Questions - INDIVIDUAL Assignment (not group)

1. Every protocol needs a name - what do you want to call this chat protocol?

CPP - Chat Platform Protocol

2. Discuss similarities and differences between this protocol (name?) and HTTP. Discuss at least <u>three</u> similarities.

CPP and HTTP are both:

   a. Text-based: Both protocols use ASCII messages to communicate. HTTP can include it as text on a webpage, while CPP uses it in messages.
   b. 3-way Handshakes: In both protocols, both users must first agree to bridge/connect before they can send messages. This is used as a TCP 3-way handshake, where the last ACK can include some data, or as a CPP register-bridge(-chat) exchange, where the last chat includes a message.
   c. Use Headers: Both protocols use headers to get information about the other user which determines some actions. Both include source and destination information in these headers.

3. What is your complete CHAT message? What headers did you use (if any) and how were they used?

```
CHAT
message: {}\r\n
id: {}\r\n
ip: {}\r\n
port: {}\r\n
\r\n
```

I used a message header to send the message itself to the other user. I used ID, IP, and Port information to help the first exchange take place since the WAIT-mode client would need that information on their end.

4. a. At what layer of the TCP/IP protocol stack is the Chat Protocol? Explain your answer.

   a. Application layer since it defines how two running applications exchange messages. It also uses sockets, which is a clear give away that it is running on the application layer since sockets are how applications interact with the transport layer.

   b. Is this application loss-tolerant?

   No, it is not loss-tolerant. It is designed in such a way that Chat messages can only be delivered once before modes are swapped. Additionally, a lost /quit message would leave the other peer waiting.

   c. Could the application have worked with UDP?

   No, the application couldn't work with UDP because it doesn't have internally built acknowledgments or retransmissions. Our current error handling wouldn't do anything because UDP doesn't let us know that an error occurred.

5. Referring to the textbook Section 2.1.1:
    a. Looking over the architecture in this project, would you characterize it as Client-Server or something else? Explain your answer.
        i. This is Peer-Peer architecture because, once the connection is established, the two clients communicate directly, without the server in between them.
    b. When in Chat mode, do the peers operate as clients, servers or both? Explain your answer.
        i. I would say they act as both a client and server because, when establishing the connection, one of them initiates the connection like a client, while the other has already been listening on its port awaiting a connection, similar to a server. During the chat, they swap between their responsibilities.

6. Is the client-server client communication using a persistent or non-persistent TCP connection? Explain your answer.
    a. Non-persistent because we are opening and closing a new TCP connection for each command that needs to be sent to the Server.

7. Is your peer-peer client program using a persistent or non-persistent TCP connection? Explain your answer.
    a. Persistent because we keep the same TCP connection for the entire chat session and only close the connection after /quit or CTRL C.

8. Why do you think the specific type of TCP connection (persistent or non-persistent) was made for each of the two connections, client-server versus peer-peer?
    a. Client-Server used a non-persistent connection because we use it infrequently and only need to perform a couple of commands per user. Peer-Peer used a persistent connection because we are continuously chatting back and forth, which would be slowed significantly by the opening and closing of new TCP connections.

9. Which aspect of your program follows half-duplex communication?

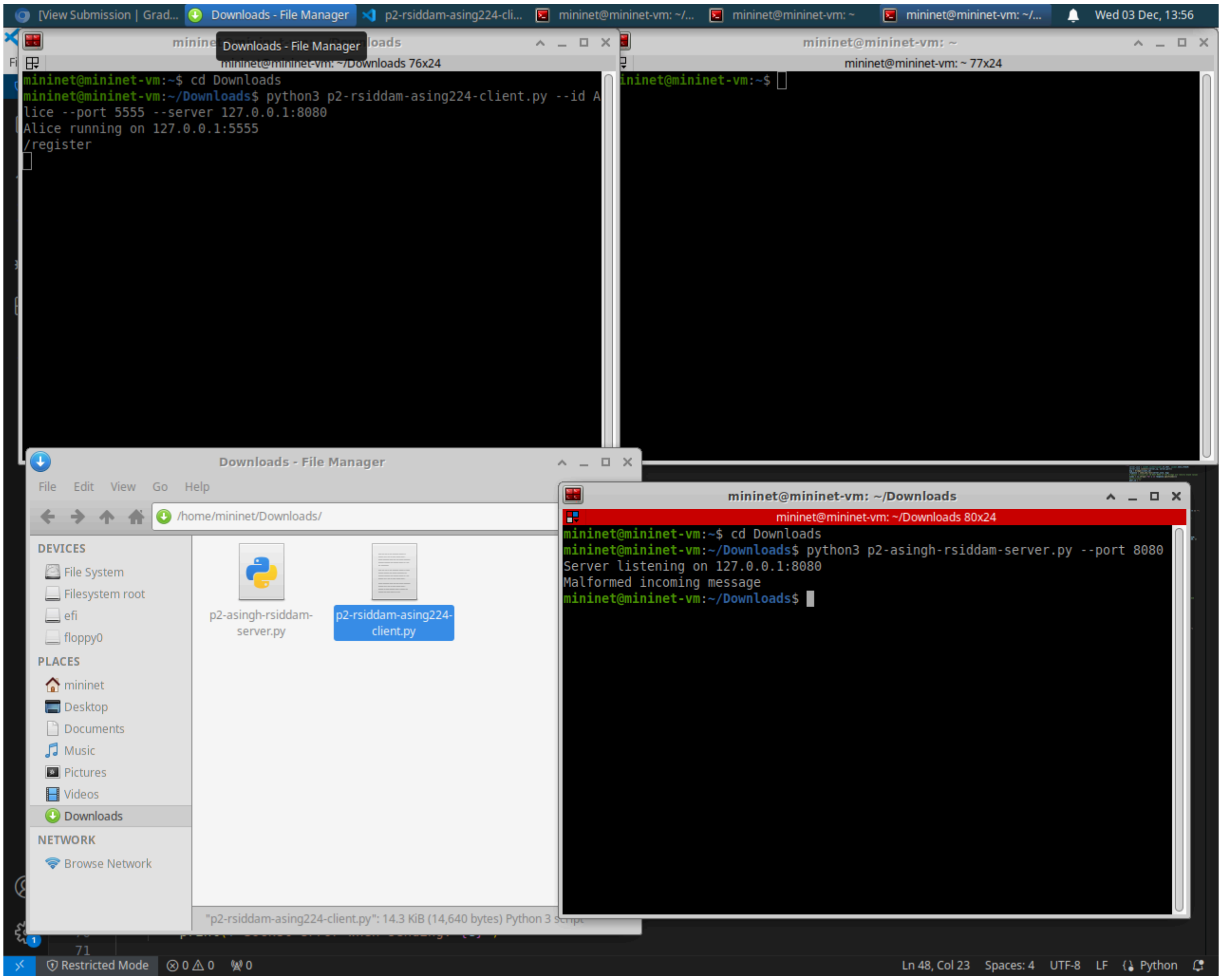
    The turn-based chatting follows half-duplex communication because only one peer is sending at a time while the other is receiving. Both users communicate but only one can do so at a time.

    .

10. If the two clients are chatting on the same host computer, what restrictions are there for the port numbers that can be used by the clients?
    a. The two clients' sockets cannot bind to the same IP:Port combo. Each needs to listen to its own port.

