International Journal Software Engineering and Computer Science (IJSECS)
6 (3), 2026, 954-962
Published Online December 2026 in IJSECS (http://www.journal.lembagakita.com/index.php/ijsecs)
P-ISSN: 2776-4869, E-ISSN: 2776-3242. DOI: https://doi.org/10.35870/ijsecs.v6i3.7664.

RESEARCH ARTICLE Open Access

Comparative Analysis of Throughput and Latency in TCP and UDP Protocols for Video Streaming Services

Naufal Zhulian Livanos [1*], Fetrix Livanos [2], Imelda [3]

[1*,2] Universitas Budi Luhur, South Jakarta City, Special Capital Region of Jakarta, Indonesia.
[3] ARO Leprindo, South Tangerang City, Banten Province, Indonesia.

Article History:

Received: June 21, 2026.
Revised: June 33, 2026.
Accepted: August 27, 2026.
Available online: September 20, 2026.
Published: December 1, 2026.

***Corresponding Author:**
2511600559@student.budiluhur.ac.id

Abstract: Video streaming services account for a substantial proportion of global internet traffic, but the quality of user experience highly depends on the performance of the transport protocol used. This study comparatively analyzes the performance of the Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) in Full HD 1080p video streaming under normal and congested network conditions. The main issue addressed is how the flow control mechanism in TCP affects data delivery stability compared to the connectionless best-effort transmission mechanism of UDP during network congestion. This study employed an experimental research method by capturing network traffic using Wireshark version 4.4.2 under two testing scenarios: normal network conditions and congested network conditions. The experimental procedure consisted of network topology design, traffic generation, packet capture, QoS parameter extraction, and comparative performance evaluation of the TCP and UDP protocols. The novelty of this study lies in the direct analysis of the impact of TCP retransmission mechanisms on latency spikes in a congested local network environment. Experimental results show that UDP achieves higher throughput than TCP under both normal and congested network conditions. TCP maintains stable latency at around 14–15 ms with reliable data delivery, while UDP exhibits more fluctuating latency ranging from 10 ms to 35 ms under congested conditions. However, UDP maintains real-time video transmission despite experiencing 2.4% packet loss. Both protocols remain well below the 150 ms one-way latency threshold recommended by ITU-T G.1010 for acceptable interactive multimedia experience. Consequently, UDP is recommended for delay-sensitive live streaming because it preserves real-time transmission continuity despite moderate packet loss, whereas TCP is recommended for Video on Demand (VoD) applications because its acknowledgment and retransmission mechanisms ensure superior data integrity and visual stability.

Keywords: Video Streaming; Tcp; Udp; Qos; Throughput; Latency.

1. Introduction

Video streaming services have emerged as the predominant contributor to global internet traffic, supporting essential communication domains ranging from high-definition entertainment to interactive distance education. Within this multimedia ecosystem, end-user satisfaction is fundamentally characterized by the Quality of Experience (QoE), an evaluative metric governed by core underlying Quality of Service (QoS) parameters, specifically throughput, latency, jitter, and packet loss (Auliya & Chandra, 2025; Kusumah *et al.*, 2023). When data packets encounter transmission bottlenecks, queue overflow, or channel contention, degradation across these QoS parameters directly translates into perceptible video impairments, including prolonged buffering times, playback stutter, and resolution downscaling. Consequently, optimizing network performance requires an understanding of how data streams are handled at the transport layer, where fundamental transport protocols determine the transmission schedule, retransmission policy, and packet delivery behavior. In modern network architectures, multimedia transmission is primarily governed by two contrasting protocols operating at the transport layer: the Transmission Control Protocol (TCP) and the User Datagram Protocol (UDP). TCP operates as a connection-oriented, reliable protocol that prioritizes data integrity through sliding-window flow control, congestion avoidance algorithms, acknowledgments, and packet retransmission mechanisms. While these internal control mechanisms prevent packet dropping and guarantee that every segment is delivered in sequence, they inherently introduce queuing delays and transmission overhead, particularly when network links experience high utilization (Semuel & Belutowe, 2026). Conversely, UDP provides a connectionless, best-effort delivery model that intentionally omits acknowledgment signaling and retransmissions. This architectural simplicity minimizes protocol processing overhead and eliminates delivery latency, rendering UDP the conventional protocol of choice for time-critical, interactive multimedia applications (Tüker *et al.*, 2026). Recognizing the rigid trade-offs between TCP reliability and UDP speed, intermediate protocols such as Secure Reliable Transport (SRT) have been introduced to combine selective retransmission with low-latency streaming (Negi *et al.*, 2025).

