Computer Networks 289 (2026) 112722

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Last-mile queueing and HTTP/3 performance: A large-scale empirical evaluation on real access networks

Astrit Krasniqi *, Shahram Salekzamankhani, Bal Virdee

London Metropolitan University, UK

ARTICLE INFO

Keywords:
HTTP/3
QUIC
HTTP/2
RTT
Packet loss
Uplink contention
TCP/UDP unfairness
SOHO networks

ABSTRACT

This paper presents a large-scale empirical evaluation of HTTP/3 (QUIC) and HTTP/2 performance in real last-mile access networks. Unlike prior work that relies heavily on cloud-hosted virtual machines or simulated testbeds, our measurements were collected using physical routers, Ethernet and Wi-Fi links, and consumer-grade access equipment representative of typical small office/home office (SOHO) deployments. More than half a million end-to-end RTT samples were gathered under baseline conditions, controlled packet loss, uplink saturation, and combined delay-plus-loss scenarios.

Across both wired and wireless paths, HTTP/3 and HTTP/2 exhibit comparable median latencies. However, their tail behaviour diverges significantly when the access link is congested. Under a competing TCP upload, HTTP/3 experiences elevated 95th–99th percentile delays and occasional timeouts, consistent with mixed TCP/UDP queueing unfairness at the consumer access link. Consumer NAT and state-handling behaviour may contribute further pressure, but were not directly instrumented in this study. When the competing flow also uses QUIC, HTTP/3 tail performance stabilises and can match or exceed HTTP/2.

These results show that HTTP/3 performance is shaped more by last-mile queueing behaviour than by the transport protocol itself. In typical SOHO environments, switching between HTTP/2 and HTTP/3 does not guarantee lower latency; instead, performance depends primarily on access-link buffering, uplink load, and queue-management discipline. This study highlights the need to evaluate transport innovations in realistic access-network conditions to understand their practical behaviour.

1. Introduction

HTTP/3, built on top of QUIC over UDP, represents a major shift in the transport layer of the web. It aims to reduce head-of-line blocking, improve connection migration, and offer robust performance under unreliable or mobile network conditions. Adoption has accelerated among major browsers and content providers, yet the behaviour of HTTP/3 on the last mile of the Internet remains insufficiently understood. Much of the existing literature evaluates QUIC using cloud-hosted virtual machines, backbone links, or controlled emulation environments [1,2]. While these studies provide valuable insights into protocol mechanics and high-level trends, they do not reflect the constraints of home and small office networks where most end users actually experience the Internet.

In practice, end-user performance is shaped primarily by the characteristics of the access link: consumer Wi-Fi, DSL or cable uplinks, NAT devices, and shallow or unmanaged queues. These environments frequently involve mixed TCP and UDP traffic, competing uploads, bufferbloat, and inconsistent queue-management policies. Such factors can outweigh protocol-level improvements, yet they are rarely captured in cloud-based experiments. Prior work shows that Wi-Fi aggregation, buffer policies, and contention behaviours can significantly distort transport-layer performance [3,4]. However, there is limited empirical evidence on how these effects influence HTTP/3 and HTTP/2 in realistic SOHO networks.

This paper addresses this gap by presenting a large-scale empirical study of HTTP/3 and HTTP/2 performance on real access networks. Using physical routers, Ethernet and Wi-Fi links, and consumer-grade SOHO equipment, we collect more than 500,000 end-to-end RTT samples across a wide range of scenarios, including baseline operation, controlled packet loss, uplink contention, and combined delay-plus-loss impairments. Unlike virtualised or cloud-hosted testbeds, this setup reveals interactions between small-packet UDP traffic, TCP uploads, consumer-router state handling, and the queueing behaviour of consumer routers.

Our findings show that HTTP/3 and HTTP/2 exhibit broadly similar median latency across wired and wireless paths, but their tail behaviour

* Corresponding author.
E-mail address: a.krasniqi@londonmet.ac.uk (A. Krasniqi).

https://doi.org/10.1016/j.comnet.2026.112722
Received 1 December 2025; Received in revised form 11 June 2026; Accepted 2 August 2026
Available online 4 September 2026

diverges significantly under uplink saturation. When a competing TCP upload saturates the access link, HTTP/3 experiences elevated 95th–99th percentile delays and occasional timeouts, consistent with queueing unfairness under mixed TCP/UDP contention rather than with an intrinsic weakness of QUIC. In contrast, when competing flows are also QUIC-based, HTTP/3 tail latency stabilises and can match or exceed that of HTTP/2. These results indicate that the practical performance of HTTP/3 is governed largely by last-mile queueing rather than by the transport protocol alone.

The remainder of this paper describes the measurement environment and methodology, presents results for each scenario, compares them with prior work, and discusses their implications for protocol deployment and last-mile network design.