11. Include a screenshot showing **<u>your</u>** server's response to receiving a malformed message from the client.

a. Sent REGISTERs instead of REGISTER



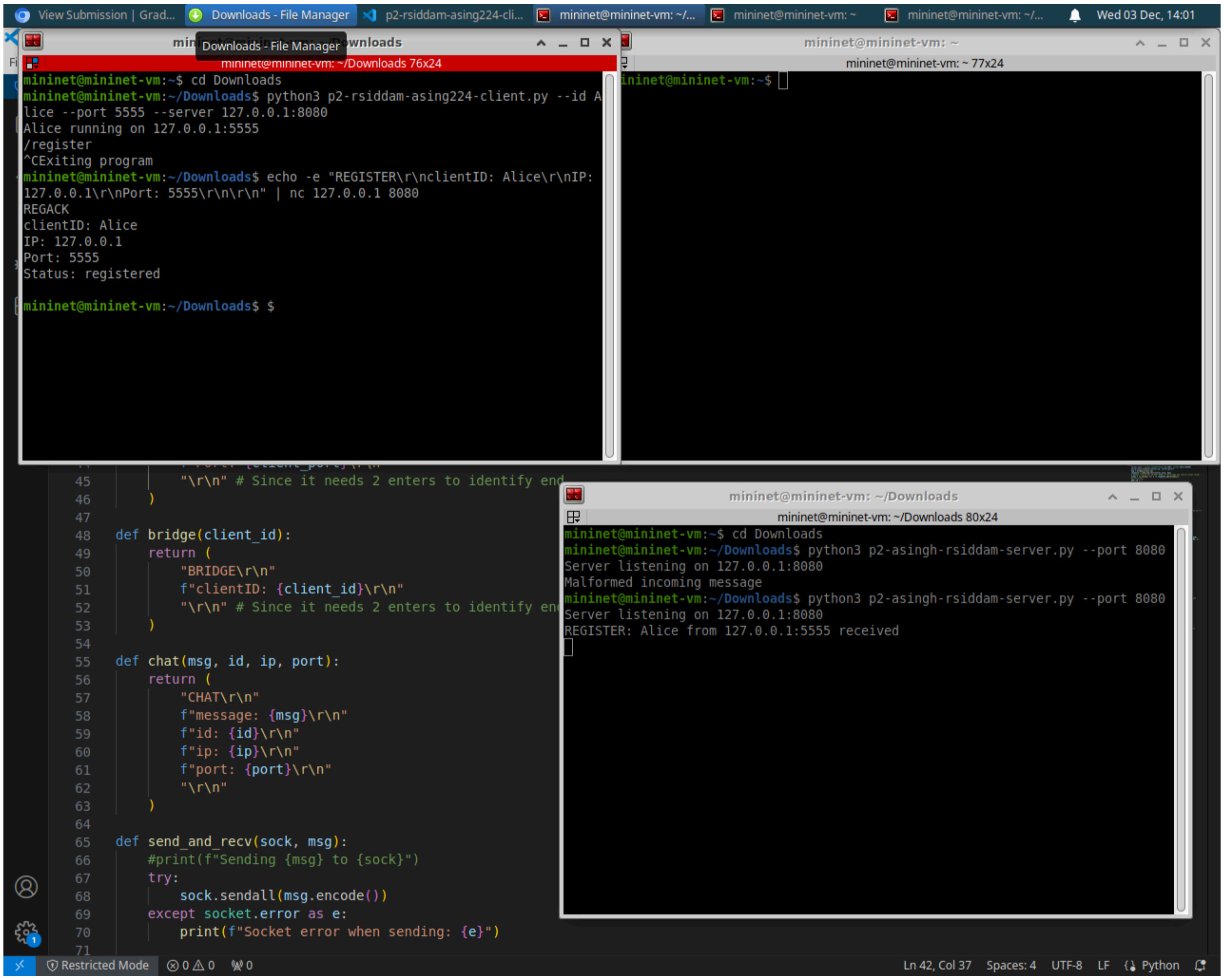
```
mininet@mininet-vm:~$ cd Downloads
mininet@mininet-vm:~/Downloads$ python3 p2-rsiddam-asing224-client.py --id A
lice --port 5555 --server 127.0.0.1:8080
Alice running on 127.0.0.1:5555
/register
```

```
ininet@mininet-vm:~$
```

```
mininet@mininet-vm:~$ cd Downloads
mininet@mininet-vm:~/Downloads$ python3 p2-asingh-rsiddam-server.py --port 8080
Server listening on 127.0.0.1:8080
Malformed incoming message
mininet@mininet-vm:~/Downloads$
```

[View Submission | Grad...
Downloads - File Manager
p2-rsiddam-asing224-cli...
mininet@mininet-vm: ~/...
mininet@mininet-vm: ~
mininet@mininet-vm: ~/...
Wed 03 Dec, 13:56
mininet@mininet-vm: ~/Downloads 76x24
mininet@mininet-vm: ~
mininet@mininet-vm: ~ 77x24
Downloads - File Manager
File Edit View Go Help
/home/mininet/Downloads/
DEVICES
File System
Filesystem root
efi
floppy0
PLACES
mininet
Desktop
Documents
Music
Pictures
Videos
Downloads
NETWORK
Browse Network
p2-asingh-rsiddam-server.py
p2-rsiddam-asing224-client.py
"p2-rsiddam-asing224-client.py": 14.3 KiB (14,640 bytes) Python 3 script
mininet@mininet-vm: ~/Downloads
mininet@mininet-vm: ~/Downloads 80x24
Restricted Mode
Ln 48, Col 23 Spaces: 4 UTF-8 LF Python

12. Send a REGISTER request from your client program to your server program using the ‘**netcat**’ utility. Include screenshots of both terminals showing successful receipt of the message at the server.


View Submission | Grad...
Downloads - File Manager
p2-rsiddam-asing224-cli...
mininet@mininet-vm: ~/...
mininet@mininet-vm: ~
mininet@mininet-vm: ~/...
Wed 03 Dec, 14:01
Downloads - File Manager
mininet@mininet-vm: ~/Downloads 76x24
mininet@mininet-vm: ~
mininet@mininet-vm: ~ 77x24
mininet@mininet-vm: ~/Downloads
mininet@mininet-vm: ~/Downloads 80x24
Restricted Mode
Ln 42, Col 37
Spaces: 4
UTF-8
LF
Python

```
mininet@mininet-vm:~$ cd Downloads
mininet@mininet-vm:~/Downloads$ python3 p2-rsiddam-asing224-client.py --id A
lice --port 5555 --server 127.0.0.1:8080
Alice running on 127.0.0.1:5555
/register
^CExiting program
mininet@mininet-vm:~/Downloads$ echo -e "REGISTER\r\nclientID: Alice\r\nIP:
127.0.0.1\r\nPort: 5555\r\n\r\n" | nc 127.0.0.1 8080
REGACK
clientID: Alice
IP: 127.0.0.1
Port: 5555
Status: registered

mininet@mininet-vm:~/Downloads$ $
```

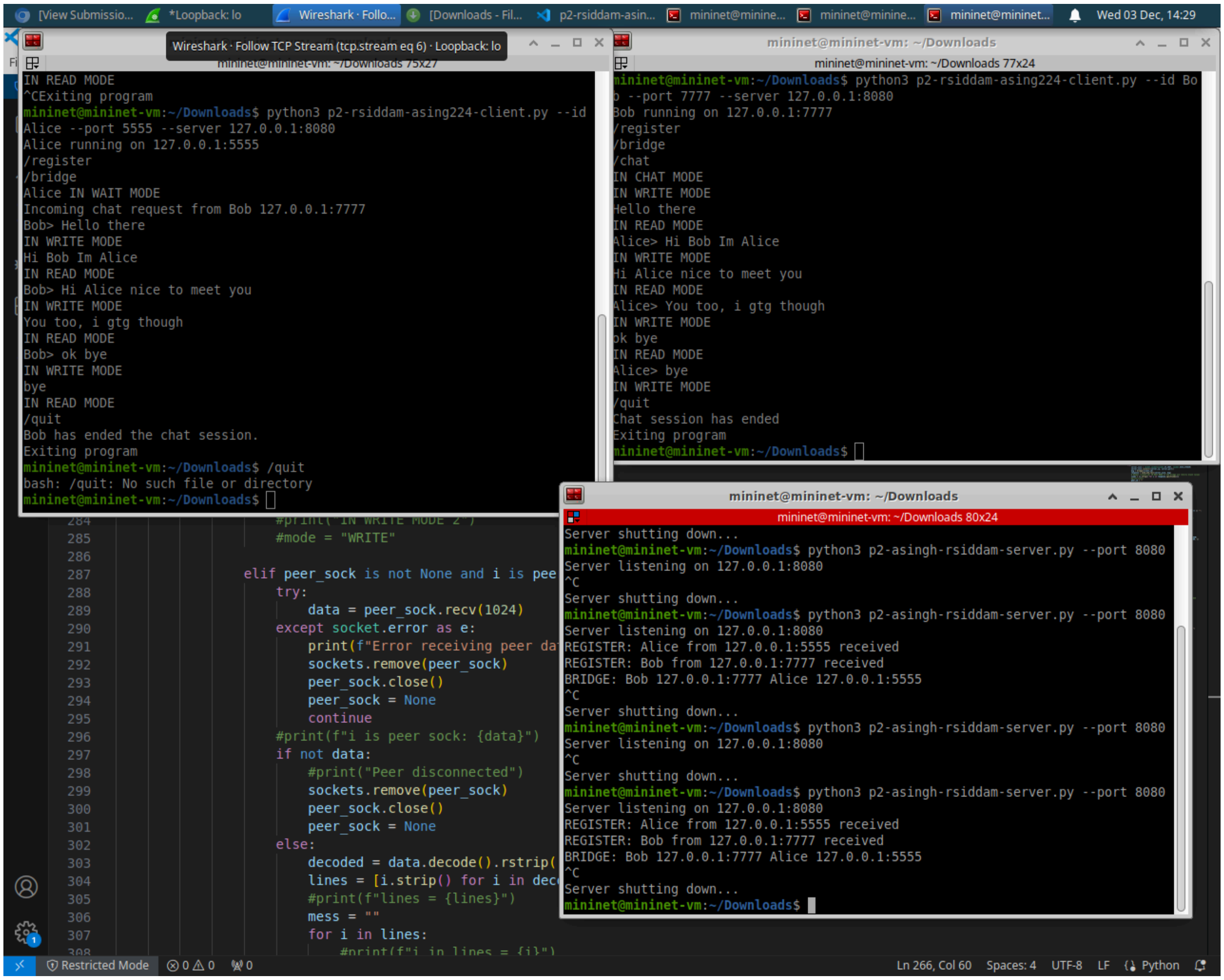
```
ininet@mininet-vm:~$
```

```
        "\r\n" # Since it needs 2 enters to identify end
    )

def bridge(client_id):
    return (
        "BRIDGE\r\n"
        f"clientID: {client_id}\r\n"
        "\r\n" # Since it needs 2 enters to identify en
    )

def chat(msg, id, ip, port):
    return (
        "CHAT\r\n"
        f"message: {msg}\r\n"
        f"id: {id}\r\n"
        f"ip: {ip}\r\n"
        f"port: {port}\r\n"
        "\r\n"
    )

def send_and_recv(sock, msg):
    #print(f"Sending {msg} to {sock}")
    try:
        sock.sendall(msg.encode())
    except socket.error as e:
        print(f"Socket error when sending: {e}")
```

```
mininet@mininet-vm:~$ cd Downloads
mininet@mininet-vm:~/Downloads$ python3 p2-asingh-rsiddam-server.py --port 8080
Server listening on 127.0.0.1:8080
Malformed incoming message
mininet@mininet-vm:~/Downloads$ python3 p2-asingh-rsiddam-server.py --port 8080
Server listening on 127.0.0.1:8080
REGISTER: Alice from 127.0.0.1:5555 received
```

