

NAME : Bhavik Ransubhe  
CLASS : TE (B) COMP  
ROLL NO : 39055

## CNL ASSIGNMENT 2 - (B1)

### PROBLEM STATEMENT:

Write a Program with following options to transfer- (Use any Framing Technic). a. Characters separated by space b. One Strings at a time c. One Sentence at a time d. file

### PROGRAM:

Sender.java:

```
package com.company;

import org.jetbrains.annotations.NotNull;

import java.io.*;
import java.net.ServerSocket;
import java.net.Socket;
import java.util.Scanner;
import java.util.regex.Pattern;

public class Sender {

    private DataOutputStream dataOutputStream;
    private static Scanner scanner=new Scanner(System.in);
    Socket socket;

    Sender(){

        try {
            ServerSocket serverSocket = new ServerSocket(3000);
            socket = serverSocket.accept();
            dataOutputStream=new DataOutputStream(socket.getOutputStream());
            if(socket.isConnected()){
                System.out.print("\nClient Connected");
            }
            while (true) {
                sendData();
            }
        } catch (IOException e) {
            e.printStackTrace();
        }
    }

    private void sendData() throws IOException {
```

```

        System.out.print("Select The Method :\n1.Char With Spaces\n2.String At
A Time\n3.Sentence At A Time\n4.File");
        int ch=scanner.nextInt();
        switch (ch){
            case 1:scanner.nextLine();
                System.out.print("Enter Characters with spaces :");
                String ip=scanner.nextLine();
                dataOutputStream.writeInt(1);
                dataOutputStream.writeInt(ip.length());

                for(int i=0;i<ip.length();i++){
                    dataOutputStream.writeUTF(String.valueOf(ip.charAt(i)));
                }

                break;

            case 2:scanner.nextLine();
                System.out.print("Enter String :");
                String ip1=scanner.nextLine();
                String[] str=ip1.split(" ");
                dataOutputStream.writeInt(2);
                dataOutputStream.writeInt(5);
                sendBuffer(getCountFrame(str));
                dataOutputStream.writeInt(2);
                break;

            case 3:scanner.nextLine();
                System.out.print("Enter Sentences with \".\" :");
                String ip2=scanner.nextLine();
                String[] str1=ip2.split(Pattern.quote("."));
                dataOutputStream.writeInt(3);
                dataOutputStream.writeInt(5);
                sendBuffer(getCountFrame(str1));
                dataOutputStream.writeInt(2);
                break;

            case 4:
                dataOutputStream.writeInt(4);
                dataOutputStream.writeInt(6);
                sendFile();
                dataOutputStream.writeInt(2);
        }
    }

    private void sendFile(){
        try {
            byte b[]=new byte[10000];
            byte send[][]=new byte[100][10000];
            InputStream in = new FileInputStream(new
File("D:\\sendFile.txt"));
            BufferedReader reader = new BufferedReader(new
InputStreamReader(in));
            StringBuilder out = new StringBuilder();
            String line;
            while ((line = reader.readLine()) != null) {

```

```

        out.append(line);
    }
    String outData=out.toString();
    String str[]=outData.split(Pattern.quote(" "));
    String op=getCountFrame(str);

    OutputStream outputStream = socket.getOutputStream();

    for(int i=0;i<op.length();i++){
        if(Character.isDigit(op.charAt(i))){
            StringBuilder sendB = new StringBuilder("");
            int l=Integer.parseInt(String.valueOf(op.charAt(i)));
            for(int j=i+1;j<l+i+1;j++){
                sendB.append(op.charAt(j));
            }
            send[i]=sendB.toString().getBytes();
            dataOutputStream.writeInt(1);
            outputStream.write(send[i],0,send[i].length);
        }
    }
    System.out.println("\nFile sent Successfully!");
} catch (IOException e){
    e.printStackTrace();
}
}

private void sendBuffer(String frame) throws IOException {
    for(int i=0;i<frame.length();i++){
        if(Character.isDigit(frame.charAt(i))){
            StringBuilder sendB = new StringBuilder("");
            int l=Integer.parseInt(String.valueOf(frame.charAt(i)));
            for(int j=i+1;j<l+i+1;j++){
                sendB.append(frame.charAt(j));
            }
            dataOutputStream.writeInt(1);
            dataOutputStream.writeUTF(sendB.toString());
        }
    }
}

@NotNull
private String getCountFrame(String[] arr){
    StringBuilder frame = new StringBuilder("");
    for (String s : arr) {
        frame.append(s.length());
        frame.append(s);
    }
    return frame.toString();
}

public static void main(String[] args){
    new Sender();
}

