[2021 Advanced Computer Networks Homework 2]

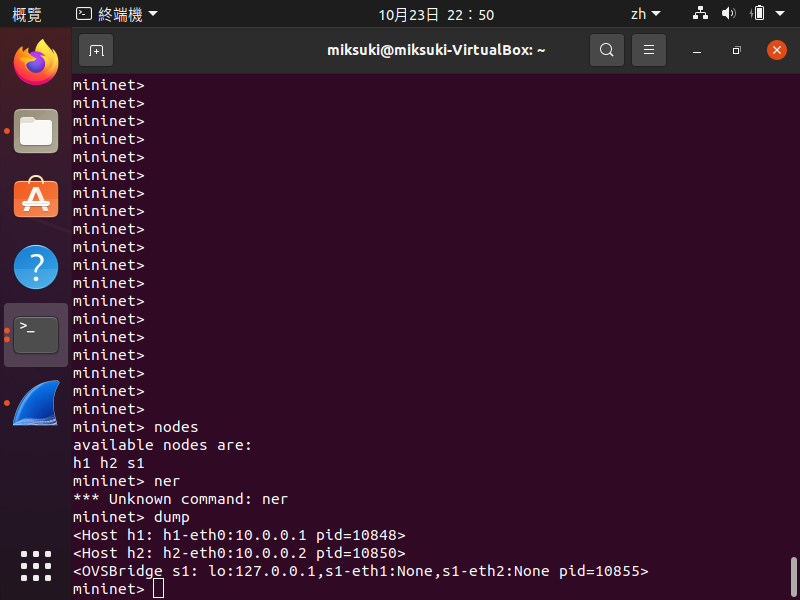
# Part1:

**Mininet**

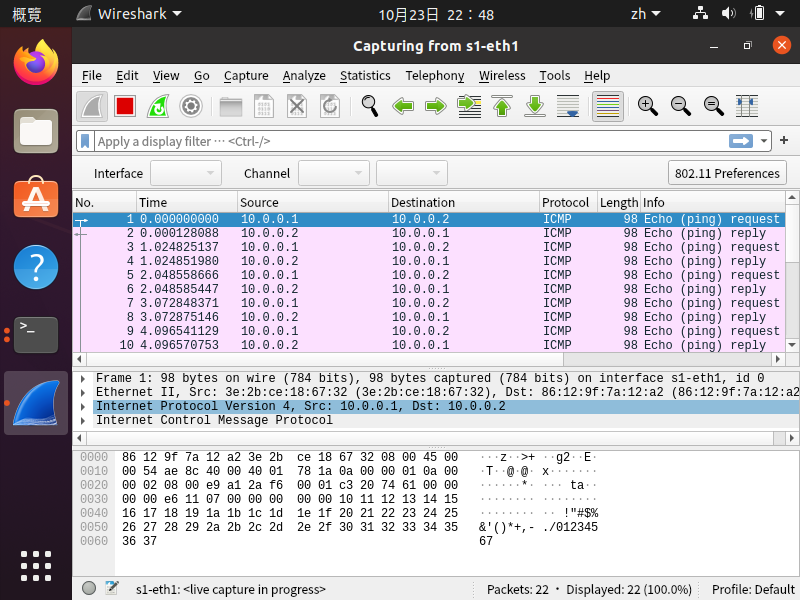
本部分將讓同學練習安裝mininet，並利用mininet 建立一個簡單的虛擬網路平台。請上網查詢安裝方式及基本使用的指令。

* 1. 安裝mininet，可藉由**apt-get install mininet** 或其他方式完成安裝。
  2. 使用”**mn**”可建立基本的虛擬拓樸。
  3. 有幾個基本指令可以顯示現在的虛擬拓樸中節點資訊或鏈結的訊息等， 如”**nodes**”、”**net**”、”**dump**”等，請嘗試使用這些指令，並觀察顯示的訊息，可使用**help** 查詢mininet 可支援的指令。
  4. 請打開**wireshark**，擷取兩個虛擬host 的網卡，再使用mininet 從**h1 ping h2**， 將你所看到的wireshark 畫面擷取下來。