Contributions

This study makes the following key contributions:

1. **Large-scale empirical dataset on real access networks:** We collect over half a million end-to-end RTT measurements using physical routers, Ethernet and Wi-Fi links, and consumer-grade SOHO equipment, rather than virtual machines or simulated cloud testbeds commonly used in prior QUIC studies.
2. **Systematic comparison of HTTP/3 and HTTP/2 under realistic last-mile conditions:** We evaluate both protocols across baseline operation, controlled packet loss, uplink saturation, and combined delay-plus-loss scenarios, capturing behaviours unique to real access-network queues and consumer-edge devices.
3. **New evidence of tail-latency divergence under uplink contention:** We show that when a competing TCP upload saturates the access link, HTTP/3 experiences elevated 95th–99th percentile delays and occasional timeouts under mixed TCP/UDP contention, while median performance remains similar to HTTP/2.
4. **Analysis of fairness dynamics in mixed-protocol contention:** Our results show that HTTP/3's degradation is not intrinsic to QUIC but emerges under access-link contention patterns that disadvantage short foreground QUIC bursts. When the competing flow is also QUIC-based, HTTP/3's tail latency stabilises and can match or exceed HTTP/2.
5. **Practical implications for protocol deployment and access-link configuration:** We show that HTTP/3's real-world latency is shaped more by last-mile queueing behaviour than protocol design alone. Access-link buffering, uplink load, and queue-management discipline have a greater impact on performance than switching between HTTP/2 and HTTP/3.

These contributions clarify how access-network behaviour interacts with modern web transports and establish a reproducible, hardware-based baseline for future QUIC and HTTP/3 performance research in everyday network environments.

The paper proceeds by outlining the measurement environment and methodology, presenting results for each scenario, analysing how they align with prior studies, and translating them into practical recommendations.

2. Background and related work

HTTP/3, built on top of QUIC over UDP, introduces several transport-layer improvements including reduced head-of-line blocking, integrated encryption, and support for connection migration. These design choices aim to provide better performance on paths with loss, mobility, or reordering. Adoption has grown across major browsers and content providers, but the behaviour of HTTP/3 on the last mile of the Internet is still not well understood.

A substantial portion of existing work evaluates HTTP/3 using virtual machines, backbone links, or controlled laboratory setups. Perna et al. [1] provide a large-scale characterisation of HTTP/3 adoption and performance, but their evaluation relies on cloud-based measurements that do not capture Wi-Fi contention or consumer router behaviour. Gupta and Bartos [2] study HTTP/3 in real deployment scenarios, with a focus on user experience and page-load times; however, their work does not examine transport-layer effects under uplink competition or NAT bottlenecks typical of SOHO environments.

Other studies investigate protocol-level mechanisms. Sander and Kunze [5] analyse HTTP/3 resource prioritisation and identify scenarios where application-layer scheduling can mitigate or expose head-of-line blocking. These works deepen understanding of QUIC internals but typically operate on well-provisioned wired links that lack the queueing dynamics of consumer access networks.

A growing body of literature highlights that Wi-Fi behaviour and access-link buffering often dominate transport-layer performance. Grazia et al. [3] show that TCP performance on Wi-Fi can degrade substantially due to MAC-layer aggregation and buffer interaction, revealing fairness issues that persist across protocol versions. Similarly, Du et al. [4] revisit congestion control for Wi-Fi and demonstrate that MAC-layer effects frequently outweigh protocol-level improvements. These findings suggest that evaluations of HTTP/3 confined to cloud or backbone paths overlook key bottlenecks that affect real users.

Recent studies also explore HTTP/3 suitability in CDN and high-performance environments. Zhou et al. [6] dissect HTTP/3 applicability for content delivery networks, emphasising that performance gains are highly dependent on network conditions and CDN structure. Zhang et al. [7] show that QUIC can provide limited benefit on fast and uncongested networks, further reinforcing that protocol advantages are sensitive to path characteristics rather than guaranteed by protocol design alone.

Additionally, more general research in constrained wireless and heterogeneous edge environments supports the idea that contention structure, queue discipline, and communication asymmetries often dominate protocol-level outcomes. For example, Sun et al. [8] show that orderly queue access and access coordination can materially influence performance in dense wireless systems, highlighting the importance of how access opportunities are scheduled under contention. Similarly, Jiang et al. [9] show that pre-deployment performance in SDN-enabled networks can depend strongly on topology, configuration, and predicted network conditions, while Liu et al. [10] demonstrate in heterogeneous edge computing that communication imbalance and constrained transmission opportunities can dominate overall system behaviour even when the higher-level application logic remains unchanged. Although these studies do not evaluate HTTP/3 directly, they reinforce a broader principle relevant to this paper: in constrained access environments, realised performance is often shaped less by protocol design in isolation and more by queueing, contention, and access-network behaviour.