13. Include screenshot(s) showing <u>all three terminals</u>, clearly displaying:

- Start up of your server
- Chat exchange of at least 3 lines between peers
- Wireshark capture of this conversation (might require 2 screenshots)
    - Circle in RED the TCP handshake and all TCP segments exchanged during your chat exchange.
    - Circle in BLUE the messages sent within the TCP payload.   This is similar to how you are able to see Alice in Wonderland text in Lab 6.

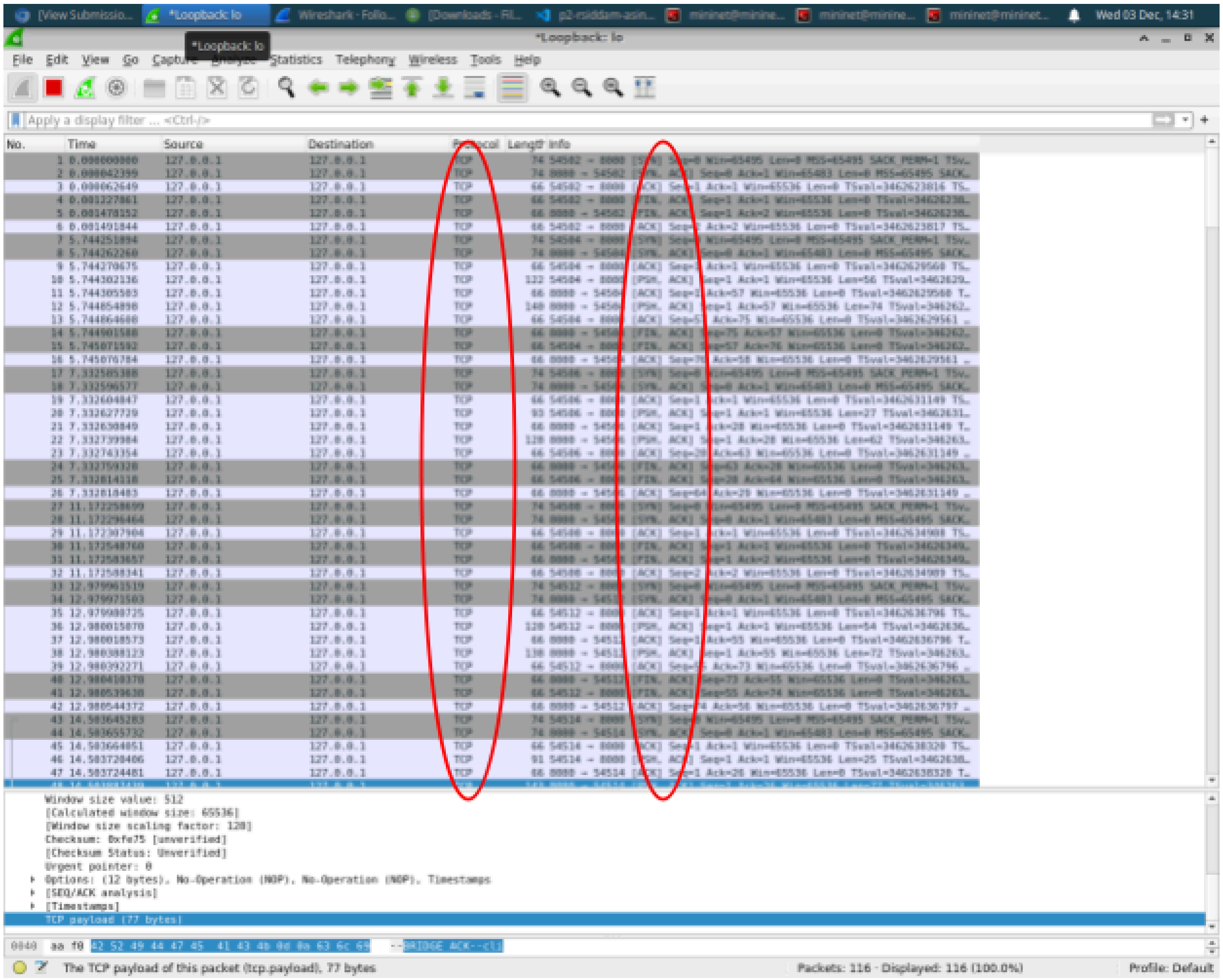

[View Submissio...
*Loopback: lo
Wireshark · Follo...
[Downloads - Fil...
p2-rsiddam-asin...
mininet@minine...
mininet@minine...
mininet@mininet...
Wed 03 Dec, 14:29
Wireshark · Follow TCP Stream (tcp.stream eq 6) · Loopback: lo
mininet@mininet-vm: ~/Downloads 75x27
mininet@mininet-vm: ~/Downloads
mininet@mininet-vm: ~/Downloads 77x24
mininet@mininet-vm: ~/Downloads
mininet@mininet-vm: ~/Downloads 80x24
Restricted Mode
0 ⚠ 0
0
Ln 266, Col 60
Spaces: 4
UTF-8
LF
Python

```
IN READ MODE
^CExiting program
mininet@mininet-vm:~/Downloads$ python3 p2-rsiddam-asing224-client.py --id
Alice --port 5555 --server 127.0.0.1:8080
Alice running on 127.0.0.1:5555
/register
/bridge
Alice IN WAIT MODE
Incoming chat request from Bob 127.0.0.1:7777
Bob> Hello there
IN WRITE MODE
Hi Bob Im Alice
IN READ MODE
Bob> Hi Alice nice to meet you
IN WRITE MODE
You too, i gtg though
IN READ MODE
Bob> ok bye
IN WRITE MODE
bye
IN READ MODE
/quit
Bob has ended the chat session.
Exiting program
mininet@mininet-vm:~/Downloads$ /quit
bash: /quit: No such file or directory
mininet@mininet-vm:~/Downloads$ 
```

```
mininet@mininet-vm:~/Downloads$ python3 p2-rsiddam-asing224-client.py --id Bo
b --port 7777 --server 127.0.0.1:8080
Bob running on 127.0.0.1:7777
/register
/bridge
/chat
IN CHAT MODE
IN WRITE MODE
Hello there
IN READ MODE
Alice> Hi Bob Im Alice
IN WRITE MODE
Hi Alice nice to meet you
IN READ MODE
Alice> You too, i gtg though
IN WRITE MODE
ok bye
IN READ MODE
Alice> bye
IN WRITE MODE
/quit
Chat session has ended
Exiting program
mininet@mininet-vm:~/Downloads$ 
```

```
Server shutting down...
mininet@mininet-vm:~/Downloads$ python3 p2-asingh-rsiddam-server.py --port 8080
Server listening on 127.0.0.1:8080
^C
Server shutting down...
mininet@mininet-vm:~/Downloads$ python3 p2-asingh-rsiddam-server.py --port 8080
Server listening on 127.0.0.1:8080
REGISTER: Alice from 127.0.0.1:5555 received
REGISTER: Bob from 127.0.0.1:7777 received
BRIDGE: Bob 127.0.0.1:7777 Alice 127.0.0.1:5555
^C
Server shutting down...
mininet@mininet-vm:~/Downloads$ python3 p2-asingh-rsiddam-server.py --port 8080
Server listening on 127.0.0.1:8080
^C
Server shutting down...
mininet@mininet-vm:~/Downloads$ python3 p2-asingh-rsiddam-server.py --port 8080
Server listening on 127.0.0.1:8080
REGISTER: Alice from 127.0.0.1:5555 received
REGISTER: Bob from 127.0.0.1:7777 received
BRIDGE: Bob 127.0.0.1:7777 Alice 127.0.0.1:5555
^C
Server shutting down...
mininet@mininet-vm:~/Downloads$ 
```

```
    #print("IN WRITE MODE 2")
    #mode = "WRITE"

elif peer_sock is not None and i is pee
    try:
        data = peer_sock.recv(1024)
    except socket.error as e:
        print(f"Error receiving peer da
        sockets.remove(peer_sock)
        peer_sock.close()
        peer_sock = None
        continue
    #print(f"i is peer sock: {data}")
    if not data:
        #print("Peer disconnected")
        sockets.remove(peer_sock)
        peer_sock.close()
        peer_sock = None
    else:
        decoded = data.decode().rstrip(
        lines = [i.strip() for i in dec
        #print(f"lines = {lines}")
        mess = ""
        for i in lines:
            #print(f"i in lines = {i}")
```



```
No.  Time          Source      Destination  Protocol Length Info
   1 0.000000000   127.0.0.1   127.0.0.1    TCP      74 54502 → 8000 [SYN] Seq=0 Win=65495 Len=0 MSS=65495 SACK_PERM=1 TSv…
   2 0.000042399   127.0.0.1   127.0.0.1    TCP      74 8000 → 54502 [SYN, ACK] Seq=0 Ack=1 Win=65483 Len=0 MSS=65495 SACK…
   3 0.000062649   127.0.0.1   127.0.0.1    TCP      66 54502 → 8000 [ACK] Seq=1 Ack=1 Win=65536 Len=0 TSval=3462623816 TS…
   4 0.001227861   127.0.0.1   127.0.0.1    TCP      66 54502 → 8000 [FIN, ACK] Seq=1 Ack=1 Win=65536 Len=0 TSval=34626238…
   5 0.001478152   127.0.0.1   127.0.0.1    TCP      66 8000 → 54502 [FIN, ACK] Seq=1 Ack=2 Win=65536 Len=0 TSval=34626238…
   6 0.001491844   127.0.0.1   127.0.0.1    TCP      66 54502 → 8000 [ACK] Seq=2 Ack=2 Win=65536 Len=0 TSval=3462623817 TS…
   7 5.744251894   127.0.0.1   127.0.0.1    TCP      74 54504 → 8000 [SYN] Seq=0 Win=65495 Len=0 MSS=65495 SACK_PERM=1 TSv…
   8 5.744262260   127.0.0.1   127.0.0.1    TCP      74 8000 → 54504 [SYN, ACK] Seq=0 Ack=1 Win=65483 Len=0 MSS=65495 SACK…