mininet 指令



Winshark 畫面



# Part2:

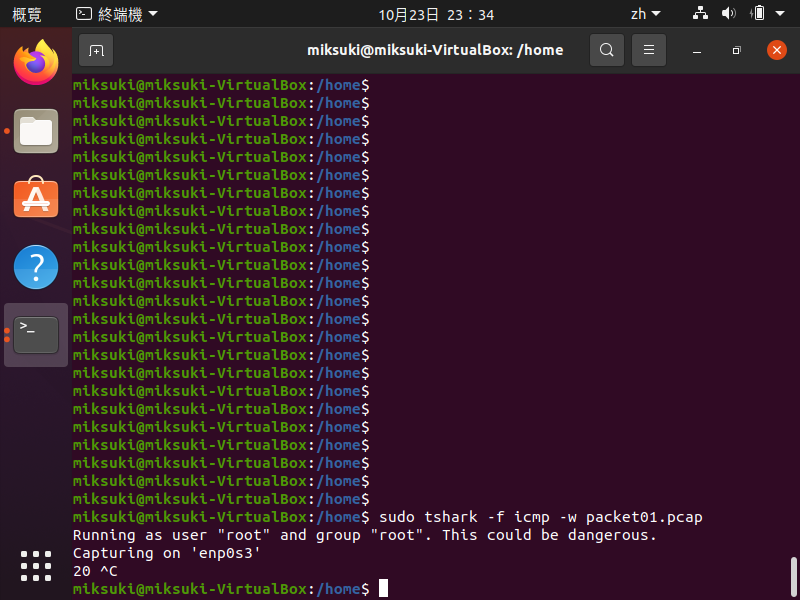
1. **tshark**

請使用tshark 指令搭配正確的參數達到以下的要求：

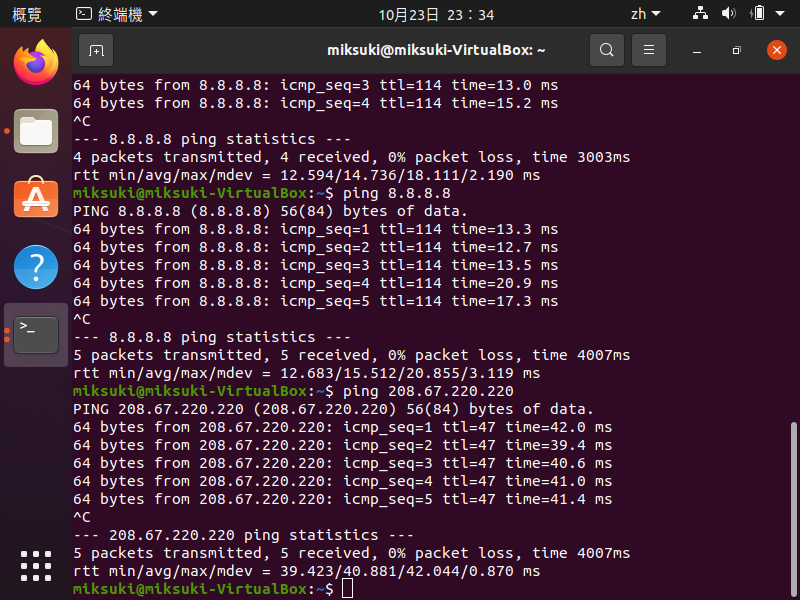
* 1. 抓取”**icmp**”的封包，並且來源或目的是”**8.8.8.8**”
  2. 將擷取的封包儲存為一個檔案，名稱為”**packet01**”
  3. 下達正確的指令後，開啟另一個視窗ping 8.8.8.8 及 ping 208.67.220.220
  4. 結束後使用tshark 指令查看擷取的檔案
  5. 以上請寫出正確的指令並截圖證明

tshark 抓取 icmp 封包，且寫入檔案 packet01 指令:

sudo tshark -f icmp -w packet01.pcap

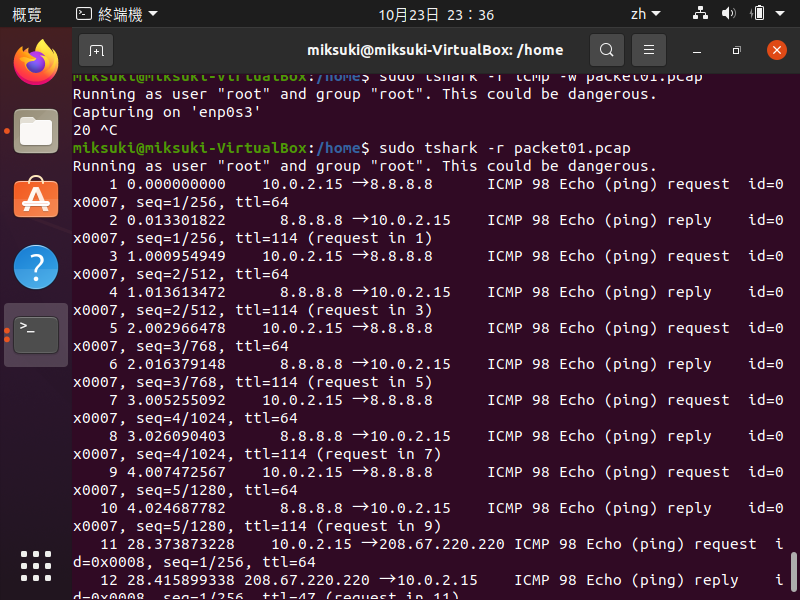


ping 8.8.8.8 及 ping 208.67.220.220



查看擷取檔案指令:

sudo tshark -r packet01.pcap



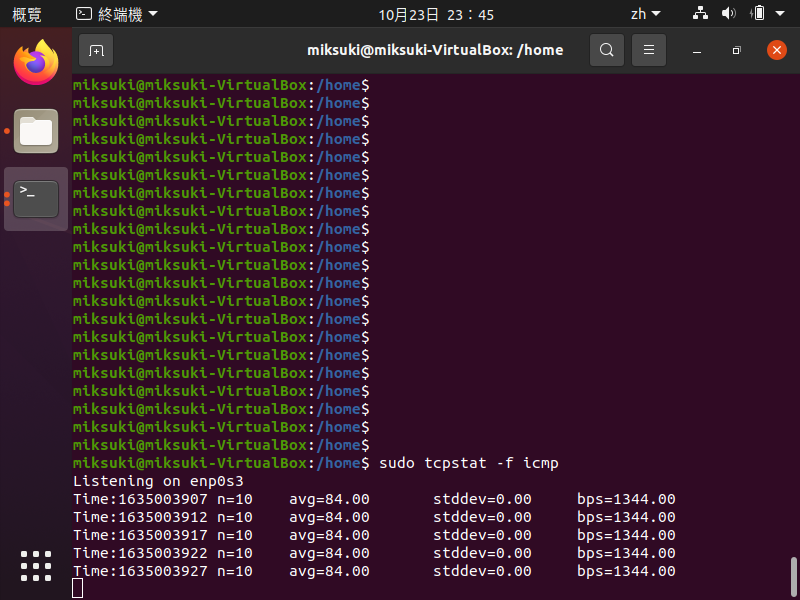
# tcpstat

請使用tcpstat 指令搭配正確的參數達到以下的要求：

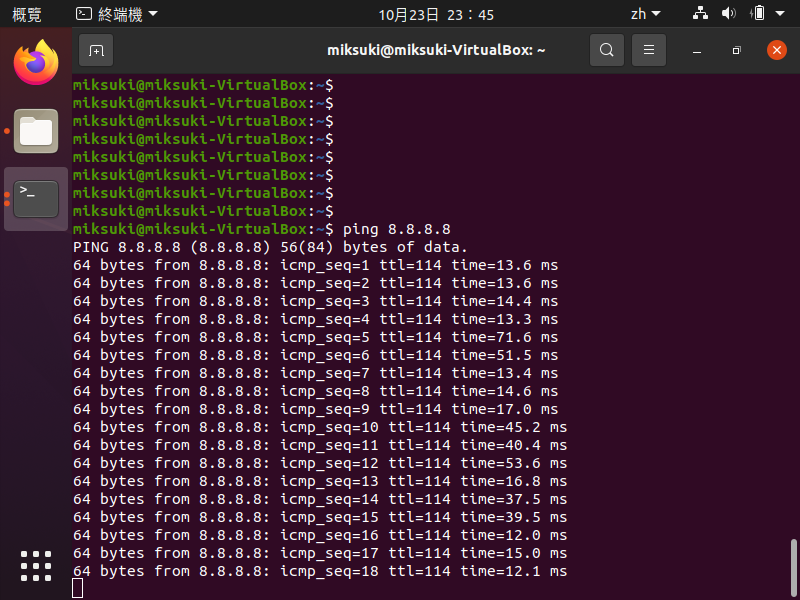
* 1. 抓取”**icmp**”的封包
  2. 開啟另一個視窗ping 任意位址
  3. 完成後中斷tcpstat，將顯示的結果截圖，並寫出正確的指令