More recent work in data-centre and programmable-network environments further shows that queueing, load balancing, packet ordering, and measurement resource allocation remain central to network performance. Hu et al. [11] show that packet reordering in lossless data-centre networks can substantially harm flow completion time and propose in-network recirculation to reduce this effect. In related work, Hu et al. [12] demonstrate that load-balancing decisions require multiple congestion signals because individual signals such as queue length, RTT, or link utilisation may not capture transient queueing behaviour accurately. Hu et al. [13] further show that adaptive memory allocation can improve priority-aware traffic measurement under changing flow distributions. Although these studies focus on data-centre and measurement systems rather than HTTP/3 in SOHO access networks, they support the broader principle that realised protocol performance is strongly shaped by queueing, contention, packet scheduling, and resource-management behaviour.

Despite these contributions, there remains a clear gap in understanding HTTP/3 behaviour on real SOHO access networks. To the best of our knowledge, there is still limited empirical evidence from

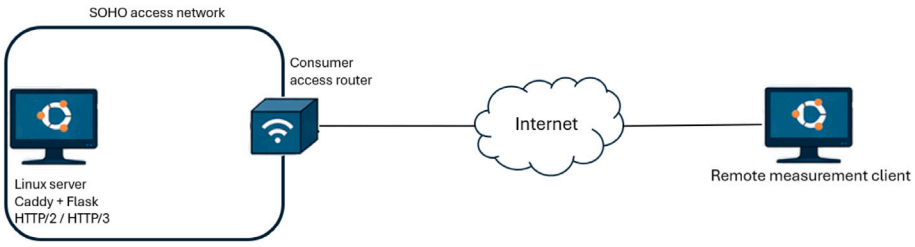

Fig. 1. Physical testbed used for the measurements. A Linux server running Caddy and Flask with HTTP/2 and HTTP/3 support was deployed behind a consumer NAT router with TCP/443 and UDP/443 forwarded to the server. Remote measurement traffic reached the server over the Internet, and the same topology was used across all reported scenarios. Across the study, measurements were repeated under both wired Ethernet and Wi-Fi conditions.

large-scale, hardware-based comparisons of HTTP/3 and HTTP/2 under mixed TCP and UDP contention, uplink saturation, combined delay-plus-loss impairment, and consumer-router queue dynamics. This paper addresses that gap by evaluating more than 500,000 end-to-end RTT samples across wired and wireless access links, revealing performance behaviours that are not observable in virtualised or datacentre measurements.

Our study builds on this insight by presenting what we believe is one of the first large-scale, hardware-based measurements of HTTP/3 and HTTP/2 performance across more than 500,000 RTT samples, using physical routers, uplink contention, and real wireless links. The aim is not to propose a new protocol or algorithm, but to surface and explain the often-overlooked access-network effects that shape HTTP/3 behaviour under pressure.

3. Methodology

This section describes the test environment, hardware setup, software stack, impairment conditions, and measurement process used to evaluate HTTP/3 and HTTP/2 performance on real access networks. The goal is not only to compare protocol behaviour, but to do so under realistic, reproducible last-mile conditions representative of SOHO deployments.

3.1. Testbed overview

All experiments were conducted using physical hardware connected through consumer-grade access equipment. The core testbed consists of: (1) a Linux-based HTTP server running a QUIC-capable web stack (Caddy and Flask), (2) a measurement client connected over Ethernet or Wi-Fi, (3) a consumer router responsible for NAT, queueing, and uplink buffering, and (4) an external network path between server and client traversing ISP infrastructure.

Unlike cloud-hosted experiments, this configuration naturally captures queueing behaviour, routing asymmetry, Wi-Fi variability, and uplink contention. These factors are central to the research question, as they dominate end-user latency and are not accurately represented in virtualised environments (see Fig. 1).

3.2. Server configuration

The server was deployed on a Linux host and exposed through a Caddy front-end configured to negotiate HTTP/2 or HTTP/3 automatically using standard ALPN-based TLS negotiation. Behind Caddy, a lightweight Flask application provided a small set of measurement endpoints used consistently across all experiments. These included `/health` for RTT probing, `/upload` for upload tests, and `/download/<filename>` for streaming download tests.

The `/health` endpoint was used for repeated latency probing because it returned a lightweight response with minimal application processing overhead. In the implementation used for these measurements, the endpoint introduced a small synthetic server-side delay of a few milliseconds in order to avoid benchmarking a completely empty handler. This behaviour was kept identical regardless of whether the request was served over HTTP/2 or HTTP/3, so protocol comparisons remained like-for-like.

