

# Reflection

1. What new knowledge you learned through this experiment? If you have used LLMs, any specific insights/knowledge/skills your acquired through LLMs.

Through this experiment, our team learned how dynamic and context-dependent transport protocols truly are. Initially, we approached **retransmission** and **acknowledgment** as static, mechanical processes. But through iterative design, testing, and debugging, we discovered that real networks are **unpredictable** — full of jitter, packet reordering, and random loss. This realization led us to explore adaptive mechanisms like **dynamic RTO** (Retransmission Timeout) and **skip control**, which allowed our protocol to “learn” from the network’s behavior rather than simply react to it.

By using Large Language Models (LLMs) such as ChatGPT, we also learned how to bridge theoretical networking knowledge with implementation-level reasoning. LLMs helped us understand why TCP’s Jacobson/Karels RTO algorithm works, and how SACK (Selective Acknowledgment) builds upon cumulative ACK to improve efficiency. Beyond coding suggestions, we used LLMs to explain the logic behind each trade-off, verify timing formulas, and even rephrase technical documentation in clearer language. Through this, we gained both technical and communication skills — learning how to reason systematically about timing, reliability, and fairness in data transmission.

2. What are the trade-offs in transport layer protocol design?

Throughout the experiment, we became more aware that every performance gain comes with a **trade-off**. Increasing reliability through retransmissions improves correctness but adds latency and congestion. Reducing waiting time through skipping or early retransmission improves responsiveness but risks data gaps or wasted bandwidth. Similarly, maintaining large buffers helps absorb jitter but increases memory use and processing overhead.

Designing SACK-based acknowledgments also showed us the tension between simplicity and precision — **cumulative ACKs** are simple and robust, but **selective ACKs** give finer control at the cost of extra complexity. Dynamic RTO tuning further reminded us that adaptability can make a system responsive, but too much sensitivity can cause instability.

Ultimately, the biggest insight we gained is that good protocol design is not about **eliminating** trade-offs, but about **balancing** them in context. A real-world transport protocol must adapt to unpredictable conditions, prioritize the user’s experience, and find equilibrium between efficiency, fairness, and reliability — values that are as much human as they are technical.