   9 5.744270675   127.0.0.1   127.0.0.1    TCP      66 54504 → 8000 [ACK] Seq=1 Ack=1 Win=65536 Len=0 TSval=3462629560 TS…
  10 5.744302136   127.0.0.1   127.0.0.1    TCP     122 54504 → 8000 [PSH, ACK] Seq=1 Ack=1 Win=65536 Len=56 TSval=3462629…
  11 5.744305503   127.0.0.1   127.0.0.1    TCP      66 8000 → 54504 [ACK] Seq=1 Ack=57 Win=65536 Len=0 TSval=3462629560 T…
  12 5.744854898   127.0.0.1   127.0.0.1    TCP     140 8000 → 54504 [PSH, ACK] Seq=1 Ack=57 Win=65536 Len=74 TSval=346262…
  13 5.744864600   127.0.0.1   127.0.0.1    TCP      66 54504 → 8000 [ACK] Seq=57 Ack=75 Win=65536 Len=0 TSval=3462629561 …
  14 5.744901588   127.0.0.1   127.0.0.1    TCP      66 8000 → 54504 [FIN, ACK] Seq=75 Ack=57 Win=65536 Len=0 TSval=346262…
  15 5.745071592   127.0.0.1   127.0.0.1    TCP      66 54504 → 8000 [FIN, ACK] Seq=57 Ack=76 Win=65536 Len=0 TSval=346262…
  16 5.745076784   127.0.0.1   127.0.0.1    TCP      66 8000 → 54504 [ACK] Seq=76 Ack=58 Win=65536 Len=0 TSval=3462629561 …
  17 7.332585388   127.0.0.1   127.0.0.1    TCP      74 54506 → 8000 [SYN] Seq=0 Win=65495 Len=0 MSS=65495 SACK_PERM=1 TSv…
  18 7.332596577   127.0.0.1   127.0.0.1    TCP      74 8000 → 54506 [SYN, ACK] Seq=0 Ack=1 Win=65483 Len=0 MSS=65495 SACK…
  19 7.332604847   127.0.0.1   127.0.0.1    TCP      66 54506 → 8000 [ACK] Seq=1 Ack=1 Win=65536 Len=0 TSval=3462631149 TS…
  20 7.332627729   127.0.0.1   127.0.0.1    TCP      93 54506 → 8000 [PSH, ACK] Seq=1 Ack=1 Win=65536 Len=27 TSval=3462631…
  21 7.332630849   127.0.0.1   127.0.0.1    TCP      66 8000 → 54506 [ACK] Seq=1 Ack=28 Win=65536 Len=0 TSval=3462631149 T…
  22 7.332739984   127.0.0.1   127.0.0.1    TCP     128 8000 → 54506 [PSH, ACK] Seq=1 Ack=28 Win=65536 Len=62 TSval=346263…
  23 7.332743354   127.0.0.1   127.0.0.1    TCP      66 54506 → 8000 [ACK] Seq=28 Ack=63 Win=65536 Len=0 TSval=3462631149 …
  24 7.332759328   127.0.0.1   127.0.0.1    TCP      66 8000 → 54506 [FIN, ACK] Seq=63 Ack=28 Win=65536 Len=0 TSval=346263…
  25 7.332814118   127.0.0.1   127.0.0.1    TCP      66 54506 → 8000 [FIN, ACK] Seq=28 Ack=64 Win=65536 Len=0 TSval=346263…
  26 7.332818483   127.0.0.1   127.0.0.1    TCP      66 8000 → 54506 [ACK] Seq=64 Ack=29 Win=65536 Len=0 TSval=3462631149 …
  27 11.172258699  127.0.0.1   127.0.0.1    TCP      74 54508 → 8000 [SYN] Seq=0 Win=65495 Len=0 MSS=65495 SACK_PERM=1 TSv…
  28 11.172296464  127.0.0.1   127.0.0.1    TCP      74 8000 → 54508 [SYN, ACK] Seq=0 Ack=1 Win=65483 Len=0 MSS=65495 SACK…
  29 11.172307904  127.0.0.1   127.0.0.1    TCP      66 54508 → 8000 [ACK] Seq=1 Ack=1 Win=65536 Len=0 TSval=3462634908 TS…
  30 11.172548760  127.0.0.1   127.0.0.1    TCP      66 54508 → 8000 [FIN, ACK] Seq=1 Ack=1 Win=65536 Len=0 TSval=34626349…
  31 11.172583657  127.0.0.1   127.0.0.1    TCP      66 8000 → 54508 [FIN, ACK] Seq=1 Ack=2 Win=65536 Len=0 TSval=34626349…
  32 11.172588341  127.0.0.1   127.0.0.1    TCP      66 54508 → 8000 [ACK] Seq=2 Ack=2 Win=65536 Len=0 TSval=3462634909 TS…
  33 12.979961519  127.0.0.1   127.0.0.1    TCP      74 54512 → 8000 [SYN] Seq=0 Win=65495 Len=0 MSS=65495 SACK_PERM=1 TSv…
  34 12.979971503  127.0.0.1   127.0.0.1    TCP      74 8000 → 54512 [SYN, ACK] Seq=0 Ack=1 Win=65483 Len=0 MSS=65495 SACK…
  35 12.979980725  127.0.0.1   127.0.0.1    TCP      66 54512 → 8000 [ACK] Seq=1 Ack=1 Win=65536 Len=0 TSval=3462636796 TS…
  36 12.980015870  127.0.0.1   127.0.0.1    TCP     120 54512 → 8000 [PSH, ACK] Seq=1 Ack=1 Win=65536 Len=54 TSval=3462636…
  37 12.980018573  127.0.0.1   127.0.0.1    TCP      66 8000 → 54512 [ACK] Seq=1 Ack=55 Win=65536 Len=0 TSval=3462636796 T…
  38 12.980388123  127.0.0.1   127.0.0.1    TCP     138 8000 → 54512 [PSH, ACK] Seq=1 Ack=55 Win=65536 Len=72 TSval=346263…
  39 12.980392271  127.0.0.1   127.0.0.1    TCP      66 54512 → 8000 [ACK] Seq=55 Ack=73 Win=65536 Len=0 TSval=3462636796 …
  40 12.980410370  127.0.0.1   127.0.0.1    TCP      66 8000 → 54512 [FIN, ACK] Seq=73 Ack=55 Win=65536 Len=0 TSval=346263…
  41 12.980539630  127.0.0.1   127.0.0.1    TCP      66 54512 → 8000 [FIN, ACK] Seq=55 Ack=74 Win=65536 Len=0 TSval=346263…
  42 12.980544372  127.0.0.1   127.0.0.1    TCP      66 8000 → 54512 [ACK] Seq=74 Ack=56 Win=65536 Len=0 TSval=3462636797 …
  43 14.503645283  127.0.0.1   127.0.0.1    TCP      74 54514 → 8000 [SYN] Seq=0 Win=65495 Len=0 MSS=65495 SACK_PERM=1 TSv…
  44 14.503655732  127.0.0.1   127.0.0.1    TCP      74 8000 → 54514 [SYN, ACK] Seq=0 Ack=1 Win=65483 Len=0 MSS=65495 SACK…
  45 14.503664051  127.0.0.1   127.0.0.1    TCP      66 54514 → 8000 [ACK] Seq=1 Ack=1 Win=65536 Len=0 TSval=3462638320 TS…
  46 14.503720406  127.0.0.1   127.0.0.1    TCP      91 54514 → 8000 [PSH, ACK] Seq=1 Ack=1 Win=65536 Len=25 TSval=3462638…
  47 14.503724481  127.0.0.1   127.0.0.1    TCP      66 8000 → 54514 [ACK] Seq=1 Ack=26 Win=65536 Len=0 TSval=3462638320 T…
```

