ADDIS ABABA UNIVERSITY

INSTITUTE OF TECHNOLOGY

SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING

Data communication and Computer Networks(ECEG4211)

Assignment one

Group members

	Name	ID
1.	Adane Eshete	UGR/3760/12
2.	Ebisa Adugna	UGR/7063/12
3.	Yinges Damtie	UGR/7186/12

Submitted to:Dr.Ephrem Teshale
Date of Submission:4/13/2023

Table of contents

Contents page

1.1 INTRODUCTION

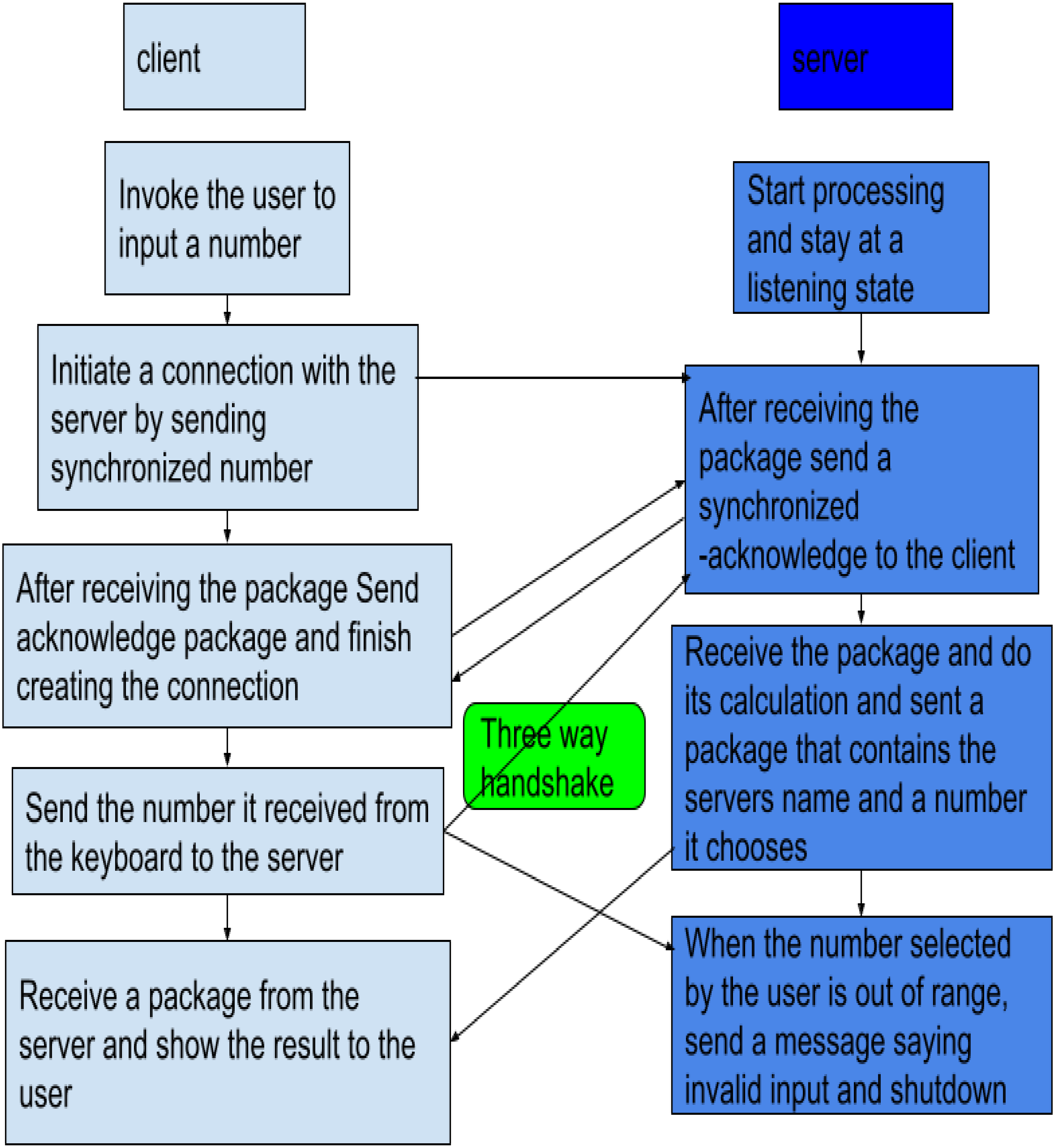
There are two types of data transmission protocols,UDP(user datagram protocol) and TCP(Transmission control protocol).UDP is a connectionless protocol ,does not establish a dedicated communication channel between end points before data is transmitted.It is not our concern here.
TCP on the other hand is a connection-oriented protocol and one of the most used communication protocol.It transmite reliable,ordered and error checked data between server and host.
TCP helps to manage cognition control,flow control and error detection mechanisms to transmit valuable data.

