

電網導 HW2 report

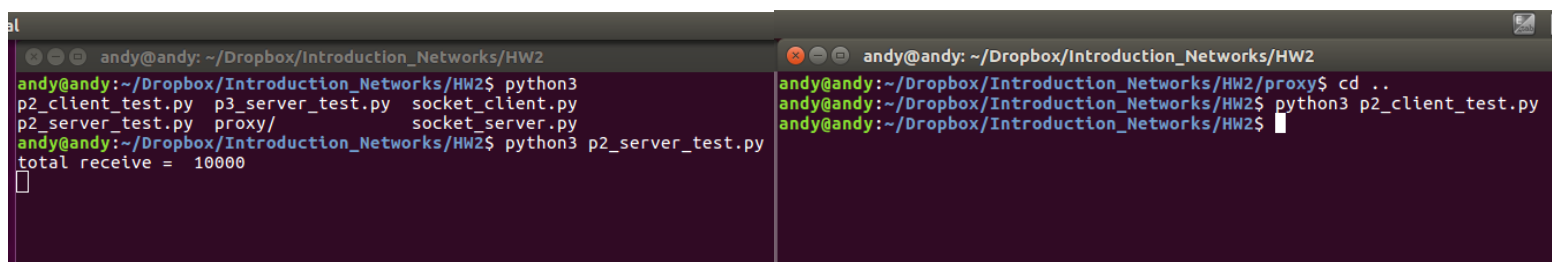
電機三 陳建良 B04901071

1. [Socket Programming - TCP]

```
andy: ~/Dropbox/電網導/HW2
File Edit View Search Terminal Help
andy@andy:~/Dropbox/電網導/HW2$ python3 socket_client.py
Welcome to HW2 P1 Local Server. Please give me your identity. What's your name?
陳建良
What's your student ID?
b04901071
What's your favorite food?
炸豬排飯
Hi 陳建良, your student id is B04901071. And you love 炸豬排飯. Is it correct? (Y/N)
Y
Thanks. Your response has been recorded. Please remeber to print-screen this execution, and have a nice day! (Session End)
```

2. [Socket Programming - UDP]

(a)



```
andy@andy: ~/Dropbox/Introduction_Networks/HW2
andy@andy:~/Dropbox/Introduction_Networks/HW2$ python3
p2_client_test.py p3_server_test.py socket_client.py
p2_server_test.py proxy/ socket_server.py
andy@andy:~/Dropbox/Introduction_Networks/HW2$ python3 p2_server_test.py
total receive = 10000

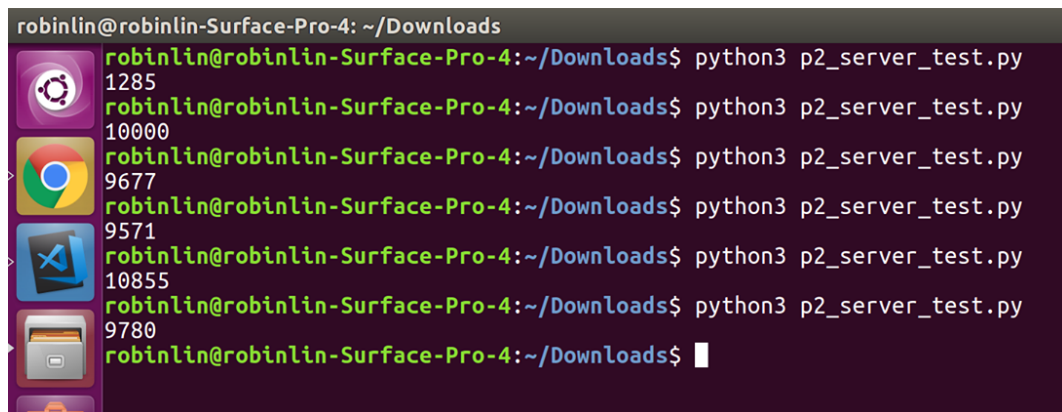
```

```
andy@andy: ~/Dropbox/Introduction_Networks/HW2
andy@andy:~/Dropbox/Introduction_Networks/HW2/proxy$ cd ..
andy@andy:~/Dropbox/Introduction_Networks/HW2$ python3 p2_client_test.py
andy@andy:~/Dropbox/Introduction_Networks/HW2$
```

如圖所示，左邊是 server，右邊是 client，10000 個封包都收到了。

實驗方式很簡單，就是 client 端用 for loop 傳 10000 次相同的訊息，另一方面，server 端每收到這樣的訊息就計數 1 次，如此重複 10000 次。另外為了避免傳送頻率太高，我在封包之間加了一點延遲。