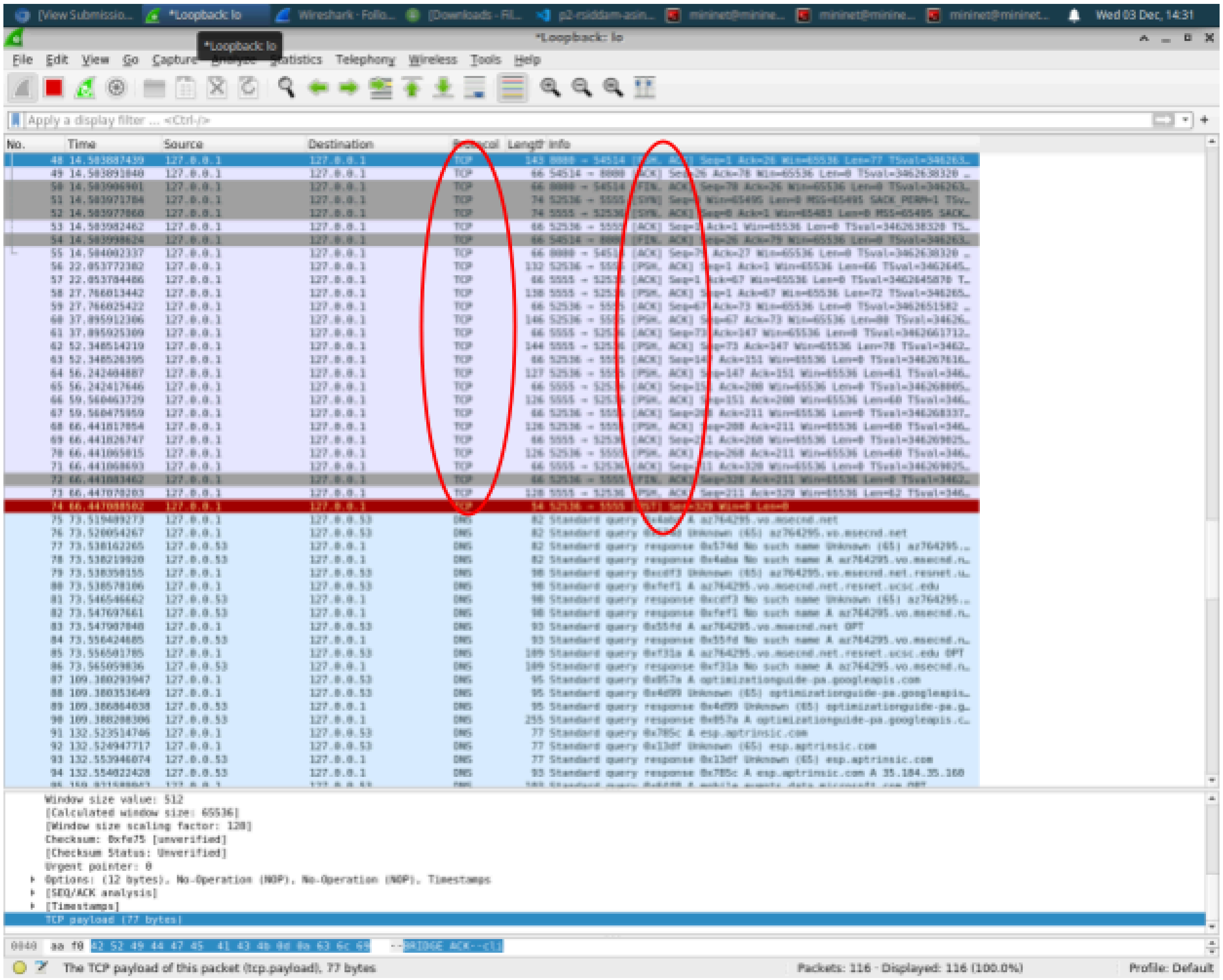
```
    Window size value: 512
    [Calculated window size: 65536]
    [Window size scaling factor: 128]
    Checksum: 0xfe75 [unverified]
    [Checksum Status: Unverified]
    Urgent pointer: 0
  ▸ Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
  ▸ [SEQ/ACK analysis]
  ▸ [Timestamps]
    TCP payload (77 bytes)
```

```
0040  aa f0 42 52 49 44 47 45  41 43 4b 0d 0a 63 6c 69   ··BRIDGE ACK··cli
```

The TCP payload of this packet (tcp.payload), 77 bytes

Packets: 116 · Displayed: 116 (100.0%)