```

Client.java:

```
package com.company;

import java.io.*;
import java.net.Socket;
import java.util.Arrays;

public class Client {

    private DataInputStream dataInputStream;
    private DataOutputStream dataOutputStream;
    Socket socket;

    Client() {
        try {
            socket = new Socket("localhost", 3000);
            dataInputStream = new DataInputStream(socket.getInputStream());
            dataOutputStream = new DataOutputStream(socket.getOutputStream());
            if (socket.isConnected()) {
                System.out.print("\nConnected To Server\n");
            }
        } catch (IOException e) {
            e.printStackTrace();
        }
        while (true){
            try {
                receiveData();
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }

    private void receiveData() throws IOException {
        int ch = dataInputStream.readInt();
        switch (ch) {
            case 1:
                System.out.print("Characters At A Time : ");
                break;
            case 2:
                System.out.print("\nString At A Time : ");
                break;
            case 3:
                System.out.print("\nSentences At A Time : ");
                break;
            case 4:
                System.out.print("\nFile : ");
                break;
            case 5: int loops;
                boolean stat;
                do{
                    loops=dataInputStream.readInt();
                    if(loops==1) {
                        String buffer = dataInputStream.readUTF();
                        System.out.println(buffer);
                    }
                } while (stat);
                break;
        }
    }
}
```

```

        stat=true;
    }else{
        stat=false;
    }

    }while (stat);
    break;
case 6:
    byte[] c = new byte[10000];
    do{
        loops=dataInputStream.readInt();
        if(loops==1) {
            receiveFile(c);
            stat=true;
        }else{
            stat=false;
        }
    }while (stat);
    break;
}
}

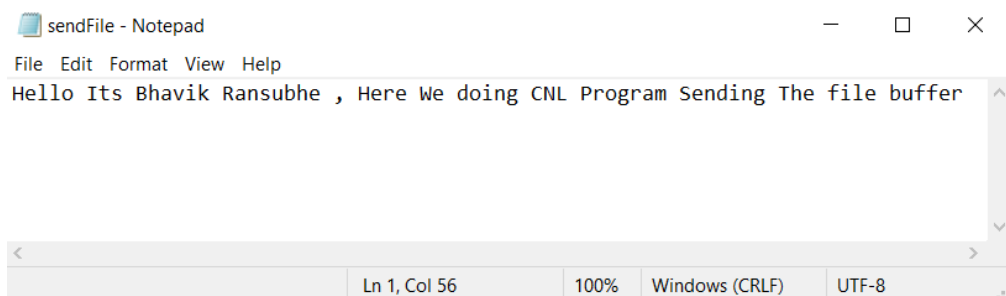
private void receiveFile(byte[] c){
    InputStream inputStream = null;
    byte[] b = new byte[10000];
    try {
        inputStream = socket.getInputStream();
        inputStream.read(b,0,b.length);
        FileOutputStream fos = new FileOutputStream("C:\\Users\\bhavi
Desktop\\CNL\\Framing Technique\\Client\\src\\com\\company\\receivedFile");
        fos.write(b,0,b.length);

    } catch (IOException e) {
        e.printStackTrace();
    }
    return;
}

public static void main(String[] args){
    new Client();
}
}

```

sendFile.txt:



## OUTPUT:

Sender:

```
Client Connected
Select The Method :
1.Char With Spaces
2.String At A Time
3.Sentence At A Time
4.File :1
Enter Characters with spaces :B H A V I K   R A N S U B H E
Select The Method :
1.Char With Spaces
2.String At A Time
3.Sentence At A Time
4.File :2
Enter String :BHAVIK RANSUBHE
Select The Method :
1.Char With Spaces
2.String At A Time
3.Sentence At A Time
4.File :3
Enter Sentences with "." :HELLO.ITS BHAVIK RANSUBHE HERE.I AM FROM TE B
COMPUTER ENGINEERING DEPARTMENT
Select The Method :
1.Char With Spaces
2.String At A Time
3.Sentence At A Time
4.File :4

File sent Successfully!

Select The Method :
1.Char With Spaces
2.String At A Time
3.Sentence At A Time
4.File :5
```

