Comparative Analysis of QUIC and Other Network Protocols for Real-Time Robotic Control

Warre Clerix
Master of Electromechanical Engineering Technology

Problem Statement

Traditional network protocols present fundamental limitations for real-time robotic control. **TCP** delivers high reliability but introduces unacceptable latency for precise control [1]. **UDP** offers minimal latency but cannot guarantee delivery, creating risks in critical operations [2]. Robots operating in remote or hazardous environments require communication that simultaneously achieves both low latency and high reliability.

QUIC, **DCCP**, and **SCTP** represent newer transport protocols designed to address these limitations. **QUIC** combines **UDP's** speed with **TCP**-like reliability features and stream multiplexing [3]. **DCCP** introduces congestion control to datagram delivery without enforcing reliability [4]. **SCTP** provides multi-homing and multi-streaming capabilities for enhanced connection stability [5]. These emerging protocols offer potential solutions for the demanding requirements of robotic control systems. The protocols benefits are displayed in figure 1.

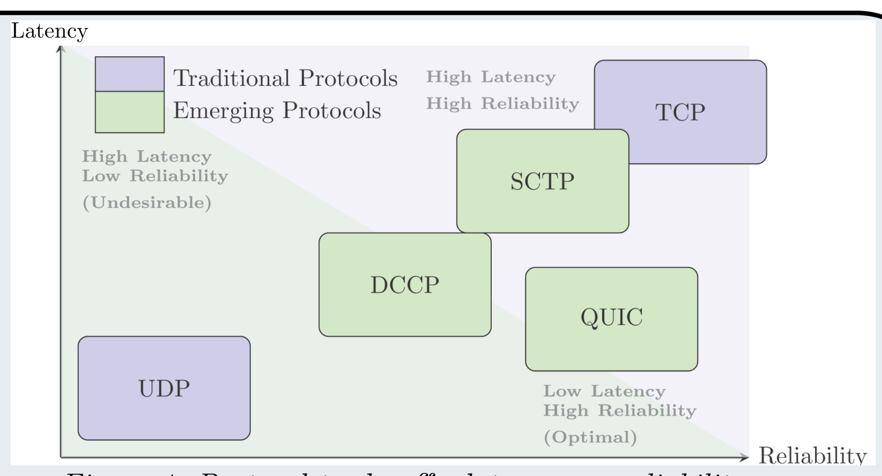

Figure 1: Protocol trade-offs: latency vs. reliability

Objectives

Primary Objective

Assess the effectiveness of **QUIC** for real-time, low-latency robot control and compare its performance against **DCCP** and **SCTP** protocols.

Secondary Objective

Evaluate protocols' network conditions while developing selection guidelines for specific robotic applications including haptic feedback, industrial automation, and emergency response systems.

Methodology

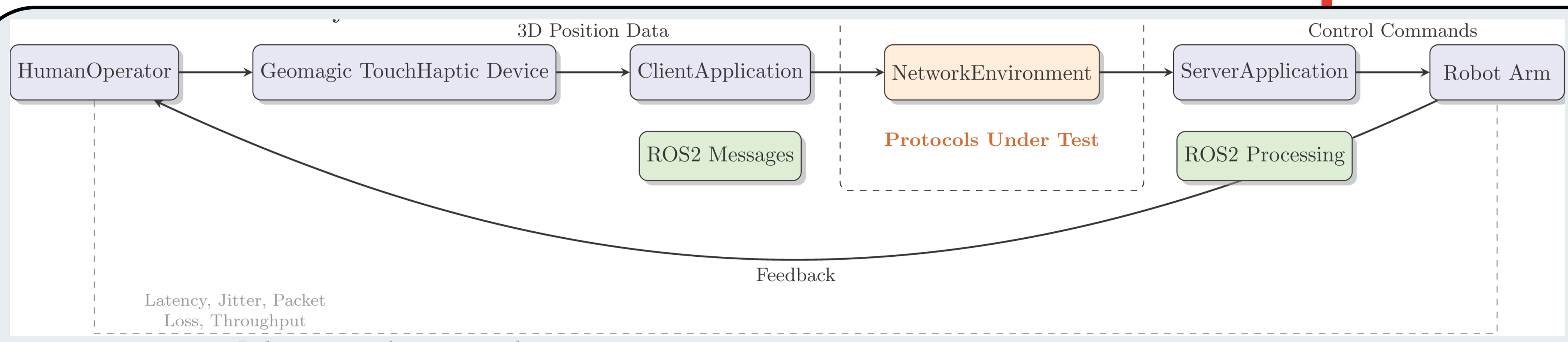

Figure 2: Robotic control system architecture

The setup uses a Geomagic Touch haptic device to capture position data, converting it to **ROS2** messages for transmission via **QUIC**, **DCCP**, or **SCTP** protocols. All protocols were implemented with equivalent architectures in **ROS2** to ensure fair comparison. The setup is shown in Figure 2.

