IJoICT
ISSN 2356-5462
https://journals.telkomuniversity.ac.id/ijoict
Intl. Journal on ICT
Vol. 12, No. 1, Jun 2026. pp. 56-65
doi:10.21108/ijoict.v12i1.10893

Analysis of the Effect of QUIC and TCP on Quality of Experience (QoE) during Handover

Muhammad Adlan Hafizha[1*], Aji Gautama Putrada[2], Ryan Lingga Wicaksono[3]

[1,2,3]*Undergraduate Program of Informatics, School of Computing*
Telkom University, Bandung, Indonesia

[*]madlanh@student.telkomuniversity.ac.id

Abstract

Video streaming on mobile devices experiences interruptions when the wireless connection switches to a new network. This study aims to evaluate Quick UDP Internet Connections (QUIC) protocol's performance as an alternative to Transmission Control Protocol (TCP) that oftentimes encounters performance setbacks during handovers. The simulation was performed using Mininet-WiFi emulator with a topology of two Access Points with various bandwidth capacities (10 Mbps to 1000 Mbps) and one mobile station (client). The simulations were repeated 30 times to ensure statistical validity for packet loss, latency, throughput, and buffering time metrics. Simulation results show that QUIC outperforms TCP in network metrics performance. QUIC's Connection ID and 0-RTT initiation features reduce packet loss by up to 6.39%, maintain latency below 0.14 ms, and instantly restore throughput. However, an anomaly was found at high bandwidth (>500 Mbps) where buffering time of QUIC was longer due to the computational load of TLS 1.3 decryption in user space. Although QUIC is strong in wireless mobility, the CPU processing of the receiver's device must be optimized on high-speed network.

Keywords: QUIC, TCP, handover, quality of experience, video streaming, Mininet-WiFi

I. Introduction

Video is a visual medium that combines motion pictures and sound, and has become an important form of entertainment and communication in the digital world [1]. Various types of videos, such as educational videos, entertainment content, and movies, have become consumer goods accessed through various devices. An easily accessible technology for video is video streaming. With video streaming, users can watch videos without having to download them first [2].

Video streaming services, such as YouTube, use Transmission Control Protocol (TCP) as their transport protocol. TCP offers high reliability with data retransmission mechanisms, but has weaknesses in dynamic network environments [3]. As an alternative, Quick UDP Internet Connections (QUIC) was developed on top of the User Datagram Protocol (UDP), which is more adaptive in dynamic network environments and has a connection migration feature that allows connection to remain active even when there is a network switch [4]. However, in high-speed networks, QUIC experiences a decline in performance due to overhead on the receiver side [5].

Network handover is the process where a device switches connection from one Access Point to another. This process occurs because the signal strength received by the device weakens and switches to a stronger signal [6]. Poor handover can cause disruptions such as buffering, packet loss, and increased latency. With the connection migration feature in QUIC, the connection can be maintained during network handover, while TCP is more prone to disruption due to its packet retransmission mechanism and is affected by the device's movement speed [6].

Received on 13 Apr 2026. Revised on 18 Jun 2026. Accepted and Published on 22 Jun 2026.

Based on the matter above, our study aims to analyze and compare the performance of TCP and QUIC with video streaming during Wi-Fi to Wi-Fi handover at frequencies 2.4 GHz and 5 GHz. By conducting network simulation, this study is expected to provide insight into the performance of TCP and QUIC on video streaming during network transition.

II. LITERATURE REVIEW

Studies evaluating TCP dan QUIC protocols has been the subject of exploration by many researchers over the past years. Many studies have thoroughly examined the performance of the protocols in various network architectures and workloads, from static web traffic transmission, cloud computing, to media streaming services. Several researches in the following literature are summarized to serve as the theoretical foundation of this study and highlights protocols performance evaluation in wireless mobility environments, including handover process.

Shreedhar et al. evaluated and compared the performance of QUIC with TLS/TCP in web access, cloud storage, and video scenarios [2]. This evaluation tests handshake time, throughput, computational load (CPU), and quality of experience (QoE) on video. The networks include Germany with high bandwidth and high RTT, and India with low bandwidth and high RTT. The results show that QUIC is superior in handshake time, which is up to 140% lowen than TLS 1.2/1.3 on TCP in network with low bandwidth and high RTT, as well as video startup delay, which is up to 40% faster. However, TLS/TCP has higher throughput for large files (>20 MB to 2 GB) than QUIC when downloading files from Google Drive. Thus, QUIC provides better QoE due to its efficient loss recovery mechanism.

