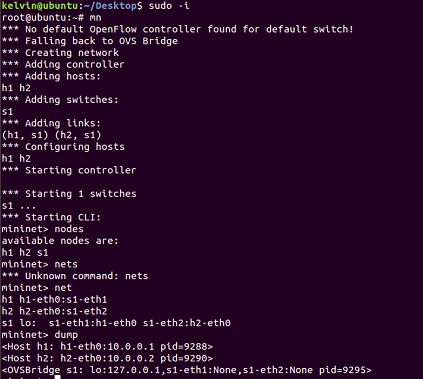
**Part.I**

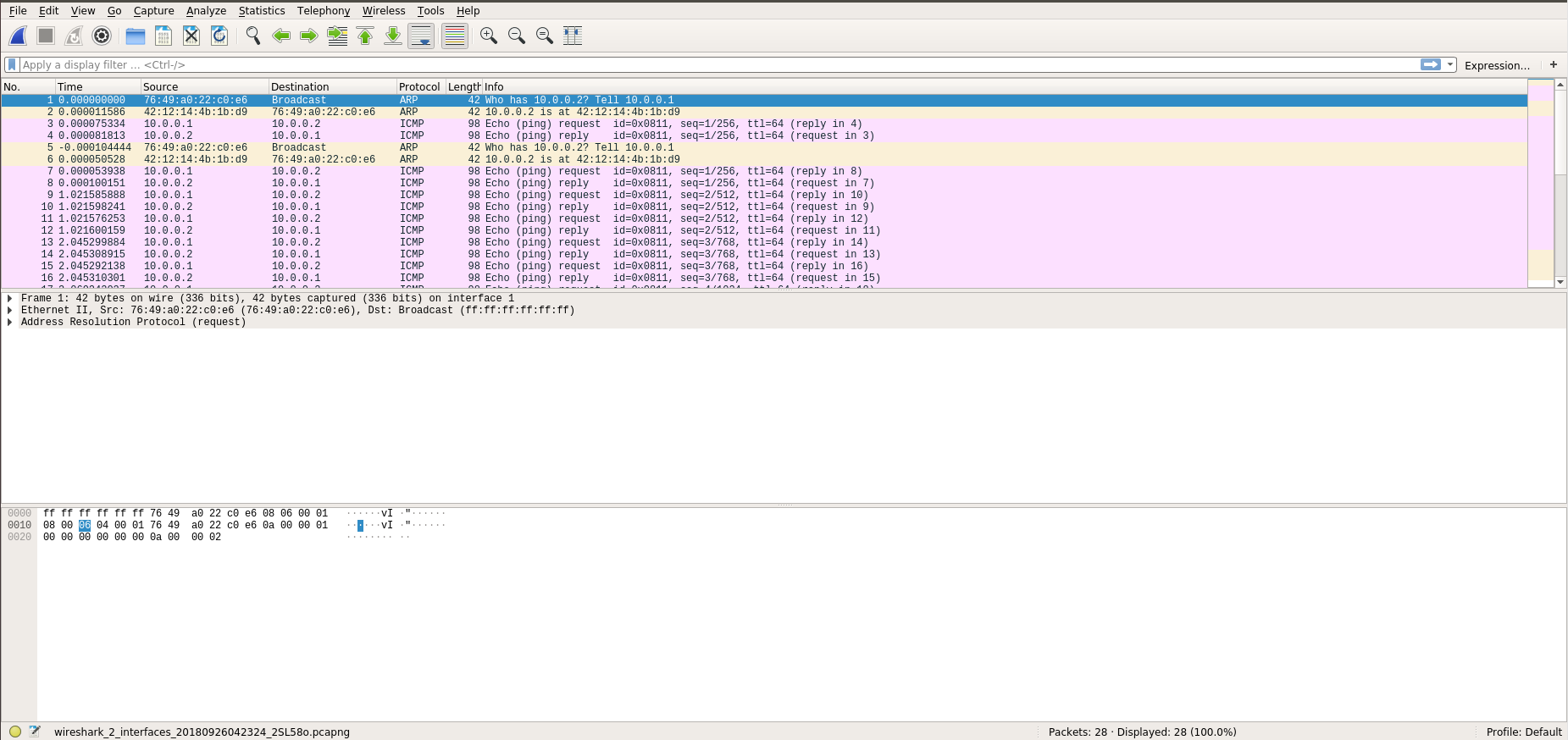
1. 安裝 mininet，可藉由 apt-get install mininet 或其他方式完成安裝。

