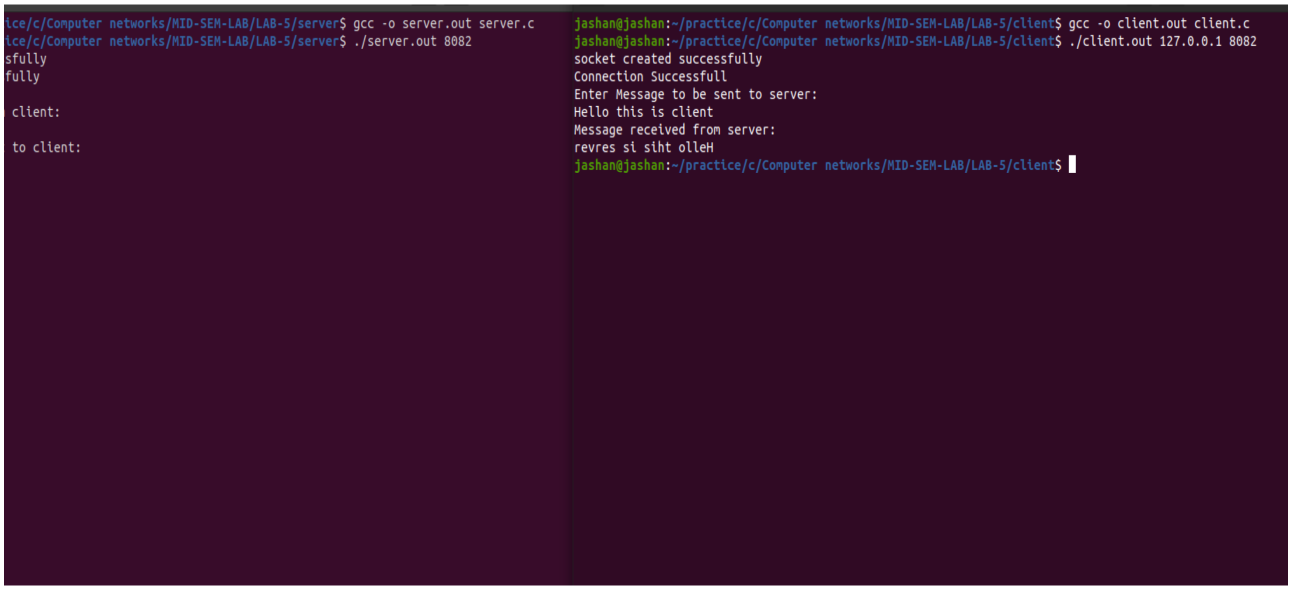
Name: Jashanpreet Singh
ID: 2018A7PS0134G

