

P10. 假設你請求一個由一份文件和五張圖片所組成的網頁，該文件的大小為 1Kb，所有圖片的大小都一樣是 50Kb，下載速率為 1Mbps，RTT 是 100ms。請問在下列條件下，獲取完整的網頁需要多長時間？(假設沒有 DNS 名稱需要查詢，而且 HTTP 中請求列和標頭的影響可以忽略)

- 使用串列連線的非永久性 HTTP。
- 使用 2 個平行連線的非永久性 HTTP。
- 使用 6 個串列連線的非永久性 HTTP。
- 一個連線的永久性 HTTP。

$$a = (100 \text{ ms}) \times 6 + (251 \text{ KB} / 125 \text{ KBps})$$

$$= 600 \text{ ms} + 2008 \text{ ms}$$

$$a = 2.608 \text{ 秒}$$

$$= 2608 \text{ ms} \div 2.608 \text{ 秒}$$

$$b : 1 \text{ KB} + (3 \times 50 \text{ KB}) = 151 \text{ KB}$$

$$2 \times 50 \text{ KB} = 100 \text{ KB}$$

$$b = 1.308 \text{ 秒}$$

$$1, RTT + (151 \text{ KB} / 125 \text{ KBps}) = 100 \text{ ms} + 1208 \text{ ms} = 1308 \text{ ms}$$

$$2, RTT + (100 \text{ KB} / 125 \text{ KBps}) = 100 \text{ ms} + 800 \text{ ms} = 900 \text{ ms}$$

$$\max(1, 2) = 1 \div 1308 \text{ ms} = 1.308$$

$$c = (100 \text{ ms}) \times 6 + (251 \text{ KB} / 125 \text{ KBps})$$

$$= 600 \text{ ms} + 2008 \text{ ms}$$

$$c = 2.608 \text{ 秒}$$

$$= 2608 \text{ ms} \div 2.608 \text{ 秒}$$

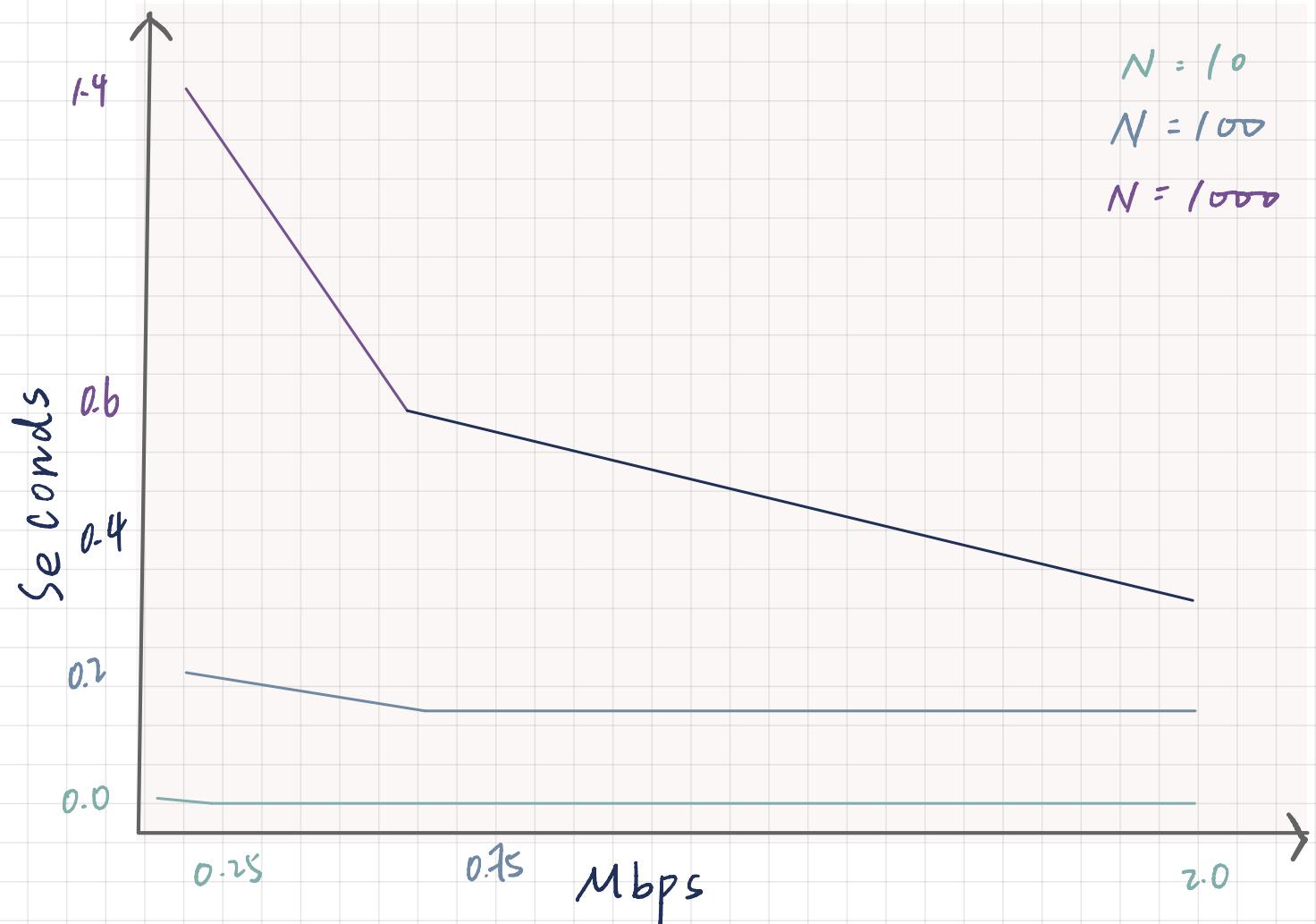
$$d = 100 \text{ ms} + (251 \text{ KB} / 125 \text{ KBps})$$

