컴퓨터 네트워크

HW2

201501462

이선호

1–a)

Client 코드 :

from socket import \*

import time

serverName = '192.168.0.102'

serverPort = 12001

#메세지 수신 실패 횟수

fail\_num = 0

#최대 소요 시간

max\_time = 0.0

#최소 소요 시간

min\_time = 99999.9

#각 ping의 소요 시간

time\_arr = []

#ping의 횟수

pingLength = 10

print('UDP 통신의 경우(%d개의 메세지)'%(pingLength))

for i in range(pingLength):

    try:

        clientSocket = socket(AF\_INET, SOCK\_DGRAM)

        clientSocket.settimeout(1)

        #전송할 메세지

        message = str(i) + "번 ping"

        #ping 시작 시간

        start\_time = time.time()

        clientSocket.sendto(message.encode(), (serverName, serverPort))

        modifiedMessage, serverAddress = clientSocket.recvfrom(2048)

        #ping 소요 시간

        elapsed\_time = time.time() - start\_time

        ms = int(elapsed\_time \* 1000)

        print(serverAddress[0], '의 응답 : ', modifiedMessage.decode(), '시간: ', ms, 'ms')

        time\_arr.append(elapsed\_time)

        if elapsed\_time > max\_time:

            max\_time = elapsed\_time

        if elapsed\_time < min\_time:

            min\_time = elapsed\_time

        clientSocket.close()

    # 서버로부터 응답이 1초 안에 오지 않을 때

    except timeout:

        print('요청시간이 만료되었습니다')

        fail\_num += 1

#실패 확률

fail\_percentage = (fail\_num / pingLength) \* 100

#모든 ping의 소요시간 합

total\_time = 0

for i in range(len(time\_arr)):

    total\_time += time\_arr[i]

print('UDP')

print(serverAddress[0], '에 대한 Ping 통계: ')

print('    패킷: 보냄 = %d, 받음 = %d, 손실 = %d (%0.1f 손실).'%(pingLength, pingLength - fail\_num, fail\_num, fail\_percentage))

print('최대 소요시간 : %0.3fms, 최소 소요시간 : %0.3fms, 평균 소요시간: %0.3fms, 총 소요시간 : %0.3fms'%(max\_time \* 1000, min\_time \* 1000, fail\_percentage, total\_time \* 1000))

server 코드 :

from socket import \*

serverPort = 12001

serverSocket = socket(AF\_INET, SOCK\_DGRAM)

serverSocket.bind(('', serverPort))

print ("The server is ready to receive")

while True:

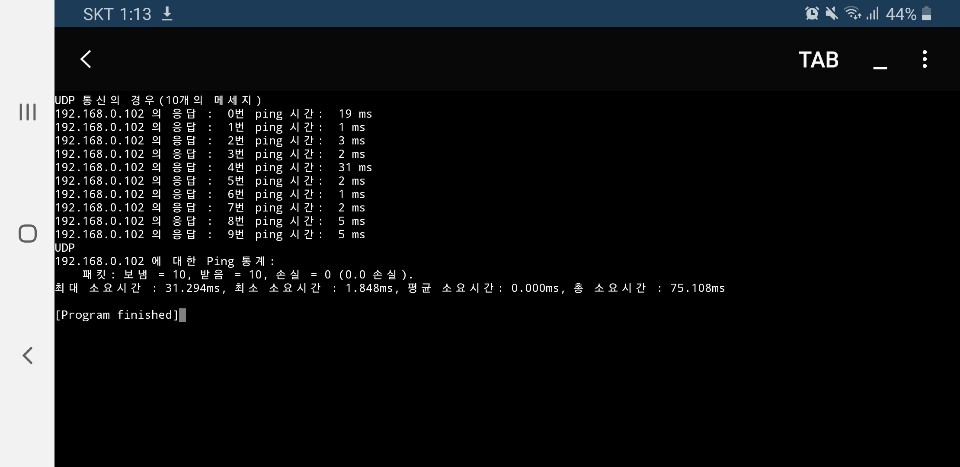
    message, clientAddress = serverSocket.recvfrom(2048)

    decodedMessage = message.decode()

    print("Received from :" + clientAddress[0] + decodedMessage)

    serverSocket.sendto(decodedMessage.encode(), clientAddress)

실행 결과:



1-b)

UDP 방식으로 1000번 수행하는 코드 :

from socket import \*

import time

serverName = '192.168.0.102'

serverPort = 12001

#메세지 수신 실패 횟수

fail\_num = 0

#최대 소요 시간

max\_time = 0.0

#최소 소요 시간

min\_time = 99999.9

#각 ping의 소요 시간

time\_arr = []

#ping의 횟수

pingLength = 1000

print('UDP 통신의 경우(%d개의 메세지)'%(pingLength))

for i in range(pingLength):

    try:

        clientSocket = socket(AF\_INET, SOCK\_DGRAM)

        clientSocket.settimeout(1)

        #전송할 메세지

        message = str(i) + "번 ping"

        #ping 시작 시간

        start\_time = time.time()

        clientSocket.sendto(message.encode(), (serverName, serverPort))

        modifiedMessage, serverAddress = clientSocket.recvfrom(2048)