Profile: Default

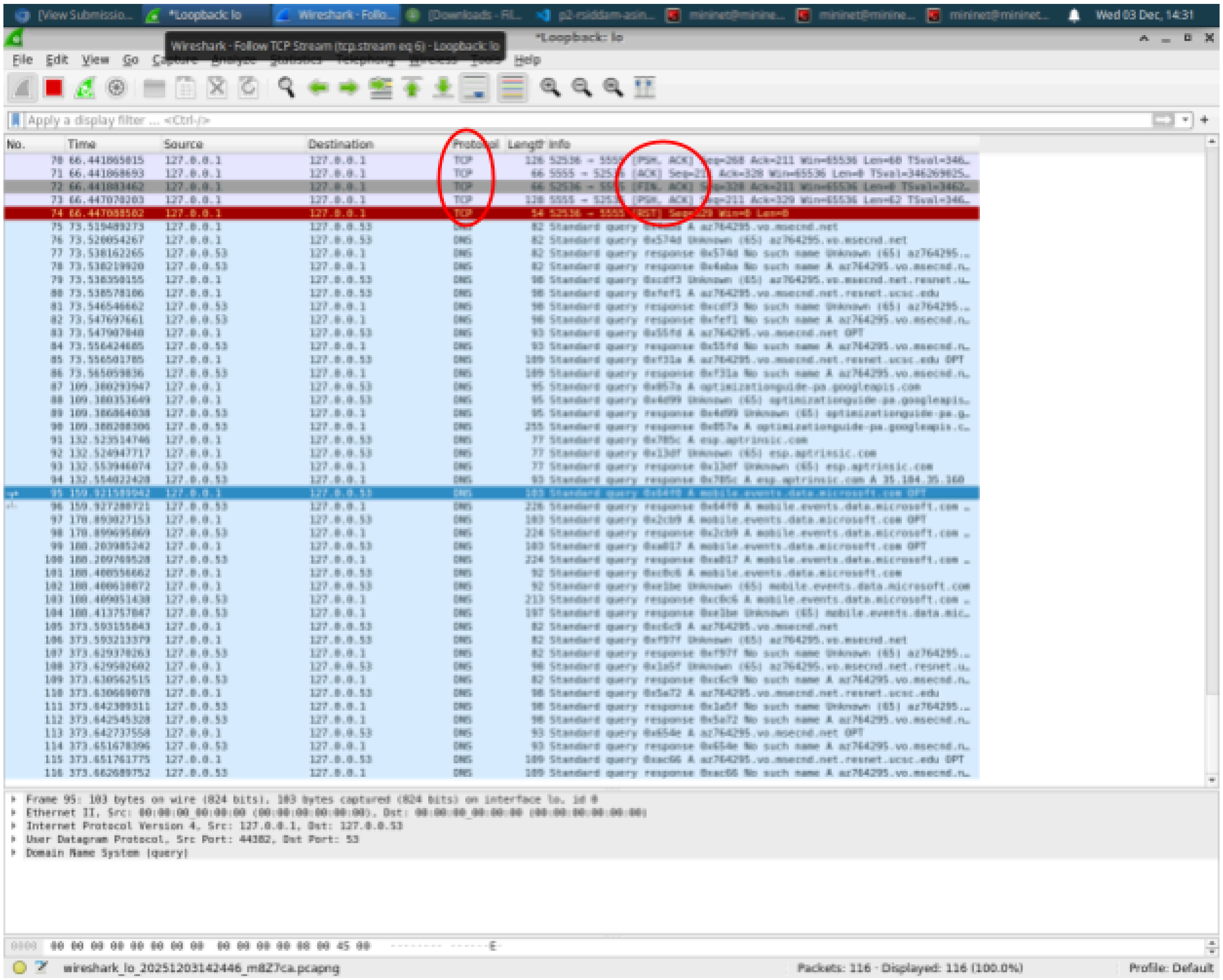


```
No.   Time           Source        Destination   Protocol Length Info
   48 14.583887439   127.0.0.1     127.0.0.1     TCP      143 8080 → 54514 [PSH, ACK] Seq=1 Ack=26 Win=65536 Len=77 TSval=346263…
   49 14.583891840   127.0.0.1     127.0.0.1     TCP       66 54514 → 8080 [ACK] Seq=26 Ack=78 Win=65536 Len=0 TSval=3462638320 …
   50 14.583906901   127.0.0.1     127.0.0.1     TCP       66 8080 → 54514 [FIN, ACK] Seq=78 Ack=26 Win=65536 Len=0 TSval=346263…
   51 14.583971784   127.0.0.1     127.0.0.1     TCP       74 52536 → 5555 [SYN] Seq=0 Win=65495 Len=0 MSS=65495 SACK_PERM=1 TSv…
   52 14.583977860   127.0.0.1     127.0.0.1     TCP       74 5555 → 52536 [SYN, ACK] Seq=0 Ack=1 Win=65483 Len=0 MSS=65495 SACK…
   53 14.583982462   127.0.0.1     127.0.0.1     TCP       66 52536 → 5555 [ACK] Seq=1 Ack=1 Win=65536 Len=0 TSval=3462638320 TS…
   54 14.583998624   127.0.0.1     127.0.0.1     TCP       66 54514 → 8080 [FIN, ACK] Seq=26 Ack=79 Win=65536 Len=0 TSval=346263…
   55 14.584002337   127.0.0.1     127.0.0.1     TCP       66 8080 → 54514 [ACK] Seq=79 Ack=27 Win=65536 Len=0 TSval=3462638320 …
   56 22.053772382   127.0.0.1     127.0.0.1     TCP      132 52536 → 5555 [PSH, ACK] Seq=1 Ack=1 Win=65536 Len=66 TSval=3462645…
   57 22.053784486   127.0.0.1     127.0.0.1     TCP       66 5555 → 52536 [ACK] Seq=1 Ack=67 Win=65536 Len=0 TSval=3462645870 T…
   58 27.766013442   127.0.0.1     127.0.0.1     TCP      138 5555 → 52536 [PSH, ACK] Seq=1 Ack=67 Win=65536 Len=72 TSval=346265…
   59 27.766025422   127.0.0.1     127.0.0.1     TCP       66 52536 → 5555 [ACK] Seq=67 Ack=73 Win=65536 Len=0 TSval=3462651582 …
   60 37.895912306   127.0.0.1     127.0.0.1     TCP      146 52536 → 5555 [PSH, ACK] Seq=67 Ack=73 Win=65536 Len=80 TSval=34626…
   61 37.895925309   127.0.0.1     127.0.0.1     TCP       66 5555 → 52536 [ACK] Seq=73 Ack=147 Win=65536 Len=0 TSval=3462661712…
   62 52.348514219   127.0.0.1     127.0.0.1     TCP      144 5555 → 52536 [PSH, ACK] Seq=73 Ack=147 Win=65536 Len=78 TSval=3462…
   63 52.348526395   127.0.0.1     127.0.0.1     TCP       66 52536 → 5555 [ACK] Seq=147 Ack=151 Win=65536 Len=0 TSval=346267616…
   64 56.242404887   127.0.0.1     127.0.0.1     TCP      127 52536 → 5555 [PSH, ACK] Seq=147 Ack=151 Win=65536 Len=61 TSval=346…
   65 56.242417646   127.0.0.1     127.0.0.1     TCP       66 5555 → 52536 [ACK] Seq=151 Ack=208 Win=65536 Len=0 TSval=346268005…
   66 59.560463729   127.0.0.1     127.0.0.1     TCP      126 5555 → 52536 [PSH, ACK] Seq=151 Ack=208 Win=65536 Len=60 TSval=346…
   67 59.560475959   127.0.0.1     127.0.0.1     TCP       66 52536 → 5555 [ACK] Seq=208 Ack=211 Win=65536 Len=0 TSval=346268337…
   68 66.441017854   127.0.0.1     127.0.0.1     TCP      126 52536 → 5555 [PSH, ACK] Seq=208 Ack=211 Win=65536 Len=60 TSval=346…
   69 66.441026747   127.0.0.1     127.0.0.1     TCP       66 5555 → 52536 [ACK] Seq=211 Ack=268 Win=65536 Len=0 TSval=346269025…
   70 66.441065015   127.0.0.1     127.0.0.1     TCP      126 52536 → 5555 [PSH, ACK] Seq=268 Ack=211 Win=65536 Len=60 TSval=346…
   71 66.441068693   127.0.0.1     127.0.0.1     TCP       66 5555 → 52536 [ACK] Seq=211 Ack=328 Win=65536 Len=0 TSval=346269025…
   72 66.441083462   127.0.0.1     127.0.0.1     TCP       66 52536 → 5555 [FIN, ACK] Seq=328 Ack=211 Win=65536 Len=0 TSval=3462…
   73 66.447070203   127.0.0.1     127.0.0.1     TCP      128 5555 → 52536 [PSH, ACK] Seq=211 Ack=329 Win=65536 Len=62 TSval=346…
   74 66.447088502   127.0.0.1     127.0.0.1     TCP       54 52536 → 5555 [RST] Seq=329 Win=0 Len=0
   75 73.519409273   127.0.0.1     127.0.0.53    DNS       82 Standard query 0x4aba A az764295.vo.msecnd.net
   76 73.520054267   127.0.0.1     127.0.0.53    DNS       82 Standard query 0x574d Unknown (65) az764295.vo.msecnd.net
   77 73.538162265   127.0.0.53    127.0.0.1     DNS       82 Standard query response 0x574d No such name Unknown (65) az764295.…
   78 73.538219920   127.0.0.53    127.0.0.1     DNS       82 Standard query response 0x4aba No such name A az764295.vo.msecnd.n…
   79 73.538350155   127.0.0.1     127.0.0.53    DNS       98 Standard query 0xcdf3 Unknown (65) az764295.vo.msecnd.net.resnet.u…
   80 73.538578106   127.0.0.1     127.0.0.53    DNS       98 Standard query 0xfef1 A az764295.vo.msecnd.net.resnet.ucsc.edu
   81 73.546546662   127.0.0.53    127.0.0.1     DNS       98 Standard query response 0xcdf3 No such name Unknown (65) az764295.…
   82 73.547697661   127.0.0.53    127.0.0.1     DNS       98 Standard query response 0xfef1 No such name A az764295.vo.msecnd.n…
   83 73.547907848   127.0.0.1     127.0.0.53    DNS       93 Standard query 0x55fd A az764295.vo.msecnd.net OPT
   84 73.556424685   127.0.0.53    127.0.0.1     DNS       93 Standard query response 0x55fd No such name A az764295.vo.msecnd.n…
   85 73.556501785   127.0.0.1     127.0.0.53    DNS      109 Standard query 0xf31a A az764295.vo.msecnd.net.resnet.ucsc.edu OPT
   86 73.565059836   127.0.0.53    127.0.0.1     DNS      109 Standard query response 0xf31a No such name A az764295.vo.msecnd.n…
   87 109.380293947  127.0.0.1     127.0.0.53    DNS       95 Standard query 0x057a A optimizationguide-pa.googleapis.com
   88 109.380353649  127.0.0.1     127.0.0.53    DNS       95 Standard query 0x4d99 Unknown (65) optimizationguide-pa.googleapis…
   89 109.386864038  127.0.0.53    127.0.0.1     DNS       95 Standard query response 0x4d99 Unknown (65) optimizationguide-pa.g…
   90 109.388208306  127.0.0.53    127.0.0.1     DNS      255 Standard query response 0x057a A optimizationguide-pa.googleapis.c…
   91 132.523514746  127.0.0.1     127.0.0.53    DNS       77 Standard query 0x785c A esp.aptrinsic.com
   92 132.524947717  127.0.0.1     127.0.0.53    DNS       77 Standard query 0x13df Unknown (65) esp.aptrinsic.com
   93 132.553946074  127.0.0.53    127.0.0.1     DNS       77 Standard query response 0x13df Unknown (65) esp.aptrinsic.com
   94 132.554022428  127.0.0.53    127.0.0.1     DNS       93 Standard query response 0x785c A esp.aptrinsic.com A 35.184.35.160
   95 159.921588062  127.0.0.1     127.0.0.53    DNS      103 Standard query 0x6420 A mobile.events.data.microsoft.com OPT
```

```
      Window size value: 512
      [Calculated window size: 65536]
      [Window size scaling factor: 128]
      Checksum: 0xfe75 [unverified]
      [Checksum Status: Unverified]
      Urgent pointer: 0
    ▸ Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
    ▸ [SEQ/ACK analysis]
    ▸ [Timestamps]
      TCP payload (77 bytes)
```

```
0040  aa f0 42 52 49 44 47 45  41 43 4b 0d 0a 63 6c 69   ··BRIDGE ACK··cli
```