Client:

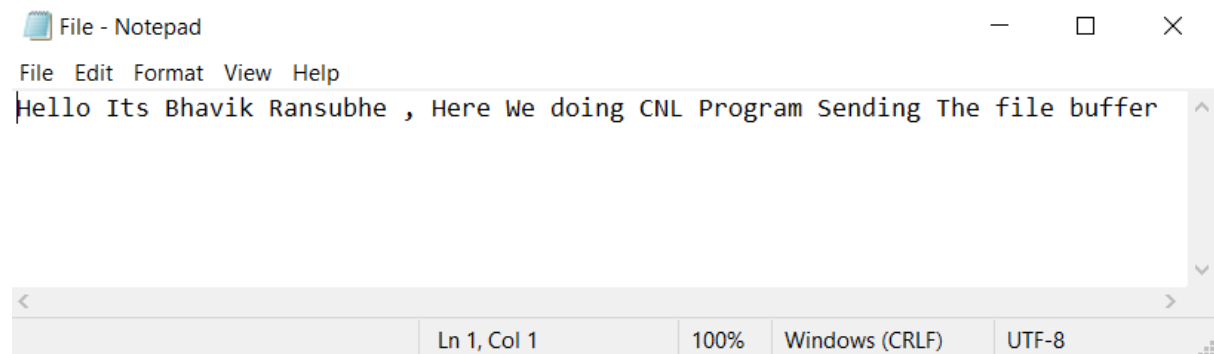
```
Connected To Server
Characters At A Time : B
H
A
V
I
K
R
A
N
S
U
B
H
E
```

```
String At A Time :
BHAVIK
RANSUBHE

Sentences At A Time :
HELLO
ITS BHAVIK RANSUBHE HERE
I AM FROM TE B COMPUTER ENGINEERING DEPARTMENT

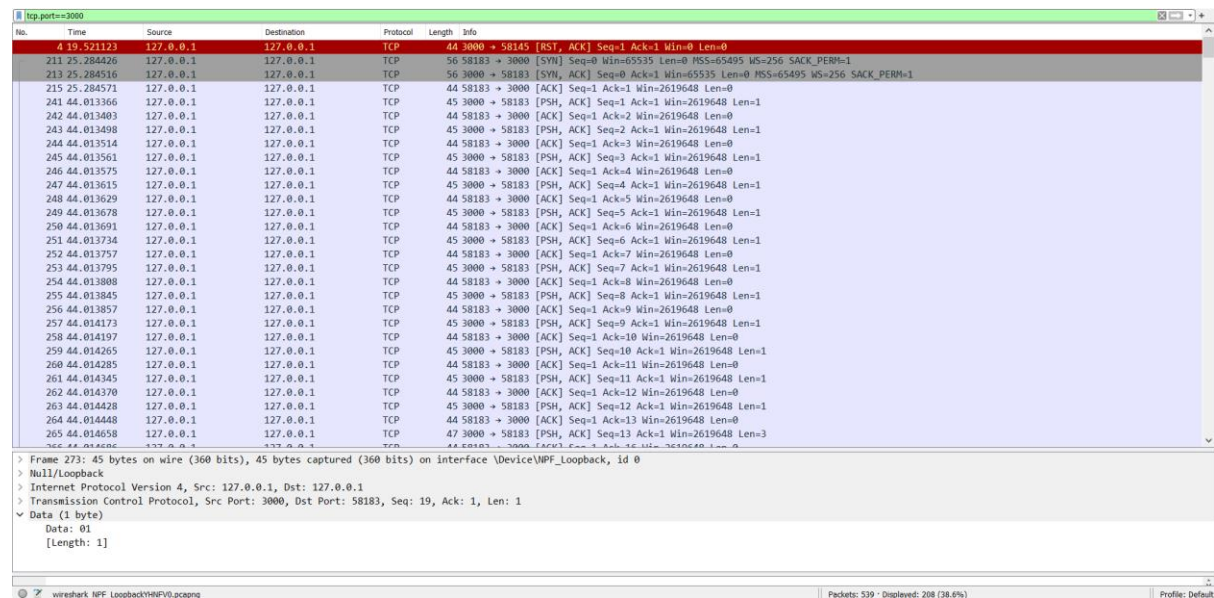
File :
```

