 1.20 MBytes 10.0 Mbits/sec 0.005 ms 0/153 (0%)
[ 13] 2.00-3.00 sec 1.19 MBytes 9.76 Mbits/sec 0.020 ms 1/150 (0.67%)
[ 13] 3.00-4.00 sec 1.15 MBytes 9.51 Mbits/sec 0.005 ms 10/155 (6.53%)
[ 13] 4.00-5.00 sec 1.15 MBytes 9.57 Mbits/sec 0.005 ms 0/152 (0%)
[ 13] 5.00-6.00 sec 1.20 MBytes 10.0 Mbits/sec 0.007 ms 0/153 (0%)
[ 13] 6.00-7.00 sec 1.19 MBytes 9.57 Mbits/sec 0.005 ms 0/152 (0%)
[ 13] 7.00-8.00 sec 1.19 MBytes 9.56 Mbits/sec 0.005 ms 0/152 (0%)
[ 13] 8.00-9.00 sec 1.20 MBytes 10.0 Mbits/sec 0.004 ms 0/153 (0%)
[ 13] 9.00-10.00 sec 1.19 MBytes 9.56 Mbits/sec 0.004 ms 0/152 (0%)
[ 13] 10.00-10.04 sec 8.00 KBytes 1.58 Mbits/sec 0.007 ms 0/1 (0%)
[ 10] Interval Transfer Bandwidth Jitter Lost/Total Datag
rame
[ 13] 0.00-10.04 sec 0.00 Bytes 0.00 bits/sec 0.007 ms 11/1512 (0.73%)
Server listening on 5201
[ ]
```

```
root@pinu-virtual-machine:~# cat udp_s1 | grep sec | head -n 10 | tr - " " | awk '{print $4,$8}' > myudp_s1
```

## Udp - 2 nodes

```
pinu@pinu-virtual-machine:~$ sudo mn --custom topo-2sw-2host.py --topo mytopo
[sudo] password for pinu:
*** No default OpenFlow controller found for default switch!
*** Falling back to OVS Bridge
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1 s2
*** Adding links:
(h1, s1) (s1, s2) (s2, h2)
*** Configuring hosts
h1 h2
*** Starting controller
*** Starting 2 switches
s1 s2 ...
*** Starting CLI:
mininet> xterm h1 h2
mininet> [ ]
```

"Node: h1"

```
Accepted connection from 10.0.0.2, port 53962
[ 15] local 10.0.0.1 port 5201 connected to 10.0.0.2 port 49045
[ 10] Interval Transfer Bandwidth Jitter Lost/Total Datag
rame
[ 15] 0.00-1.00 sec 1.06 MBytes 8.90 Mbits/sec 0.005 ns 0/138 (0%)
[ 15] 1.00-2.00 sec 1.12 MBytes 9.38 Mbits/sec 0.008 ns 12/155 (7.7%)
[ 15] 2.00-3.00 sec 1.18 MBytes 9.76 Mbits/sec 0.020 ns 1/150 (0.67%)
[ 15] 3.00-4.00 sec 1.19 MBytes 9.56 Mbits/sec 0.005 ns 0/152 (0%)
[ 15] 4.00-5.00 sec 1.19 MBytes 9.56 Mbits/sec 0.005 ns 0/152 (0%)
[ 15] 5.00-6.00 sec 1.18 MBytes 9.69 Mbits/sec 0.009 ns 5/152 (3.3%)
[ 15] 6.00-7.00 sec 1.17 MBytes 9.63 Mbits/sec 0.009 ns 2/152 (1.3%)
[ 15] 7.00-8.00 sec 1.18 MBytes 9.50 Mbits/sec 0.005 ns 2/152 (1.3%)
[ 15] 8.00-9.00 sec 1.18 MBytes 9.50 Mbits/sec 0.005 ns 2/152 (1.3%)
[ 15] 9.00-10.00 sec 1.16 MBytes 9.70 Mbits/sec 0.006 ns 4/152 (2.6%)
[ 15] 10.00-10.04 sec 8.00 KBytes 1.53 Mbits/sec 0.012 ns 42/1512 (2.8%)
[ 10] Interval Transfer Bandwidth Jitter Lost/Total Datag
rame
[ 15] 0.00-10.04 sec 0.00 Bytes 0.00 bits/sec 0.012 ns 42/1512 (2.8%)
Server listening on 5201
[ ]
```

"Node: h2"

```
root@pinu-virtual-machine:~# iperf3 -c 10.0.0.1 -u -b 10M > udp_s2
root@pinu-virtual-machine:~# cat udp_s2 | grep sec | head -n 10 | tr - " " | awk '{print $4,$8}' > myudp_s2
root@pinu-virtual-machine:~# [ ]
```

## Udp – 3 nodes

The screenshot shows a Linux desktop environment with two terminal windows. The left window is titled "pinu@pinu-virtual-machine: ~" and displays the command-line interface for setting up a 3-node UDP network. The right window is titled "Node: h1" and shows the iperf3 server listening on port 53970. The terminal output includes performance metrics for the connection between h1 and h2.

```
examples.desktop mytcp_s3 tcp_s1 topo-3sw-2host.py
Laravel60 mytcp_s4 tcp_s1 topo-3sw-2host.py
mytcp myudp_s1 tcp_s2 topo-4sw-2host.py
mytcp_s1 myudp_s2 tcp_s3 topo-5sw-2host.py
pinu@pinu-virtual-machine:~$ sudo mn --custom topo-3sw-2host.py --topo mytopo
*** No default OpenFlow controller found for default switch!
*** Falling back to OVS Bridge
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1 s2 s3
*** Adding links:
(h1, s1) (s1, s2) (s2, s3) (s3, h2)
*** Configuring hosts
h1 h2
*** Starting controller
*** Starting 3 switches
s1 s2 s3 ...
*** Starting CLI:
mininet> xterm h1 h2
mininet> 
```

```
Accepted connection from 10.0.0.2, port 53970
[ 17] local 10.0.0.1 port 53970 connected to 10.0.0.2 port 44317
[ 18] Interval Transfer Bandwidth Jitter Lost/Total Datagrams
[ 17] 0.00-1.00 sec 1.08 MBytes 9.04 Mbits/sec 0.005 ms 0/138 (0%)
[ 17] 1.00-2.00 sec 1.20 MBytes 10.0 Mbits/sec 0.005 ms 0/153 (0%)
[ 17] 2.00-3.00 sec 1.19 MBytes 9.95 Mbits/sec 0.005 ms 0/152 (0%)
[ 17] 3.00-4.00 sec 1.19 MBytes 9.95 Mbits/sec 0.005 ms 0/152 (0%)
[ 17] 4.00-5.00 sec 1.19 MBytes 9.95 Mbits/sec 0.005 ms 0/152 (0%)
[ 17] 5.00-6.00 sec 1.20 MBytes 10.0 Mbits/sec 0.005 ms 0/153 (0%)
[ 17] 6.00-7.00 sec 1.20 MBytes 10.0 Mbits/sec 0.005 ms 0/153 (0%)
[ 17] 7.00-8.00 sec 1.19 MBytes 9.95 Mbits/sec 0.005 ms 0/152 (0%)
[ 17] 8.00-9.00 sec 1.20 MBytes 10.0 Mbits/sec 0.005 ms 0/153 (0%)
[ 17] 9.00-10.00 sec 1.19 MBytes 9.95 Mbits/sec 0.005 ms 0/152 (0%)
[ 17] 10.00-10.04 sec 0.00 KBytes 1.51 Mbytes 0.010 ms 0/1 (0%)
[ 18] Interval Transfer Bandwidth Jitter Lost/Total Datagrams
[ 17] 0.00-10.04 sec 0.00 Bytes 0.00 bits/sec 0.010 ms 0/1512 (0%)
Server listening on 5201

```