(b)



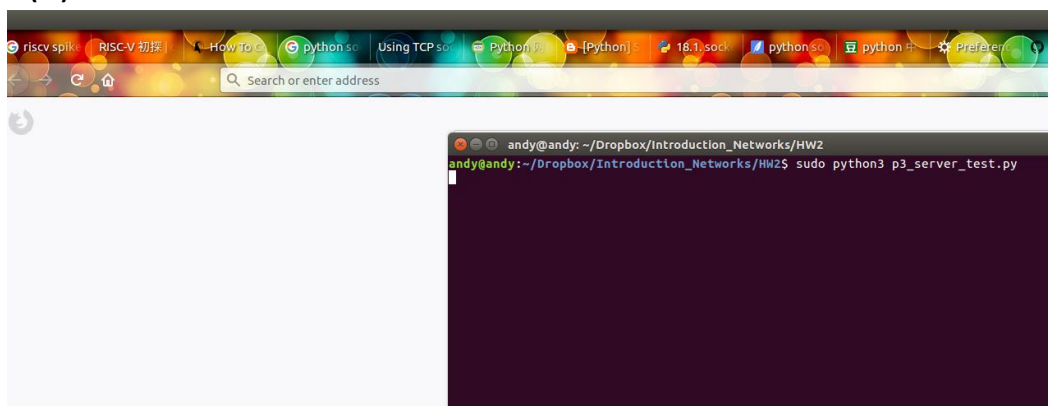
```
robinlin@robinlin-Surface-Pro-4: ~/Downloads
robinlin@robinlin-Surface-Pro-4:~/Downloads$ python3 p2_server_test.py
1285
robinlin@robinlin-Surface-Pro-4:~/Downloads$ python3 p2_server_test.py
10000
robinlin@robinlin-Surface-Pro-4:~/Downloads$ python3 p2_server_test.py
9677
robinlin@robinlin-Surface-Pro-4:~/Downloads$ python3 p2_server_test.py
9571
robinlin@robinlin-Surface-Pro-4:~/Downloads$ python3 p2_server_test.py
10855
robinlin@robinlin-Surface-Pro-4:~/Downloads$ python3 p2_server_test.py
9780
robinlin@robinlin-Surface-Pro-4:~/Downloads$
```

這是以同學電腦作為 server 端做的實驗，第一次只有收到 1285 次，這是因為我這次並沒有在封包之間加上延遲，所以 buffer 來不及消化，導致 lost 掉許多封包，之後幾次加上延遲之後有很大的進步，都有收到 9 成 5 以上的封包。

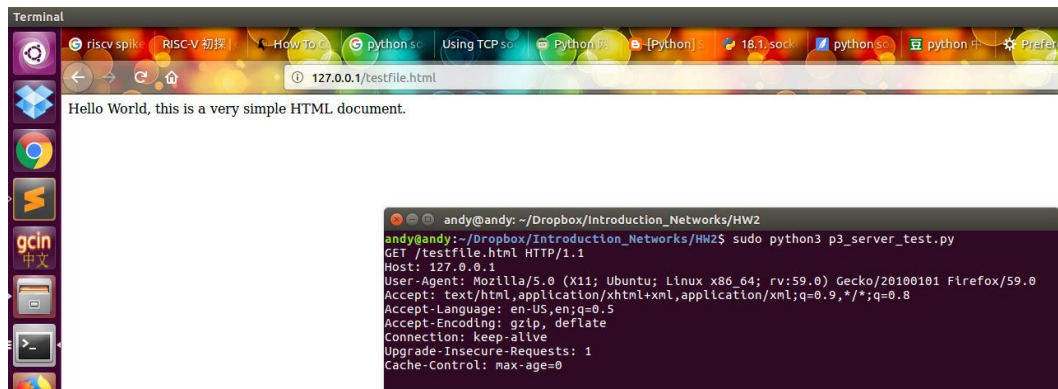
我想這就是 UDP 最大的特徵，因為不需要 handshaking 的關係，如果一直瘋狂不間段的傳送封包，buffer 就會承受不了，這點在有沒有加上時間 delay 可以很明顯的看出。

3. [Web Server]