File.txt:



## Wireshark:

Characters Separate By Space:



String At A Time:

No.	Time	Source	Destination	Protocol	Length	Info
26	0.000322	127.0.0.1	127.0.0.1	TCP	44	58183 → 3000 [ACK] Seq=1 Ack=21 Win=10232 Len=0
27	0.001024	127.0.0.1	127.0.0.1	TCP	45	3000 → 58183 [PSH, ACK] Seq=21 Ack=1 Win=10233 Len=1
28	0.001039	127.0.0.1	127.0.0.1	TCP	44	58183 → 3000 [ACK] Seq=1 Ack=22 Win=10232 Len=0
29	0.001089	127.0.0.1	127.0.0.1	TCP	45	3000 → 58183 [PSH, ACK] Seq=22 Ack=1 Win=10233 Len=1
30	0.001103	127.0.0.1	127.0.0.1	TCP	44	58183 → 3000 [ACK] Seq=1 Ack=23 Win=10232 Len=0
31	0.001141	127.0.0.1	127.0.0.1	TCP	45	3000 → 58183 [PSH, ACK] Seq=23 Ack=1 Win=10233 Len=1
32	0.001154	127.0.0.1	127.0.0.1	TCP	44	58183 → 3000 [ACK] Seq=1 Ack=24 Win=10232 Len=0
33	0.001192	127.0.0.1	127.0.0.1	TCP	45	3000 → 58183 [PSH, ACK] Seq=24 Ack=1 Win=10233 Len=1
34	0.001213	127.0.0.1	127.0.0.1	TCP	44	58183 → 3000 [ACK] Seq=1 Ack=25 Win=10232 Len=0
35	0.001276	127.0.0.1	127.0.0.1	TCP	52	3000 → 58183 [PSH, ACK] Seq=25 Ack=1 Win=10233 Len=8
36	0.001291	127.0.0.1	127.0.0.1	TCP	44	58183 → 3000 [ACK] Seq=1 Ack=33 Win=10232 Len=0
37	0.001367	127.0.0.1	127.0.0.1	TCP	45	3000 → 58183 [PSH, ACK] Seq=33 Ack=1 Win=10233 Len=1
38	0.001381	127.0.0.1	127.0.0.1	TCP	44	58183 → 3000 [ACK] Seq=1 Ack=34 Win=10232 Len=0
39	0.001441	127.0.0.1	127.0.0.1	TCP	45	3000 → 58183 [PSH, ACK] Seq=34 Ack=1 Win=10233 Len=1
40	0.001469	127.0.0.1	127.0.0.1	TCP	44	58183 → 3000 [ACK] Seq=1 Ack=35 Win=10232 Len=0
41	0.001532	127.0.0.1	127.0.0.1	TCP	45	3000 → 58183 [PSH, ACK] Seq=35 Ack=1 Win=10233 Len=1
42	0.001555	127.0.0.1	127.0.0.1	TCP	44	58183 → 3000 [ACK] Seq=1 Ack=36 Win=10232 Len=0
43	0.001639	127.0.0.1	127.0.0.1	TCP	45	3000 → 58183 [PSH, ACK] Seq=36 Ack=1 Win=10233 Len=1
44	0.001664	127.0.0.1	127.0.0.1	TCP	44	58183 → 3000 [ACK] Seq=1 Ack=37 Win=10232 Len=0
45	0.001836	127.0.0.1	127.0.0.1	TCP	52	3000 → 58183 [PSH, ACK] Seq=37 Ack=1 Win=10233 Len=8
46	0.001866	127.0.0.1	127.0.0.1	TCP	44	58183 → 3000 [ACK] Seq=1 Ack=45 Win=10232 Len=0
47	0.001936	127.0.0.1	127.0.0.1	TCP	45	3000 → 58183 [PSH, ACK] Seq=45 Ack=1 Win=10233 Len=1
48	0.001950	127.0.0.1	127.0.0.1	TCP	44	58183 → 3000 [ACK] Seq=1 Ack=46 Win=10232 Len=0
49	0.002016	127.0.0.1	127.0.0.1	TCP	45	3000 → 58183 [PSH, ACK] Seq=46 Ack=1 Win=10233 Len=1
50	0.002034	127.0.0.1	127.0.0.1	TCP	44	58183 → 3000 [ACK] Seq=1 Ack=47 Win=10232 Len=0
51	0.002116	127.0.0.1	127.0.0.1	TCP	45	3000 → 58183 [PSH, ACK] Seq=47 Ack=1 Win=10233 Len=1
52	0.002135	127.0.0.1	127.0.0.1	TCP	44	58183 → 3000 [ACK] Seq=1 Ack=48 Win=10232 Len=0
53	0.002181	127.0.0.1	127.0.0.1	TCP	45	3000 → 58183 [PSH, ACK] Seq=48 Ack=1 Win=10233 Len=1
54	0.002193	127.0.0.1	127.0.0.1	TCP	44	58183 → 3000 [ACK] Seq=1 Ack=49 Win=10232 Len=0
Frame 25: 52 bytes on wire (416 bits), 52 bytes captured (416 bits) on interface \Device\NPF_{...} id 0						
Null/Loopback						
Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1						
Transmission Control Protocol, Src Port: 3000, Dst Port: 58183, Seq: 13, Ack: 1, Len: 8						
Data (8 bytes)						
Data: 0006414456414954						
[Length: 8]						
Frame (Frame), 52 bytes						
					Packets: 58 · Displayed: 54 (93.1%)	Profile: Default

