Assignment 2 computer network 2018007692 lee ji yun

1~3.

Assignment 2.

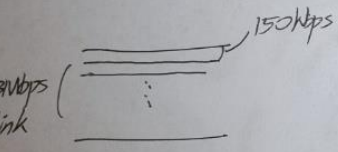
1. The IP address of the destination host and the port number of the destination socket.

2. a) $d_{prop} = m/s$

b) $d_{trans} = L/R$

c) $d_{prop} + d_{trans} = L/R + m/s$ (Ignoring $d_{proc} + d_{queue}$)

3.



a) 20 users (3000kbps/150kbps)

b) n users are transmitting simultaneously, each user transmits only 10percent of the time.

c) $P(N) = {}_{20}C_N \times \left(\frac{1}{10}\right)^N \times \left(\frac{9}{10}\right)^{120-N}$

4. I used Java but it is different like code in ppt. I almost do on my own (searching google,naver..)

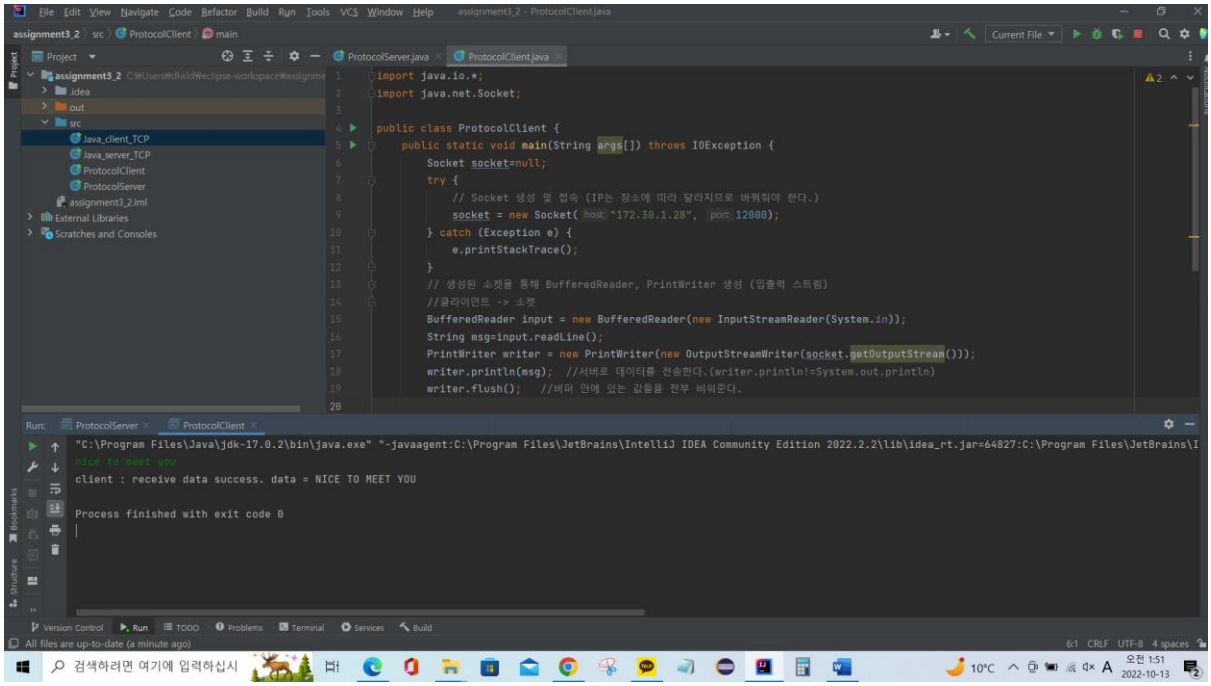
(client code)

