TRANSPORT-PROTOCOL RESILIENCE FOR SMART-CITY DIGITAL SERVICES UNDER UNSAFE NETWORKS: COMPARATIVE EVIDENCE FROM QUIC/UDP AND TLS/TCP

Ping Xu

Urban digital services increasingly depend on resilient, low-latency, and secure network transport. This requirement is especially acute in smart-city environments, where edge-connected public services, web platforms, and data-intensive civic systems must remain available under hostile traffic conditions. This paper presents a structured comparative study of Quick UDP Internet Connections (QUIC/UDP) and TLS over Transmission Control Protocol (TLS/TCP) under unsafe network conditions, with emphasis on operational implications for smart-city digital infrastructure. A real-world four-node platform comprising a client, server, router, and attacker is used to evaluate the two protocol stacks under three classes of attack: denial-of-service attacks, man-in-the-middle attacks, and traffic analysis attacks. Efficiency is assessed through client delay, packet loss, server CPU utilization, and server memory utilization, while security is examined using website-fingerprinting performance and protocol-conditional model accuracy. The experimental evidence shows a clear attack-dependent asymmetry. Under connection-flooding denial-of-service conditions, QUIC maintains substantially lower latency; at 100 attack handshakes per second, the client delay observed with TLS/TCP is almost 20 times that of QUIC/UDP. TCP packet loss begins to increase once the attack rate exceeds 60 handshakes per second, and the TCP server becomes fully CPU-bound above roughly 50 attack handshakes per second, whereas QUIC does not exhaust CPU capacity even at 100 attacks per second. Under slowloris conditions, both protocols avoid packet loss and maintain low delay, but QUIC retains a memory-efficiency advantage. Under man-in-the-middle attacks, the performance ordering reverses: QUIC incurs longer client delays and greater data loss than TLS/TCP, while server replay produces delays in the 40–50 ms range for both stacks, with QUIC remaining slightly slower. In traffic analysis, TCP achieves the higher classifier F1-score (0.67495 versus 0.63497), indicating stronger overall confidentiality in this implementation, although QUIC traffic patterns remain harder to learn by machine-learning methods. Taken together, the findings support a deployment principle of contextual protocol selection for smart-city systems: QUIC is preferable when availability and connection efficiency dominate, whereas TLS/TCP is preferable when manipulation resistance and confidentiality are the primary concern.

Index Terms — smart cities, urban digital infrastructure, QUIC, TCP, TLS, unsafe networks, denial-of-service, man-in-the-middle, traffic analysis

https://doi.org/10.66033/judsc2025-211

INTRODUCTION

Cities now rely on dense layers of digital infrastructure to deliver planning, mobility, utility, administrative, and public-information services. Web portals, edge gateways, sensor platforms, and citizen-facing applications all depend on transport protocols that can preserve service continuity while maintaining acceptable latency and confidentiality. In this setting, transport choice is not merely a low-level networking preference; it is an operational design decision with consequences for urban resilience, system safety, and public-service reliability.

Quick UDP Internet Connections (QUIC) has emerged as a prominent transport for modern web systems because it combines transport negotiation and cryptographic setup, supports connection migration, and reduces connection-establishment overhead relative to conventional TLS/TCP in favorable conditions [2, 5, 6]. These properties make QUIC attractive for smart-city digital services that depend on responsive edge-to-cloud exchanges. Yet lower handshake overhead does not automatically imply superior behavior under attack. When public-service systems are exposed to malicious traffic, protocol performance depends on the interaction between protocol mechanics and attack semantics.

The empirical record demonstrates that QUIC and TLS/TCP do not dominate one another uniformly. In controlled comparative testing under unsafe network conditions, QUIC/UDP performs better than TLS/TCP in denial-of-service (DoS) conditions, especially where repeated connection setup dominates the workload, but TLS/TCP becomes more efficient under man-in-the-middle (MitM) manipulation and more secure overall in traffic-analysis terms [1, 3, 4]. This attack-dependent reversal is highly relevant for smart-city computing, where service back-ends may encounter both volumetric pressure and integrity-oriented disruption.

This manuscript presents a polished, source-grounded comparative analysis of QUIC/UDP and TLS/TCP under unsafe network conditions, written for an urban-development and smart-cities readership. Rather than introducing unsupported follow-on metrics, it focuses strictly on the empirical platform, measurements, and conclusions substantiated by the original experimental record. The paper contributes three forms of value. First, it organizes the protocol comparison around smart-city operational priorities: availability, efficiency, and confidentiality. Second, it consolidates the experimentally reported thresholds, ranges, and platform characteristics into a clearer academic structure. Third, it translates the networking findings into deployment implications for urban digital infrastructure, including edge services and public-facing platforms.