Kyratzis and Cottis evaluated the performance of QUIC and TCP in LTE networks using NS-3 as a simulator [3]. The main targets in this evaluation are throughput, flow fairness, file transfer time, and the effect of multiplexing on QUIC. The results show that QUIC is superior in stable state throughput, up to 2x greater than TCP. Transfer time for small-sized files is 22-27% faster than TCP, but is the same as TCP in poor signal quality. Furthermore, QUIC receives a larger share of available bandwidth than TCP, indicating that there is no flow fairness between QUIC and TCP. QUIC has other advantages such as 0-RTT handshake (cutting latency by 50 ms) and stream multiplexing, which increases throughput by 37% and overcomes HOL blocking, but its performance declines when packet reordering occurs in wireless networks.

Chaudhary et al. evaluated connection racing in video streaming in HTTP/3 browsers and improved quality of experience (QoE) [4]. They analyzed the impact of connection racing between QUIC and TCP on YouTube streaming media. Testing was conducted on more than 5,474 hours of streaming in Chromium browsers under various network conditions. The results showed that repeated connection racing can reduce QoE due to the instability of the transport protocol, especially in weak network conditions. Then, they modified the browser code to dynamically regulate connection racing based on RTT. The modification successfully improved QoE by up to 70% by minimizing unnecessary protocol switching. This study emphasizes the importance of protocol management in browsers of stable streaming.

Zhang et al. identified the causes of slow QUIC performance on high-speed networks [5]. They conducted a systematic examination by comparing the UDP+QUIC+HTTP/3 and TCP+TLS+HTTP/2 protocols. The methods used included cuRL and quic_client as tools for file download experiments, as well as testing on browsers such as Chrome, Edge, Firefox, and Opera. On networks with speed >500 Mbps, QUIC experienced a data speed reduction of up to 45.2% compared to HTTP/2. Through packet trace analysis and system profiling, it was revealed that the main cause was processing overhead on the receiver side due to the large number of data packets received. Based on the results, the use of UDP GRO on the receiver side is recommended to overcome this problem.

Yan and Yang identified the causes of deadlock in QUIC connection migration on Wi-Fi hotspot networks [6]. They evaluated the performance of passive QUIC migration by testing the opening of a web page containing 50 images and downloading a 500 KB file in a stable network and during suspension. The results showed that 65% of suspension cases experienced deadlock, where the server continued to send problematic packets because there was no migration notification. To address this issue, they provided a “Delay and Replay” (D&R) scheme solution on the relay. The solution provided successfully reduced the chance of deadlock to 3% and improved download latency.

III. RESEARCH METHOD

A. Simulation Procedure

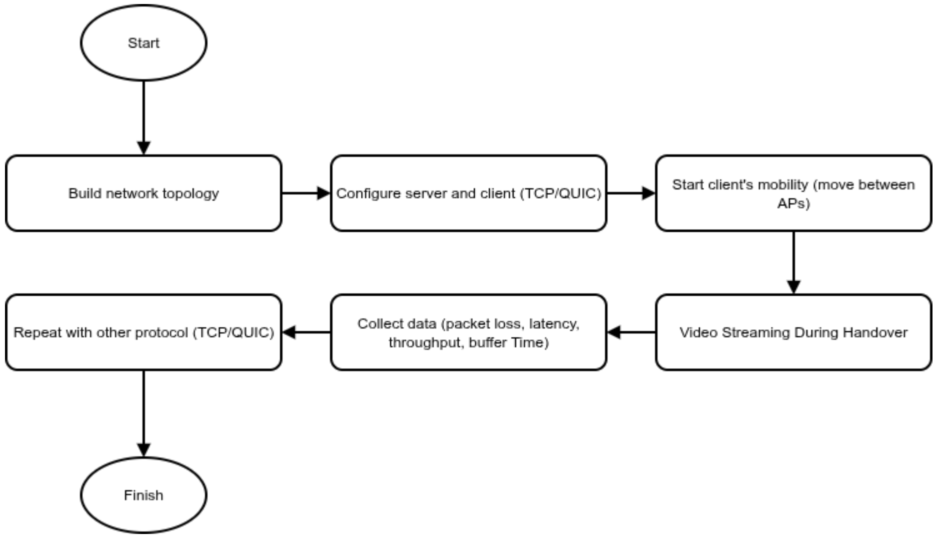

Fig. 1. Simulation Flowchart

Python script is used in the simulation flow to reduce manual execution bias. The testing procedure is as depicted in the flowchart (Fig. 1):

1. Topology Initialization: Building the nodes, determining coordinates of the APs, assigning IP addresses to each node, and limiting bandwidth parameters.
2. Server dan Client Configuration: Setting MPV media player on the client side and operating video server services for TCP and QUIC.
3. Start Mobility: Client automatically moves across APs at a constant speed. When client's signal (RSSI) to the connected AP starts to fall down, the system triggers a handover to the nearest AP with a stronger signal.
4. Data Collection: The evaluated metrics are recorded in real-time every second during client's mobility.

The overall procedure for each scenario runs for 130 seconds. This duration was taken into consideration based on the time required for the client to finish its whole mobility path, including the initial steady state, the signal disruption and handover, and the network recovery [3].