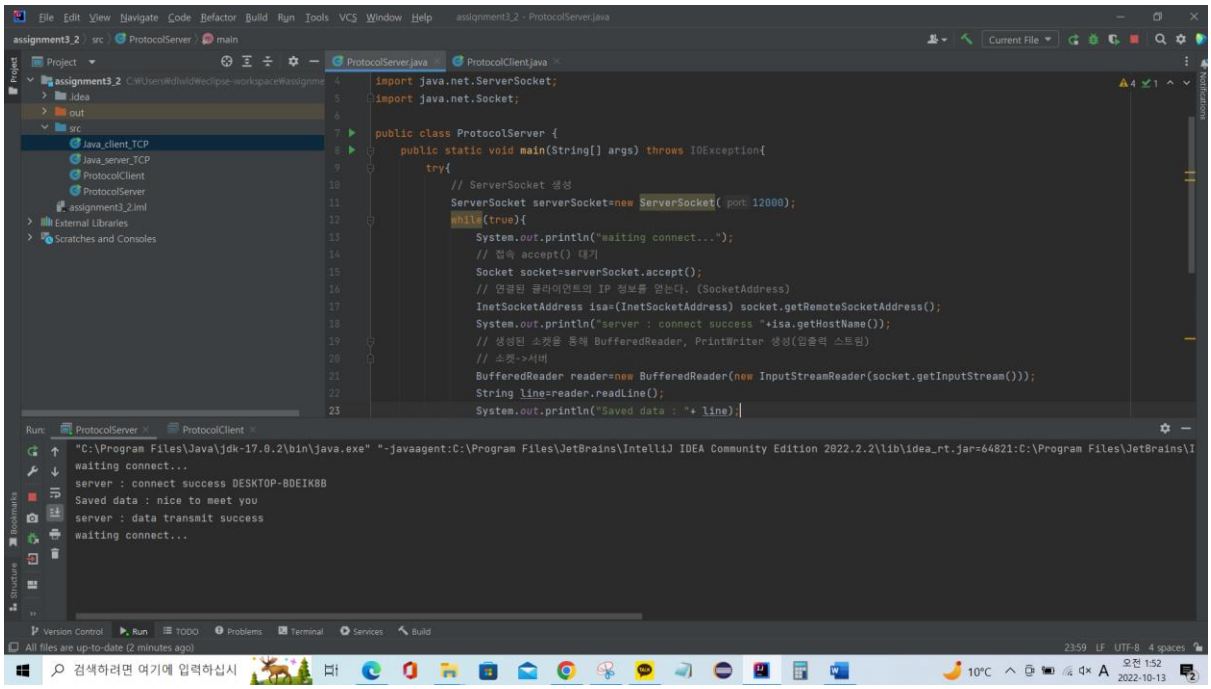
```
import java.io.*;
import java.net.Socket;

public class ProtocolClient {
    public static void main(String args[]) throws IOException {
        Socket socket=null;
        try {
            // Socket 생성 및 접속 (IP는 장소에 따라 달라지므로 바꿔줘야 한다.)
            socket = new Socket("172.30.1.28", 12000);
        } catch (Exception e) {
            e.printStackTrace();
        }
        // 생성된 소켓을 통해 BufferedReader, PrintWriter 생성 (입출력 스트림)
        //클라이언트 -> 소켓
        BufferedReader input = new BufferedReader(new
InputStreamReader(System.in));
        String msg=input.readLine();
        PrintWriter writer = new PrintWriter(new
OutputStreamWriter(socket.getOutputStream()));
        writer.println(msg);  //서버로 데이터를
전송한다.(writer.println!=System.out.println)
        writer.flush();   //버퍼 안에 있는 값들을 전부 비워준다.

        //소켓 -> 클라이언트
        BufferedReader reader = new BufferedReader(new
InputStreamReader(socket.getInputStream()));
        String line = reader.readLine();
        System.out.println("client : receive data success. data = " + line);
        //서버와 통신이 완료되어 서버 출력값을 가지고 온다.

        writer.close();
        reader.close();
    }
}
```

