

分散式系統

Lab: Networking

請務必填寫學號系級姓名，以免成績登錄錯誤。

學號系級姓名:

110703056 資科三 徐宏宇

請依問題與提示在指定區域回答問題，並依規定時間內上傳至 moodle。

第一部份 UDP

1. 請依據課程投影片中的範例，寫作符合下列要求的「UDP Client 與 Server」，其中，Server 請以 Node.js 寫作，Client 請以 Python 寫作。
2. Server 的需求:
 - 以 Node.js 寫作
 - 傾聽本地端(127.0.0.1)的 port 20213，接收 Client 傳來的訊息
 - 在 console 中印出所接收到的訊息
 - 將收到的訊息的最前面加上「XXXX:」再送回給 Client，其中，XXXX 是 Client 的 port number。(client 的 port number 請由 node.js 的 API 取得)
https://nodejs.org/docs/latest/api/dgram.html#dgram_event_message
提示: 使用 server.send 方法要嵌入在 on message 的 call back function 中，才會在收到訊息之後執行。
 - 印出後立即關閉連線(提示: server.close 方法要嵌入在 server.send 方法中)
3. Client 的需求:
 - 以 Python 寫作
 - 連接到本地端的 UDP Server，port 為 20213
 - 送出訊息: 「This is a test from python client」，記得要先將訊息轉為 binary: b"This is....."
 - 在 console 中印出所接收到的回應訊息 (記得用 message.decode('utf-8') 將 binary 轉回文字，正常的話應該會印出 xxxxx:This is a test from python client，而不是 b'xxxxx:This is a test form python client)
 - 收到後關閉連線

請將 Server 的程式碼則在下面:

答:

```
// Import the dgram module for UDP socket operations
const dgram = require('dgram');
// Create a UDP socket for IPv4
const server = dgram.createSocket('udp4');

// Handle errors that occur within the server
server.on('error', (err) => {
  // Print the error stack trace
  console.log(`server error:\n${err.stack}`);
  server.close();
});

// Event listener for receiving messages
server.on('message', (msg, rinfo) => {
  console.log(`server got: ${msg} from ${rinfo.address}:${rinfo.port}`);

  // Prepend the client's port number to the received message
  const message = `${rinfo.port}: ${msg}`;

  // Send the modified message back to the client
  server.send(message, rinfo.port, rinfo.address, () => {
    console.log('Message sent back to client');
    // Close the server after sending the message
    server.close();
  });
});

// Listener for when the server starts listening
server.on('listening', () => {
  // Get the address information of the server
  const address = server.address();
  // Log the listening address and port
  console.log(`server listening ${address.address}:${address.port}`);
});

// Server listens on port 20213
server.bind(20213);
```

請將 Client 的程式碼貼在下面:

答:

```
import socket

# Server address and port
server_address = ('127.0.0.1', 20213)
message = b'This is a test from python client'

# Create a UDP socket
sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)

try:
    # Sending the message
    print('Sending: ', message.decode())
    sent = sock.sendto(message, server_address)

    # Receiving the response
    # buffer size 4096 bytes
    data, server = sock.recvfrom(4096)
    # Decode bytes to string and print the response
    print('Received: ', data.decode())

finally:
    print('closing socket')
    sock.close()
```

第二部份 群播

1. 執行 multicastReceiver.js 與 multicastSender.js，此時，multicastSender 會每 5 秒送一個封包到群播位址，並被 multicastReceiver 所接收
2. 請問它送到那一個群播位址(不含 port)? 所傳送的内容為何?
答: 239.255.255.250, This is a test for networking lab
3. 使用 wireshark，設定 filter 為 udp.port==2391，抓取 multicastSender 送出的封包並觀察它 UDP 的 Length 欄位值為多少? UDP Payload 的大小(值)為多少?
答: 41, 33(bytes)
4. UDP Header 中的 Length 欄位代表什麼意思? 為什麼它會比 UDP Payload 的

大小多 8 bytes?

答:

UDP Length = UDP Header + UDP Payload

UDP Payload (8 bytes) = Source Port (2 bytes) + Destination Port (2 bytes) + Length (2 bytes) + Checksum (2bytes)