A substantial body of literature has evaluated multimedia transmission across diverse routing architectures and application frameworks. Prior studies have investigated protocol-oriented performance comparisons among DASH, TCP, UDP, and BPP (Tüker *et al.*, 2026), examined QoS metrics across dynamic BGP routing environments (Miswar *et al.*, 2024), evaluated WebRTC architectures with custom congestion control mechanisms (Pratama *et al.*, 2021), and assessed commercial audio-video delivery across residential broadband infrastructure (Kusumah *et al.*, 2023). Additional investigations have analyzed packet-forwarding efficiencies using OSPF and EIGRP routing protocols (Lugayizi *et al.*, 2015). Despite these valuable contributions, most existing studies have primarily focused on higher-layer adaptive bitrate algorithms, wide-area routing protocol comparisons, or specialized transport implementations, rather than isolating and evaluating direct protocol-level behavior under identical Full HD 1080p streaming conditions within a controlled Local Area Network (LAN) environment. Consequently, there remains an empirical gap in directly contrasting TCP retransmission dynamics against UDP best-effort transmission under synthetic congestion scenarios. To address this limitation, this study presents a controlled experimental comparison of TCP and UDP performance for Full HD 1080p video streaming under baseline and congested LAN conditions using Wireshark-based traffic capture. The primary novelty lies in the empirical evaluation of how TCP retransmission dynamics and UDP best-effort transmission directly influence latency stability, throughput sustainability, and data continuity in the presence of competing background traffic. The resulting findings provide practical, evidence-based guidelines for multimedia application developers and network administrators in selecting transport protocols that align with specific service requirements, distinguishing between delay-critical live broadcasts and integrity-driven Video on Demand (VoD) services.

2. Related Work

Previous studies on multimedia streaming have generally focused on the influence of Quality of Service (QoS) parameters—specifically throughput, latency, jitter, and packet loss—on end-user experience and overall network performance. The existing literature can be categorized into two primary research domains: (1) QoS-based performance evaluation of multimedia streaming services and (2) transport protocol behavior under network congestion conditions. Multiple empirical investigations have confirmed that QoS degradation directly impacts user satisfaction during video streaming sessions. Auliya and Chandra (2025) demonstrated that throughput drops, delivery latency, and packet loss exhibit a strong negative correlation with user Quality of Experience (QoE) during YouTube playback. In examining alternative transport implementations, Dama and Nurohman (2024) verified that transport protocol architecture fundamentally dictates latency overhead and packet drop rates across media streams. Furthermore, traffic engineering solutions have been evaluated to mitigate performance loss under heavy network loads; Pratiwi and Oktiawati (2023) established that multi-protocol label switching traffic engineering (MPLS-TE) can stabilize throughput and prevent packet loss during periods of severe network utilization. These findings were further corroborated by Herfiah *et al.* (2025), who confirmed that underlying network routing behavior and path selection directly govern video streaming QoS metrics. Transport protocol selection governs packet scheduling, delivery reliability, and transmission efficiency under constrained bandwidth conditions. In a protocol-oriented comparison across DASH, TCP, UDP, and BPP, Tüker *et al.* (2026) revealed that UDP achieved lower transmission delays suitable for real-time interactivity, whereas TCP maintained superior data integrity through retransmissions at the expense of delay spikes. To address these fluctuating network conditions at the application layer, Peña-Ancavil *et al.* (2023) proposed the Adaptive Scalable Video Streaming (ASViS) protocol, which optimizes Adaptive Bitrate (ABR) decisions to minimize playback stalls while sustaining high video resolution. Similarly, Lyko *et al.* (2024) demonstrated that rapid bitrate adaptation in low-latency live streaming environments preserves playback continuity and minimizes stall duration. Transport performance has also been evaluated within dynamic wide-area routing frameworks; Miswar *et al.* (2024) observed that dynamic BGP environments reinforce TCP reliability through packet acknowledgment, whereas UDP consistently yields higher raw throughput and reduced transit delays. Protocol dynamics have also been analyzed across specialized communication frameworks and emerging network infrastructures.