$$= 100 \text{ ms} + 2008 \text{ ms} = 2108 \text{ ms} \div 2.108 \text{ 秒}$$

J, 11222030
資料 = 9
1 2 3 4 5 6 7 8 9

$$d : 2.108 \text{ 秒}$$

P22. 請考量分享一個 $F=15\text{Gbits}$ 的檔案到 N 個對等點。伺服器的上傳速率為 $u_s = 30 \text{ Mbps}$, 各個對等點的下傳速率為 $d_i = 2 \text{ Mbps}$, 上傳速率則為 u 。針對 $N = 10$ 、 100 和 $1,000$, 以及 $u = 300\text{Kbps}$ 、 700 Kbps 和 2Mbps , 請準備一張圖表, 畫出每一對 N 與 u 的組合在使用用戶端-伺服端散布與 P2P 散布時, 其最小散布時間為何。



$$T_{CS} = \max \left(\frac{F}{u_s}, \frac{F}{d_i} \right)$$

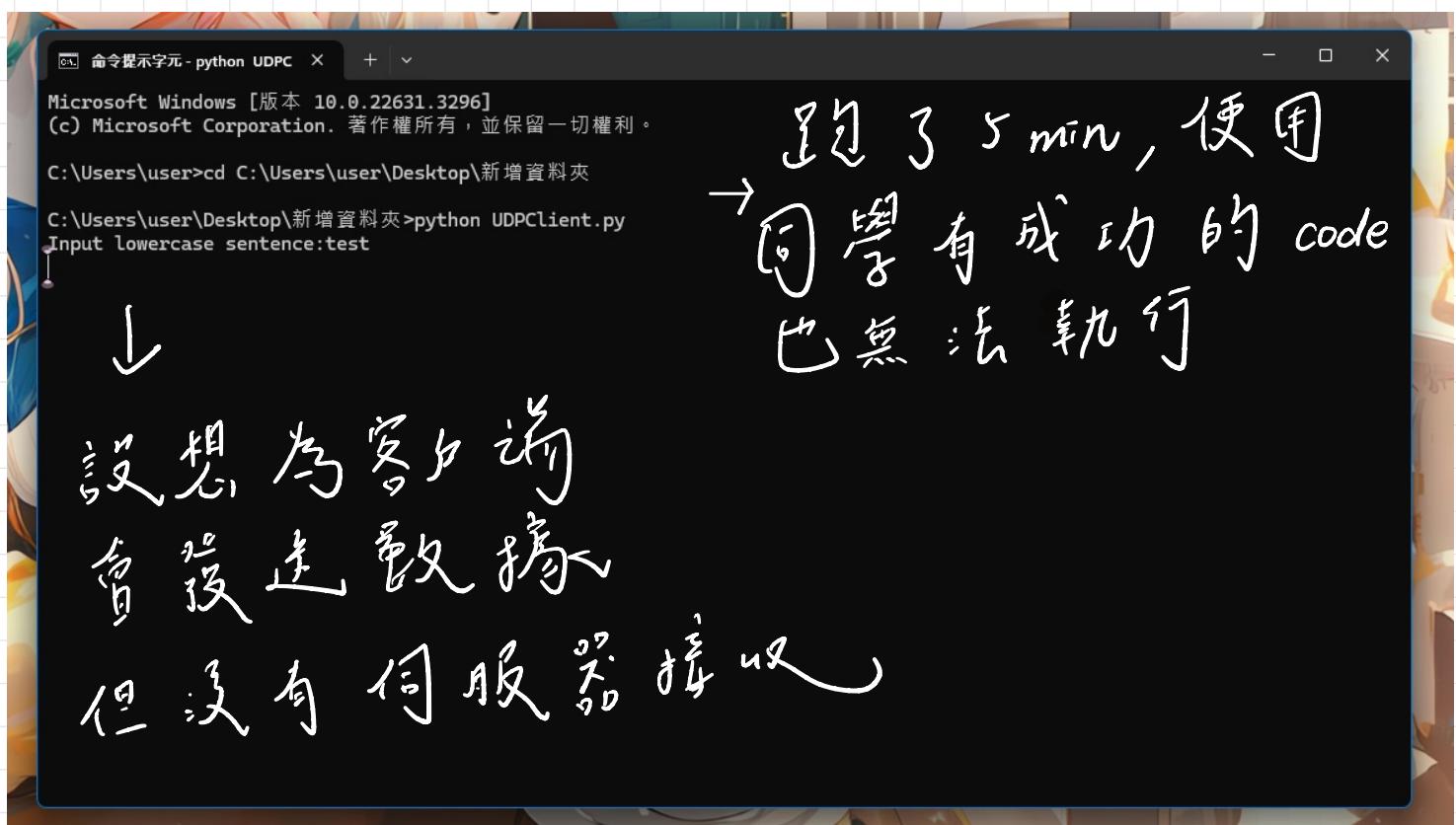
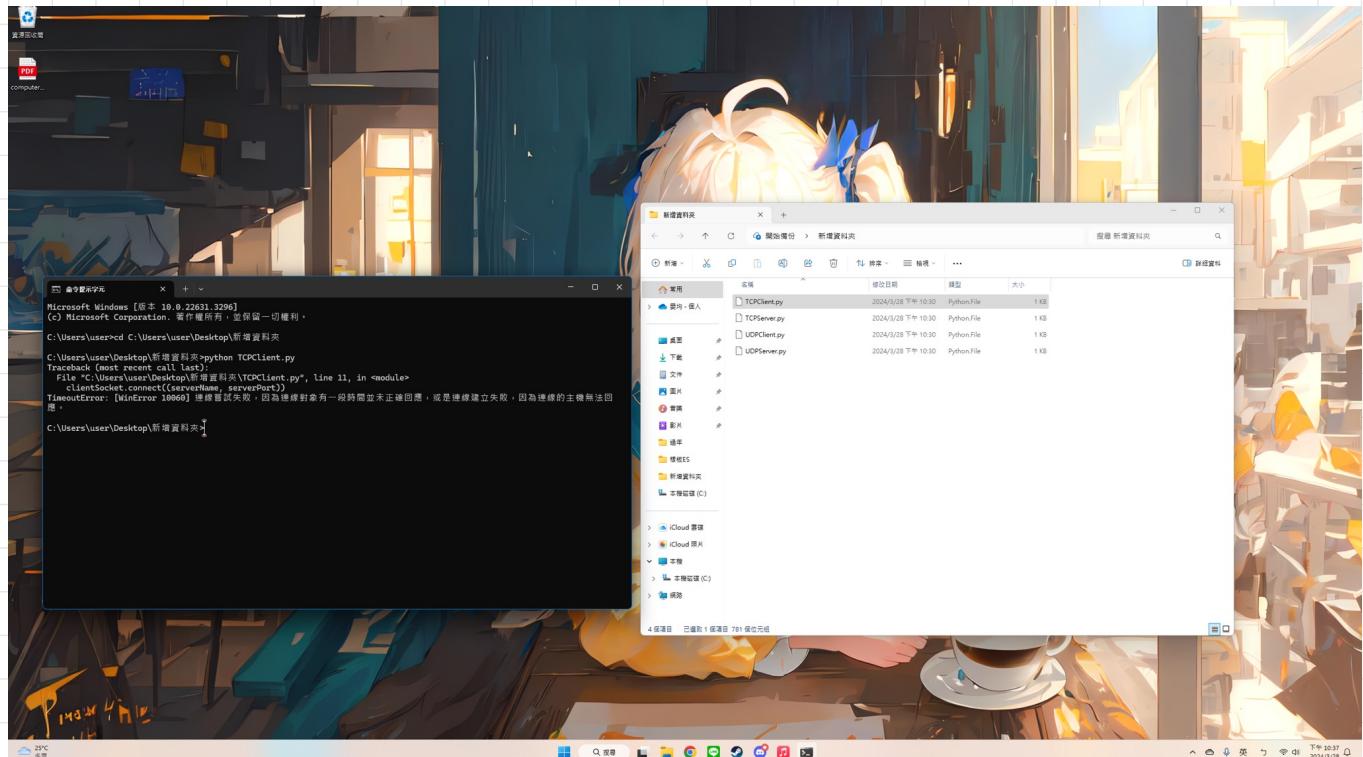
$$T_{P2P} = \frac{F}{\min(u, d_i)}$$