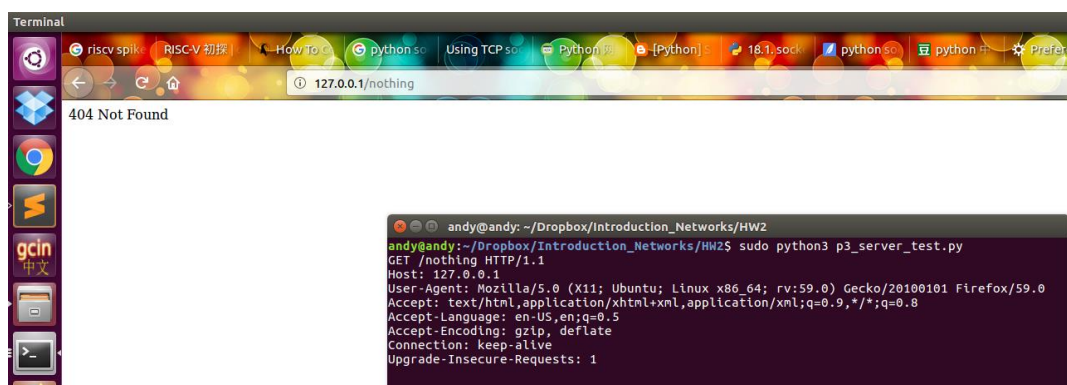
(a)



一開始什麼都沒有，執行 server

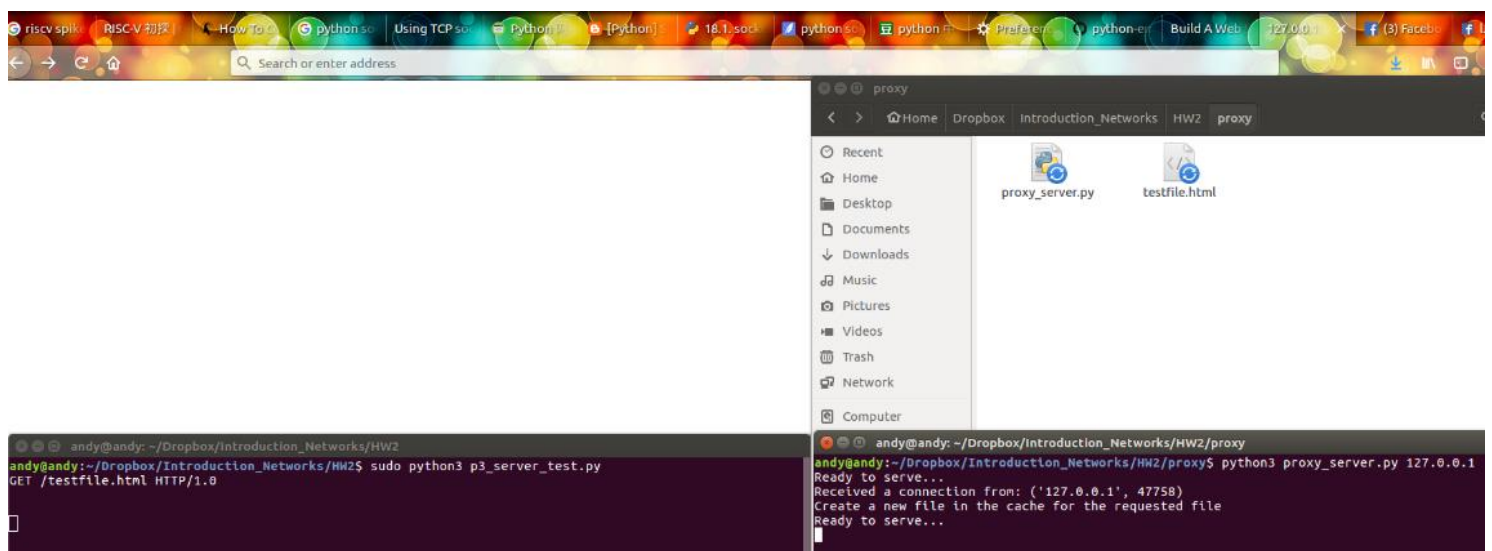


我事先在 server 端資料夾創立 `testfile.html` 來做測試，如圖所示，當瀏覽器要求檔案之後，server 端做出回應並將此 html 檔回傳，內容顯示在瀏覽器