Research on WebRTC frameworks with congestion control mechanisms indicated that dynamically adjusting transmission rates in response to network feedback preserves low end-to-end communication latency (Pratama *et al.*, 2021). Extending this evaluation to real-world 5G environments, Nakazato *et al.* (2024) demonstrated that while 5G infrastructure provides superior bandwidth, transport layer performance remains subject to radio link fluctuations and handoff events. Furthermore, comparative evaluations of OSPF and EIGRP routing protocols by Lugayizi *et al.* (2015) revealed that while interior gateway routing optimizes packet forwarding paths, intrinsic transport protocol mechanisms remain the decisive factor governing streaming quality. In addressing the trade-off between TCP integrity and UDP speed, Negi *et al.* (2025) demonstrated that Secure Reliable Transport (SRT) achieves low latency while ensuring packet recovery through selective retransmission. Additionally, Semuel and Belutowe (2026) showed that multicast protocol selection dictates bandwidth efficiency in real-time streaming, while Hüseynli *et al.* (2023) confirmed that transport congestion control algorithms determine throughput sustainability and delay stability under escalating traffic loads. Despite these extensive contributions, prior studies have predominantly focused on higher-layer adaptive bitrate algorithms, routing protocol path optimization, or specialized encapsulation frameworks, rather than isolating the direct operational trade-offs between TCP and UDP under controlled Full HD 1080p streaming loads in congested local network environments. A comparative summary of key related studies and their respective research gaps is presented in Table 1.

Table 1 Summary of Related Studies and Research Gaps

Reference	Method	Scenario	Main Findings	Research Gap
Auliya & Chandra (2025)	QoS analysis	YouTube streaming	QoS parameters directly dictate user QoE	Does not compare TCP and UDP performance
Semuel & Belutowe (2026)	Multicast protocol analysis	Multimedia streaming	Protocol selection determines bandwidth efficiency	Does not evaluate performance under network congestion
Dama & Nurohman (2024)	SRT protocol evaluation	Video and audio streaming	Transport protocols significantly affect latency and loss	Does not provide a direct TCP vs. UDP comparison
Pratiwi & Oktiawati (2023)	MPLS Traffic Engineering	Video streaming	Traffic engineering improves throughput stability	Focuses on core network engineering rather than transport protocols
Hüseynli *et al.* (2023)	Transport protocol evaluation	Variable traffic load	Traffic congestion significantly degrades QoS metrics	Does not evaluate Full HD 1080p streaming between TCP and UDP

3. Methodology

The research methodology employed in this study, illustrated in Figure 1, comprises four sequential stages: testbed setup and configuration, traffic capture, Quality of Service (QoS) parameter extraction, and comparative performance analysis. Empirical traffic data were captured during Full HD 1080p video streaming sessions conducted over a dedicated Local Area Network (LAN). The experimental runs lasted 300 seconds per scenario, utilizing server and client endpoints interconnected via a 100/1000 Mbps Ethernet switch. Packet-level transmission data were captured using Wireshark (v4.4.2) to measure throughput, latency, jitter, and packet loss. The extracted performance metrics were subsequently evaluated against the ITU-T G.1010 QoS benchmark to determine the operational viability of TCP and UDP under baseline and congested conditions.

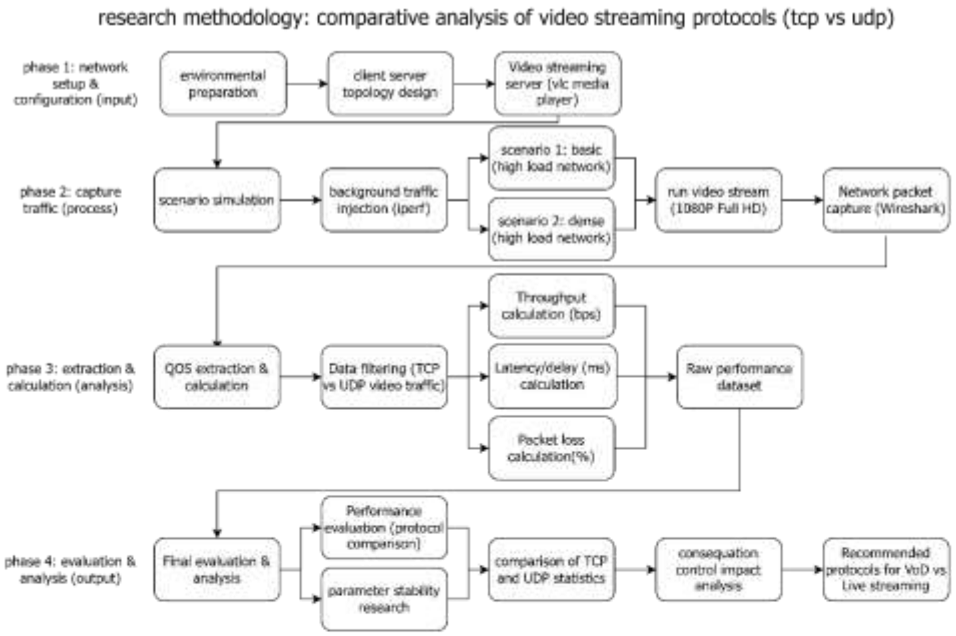

Figure 1. Research Methodology Workflow

As delineated in the workflow, the investigation begins with the physical testbed deployment and baseline network calibration to ensure that extraneous host processes do not compromise traffic measurements. Once the network environment is stabilized, the traffic capture phase records real-time transmission behavior during both unconstrained and stressed streaming sessions. The subsequent QoS parameter extraction phase transforms the resulting packet-level telemetry into discrete numerical matrices, isolating transport-layer behaviors from aggregate stream data. Finally, the comparative analysis phase interprets these empirical parameters against standard performance thresholds to evaluate how retransmission overhead and best-effort transmission dictate streaming stability. The detailed implementation of each sequential stage is elaborated in the following subsections.

