

Project 1: TCP Socket Programming and Packet Capture Report

Team Details

- **Student Name:** Brian Zhang
- **NetID:** bz271
- **Partner Name:** Christopher Doss
- **Partner NetID:** cad407
- **Assigned Server Port Number:** 30147

Collaboration

- No collaboration with other students outside of team member
- Used course materials and assignment instructions
- Consulted Python socket documentation

Analysis Questions

Part A. Protocol Basics

1. What transport-layer protocol is used in your capture? **TCP**
2. What port number did the server use? **30147**
3. What ephemeral port did the client use? (in part1, and part2) **48342, 54624**
4. What interface was used when you ran the C&S on the same machine(rlab5)? **Loopback**
5. What interface was used when you ran the C&S on different machines (client on rlab5)? **Ethernet**

Part B. Application Layer

7. In part1, Which packet carries the "Welcome to CS 352!" message? How many bytes of payload are in that packet? **Packet 4, 18 bytes**
8. In part2, Identify one packet carrying a client message. **Packet 6** Show which field contains the application data. **Data field (payload)**
9. In part2, Identify the corresponding echo reply from the server. Indicate the line #, or copy paste that line. **Packet 8: !253 sc OT EMOCLEw**

Part C. Packet Sizes

9. How many total bytes were sent from client → server, **435 bytes** and from server → client? (in both cases-part1 and part2) **385 bytes** (Total 820 bytes per capture)
10. Which packet in the capture is largest in terms of payload size? **Packets 6 and 8 (19 bytes payload)**

Conclusion

The assignment successfully demonstrated:

- TCP socket programming with client-server architecture

- Packet capture and analysis using tshark
- Understanding of transport vs application layer protocols
- Comparison between loopback and network interface traffic
- String processing with reverse and case-swap operations

Both packet captures show identical application behavior with proper TCP connection management, confirming the reliability of the socket implementation across different network scenarios.