B. Simulation Environment and Topology

The simulation was performed using Mininet-WiFi [7]. It is a Software-Defined Wireless Network (SDWN) emulation platform that allows real-time evaluation of node mobility by simulating physical wireless parameters like Received Signal Strength Indicator (RSSI). This platform was operated on Ubuntu 22.04 operating system to provide a reliable and isolated testing environment which is a method frequently used in cellular and wireless network research [8].

The topology was designed linearly to simulate the dynamic mobility of client within the wireless coverage area. The node structure includes two access points (APs), one mobile station (client), one switch, and video server (Fig. 2).

All nodes in this topology are assigned static IP addresses to separate the protocols' performance and minimize additional latency caused by the Dynamic Host Configuration Protocol (DHCP). Also, in wireless link between access points and mobile station, no parameters for artificial air propagation delay or signal fading are included. This approach ensures that any delays or packet loss are only caused by physical handover processes and protocol features, instead of by IP address reassignment at the network layer and propagation delay [2].

This study focuses on end-to-end performance evaluation of transport layer (Layer 4) protocols. Network handover simulation uses the standard RSSI-based method built into Mininet-WiFi. Since TCP and QUIC operate without relying on intermediate device configurations at the data link layer, the simulation does not modify or propose new handover algorithm on Access Point (APs), so the evaluated architectural design is universal and can be applied to real-world APs [6], [9].

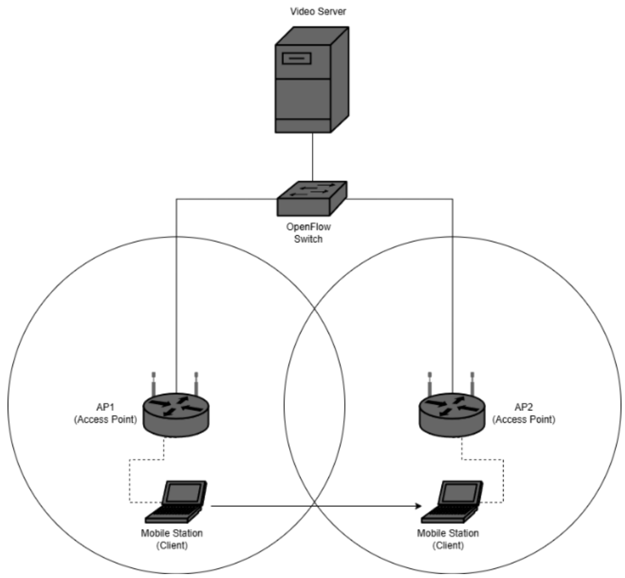

Fig. 2. Network Topology

C. Test Scenarios

The test focused on client mobility between APs that trigger the handover mechanism. Access Points were configured using a variety of 2.4 GHz and 5 GHz channel frequencies to create realistic heterogeneous network environment. The test scenarios were divided into two main categories:

1. Inter-frequency Handover: A scenario of client's device moves and connects from one AP to another with different frequencies, namely from 2.4 GHz to 5 GHz, and from 5 GHz to 2.4 GHz.
2. Intra-frequency Handover: A scenario of client's device moves and connects from one AP to another with the same frequencies, namely from 2.4 GHz to 2.4 GHz, and from 5 GHz to 5 GHz.

Each scenario used variance bandwidth capacities from 10 Mbps, 100 Mbps, 500 Mbps, to 1000 Mbps.

D. Video Traffic Generation

Video streaming data is transmitted from server to client using the HTTP/1.1 for TCP and HTTP/3 for QUIC. Mpv is the media player that used to video streaming in this simulation. The selection of mpv media player was based on its ability to run via a command-line interface and its JSON-based Inter-Process Communication (IPC) feature, which allows for the precise extraction of Quality of Experience (QoE) parameters in a headless simulation environment without a Graphical User Interface (GUI) [10].

The transmitted video sample has a 1080p resolution with an average bit rate of 2609 Kbps. This specification was chosen because it resembles the standard workload of today's media consumption on mobile devices, allowing for an accurate visualization of the protocols' performance when handling large data packets [11].

E. Metrics Evaluation and Data Analysis

This study uses four performance metrics to evaluate TCP and QUIC performance, including:

1. Packet Loss (%): Reliability of the protocols during handover is shown by the percentage of data packet lost.
2. Latency (ms): The protocols' responsiveness to route disconnections is indicated by the delay time estimated from Round-Trip Time (RTT).
3. Throughput (Mbps): The actual rate of successful data transfer over a network connection measured with the iperf tool.
4. Buffering Time (s): Total delay time during video streaming on the client side to hold data in transmission.

Each scenario in this study was repeated 30 times ($N = 30$) to assure statistical validity and reduce the impacts of random anomalies, which are common in wireless simulations. The simulation's sample size was chosen based on the Central Limit Theorem, which stipulates that a sample size of $N \geq 30$ results in a data distribution that asymptotically similar to a normal distribution [8], [12]. This iterative approach considerable improves confidence in the estimates and the reliability of the collected data.

