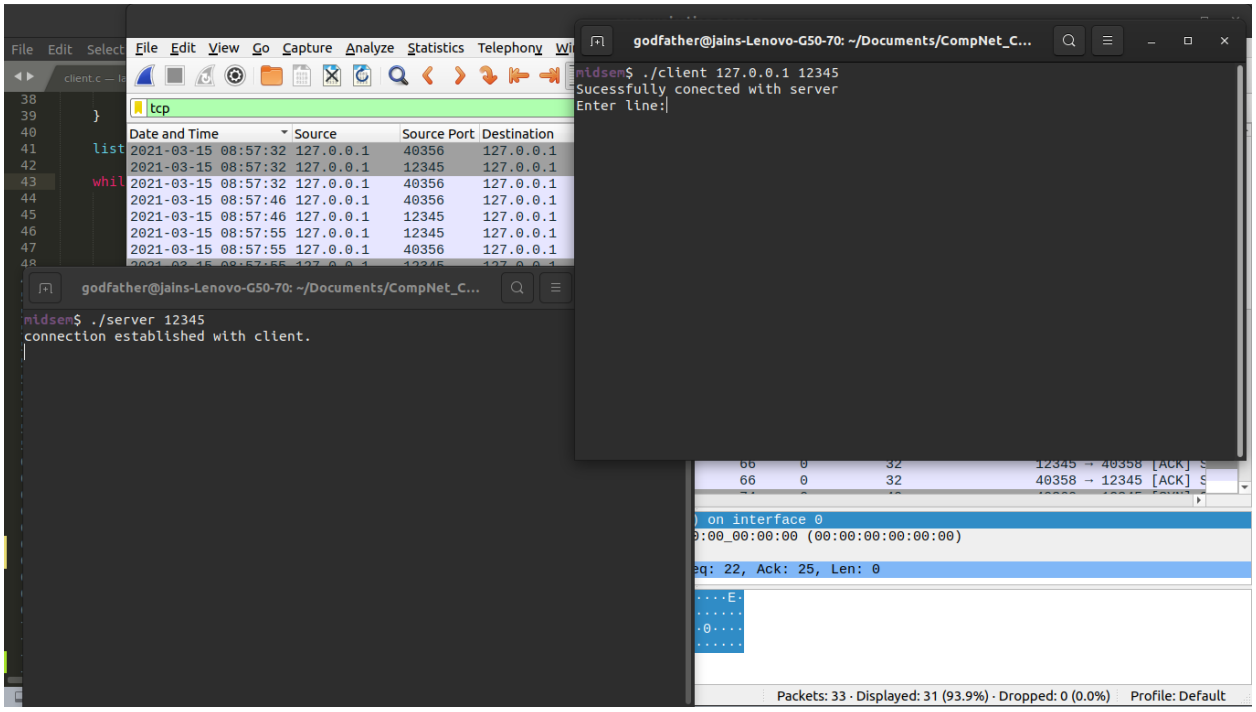