1.2 Objectives

- **Testing network connectivity:** A TCP connection can be used to test whether two network devices are able to establish a connection and exchange data.
- **Measuring network performance:** By establishing a TCP connection and measuring various metrics such as throughput, latency, and packet loss, we can analyze the performance of the network.
- **Developing and testing network applications:** TCP connections are commonly used in the development and testing of network applications.
- **Simulating network conditions**:By using wireshark we analysis the the tcp connection,the device connected ,ip address of server and client ,and the encrypted data transmitted between server and client or client to client

1.3 Working principles

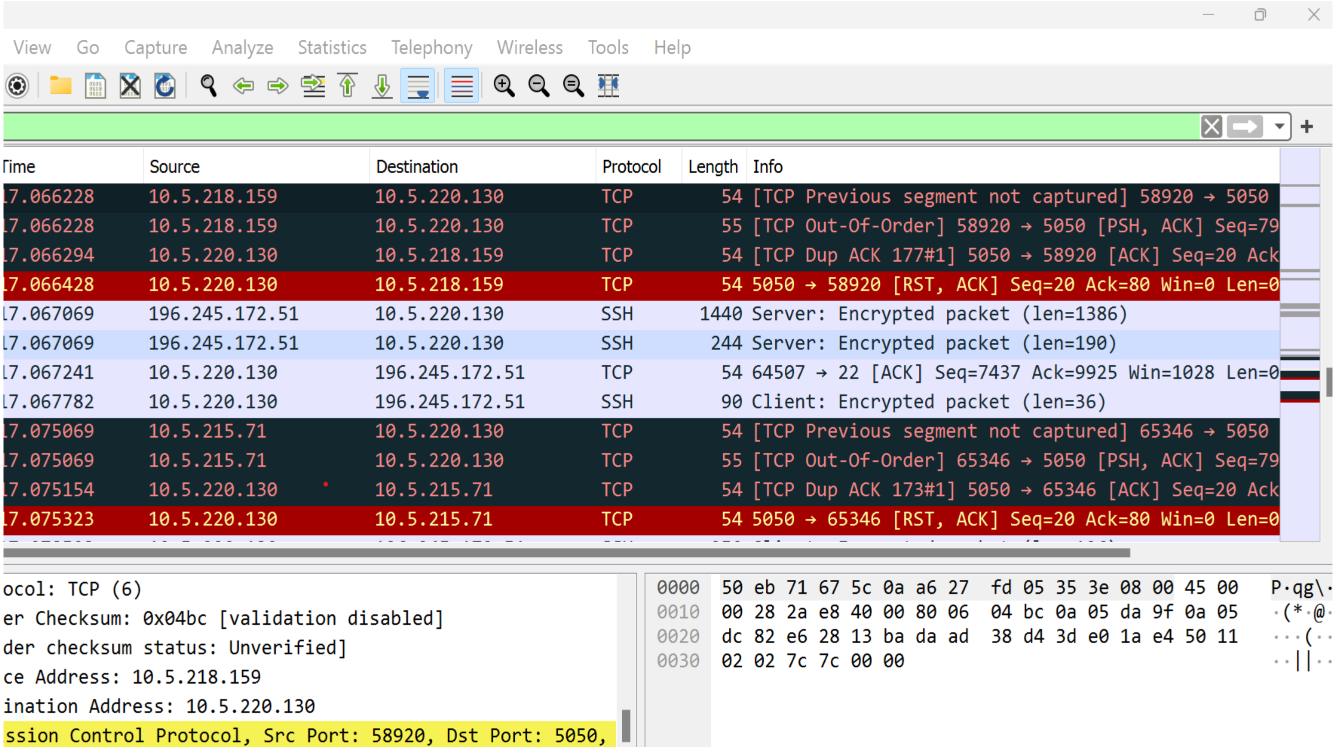
TCP IS a three way handshake transmission protocol,to get permission to connect with the server(between server and clients) or other clients(between clients).The figure below describes the working principle.


client
Invoke the user to input a number
Initiate a connection with the server by sending synchronized number
After receiving the package Send acknowledge package and finish creating the connection
Send the number it received from the keyboard to the server
Receive a package from the server and show the result to the user
Three way handshake
server
Start processing and stay at a listening state
After receiving the package send a synchronized -acknowledge to the client
Receive the package and do its calculation and sent a package that contains the servers name and a number it chooses
When the number selected by the user is out of range, send a message saying invalid input and shutdown