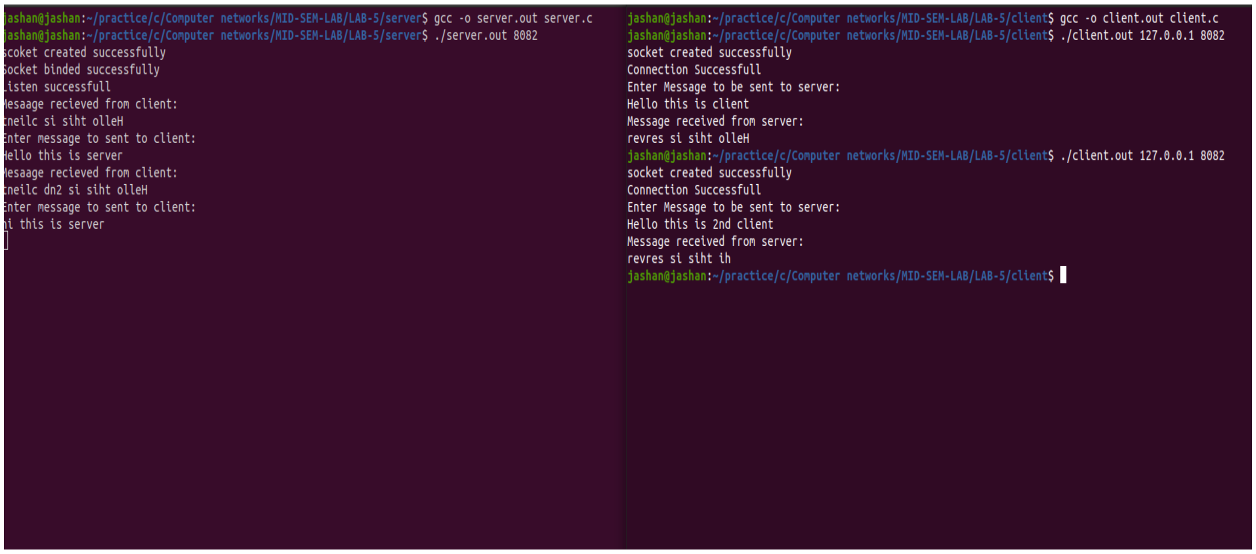
1) Server accepts first client



```
ice/c/Computer networks/MID-SEM-LAB/LAB-5/server$ gcc -o server.out server.c
ice/c/Computer networks/MID-SEM-LAB/LAB-5/server$ ./server.out 8082
sfully
fully

 client:

 to client:
```

```
jashan@jashan:~/practice/c/Computer networks/MID-SEM-LAB/LAB-5/client$ gcc -o client.out client.c
jashan@jashan:~/practice/c/Computer networks/MID-SEM-LAB/LAB-5/client$ ./client.out 127.0.0.1 8082
socket created successfully
Connection Successfull
Enter Message to be sent to server:
Hello this is client
Message received from server:
revres si siht olleH
jashan@jashan:~/practice/c/Computer networks/MID-SEM-LAB/LAB-5/client$ 
```

2) Server accepts second client

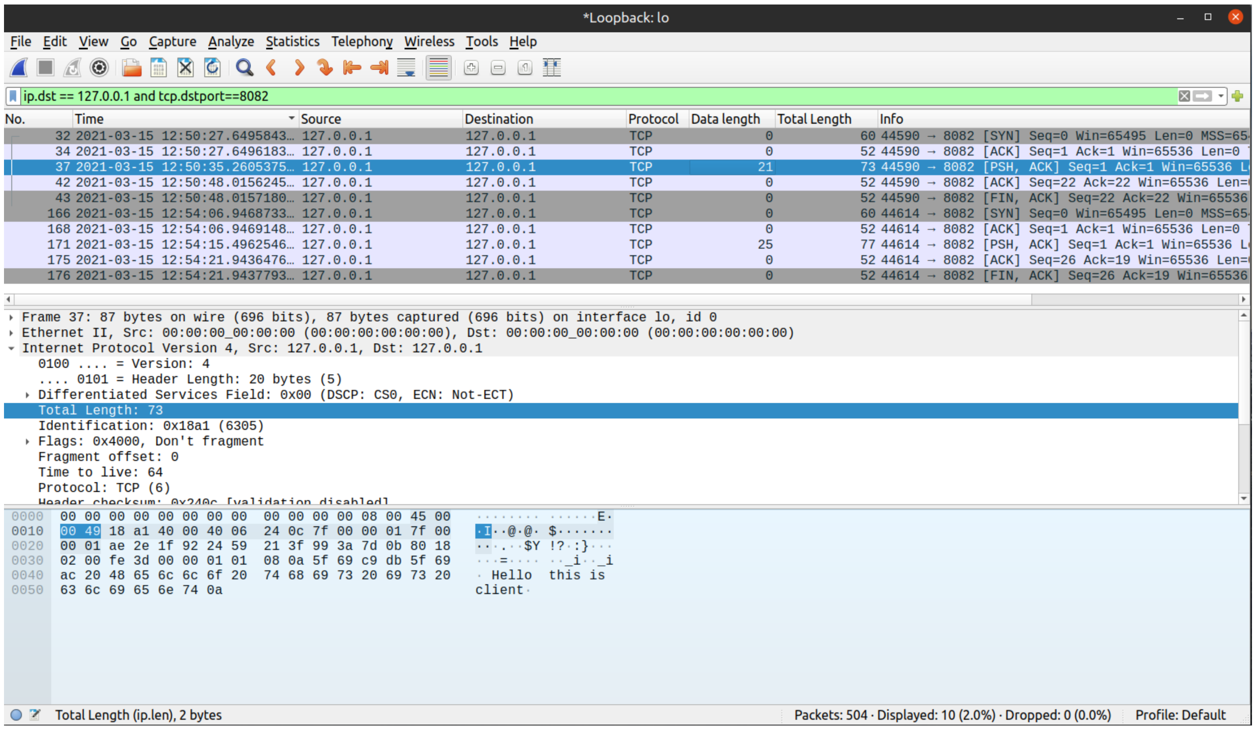


```
jashan@jashan:~/practice/c/Computer networks/MID-SEM-LAB/LAB-5/server$ gcc -o server.out server.c
jashan@jashan:~/practice/c/Computer networks/MID-SEM-LAB/LAB-5/server$ ./server.out 8082
scoket created successfully
Socket binded successfully
Listen successfull
Mesaage recieved from client:
tneilc si siht olleH
Enter message to sent to client:
Hello this is server
Mesaage recieved from client:
tneilc dn2 si siht olleH
Enter message to sent to client:
hi this is server
```

```
jashan@jashan:~/practice/c/Computer networks/MID-SEM-LAB/LAB-5/client$ gcc -o client.out client.c
jashan@jashan:~/practice/c/Computer networks/MID-SEM-LAB/LAB-5/client$ ./client.out 127.0.0.1 8082
socket created successfully
Connection Successfull
Enter Message to be sent to server:
Hello this is client
Message received from server:
revres si siht olleH
jashan@jashan:~/practice/c/Computer networks/MID-SEM-LAB/LAB-5/client$ ./client.out 127.0.0.1 8082
socket created successfully
Connection Successfull
Enter Message to be sent to server:
Hello this is 2nd client
Message received from server:
revres si siht ih
jashan@jashan:~/practice/c/Computer networks/MID-SEM-LAB/LAB-5/client$ 
```