The collected data were then analyzed to get the mean value and variance of deviance using standard deviation. The mean value ($\bar{X}$) was calculated using Equation (1).

$$\bar{X} = \frac{1}{N}\sum_{i=1}^{N} X_i \tag{1}$$

Protocol performance stability was calculated through standard deviation (S) using Equation (2).

$$S = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(X_i - \bar{X})^2} \tag{2}$$

Where $\bar{X}$ is the mean of metrics evaluation, N is the total time of repetition ($N \geq 30$), X_i is the actual result at iteration i, and S is the sample standard deviation. In the data visualization, the standard deviation is shown as error bars to show the protocol's consistency and stability.

IV. RESULTS AND DISCUSSION

A. *Results*

TCP and QUIC performance were assessed using several mobility simulations that triggered handover processes across varying bandwidth capacities. This analysis focused on four metrics: packet loss, latency, throughput, and buffering time. The results indicate that, despite some computational overload at the application layer, QUIC successfully maintained connection stability throughout network handovers. Based on the results of packet loss (Fig. 3), TCP shows a relatively high and constant packet loss rate throughout all test scenarios, resulting from 11% to 13% loss. While QUIC exhibited better resilience, specifically in the intra-frequency scenario. In 2.4 GHz to 2.4 GHz with a bandwidth of 100 Mbps handover scenario, QUIC was able to reduce packet loss to 6.39%. While TCP has higher packet loss to 11.06% under the same scenario

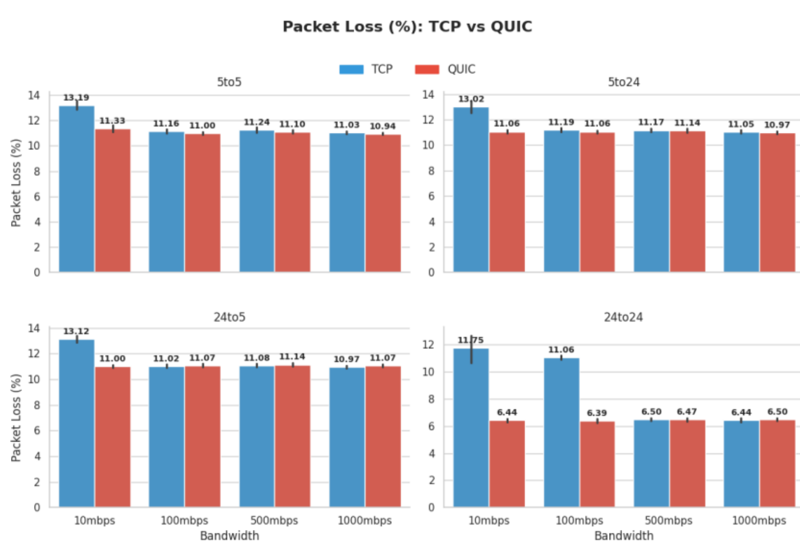

Fig. 3. Packet Loss Evaluation

.