tcpstat 指令:

sudo tcpstat -f icmp



ping任意位址: ping 8.8.8.8



# tcpdump & tcpstat & gnuplot

本題要讓同學練習使用以上三個工具，將網路流量監測的結果繪製成圖表，繳交作業時請一併附上繪製出的圖檔，請依照下列步驟操作

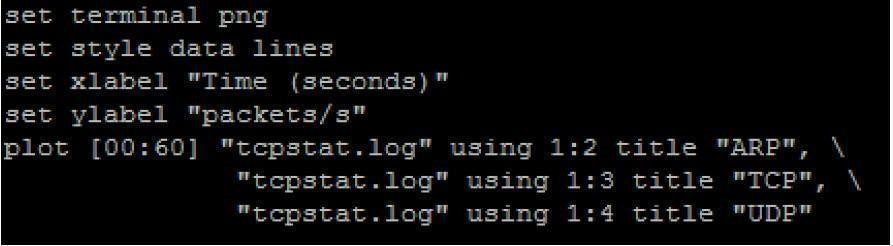
* 1. 使用tcpdump 擷取網路封包，每台電腦的網卡代號可能不同

# tcpdump -i eth1 -w rawdata.dmp

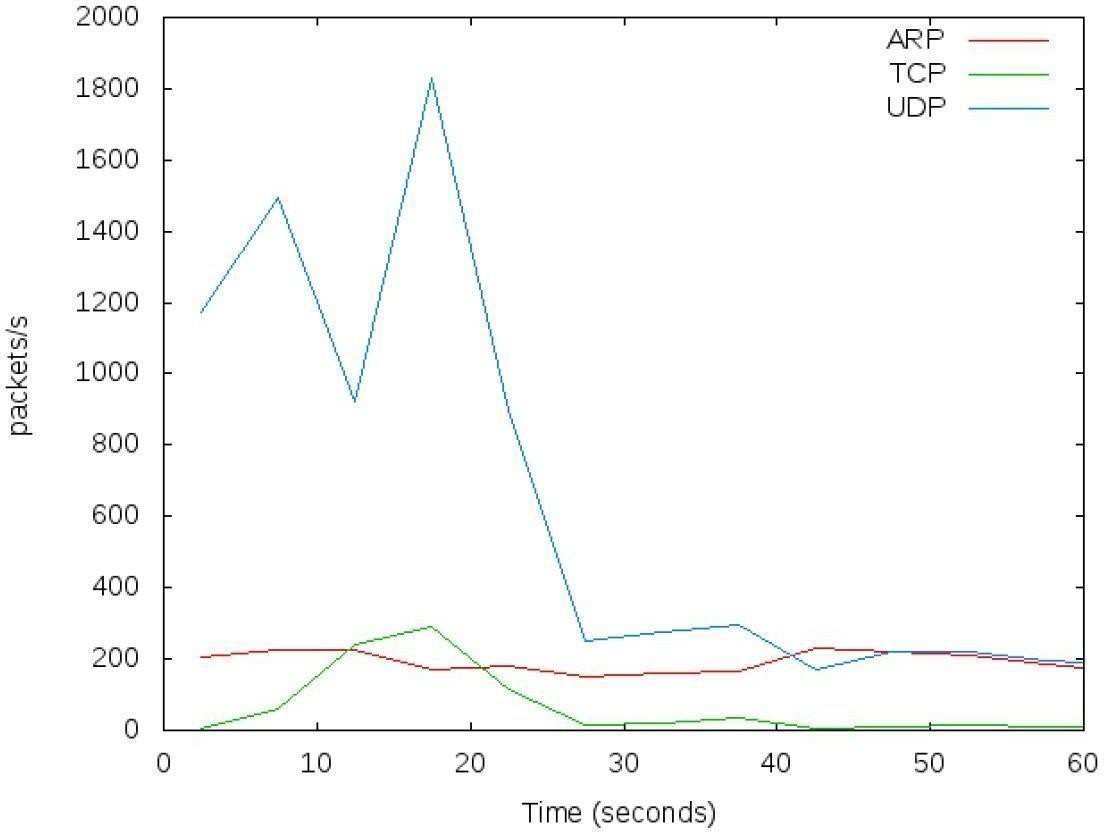
* 1. 開啟瀏覽器瀏覽網頁約一分鐘
  2. 中斷tcpdump
  3. 使用tcpstat 將擷取的檔案做格式化

# tcpstat -r rawdata.dmp -o "%r %A %T %U %l %b\n" > tcpstat.log

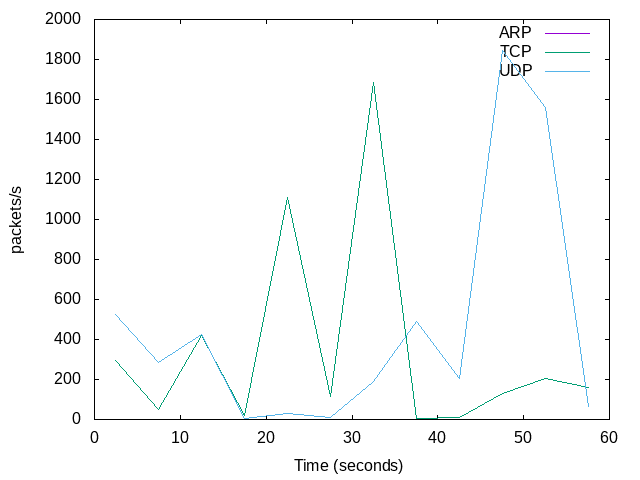
1. 使用vim 寫一個script，名稱為”**script1**”，script 內容如下圖