```
"Node: h2"
root@pinu-virtual-machine:~# iperf3 -c 10.0.0.1 -u -b 10M > udp_s3
root@pinu-virtual-machine:~# cat udp_s3 | grep sec | head -n 10 | tr -s " "
root@pinu-virtual-machine:~# 
```

## Udp – 4 nodes

The screenshot shows a Linux desktop environment with two terminal windows. The left window is titled "pinu@pinu-virtual-machine: ~" and displays the command-line interface for setting up a 4-node UDP network. The right window is titled "Node: h1" and shows the iperf3 server listening on port 53984. The terminal output includes performance metrics for the connection between h1 and h2.

```
examples.desktop mytcp_s3 tcp_s1 topo-2sw-2host.py udp_s2
Laravel60 mytcp_s4 tcp_s1 topo-3sw-2host.py udp_s3
mytcp myudp_s1 tcp_s2 topo-4sw-2host.py
mytcp_s1 myudp_s2 tcp_s3 topo-5sw-2host.py
pinu@pinu-virtual-machine:~$ sudo mn --custom topo-4sw-2host.py --topo mytopo
*** No default OpenFlow controller found for default switch!
*** Falling back to OVS Bridge
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1 s2 s3 s4
*** Adding links:
(h1, s1) (s1, s2) (s2, s3) (s3, s4) (s4, h2)
*** Configuring hosts
h1 h2
*** Starting controller
*** Starting 4 switches
s1 s2 s3 s4 ...
*** Starting CLI:
mininet> xterm h1 h2
mininet> 
```

```
Accepted connection from 10.0.0.2, port 53984
[ 19] local 10.0.0.1 port 53984 connected to 10.0.0.2 port 49298
[ 19] Interval Transfer Bandwidth Jitter Lost/Total Datagrams
[ 19] 0.00-1.00 sec 1.08 MBytes 9.05 Mbits/sec 0.007 ms 0/138 (0%)
[ 19] 1.00-2.00 sec 1.14 MBytes 9.57 Mbits/sec 0.036 ms 4/150 (2.7%)
[ 19] 2.00-3.00 sec 1.13 MBytes 9.51 Mbits/sec 0.478 ms 10/155 (6.5%)
[ 19] 3.00-4.00 sec 1.14 MBytes 9.57 Mbits/sec 0.007 ms 7/153 (4.6%)
[ 19] 4.00-5.00 sec 1.19 MBytes 9.95 Mbits/sec 0.005 ms 0/152 (0%)
[ 19] 5.00-6.00 sec 1.14 MBytes 9.57 Mbits/sec 0.005 ms 14/153 (9.2%)
[ 19] 6.00-7.00 sec 1.14 MBytes 9.57 Mbits/sec 0.011 ms 7/153 (4.6%)
[ 19] 7.00-8.00 sec 1.19 MBytes 9.97 Mbits/sec 0.005 ms 0/152 (0%)
[ 19] 8.00-9.00 sec 1.20 MBytes 10.0 Mbits/sec 0.005 ms 0/153 (0%)
[ 19] 9.00-10.00 sec 1.19 MBytes 9.95 Mbits/sec 0.005 ms 0/152 (0%)
[ 19] 10.00-10.04 sec 8.00 KBytes 1.53 Mbytes 0.008 ms 0/1 (0%)
[ 19] Interval Transfer Bandwidth Jitter Lost/Total Datagrams
[ 19] 0.00-10.04 sec 0.00 Bytes 0.00 bits/sec 0.008 ms 42/1512 (2.8%)
Server listening on 5201

```

```
"Node: h2"
root@pinu-virtual-machine:~# iperf3 -c 10.0.0.1 -u -b 10M > udp_s4
root@pinu-virtual-machine:~# cat udp_s4 | grep sec | head -n 10 | tr -s " "
root@pinu-virtual-machine:~# 
```

## Udp – 5 nodes

```
pinu@pinu-virtual-machine: ~
檔案(F) 編輯(E) 檢視(V) 搜尋(S) 終端機(T) 助手(H)

Laravel60 myudp_s1 tcp_s2 topo-5sw-2host.py 公共 圖片 音樂
mytcp myudp_s2 tcp_s3 udp_s1 圖片 影音
mytcp_s1 myudp_s3 tcp_s4 udp_s2 影音
mytcp_s2 myudp_s4 topo-2sw-2host.py udp_s3 文件
pinu@pinu-virtual-machine: $ sudo mn --custom topo-5sw-2host.py --topo mytopo
*** No default OpenFlow controller found for default switch!
*** Falling back to OVS Bridge
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1 s2 s3 s4 s5
*** Adding links:
(h1, s1) (s1, s2) (s2, s3) (s3, s4) (s4, s5) (s5, h2)
*** Configuring hosts
h1 h2
*** Starting controller

*** Starting 5 switches
s1 s2 s3 s4 s5 ...
*** Starting CLI:
mininet> xterm h1 h2
mininet> [ ]
```

"Node: h1"

```
Accepted connection from 10.0.0.2, port 5390
[ 21] local 10.0.0.1 port 5201 connected to 10.0.0.2 port 39324
[ ID] Interval Transfer Bandwidth Jitter Lost/Total Datag
[ 21] 0.00-1.00 sec 1.08 MBytes 9.04 Mbits/sec 0.005 ms 0/138 (OK)
[ 21] 1.00-2.00 sec 1.28 MBytes 10.0 Mbits/sec 0.005 ms 0/152 (OK)
[ 21] 2.00-3.00 sec 1.19 MBytes 9.96 Mbits/sec 0.005 ms 0/152 (OK)
[ 21] 3.00-4.00 sec 1.20 MBytes 10.0 Mbits/sec 0.005 ms 0/153 (OK)
[ 21] 4.00-5.00 sec 1.13 MBytes 9.96 Mbits/sec 0.005 ms 0/152 (OK)
[ 21] 5.00-6.00 sec 1.20 MBytes 10.0 Mbits/sec 0.005 ms 0/153 (OK)
[ 21] 6.00-7.00 sec 1.20 MBytes 10.0 Mbits/sec 0.005 ms 0/153 (OK)
[ 21] 7.00-8.00 sec 1.17 MBytes 9.84 Mbits/sec 0.005 ms 0/150 (OK)
[ 21] 8.00-9.00 sec 1.17 MBytes 9.83 Mbits/sec 0.005 ms 0/150 (OK)
[ 21] 9.00-10.00 sec 1.17 MBytes 9.83 Mbits/sec 0.005 ms 5/155 (3.2%)
[ 21] 10.00-10.04 sec 8.00 KBytes 1.52 Mbits/sec 0.008 ms 0/1 (OK)
```

"Node: h2"