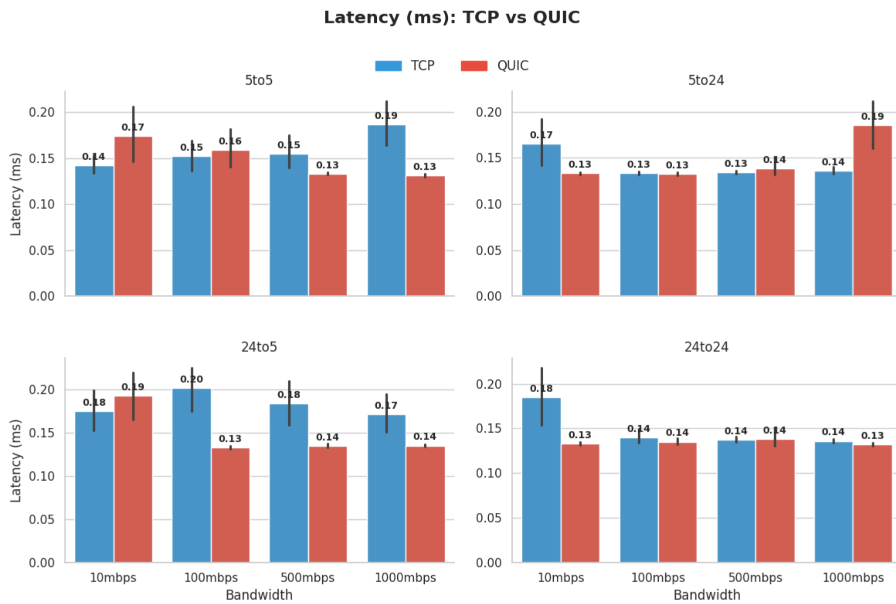

Fig. 4. Latency Evaluation

In Fig. 4 shows QUIC has lower times latency across all test scenarios, resulting between 0.13 ms and 0.19 ms, which indicates that QUIC does not cause any delays during handover. While TCP showed a higher times latency, resulting from 0.13 ms to 0.20 ms. This high latency for TCP indicates an increase of wait time throughout the test scenarios. While Fig. 5 shows that at the exact time of the physical disconnection caused by handover from one Access Point to another, TCP experiences an extreme latency spike for most of the times. This spike shows a performance gap. On the other hand, QUIC's latency spike shows minimal fluctuations. QUIC proved capable of reducing the effects of network disconnections quickly, preventing extreme latency spikes, and immediately stabilizing the connection back to its regular latency.

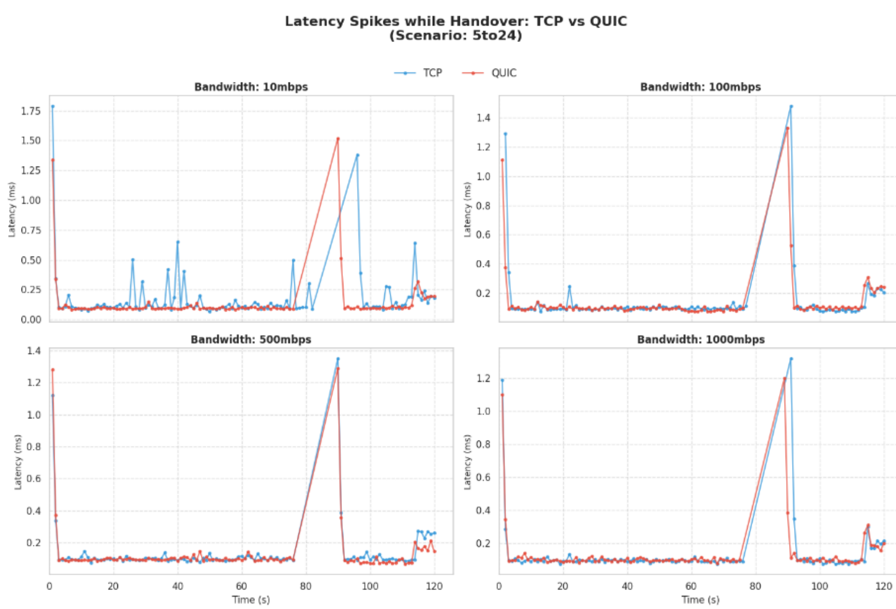

Fig. 5. Latency Spike in Scenario 5 GHz to 2.4 GHz

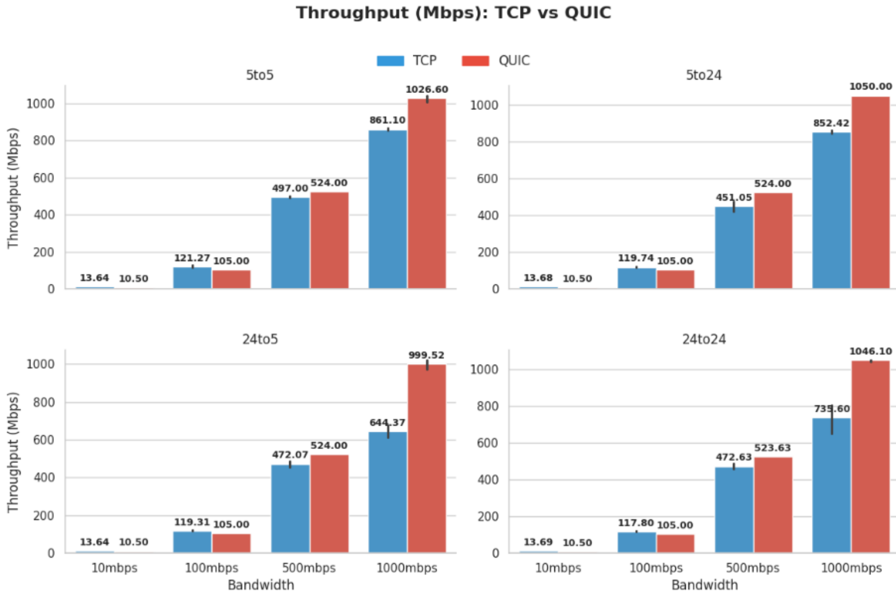

Fig. 6. Throughput Evaluation

Throughput performance shows a striking difference between the protocols, particularly in terms of network capacity variation (Fig. 6). TCP and QUIC performance are comparatively equivalent and can reach the full capacity under low bandwidth (10 Mbps and 100 Mbps). However, high-speed networks (500 Mbps and 1000 Mbps) exhibit significant performance variance. QUIC outperformed TCP, which only managed to attain 852.42 Mbps, with an average throughput of 1050.00 Mbps in a 5 GHz to 2.4 GHz scenario with a bandwidth of 1000 Mbps. A sharp performance degradation pattern can be seen in the TCP curve (Fig. 7). TCP throughput drastically decreases at the handover point, even getting close to 0 Mbps. TCP curve also shows very slow recovery processes before finally returning to its full-speed. On the contrary, QUIC curve shows resilience, although the protocol also experiences a minor drop in data rate due to physical interruptions. QUIC curve immediately rises, restoring transmission speed the moment the connection is reestablished, ensuring high throughput is maintained throughout the tests.