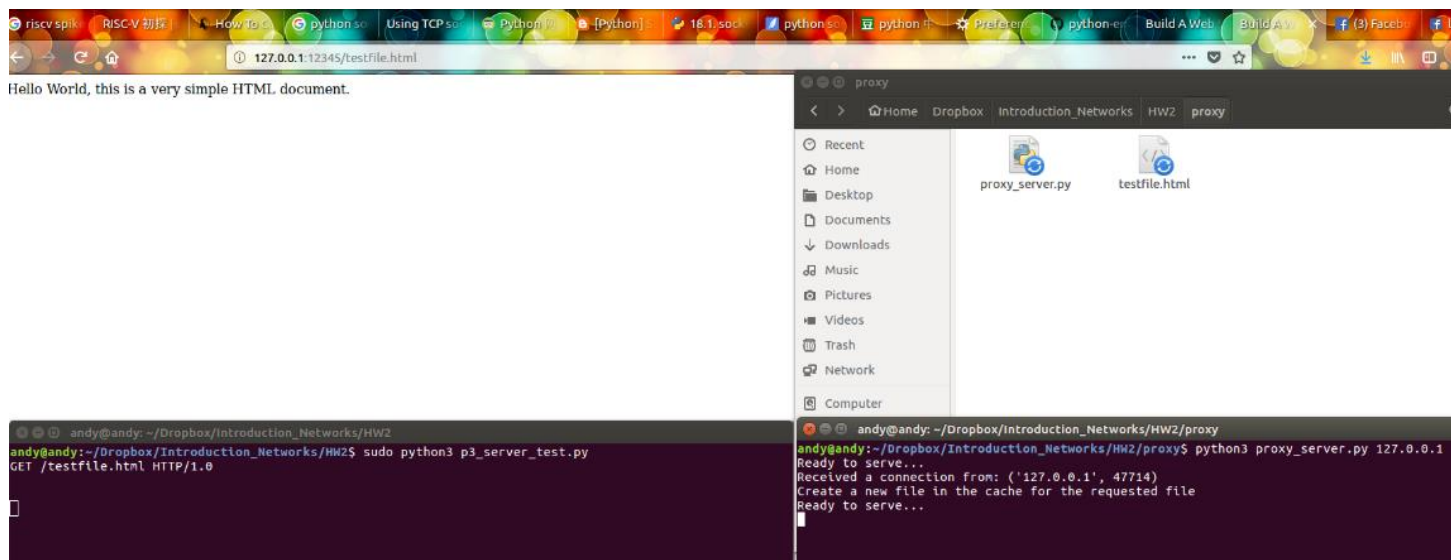


如果要求的檔案並不存在 server 端資料夾，則 server 回傳 404 Not Found

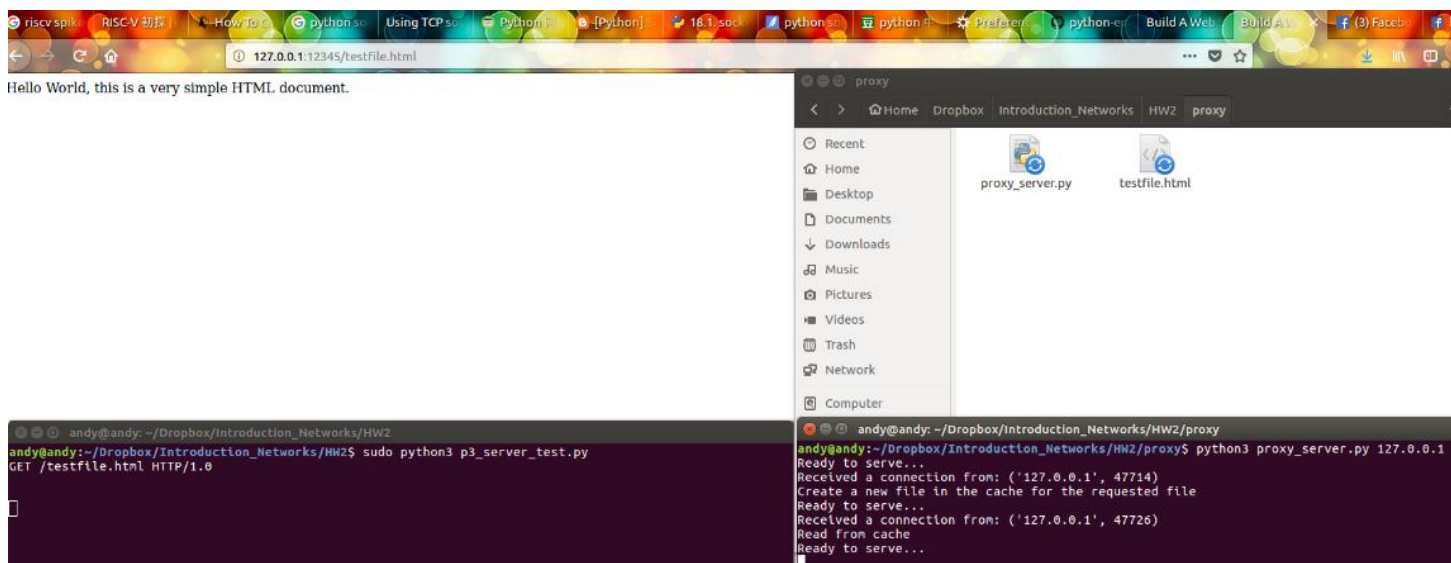
(b)