## Sentence At A Time:

No.	Time	Source	Destination	Protocol	Length	Info
30	29.669641	127.0.0.1	127.0.0.1	TCP	45	3000 → 58183 [PSH, ACK] Seq=9 Ack=1 Win=10233 Len=1
31	29.669664	127.0.0.1	127.0.0.1	TCP	44	58183 → 3000 [ACK] Seq=1 Ack=10 Win=10232 Len=0
32	29.669723	127.0.0.1	127.0.0.1	TCP	45	3000 → 58183 [PSH, ACK] Seq=10 Ack=1 Win=10233 Len=1
33	29.669744	127.0.0.1	127.0.0.1	TCP	44	58183 → 3000 [ACK] Seq=1 Ack=11 Win=10232 Len=0
34	29.669799	127.0.0.1	127.0.0.1	TCP	45	3000 → 58183 [PSH, ACK] Seq=11 Ack=1 Win=10233 Len=1
35	29.669818	127.0.0.1	127.0.0.1	TCP	44	58183 → 3000 [ACK] Seq=1 Ack=12 Win=10232 Len=0
36	29.669857	127.0.0.1	127.0.0.1	TCP	45	3000 → 58183 [PSH, ACK] Seq=12 Ack=1 Win=10233 Len=1
37	29.669870	127.0.0.1	127.0.0.1	TCP	44	58183 → 3000 [ACK] Seq=1 Ack=13 Win=10232 Len=0
38	29.669921	127.0.0.1	127.0.0.1	TCP	51	3000 → 58183 [PSH, ACK] Seq=13 Ack=1 Win=10233 Len=7
39	29.669938	127.0.0.1	127.0.0.1	TCP	44	58183 → 3000 [ACK] Seq=1 Ack=20 Win=10232 Len=0
40	29.670009	127.0.0.1	127.0.0.1	TCP	45	3000 → 58183 [PSH, ACK] Seq=20 Ack=1 Win=10233 Len=1
41	29.670021	127.0.0.1	127.0.0.1	TCP	44	58183 → 3000 [ACK] Seq=1 Ack=21 Win=10232 Len=0
42	29.670058	127.0.0.1	127.0.0.1	TCP	45	3000 → 58183 [PSH, ACK] Seq=21 Ack=1 Win=10233 Len=1
43	29.670078	127.0.0.1	127.0.0.1	TCP	44	58183 → 3000 [ACK] Seq=1 Ack=22 Win=10232 Len=0
44	29.670104	127.0.0.1	127.0.0.1	TCP	45	3000 → 58183 [PSH, ACK] Seq=22 Ack=1 Win=10233 Len=1
45	29.670116	127.0.0.1	127.0.0.1	TCP	44	58183 → 3000 [ACK] Seq=1 Ack=23 Win=10232 Len=0
46	29.670157	127.0.0.1	127.0.0.1	TCP	45	3000 → 58183 [PSH, ACK] Seq=23 Ack=1 Win=10233 Len=1
47	29.670178	127.0.0.1	127.0.0.1	TCP	44	58183 → 3000 [ACK] Seq=1 Ack=24 Win=10232 Len=0
48	29.670258	127.0.0.1	127.0.0.1	TCP	67	3000 → 58183 [PSH, ACK] Seq=24 Ack=1 Win=10233 Len=23
49	29.670276	127.0.0.1	127.0.0.1	TCP	44	58183 → 3000 [ACK] Seq=1 Ack=47 Win=10232 Len=0
50	29.670365	127.0.0.1	127.0.0.1	TCP	45	3000 → 58183 [PSH, ACK] Seq=47 Ack=1 Win=10233 Len=1
51	29.670382	127.0.0.1	127.0.0.1	TCP	44	58183 → 3000 [ACK] Seq=1 Ack=48 Win=10232 Len=0
52	29.670422	127.0.0.1	127.0.0.1	TCP	45	3000 → 58183 [PSH, ACK] Seq=48 Ack=1 Win=10233 Len=1
53	29.670435	127.0.0.1	127.0.0.1	TCP	44	58183 → 3000 [ACK] Seq=1 Ack=49 Win=10232 Len=0
54	29.670470	127.0.0.1	127.0.0.1	TCP	45	3000 → 58183 [PSH, ACK] Seq=49 Ack=1 Win=10233 Len=1
55	29.670483	127.0.0.1	127.0.0.1	TCP	44	58183 → 3000 [ACK] Seq=1 Ack=50 Win=10232 Len=0
56	29.670523	127.0.0.1	127.0.0.1	TCP	45	3000 → 58183 [PSH, ACK] Seq=50 Ack=1 Win=10233 Len=1
57	29.670541	127.0.0.1	127.0.0.1	TCP	44	58183 → 3000 [ACK] Seq=1 Ack=51 Win=10232 Len=0
58	29.670588	127.0.0.1	127.0.0.1	TCP	62	3000 → 58183 [PSH, ACK] Seq=51 Ack=1 Win=10233 Len=18
Frame 48: 67 bytes on wire (536 bits), 67 bytes captured (536 bits) on interface \Device\NPF_{...} id 0						
Null/Loopback						
Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1						
Transmission Control Protocol, Src Port: 3000, Dst Port: 58183, Seq: 24, Ack: 1, Len: 23						
Data (23 bytes)						
Data: 0015204144564149542046404c494e44205448414b5552						
[Length: 23]						
Frame (Frame), 67 bytes						
					Packets: 66 · Displayed: 47 (71.2%)	Profile: Default