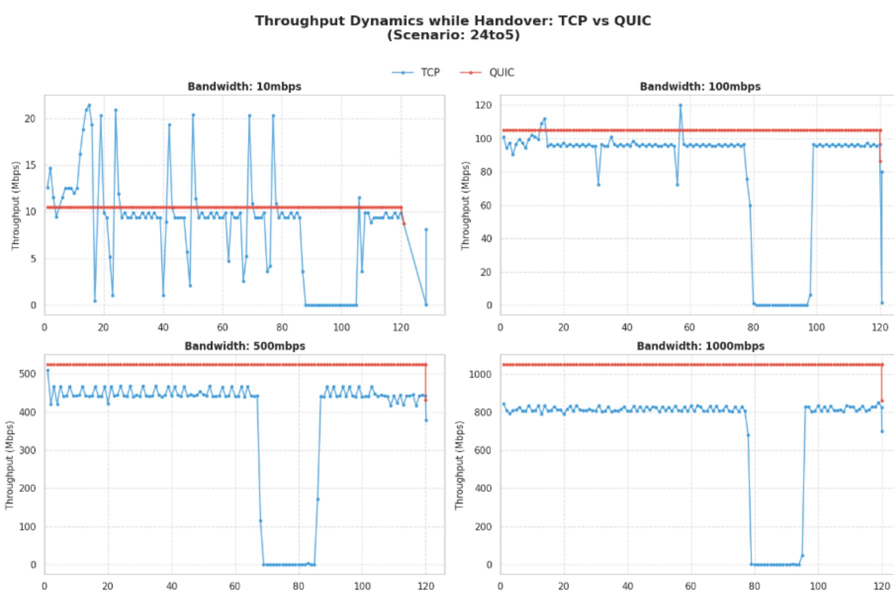

Fig. 7. Throughput Dynamics in Scenario 2.4 GHz to 5 GHz

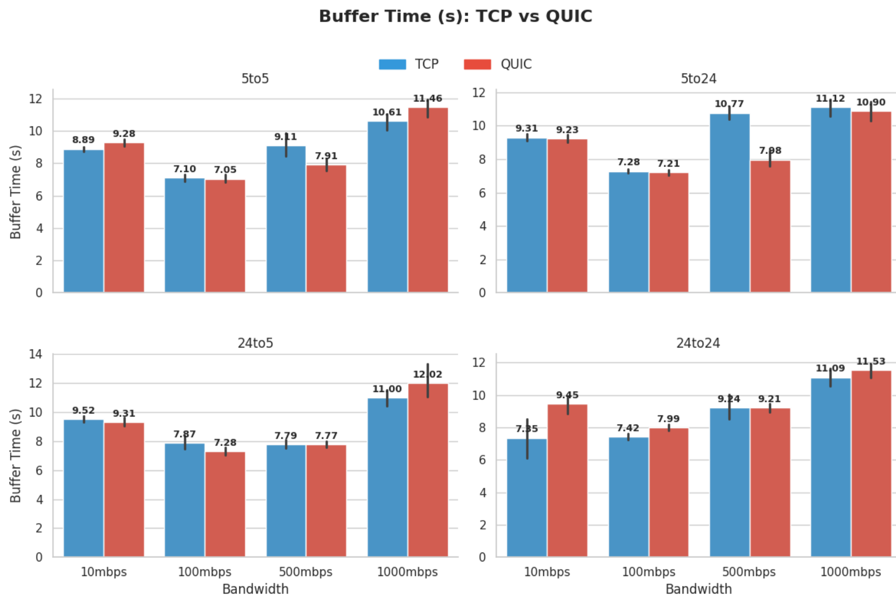

Fig. 8. Buffering Time Evaluation

Although QUIC dominates all network metrics, measurements of application-level metric show unexpected results. In Fig. 8 shows that on high bandwidth network, in this case 1000 Mbps, the video player in QUIC transmission experienced longer stalling or buffering times. For example, in 5 GHz to 5 GHz with bandwidth of 1000 Mbps scenario, QUIC resulted of 11.46 seconds buffering duration, while TCP was able to resume video with 10.61 seconds that is shorter buffering duration.