The upload and download endpoints were implemented in a simple and transparent way. Upload requests stored files on the server and logged transfer metadata, while download responses were streamed in chunks rather than returned as a single buffered object. No protocol-specific optimisation, kernel bypass, or custom transport modification was introduced. The server therefore reflects a realistic deployment that combines standard web software with a QUIC-capable reverse proxy.

It is important to clarify that this paper does not propose a new transport protocol, congestion-control algorithm, or measurement framework. Instead, it uses established and openly available software components to build a reproducible real-hardware testbed, with the research contribution lying in the empirical evaluation of HTTP/2 and HTTP/3 under realistic last-mile conditions.

3.3. Client configuration

The remote measurement client was connected over wired Ethernet. The wired and Wi-Fi conditions reported in this paper were created at the server-side access network by enabling either the server's wired Ethernet path or its Wi-Fi path, while the other was disabled.

HTTP/2 measurements were conducted using the `httpx` library with HTTP/2 enabled. HTTP/3 measurements used an `aioquic`-based client that issued individual requests over a persistent QUIC connection and recorded:

- application-layer RTT,
- QUIC smoothed RTT when available,
- jitter (sample-to-sample variation).

This approach avoids browser noise while preserving standard HTTP semantics.

3.4. Impairment scenarios

To evaluate protocol behaviour under diverse access conditions, four representative scenarios were applied:

1. **Baseline:** No artificial impairment was added, allowing measurements to capture normal SOHO behaviour over the access path.
2. **Packet Loss:** Controlled random packet loss was introduced using Linux `tc` in order to emulate access-link variability and transient network impairment.
3. **Uplink Saturation:** A long-lived competing TCP upload flow was generated in parallel with the RTT probes so that the router uplink remained under sustained queue pressure. The purpose of this scenario was not to benchmark bulk upload throughput itself, but to observe how small request–response transactions behaved when sharing the uplink with a persistent TCP transfer.

The competing flow was generated using standard user-space tools such as `curl` and `iperf3`, without custom kernel tuning or protocol modification, so that queue build-up arose under realistic consumer-router conditions.

4. **Delay + Loss Combination:** Fixed delay and random packet loss were applied together in order to evaluate protocol behaviour under more stressful and realistic multi-factor conditions.

The impairment design was chosen to move progressively from clean baseline conditions to increasingly challenging last-mile scenarios. In particular, the uplink-saturation case is important because consumer routers often exhibit buffer build-up, NAT pressure, and unequal treatment of competing traffic when long-lived TCP uploads coexist with smaller latency-sensitive exchanges. The study therefore focuses on realistic queue occupancy effects rather than on artificially isolated transport behaviour alone.

The competing upload flow used the default behaviour of the host TCP stack and was not separately tuned for congestion control, packet sizing, or pacing. This choice was deliberate: the goal of the experiment was to reproduce practical queueing pressure created by ordinary bulk-transfer traffic, rather than to optimise the competing flow itself. In the saturation scenario, the upload was configured to create sustained uplink load approaching the available access capacity, thereby stressing the queueing behaviour of the consumer router.

3.5. Measurement procedure

For each scenario, the client issued repeated lightweight HTTP requests to the `/health` endpoint and recorded one measurement per request in structured CSV output. The primary metric was end-to-end application RTT, defined as the time elapsed between request dispatch and receipt of the first response bytes at the client. In addition, the scripts recorded:

- application-layer RTT,
- jitter, computed as the absolute difference from the previous RTT sample,
- QUIC smoothed RTT for HTTP/3 when available from the transport stack.

To avoid handshake cost dominating the measurements, HTTP/3 probes were sent over a single persistent QUIC connection, while HTTP/2 measurements reused a stable client session over TLS. Probe pacing was kept approximately periodic, with a nominal inter-request interval of 80 ms and a small random perturbation to avoid strict synchronisation effects during long runs.

Timeouts reported in this paper refer to application-level request timeouts in the probing scripts rather than QUIC idle timeout expiry or TCP retransmission timeout events. In both protocol clients, a request was treated as failed if no valid response was obtained within the configured 10 s application-level timeout threshold. This distinction is important because the observed failures represent unsuccessful application transactions under queueing pressure, rather than a direct measurement of lower-layer retransmission timers.

For each run, the first 200 samples were treated as warm-up and removed before plotting and summary analysis, reducing the influence of connection establishment and early transient behaviour. Depending on the scenario, each experiment collected between 10,000 and 50,000 samples, producing more than 500,000 RTT measurements in total across the full study.

3.6. Data processing and reproducibility

All measurement scripts, server configurations, and impairment settings were maintained under version control to support repeatability. The testbed was constructed using openly available software components, including `aioquic` for HTTP/3 probing, `httpx` for HTTP/2 probing, Flask and Caddy on the server side, and Linux `tc` for impairment injection. Because the software stack is based on publicly available tools and commodity hardware, the experimental workflow can be reproduced without proprietary components.