BACKGROUND AND SMART-CITY RELEVANCE

Why transport protocols matter in urban digital systems

Smart-city systems depend on digital exchanges that are often continuous, time-sensitive, and distributed across heterogeneous hardware. Public dashboards, permit portals, emergency-notification systems, mobility platforms, and urban sensing backbones all require a transport substrate that remains reliable under adverse conditions. In practice, the wrong transport choice can amplify congestion, degrade response times, or expose communication patterns that weaken confidentiality.

In these settings, three requirements dominate:

1. **Availability under load**, because public systems must remain reachable when request rates surge or when attack traffic competes with legitimate demand.
2. **Integrity under manipulation**, because intercepted or replayed traffic can corrupt service state even

when links remain nominally online.

3. **Confidentiality under observation**, because encrypted traffic may still leak information through packet structure, timing, and auxiliary protocol behavior.

These requirements make the QUIC-versus-TCP decision directly relevant to smart-city research, particularly in contexts involving edge computing, digital public services, and secure urban data exchange.

Protocol characteristics

QUIC preserves several desirable properties of modern secure transport while reducing setup overhead. The source study highlights seven major functionalities in comparing TCP, UDP, and QUIC: connection ID and migration, 1-RTT handshakes, 0-RTT handshakes, in-order delivery guarantees, connection verification, padding frames, and anti-amplification limits [1]. Table 1 restates these features in compact form.

Table 1: Key QUIC functionalities relative to TCP and UDP.

Functionality	TCP	UDP	QUIC
Connection ID and migration	No	No	Yes
1-RTT handshake	No	Yes	Yes
0-RTT handshake	No	Yes	Yes
In-order delivery guarantee	Yes	No	Yes
Connection verification	Yes	No	Yes
Padding frames	No	No	Yes
Anti-amplification limit	Yes	No	Yes

These capabilities explain why QUIC can outperform TLS/TCP in rapid connection-establishment scenarios. They do not, however, guarantee superior behavior when the dominant stressor is packet manipulation, replay, or protocol-observable leakage.

EXPERIMENTAL DESIGN

Unsafe-network attack scenarios

The study evaluates QUIC/UDP and TLS/TCP under three broad classes of network attack [1]:

1. **Denial-of-service attacks**, represented by connection flooding and slowloris, to test availability and efficiency under pressure on connection management and server resources.
2. **Man-in-the-middle attacks**, represented by immediate client echo and random server replay, to test resilience against packet interception, reflection, and replay.
3. **Traffic analysis attacks**, implemented as website fingerprinting, to examine confidentiality risks in encrypted traffic patterns.

The attack selection is methodologically important because it covers different operational failure modes. Connection flooding emphasizes handshake overhead; slowloris emphasizes long-lived connection occupation;

immediate client echo emphasizes endpoint confusion via reflected client packets; random server replay emphasizes stale but plausible server responses; and website fingerprinting evaluates what an observer can infer from encrypted traffic without reading its content.

Platform architecture and hardware

The experimental platform consists of four nodes: a client, a server, a router, and an attacker. The server uses Nginx, enabling a controlled comparison of TCP and QUIC in the same web-server environment. The router is a Raspberry Pi configured with a generic DHCP service to connect the other nodes. To reduce confounding factors, experiments are conducted on a local wireless network, endpoints are kept consistent across tests, and systems are restarted between experiments so that one attack does not influence the next [1].

Table 2 reports the hardware configuration used in the source experiments.

Table 2: Testing platform hardware specifications.

Device	CPU	Threads	Max speed	Memory	Operating system
Server	Celeron N3050	2	2.16 GHz	4 GB	Ubuntu
Client	Ryzen 7 3700X	16	4.4 GHz	32 GB	Arch Linux
Attacker	Ryzen 7 5800H	16	4.4 GHz	16 GB	Arch Linux
Router	ARM Cortex-A7	4	900 MHz	1 GB	Raspbian

Measurement strategy

The evaluation uses five metrics [1]:

1. **Client delay**: the time from handshake initialization to connection closure after data is received.
2. **Loss rate**: the ratio of lost data to total transmitted data.
3. **Server CPU utilization**: measured per process at 0.1 s intervals over 50 s.
4. **Server memory utilization**: observed over a 50 s interval.
5. **Traffic-analysis security**: assessed through macro F1-score and protocol-conditional model accuracy.

For the traffic-analysis experiment, 50,000 packets are collected from three Google services—Sheets, Docs, and Slides—and then preprocessed for a deep convolutional neural network used in website fingerprinting [1]. This provides a concrete empirical basis for comparing protocol-observable information leakage.