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。

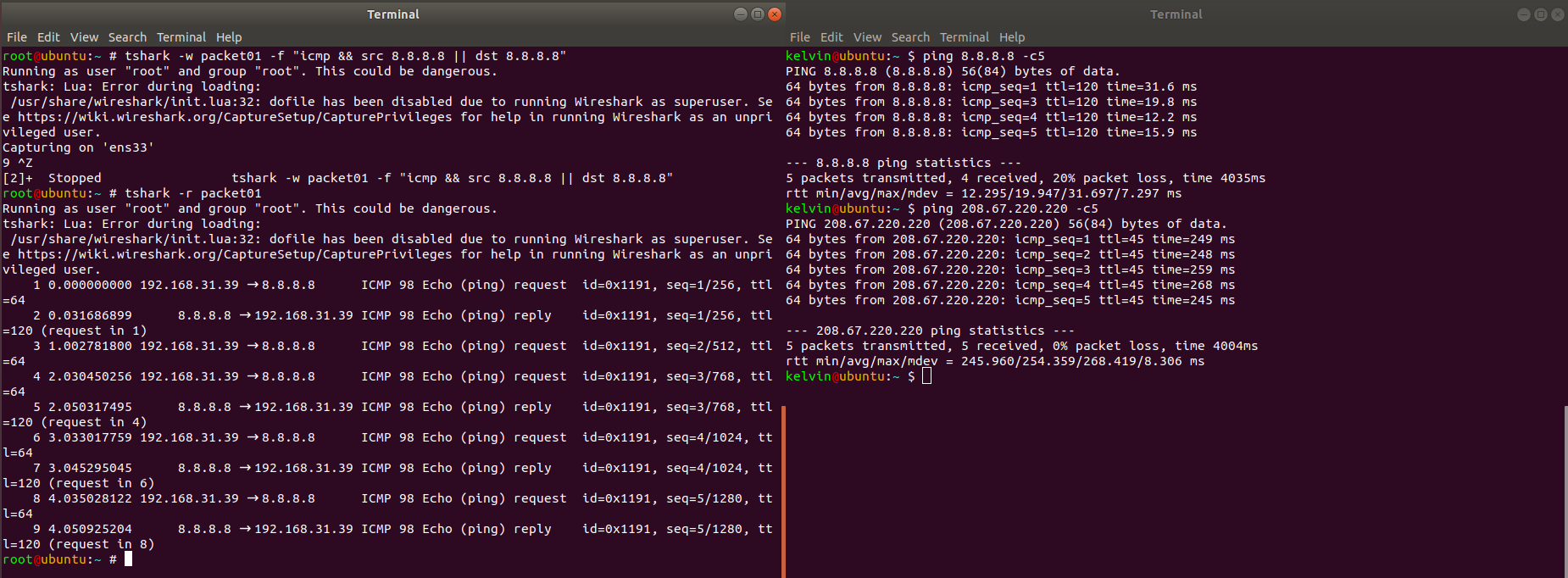


4. 請打開 wireshark，擷取兩個虛擬 host 的網卡，再使用 mininet從h1 ping h2，將你所看到的 wireshark 畫面擷取下來。



**Part.II**

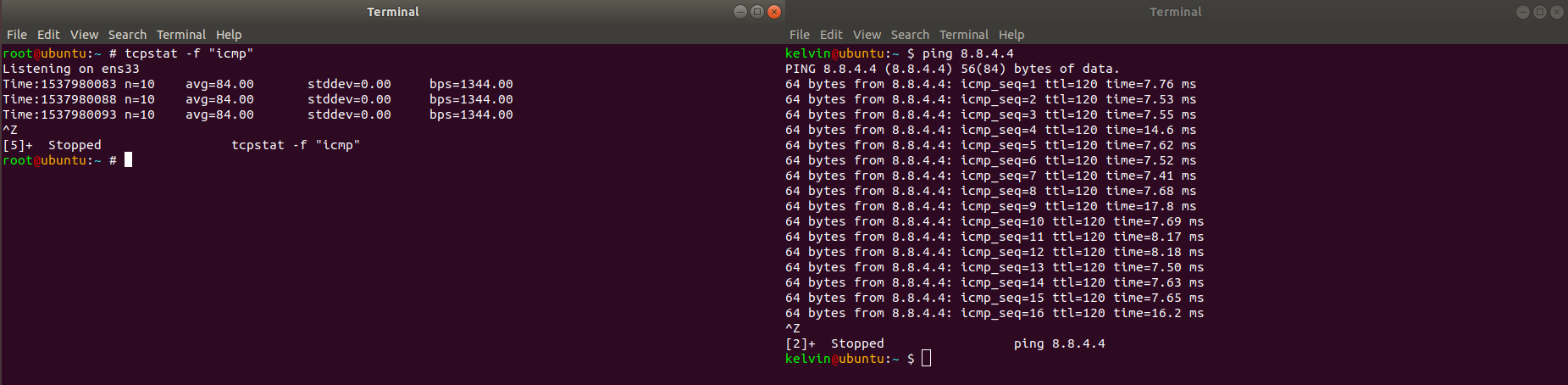
**1. tshark**請使用 tshark 指令搭配正確的參數達到以下的要求：  
a. 抓取”icmp”的封包， 並且來源或目的是”8.8.8.8”  
b. 將擷取的封包儲存為一個檔案， 名稱為”packet01”  