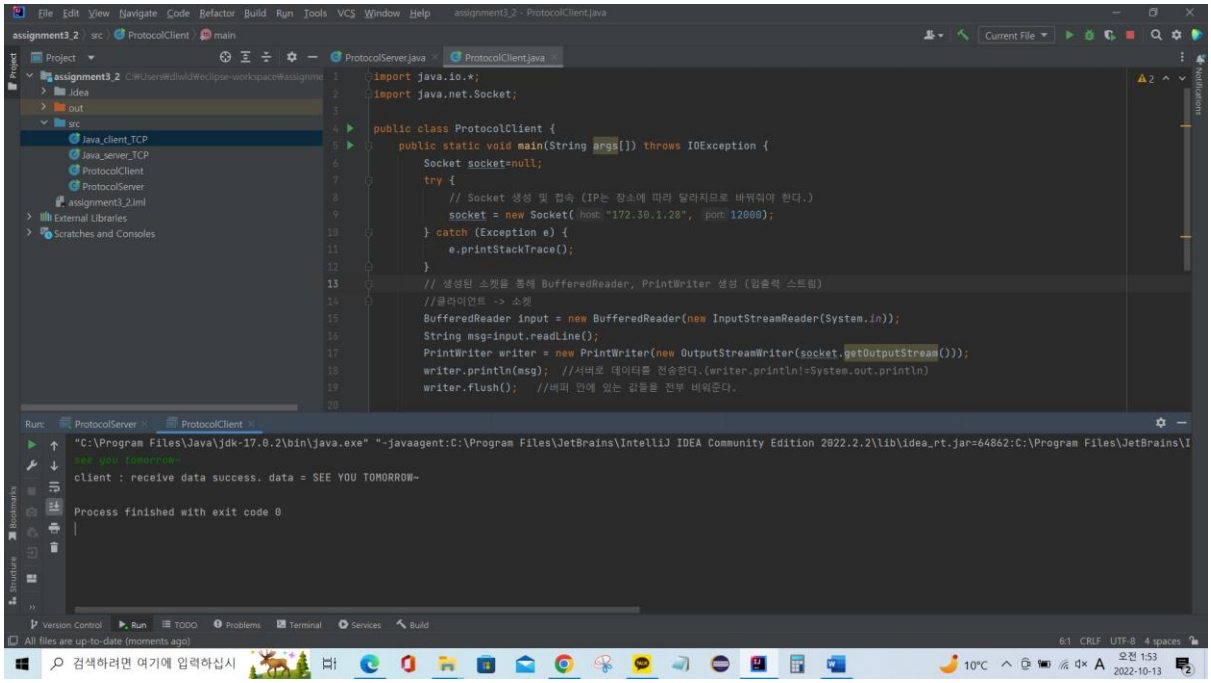
(server code)