RESULTS

Denial-of-service performance

The connection-flooding experiment shows the clearest efficiency advantage for QUIC. When the attack rate reaches 100 attack handshakes per second, the client delay observed with TLS/TCP is almost 20 times the delay observed with QUIC/UDP [1]. This gap is attributed to handshake structure: QUIC completes key

exchange and transport setup in a shorter initialization sequence, whereas TLS/TCP must absorb both TCP's three-way handshake and the TLS handshake.

The packet-loss threshold is also explicitly reported. Both protocols avoid packet loss when the attack rate remains below 60 attacks per second, but TCP packet loss increases sharply once the attack rate exceeds that level [1]. The source paper ties this behavior to CPU saturation on the server, caused by the greater handshake-processing demand of TLS/TCP.

Server-resource measurements reinforce the same conclusion. The TCP server becomes fully occupied when the flooding rate rises above roughly 50 attack handshakes per second, while QUIC does not exhaust CPU capacity even at 100 attack handshakes per second [1]. In addition, QUIC maintains lower memory use because it generates less handshake overhead and caches less state than TCP. Under short-term connection-flooding pressure, QUIC therefore offers the more resilient availability profile.

The slowloris experiment produces a different but still favorable pattern for QUIC. As the number of slowloris attack connections increases from 0 to 1000, both protocol stacks maintain low delay and avoid packet loss in the reported test conditions [1]. Because slowloris holds connections open rather than repeatedly forcing new handshakes, the connection-establishment advantage of QUIC becomes less pronounced. Even so, QUIC remains more memory efficient, and CPU utilization for both protocols is reported as rarely exceeding 5% throughout the experiment [1]. The principal conclusion is that QUIC clearly outperforms TCP under short-term DoS pressure and remains at least as effective, while more memory efficient, under long-lived DoS pressure.

Man-in-the-middle performance

Under MitM conditions, the relative ordering reverses. In the immediate client echo attack, the source study reports that QUIC requires longer time to accept a connection request than TCP and also loses more data [1]. The explanation is protocol-mechanical: QUIC packets are largely encrypted and authenticated, so additional processing is required at endpoints when malformed or reflected traffic disrupts the expected packet sequence.

In the random server replay attack, the server CPU and memory remain at baseline levels, but delay and packet-loss behavior again favor TLS/TCP [1]. The source paper reports that client delays for both protocols fall within the 40–50 ms range, with QUIC still requiring slightly longer delay than TCP [1]. Compared with immediate client echo, both protocols experience longer delays in the replay scenario because replay forces the server to process more connection activity. Nevertheless, the paper's overall interpretation is unambiguous: under MitM attacks, QUIC connections experience worse performance than TCP connections.

Traffic-analysis security

The traffic-analysis experiment evaluates confidentiality through website fingerprinting. A deep convolutional neural network is trained on encrypted traffic generated by the three target websites, and its performance is assessed on a 10% test subset [1]. The reported classifier outcomes are shown in Table 3.

Table 3: Website-fingerprinting classifier performance reported for the two protocol stacks.

Protocol	F1-score
QUIC	0.63497
TCP	0.67495

TCP attains the higher F1-score, indicating that in this implementation it affords stronger overall protection against the specific information leakage measured by the study [1]. At the same time, the authors emphasize a subtler point: QUIC traffic patterns are harder to learn by machine-learning methods, even though TCP is more secure overall in this implementation. The source interpretation is that much of the leakage associated with QUIC arises from the presence of UDP packets in the flow rather than from QUIC packet structure alone [1]. This is precisely why both F1-score and protocol-conditional model accuracy are needed to interpret confidentiality outcomes responsibly.

Consolidated comparative findings

Table 4 consolidates the empirically reported, source-grounded results that are most relevant for deployment decisions.

Table 4: Source-grounded comparison of protocol behavior under unsafe-network conditions.

Attack context	QUIC/UDP	TLS/TCP
Connection flooding	Lower client delay; at 100 attacks/s, TCP delay is almost 20 times higher; lower CPU and memory burden; no CPU exhaustion at 100 attacks/s	No packet loss below 60 attacks/s, but loss rises beyond 60; CPU becomes fully occupied above about 50 attacks/s
Slowloris (0–1000 attacks)	Low delay; no packet loss; lower memory usage	Low delay; no packet loss; higher memory usage
Immediate client echo	Longer delay and more data loss than TCP	Better delay and loss performance than QUIC
Random server replay	Delay remains in the 40–50 ms range but is slightly longer than TCP	Delay remains in the 40–50 ms range and is slightly shorter than QUIC
Traffic analysis	Lower classifier F1-score (0.63497); harder to learn by machine learning; leakage strongly associated with UDP within the flow	Higher classifier F1-score (0.67495); more secure overall in the tested implementation