min 散布 time \rightarrow 對等數為 10, 最小散布 time
為 7500 秒

27. 請在一台主機上安裝並編譯 Python 程式 TCPClient 和 UDPClient，並在另一台主機上安裝並編譯 TCPServer 和 UDPServer。

- a. 假設你在執行 TCPServer 之前先執行 TCPClient。請問發生了什麼事？為什麼？
- b. 假設你在執行 UDPServer 之前先執行 UDPClient。請問發生了什麼事？為什麼？
- c. 如果你在用戶端和伺服端使用不同的埠號，會發生什麼事？

a= 會因為伺服器在監聽客戶端嘗試連接
端口而導致錯誤



命令提示字元

```
C:\Users\user>cd C:\Users\user\Desktop\新增資料夾
C:\Users\user\Desktop\新增資料夾>python TCPClient.py
python: can't open file 'C:\\\\Users\\\\user\\\\Desktop\\\\新增資料夾\\\\TCPClient.py': [Errno 2] No such file or directory

C:\Users\user\Desktop\新增資料夾>\python TCPClient
'\\python' 不是內部或外部命令、可執行的程式或批次檔。
```

C:\Users\user\Desktop\新增資料夾>python TCPClient.py

```
Traceback (most recent call last):
  File "C:\\\\Users\\\\user\\\\Desktop\\\\新增資料夾\\\\TCPClient.py", line 6, in <module>
    clientSocket.connect((serverName, serverPort))
TimeoutError: [WinError 10060] 連線嘗試失敗，因為連線對象有一段時間並未正確回應，或是連線建立失敗，因為連線的主機無法回應。
```