```
[ ID] Interval Transfer Bandwidth Jitter Lost/Total Datag
[ 21] 0.00-10.04 sec 0.00 Bytes 0.00 bits/sec 0.008 ms 5/1512 (0.33%)
```

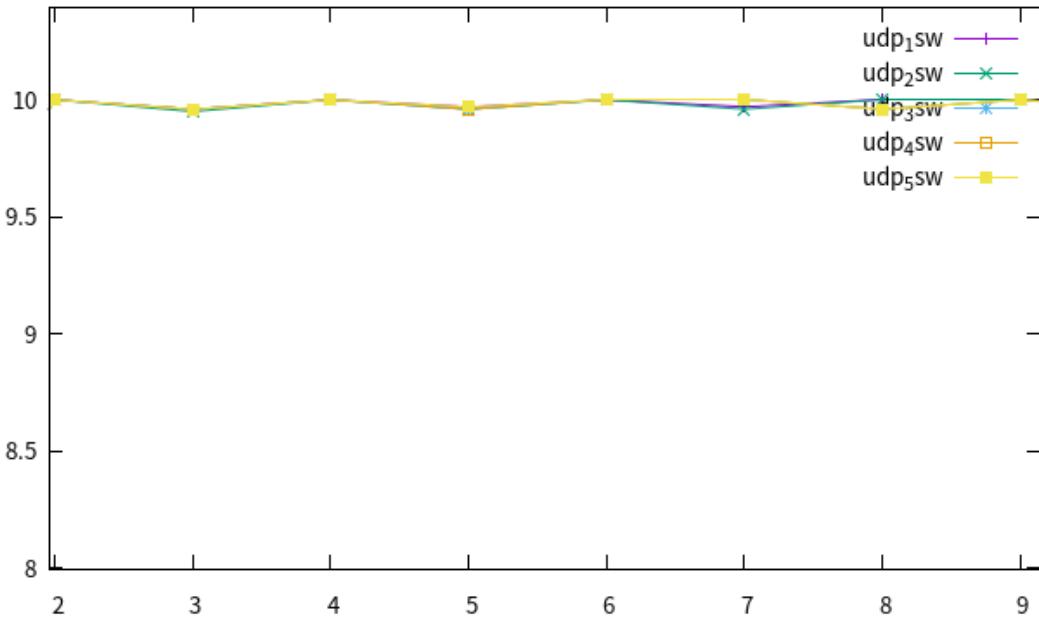
```
Server listening on 5201
```

[ ]