Each RTT probe run wrote structured CSV output containing iteration number, application-layer RTT, jitter, and a grouping index for later aggregation. For HTTP/3, the client additionally recorded QUIC smoothed RTT when available. To reduce warm-up effects, the first 200 samples of each run were discarded prior to plotting and summary analysis. For long runs, later samples were grouped in consecutive blocks of 50 to improve readability while preserving overall time-series behaviour. This grouping was applied consistently across the visualisation workflow.

Probe pacing was approximately periodic but not perfectly fixed. In both HTTP/2 and HTTP/3 scripts, requests were issued with a nominal inter-probe interval of 80 ms, together with a small random perturbation of approximately ±10 ms. This was done to avoid strict synchronisation artefacts while keeping the probing rate stable and low enough not to dominate the access link by itself.

The plotting workflow generated consistent figures directly from the recorded CSV files. The visualisations reported grouped RTT and jitter trends and included reference lines for mean RTT, median RTT, and mean jitter where applicable. For HTTP/3, grouped QUIC smoothed RTT could also be inspected where it was recorded. Output figures were exported automatically, ensuring that the same processing steps were applied across scenarios. In addition to protocol selection, the recorded logs preserved run metadata such as timestamps, impairment settings, and environment labels, allowing each experiment to be traced and repeated (see Fig. 2).

3.7. Justification for using real hardware

Using physical routers and real Wi-Fi is essential for this study. Consumer NAT devices and wireless access points exhibit behaviours such as bursty queueing, per-flow unfairness, inconsistent buffer sizing, MAC-layer aggregation, and vendor-specific timeout or state-handling policies. These behaviours can influence observed HTTP/2 and HTTP/3 performance in ways that are difficult to reproduce faithfully in cloud-based or fully virtualised environments.

In particular, last-mile performance is often shaped not only by the transport protocol itself but also by how consumer hardware manages uplink contention, packet scheduling, wireless retransmissions, and connection state. This is especially relevant when long-lived bulk transfers coexist with smaller latency-sensitive exchanges. A real-hardware methodology is therefore necessary to capture the access-network effects that motivate this study.

4. Results and discussion

This section presents results from four experimental regimes representative of small office/home office (SOHO) environments: baseline RTT and jitter, controlled packet loss, concurrent uplink load, and combined delay plus loss. Each test was performed for both HTTP/2 and HTTP/3 over wired and wireless connections.

4.1. Measuring latency with Round-Trip Time (RTT)

Fig. 3 summarises baseline RTT behaviour for HTTP/2 and HTTP/3 on both wired Ethernet and Wi-Fi. Each run contains 50,000 probes; the first 200 warm-up samples are removed and the remainder is averaged in groups of 50 to expose steady-state behaviour.

Wired. On the wired path the two protocols are closely matched and highly stable. HTTP/2 settles at **7.96 ms mean/7.90 ms median** with a narrow **0.52 ms interquartile range**. HTTP/3 is slightly lower and a touch tighter—**7.04 ms mean/6.99 ms median**, **0.51 ms IQR**. Both traces remain close to their central values with only occasional narrow spikes. On a clean link, transport choice does not materially change baseline RTT; HTTP/3's advantage is modest.

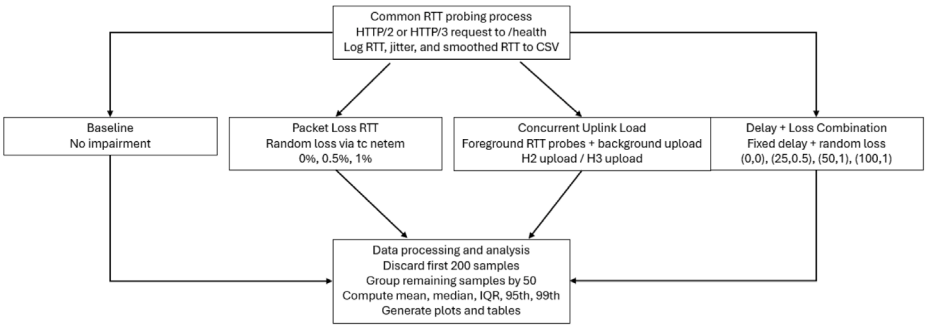

Fig. 2. Experimental workflow used across all reported scenarios. A common RTT probing process was applied using HTTP/2 or HTTP/3 requests to the `/health` endpoint, with RTT, jitter, and QUIC smoothed RTT logged to CSV where available. The same probing workflow was then executed under four conditions: baseline, packet-loss RTT, concurrent uplink load, and combined delay-plus-loss impairment. Recorded measurements were post-processed by discarding the first 200 samples, grouping the remainder in blocks of 50, and computing the summary statistics and plots reported in the paper.