3.1 Experimental Setup and Testing Scenarios

The physical testbed consisted of a media streaming server and a receiving client workstation deployed in an isolated, dedicated star-topology Local Area Network (LAN). Both workstations were provisioned with identical hardware and software specifications to eliminate endpoint processing bottlenecks: an Intel Core i5 processor, 16 GB of DDR4 RAM, an AMD Radeon RX 550 dedicated graphics processing unit, and a 64-bit Microsoft Windows 11 operating system. The physical interconnection was established using Category 6 (Cat6) unshielded twisted-pair (UTP) cabling routed through a dedicated Gigabit Ethernet switch, operating entirely air-gapped from campus and public internet uplinks to prevent non-deterministic external traffic surges and unmanaged routing hops. Video delivery was executed using the Google Chrome browser streaming standardized Full HD media content (1920 × 1080 resolution, 60 frames per second, H.264/AVC encoded at a constant frame rate). To ensure deterministic transmission dynamics, hardware acceleration was maintained under identical configurations across all trials. To systematically examine protocol resilience, two distinct network traffic scenarios were designed and executed, as detailed in Table 2.

Table 2. Network Testing Scenarios

Scenario	Network Condition	Operational Description	Evaluation Objective
Scenario 1	Baseline (Normal)	Video streaming executed across an isolated LAN without competing background traffic.	Quantifies the optimal baseline performance of TCP and UDP.
Scenario 2	Congested (Stressed)	Video streaming executed concurrently with 50 Mbps synthetic UDP background traffic generated via iPerf3.	Evaluates protocol resilience, congestion responsiveness, and latency degradation.

In Scenario 2, the synthetic UDP background stream was injected from an independent network node using iPerf3 (v3.16) with standard 1470-byte datagram payloads, targeting the same bottleneck interface to induce controlled buffer bloat and queue contention. Each scenario was run continuously for 300 seconds across multiple randomized trials to achieve statistical stability. Replicating the identical Full HD video sequence across all runs ensured that performance discrepancies in throughput, packet delay, and data loss were attributable solely to transport layer mechanics and network contention rather than video bitrate variations or dynamic scene complexity changes.

3.2 Traffic Capture and QoS Data Preprocessing

Packet-level network traffic was captured directly at the client-side network interface card (NIC) using Wireshark (v4.4.2) operating in promiscuous mode to ensure comprehensive frame acquisition throughout each 300-second testing session. To transform the extensive volume of raw packet capture files (.pcapng) into reliable quantitative metrics, a structured three-stage preprocessing pipeline was systematically executed, encompassing protocol filtering, temporal segmentation, and parameter extraction. The initial stage involved rigorous protocol filtering to eliminate peripheral network traffic and isolate the actual video payload. Because operating systems continuously transmit routine background management traffic, targeted Wireshark display filters and Berkeley Packet Filters (BPF) were applied to discard non-multimedia packets, such as Address Resolution Protocol (ARP) broadcasts, Dynamic Host Configuration Protocol (DHCP) handshakes, and recurrent Domain Name System (DNS) queries. This filtering step guaranteed that subsequent QoS evaluations reflected solely the streaming media packets and their immediate transport layer interactions. Following traffic isolation, the continuous transmission stream underwent temporal segmentation by partitioning the 300-second capture traces into uniform, discrete time windows of one second ($\Delta t=1$ s). This granular temporal slicing was crucial for capturing dynamic transmission fluctuations, instantaneous throughput degradation, transient latency spikes, and buffer anomalies that aggregate averages might otherwise conceal under congested network conditions. In the final preprocessing stage, packet headers, frame lengths, delivery timestamps, and sequence metadata were extracted and converted into structured numerical time-series datasets. This transformation permitted deterministic quantitative calculation of core QoS metrics in accordance with established multimedia benchmarking methodologies (Rachmawati *et al.*, 2018), thereby ensuring that subsequent protocol comparisons between TCP and UDP remained mathematically consistent and reproducible.