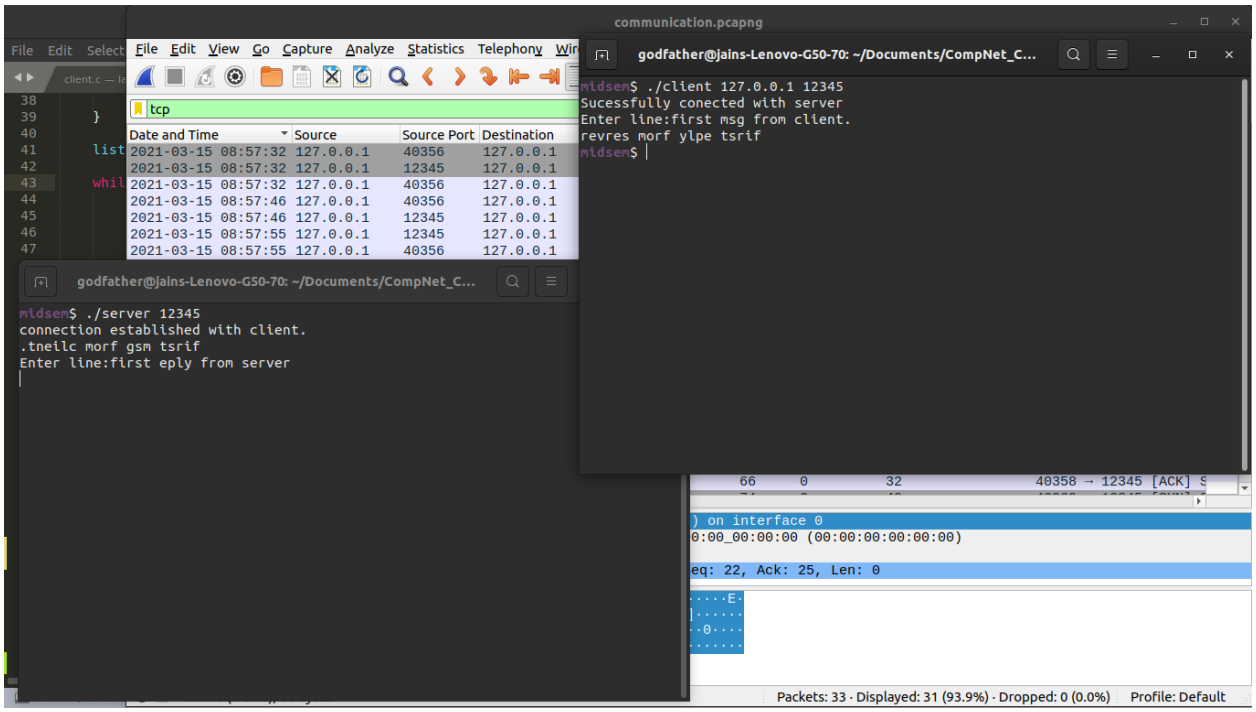
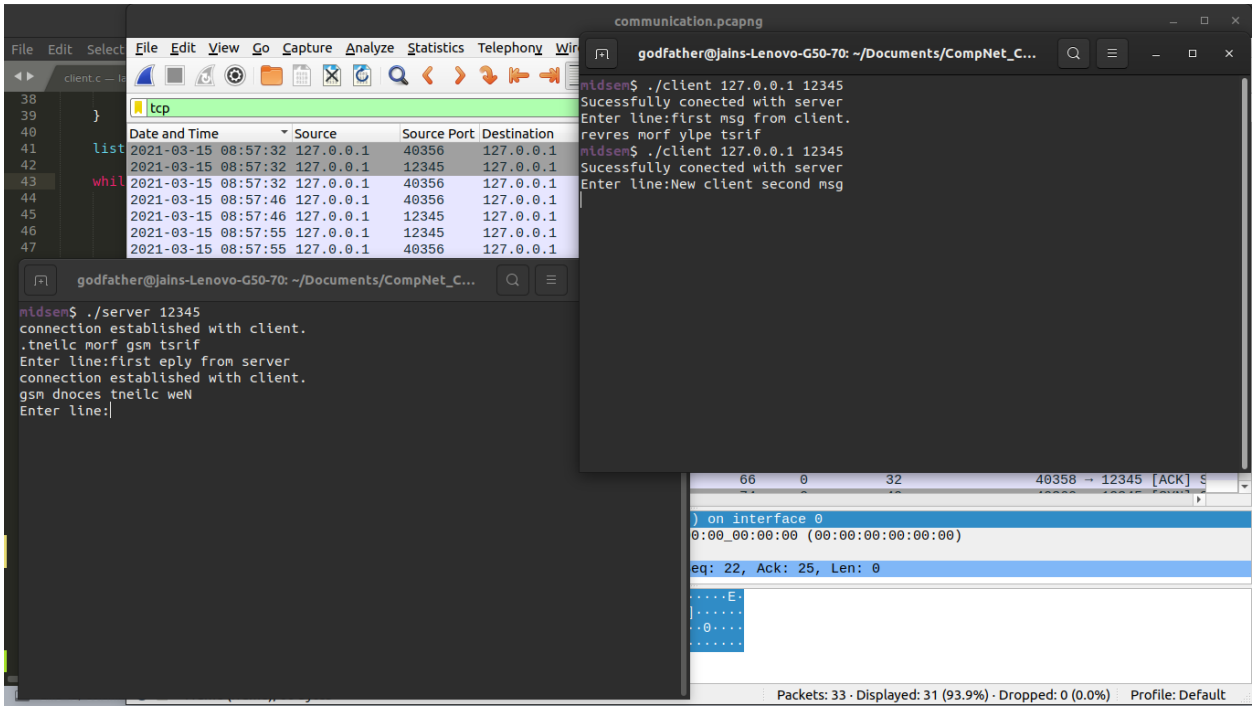