Testing Methodology: Performance metrics (latency, jitter, packet loss, connection stability) were measured under ideal conditions. All protocols were tested with identical data patterns and control sequences.

Results

Latency Performance

Figure 3 demonstrates that **QUIC** achieves excellent latency (1.198 ms) with minimal jitter (0.036 ms) and 98.5% delivery reliability under controlled conditions. **DCCP** shows moderate latency (2.45 ms) but suffers from significant packet loss resulting in only 86.6% success rate, while **SCTP** exhibits substantially higher delays (5.231 ms) despite good reliability (97.6%). **QUIC** delivers 51% lower latency than **DCCP** and 77% lower than **SCTP**, offering substantial advantages for haptic teleoperation systems requiring sub-millisecond precision for stable control loops.

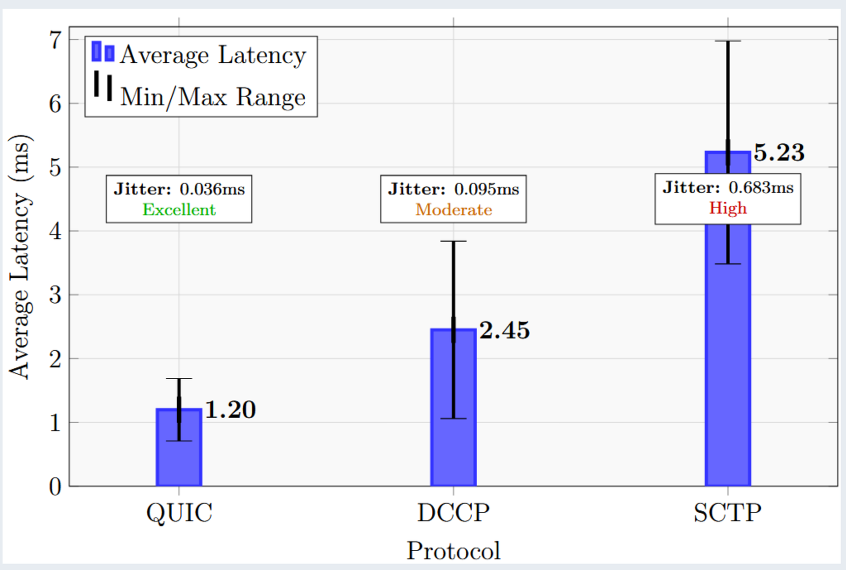

Figure 3: Average latency and jitter comparison across protocols

Multidimensional Comparison

Figure 4 reveals that **QUIC** offers the most balanced performance across all metrics among the tested protocols. It successfully combines low latency (1.198 ms) and high reliability (98.5%). **SCTP** excels in connection stability but exhibits significantly higher latency (5.231 ms). **DCCP** shows moderate latency but suffers from poor reliability (86.6%). **TCP** and **UDP** values are theoretical benchmarks for comparison only, not empirically tested in this study. **QUIC** provides the optimal balance for critical robotic control systems.