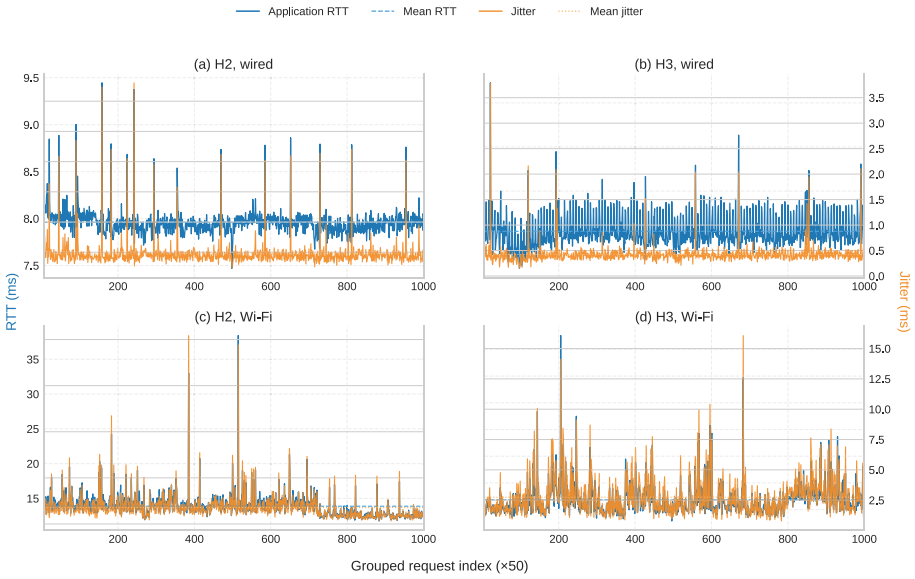

Fig. 3. Baseline RTT for HTTP/2 and HTTP/3 over wired Ethernet and Wi-Fi (50,000 probes per panel; first 200 samples trimmed; grouped ×50). Blue: application RTT. Orange (right axis): jitter. Dashed lines indicate mean values. The figure highlights temporal stability and burst behaviour, while exact summary statistics are given in the corresponding table. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Wireless. Over Wi-Fi, the medium dominates. HTTP/2 rises to **13.87 ms mean/11.97 ms median** with an **1.74 ms IQR** and frequent short bursts. HTTP/3 records **12.74 ms mean/11.46 ms median** and **1.47 ms IQR**, with the right tail extending to brief peaks around 18–33 ms. The spread is several-fold greater than on Ethernet, reflecting contention and MAC-layer retransmissions rather than transport design. Any small protocol advantage is largely masked by the wireless channel.

Interpretation. Baseline latency is governed primarily by the access medium: Ethernet yields tight, low-jitter traces for both protocols, whereas Wi-Fi inflates spread and introduces frequent short spikes. With no additional impairment, HTTP/2 and HTTP/3 deliver comparable RTTs; HTTP/3 is modestly lower and slightly more compact, but the difference is small compared with the variability introduced by the wireless medium. This baseline frames the loss and load experiments that follow.

Quantitative summary. To complement Fig. 3, Table 1 summarises central tendency and spread for each access medium. HTTP/3 consistently shows a slightly lower median and tighter IQR than HTTP/2. On wired Ethernet the gap is about one millisecond, while on Wi-Fi both protocols experience higher variance driven by contention and radio retries. The long right tails (95th–99th percentiles) on Wi-Fi represent short retransmission bursts rather than persistent congestion.

Table 1
Baseline RTT results (50 000 probes per condition).

Medium	Protocol	Mean [ms]	Median [ms]	IQR [ms]	95th [ms]	99th [ms]
Wired	H2	7.96	7.90	0.52	8.58	9.49
	H3	7.04	6.99	0.51	7.73	8.49
Wi-Fi	H2	13.87	11.97	1.74	22.04	39.97
	H3	12.74	11.46	1.47	18.24	32.70

4.2. Packet-loss RTT (0%–1%)

This section examines how small amounts of random loss change application-level RTT for HTTP/2 and HTTP/3. We apply 0%, 0.5%, and 1% random loss with `tc netem` on the client side. Each condition is run for 50,000 probes per protocol and access medium (wired and Wi-Fi), with the first 200 samples trimmed and the remainder grouped in blocks of 50, as in Section 4.1.