B. Discussion

Data from the test results show that the transport layer architecture determines the level of a protocol's resilience there are physical connectivity disruptions. High packet loss rates and extreme latency spikes in TCP are caused by from its strict connection model. TCP validates connections based on a 4-tuple identification (source/destination IP address and port), so TCP considers the communication session to have been terminated completely when client switches Access Points [13]. This event urges TCP to renegotiate the 3-way handshake process that increases latency. While QUIC's performance is supported by its Connection Migration mechanism. With the implementation of Connection ID (CID) and regardless of the physical IP address, communication sessions in QUIC remain uninterrupted at the application layer even the route changes due to the handover [14], [9]. Integrated with Zero Round-Trip Time (0-RTT), QUIC eliminates reconnection delays, resulting in stable latency that is resilient to client mobility. In Fig. 4 shows very low average latency values, from 0.13 ms to 0.20 ms. These values represent the purity of processing time within the emulator kernel, which is isolated from physical transmission delays [7]. Therefore, latency metric is intended primarily as a comparative baseline for demonstrating the efficiency of QUIC's 0-RTT compared to TCP's 3-way handshake. In addition, since TCP and QUIC operate at the transport layer on an end-to-end basis between server and client, this implementation is universal and does not require any hardware modifications or configurations on Access Point in real-world environment [6].

The architectural difference also has a direct impact on the recovery of transmission throughput. In Fig. 7 shows drastic drop in TCP transmission rate to nearly 0 Mbps is caused by a misinterpretation of the Additive Increase Multiplicative Decrease (AIMD)-based Congestion Control algorithm. The accumulated packets during handover cause buffer overflow and packet loss, which TCP mistakenly interprets as severe network congestion [15]. As a result, TCP reduces the congestion window (cwnd) size exponentially by 63.3% and goes into the slow-start phase that is causing throughput recovery to proceed very slowly [15]. While QUIC addresses this interruption using stream multiplexing architecture. Packet loss caused by burst loss during handover affects only the problematic stream without delaying the transmission of other streams. This capability effectively prevents Head-of-Line (HoL) Blocking and allows the transmission rate to rebound instantly to its highest speed [2].

Although QUIC dominates network metrics, an analysis of buffering time reveals an unexpected result. Under high-speed bandwidth network conditions, QUIC fast transmission rate actually shows longer video buffering duration compared to TCP. This contradiction is an example of a computational trade-off or CPU-bound bottleneck. TCP operates natively in the operating system's kernel space with highly efficient CPU optimization and minimal overhead. While QUIC operates in user space and requires Transport Layer Security or TLS 1.3 encryption on all data packets and their headers [16]. As a result, when the client's media player or mpv receives a Gigabit-speed data stream, the receiver's CPU struggles to complete the decryption process from user space on time (decoding delay). This delay can cause to sudden stops of the video playback which affecting user experience [16]

Several studies have evaluated the performance comparison between TCP and QUIC protocols in various scenarios. Paper [3] evaluated the protocols using NS-3 simulation, paper [13] used M2HO solution to mitigate the negative effects of handover on TCP, and paper [9] proposed mQUIC method to support seamless handover. In this study, we compared TCP and QUIC performance in video streaming during handover process. This study found a conflicting result on buffering duration. Although QUIC's adaptive features have proven effective in maintaining throughput, yet it can be compromised by high processing overhead when handling high-speed network (>500 Mbps). This study contribution reveals that while QUIC is very important for handling handover stability, its implementation in video streaming requires additional optimization for client efficiency.

V. CONCLUSION

This study shows that QUIC protocol generally demonstrates dominance network performance than TCP in supporting video streaming services in handover scenarios. This advantage is empirically proven through lower packet loss and latency rates, as well as more instantly restored throughput, supported by the effectiveness of the connection ID (CID) feature and 0-RTT initiation in eliminating the need for connection re-handshake during access point transitions. This zero-delay connection recovery performance is proven to sustain line-rate stability up to the maximum bandwidth limit of 1000 Mbps.

Although QUIC successfully overcomes HoL Blocking constraints at the transport layer to maximize bandwidth utilization, evaluation of application-level metric reveals a crucial performance trade-off on high-speed networks. Under these conditions, QUIC buffering time is slightly longer than TCP due to the high computational load of the TLS 1.3 decryption process running in user space. Therefore, QUIC implementation is highly recommended to maintain data continuity stability on dynamic wireless network, with the qualification that optimizing the processing efficiency of receiving devices remains a major challenge that needs to be addressed in the future to keep pace with high-speed network capabilities.

DATA AND COMPUTER PROGRAM AVAILABILITY

Data and program used in this paper can be accessed in the following site https://github.com/madlanh/tcp_and_quic_handover_vidStream.git.

ACKNOWLEDGMENT

The authors would like to thank PT Packet Systems Indonesia for the financial support provided through the PSI Scholarship Program for this study. This support has been contributed to facilitating the study process.