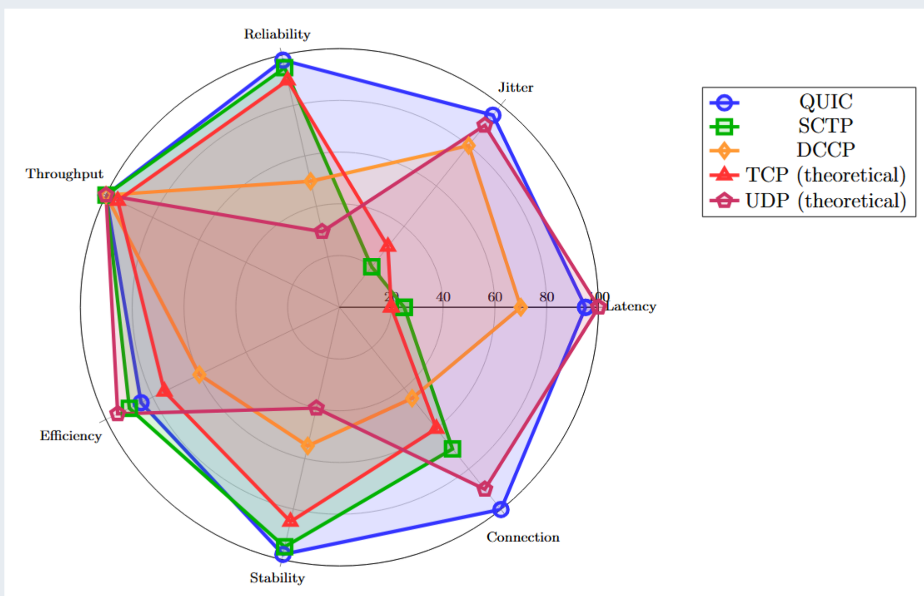

Figure 4: Protocol performance comparison across metrics

Throughput Performance

Figure 5 reveals distinct throughput behavioral patterns highlighting fundamental protocol differences under varying network conditions. **QUIC** demonstrates superior stability, maintaining consistent 1.95 Mbps performance throughout most experimental periods. The protocol exhibits intelligent congestion handling during a brief network stress event at approximately 38 seconds, where throughput temporarily drops to 1.65 Mbps before rapidly recovering within 4 seconds, demonstrating advanced adaptive algorithms that distinguish between temporary fluctuations and sustained congestion effectively. **SCTP** provides the most consistent performance, maintaining steady 1.93 Mbps throughput with minimal variation throughout the entire test duration. This rock-solid stability reflects SCTP's mature congestion control and multi-streaming architecture. **DCCP** exhibits significant instability with frequent fluctuations between 1.50-1.73 Mbps due to constant rate adjustments triggered by high packet loss rates, creating problematic feedback loops where reduced throughput temporarily improves delivery success but substantially reduces overall system efficiency for robotic control applications.

