

## ● http/2.0 drop condition

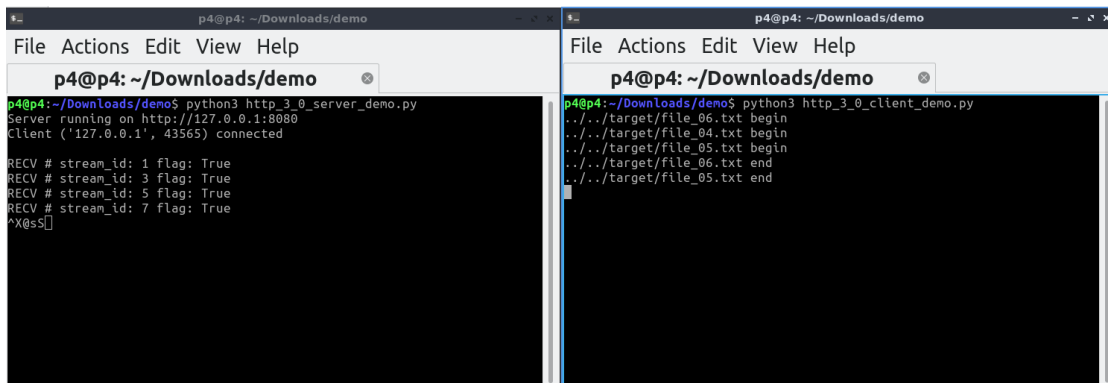
h1 裡面只有收到第一次的 request 和第一個 file 的 request 有收到，後面的 request 收不到



```
"Node: h1"
root@p4:/home/p4/tutorials/exercises/http_drop# python3 hw6/http_2_0_server_demo.py
Server listening on 10.0.1.1:8080
recv stream_id: 1 type 1
recv stream_id: 3 type 1
```

## ● http/3.0 drop condition

儘管 stream\_id = 5 的 request drop 掉了，其他兩個檔案依舊可以完成



```
p4@p4: ~/Downloads/demo
File Actions Edit View Help
p4@p4: ~/Downloads/demo
p4@p4:~/Downloads/demo$ python3 http_3_0_server_demo.py
Server running on http://127.0.0.1:8080
Client ('127.0.0.1', 43565) connected
RECV # stream_id: 1 flag: True
RECV # stream_id: 3 flag: True
RECV # stream_id: 5 flag: True
RECV # stream_id: 7 flag: True
^X@ss[]

p4@p4: ~/Downloads/demo
File Actions Edit View Help
p4@p4:~/Downloads/demo$ python3 http_3_0_client_demo.py
../target/file_06.txt begin
../target/file_04.txt begin
../target/file_05.txt begin
../target/file_06.txt end
../target/file_05.txt end
```

**Q7:** Describe why there was only one request received by the server. Write your answers in the report. Hints: The socket used is a TCP socket.

**A7:** http2 是基於 TCP 協定傳輸的，在 TCP 協定中前面的封包如果未能成功接收，後面的數據也會無法接收直到失敗的莊況被排除，因此，當 stream\_id = 5 掉包後，後面的 request 也無法傳輸。

**Q8:** Describe the differences between HTTP/3 and HTTP/2 according to experimental results. Write your answer in the report.

**A3:** http3 利用 QUIC 創造多個數據流，讓封包可以在同時間傳送，而不用受到其他數據流的影響，所以即使 stream\_id = 5 的封包丟包，其他的封包依然能正常傳輸。