The TCP payload of this packet (tcp.payload), 77 bytes   Packets: 116 · Displayed: 116 (100.0%)   Profile: Default

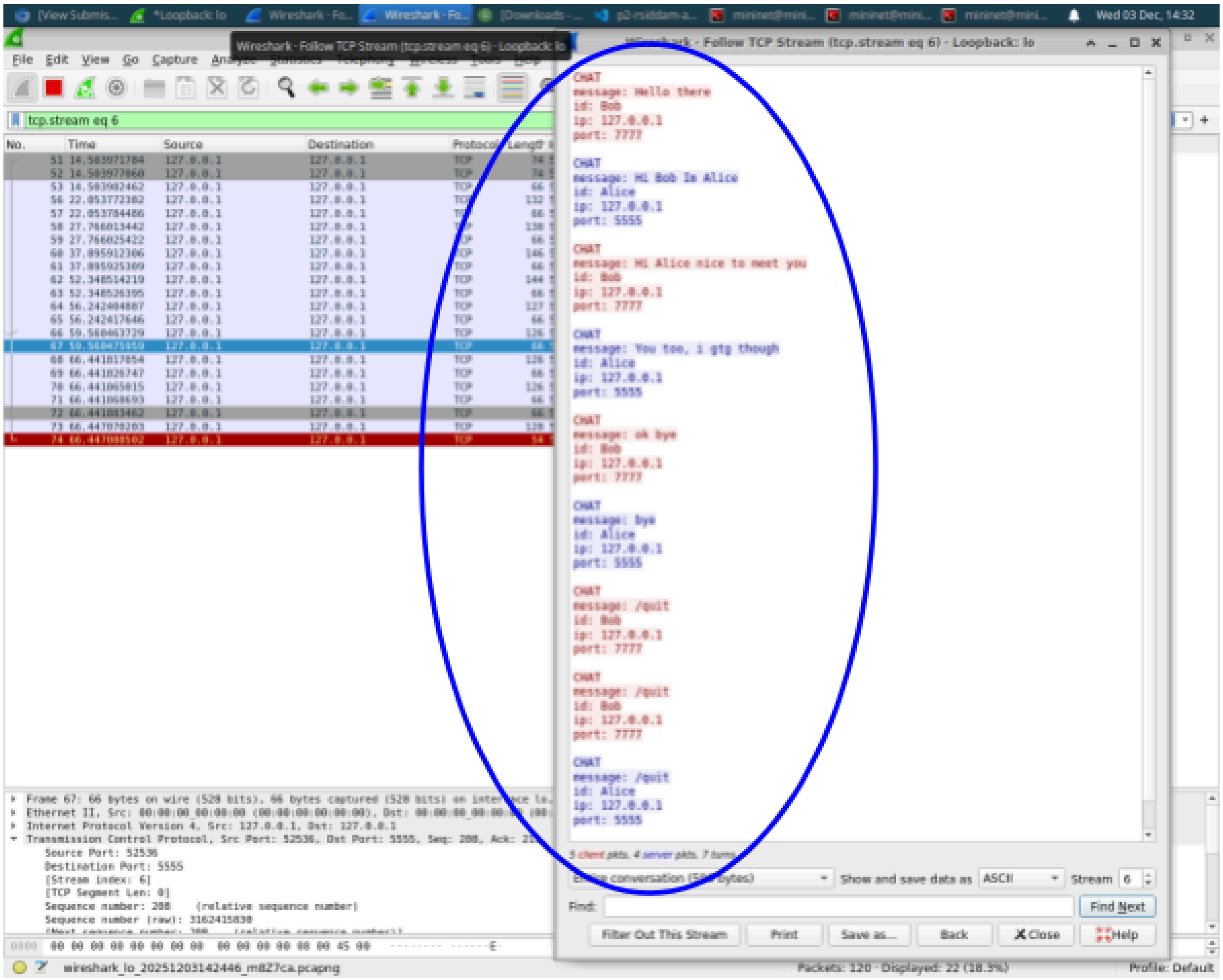


```
No.  Time            Source       Destination  Protocol Length Info
  70 66.441865815    127.0.0.1    127.0.0.1    TCP      126 52536 → 5555 [PSH, ACK] Seq=268 Ack=211 Win=65536 Len=60 TSval=346…
  71 66.441868693    127.0.0.1    127.0.0.1    TCP       66 5555 → 52536 [ACK] Seq=211 Ack=328 Win=65536 Len=0 TSval=346269025…
  72 66.441883462    127.0.0.1    127.0.0.1    TCP       66 52536 → 5555 [FIN, ACK] Seq=328 Ack=211 Win=65536 Len=0 TSval=3462…
  73 66.447070203    127.0.0.1    127.0.0.1    TCP      128 5555 → 52536 [PSH, ACK] Seq=211 Ack=329 Win=65536 Len=62 TSval=346…
  74 66.447088582    127.0.0.1    127.0.0.1    TCP       54 52536 → 5555 [RST] Seq=329 Win=0 Len=0
  75 73.519489273    127.0.0.1    127.0.0.53   DNS       82 Standard query 0x4aba A az764295.vo.msecnd.net
  76 73.520054267    127.0.0.1    127.0.0.53   DNS       82 Standard query 0x574d Unknown (65) az764295.vo.msecnd.net
  77 73.538162265    127.0.0.53   127.0.0.1    DNS       82 Standard query response 0x574d No such name Unknown (65) az764295.…
  78 73.538219920    127.0.0.53   127.0.0.1    DNS       82 Standard query response 0x4aba No such name A az764295.vo.msecnd.n…
  79 73.538350155    127.0.0.1    127.0.0.53   DNS       98 Standard query 0xcdf3 Unknown (65) az764295.vo.msecnd.net.resnet.u…
  80 73.538578106    127.0.0.1    127.0.0.53   DNS       98 Standard query 0xfef1 A az764295.vo.msecnd.net.resnet.ucsc.edu
  81 73.546546662    127.0.0.53   127.0.0.1    DNS       98 Standard query response 0xcdf3 No such name Unknown (65) az764295.…
  82 73.547697661    127.0.0.53   127.0.0.1    DNS       98 Standard query response 0xfef1 No such name A az764295.vo.msecnd.n…
  83 73.547907040    127.0.0.1    127.0.0.53   DNS       93 Standard query 0x55fd A az764295.vo.msecnd.net OPT
  84 73.556424685    127.0.0.53   127.0.0.1    DNS       93 Standard query response 0x55fd No such name A az764295.vo.msecnd.n…
  85 73.556501785    127.0.0.1    127.0.0.53   DNS      109 Standard query 0xf31a A az764295.vo.msecnd.net.resnet.ucsc.edu OPT
  86 73.565059836    127.0.0.53   127.0.0.1    DNS      109 Standard query response 0xf31a No such name A az764295.vo.msecnd.n…
  87 109.380293947   127.0.0.1    127.0.0.53   DNS       95 Standard query 0x057a A optimizationguide-pa.googleapis.com
  88 109.380353649   127.0.0.1    127.0.0.53   DNS       95 Standard query 0x4d99 Unknown (65) optimizationguide-pa.googleapis…
  89 109.386864038   127.0.0.53   127.0.0.1    DNS       95 Standard query response 0x4d99 Unknown (65) optimizationguide-pa.g…
  90 109.388208306   127.0.0.53   127.0.0.1    DNS      255 Standard query response 0x057a A optimizationguide-pa.googleapis.c…
  91 132.523514746   127.0.0.1    127.0.0.53   DNS       77 Standard query 0x785c A esp.aptrinsic.com
  92 132.524947717   127.0.0.1    127.0.0.53   DNS       77 Standard query 0x13df Unknown (65) esp.aptrinsic.com
  93 132.553946074   127.0.0.53   127.0.0.1    DNS       77 Standard query response 0x13df Unknown (65) esp.aptrinsic.com
  94 132.554022428   127.0.0.53   127.0.0.1    DNS       93 Standard query response 0x785c A esp.aptrinsic.com A 35.184.35.160
  95 159.921589042   127.0.0.1    127.0.0.53   DNS      103 Standard query 0x64f0 A mobile.events.data.microsoft.com OPT
  96 159.927280721   127.0.0.53   127.0.0.1    DNS      226 Standard query response 0x64f0 A mobile.events.data.microsoft.com …
  97 170.893027153   127.0.0.1    127.0.0.53   DNS      103 Standard query 0x2cb9 A mobile.events.data.microsoft.com OPT
  98 170.899695869   127.0.0.53   127.0.0.1    DNS      224 Standard query response 0x2cb9 A mobile.events.data.microsoft.com …
  99 188.203985242   127.0.0.1    127.0.0.53   DNS      103 Standard query 0xa017 A mobile.events.data.microsoft.com OPT
 100 188.209769528   127.0.0.53   127.0.0.1    DNS      224 Standard query response 0xa017 A mobile.events.data.microsoft.com …
 101 188.408556662   127.0.0.1    127.0.0.53   DNS       92 Standard query 0xc0c6 A mobile.events.data.microsoft.com
 102 188.408610072   127.0.0.1    127.0.0.53   DNS       92 Standard query 0xe1be Unknown (65) mobile.events.data.microsoft.com
 103 188.409051438   127.0.0.53   127.0.0.1    DNS      213 Standard query response 0xc0c6 A mobile.events.data.microsoft.com …
 104 188.413757047   127.0.0.53   127.0.0.1    DNS      197 Standard query response 0xe1be Unknown (65) mobile.events.data.mic…
 105 373.593155043   127.0.0.1    127.0.0.53   DNS       82 Standard query 0xc6c9 A az764295.vo.msecnd.net
 106 373.593213379   127.0.0.1    127.0.0.53   DNS       82 Standard query 0xf97f Unknown (65) az764295.vo.msecnd.net
 107 373.629370263   127.0.0.53   127.0.0.1    DNS       82 Standard query response 0xf97f No such name Unknown (65) az764295.…
 108 373.629502602   127.0.0.1    127.0.0.53   DNS       98 Standard query 0x1a5f Unknown (65) az764295.vo.msecnd.net.resnet.u…
 109 373.630562515   127.0.0.53   127.0.0.1    DNS       82 Standard query response 0xc6c9 No such name A az764295.vo.msecnd.n…
 110 373.630669078   127.0.0.1    127.0.0.53   DNS       98 Standard query 0x5a72 A az764295.vo.msecnd.net.resnet.ucsc.edu
 111 373.642309311   127.0.0.53   127.0.0.1    DNS       98 Standard query response 0x1a5f No such name Unknown (65) az764295.…
 112 373.642545328   127.0.0.53   127.0.0.1    DNS       98 Standard query response 0x5a72 No such name A az764295.vo.msecnd.n…
 113 373.642737558   127.0.0.1    127.0.0.53   DNS       93 Standard query 0x654e A az764295.vo.msecnd.net OPT
 114 373.651678396   127.0.0.53   127.0.0.1    DNS       93 Standard query response 0x654e No such name A az764295.vo.msecnd.n…
 115 373.651761775   127.0.0.1    127.0.0.53   DNS      109 Standard query 0xac66 A az764295.vo.msecnd.net.resnet.ucsc.edu OPT
 116 373.662689752   127.0.0.53   127.0.0.1    DNS      109 Standard query response 0xac66 No such name A az764295.vo.msecnd.n…
```

```
▸ Frame 95: 103 bytes on wire (824 bits), 103 bytes captured (824 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: 127.0.0.53
▸ User Datagram Protocol, Src Port: 44382, Dst Port: 53
▸ Domain Name System (query)
```

```
0000  00 00 00 00 00 00 00 00  00 00 00 00 00 00 45 00   ········ ······E·
```

```
wireshark_lo_20251203142446_m8Z7ca.pcapng          Packets: 116 · Displayed: 116 (100.0%)          Profile: Default
```