As seen from the figure above we do the following tasks:

- The entities of the client and server applications are identified by specifying port numberIP address and server and client name.
- A socket is created for both server and client applications and specified way of
  communication which is in the transport layer, TCP
- The server is programmed to stay at a listening state snice that is how servers operate in the real world but according to the instructions we limited it's on state on a given information.
- The client application is designed to accept a data from a user and send the data after it creates the connection
- Then we made the server application to accept and do some manipulations and chose a number by itself and send it to the client attached with itAnd both applications to show the results

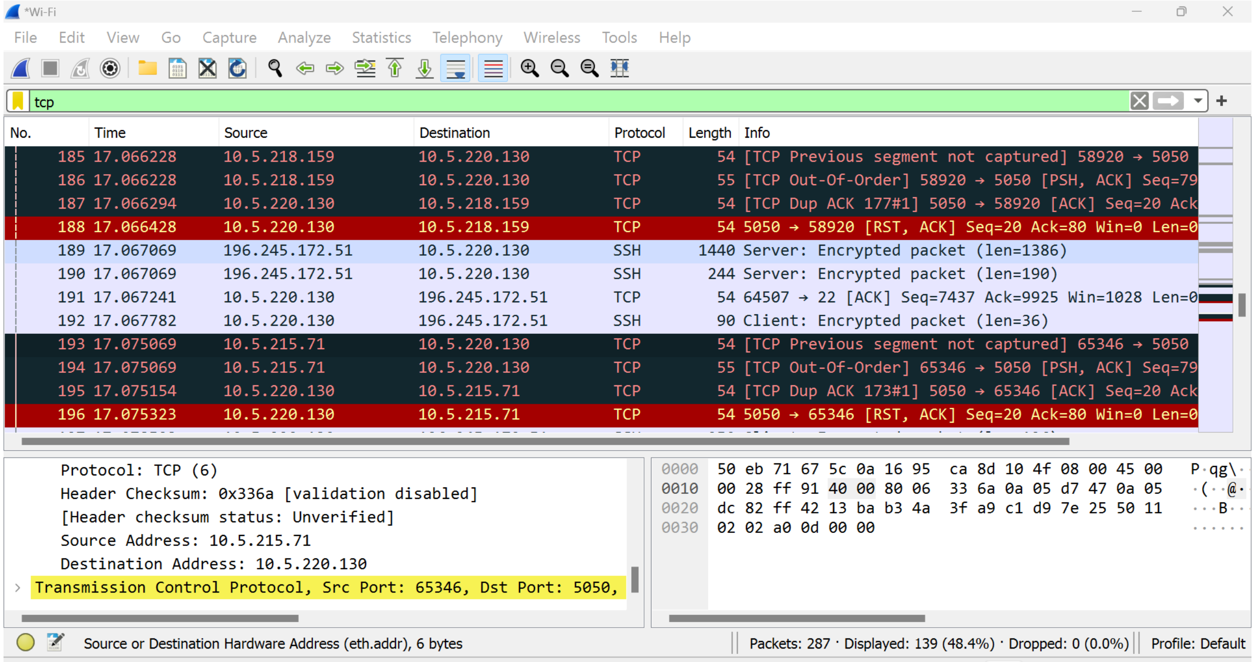
1.4 Wireshark analysis



```
Time       Source          Destination     Protocol Length Info
17.066228  10.5.218.159    10.5.220.130    TCP        54 [TCP Previous segment not captured] 58920 → 5050
17.066228  10.5.218.159    10.5.220.130    TCP        55 [TCP Out-Of-Order] 58920 → 5050 [PSH, ACK] Seq=79
17.066294  10.5.220.130    10.5.218.159    TCP        54 [TCP Dup ACK 177#1] 5050 → 58920 [ACK] Seq=20 Ack
17.066428  10.5.220.130    10.5.218.159    TCP        54 5050 → 58920 [RST, ACK] Seq=20 Ack=80 Win=0 Len=0
17.067069  196.245.172.51  10.5.220.130    SSH      1440 Server: Encrypted packet (len=1386)
17.067069  196.245.172.51  10.5.220.130    SSH       244 Server: Encrypted packet (len=190)
17.067241  10.5.220.130    196.245.172.51  TCP        54 64507 → 22 [ACK] Seq=7437 Ack=9925 Win=1028 Len=0
17.067782  10.5.220.130    196.245.172.51  SSH        90 Client: Encrypted packet (len=36)
17.075069  10.5.215.71     10.5.220.130    TCP        54 [TCP Previous segment not captured] 65346 → 5050
17.075069  10.5.215.71     10.5.220.130    TCP        55 [TCP Out-Of-Order] 65346 → 5050 [PSH, ACK] Seq=79
17.075154  10.5.220.130    10.5.215.71     TCP        54 [TCP Dup ACK 173#1] 5050 → 65346 [ACK] Seq=20 Ack
17.075323  10.5.220.130    10.5.215.71     TCP        54 5050 → 65346 [RST, ACK] Seq=20 Ack=80 Win=0 Len=0
```

```
ocol: TCP (6)
er Checksum: 0x04bc [validation disabled]
der checksum status: Unverified]
ce Address: 10.5.218.159
ination Address: 10.5.220.130
ssion Control Protocol, Src Port: 58920, Dst Port: 5050,
```

```
0000  50 eb 71 67 5c 0a a6 27  fd 05 35 3e 08 00 45 00   P·qg\·
0010  00 28 2a e8 40 00 80 06  04 bc 0a 05 da 9f 0a 05   ·(*·@·
0020  dc 82 e6 28 13 ba da ad  38 d4 3d e0 1a e4 50 11   ···(··
0030  02 02 7c 7c 00 00                                  ··||··
```