1. 利用gnuplot 繪圖，產生如下的圖表**gnuplot script1 > graph1.png** Sample output:



繪圖結果:



# mininet & iperf & gnuplot

本題要讓同學練習在mininet下使用iperf，將分為以下 4個部份:

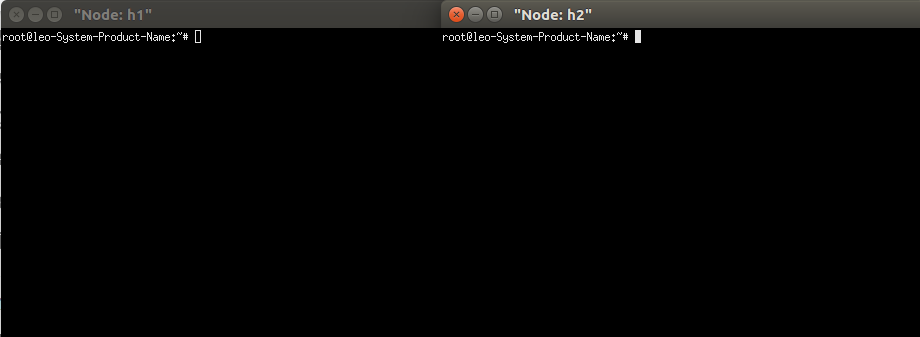
<請使用iperf3 指令完成作業> 4-1.

請說明iperf 的用途，以及在什麼情況下你會需要使用它？

4-2.

請在**mininet** 下設計**topology** ，使得**hosts(Server** 及**Client)**之間有**1**、**3**、**5** 個節點，測量**TCP** 及**UDP** 傳輸時，不同數量節點的頻寬變化，並將結果存成檔案。【利用**xterm <host>**指令開啟**host** 視窗，分別在兩個不同**host** 的**xterm**視窗下**iperf3** 指令，即可開始測量】

此題有以下幾點事項注意:

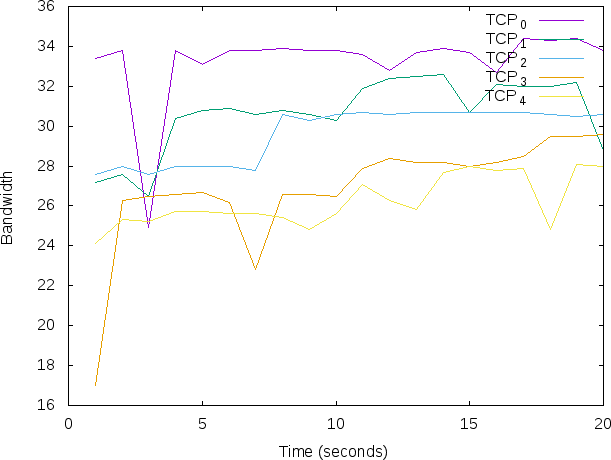


1. 需在mininet nodes 上完成，mininet nodes 如下圖所示。
2. **1**、**3**、**5** 個節點表示**Server** 和**Client** 之間有幾個**hops**。
3. 請在Client 端指令後加入”> [檔案名稱]”將輸出導入檔案中。
4. 請注意UDP 頻寬有預設的最大值，需調整頻寬最大值才能看出差異。

4-3.

請參考Part2 第三小題自行修改script，將第二步tcp 及udp 的結果使用參考指令處理後，利用gnuplot 繪製兩張結果圖。輸出圖會類似此範例，UDP 請同學自行繪製。

Sample output :



4-4.