c. 下達正確的指令後，請 ping 8.8.8.8 及 ping 208.67.220.220  
d. 結束後使用 tshark 指令查看擷取的檔案  
e. 以上請寫出正確的指令並截圖證明



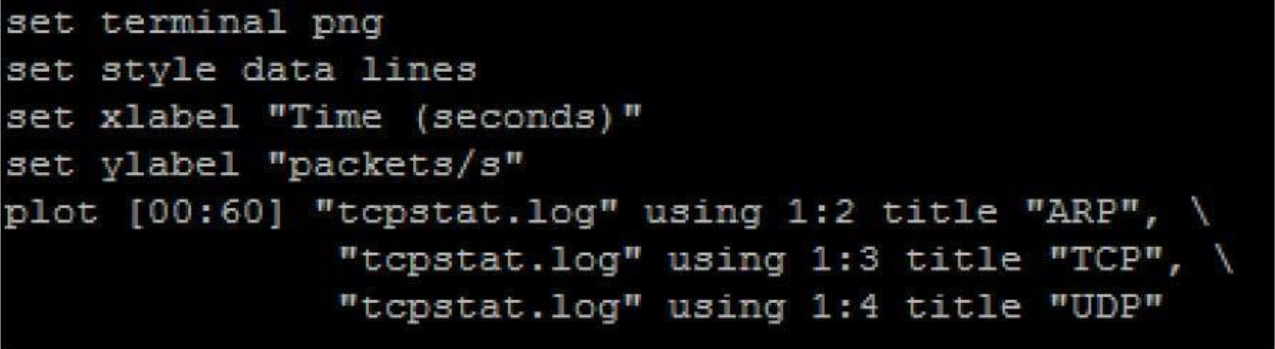
截取指令(root)：tshark -w packet01 -f "icmp && host 8.8.8.8"

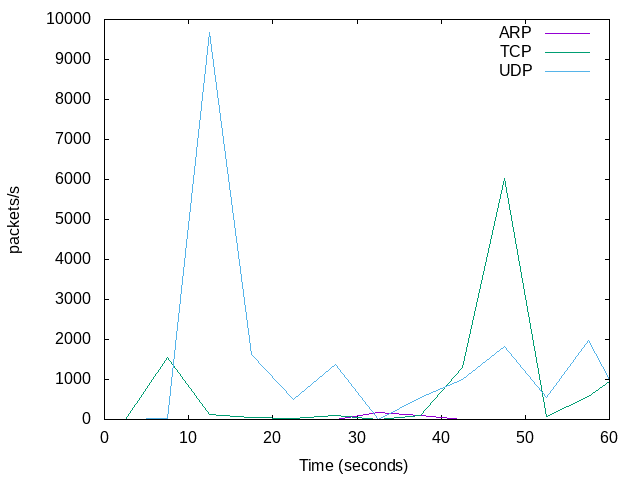
讀取指令(root)：tshark -r packet01

2. tcpstat請使用 tcpstat 指令搭配正確的參數達到以下的要求：  
a. 抓取”icmp”的封包  
b. ping 任意位址  
c. 完成後中斷 tcpstat，將顯示的結果截圖，並寫出正確的指令