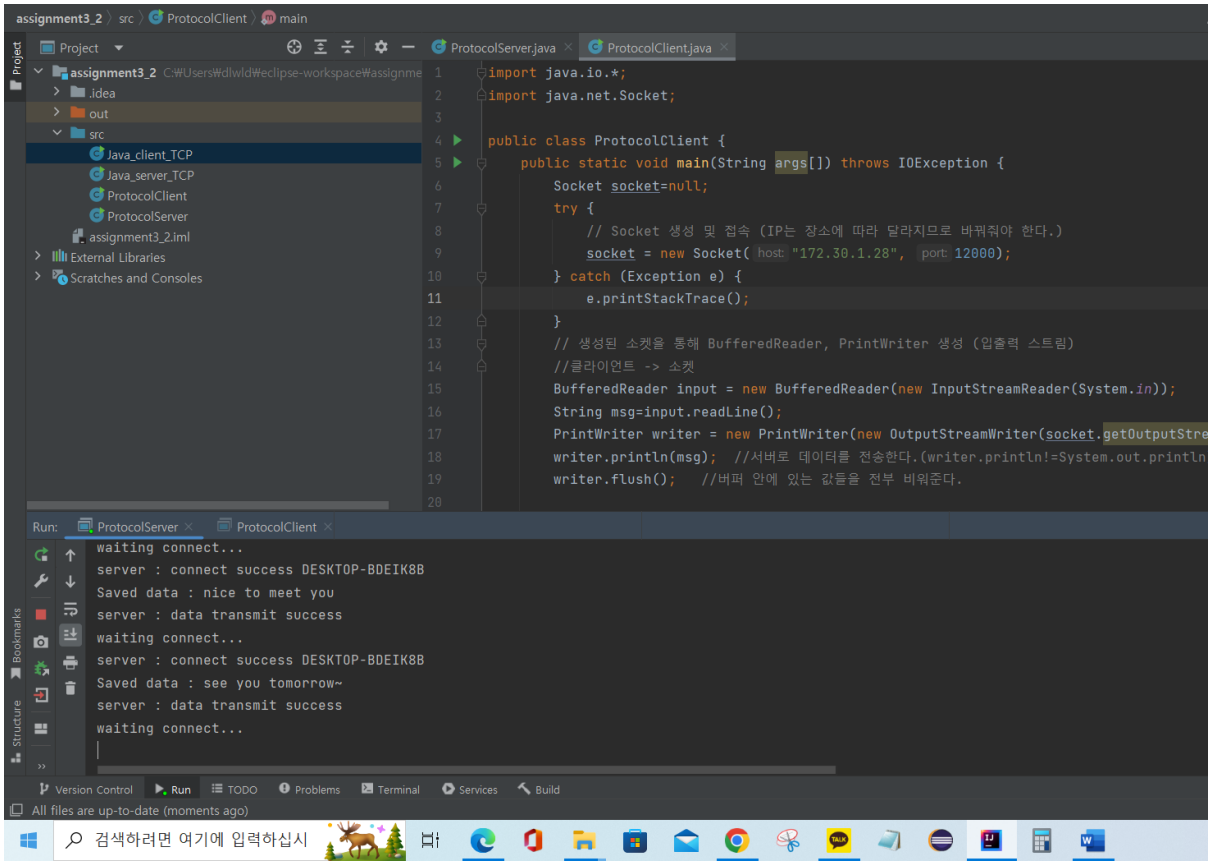
```
import java.net.ServerSocket;
import java.net.Socket;

public class ProtocolServer {
    public static void main(String[] args) throws IOException{
```

```
        try{
            // ServerSocket 생성
            ServerSocket serverSocket=new ServerSocket(12000);
            while(true){
                System.out.println("waiting connect...");
                // 접속 accept() 대기
                Socket socket=serverSocket.accept();
                // 연결된 클라이언트의 IP 정보를 얻는다. (SocketAddress)
                InetSocketAddress isa=(InetSocketAddress)
socket.getRemoteSocketAddress();
                System.out.println("server : connect success
"+isa.getHostName());
                // 생성된 소켓을 통해 BufferedReader, PrintWriter 생성(입출력
스트림)
                // 소켓->서버
                BufferedReader reader=new BufferedReader(new
InputStreamReader(socket.getInputStream()));
                String line=reader.readLine();
                System.out.println("Saved data : "+ line);
                line=line.toUpperCase();
                // 서버 클래스 안의 line 변수 안에 클라이언트의 입력이 들어옴
                PrintWriter writer = new PrintWriter(new
OutputStreamWriter(socket.getOutputStream()));
                // 소켓으로 데이터를 클라이언트로 보냄
                writer.println(line);
                writer.flush();
                System.out.println("server : data transmit success");
            }   // 스트림을 닫지 않기 때문에 close() 대신 flush()를 해준다.
        }catch(IOException e){

        }
    }
}
```


ProtocolClient.java
import java.io.*;
import java.net.Socket;
public class ProtocolClient {
public static void main(String args[]) throws IOException {
Socket socket=null;
try {
// Socket 생성 및 접속 (IP는 장소에 따라 달라지므로 바꿔줘야 한다.)
socket = new Socket(host: "172.30.1.28", port: 12000);
} catch (Exception e) {
e.printStackTrace();
}
// 생성된 소켓을 통해 BufferedReader, PrintWriter 생성 (입출력 스트림)
//클라이언트 -> 소켓
BufferedReader input = new BufferedReader(new InputStreamReader(System.in));
String msg=input.readLine();
PrintWriter writer = new PrintWriter(new OutputStreamWriter(socket.getOutputStream()));
writer.println(msg); //서버로 데이터를 전송한다.(writer.println!=System.out.println)
writer.flush(); //버퍼 안에 있는 값들을 전부 비워준다.
Run: ProtocolServer ProtocolClient
"C:\Program Files\Java\jdk-17.0.2\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2022.2.2\lib\idea_rt.jar=64827:C:\Program Files\JetBrains\I
nice to meet you
client : receive data success. data = NICE TO MEET YOU
Process finished with exit code 0


