Testing and Analysis

### **1. TCP Connection Setup and Handshake Verification**

The TCP three-way handshake process is visible between the client and server IP addresses (e.g., 192.168.1.248 and 127.0.0.1). The handshake sequence (SYN, SYN-ACK, ACK) confirms a reliable connection setup, as each client and server successfully establish sessions on the designated port (e.g., 5000).

### **2. Message Relay and Broadcasting**

We can see TCP packets transferring data, indicating message exchange between clients and the server. Client Josh (IP: 192.168.1.248) sends messages to the server, which are relayed back to connected clients like Tony (IP: 127.0.0.1). Confirming that the server is successfully broadcasting messages to all active clients. TCP packets labeled [PSH, ACK] typically signify data transmissions, confirming that the messages are relayed as expected.

### **3. Client Connection and Disconnection**

The logs display when clients connect and disconnect. In the screenshots, it shows clients joining and leaving, along with messages in the server logs, which match the TCP reset packets (e.g., RST). The RST packets indicate disconnections. This signifies the end of a session. The server logs correctly reflect the changes in client connections, which align with the RST packets in Wireshark.

### **4. Private Messaging Verification**

Although the exact packet contents are not visible in the screenshots, it would be expected that private messages would have unique identifiers in the data payload to indicate their targeted recipient. If the chat application includes private messaging, you could look for specific data strings or payloads addressed to individual IPs without a broadcast indicator. Unfortunately, without seeing this in the capture, the functionality cannot be fully verified.

### **5. Errors or Reset Packets**

Some RST (Reset) packets are visible in the capture.

* These packets may be due to clients disconnecting or application errors. It’s essential to monitor these occurrences, as frequent resets could indicate an issue with session management or network stability.

### **6. Data Integrity and Complete Message Delivery**

TCP packets show ACK for each segment, which confirms delivery. No visible packet fragmentation or FIN-based interruptions for incomplete transmissions. Message delivery is stable and complete without issues, as TCP inherently ensures the complete and ordered delivery of data segments.

### **7. Filtering for Relevant Data**

Filters such as tcp.port == 5000 have been applied in Wireshark to narrow down relevant traffic. This filter is effective for isolating traffic on port 5000, which your application uses. Additionally, filtering by IP addresses (e.g., 192.168.1.248 for the server) can help analyze client-server exchanges and confirm targeted communication.

### **Conclusion**

The captured data demonstrates a functioning chat server where TCP handshakes, data exchanges, connection/disconnection logs, and message relay appear to operate correctly. For further analysis:

* **Check** private messaging by inspecting payloads or unique identifiers in messages.
* **Monitor** RST packets closely to understand if they correlate with intentional disconnections or potential issues.
* **Ensure** secure handling of credentials, if applicable, to prevent exposure over the network.

# Photos