## File:

No.	Time	Source	Destination	Protocol	Length	Info
171	74.228011	127.0.0.1	127.0.0.1	TCP	45	3000 → 58198 [PSH, ACK] Seq=7 Ack=1 Win=2619648 Len=1
172	74.228024	127.0.0.1	127.0.0.1	TCP	44	58198 → 3000 [ACK] Seq=1 Ack=8 Win=2619648 Len=0
173	74.228060	127.0.0.1	127.0.0.1	TCP	45	3000 → 58198 [PSH, ACK] Seq=8 Ack=1 Win=2619648 Len=1
174	74.228073	127.0.0.1	127.0.0.1	TCP	44	58198 → 3000 [ACK] Seq=1 Ack=9 Win=2619648 Len=0
175	74.230077	127.0.0.1	127.0.0.1	TCP	45	3000 → 58198 [PSH, ACK] Seq=9 Ack=1 Win=2619648 Len=1
176	74.230121	127.0.0.1	127.0.0.1	TCP	44	58198 → 3000 [ACK] Seq=1 Ack=10 Win=2619648 Len=0
177	74.230186	127.0.0.1	127.0.0.1	TCP	45	3000 → 58198 [PSH, ACK] Seq=10 Ack=1 Win=2619648 Len=1
178	74.230204	127.0.0.1	127.0.0.1	TCP	44	58198 → 3000 [ACK] Seq=1 Ack=11 Win=2619648 Len=0
179	74.230244	127.0.0.1	127.0.0.1	TCP	45	3000 → 58198 [PSH, ACK] Seq=11 Ack=1 Win=2619648 Len=1
180	74.230258	127.0.0.1	127.0.0.1	TCP	44	58198 → 3000 [ACK] Seq=1 Ack=12 Win=2619648 Len=0
181	74.230294	127.0.0.1	127.0.0.1	TCP	45	3000 → 58198 [PSH, ACK] Seq=12 Ack=1 Win=2619648 Len=1
182	74.230311	127.0.0.1	127.0.0.1	TCP	44	58198 → 3000 [ACK] Seq=1 Ack=13 Win=2619648 Len=0
183	74.230408	127.0.0.1	127.0.0.1	TCP	49	3000 → 58198 [PSH, ACK] Seq=13 Ack=1 Win=2619648 Len=5
184	74.230505	127.0.0.1	127.0.0.1	TCP	44	58198 → 3000 [ACK] Seq=1 Ack=18 Win=2619648 Len=0
185	74.230572	127.0.0.1	127.0.0.1	TCP	45	3000 → 58198 [PSH, ACK] Seq=18 Ack=1 Win=2619648 Len=1
186	74.230587	127.0.0.1	127.0.0.1	TCP	44	58198 → 3000 [ACK] Seq=1 Ack=19 Win=2619648 Len=0
187	74.230625	127.0.0.1	127.0.0.1	TCP	45	3000 → 58198 [PSH, ACK] Seq=19 Ack=1 Win=2619648 Len=1