```
root@pinu-virtual-machine: # iperf3 -c 10.0.0.1 -u -b 10M > udp.s5
root@pinu-virtual-machine: # cat udp.s5 | grep sec | head -n 10 | tr - " "
root '|print $4;83|' > myudp_s5
root@pinu-virtual-machine: #
```

# Gunplot 繪圖

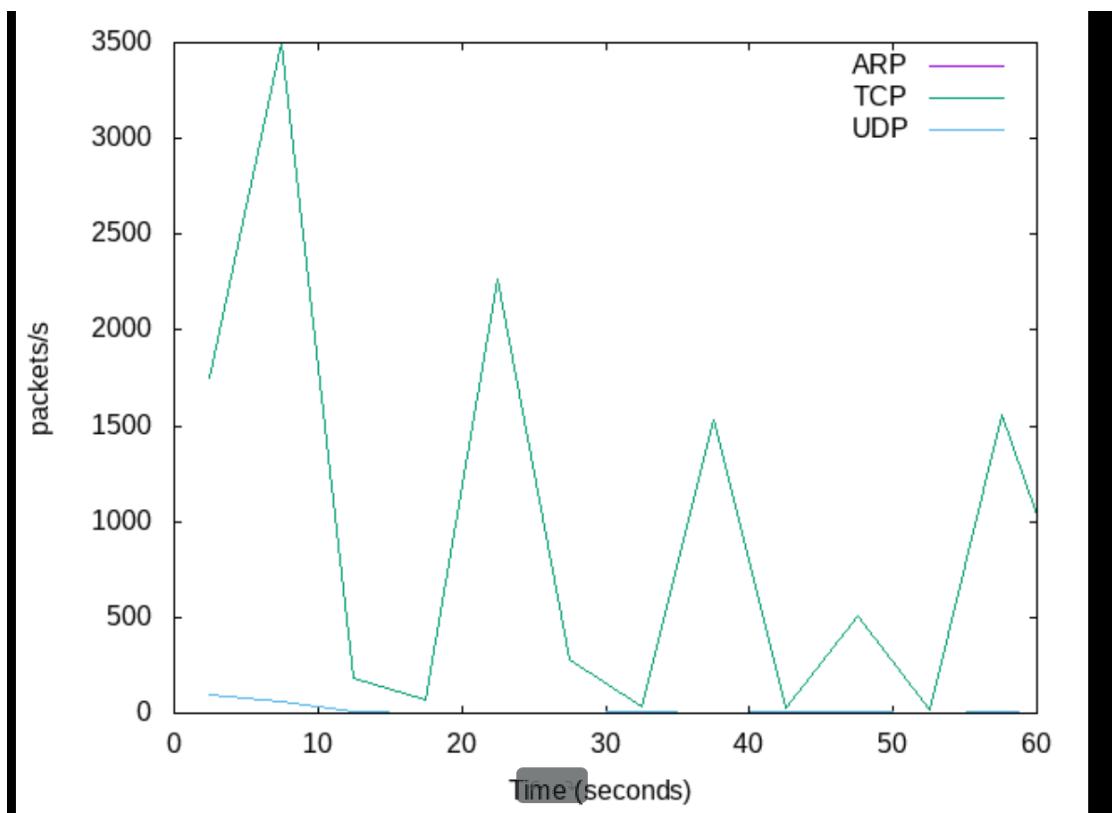
```
gnuplot> plot "myudp_s1" title "udp_1sw" with linespoints, "myudp_s2" title "udp_2sw" with linespoints, "myudp_s3" title "udp_3sw" with linespoints, "myudp_s4" title "udp_4sw" with linespoints, "myudp_s5" title "udp_5sw" with linespoints  
gnuplot> set yrange[9:12]  
gnuplot> replot
```



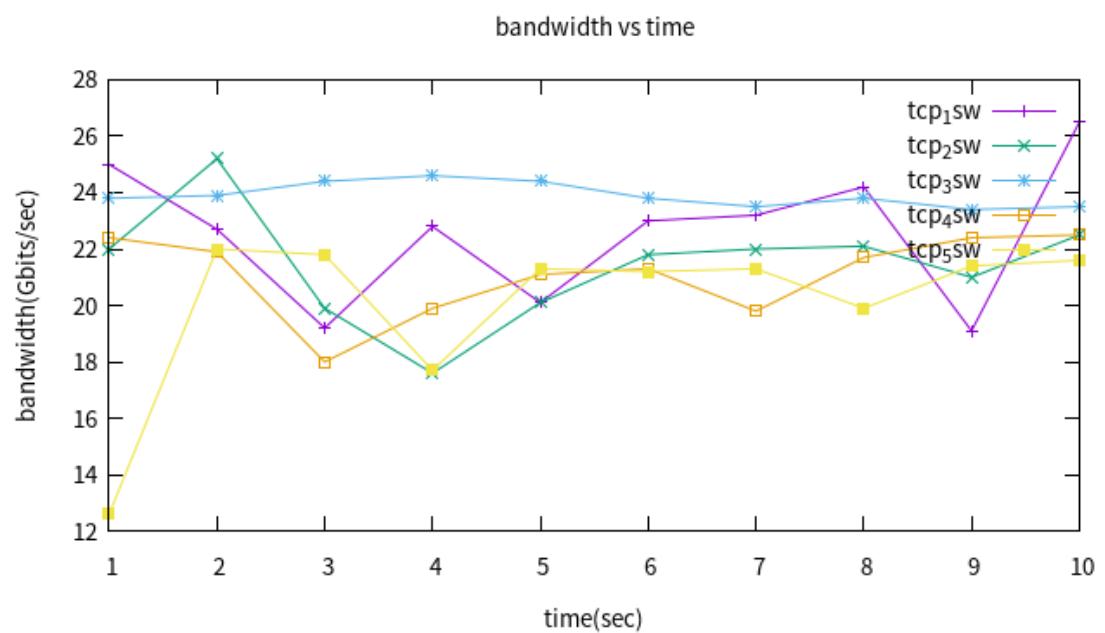
X 軸為 時間 , y 軸為 頻寬 Gbits/sec

## [Summary]

第一張圖

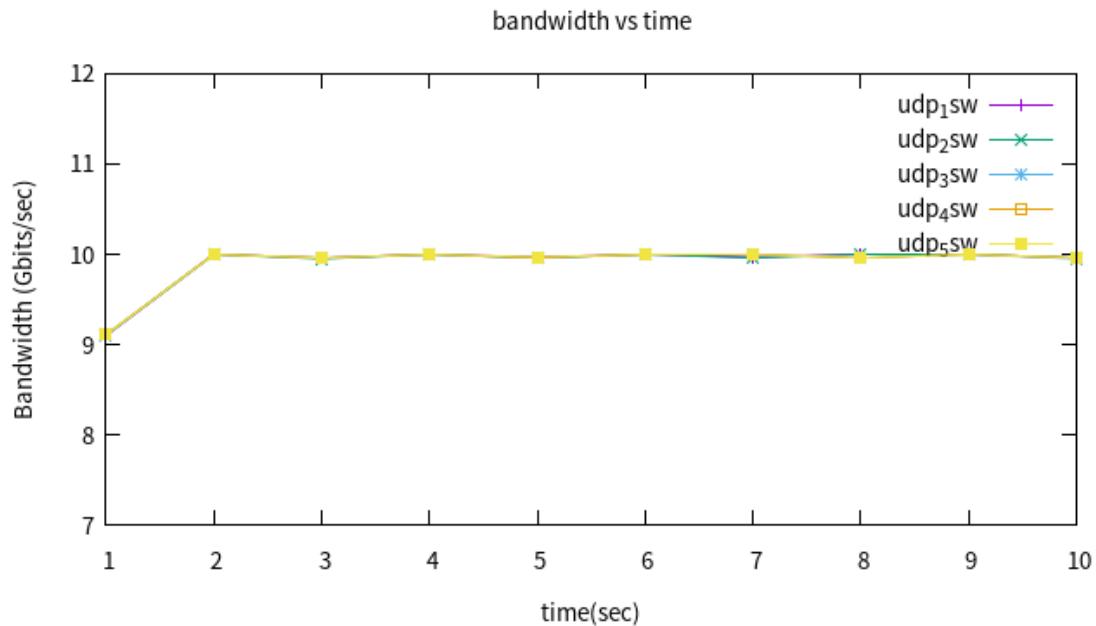


第二張圖 – TCP 加上 label & title

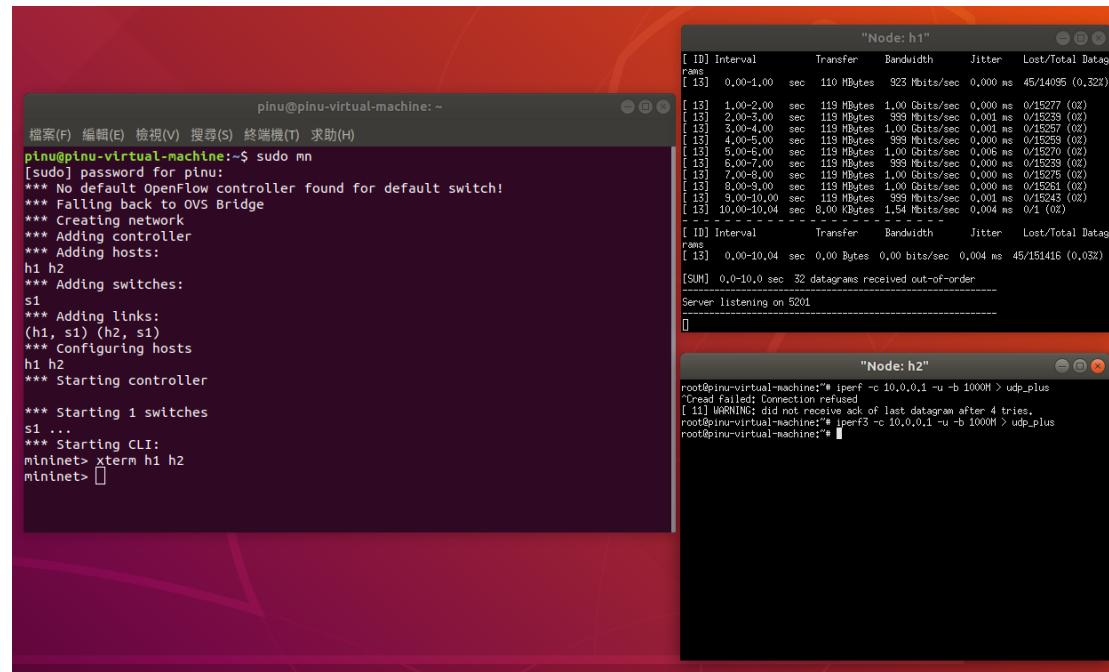


第三張圖 – UDP 加上 label & title

[10M 版]



[改進版 :1000M]



pinu@pinu-virtual-machine:~

```
[sudo] password for pinu:
*** No default OpenFlow controller found for default switch!
*** Falling back to OVS Bridge
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1)
*** Configuring hosts
h1 h2
*** Starting controller