DISCUSSION: IMPLICATIONS FOR SMART-CITY INFRASTRUCTURE

The central result is not that one protocol is universally superior. Rather, the evidence supports a conditional deployment logic. When urban systems face availability-oriented stress—especially rapid connection churn associated with volumetric or handshake-heavy attacks—QUIC offers clear operational advantages. Its lower connection-establishment overhead allows smart-city services to retain responsiveness under conditions that push TLS/TCP into CPU saturation and packet loss.

By contrast, when urban systems face manipulation-oriented threats, especially replay or reflection designed to disrupt protocol state, TLS/TCP offers a more conservative and effective operating profile. For municipal

platforms that prioritize transaction integrity, administrative reliability, and predictable behavior under packet tampering, TLS/TCP remains the safer choice.

The confidentiality findings further reinforce the importance of deployment context. In this implementation, TCP is more secure overall under the study's traffic-analysis metrics, yet QUIC traffic is harder to learn by machine-learning methods. For smart-city platforms, this means security assessment should not be reduced to a single metric. Systems handling sensitive service metadata, citizen access patterns, or operational telemetry should evaluate both overall leakage and what specific lower-layer artifacts are contributing to that leakage.

These findings map naturally to smart-city architecture. Edge-heavy and latency-sensitive urban services may benefit from QUIC where public availability is the dominant risk. Administrative systems, supervisory control portals, and integrity-sensitive applications may benefit from TLS/TCP where traffic manipulation is the more credible threat. In practice, protocol selection should be tied to service function, threat model, and acceptable failure mode.

LIMITATIONS

The conclusions are bounded by the tested implementation and the specific attack scenarios. The platform is a controlled local network, and the observed results depend on the particular hardware, server software, and attack scripts used in the study. The traffic-analysis experiment is also limited to three Google services and a specific deep-learning pipeline. Accordingly, the findings should be interpreted as implementation-grounded operational evidence rather than as universal claims about all QUIC or TCP deployments.

Even with these limitations, the reported thresholds and performance reversals are valuable because they expose the conditions under which protocol assumptions fail. For smart-city research, this is often more useful than an abstract claim of average superiority: planners and system designers need to know when a protocol breaks down, not only when it performs well.

CONCLUSION

Secure and efficient transport is foundational to urban digital resilience. The comparative evidence synthesized here shows that QUIC/UDP and TLS/TCP have distinct strengths under unsafe network conditions, and those strengths align with different smart-city operational priorities. QUIC is the more efficient choice under denial-of-service conditions, particularly when repeated connection setup dominates system load. TLS/TCP is the stronger choice under man-in-the-middle manipulation and provides better overall confidentiality in the reported traffic-analysis experiment. For smart-city digital services, the most defensible engineering position is therefore context-aware protocol deployment rather than universal protocol preference. Transport selection should follow the dominant urban service requirement—availability, integrity, or confidentiality—and should be evaluated as part of infrastructure resilience planning rather than as a purely backend networking decision.

REFERENCES

[1] A. Simpson, M. Alshaali, W. Tu, and M. R. Asghar, "Quick UDP Internet Connections and Transmission Control Protocol in unsafe networks: A comparative analysis," *IET Smart Cities*, vol. 6, no. 4, pp. 351–360, 2024.

[2] J. Iyengar and M. Thomson, "QUIC: A UDP-Based Multiplexed and Secure Transport," RFC 9000, IETF, 2021.

[3] R. Lychev, S. Jero, A. Boldyreva, and C. Nita-Rotaru, "How Secure and Quick Is QUIC? Provable Security and Performance Analyses," in *Proceedings of the IEEE Symposium on Security and Privacy*, 2015, pp. 214–231.

[4] S. Chen, T. F. J. M. Pasquier, J. Singh, D. Eyers, and J. Bacon, "Secure communication channel establishment: TLS 1.3 (over TCP Fast Open) vs. QUIC," in *Computer Security – ESORICS 2019*, 2019, pp. 404–426.

[5] A. Yu and T. A. Benson, "Dissecting performance of production QUIC," in *Proceedings of the Web Conference 2021*, 2021, pp. 1157–1168.

[6] P. Biswal and O. Gnawali, "Does QUIC make the web faster?" in *IEEE Global Communications Conference*, 2016, pp. 1–6.

Xu, P. University of Southampton, Southampton, UK

Manuscript Published; 30 June 2025.