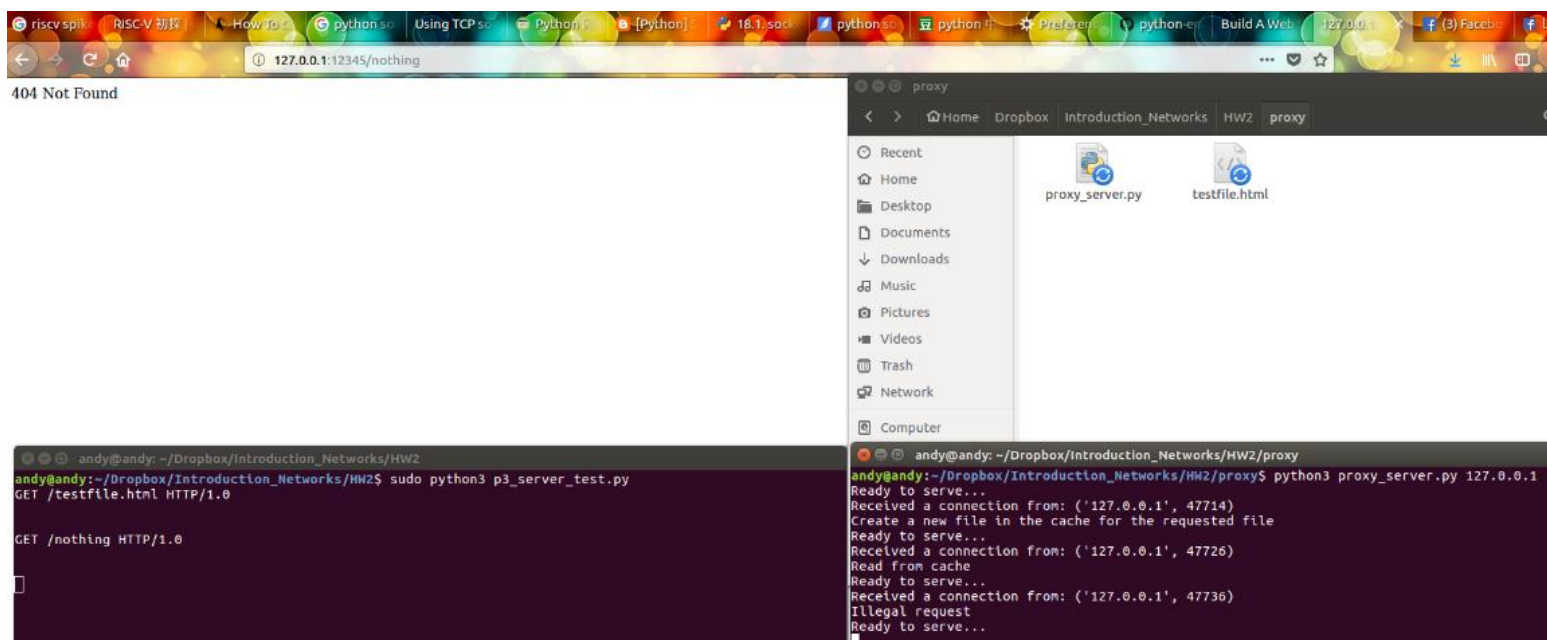
初始狀態，瀏覽器沒有東西，右方是 proxy 的資料夾，下面的 terminal 分別為 proxy 以及 server



瀏覽器向 proxy 要求 testfile.html，這個檔案已經存在在真正的 server 端，而 proxy 的 cache 並沒有，所以 proxy 向 server 要求檔案，server 確認有這個檔案，將內容回傳，proxy 再將內容給瀏覽器，同時建立檔案在自己的 cache(資料夾)，最後顯示在瀏覽器。



瀏覽器向 proxy 要求 testfile.html，這個檔案已經存在 proxy 的 cache，所以 proxy 並不需要向 server 要求檔案，直接將內容回傳，最後顯示在瀏覽器。



瀏覽器向 proxy 要求 nothing，這個檔案不存在，所以 proxy 向 server 要求檔案之後，server 確認並沒有這個檔案，回傳 Not Found，proxy 再將內容給瀏覽器，最後顯示在瀏覽器。此時 proxy 並不需要建檔在 cache