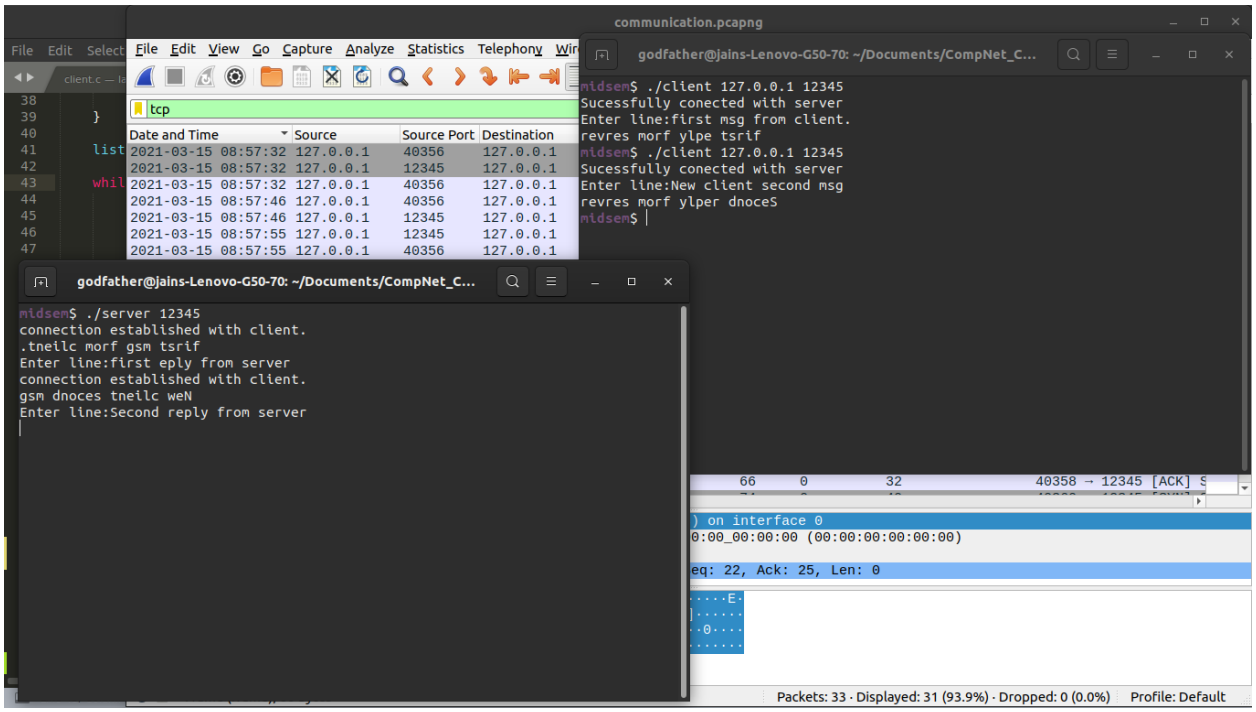
CN MIDSEM