*** Starting 1 switches
s1 ...
*** Starting CLI:
mininet> xterm h1 h2
mininet> 
```

"Node: h1"

ID	Interval	Transfer	Bandwidth	Jitter	Lost/Total Data
[ 13]	0.00-1.00	sec 110 MBytes	923 Mbits/sec	0.000 ms	45/14095 (0.32%)
[ 13]	1.00-2.00	sec 119 MBytes	1.00 Gbits/sec	0.000 ms	0/15239 (0%)
[ 13]	2.00-3.00	sec 119 MBytes	999 Mbits/sec	0.001 ms	0/15239 (0%)
[ 13]	3.00-4.00	sec 119 MBytes	1.00 Gbits/sec	0.001 ms	0/15257 (0%)
[ 13]	4.00-5.00	sec 119 MBytes	999 Mbits/sec	0.000 ms	0/15257 (0%)
[ 13]	5.00-6.00	sec 119 MBytes	1.00 Gbits/sec	0.000 ms	0/15270 (0%)
[ 13]	6.00-7.00	sec 119 MBytes	999 Mbits/sec	0.000 ms	0/15239 (0%)
[ 13]	7.00-8.00	sec 119 MBytes	1.00 Gbits/sec	0.000 ms	0/15275 (0%)
[ 13]	8.00-9.00	sec 119 MBytes	1.00 Gbits/sec	0.000 ms	0/15261 (0%)
[ 13]	9.00-10.00	sec 119 MBytes	999 Mbits/sec	0.001 ms	0/15245 (0%)
[ 13]	10.00-10.04	sec 8.00 KBytes	1.54 Mbits/sec	0.004 ms	0/1 (0%)

[ ID] Interval Transfer Bandwidth Jitter Lost/Total Data

[SUM] 0.0-10.04 sec 0.00 Bytes 0.00 bits/sec 0.004 ms 45/151416 (0.03%)

[SUM] 0.0-10.0 sec 32 datagrams received out-of-order

Server listening on 5201

"Node: h2"

```
root@pinu-virtual-machine:~# iperf -c 10.0.0.1 -u -b 1000M > udp_plus
[11] WARNING: did not receive ack of last datagram after 4 tries.
root@pinu-virtual-machine:~# iperf3 -c 10.0.0.1 -u -b 1000M > udp_plus
root@pinu-virtual-machine:~# cat udp_plus | grep see | head -n 10 | tr - " " | awk '{print $4,$5}' > myudp_plus
root@pinu-virtual-machine:~# 
```

"Node: h1"

ID	Interval	Transfer	Bandwidth	Jitter	Lost/Total Data
[ 13]	0.00-1.00	sec 110 MBytes	923 Mbits/sec	0.000 ms	45/14095 (0.32%)
[ 13]	1.00-2.00	sec 119 MBytes	1.00 Gbits/sec	0.000 ms	0/15277 (0%)
[ 13]	2.00-3.00	sec 119 MBytes	999 Mbits/sec	0.000 ms	0/15239 (0%)
[ 13]	3.00-4.00	sec 119 MBytes	1.00 Gbits/sec	0.000 ms	0/15257 (0%)
[ 13]	4.00-5.00	sec 119 MBytes	999 Mbits/sec	0.000 ms	0/15257 (0%)
[ 13]	5.00-6.00	sec 119 MBytes	1.00 Gbits/sec	0.006 ms	0/15270 (0%)
[ 13]	6.00-7.00	sec 119 MBytes	999 Mbits/sec	0.000 ms	0/15239 (0%)
[ 13]	7.00-8.00	sec 119 MBytes	1.00 Gbits/sec	0.000 ms	0/15275 (0%)
[ 13]	8.00-9.00	sec 119 MBytes	1.00 Gbits/sec	0.000 ms	0/15261 (0%)
[ 13]	9.00-10.00	sec 119 MBytes	999 Mbits/sec	0.001 ms	0/15245 (0%)
[ 13]	10.00-10.04	sec 8.00 KBytes	1.54 Mbits/sec	0.004 ms	0/1 (0%)

[ ID] Interval Transfer Bandwidth Jitter Lost/Total Data

[SUM] 0.0-10.04 sec 0.00 Bytes 0.00 bits/sec 0.004 ms 45/151416 (0.03%)

[SUM] 0.0-10.0 sec 32 datagrams received out-of-order

Server listening on 5201

"Node: h2"

```
G N U P L O T
Version 5.2 patchlevel 2  last modified 2017-11-01
Copyright (C) 1986-1993, 1998, 2004, 2007-2017
Thomas Williams, Colin Kelley and many others
gnuplot home: http://www.gnuplot.info
faq, bugs, etc: type "help FAQ"
immediate help: type "help" (plot window: hit 'h')
terminal type is now 'wxt'
gnuplot> plot "myudp_plus" title "udp_s1_plus" with linepoints
[tk-Messag: 15:57:04,000]: Failed to load module 'camberra-gtk-module'
gnuplot:38(17): [IBUS-WARNING **: 15:57:05,398]: The owner of '/home/pinu/.config/ibus/bus' is not root!
gnuplot:38(17): [IBUS-WARNING **: 15:57:05,440]: Unable to connect to ibus: Could not connect: Connection refused
gnuplot> 
```

G N U P L O T

Version 5.2 patchlevel 2 last modified 2017-11-01

Copyright (C) 1986-1993, 1998, 2004, 2007-2017

Thomas Williams, Colin Kelley and many others

gnuplot home: <http://www.gnuplot.info>

faq, bugs, etc: type "help FAQ"

immediate help: type "help" (plot window: hit 'h')

terminal type is now 'wxt'

gnuplot> plot "myudp\_plus" title "udp\_s1\_plus" with linepoints

[tk-Messag: 15:57:04,000]: Failed to load module 'camberra-gtk-module'

gnuplot:38(17): [IBUS-WARNING \*\*: 15:57:05,398]: The owner of '/home/pinu/.config/ibus/bus' is not root!

gnuplot:38(17): [IBUS-WARNING \*\*: 15:57:05,440]: Unable to connect to ibus: Could not connect: Connection refused

gnuplot>

6.53712, 1058.62

Gnuplot (window id : 0)

pinu@pinu-virtual-machine: ~

```
*** Done
completed in 673.543 seconds
pinu@pinu-virtual-machine:~$ sudo mn --custom topo-2sw-2host.py --topo mytopo
*** No default OpenFlow controller found for default switch!
*** Falling back to OVS Bridge
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1 s2
*** Adding links:
(h1, s1) (s1, s2) (s2, h2)
*** Configuring hosts
h1 h2
*** Starting controller

*** Starting 2 switches
s1 s2 ...
*** Starting CLI:
mininet> xterm
usage: xterm node1 node2 ...
mininet> xterm h1 h2
mininet> 
```

"Node: h1"