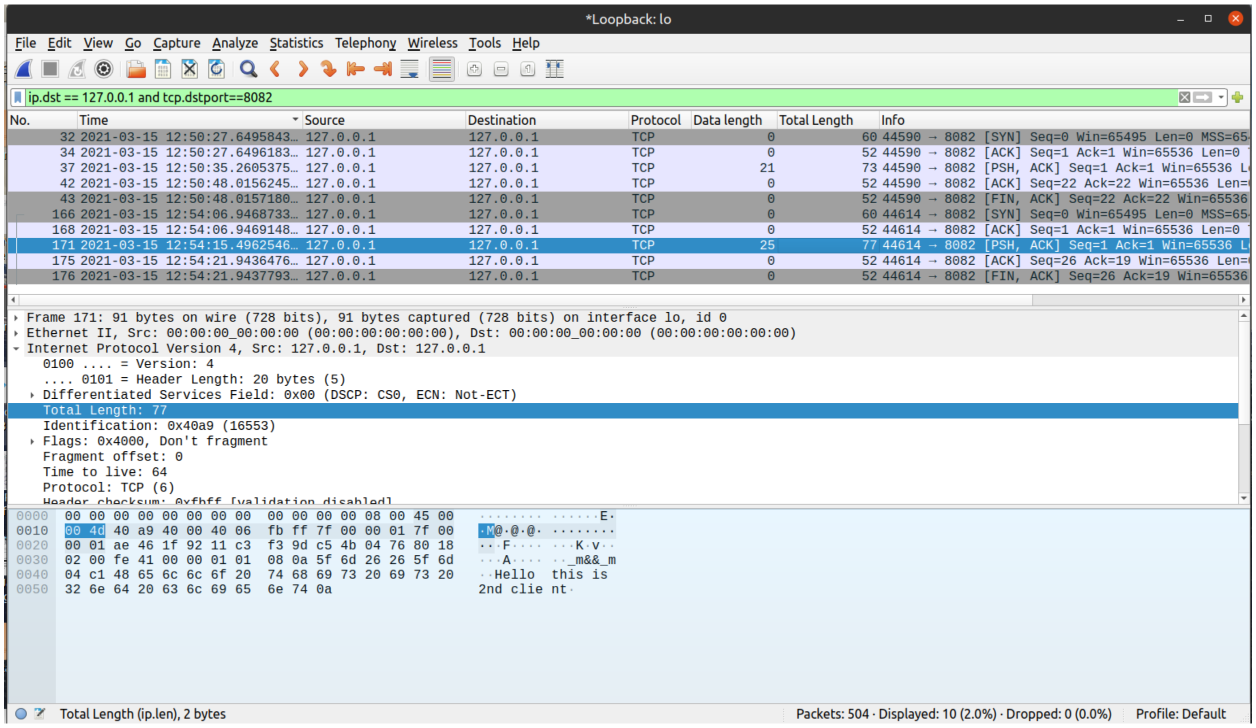
WIRESHARK

1) This shows packet sent by first client


*Loopback: lo
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

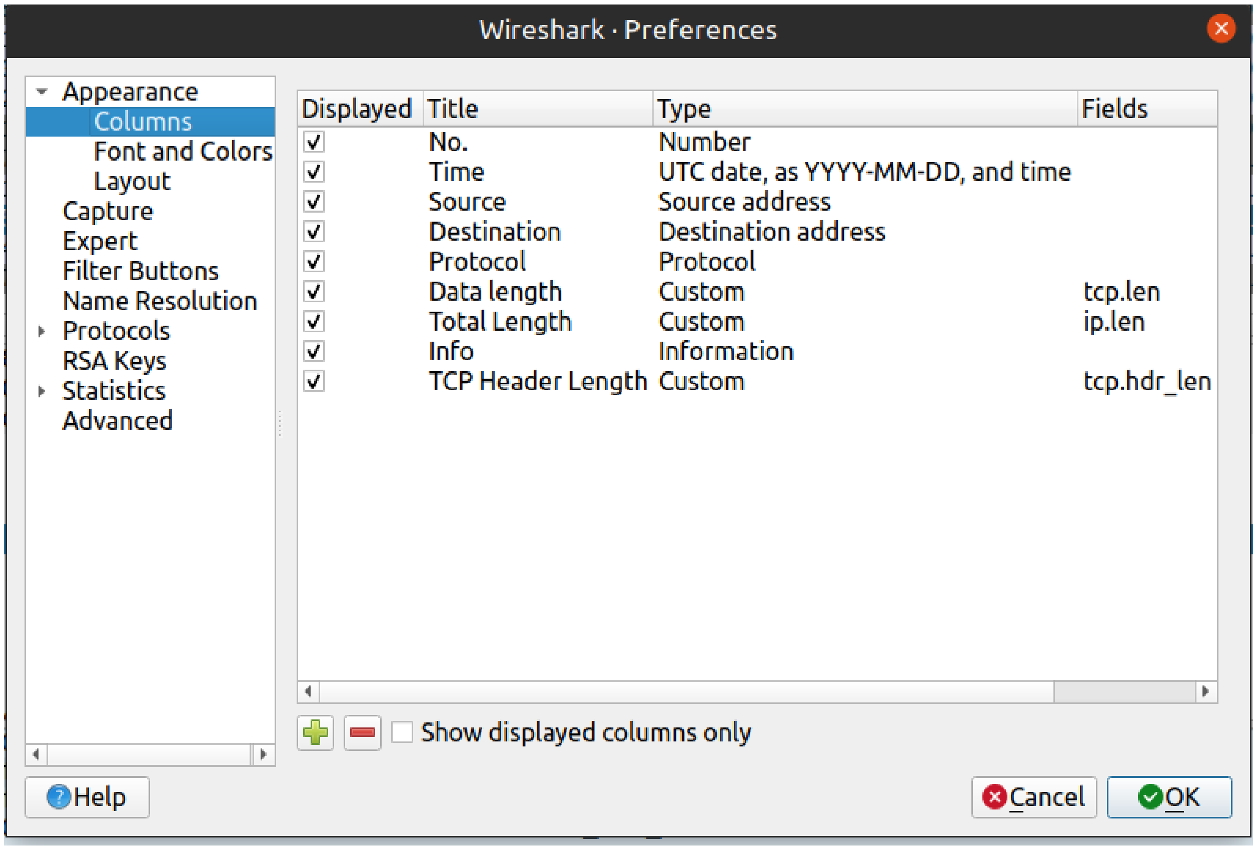
```
ip.dst == 127.0.0.1 and tcp.dstport==8082

No.  Time                          Source     Destination  Protocol  Data length  Total Length  Info
  32 2021-03-15 12:50:27.6495843… 127.0.0.1  127.0.0.1    TCP        0            60 44590 → 8082 [SYN] Seq=0 Win=65495 Len=0 MSS=65
  34 2021-03-15 12:50:27.6496183… 127.0.0.1  127.0.0.1    TCP        0            52 44590 → 8082 [ACK] Seq=1 Ack=1 Win=65536 Len=0
  37 2021-03-15 12:50:35.2605375… 127.0.0.1  127.0.0.1    TCP       21            73 44590 → 8082 [PSH, ACK] Seq=1 Ack=1 Win=65536 L
  42 2021-03-15 12:50:48.0156245… 127.0.0.1  127.0.0.1    TCP        0            52 44590 → 8082 [ACK] Seq=22 Ack=22 Win=65536 Len=
  43 2021-03-15 12:50:48.0157180… 127.0.0.1  127.0.0.1    TCP        0            52 44590 → 8082 [FIN, ACK] Seq=22 Ack=22 Win=65536
 166 2021-03-15 12:54:06.9468733… 127.0.0.1  127.0.0.1    TCP        0            60 44614 → 8082 [SYN] Seq=0 Win=65495 Len=0 MSS=65
 168 2021-03-15 12:54:06.9469148… 127.0.0.1  127.0.0.1    TCP        0            52 44614 → 8082 [ACK] Seq=1 Ack=1 Win=65536 Len=0