ProtocolServer.java
import java.net.ServerSocket;
import java.net.Socket;
public class ProtocolServer {
public static void main(String[] args) throws IOException{
try{
// ServerSocket 생성
ServerSocket serverSocket=new ServerSocket(port: 12000);
while(true){
System.out.println("waiting connect...");
// 접속 accept() 대기
Socket socket=serverSocket.accept();
// 연결된 클라이언트의 IP 정보를 얻는다. (SocketAddress)
InetSocketAddress isa=(InetSocketAddress) socket.getRemoteSocketAddress();
System.out.println("server : connect success "+isa.getHostName());
// 생성된 소켓을 통해 BufferedReader, PrintWriter 생성(입출력 스트림)
// 소켓->서버
BufferedReader reader=new BufferedReader(new InputStreamReader(socket.getInputStream()));
String line=reader.readLine();
System.out.println("Saved data : "+ line);
Run: ProtocolServer ProtocolClient
"C:\Program Files\Java\jdk-17.0.2\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2022.2.2\lib\idea_rt.jar=64821:C:\Program Files\JetBrains\I
waiting connect...
server : connect success DESKTOP-BDEIK8B
Saved data : nice to meet you
server : data transmit success
waiting connect...


assignment3_2 〉 src 〉 ProtocolClient 〉 main
ProtocolServer.java
ProtocolClient.java
import java.io.*;
import java.net.Socket;
public class ProtocolClient {
public static void main(String args[]) throws IOException {
Socket socket=null;
try {
// Socket 생성 및 접속 (IP는 장소에 따라 달라지므로 바뀌줘야 한다.)
socket = new Socket(host: "172.30.1.28", port: 12000);
} catch (Exception e) {
e.printStackTrace();
}
// 생성된 소켓을 통해 BufferedReader, PrintWriter 생성 (입출력 스트림)
//클라이언트 -> 소켓
BufferedReader input = new BufferedReader(new InputStreamReader(System.in));
String msg=input.readLine();
PrintWriter writer = new PrintWriter(new OutputStreamWriter(socket.getOutputStream()));
writer.println(msg); //서버로 데이터를 전송한다.(writer.println!=System.out.println)
writer.flush(); //버퍼 안에 있는 값들을 전부 비워준다.
Run: ProtocolServer ProtocolClient
"C:\Program Files\Java\jdk-17.0.2\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2022.2.2\lib\idea_rt.jar=64862:C:\Program Files\JetBrains\I
see you tomorrow~
client : receive data success. data = SEE YOU TOMORROW~
Process finished with exit code 0
Version Control Run TODO Problems Terminal Services Build
All files are up-to-date (moments ago)
검색하려면 여기에 입력하십시
10°C
오전 1:53
2022-10-13


assignment3_2 〉 src 〉 ProtocolClient 〉 main
Project
ProtocolServer.java
ProtocolClient.java
assignment3_2 C:₩Users₩dlwld₩eclipse-workspace₩assignme
.idea
out
src
Java_client_TCP
Java_server_TCP
ProtocolClient
ProtocolServer
assignment3_2.iml
External Libraries
Scratches and Consoles
import java.io.*;
import java.net.Socket;
public class ProtocolClient {
public static void main(String args[]) throws IOException {
Socket socket=null;
try {
// Socket 생성 및 접속 (IP는 장소에 따라 달라지므로 바뀌줘야 한다.)
socket = new Socket(host: "172.30.1.28", port: 12000);
} catch (Exception e) {
e.printStackTrace();
}
// 생성된 소켓을 통해 BufferedReader, PrintWriter 생성 (입출력 스트림)
//클라이언트 -> 소켓
BufferedReader input = new BufferedReader(new InputStreamReader(System.in));
String msg=input.readLine();
PrintWriter writer = new PrintWriter(new OutputStreamWriter(socket.getOutputStre
writer.println(msg); //서버로 데이터를 전송한다.(writer.println!=System.out.println
writer.flush(); //버퍼 안에 있는 값들을 전부 비워준다.
Run: ProtocolServer ProtocolClient
waiting connect...
server : connect success DESKTOP-BDEIK8B
Saved data : nice to meet you
server : data transmit success
waiting connect...
server : connect success DESKTOP-BDEIK8B
Saved data : see you tomorrow~
server : data transmit success
waiting connect...
Version Control Run TODO Problems Terminal Services Build
All files are up-to-date (moments ago)
검색하려면 여기에 입력하십시

5. capture traceroute nmap – naver.com, kt dns server..