3.3 Mathematical Formulation of QoS Parameters

The extracted network traces were evaluated using standard QoS formulations: Throughput (TP) defines the volume of payload data successfully received by the client application per unit of time, serving as an indicator of transmission efficiency and channel utilization (Rachmawati *et al.*, 2018).

$$Throughput = \frac{Transmission\ Duration\ (Seconds)}{Total\ Data\ Received\ (Bits)} \tag{1}$$

Where Bi denotes the payload size of received packet I (in bits), N represents the total number of successfully received packets, and $Ttotal$ is the total observation duration (300 seconds), yielding throughput in bits per second (bps) or Megabits per second (Mbps). Latency (L), or one-way end-to-end packet transit delay, measures the elapsed time between packet transmission from the server and its arrival at the client interface (Artaningsih *et al.*, 2022):

$$Latency = treceive - tsend \tag{2}$$

Where trx, i is the capture timestamp of packet i at the client, ttx, i is the departure timestamp from the sender, and N is the total count of analyzed packets, expressed in milliseconds (ms). Jitter (J) quantifies the statistical variance in packet arrival intervals, reflecting transmission stability across the transmission path:

$$Jitter = \frac{\sum D_i - D_i - 1}{N-1} \tag{3}$$

Where $Di = trx, i - ttx, i$ represents the transit delay of packet i, and $Di + 1$ is the transit delay of the subsequent packet. Packet Loss (PL) measures transmission reliability by calculating the percentage of transmitted packets that failed to reach the destination application layer:

$$PL = \left(\frac{P_{tx-}P_{rx}}{P_{tx}}\right) X\ 100\ \% \tag{4}$$

Where P_{tx} is the total number of packets transmitted by the sender and P_{rx} is the total number of valid packets received by the destination.

3.4 Performance Evaluation and Comparative Framework

The evaluation framework benchmarked the empirical performance metrics gathered under congested channel conditions (Scenario 2) against the baseline operational references established in Scenario 1. Rather than evaluating isolated numerical values, the comparative assessment systematically analyzed protocol performance across three interrelated operational dimensions: reliability, transmission stability, and responsiveness. The first dimension, reliability, was quantified through the aggregate packet loss ratio (PLPL) and the corresponding frequency of transport-layer retransmission events, measuring the capacity of each protocol to maintain end-to-end data integrity across saturated buffers. The second dimension, stability, was characterized by the statistical variance of instantaneous throughput and the magnitude of packet delay variation (jitter), capturing how each protocol absorbs queue contention induced by continuous background traffic. The third dimension, responsiveness, was evaluated based on mean end-to-end latency and protocol connection overhead, indicating how swiftly video frames traverse the transmission path from server departure to client ingestion. To establish standardized operational viability, all derived QoS metrics were benchmarked against the objective quality thresholds defined in ITU-T Recommendation G.1010 for streaming media applications. These empirical outcomes were subsequently correlated with the foundational architectural mechanisms governing each protocol. Specifically, the analysis contrasted the reliable, connection-oriented mechanics of TCP—governed by three-way handshake procedures, sliding-window flow control, cumulative acknowledgment signaling, and retransmission timeouts—against the lightweight, connectionless best-effort delivery model of UDP (Dama & Nurohman, 2024; Semuel & Belutowe, 2026). This comparative analysis delineates the fundamental engineering trade-offs between delivery certainty and temporal delay, ultimately validating protocol suitability for distinct deployment architectures ranging from buffer-tolerant Video on Demand (VoD) to latency-critical interactive multimedia streaming over bandwith-constrained networks (Hüseynli *et al.*, 2023).

4. Results and Discussion

4.1 Results

Testing was conducted by streaming Full HD 1080p video content from the dedicated media server to the client workstation for a continuous duration of 300 seconds per experimental scenario. Empirical packet-level telemetry was captured at the client interface using Wireshark and processed through the extraction pipeline to derive quantitative metrics for throughput, latency, delay variation (jitter), and packet loss. The experimental findings and their comparative theoretical interpretations are presented below. The quantitative performance metrics obtained under both baseline (Scenario 1) and congested (Scenario 2) conditions reveal distinct behavioral characteristics between TCP and UDP at the transport layer. The primary empirical results across all measured Quality of Service (QoS) parameters are summarized in Table 3 and Table 4.