 171 2021-03-15 12:54:15.4962546… 127.0.0.1  127.0.0.1    TCP       25            77 44614 → 8082 [PSH, ACK] Seq=1 Ack=1 Win=65536 L
 175 2021-03-15 12:54:21.9436476… 127.0.0.1  127.0.0.1    TCP        0            52 44614 → 8082 [ACK] Seq=26 Ack=19 Win=65536 Len=
 176 2021-03-15 12:54:21.9437793… 127.0.0.1  127.0.0.1    TCP        0            52 44614 → 8082 [FIN, ACK] Seq=26 Ack=19 Win=65536

Frame 37: 87 bytes on wire (696 bits), 87 bytes captured (696 bits) on interface lo, id 0
Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)
Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
  Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
  Total Length: 73
  Identification: 0x18a1 (6305)
  Flags: 0x4000, Don't fragment
  Fragment offset: 0
  Time to live: 64
  Protocol: TCP (6)
  Header checksum: 0x240c [validation disabled]

0000  00 00 00 00 00 00 00 00  00 00 00 00 08 00 45 00   ········ ······E·
0010  00 49 18 a1 40 00 40 06  24 0c 7f 00 00 01 7f 00   ·I··@·@· $·······
0020  00 01 ae 2e 1f 92 24 59  21 3f 99 3a 7d 0b 80 18   ···.··$Y !?·:}···
0030  02 00 fe 3d 00 00 01 01  08 0a 5f 69 c9 db 5f 69   ···=···· ··_i··_i
0040  ac 20 48 65 6c 6c 6f 20  74 68 69 73 20 69 73 20   · Hello  this is
0050  63 6c 69 65 6e 74 0a                               client·

Total Length (ip.len), 2 bytes    Packets: 504 · Displayed: 10 (2.0%) · Dropped: 0 (0.0%)    Profile: Default
```