```
Accepted connection from 10.0.0.2, port 54354
[ 15] local 10.0.0.1 port 5201 connected to 10.0.0.2 port 41953
[ 15] Interval Transfer Bandwidth Jitter Lost/Total Datаг
рамs
[ 15] 0.00~1.00 sec 111 MBBytes 927 Mbit/sec 0.001 ms 0/14147 (0%)
[ 15] 1.00~2.00 sec 119 MBBytes 1.00 Gbit/sec 0.002 ms 0/15267 (0%)
[ 15] 2.00~3.00 sec 120 MBBytes 1.00 Gbit/sec 0.001 ms 0/15301 (0%)
[ 15] 3.00~4.00 sec 119 MBBytes 1.00 Gbit/sec 0.001 ms 0/15260 (0%)
[ 15] 4.00~5.00 sec 119 MBBytes 997 Mbit/sec 0.006 ms 0/15214 (0%)
[ 15] 5.00~6.00 sec 119 MBBytes 1000 Mbit/sec 0.001 ms 0/15251 (0%)
[ 15] 6.00~7.00 sec 119 MBBytes 999 Mbit/sec 0.001 ms 0/15223 (0%)
[ 15] 7.00~8.00 sec 119 MBBytes 998 Mbit/sec 0.006 ms 0/15239 (0%)
[ 15] 8.00~9.00 sec 119 MBBytes 999 Mbit/sec 0.001 ms 0/15275 (0%)
[ 15] 9.00~10.00 sec 119 MBBytes 999 Mbit/sec 0.001 ms 0/15251 (0%)
[ 15] 10.00~10.04 sec 8.00 KBytes 1.55 Mbit/sec 0.004 ms 0/1 (0%)
[ 15] Interval Transfer Bandwidth Jitter Lost/Total Datаг
рамс
[ 15] 0.00~10.04 sec 0.00 Bytes 0.00 bits/sec 0.004 ms 0/151475 (0%)
Server listening on 5201
[ ]
```

"Node: h2"

```
root@pinu-virtual-machine:~# ./portf -c 10.0.0.1 -u -b 1000M > udp_s2_plus
root@pinu-virtual-machine:~# cat udp_s2_plus | grep sec | head -n 10 | tr -s ' '
I ask '$print $4,$8' > ./udp_s2_plus
root@pinu-virtual-machine:~# 
```

"Node: h1"

```
[ 17] Interval Transfer Bandwidth Jitter Lost/Total Datаг
рамс
[ 17] 0.00~1.00 sec 111 MBBytes 950 Mbit/sec 0.001 ms 0/14195 (0%)
[ 17] 1.00~2.00 sec 118 MBBytes 994 Mbit/sec 0.001 ms 119/15248 (0.78%)
[ 17] 2.00~3.00 sec 118 MBBytes 994 Mbit/sec 0.004 ms 120/15275 (0.79%)
[ 17] 3.00~4.00 sec 119 MBBytes 999 Mbit/sec 0.000 ms 0/15248 (0%)
[ 17] 4.00~5.00 sec 119 MBBytes 1.00 Gbit/sec 0.001 ms 0/15248 (0%)
[ 17] 5.00~6.00 sec 119 MBBytes 995 Mbit/sec 0.001 ms 0/15193 (0%)
[ 17] 6.00~7.00 sec 120 MBBytes 1.00 Gbit/sec 0.001 ms 0/15193 (0%)
[ 17] 7.00~8.00 sec 119 MBBytes 1000 Mbit/sec 0.001 ms 0/15226 (0%)
[ 17] 8.00~9.00 sec 120 MBBytes 1.00 Gbit/sec 0.002 ms 0/15206 (0%)
[ 17] 9.00~10.00 sec 119 MBBytes 997 Mbit/sec 0.000 ms 0/15212 (0%)
[ 17] 10.00~10.04 sec 8.00 KBytes 1.55 Mbit/sec 0.004 ms 0/1 (0%)
[ 17] Interval Transfer Bandwidth Jitter Lost/Total Datаг
рамс
[ 17] 0.00~10.04 sec 0.00 Bytes 0.00 bits/sec 0.004 ms 233/151527 (0.16%)
Server listening on 5201
[ ]
```

"Node: h2"

```
GNUPLOT Version 5.2 patchlevel 2 last modified 2017-11-01
Copyright (C) 1998-1999, 1998, 2004, 2007-2017
Thomas Williams, Colin Kelley and many others
gnuplot home: http://www.gnuplot.info
fat, buss, etc: type "help FOO"
immediate help: type "help" "(plot window; hit 'h')"
Terminal type is now 'xwt'.
gnuplot> plot "udp_s3_plus" title "udp_s3_plus" with linespoints
Sta-Message: 13:54:22.845: Failed to load module "cumbera-gtk-module"
ibus/bus is not root!
(gnuplot:39826): IBUS-WARNING **: 13:54:22.845: The owner of /home/pinu/.config/ibus/bus was not found
(gnuplot:39826): GLib-CRITICAL **: 13:54:22.845: Unable to connect to ibus: Could
not connect; Connection refused
gnuplot>
(gnuplot:39826): GLib-CRITICAL **: 13:54:22.845: Source ID 13 was not found when
attempting to remove it
[ ]
```

Gnuplot (window id : 0)

pinu@pinu-virtual-machine: ~

```
檔案(F) 編輯(E) 檢視(V) 搜尋(S) 終端機(T) 求助(H)
*** Stopping 2 hosts
h1 h2
*** Done
completed in 266.196 seconds
pinu@pinu-virtual-machine:~$ sudo mn --custom topo-4sw-2host.py --topo mytopo
*** No default OpenFlow controller found for default switch!
*** Falling back to OVS Bridge
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1 s2 s3 s4
*** Adding links:
(h1, s1) (s1, s2) (s2, s3) (s3, s4) (s4, h2)
*** Configuring hosts
h1 h2
*** Starting controller
*** Starting 4 switches
s1 s2 s3 s4 ...

```

Gnuplot (window id : 0)

7.37920, 1033.94

"Node: h1"

```
Accepted connection from 10.0.0.2, port 5436
[ 19] local 10.0.0.1 port 5201 connected to 10.0.0.2 port 3800
[ 19] Interval Transfer Bandwidth Jitter Lost/Total Data
[ 19] 0.00-1.00 sec 111 MBytes 934 Mbit/sec 0.004 ms 0/14955 (0%)
[ 19] 1.00-2.00 sec 119 MBytes 998 Mbit/sec 0.002 ms 0/15242 (0%)
[ 19] 2.00-3.00 sec 119 MBytes 998 Mbit/sec 0.002 ms 0/15232 (0%)
[ 19] 3.00-4.00 sec 119 MBytes 998 Mbit/sec 0.001 ms 0/15238 (0%)
[ 19] 4.00-5.00 sec 119 MBytes 1.00 Mbit/sec 0.004 ms 0/15272 (0%)
[ 19] 5.00-6.00 sec 119 MBytes 1.00 Mbit/sec 0.001 ms 0/15266 (0%)
[ 19] 6.00-7.00 sec 119 MBytes 1.00 Mbit/sec 0.002 ms 0/15275 (0%)
[ 19] 7.00-8.00 sec 120 MBytes 1.00 Mbit/sec 0.001 ms 0/15267 (0%)
[ 19] 8.00-9.00 sec 119 MBytes 997 Mbit/sec 0.002 ms 0/15209 (0%)
[ 19] 9.00-10.00 sec 119 MBytes 998 Mbit/sec 0.003 ms 0/15236 (0%)
[ 19] 10.00-10.04 sec 8.00 KBytes 1.61 Mbit/sec 0.006 ms 0/1 (0%)
[ 19] Interval Transfer Bandwidth Jitter Lost/Total Data
[ 19] 0.00-10.04 sec 0.00 Bytes 0.00 bits/sec 0.006 ms 0/151539 (0%)
Server listening on 5201
[ 19]
```

"Node: h2"