*Wi-Fi
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

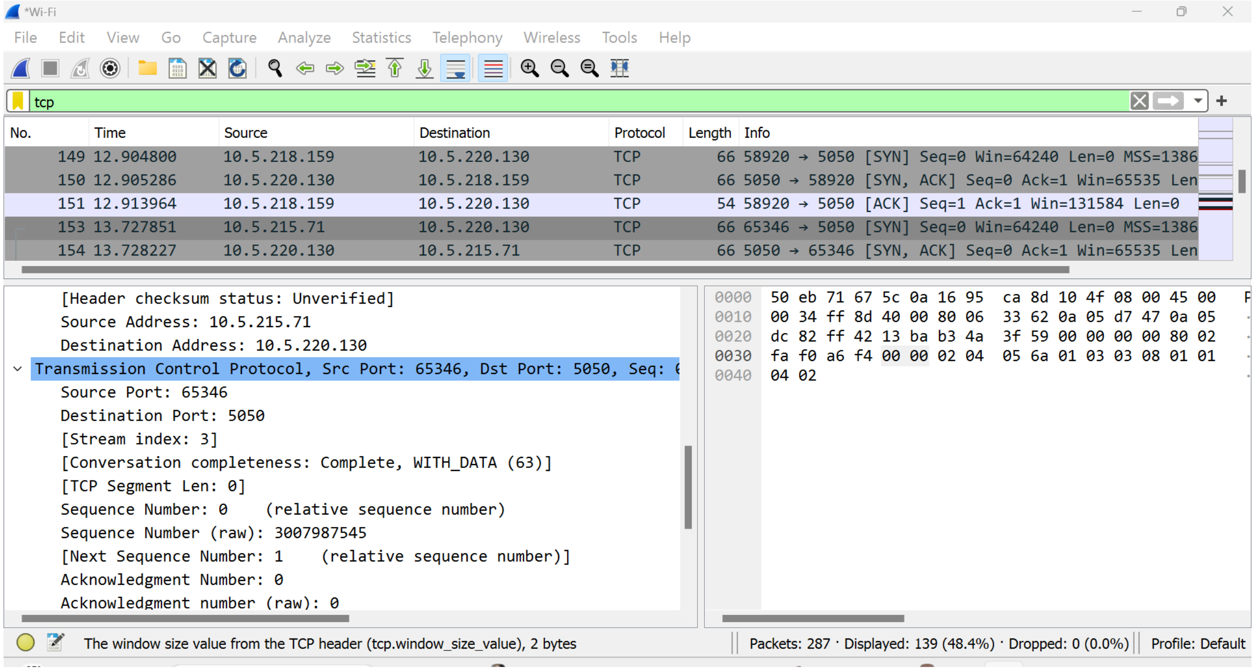
```
tcp

No.   Time        Source          Destination     Protocol Length Info
  185 17.066228   10.5.218.159    10.5.220.130    TCP        54 [TCP Previous segment not captured] 58920 → 5050
  186 17.066228   10.5.218.159    10.5.220.130    TCP        55 [TCP Out-Of-Order] 58920 → 5050 [PSH, ACK] Seq=79
  187 17.066294   10.5.220.130    10.5.218.159    TCP        54 [TCP Dup ACK 177#1] 5050 → 58920 [ACK] Seq=20 Ack
  188 17.066428   10.5.220.130    10.5.218.159    TCP        54 5050 → 58920 [RST, ACK] Seq=20 Ack=80 Win=0 Len=0
  189 17.067069   196.245.172.51  10.5.220.130    SSH      1440 Server: Encrypted packet (len=1386)
  190 17.067069   196.245.172.51  10.5.220.130    SSH       244 Server: Encrypted packet (len=190)
  191 17.067241   10.5.220.130    196.245.172.51  TCP        54 64507 → 22 [ACK] Seq=7437 Ack=9925 Win=1028 Len=0
  192 17.067782   10.5.220.130    196.245.172.51  SSH        90 Client: Encrypted packet (len=36)
  193 17.075069   10.5.215.71     10.5.220.130    TCP        54 [TCP Previous segment not captured] 65346 → 5050
  194 17.075069   10.5.215.71     10.5.220.130    TCP        55 [TCP Out-Of-Order] 65346 → 5050 [PSH, ACK] Seq=79
  195 17.075154   10.5.220.130    10.5.215.71     TCP        54 [TCP Dup ACK 173#1] 5050 → 65346 [ACK] Seq=20 Ack
  196 17.075323   10.5.220.130    10.5.215.71     TCP        54 5050 → 65346 [RST, ACK] Seq=20 Ack=80 Win=0 Len=0
```

```
    Protocol: TCP (6)
    Header Checksum: 0x336a [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 10.5.215.71
    Destination Address: 10.5.220.130
>  Transmission Control Protocol, Src Port: 65346, Dst Port: 5050,
```

```
0000  50 eb 71 67 5c 0a 16 95  ca 8d 10 4f 08 00 45 00   P·qg\·
0010  00 28 ff 91 40 00 80 06  33 6a 0a 05 d7 47 0a 05   ·(··@·
0020  dc 82 ff 42 13 ba b3 4a  3f a9 c1 d9 7e 25 50 11   ···B··
0030  02 02 a0 0d 00 00                                  ······
```