188	74.230638	127.0.0.1	127.0.0.1	TCP	44	58198 → 3000 [ACK] Seq=1 Ack=20 Win=2619648 Len=0
189	74.230679	127.0.0.1	127.0.0.1	TCP	45	3000 → 58198 [PSH, ACK] Seq=20 Ack=1 Win=2619648 Len=1
190	74.230693	127.0.0.1	127.0.0.1	TCP	44	58198 → 3000 [ACK] Seq=1 Ack=21 Win=2619648 Len=0
191	74.230729	127.0.0.1	127.0.0.1	TCP	45	3000 → 58198 [PSH, ACK] Seq=21 Ack=1 Win=2619648 Len=1
192	74.230742	127.0.0.1	127.0.0.1	TCP	44	58198 → 3000 [ACK] Seq=1 Ack=22 Win=2619648 Len=0
193	74.230778	127.0.0.1	127.0.0.1	TCP	47	3000 → 58198 [PSH, ACK] Seq=22 Ack=1 Win=2619648 Len=3
194	74.230792	127.0.0.1	127.0.0.1	TCP	44	58198 → 3000 [ACK] Seq=1 Ack=25 Win=2619648 Len=0
195	74.230859	127.0.0.1	127.0.0.1	TCP	45	3000 → 58198 [PSH, ACK] Seq=25 Ack=1 Win=2619648 Len=1
196	74.230894	127.0.0.1	127.0.0.1	TCP	44	58198 → 3000 [ACK] Seq=1 Ack=26 Win=2619648 Len=0
197	74.230913	127.0.0.1	127.0.0.1	TCP	45	3000 → 58198 [PSH, ACK] Seq=26 Ack=1 Win=2619648 Len=1
198	74.230950	127.0.0.1	127.0.0.1	TCP	44	58198 → 3000 [ACK] Seq=1 Ack=27 Win=2619648 Len=0
199	74.230986	127.0.0.1	127.0.0.1	TCP	45	3000 → 58198 [PSH, ACK] Seq=27 Ack=1 Win=2619648 Len=1
Frame 183: 49 bytes on wire (392 bits), 49 bytes captured (392 bits) on interface \Device\NPF_{...} id 0						
Null/Loopback						
Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1						
Transmission Control Protocol, Src Port: 3000, Dst Port: 58198, Seq: 13, Ack: 1, Len: 5						
Data (5 bytes)						
Data: 48656c66f						
[Length: 5]						
Frame (Frame), 49 bytes						
					Packets: 317 · Displayed: 204 (64.4%)	Profile: Default