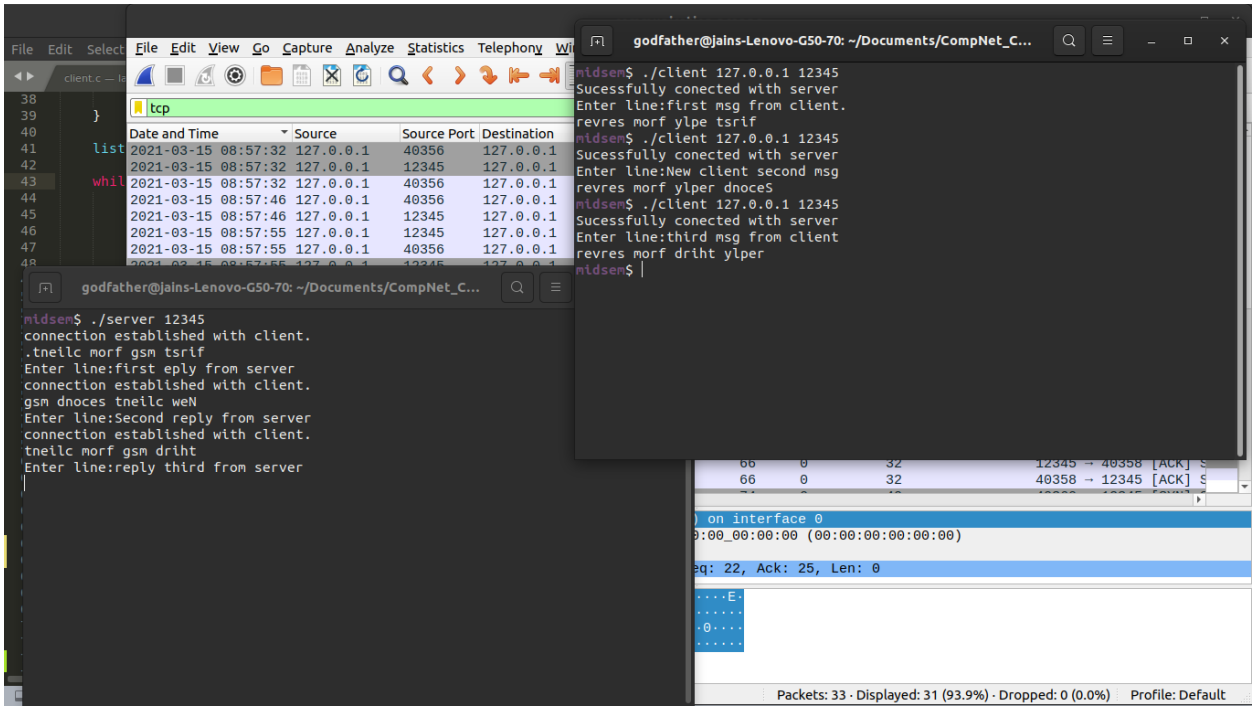
Aditya Jain
2018A7PS0243G





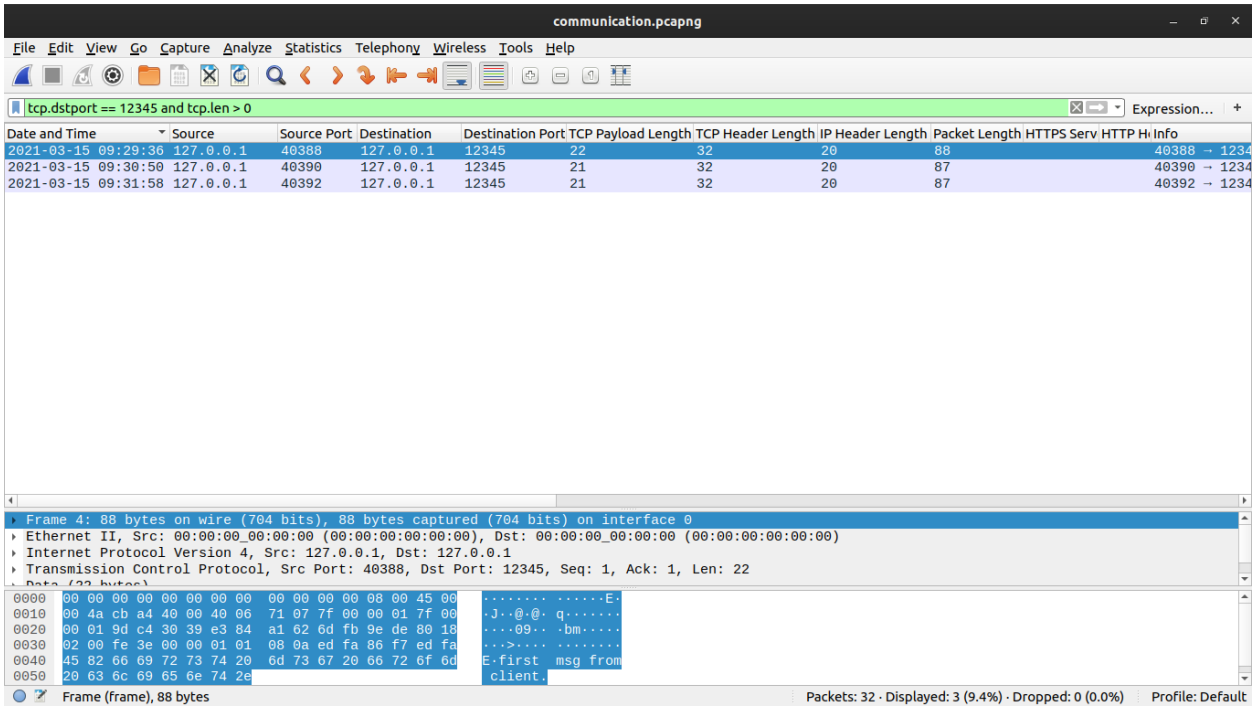

communication.pcapng
File Edit View Go Capture Analyze Statistics Telephony
tcp
Date and Time Source Source Port Destination
2021-03-15 08:57:32 127.0.0.1 40356 127.0.0.1
2021-03-15 08:57:32 127.0.0.1 12345 127.0.0.1
2021-03-15 08:57:32 127.0.0.1 40356 127.0.0.1
2021-03-15 08:57:46 127.0.0.1 40356 127.0.0.1
2021-03-15 08:57:46 127.0.0.1 12345 127.0.0.1
2021-03-15 08:57:55 127.0.0.1 12345 127.0.0.1
2021-03-15 08:57:55 127.0.0.1 40356 127.0.0.1
godfather@jains-Lenovo-G50-70: ~/Documents/CompNet_C...
midsem$ ./server 12345
connection established with client.
.tneilc morf gsm tsrif
Enter line:first eply from server
connection established with client.
gsm dnoces tneilc weN
Enter line:
godfather@jains-Lenovo-G50-70: ~/Documents/CompNet_C...
midsem$ ./client 127.0.0.1 12345
Sucessfully conected with server
Enter line:first msg from client.
revres morf ylpe tsrif
midsem$ ./client 127.0.0.1 12345
Sucessfully conected with server
Enter line:New client second msg
66 0 32 40358 → 12345 [ACK] S
) on interface 0
0:00_00:00:00 (00:00:00:00:00:00)
eq: 22, Ack: 25, Len: 0
Packets: 33 · Displayed: 31 (93.9%) · Dropped: 0 (0.0%) Profile: Default