指令：tcpstat –f “icmp”

3. tcpdump & tcpstat & gnuplot  
a. 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.dmpb. 開啟瀏覽器瀏覽網頁約一分鐘  
c. 中斷 tcpdump  
d. 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.loge. 使用 vim 寫一個 script， 名稱為”script1”， script 內容如下圖

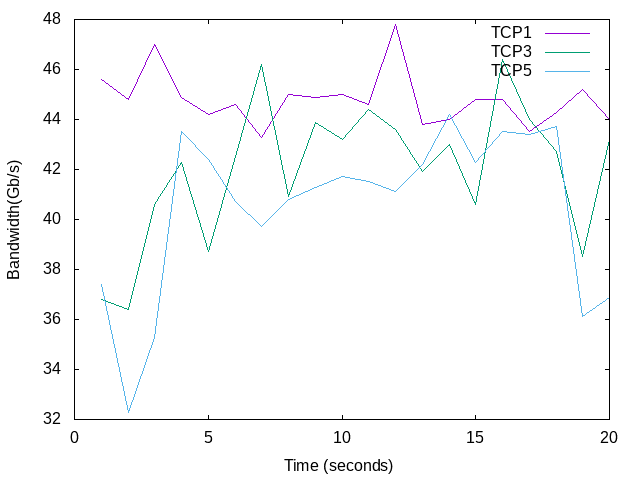
f. 利用 gnupolt 繪圖，產生如下的圖表  
gnuplot script1 > graph1.png

4.mininet & iperf & gnuplot4-1. 請說明 iperf 的用途，以及在什麼情況下你會需要使用它？

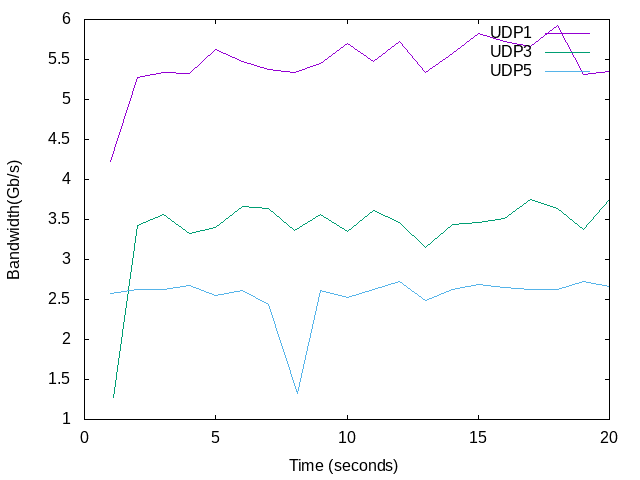
iperf是一個測試網路性能的工具，可以測試TCP和UDP的頻寬性能，可以根據需要調整參數，根據結果回報頻寬、延遲、封包丟失等資料。

4-2. 請在 mininet 下設計 topology，使用 iperf3 指令開啟 Server 和 Client，使Server 及 Client 之間有 0~4 個節點，並測量使用 TCP 及 UDP 傳輸時不同數量節點的頻寬變化，將結果存成檔案。

**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。

TCP:

UDP:



**4-4.** 請說明 TCP 及 UDP 產生結果差異的原因。

從兩個結果皆可看出節點越多頻寬會有些微的下降，UDP下降的幅度更是顯著，而UDP頻寬遠低於TCP的原因在於，UDP並不會確認封包是否送達，而伺服器將掉包視為無效的頻寬，因此沒有送到的部分將會變成頻寬的浪費，導致最後頻寬無法上升。

**5. 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】

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. 請寫出正確的指令並截圖證明

server: netserver

client(TCP): netperf –H 10.0.0.1 –l 20

client(UDP): netperf –t UDP\_STREAM –H 10.0.0.1 –l 20

