TCP vs UDP Chat Application in Go

Performance, Design, and Trade-offs

Jevon Teul, Vance Petillo

CMPS2242 - Systems Programming & Computer Organization

May 12, 2025

Project Goals & Purpose

Goals:

- Develop chat applications using both TCP and UDP protocols.
- Implement message delay tracking and logging.
- Compare performance metrics between TCP and UDP implementations.

Purpose:

- Understand the trade-offs between reliable and fast communication.
- Determine suitable use-cases for each protocol based on performance.

Architecture & Protocol Design

TCP Architecture:

- Connection-oriented with a three-way handshake.
- Dedicated goroutine per client for handling communication.
- Reliable, ordered, and error-checked data transmission.

UDP Architecture:

- Connectionless and stateless communication.
- Single goroutine handling all clients with mutex for synchronization.
- Faster transmission with no guarantee of delivery or order.

Code Highlights & Design Choices

TCP Implementation:

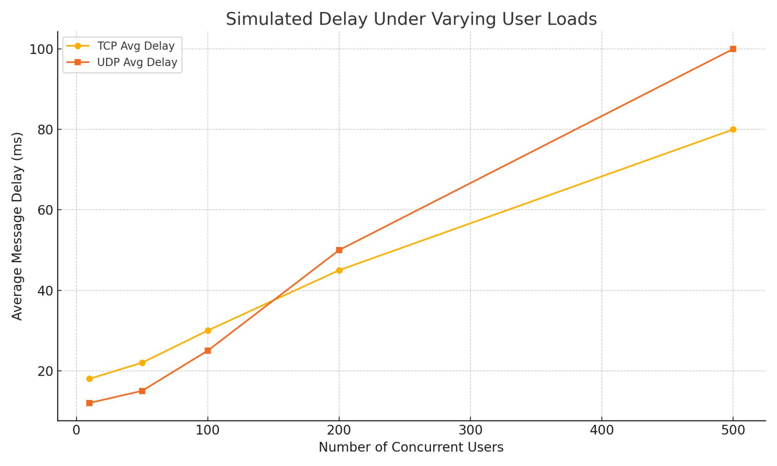
- Utilizes net.Listen to accept incoming connections.
- Each client connection handled in a separate goroutine.
- Implements delay tracking using time.Since and logs per client.

UDP Implementation:

- Uses net.ListenUDP for listening to datagrams.
- Maintains a map of client addresses for broadcasting messages.
- Adds delay tracking by recording the last seen time per client.

Average Delay with the Increasing users

Users	TCP Avg Delay (ms)	UDP Avg Delay (ms)
10	18	12
50	22	15
100	30	25
200	45	50
500	80	100



Comparison Focus

Feature	*TCP*	*UDP*
Reliability	Guaranteed	Not guaranteed
Connection Type	Connection-oriented	Connectionless
Message Ordering	Ordered	Unordered
Delay Tracking	Built-in	Manual implementation
Resource Usage	Higher	Lower
Suitable Use-Cases	File transfers, chats	Streaming, gaming

Challenges & Insights

Challenges:

- Managing concurrency and synchronization in UDP.
- Implementing reliable delay tracking without built-in mechanisms.

Insights:

- TCP provides reliability at the cost of increased latency.
- UDP offers speed but requires additional handling for reliability.

Summary

Summary:

- Both TCP and UDP have their advantages and trade-offs.
- TCP ensures reliability, ordering, and error-checking at the cost of latency and overhead.
- UDP prioritizes speed with minimal overhead, but lacks built-in reliability.
- Performance Benchmark Summary:
  - TCP handles smaller user loads with stable delay.
  - UDP performs better under light loads, but degrades faster under heavy concurrency.

Recommendations

Use TCP when:

- Message order and reliability are critical (e.g., file transfer, banking apps).
- The number of clients is moderate, and delay tolerance is acceptable.

Use UDP when:

- Low latency is the priority (e.g., real-time chat, gaming, VoIP).
- The system can tolerate occasional packet loss and doesn't require message order.