C:\Users\user\Desktop\新增資料夾>

客户端與伺服器使用不同端口
導致無法建立連接

TCPClient.py

```
檔案 編輯 檢視
import socket

# 伺服器的位址與埠號
serverName = '192.168.0.80' # 請更改為您實際的伺服器位址
serverPort = 12000 # 請更改為伺服器正在運行的埠號

# 建立一個 TCP 連接的客戶端 socket
clientSocket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

# 連接到伺服器
clientSocket.connect((serverName, serverPort))

# 提示用戶輸入一個小寫句子
message = input('Input lowercase sentence:')

# 將輸入的訊息編碼後發送給伺服器
clientSocket.send(message.encode())

# 從伺服器接收修改後的訊息
modifiedMessage = clientSocket.recv(1024)

# 將修改後的訊息解碼並輸出到終端
print('From Server:', modifiedMessage.decode())

# 關閉客戶端 socket
clientSocket.close()
```

UDPClient.py

```
檔案 編輯 檢視
import socket

# 伺服器埠號
serverName = '192.168.0.80' # 請更改為您實際的伺服器位址
serverPort = 12000 # 請更改為伺服器正在運行的埠號

# 建立一個 UDP 連接的客戶端 socket
clientSocket = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)

# 提示用戶輸入一個小寫句子
message = input('Input lowercase sentence:')

# 將輸入的訊息編碼後發送給伺服器
clientSocket.sendto(message.encode(), (serverName, serverPort))

# 從伺服器接收修改後的訊息及伺服器位址
modifiedMessage, serverAddress = clientSocket.recvfrom(2048)

# 將修改後的訊息解碼並輸出到終端
print('From Server:', modifiedMessage.decode())

# 關閉客戶端 socket
clientSocket.close()
```

UDPServer.py

```
檔案 編輯 檢視
import socket

# 伺服器埠號
serverPort = 12000

# 建立一個 UDP 伺服器 socket
serverSocket = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)

# 將伺服器 socket 綁定到空白位址和指定的埠號
serverSocket.bind(('', serverPort))

# 印出伺服器已經準備好接收訊息
print('The server is ready to receive')

while True:
    # 從客戶端接收訊息及客戶端位址
    message, clientAddress = serverSocket.recvfrom(2048)

    # 將接收到的訊息轉換為大寫
    modifiedMessage = message.decode().upper()

    # 將修改後的訊息發送回客戶端
    serverSocket.sendto(modifiedMessage.encode(), clientAddress)
```

TCPServer.py

```
檔案 編輯 檢視
import socket

# 伺服器埠號
serverPort = 12001

# 建立一個 TCP 伺服器 socket
serverSocket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

# 將伺服器 socket 綁定到空白位址和指定的埠號
serverSocket.bind(('', serverPort))

# 開始監聽連接，參數指定最大等待連接數為 1
serverSocket.listen(1)

# 印出伺服器已經準備好接收訊息
print('The server is ready to receive')

while True:
    # 等待客戶端的連接，accept() 方法會返回一個新的 socket 和客戶端的位址
    connectionSocket, addr = serverSocket.accept()

    # 從客戶端接收訊息
    message = connectionSocket.recv(1024)

    # 將接收到的訊息轉換為大寫
    modifiedMessage = message.decode().upper()

    # 將修改後的訊息發送回客戶端
    connectionSocket.send(modifiedMessage.encode())

    # 關閉與該客戶端的連接
    connectionSocket.close()
```

P31. Netflix 為其CDN採用的伺服器配置策略是什麼？內容如何在不同的伺服器上複製？

Netflix 使用 Open Connect CDN 架構
其中針對每個地區需求配置伺服器。
他們在不同的地點放置 OCAc 的專用
硬體，存儲最愛歡迎的內容副本。
內容自動複製到這些伺服器。
以確保用戶可以流動地觀看內容。