[View Submis... *Loopback: lo Wireshark · Fo... Wireshark · Fo... [Downloads - ... p2-rsiddam-a... mininet@mini... mininet@mini... mininet@mini... Wed 03 Dec, 14:32
Wireshark · Follow TCP Stream (tcp.stream eq 6) · Loopback: lo
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Show and save data as
Stream
Find:
Find Next
Filter Out This Stream
Print
Save as...
Back
Close
Help
Profile: Default

```
tcp.stream eq 6
```

```
No.  Time          Source      Destination  Protocol  Length
51  14.583971784  127.0.0.1   127.0.0.1    TCP        74
52  14.583977860  127.0.0.1   127.0.0.1    TCP        74
53  14.583982462  127.0.0.1   127.0.0.1    TCP        66
56  22.053772382  127.0.0.1   127.0.0.1    TCP       132
57  22.053784486  127.0.0.1   127.0.0.1    TCP        66
58  27.766013442  127.0.0.1   127.0.0.1    TCP       138
59  27.766025422  127.0.0.1   127.0.0.1    TCP        66
60  37.895912306  127.0.0.1   127.0.0.1    TCP       146
61  37.895925309  127.0.0.1   127.0.0.1    TCP        66
62  52.348514219  127.0.0.1   127.0.0.1    TCP       144
63  52.348526395  127.0.0.1   127.0.0.1    TCP        66
64  56.242404887  127.0.0.1   127.0.0.1    TCP       127
65  56.242417646  127.0.0.1   127.0.0.1    TCP        66
66  59.560463729  127.0.0.1   127.0.0.1    TCP       126
67  59.560475959  127.0.0.1   127.0.0.1    TCP        66
68  66.441817054  127.0.0.1   127.0.0.1    TCP       126
69  66.441826747  127.0.0.1   127.0.0.1    TCP        66
70  66.441065015  127.0.0.1   127.0.0.1    TCP       126
71  66.441068693  127.0.0.1   127.0.0.1    TCP        66
72  66.441083462  127.0.0.1   127.0.0.1    TCP        66
73  66.447070203  127.0.0.1   127.0.0.1    TCP       128
74  66.447088502  127.0.0.1   127.0.0.1    TCP        54
```

```
▸ Frame 67: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface lo,
▸ Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:
▸ Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
▾ Transmission Control Protocol, Src Port: 52536, Dst Port: 5555, Seq: 208, Ack: 2
    Source Port: 52536
    Destination Port: 5555
    [Stream index: 6]
    [TCP Segment Len: 0]
    Sequence number: 208    (relative sequence number)
    Sequence number (raw): 3162415838
    [Next sequence number: 208    (relative sequence number)]
0000  00 00 00 00 00 00 00 00  00 00 00 00 00 00 45 00   ........ ......E.
```

wireshark_lo_20251203142446_m8Z7ca.pcapng

Packets: 120 · Displayed: 22 (18.3%)

```
CHAT
message: Hello there
id: Bob
ip: 127.0.0.1
port: 7777

CHAT
message: Hi Bob Im Alice
id: Alice
ip: 127.0.0.1
port: 5555

CHAT
message: Hi Alice nice to meet you
id: Bob
ip: 127.0.0.1
port: 7777

CHAT
message: You too, i gtg though
id: Alice
ip: 127.0.0.1
port: 5555

CHAT
message: ok bye
id: Bob
ip: 127.0.0.1
port: 7777

CHAT
message: bye
id: Alice
ip: 127.0.0.1
port: 5555

CHAT
message: /quit
id: Bob
ip: 127.0.0.1
port: 7777

CHAT
message: /quit
id: Bob
ip: 127.0.0.1
port: 7777

CHAT
message: /quit
id: Alice
ip: 127.0.0.1
port: 5555
```

5 client pkts, 4 server pkts, 7 turns.

Entire conversation (5?? bytes) · Show and save data as: ASCII · Stream: 6

14. Consider the simplifying assumptions for this assignment. Which simplification in particular needs to be addressed for the server to function as a “real” server?

    The assumption “There are only two clients and one server” needs to be addressed for a “real” server because a real server should be able to handle a lot more clients, be able to match them up with each other, and manage commands as it does now for all clients.

15. Suppose you are in an interview, what would you have to say about this project?

    This project was an interactive exploration into network programming. I created a custom application-layer protocol using TCP as a base that combines client-server discovery with peer-peer communication. I learned how to handle commands, messages, and headers, implemented error resilience and graceful shutdown, and utilized both persistent and non-persistent connections. I also learned that, despite TCP providing reliability, it is still up to the protocol makers to handle the errors that it catches.