Wired. Adding 0.5% loss raises the centre of both protocols but in different ways. HTTP/2 tends to lift the whole distribution smoothly

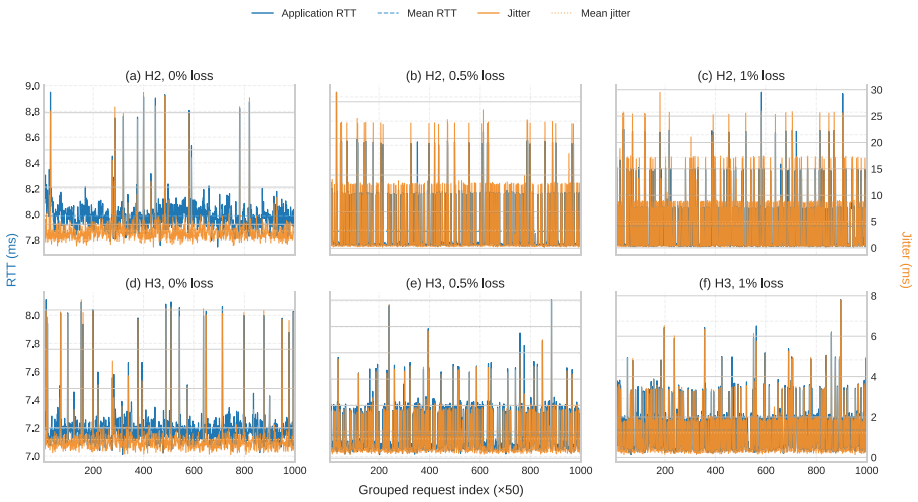

Fig. 4. Packet-loss RTT on **wired Ethernet** for HTTP/2 (top row) and HTTP/3 (bottom row) at 0%, 0.5%, and 1% random loss (50,000 probes per panel; first 200 samples trimmed; grouped ×50). Blue: application RTT. Orange (right axis): jitter. Dashed lines show mean values. Exact summary statistics are reported in the corresponding table; the plots are intended to show temporal variation and burst behaviour. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 2
Wired packet-loss RTT results (50 000 probes per condition).

Loss	Protocol	Mean [ms]	Median [ms]	IQR [ms]	95th [ms]	99th [ms]
0%	H2	7.99	7.93	0.34	8.58	9.63
	H3	7.20	7.12	0.27	7.67	8.77
0.5%	H2	8.87	7.74	0.33	8.46	9.91
	H3	7.91	7.63	0.26	8.27	9.62
1%	H2	9.86	7.67	0.33	8.50	49.93
	H3	8.19	7.66	0.27	8.59	45.58

Table 3
Wi-Fi packet-loss RTT results (50 000 probes per condition).

Loss	Protocol	Mean [ms]	Median [ms]	IQR [ms]	95th [ms]	99th [ms]
0%	H2	13.44	12.53	2.07	17.45	23.81
	H3	12.31	11.20	1.16	16.96	28.43
0.5%	H2	15.04	12.75	2.22	18.85	36.87
	H3	12.75	11.39	1.20	16.43	52.02
1%	H2	14.44	11.56	0.95	16.05	222.81
	H3	13.85	11.43	2.06	18.53	74.55

(a mostly constant offset with moderate broadening), while HTTP/3 keeps a lower central RTT at the same loss level yet shows occasional short bursts linked to recovery. At 1% loss, both degrade further: HTTP/2 remains "smoother but higher", whereas HTTP/3 stays lower in the typical case with rarer, taller excursions. For steady workloads that value predictability, HTTP/2 is slightly more even; for interactive workloads that benefit from low typical latency, HTTP/3 keeps the centre down.

Wireless. On Wi-Fi the loss signal compounds radio contention and MAC retries. At 0.5% loss, HTTP/3 still tends to hold a lower median than HTTP/2, but the gap narrows and both protocols display more frequent spikes. By 1%, the access medium dominates variance; medians and jitter move closer and short bursts are common in both traces. Protocol choice matters less than stabilising the radio channel.

Stability and tails. Across media, the coefficient of variation (σ/μ) grows with loss. On wired links, HTTP/3 often maintains a slightly lower centre, while tail behaviour varies by impairment level rather than following one uniform pattern. On Wi-Fi, σ/μ rises for both protocols and access-medium dynamics increasingly dominate variability, narrowing the practical importance of protocol-level differences (see Figs. 4–5).

Wired (quantitative results). Table 2 summarises wired results for 0%, 0.5%, and 1% random loss. HTTP/3 keeps a slightly lower median and narrower IQR across all loss levels, while HTTP/2 develops much heavier tails as retransmissions accumulate. Median RTTs remain below 8 ms for both protocols across all wired loss settings, but the 99th percentile of H2 expands to nearly 50 ms at 1% loss, whereas H3 remains slightly tighter.