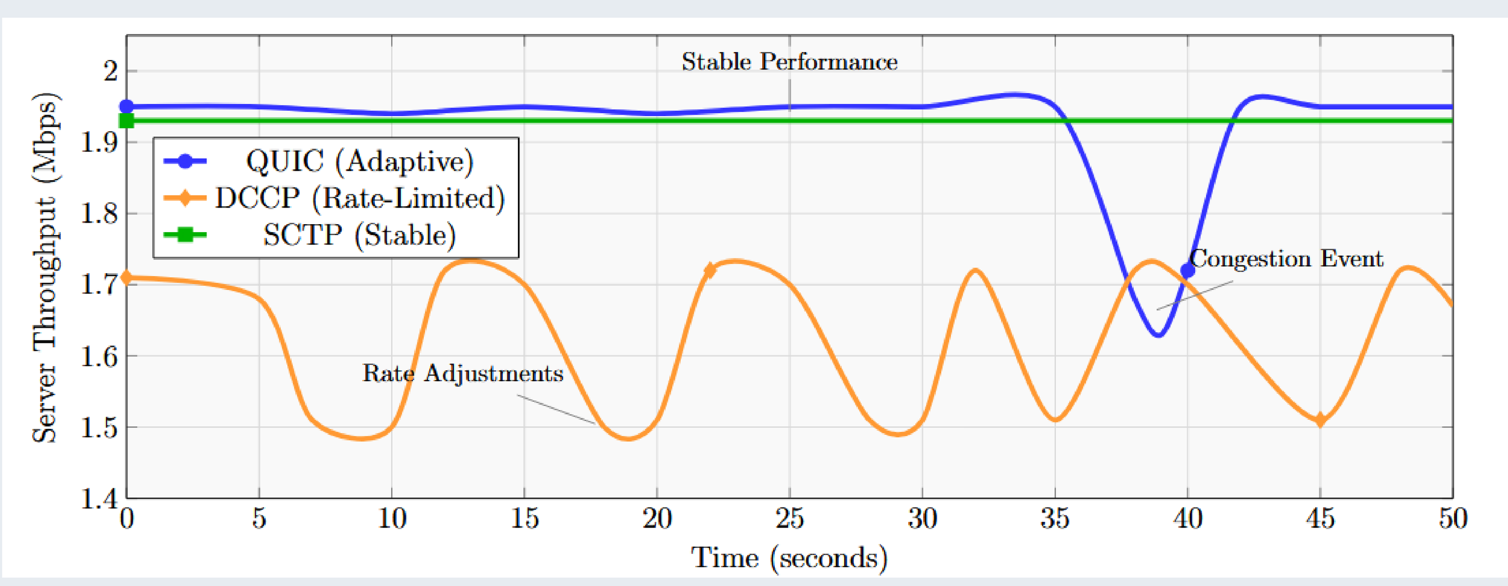

Figure 5: Server-side throughput measurements.

Conclusion

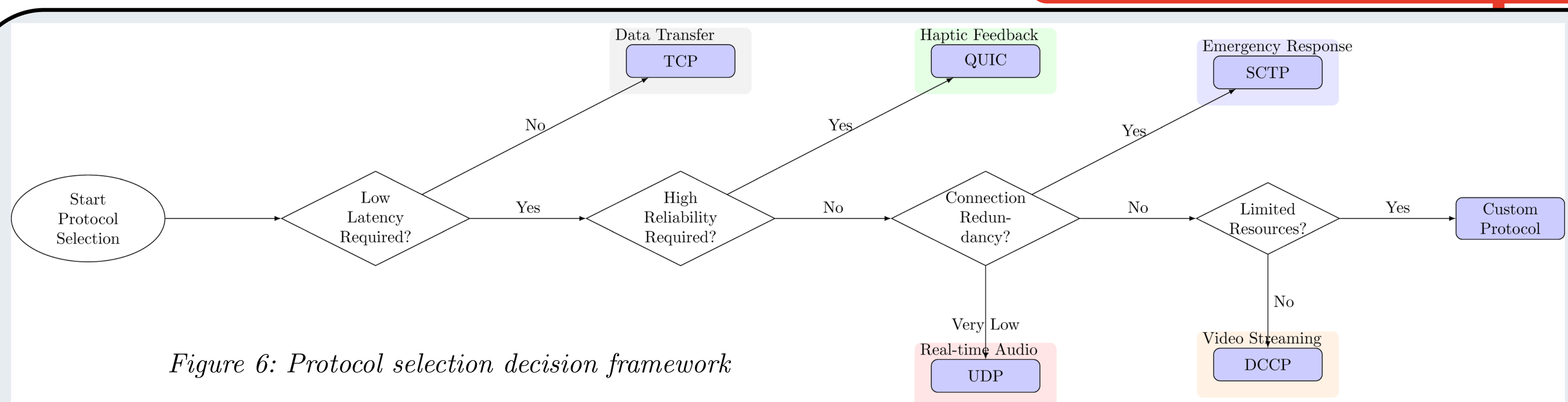

Figure 6: Protocol selection decision framework

Key Findings: QUIC's combination of **UDP**-like speed with **TCP**-like reliability makes it particularly well-suited for real-time robotic control applications. Its stream multiplexing capabilities prevent head-of-line blocking, while its connection migration features enhance stability in variable network environments. **DCCP** showed advantages in scenarios where occasional packet loss is acceptable, while **SCTP** excelled in environments requiring connection redundancy. Figure 6 shows a decision framework.

The optimal protocol depends on specific application requirements. **QUIC** is recommended for applications requiring both low latency and high reliability, particularly haptic feedback systems. **DCCP** is suitable for applications that can tolerate some packet loss but require congestion control. **SCTP** is optimal for scenarios where connection stability through redundant paths is critical.

Supervisors / Co-supervisors / Advisors: Prof. dr. Nikolaos Tsiogkas

[1] W. Eddy, "Transmission Control Protocol (TCP)." RFC 9293, Aug. 2022.
[2] J. Postel, "User Datagram Protocol." RFC 768, Aug. 1980.
[3] J. Iyengar and M. Thomson, "QUIC: A UDP-Based Multiplexed and Secure Transport." RFC 9000, May 2021.
[4] G. Fairhurst, "The Datagram Congestion Control Protocol (DCCP) Service Codes." RFC5595, Sept. 2009.
[5] E. R. Stewart, "Stream Control Transmission Protocol." RFC 4960, Sept. 2007.