請說明TCP 及UDP 產生結果差異的原因。

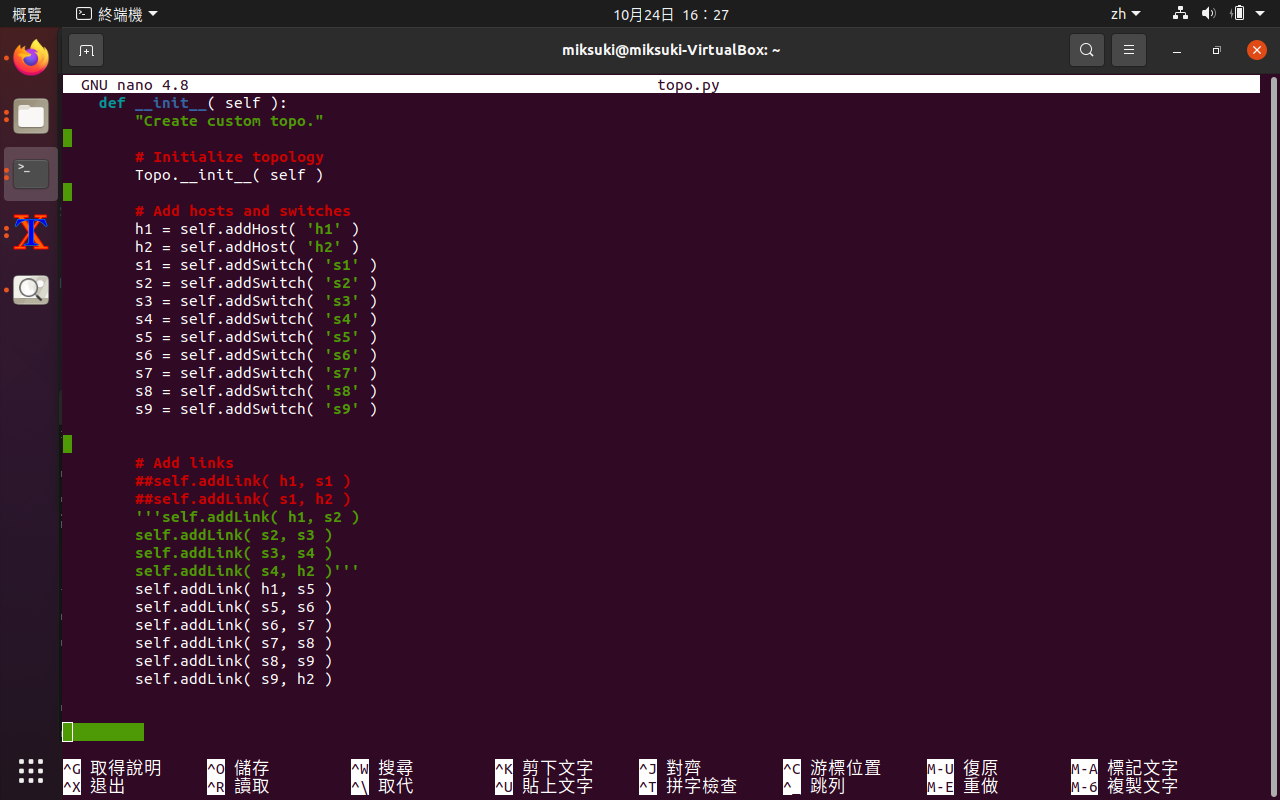
4.1

iperf 可用來測試網路效能，

當你覺得網速異常慢時，可用 iperf 來看看是哪邊出問題。

4.2 , 4.3

使用 python建立 topology



mininet 指令:

sudo mn --custom topo.py --topo mytopo

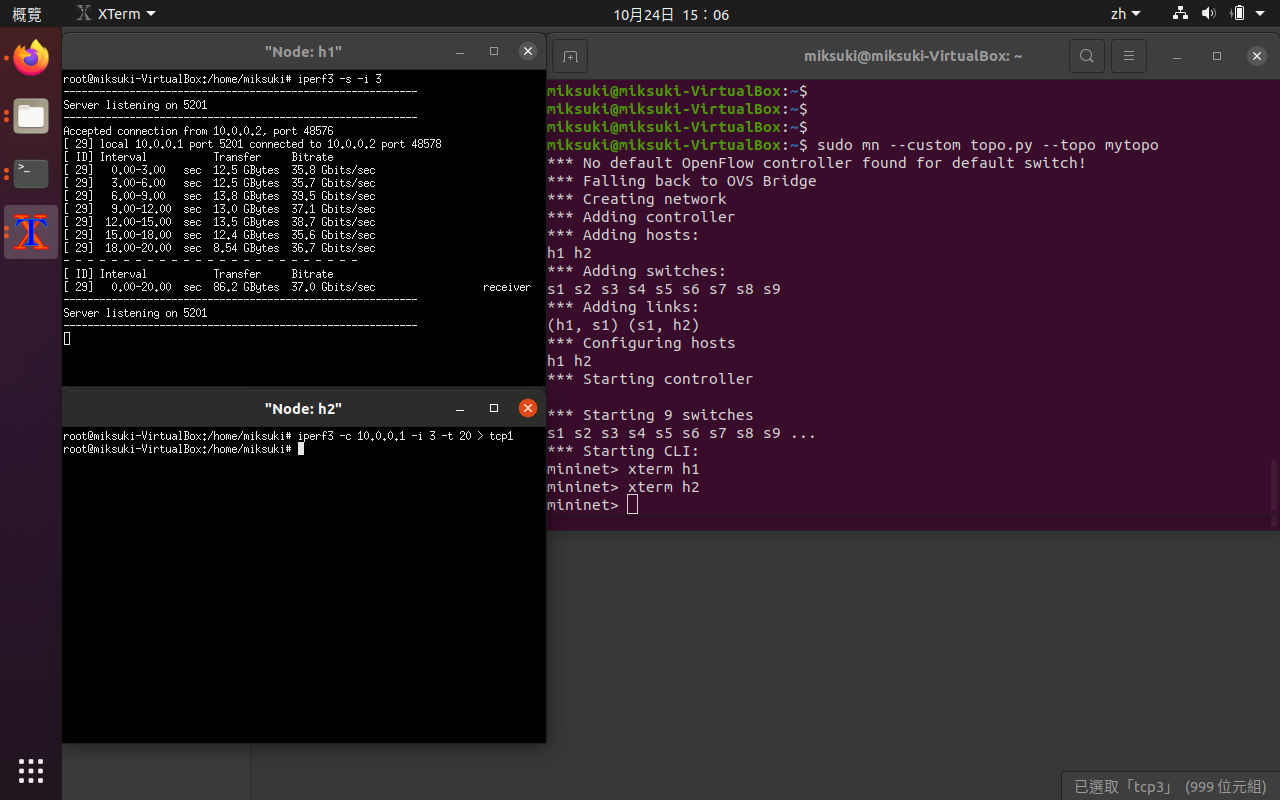
iperf-tcp 指令:

iperf3 -c 10.0.0.1 -i 3 -t 20 > filename

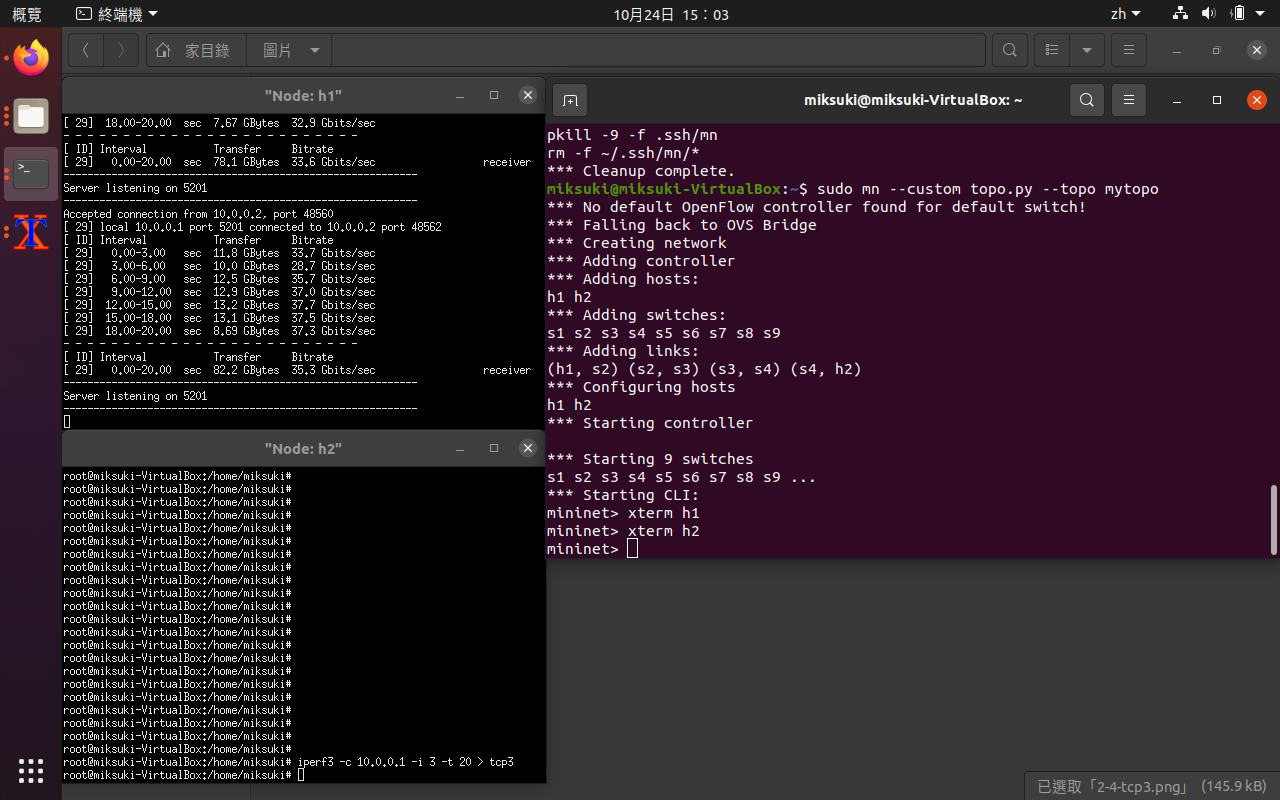
iperf-udp 指令:

iperf3 -u -c 10.0.0.1 -i 3 -t 20 -b 512G > filename

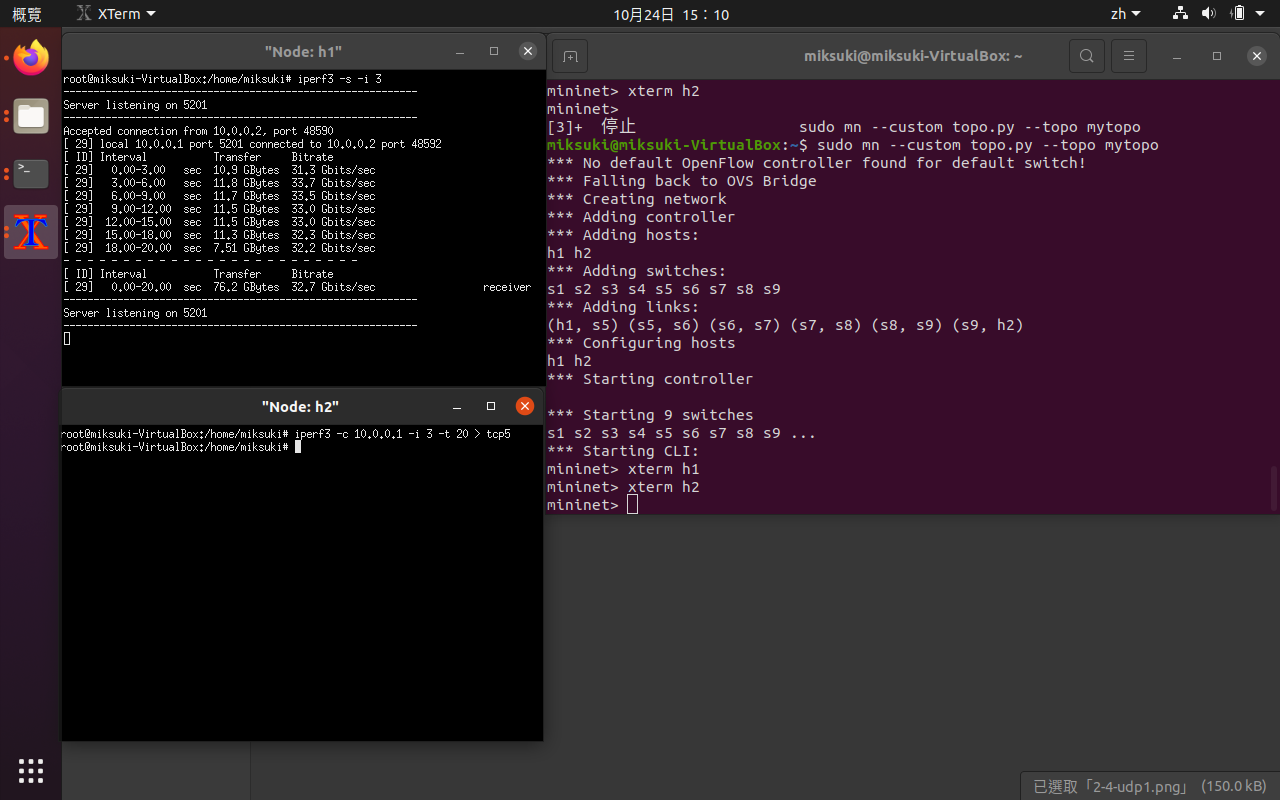
1個節點的 TCP



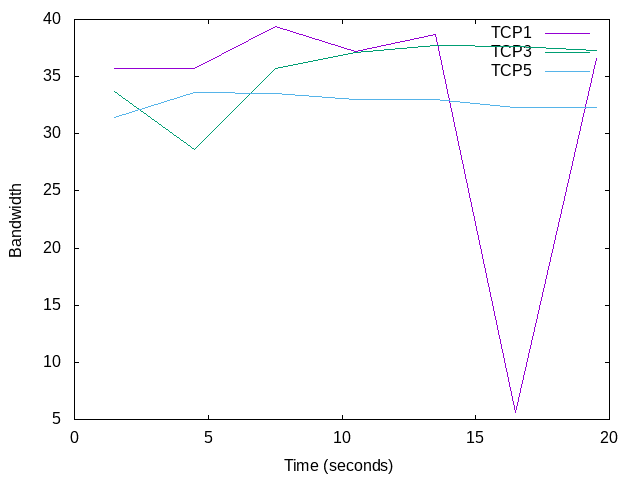
3個節點的 TCP



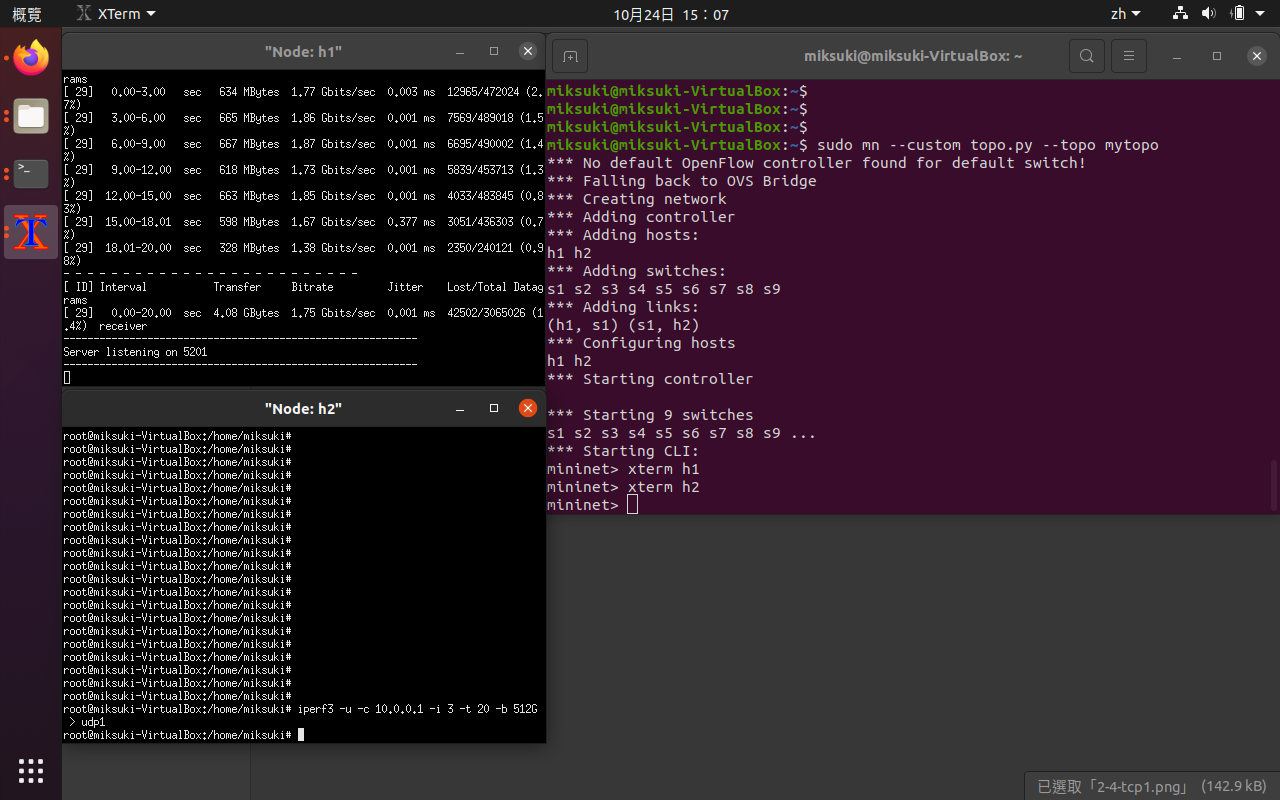
五個節點的 TCP



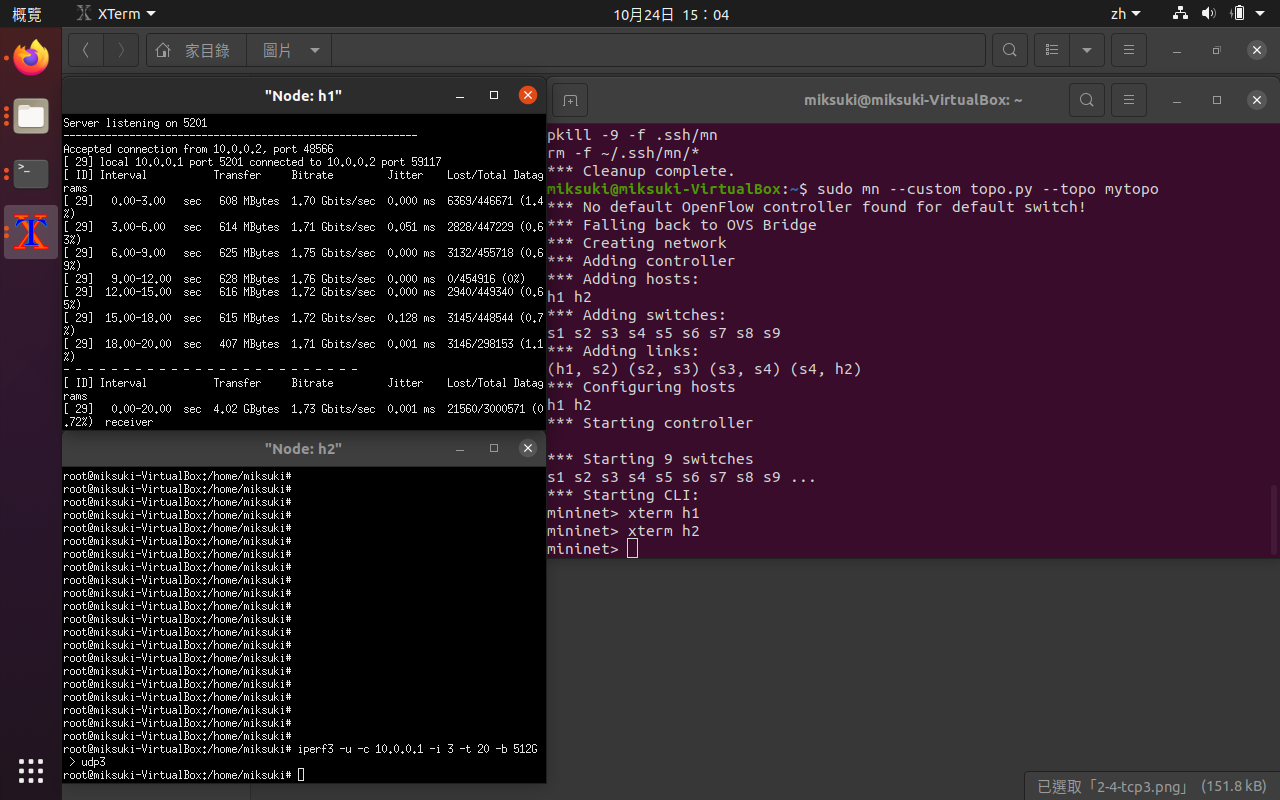
TCP 繪圖結果



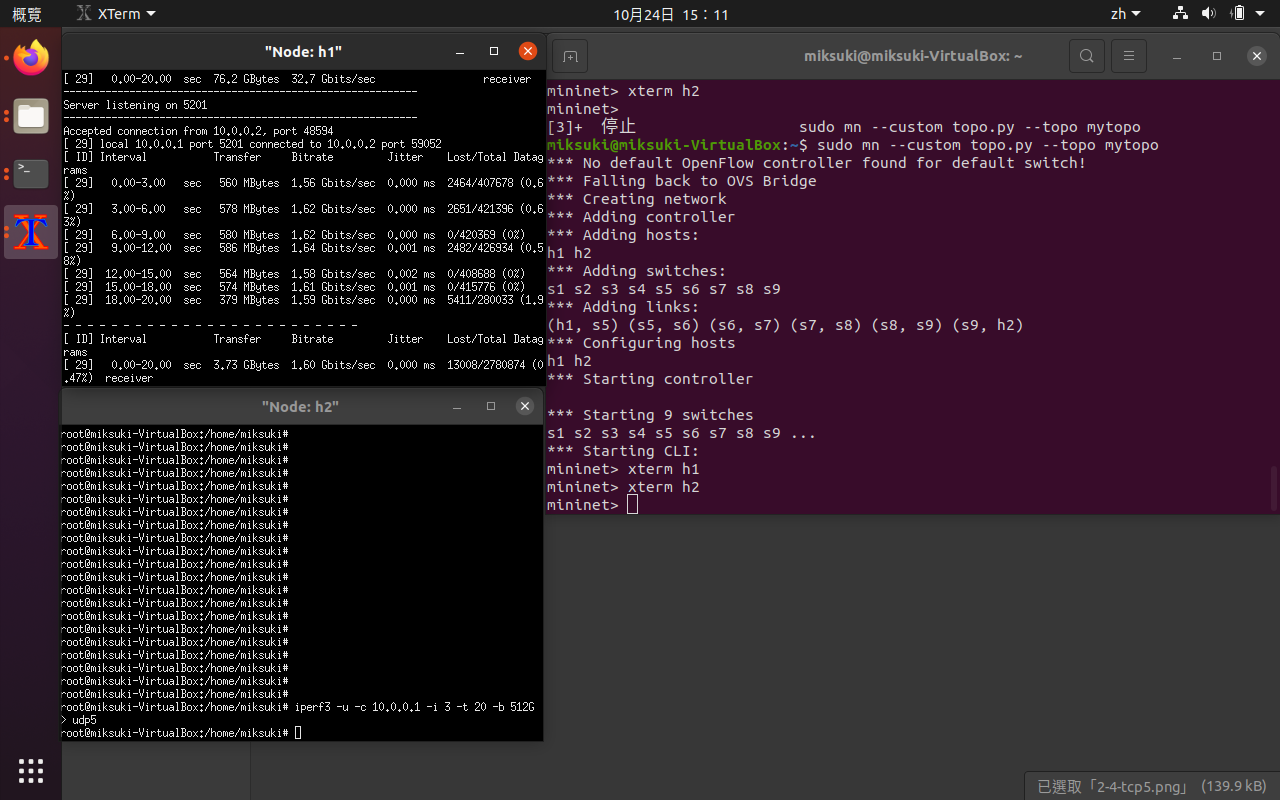
一個節點的 UDP



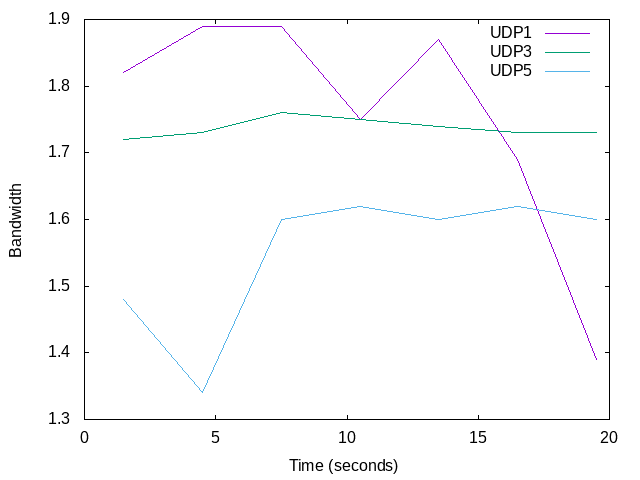
三個節點的 UDP



五個節點的 UDP



UDP 繪圖結果



4.4

buffer size 會影響傳輸的 performance

當 buffer size大的時候，傳輸量增加，但相對的延遲也高

而在 linux kernel 裡，TCP 追求的是高速、大量的資料傳輸，buffer size 大；

UDP 則是追求低延遲傳輸，buffer size 小，

所以會導致在大量傳輸時， TCP 比 UDP 快

# netperf

請利用netperf 完成以下要求【自行使用**mininet** 產生**hosts**，利用**xterm <host>**

指令開啟個別 **host** 視窗，即可進行量測】

* 1. 測量Client 與Server 間的**TCP** 網路效能
  2. 測量Client 與Server 間的**UDP** 網路效能
  3. 請寫出正確的指令並截圖證明。

Server 指令:

netsetver

Client-TCP 指令:

netperf -H 10.0.0.1 -p 12865 -t TCP\_STREAM

Client-UDP 指令:

netperf -H 10.0.0.1 -p 12865 -t UDP\_STREAM

