**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)

1. 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multicaseSender.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)? 所傳送的內容為何?

答: 239.255.255.250, This is a test for networking lab

1. 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)

1. 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)