        #ping 소요 시간

        elapsed\_time = time.time() - start\_time

        ms = int(elapsed\_time \* 1000)

        print(serverAddress[0], '의 응답 : ', modifiedMessage.decode(), '시간: ', ms, 'ms')

        time\_arr.append(elapsed\_time)

        if elapsed\_time > max\_time:

            max\_time = elapsed\_time

        if elapsed\_time < min\_time:

            min\_time = elapsed\_time

        clientSocket.close()

    # 서버로부터 응답이 1초 안에 오지 않을 때

    except timeout:

        print('요청시간이 만료되었습니다')

        fail\_num += 1

#실패 확률

fail\_percentage = (fail\_num / pingLength) \* 100

#모든 ping의 소요시간 합

total\_time = 0

for i in range(len(time\_arr)):

    total\_time += time\_arr[i]

print('UDP')

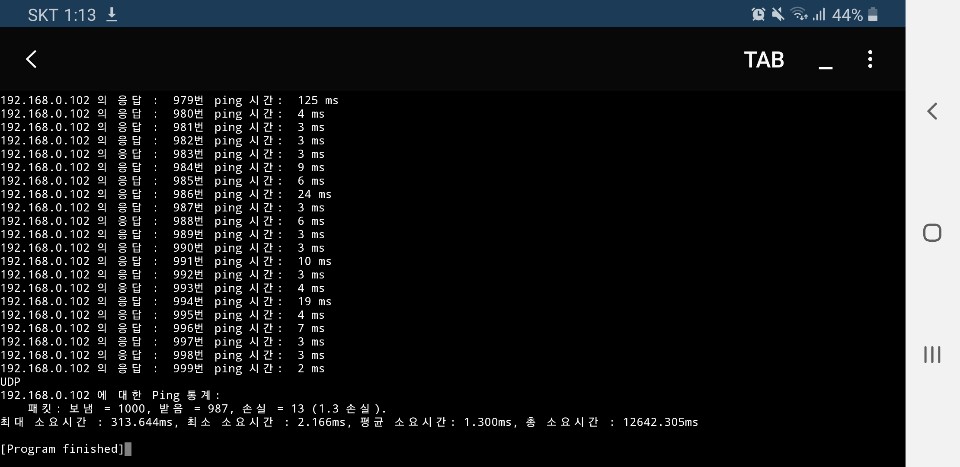
print(serverAddress[0], '에 대한 Ping 통계: ')

print('    패킷: 보냄 = %d, 받음 = %d, 손실 = %d (%0.1f 손실).'%(pingLength, pingLength - fail\_num, fail\_num, fail\_percentage))

print('최대 소요시간 : %0.3fms, 최소 소요시간 : %0.3fms, 평균 소요시간: %0.3fms, 총 소요시간 : %0.3fms'%(max\_time \* 1000, min\_time \* 1000, fail\_percentage, total\_time \* 1000))

server의 코드는 a)와 같다.

수행 결과:



1-c)

TCP 방식으로 b)를 수행하는 client 코드:

from socket import \*

import time

serverName = '192.168.0.102'

serverPort = 12000

#메세지 수신 실패 횟수

fail\_num = 0

#최대 소요 시간

max\_time = 0.0

#최소 소요 시간

min\_time = 99999.9

#각 ping의 소요 시간

time\_arr = []

#ping의 횟수

pingLength = 1000

print('tcp 통신의 경우(%d개의 메세지)'%(pingLength))

for i in range(pingLength):

    try:

        #전송할 메세지

        sentence = str(i) + "번 Ping"

        clientSocket = socket(AF\_INET, SOCK\_STREAM)

        clientSocket.settimeout(1)

        #ping 시작 시간

        start\_time = time.time()

        clientSocket.connect((serverName,serverPort))

        clientSocket.send(sentence.encode())

        modifiedSentence = clientSocket.recv(1024)

        #ping의 소요 시간

        elapsed\_time = time.time() - start\_time

        ms = int(elapsed\_time \* 1000)

        print(serverName, '의 응답 : ', modifiedSentence.decode(), '시간: ', ms, 'ms')

        time\_arr.append(elapsed\_time)

        if elapsed\_time > max\_time:

            max\_time = elapsed\_time

        if elapsed\_time < min\_time:

            min\_time = elapsed\_time

        clientSocket.close()

    # 서버로부터 응답이 1초 안에 오지 않을 때S

    except timeout:

        print('요청시간이 만료되었습니다.')

        fail\_num += 1

#실패 확률

fail\_percentage = (fail\_num / pingLength) \* 100

#모든 ping의 소요시간 합

total\_time = 0

for i in range(len(time\_arr)):

    total\_time += time\_arr[i]

print('TCP')

print(serverName, '에 대한 Ping 통계: ')

print('    패킷: 보냄 = %d, 받음 = %d, 손실 = %d (%0.1f 손실).'%(pingLength, pingLength - fail\_num, fail\_num, fail\_percentage))

print('    최대 소요시간 : %0.3fms, 최소 소요시간 : %0.3fms, 평균 소요시간: %0.3fms, 총 소요시간 : %0.7fms'%(max\_time \* 1000, min\_time \* 1000, (total\_time/len(time\_arr)) \* 1000, total\_time \* 1000))

server 코드 :

from socket import \*

serverPort = 12001

serverSocket = socket(AF\_INET, SOCK\_DGRAM)

serverSocket.bind(('', serverPort))

print ("The server is ready to receive")

while True:

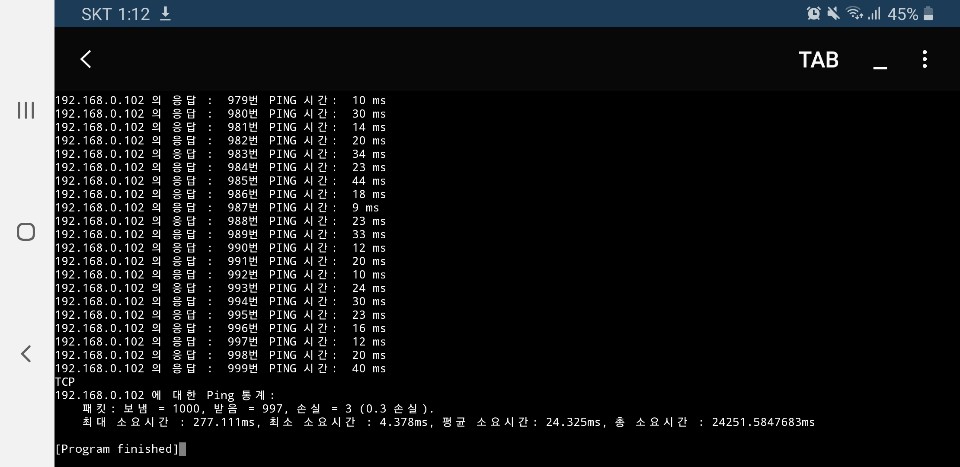
    message, clientAddress = serverSocket.recvfrom(2048)

    decodedMessage = message.decode()

    print("Received from :" + clientAddress[0] + decodedMessage)

    serverSocket.sendto(decodedMessage.encode(), clientAddress)

수행 결과:



TCP와 UDP 간 차이점:

먼저 TCP의 경우 1000개의 메시지 중 총 7개가 손실되었으며, 최대 소요시간은 277.111ms, 최소 소요시간은 4.378ms, 평균적으로 24.325ms가 소요되었다.

반대로 UDP의 경우 1000개의 메시지 중 총 13개가 손실되었으며, 최대 소요시간은 313.644ms이며, 최소 소요시간은 2.166ms, 평균적으로 1.3ms가 소요되었다.

TCP의 경우 최초에 3-way handshaking을 통하여 소켓을 구성한 이후, 해당 소켓을 이용하여 데이터를 전송, 수신한다. TCP의 경우 메시지를 전송한 이후, 일정 대기시간 이내에 메시지에 대한 응답이 도착하지 않을 시 해당 메시지를 다시 요청한다. 이러한 방식을 통하여 신뢰성 있는 메시지 통신을 할 수 있으며, 위의 TCP의 메시지 손실이 3개인데 반해 UDP의 메시지 손실이 13개임을 보아 확인할 수 있다. 그러나 3-way handshaking을 구성하는 시간, 메시지 손실 발생시 해당 메시지를 다시 요청하여 수신 받는 시간들이 추가로 소요되기 때문에 TCP는 UDP와 비교했을 때 평균 소요시간이 24ms로 1.3ms인 UDP보다 시간이 더 걸린다는 사실을 알 수 있다. 이 코드의 경우 1초 이내에 응답이 완료되지 않는 경우 메시지 소실 처리를 하였으며, 3-way handshaking, 메시지 전송간 메시지가 소실되어 해당 메시지를 다시 요청하는 과정을 거치는 동안 1초가 지났을 경우 소실 처리를 하였다. 만약 해당 소실처리를 하지 않는다면 정상적인 네트워크, client, server로 구성된 경우 언젠가 메시지를 정상적으로 수신할 수 있을 것이다.

이와 반대로 UDP의 경우 client가 전송한 메시지가 정상적으로 server에 도착하였는지, server가 전송한 메시지가 정상적으로 client에 도착하였는지 확인하지 않는다. 그에 따라 이 두가지 상황 중 하나가 발생한 메시지는 소실된다. 이러한 이유로 인해 UDP는 비 신뢰성 통신을 하며, 위의 UDP의 메시지 손실이 13개인데 반해 TCP의 메시지 손실이 3개임을 보아 확인할 수 있다. UDP는 사전 소켓 연결이 없으며(3-way handshaking을 하지 않는다) 메시지 전송 이후 그 어떤 처리도 하지 않기 때문에 속도가 매우 빠르다. 이는 UDP의 평균 속도가 1.3ms이지만, TCP의 평균 속도가 24ms임을 보아 확인할 수 있다.

이와 같이 TCP는 UDP에 비해 느리지만 신뢰성 있는 통신을 구성한다

이와 반대로 UDP는 TCP에 비해 빠른 속도를 갖고 있지만, 비 신뢰적 통신을 구성한다.