```
GNUPLOT
Version 5.2 patchlevel 2 last modified 2017-11-01
Copyright (C) 1990-1993, 1998, 2004, 2007-2017
Thomas Williams, Colin Kelley and many others

gnuplot home: http://www.gnuplot.info
FAQ, bugs, etc: type "help FAQ"
immediate help: type "help" (plot window: hit 'h')

Terminal type is now 'wxt'
gnuplot: plot "myudp_s4_plus" title "udp_s4_plus" with linespoints
Gtk-Message: 14:01:41.109: Failed to load module "canberra-gtk-module"

(gnuplot:39174): IBUS-WARNING **: 14:01:41.149: The owner of /home/pinu/.config/ibus/bus is not root!

(gnuplot:39174): IBUS-WARNING **: 14:01:41.161: Unable to connect to ibus: Could
not connect; Connection refused
(gnuplot:39174): GLib-CRITICAL **: 14:01:41.161: Source ID 14 was not found when
attempting to remove it
[ 19]
```

pinu@pinu-virtual-machine: ~

```
檔案(F) 編輯(E) 檢視(V) 搜尋(S) 終端機(T) 求助(H)
*** Stopping 2 hosts
h1 h2
*** Done
completed in 431.085 seconds
pinu@pinu-virtual-machine:~$ sudo mn --custom topo-5sw-2host.py --topo mytopo
*** No default OpenFlow controller found for default switch!
*** Falling back to OVS Bridge
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1 s2 s3 s4 s5
*** Adding links:
(h1, s1) (s1, s2) (s2, s3) (s3, s4) (s4, s5) (s5, h2)
*** Configuring hosts
h1 h2
*** Starting controller
*** Starting 5 switches
s1 s2 s3 s4 s5 ...

```

Gnuplot (window id : 0)

root@pinu-virtual-machine:~# cat udp\_s5\_plus | grep sec | head -n 10 | tr - " "
| awk '{print \$4,\$8}' > myudp\_s5\_plus
root@pinu-virtual-machine:~# gnuplot

"Node: h1"

```
[ 21] 0.00-1.00 sec 90.0 MBytes 754 Mbit/sec 0.002 ms 2687/14162 (18%)
[ 21] 1.00-2.00 sec 120 MBytes 1.01 Mbit/sec 0.002 ms 0/15379 (0%)
[ 21] 2.00-3.00 sec 119 MBytes 1000 Mbit/sec 0.002 ms 0/15258 (0%)
[ 21] 3.00-4.00 sec 119 MBytes 1.00 Mbit/sec 0.003 ms 0/15260 (0%)
[ 21] 4.00-5.00 sec 119 MBytes 1000 Mbit/sec 0.004 ms 0/15247 (0%)
[ 21] 5.00-6.00 sec 119 MBytes 1000 Mbit/sec 0.008 ms 0/15259 (0%)
[ 21] 6.00-7.00 sec 119 MBytes 1.00 Mbit/sec 0.009 ms 0/15250 (0%)
[ 21] 7.00-8.00 sec 119 MBytes 999 Mbit/sec 0.004 ms 0/15242 (0%)
[ 21] 8.00-9.00 sec 119 MBytes 1.00 Mbit/sec 0.005 ms 0/15232 (0%)
[ 21] 9.00-10.00 sec 119 MBytes 994 Mbit/sec 0.001 ms 107/15270 (0.7%)
[ 21] 10.00-10.04 sec 8.00 KBytes 1.55 Mbit/sec 0.005 ms 0/1 (0%)
[ 21] Interval Transfer Bandwidth Jitter Lost/Total Data
[ 21] 0.00-10.04 sec 0.00 Bytes 0.00 bits/sec 0.005 ms 2794/151628 (1.8%)
[SUM] 0.0-10.0 sec 45 datagrams received out-of-order
Server listening on 5201
[ 21]
```

"Node: h2"

```
GNUPLOT
Version 5.2 patchlevel 2 last modified 2017-11-01
Copyright (C) 1990-1993, 1998, 2004, 2007-2017
Thomas Williams, Colin Kelley and many others

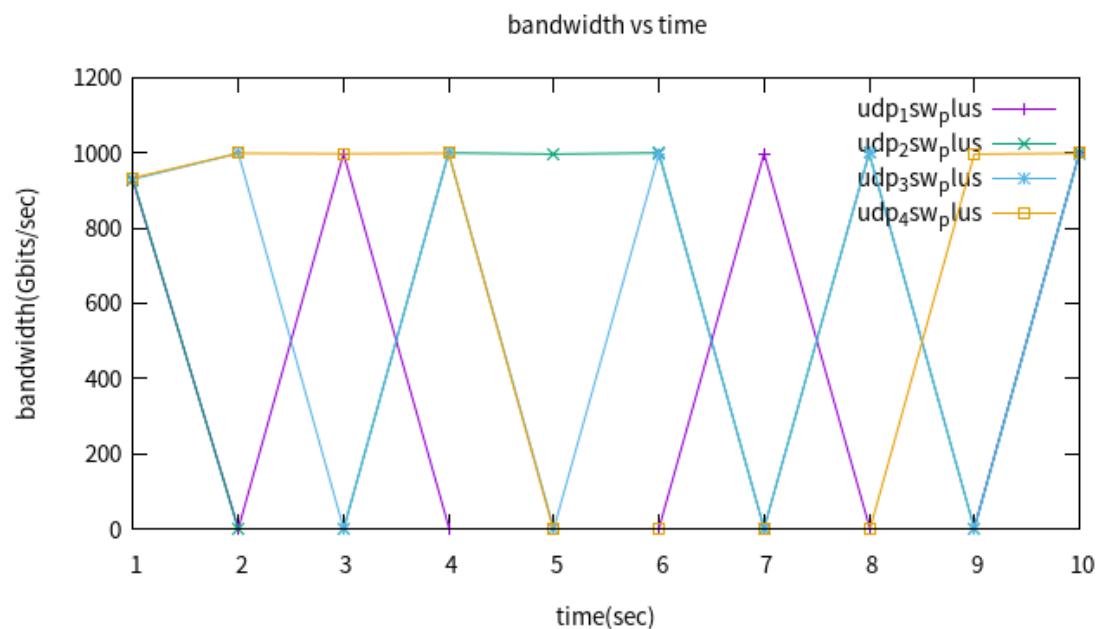
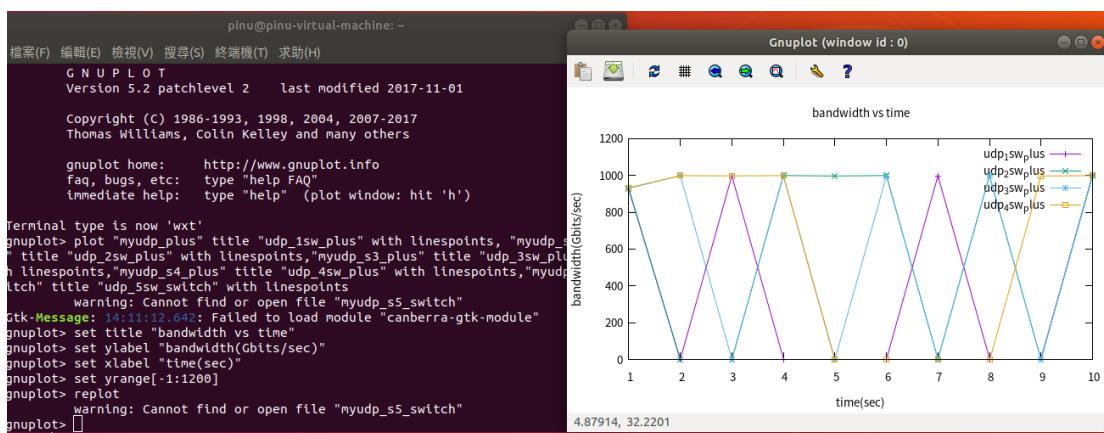
gnuplot home: http://www.gnuplot.info
FAQ, bugs, etc: type "help FAQ"
immediate help: type "help" (plot window: hit 'h')