REFERENCES

[1] J. Apostolopoulos, W.-T. Tan, and S. J. Wee, "Video Streaming: Concepts, Algorithms, and Systems", in *Handbook of Video Databases: Design and Applications*, 2003, pp. 831–864.

[2] T. Shreedhar, R. Panda, S. Podanev, and V. Bajpai, "Evaluating QUIC Performance Over Web, Cloud Storage, and Video Workloads", *IEEE Trans. Netw. Serv. Manage.*, Vol. 19, No. 2, pp. 1366–1381, Jun. 2022, doi: 10.1109/TNSM.2021.3134562.

[3] A. I. Kyratzis and P. G. Cottis, "QUIC vs TCP: A Performance Evaluation over LTE with NS-3", *CN*, Vol. 14, No. 01, pp. 12–22, 2022, doi: 10.4236/cn.2022.141002.

[4] S. Chaudhary, N. K. Shukla, P. Sachdeva, S. Chakraborty, and M. Maity, "Managing Connections by QUIC-TCP Racing: A First Look of Streaming Media Performance Over Popular HTTP/3 Browsers", *IEEE Trans. Netw. Serv. Manage.*, Vol. 21, No. 3, pp. 2962–2976, Jun. 2024, doi: 10.1109/TNSM.2024.3371069.

[5] X. Zhang *et al.*, "QUIC is not Quick Enough over Fast Internet", in *Proceedings of the ACM Web Conference 2024*, May 2024, pp. 2713–2722. doi: 10.1145/3589334.3645323.

[6] Y. Yan and Z. Yang, "When QUIC's Connection Migration Meets Middleboxes A Case Study on Mobile Wi-Fi Hotspot", in *2021 IEEE Global Communications Conference (GLOBECOM)*, Madrid, Spain: IEEE, Dec. 2021, pp. 1–6. doi: 10.1109/GLOBECOM46510.2021.9685048.

[7] R. R. Fontes, S. Afzal, S. H. B. Brito, M. A. S. Santos, and C. E. Rothenberg, "Mininet-WiFi: Emulating Software-Defined Wireless Networks", in *2015 11th International Conference on Network and Service Management (CNSM)*, 2015, pp. 384–389. doi: 10.1109/CNSM.2015.7367387.

[8] T. Mahboob, S. Tariq Shah, M. Choi, S.-H. Kim, and M. Young Chung, "Multi-Operator Spectrum and MEC Resource Sharing in Next Generation Cellular Networks", *IEEE Access*, vol. 12, pp. 91634–91648, 2024, doi: 10.1109/ACCESS.2024.3422073.

[9] S.-Y. Kim and S.-J. Koh, "mQUIC: Use of QUIC for Handover Support with Connection Migration in Wireless/Mobile Networks", *IEEE Commun. Mag.*, Vol. 62, No. 4, pp. 128–134, Apr. 2024, doi: 10.1109/MCOM.023.2300083.

[10] S. Kiani Mehr, P. Jogalekar, and D. Medhi, "Moving QoE for Monitoring DASH Video Streaming: Models and A Study of Multiple Mobile Clients", *Journal of Internet Services and Applications*, Vol. 12, No. 1, p. 1, Apr. 2021, doi: 10.1186/s13174-021-00133-y.

[11] D. Dias, M. Luís, P. Rito, and S. Sargento, "A Software Defined Vehicular Network using Cooperative Intelligent Transport System Messages", *IEEE Access*, Vol. 12, pp. 93152–93170, 2024.

[12] R. A. Khurma, E. Alhenawi, M. Braik, F. A. Hashim, A. Chhabra, and P. A. Castillo, "A Bio-Medical Snake Optimizer System Driven by Logarithmic Surviving Global Search for Optimizing Feature Selection and its Application for Disorder Recognition", *Journal of Computational Design and Engineering*, Vol. 10, No. 6, pp. 2361–2383, Nov. 2023, doi: 10.1093/jcde/qwad101.

[13] G. Rahate and N. Chopade, "Realistic Vertical Handoff Predictive Trigger Thresholding in Heterogeneous Networks", *Ingenierie des Systemes d'Information*, Vol. 27, No. 4, p. 557, 2022.

[14] M. Tauqeer, M. Gohar, S. Koh, and H. Alquhayz, "Use of QUIC for Mobile-Oriented Future Internet (Q-MOFI)", *Electronics*, Vol. 13, No. 2, p. 431, Jan. 2024, doi: 10.3390/electronics13020431.

[15] Z. Liu *et al.*, "M2HO: Mitigating The Adverse Effects of 5G Handovers on TCP", in *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking*, 2024, pp. 1089–1103.

[16] B. Jaeger, J. Zirngibl, M. Kempf, K. Ploch, and G. Carle, "QUIC on the Highway: Evaluating Performance on High-Rate Links", in *2023 IFIP Networking Conference (IFIP Networking)*, 2023, pp. 1–9. doi: 10.23919/IFIPNetworking57963.2023.10186365.