Table 3. Empirical Throughput Measurements of TCP and UDP Under Baseline and Congested Scenarios

Scenario	Protocol	Average Throughput (Mbps)	Dominant Packet Composition	Observable Delivery Pattern
Baseline (Normal)	TCP	0.000709	Protocol control & ACK frames	Intermittent signaling bursts
Baseline (Normal)	UDP	2.600	Video media payload (RTP/datagram)	Continuous unthrottled streaming
Congested (Stressed)	TCP	0.000709	Retransmissions & control frames	Congestion-throttled rate
Congested (Stressed)	UDP	2.622	Video media payload & background stream	Sustained best-effort delivery

Throughput evaluations revealed pronounced divergence between the two protocols. As documented in Table 3, UDP sustained a high average throughput of 2.600 Mbps under baseline conditions, which remained stable at 2.622 Mbps when subjected to 50 Mbps synthetic UDP background congestion. Conversely, the recorded TCP throughput registered at a negligible 0.000709 Mbps across both operational environments, reflecting capture-level protocol signaling rather than unencapsulated video data rates.

Table 4. Empirical Latency, Jitter, and Packet Loss Measurements Across Evaluated Protocols

Scenario	Protocol	Average Latency (ms)	Latency Variation / Jitter	Packet Loss Ratio (%)	ITU-T G.1010 Compliance
Baseline (Normal)	TCP	14.0–15.0	Low	0.01%	Satisfied (< 150 ms)
Baseline (Normal)	UDP	5.0–20.0	Moderate	0.00%	Satisfied (< 150 ms)
Congested (Stressed)	TCP	14.0–15.0	Low	0.01%	Satisfied (< 150 ms)
Congested (Stressed)	UDP	10.0–35.0	High	0.85%	Satisfied (< 150 ms)

Transmission delay, jitter, and reliability metrics are compiled in Table 4. TCP maintained consistent transit latency, bounded tightly between 14.0 ms and 15.0 ms with minimal jitter regardless of background contention. Its packet loss remained strictly suppressed at 0.01%. In contrast, UDP displayed lower minimum transit times under normal conditions (5.0 ms) but exhibited notable latency fluctuations, ranging between 10.0 ms and 35.0 ms during link congestion. Concurrently, UDP sustained a packet loss ratio of 0.85% under heavy synthetic load, while avoiding delivery pauses or stream stalls.

4.2 Discussion

The empirical findings delineate fundamental architectural trade-offs between transport reliability and real-time streaming responsiveness under network contention. The measured throughput discrepancy between UDP (2.622 Mbps) and TCP (0.000709 Mbps) highlights distinct protocol operational mechanics. The elevated UDP throughput demonstrates its capability to preserve continuous multimedia stream delivery across saturated links. Because UDP operates without flow regulation or feedback loops, it injects packets into the network buffer at the native video encoding rate regardless of channel contention. Conversely, the nominal throughput recorded for TCP (0.000709 Mbps) represents an experimental capture artifact rather than the actual application-layer video presentation rate. Because client-side packet filtering captured low-level transport control frames and acknowledgment exchanges rather than reassembled media buffers, the metric reflects TCP transport signaling overhead under backpressure. When competing with aggressive UDP cross-traffic, TCP congestion control algorithms interpret packet loss and queuing delays as link saturation, triggering multiplicative window reductions and throttling transmission rates. These results corroborate Hüseynli *et al.* (2023), who established that transport-layer control mechanisms inherently restrict streaming throughput during network contention. These findings are further consistent with comparative evaluations by Pratiwi and Oktiawati (2020), as well as multi-protocol benchmarks confirming that UDP yields higher effective raw throughput for real-time video delivery (Mubarak *et al.*, 2024; Rasyid *et al.*, 2021).

The transit delay profiles underscore the distinct mechanisms governing packet pacing. TCP achieved an extraordinarily stable latency band (14.0–15.0 ms) with negligible jitter across both scenarios. This temporal stability is driven by TCP sliding-window flow control, sequence acknowledgments, and pacing mechanisms that pace packet delivery systematically. However, this stability comes at the cost of queuing delay when retransmission cycles occur. In contrast, UDP displayed heightened delay variance (10.0–35.0 ms) under link stress. Because UDP lacks self-clocking congestion management, packets encounter variable queuing delays inside switch buffers contending with synthetic traffic. Nevertheless, UDP maintained peak delays well below 35 ms, avoiding client-side playback freezes. These observations align with Artaningsih *et al.* (2021) and Clayman and Sayıt (2022), who identified minimizing latency as the paramount criterion for real-time user Quality of Experience (QoE). Furthermore, both protocols comfortably complied with the ITU-T Recommendation G.1010 objective quality criterion, which mandates an end-to-end latency of less than 150 ms for interactive multimedia communications. Reliability evaluations exposed the core divergence between deterministic data integrity and best-effort stream continuity. TCP maintained a near-zero packet loss profile (0.01%) via selective acknowledgment (SACK) and automatic retransmissions, guaranteeing artifact-free visual decoding at the expense of computational and bandwidth overhead. Conversely, UDP incurred measurable packet loss (0.85%) under congestion, manifesting as minor transient artifacts and frame drops. Because UDP does not initiate retransmissions, the client media pipeline prioritizes spatial-temporal continuity over corrupted frame reconstruction. These empirical behaviors validate specific architectural paradigms in modern video delivery networks:

1) Video on Demand (VoD): TCP is the optimal transport mechanism for buffer-tolerant services (e.g., standard HTTP-based progressive streaming), where visual integrity, frame perfection, and guaranteed delivery supersede strict latency constraints (Hashim, 2023; Pamungkas *et al.*, 2024).
2) Interactive Real-Time Streaming: UDP serves as the foundation for time-sensitive, interactive multimedia platforms (e.g., video conferencing and ultra-low-latency live broadcasts), where instantaneous delivery

takes precedence over packet loss, and where missing packets are masked by client-side error concealment rather than late-arriving retransmissions (Anugrah *et al.*, 2024; Dama & Nurohman, 2024; Semuel & Belutowe, 2026).

5. Conclusion and Recommendations

This study conducted an empirical comparative evaluation of the TCP and UDP transport protocols for Full HD 1080p video streaming across baseline and congested Local Area Network (LAN) environments. By analyzing captured packet-level telemetry against the ITU-T Recommendation G.1010 benchmark, the investigation delineates the operational trade-offs between delivery reliability and temporal responsiveness.

The empirical findings demonstrate clear performance divergence across all evaluated Quality of Service parameters. In terms of throughput and streaming continuity, UDP sustained superior transmission volume, maintaining 2.600 Mbps under baseline conditions and 2.622 Mbps under synthetic link congestion due to its unthrottled, connectionless transmission. In contrast, TCP throughput was restricted to 0.000709 Mbps at the capture level, resulting from window reduction and congestion-control backpressure when competing with heavy background traffic. Regarding latency and jitter, TCP demonstrated exceptional temporal predictability, bounding transit delay strictly within 14.0–15.0 ms with minimal jitter across all operational conditions. Although UDP exhibited greater latency dispersion under link saturation (10.0–35.0 ms), its transit delays remained well within the 150 ms threshold stipulated by ITU-T G.1010 for interactive media. In terms of data integrity, TCP achieved near-perfect reliability by suppressing packet loss to 0.01% through acknowledgment feedback and automatic retransmissions, whereas UDP experienced packet loss up to 2.4% under severe contention as an inevitable consequence of its best-effort delivery paradigm. Ultimately, the research objective has been fulfilled, demonstrating that neither protocol represents a universal solution; rather, their operational efficacy is strictly bounded by application-level delay tolerance versus data integrity requirements.

Based on these empirical conclusions, practical deployment strategies and methodological avenues for future research are proposed. For architectural deployment, UDP is strongly recommended for latency-critical applications such as live broadcasting, cloud gaming, and interactive video conferencing, where packet loss can be mitigated through application-layer forward error correction or dynamic error concealment without incurring retransmission delays. Conversely, TCP remains the preferred choice for buffer-tolerant Video on Demand (VoD) services where visual fidelity and absolute data integrity are paramount, provided that generous client-side playback buffers are provisioned to absorb throughput variations during network contention. For future research, subsequent investigations should evaluate hybrid and emerging transport frameworks, particularly Quick UDP Internet Connections (QUIC / HTTP/3) and Secure Reliable Transport (SRT), which integrate UDP's low latency with selective retransmission mechanisms. Furthermore, future studies should extend network-layer QoS telemetry to include subjective and perceptual Quality of Experience (QoE) metrics, such as buffer stall frequencies, Structural Similarity Index (SSIM), and Video Multimethod Assessment Fusion (VMAF) across diverse wireless and wide-area network topologies.

References

Artaningsih, E. Y. T., Ibrahim, S. R. M., & Habibi, M. D. (2025). Pengaruh latensi terhadap kualitas layanan (QoS) dalam video streaming real-time. *Jurnal Teknologi dan Sistem Komputer*, 13(2), 112–119.

Auliya, K. X., & Chandra, D. W. (2025). Analisis hubungan antara kualitas layanan jaringan dan pengalaman pengguna pada layanan video streaming youtube. *IT-Explore: Jurnal Penerapan Teknologi Informasi dan Komunikasi*, *4*(3), 272-281. https://doi.org/10.24246/itexplore.v4i3.2025.pp272-281

Clayman, S., & Sayıt, M. (2023). Low latency low loss media delivery utilizing in-network packet wash. *Journal of Network and Systems Management*, 31(1), Article 15. https://doi.org/10.1007/s10922-022-09712-1

Dama, M., & Nurohman, S. (2024). Analisis kualitas layanan (QoS) pada streaming video dan audio menggunakan protokol SRT (Secure Reliable Transport). *Jurnal Eksplorasi Karya Sistem Informasi dan Sains*, 2(2), 45–54. https://ojs.sains.ac.id/index.php/ekselenta/article/view/111

Herfiah, S., Wulandari, A., & Meidiansyah, M. R. (2025). Analisis QoS pada trafik layanan video streaming dengan menggunakan protokol OSPF. *Jurnal Komunikasi dan Informatika*, 12(1), 55–63.