2) This shows packets captured by second client


*Loopback: lo
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

```
ip.dst == 127.0.0.1 and tcp.dstport==8082

No.  Time                           Source      Destination  Protocol  Data length  Total Length  Info
  32 2021-03-15 12:50:27.6495843…  127.0.0.1   127.0.0.1    TCP                 0            60 44590 → 8082 [SYN] Seq=0 Win=65495 Len=0 MSS=65
  34 2021-03-15 12:50:27.6496183…  127.0.0.1   127.0.0.1    TCP                 0            52 44590 → 8082 [ACK] Seq=1 Ack=1 Win=65536 Len=0
  37 2021-03-15 12:50:35.2605375…  127.0.0.1   127.0.0.1    TCP                21            73 44590 → 8082 [PSH, ACK] Seq=1 Ack=1 Win=65536 L
  42 2021-03-15 12:50:48.0156245…  127.0.0.1   127.0.0.1    TCP                 0            52 44590 → 8082 [ACK] Seq=22 Ack=22 Win=65536 Len=
  43 2021-03-15 12:50:48.0157180…  127.0.0.1   127.0.0.1    TCP                 0            52 44590 → 8082 [FIN, ACK] Seq=22 Ack=22 Win=65536
 166 2021-03-15 12:54:06.9468733…  127.0.0.1   127.0.0.1    TCP                 0            60 44614 → 8082 [SYN] Seq=0 Win=65495 Len=0 MSS=65
 168 2021-03-15 12:54:06.9469148…  127.0.0.1   127.0.0.1    TCP                 0            52 44614 → 8082 [ACK] Seq=1 Ack=1 Win=65536 Len=0
 171 2021-03-15 12:54:15.4962546…  127.0.0.1   127.0.0.1    TCP                25            77 44614 → 8082 [PSH, ACK] Seq=1 Ack=1 Win=65536 L
 175 2021-03-15 12:54:21.9436476…  127.0.0.1   127.0.0.1    TCP                 0            52 44614 → 8082 [ACK] Seq=26 Ack=19 Win=65536 Len=
 176 2021-03-15 12:54:21.9437793…  127.0.0.1   127.0.0.1    TCP                 0            52 44614 → 8082 [FIN, ACK] Seq=26 Ack=19 Win=65536

▸ Frame 171: 91 bytes on wire (728 bits), 91 bytes captured (728 bits) on interface lo, id 0
▸ Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)
▾ Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  ▸ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 77
    Identification: 0x40a9 (16553)
  ▸ Flags: 0x4000, Don't fragment
    Fragment offset: 0
    Time to live: 64
    Protocol: TCP (6)
    Header checksum: 0xfbff [validation disabled]

0000  00 00 00 00 00 00 00 00  00 00 00 00 08 00 45 00   ········ ······E·
0010  00 4d 40 a9 40 00 40 06  fb ff 7f 00 00 01 7f 00   ·M@·@·@· ········
0020  00 01 ae 46 1f 92 11 c3  f3 9d c5 4b 04 76 80 18   ···F···· ···K·v··
0030  02 00 fe 41 00 00 01 01  08 0a 5f 6d 26 26 5f 6d   ···A···· ··_m&&_m
0040  04 c1 48 65 6c 6c 6f 20  74 68 69 73 20 69 73 20   ··Hello  this is
0050  32 6e 64 20 63 6c 69 65  6e 74 0a                  2nd clie nt·

Total Length (ip.len), 2 bytes        Packets: 504 · Displayed: 10 (2.0%) · Dropped: 0 (0.0%)    Profile: Default
```

3) Columns displayed above



Displayed	Title	Type	Fields
✓	No.	Number	
✓	Time	UTC date, as YYYY-MM-DD, and time	
✓	Source	Source address	
✓	Destination	Destination address	
✓	Protocol	Protocol	
✓	Data length	Custom	tcp.len
✓	Total Length	Custom	ip.len
✓	Info	Information	
✓	TCP Header Length	Custom	tcp.hdr_len

4) Filter used: ip.dst == 127.0.0.1 and tcp.dstport==8082 and tcp.len

Here ip.dst is used to specify only packets sent to localhost, tcp.dstport is the destination port of the server.

5) Here the measured value of data length is the number of bytes of line plus one where plus one is for a new line character sent by the client.

Total length is the total segment length = TCP header length + data length