Terminal type is now 'wxt'
gnuplot: plot "myudp_s5_plus" title "udp_s5_plus" with linespoints
Gtk-Message: 14:06:34.356: Failed to load module "canberra-gtk-module"

(gnuplot:39647): IBUS-WARNING **: 14:06:34.369: The owner of /home/pinu/.config/ibus/bus is not root!

(gnuplot:39647): IBUS-WARNING **: 14:06:34.388: Unable to connect to ibus: Could
not connect; Connection refused
(gnuplot:39647): GLib-CRITICAL **: 14:06:34.388: Source ID 14 was not found when
attempting to remove it
[ 21]
```

```
pinu@pinu-virtual-machine:~$ ls
composer.phar      myudp_s2
examples.desktop   myudp_s2_plus
Laravel60          myudp_s3
mytcp              myudp_s3_plus
mytcp_s1           myudp_s4
mytcp_s2           myudp_s4_plus
mytcp_s3           myudp_s5
mytcp_s4           myudp_s5_plus
myudp_plus         tcp
myudp_s1           tcp_s1
```



#### 4-4.

請說明 TCP 及 UDP 產生結果差異的原因。

**TCP 起起伏伏**

**UDP 到某個時間趨於穩定 且不受 節點數影響(!?)**

產生結果差異的原因

可能是 TCP 有做 error recovery ，而 UDP 沒做。

整個比較下來 TCP 有高有低 而 UDP 到某個時間趨於穩定。

## 5. netperf

- a. 測量 Client 與 Server 間的 **TCP** 網路效能
- b. 測量 Client 與 Server 間的 **UDP** 網路效能
- c.

```
pinu@pinu-virtual-machine:~$ sudo apt-get install netperf
[sudo] password for pinu:
正在讀取套件清單... 完成
正在重建相依關係
正在讀取狀態資料... 完成
下列【新】套件將會被安裝：
  netperf
升級 0 個，新安裝 1 個，移除 0 個，有 113 個未被升級。
需要下載 550 kB 的套件檔。
此操作完成之後，會多佔用 1,155 kB 的磁碟空間。
下載:1 http://tw.archive.ubuntu.com/ubuntu bionic/multiverse amd64 netperf amd64 2.6.0-2.1 [550 kB]
取得 550 kB 用了 0秒 (1,266 kB/s)
選取了原先未選的套件 netperf。
(讀取資料庫 ... 目前共安裝了 171377 個檔案和目錄。)
準備解開 .../netperf_2.6.0-2.1_amd64.deb ...
解開 netperf (2.6.0-2.1) 中...
Processing triggers for ureadahead (0.100.0-21) ...
Processing triggers for systemd (237-3ubuntu10.29) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
設定 netperf (2.6.0-2.1) ...
Processing triggers for systemd (237-3ubuntu10.29) ...
Processing triggers for ureadahead (0.100.0-21) ...
```

下載 netperf

The screenshot shows two terminal windows side-by-side. The left window is titled "Node: h1" and contains the command "root@pinu-virtual-machine:~# netserver". The right window is titled "Node: h2" and contains the command "root@pinu-virtual-machine:~# netperf -H 10.0.0.1". The output from the netperf command is displayed, showing a single data transfer with a throughput of 22140.72 Gbytes/sec.

```
"Node: h1"
root@pinu-virtual-machine:~# netserver
Starting netserver with host 'IN(6)ADDR_ANY' port '12865' and family AF_UNSPEC
root@pinu-virtual-machine:~# 

"Node: h2"
root@pinu-virtual-machine:~# netperf -H 10.0.0.1
MIGRATED TCP STREAM TEST from 0.0.0.0 (0.0.0.0) port 0 AF_INET to 10.0.0.1 () port 0 AF_INET : demo
Recv  Send  Send
Socket Socket Message Elapsed
Size  Size  Size  Time   Throughput
bytes bytes bytes secs.  10^6bits/sec
87380 87380 87380 10.00  22140.72
root@pinu-virtual-machine:~#
```

[TCP]

假設 H1為 server , h2 為client  
在 h1 下 netserver  
在 h2 下 netperf -H 10.0.0.1

[UDP]

The screenshot shows two terminal windows side-by-side. The left window is titled "Node: h1" and the right window is titled "Node: h2". Both windows display the output of the netperf command.

**Node: h1**

```
root@pinu-virtual-machine:~# netserver
Starting netserver with host 'IN(6)ADDR_ANY' port '12865' and family AF_UNSPEC
root@pinu-virtual-machine:~#
```

**Node: h2**

```
root@pinu-virtual-machine:~# netperf -H 10.0.0.1
MIGRATED TCP STREAM TEST from 0.0.0.0 (0.0.0.0) port 0 AF_INET to 10.0.0.1 () port 0 AF_INET : demo
Recv Send Send
Socket Socket Message Elapsed
Size Size Size Time Throughput
bytes bytes bytes secs. 10^6bits/sec
87380 87380 87380 10.00 22140.72
root@pinu-virtual-machine:~# netperf UDP_STREAM -H 10.0.0.1
MIGRATED TCP STREAM TEST from 0.0.0.0 (0.0.0.0) port 0 AF_INET to 10.0.0.1 () port 0 AF_INET : demo
Recv Send Send
Socket Socket Message Elapsed
Size Size Size Time Throughput
bytes bytes bytes secs. 10^6bits/sec
87380 87380 87380 10.00 22089.68
root@pinu-virtual-machine:~#
```

在 UDP client 端 運行netperf時，多加一個 參數 UDP\_STREAM