Source or Destination Hardware Address (eth.addr), 6 bytes | Packets: 287 · Displayed: 139 (48.4%) · Dropped: 0 (0.0%) | Profile: Default


*Wi-Fi
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

```
tcp

No.   Time        Source          Destination     Protocol Length Info
  149 12.904800   10.5.218.159    10.5.220.130    TCP        66 58920 → 5050 [SYN] Seq=0 Win=64240 Len=0 MSS=1386
  150 12.905286   10.5.220.130    10.5.218.159    TCP        66 5050 → 58920 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len
  151 12.913964   10.5.218.159    10.5.220.130    TCP        54 58920 → 5050 [ACK] Seq=1 Ack=1 Win=131584 Len=0
  153 13.727851   10.5.215.71     10.5.220.130    TCP        66 65346 → 5050 [SYN] Seq=0 Win=64240 Len=0 MSS=1386
  154 13.728227   10.5.220.130    10.5.215.71     TCP        66 5050 → 65346 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len
```

```
    [Header checksum status: Unverified]
    Source Address: 10.5.215.71
    Destination Address: 10.5.220.130
v  Transmission Control Protocol, Src Port: 65346, Dst Port: 5050, Seq: 0
    Source Port: 65346
    Destination Port: 5050
    [Stream index: 3]
    [Conversation completeness: Complete, WITH_DATA (63)]
    [TCP Segment Len: 0]
    Sequence Number: 0    (relative sequence number)
    Sequence Number (raw): 3007987545
    [Next Sequence Number: 1    (relative sequence number)]
    Acknowledgment Number: 0
    Acknowledgment number (raw): 0
```

```
0000  50 eb 71 67 5c 0a 16 95  ca 8d 10 4f 08 00 45 00   P
0010  00 34 ff 8d 40 00 80 06  33 62 0a 05 d7 47 0a 05   ·
0020  dc 82 ff 42 13 ba b3 4a  3f 59 00 00 00 00 80 02   ·
0030  fa f0 a6 f4 00 00 02 04  05 6a 01 03 03 08 01 01   ·
0040  04 02                                              ·
```

The window size value from the TCP header (tcp.window_size_value), 2 bytes | Packets: 287 · Displayed: 139 (48.4%) · Dropped: 0 (0.0%) | Profile: Default

1.4 Conclusion

TCP is a widely used protocol that is essential for reliable data transmission over the networks.Its reliability and robustness make it suitable for a wide range of applications ,from simple file transfer to complex real-time communications. TCP guarantees for data transfer segments in packets and ensures that packets delivered and acknowledged.