communication.pcapng
File Edit View Go Capture Analyze Statistics Telephony
tcp
Date and Time Source Source Port Destination
2021-03-15 08:57:32 127.0.0.1 40356 127.0.0.1
2021-03-15 08:57:32 127.0.0.1 12345 127.0.0.1
2021-03-15 08:57:32 127.0.0.1 40356 127.0.0.1
2021-03-15 08:57:46 127.0.0.1 40356 127.0.0.1
2021-03-15 08:57:46 127.0.0.1 12345 127.0.0.1
2021-03-15 08:57:55 127.0.0.1 12345 127.0.0.1
2021-03-15 08:57:55 127.0.0.1 40356 127.0.0.1
godfather@jains-Lenovo-G50-70: ~/Documents/CompNet_C...
midsem$ ./server 12345
connection established with client.
.tneilc morf gsm tsrif
Enter line:first eply from server
connection established with client.
gsm dnoces tneilc weN
Enter line:Second reply from server
godfather@jains-Lenovo-G50-70: ~/Documents/CompNet_C...
midsem$ ./client 127.0.0.1 12345
Sucessfully conected with server
Enter line:first msg from client.
revres morf ylpe tsrif
midsem$ ./client 127.0.0.1 12345
Sucessfully conected with server
Enter line:New client second msg
revres morf ylper dnoceS
midsem$
66 0 32 40358 → 12345 [ACK] S
) on interface 0
0:00_00:00:00 (00:00:00:00:00:00)
eq: 22, Ack: 25, Len: 0
Packets: 33 · Displayed: 31 (93.9%) · Dropped: 0 (0.0%) Profile: Default



These are screenshots from communication between server and client.
Also attached pcap file along which includes all these captured packets.

Next page....



Filter used for client to server with payload size > 0 :
tcp.dstport == 12345 and tcp.len > 0

12345 because it was port given to the server as a command line argument.
Let's say port is X so filter will be:
tcp.dstport == X and tcp.len > 0

tcp.dstport = 12345 gives packets which are sent from client to server that is the destination port is 12345.
tcp.len > 0 gives packets whose payload length(or data length) is greater than 0 that is packets which contain some data in it other than tcp header.

Message sent from client to server is “first msg from client.” which has 22 bytes length and **TCP segment length includes TCP header length(tcp.hdr_len) + TCP payload length(tcp.len).** As we can see TCP payload length from column is 22 for this packet which is the same as length of msg sent. **So the number of bytes in line = 22 and TCP payload length = 22 which is the same**. And the TCP header length is 32. Which makes TCP segment length = 32(**tcp.hdr_len**) + 22(**tcp.len**) = 54 bytes for first packet.
Columns for these can be seen in the screenshot.

Same is case for other packets as well, msg sent from client to server are:
“New client second msg” which has **21 bytes length**.
“third msg from client” which has **21 bytes length**.

As you can also see from screenshots attached above and pcapng file attached.
And so TCP Payload length comes out to be the same as the number of bytes in the message sent which is 21 and 21 respectively. And TCP segment size = TCP header length + TCP payload length = 32 + 21 = 53 bytes for both, second and third messages.

TCP Payload and number of bytes in line is the same because TCP Payload is data that is sent over the connection and that is exactly the message that is being sent from client to server. Same is the case for messages from server to client also.

There is another column as well which talks about ip header length which is 20 in this case. It is not included in the count of tcp segment size.
Total packet length refers to total packet size that was actually received through the wire which includes Ethernet Header(link header) + IP header + TCP Header + TCP Payload(tcp data).
To find IP Header Length column field is set to **ip.hdr_len**.
We can see from the column of packet size that total packet sizes are 88, 87 and 87 respectively.
Which includes 14(link header) + 20(ip header) + 32(tcp header) + 22/21(tcp payload/data) = 88/87(Packet/Frame size).
We can also find ip length which includes IP header + TCP Header + TCP Payload(data) using **ip.len** as column field.