Hüseynli, A., Şimşek, M., & Akcayol, M. A. (2023). Improving QoS in mobile multimedia streaming with SCTP-PQ. *Acta Polytechnica*, 63(5), 347–355. https://doi.org/10.14311/AP.2023.63.0347

Kusumah, M. N., & Yacoub, R. R. (2024). Analisis QoS dan SINPO untuk layanan audio video pada jaringan IndiHome di Kota Sanggau. *Jurnal Teknik Elektro Universitas Tanjungpura*, 14(1), 18–26.

Lugayizi, F. L., Gasela, N., & Bukohwo Michael, E. (2015). Performance evaluation of OSPF and EIGRP routing protocols for video streaming over next generation networks. *Academic Circle of Science and Interactive Journalism (ACSIJ)*, 4(6), 1–9.

Lyko, T., Broadbent, M., Race, N., Nilsson, M., Farrow, P., & Appleby, S. (2024). Improving quality of experience in adaptive low latency live streaming. *Multimedia Tools and Applications*, 83(6), 15957–15983. https://doi.org/10.1007/s11042-023-15895-9

Miswar, N., Riadi, I., & Dahlan, A. (2024). Performance analysis of BGP dynamic routing protocol using QoS for TCP and UDP services. *International Journal of Computer Network and Information Security*, 9(1), 45–56.

Mohammed, H. H. (2024). Real time video streaming over homogeneous systems. *NTU Journal of Pure Sciences*, 3(2), 77–85. https://doi.org/10.56286/s0h8bg27

Nakazato, J., Nakagawa, K., Itoh, K., Fontugne, R., Tsukada, M., & Esaki, H. (2024). WebRTC over 5G: A study of remote collaboration QoS in mobile environment. *Journal of Network and Systems Management*, 32(1), Article 20. https://doi.org/10.1007/s10922-023-09778-5

Negi, P., Negi, V., & Pendyala, S. K. (2025). Live video streaming using the Secure Reliable Transport (SRT) protocol: Architecture and performance insights. *International Journal of Multidisciplinary Research in Science, Engineering and Technology*, 8(7), 11142–11149. https://doi.org/10.15680/IJMRSET.2025.0807005

Peña-Ancavil, E., Estevez, C., Sanhueza, A., & Orchard, M. (2023). Adaptive Scalable Video Streaming (ASViS): An advanced ABR transmission protocol for optimal video quality. *Electronics*, 12(21), Article 4542. https://doi.org/10.3390/electronics12214542

Pratama, F. Y., Putrada, A. G., & Dawani, F. (2024). Analisis QoS pada WebRTC dengan mekanisme congestion control di jaringan lokal. *Jurnal Informatika dan Komputasi*, 16(2), 101–110.

Pratiwi, R. S., & Oktiawati, U. Y. (2023). Analisis QoS pada implementasi MPLS traffic engineering-Diffserv untuk layanan video streaming. *Journal of Internet and Software Engineering*, 4(2), 34–42.

Rachmawati, Munawar, & Tari, S. (2023). Analisis Quality of Service (QoS) radio streaming menggunakan Icecast pada Wi-Fi 802.11n. *Jurnal Litek: Jurnal Listrik Telekomunikasi Elektronika*, 19(1), 22–29. https://doi.org/10.30811/litek.v19i1

Santos, C. E. M., da Silva, C. A. G., & Pedroso, C. M. (2021). Improving perceived quality of live adaptative video streaming. *Entropy*, 23(8), Article 948. https://doi.org/10.3390/e23080948

Semuel, R. A., & Belutowe, Y. S. (2026). Analisis kinerja protokol multicast dalam jaringan komputer untuk aplikasi multimedia streaming. *Jurnal Publikasi Manajemen Informatika*, 5(1), 1–17. https://doi.org/10.55606/jupumi.v5i1.4235

Tüker, M., Karakış, E., & Sayıt, M. (2026). Protocol-oriented comparative analysis of video streaming over DASH, TCP, UDP, and BPP. *Scientific Journal of Mehmet Akif Ersoy University*, 9(1), 30–46. https://doi.org/10.70030/sjmakeu.1911041