Wireless (quantitative results). Table 3 reports Wi-Fi results under 0%, 0.5%, and 1% random loss. HTTP/3 generally maintains a lower centre than HTTP/2, although spread and tail behaviour vary across settings rather than favouring one protocol uniformly in every case. While both protocols are affected by link-layer contention and retries, HTTP/2 develops especially heavy tails at 1% loss, with its 99th percentile rising above 220 ms, whereas HTTP/3 remains below 75 ms. Protocol differences therefore persist, but radio variability dominates overall dispersion.

4.3. Concurrent uplink load

We now examine how a sustained background upload on the access uplink reshapes RTT for foreground probes. Two variants are used: (i) a long-lived HTTP/2 upload and (ii) a long-lived HTTP/3 upload, both targeted at the same server and run long enough to keep the uplink queue persistently busy. Foreground probes continue to query the `/health` endpoint over HTTP/2 or HTTP/3, as in prior sections. Each condition uses 10,000 probes per panel, with warm-up trimmed and grouping ×50. As defined in Section 3.5, any timeout mentioned in this section refers to an application-level request timeout in the probing client rather than a directly observed transport-layer retransmission timer.

Wired. When the background is a long-lived **HTTP/2 upload**, the foreground traces become more bursty, with the HTTP/3 probe showing greater short-term variability and occasional application-level timeouts. Importantly, this does not appear mainly as a shift in the median; rather, the centre remains relatively close to baseline while the spread increases. HTTP/2 probes are also affected, but their degradation is smoother. When the background is a **HTTP/3 upload**, contention still inflates latency for both probe types, yet the effect is milder and more symmetric than under HTTP/2 upload.

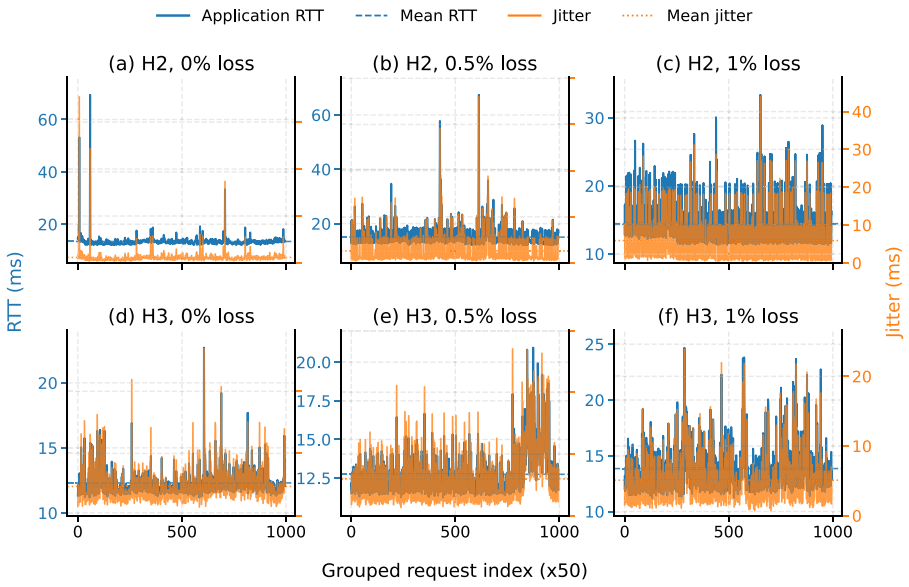

Fig. 5. Packet-loss RTT on **Wi-Fi** for HTTP/2 (top row) and HTTP/3 (bottom row) at 0%, 0.5%, and 1% random loss (50,000 probes per panel; first 200 samples trimmed; grouped ×50). Link-layer contention amplifies variability and narrows protocol differences relative to wired. Exact summary statistics are reported in the corresponding table; the plots are intended to show temporal variation and burst behaviour.

Table 4
Wired concurrent uplink load results (10 000 probes per condition).

Probe	Background upload	Mean [ms]	Median [ms]	IQR [ms]	95th [ms]	99th [ms]
H2	H2 upload	8.19	8.05	0.55	9.11	10.26
H2	H3 upload	7.97	7.86	0.40	8.67	9.89
H3	H2 upload	7.41	7.24	0.78	8.71	9.72
H3	H3 upload	7.53	7.44	0.48	8.33	9.53

Wi-Fi. The same pattern appears more strongly on Wi-Fi, but here both foreground probe types are heavily affected when the background flow is a long-lived **HTTP/2 upload**. The dominant effect is not simply a higher average RTT, but severe burstiness and tail inflation caused by queue build-up together with wireless airtime contention and MAC-layer retransmissions. When the background upload instead uses **HTTP/3**, both probe types return close to baseline behaviour, and the sharp instability seen under HTTP/2 upload is largely absent.

Interpretation. These busy-uplink results indicate that foreground latency under load is shaped primarily by access-link queueing and mixed-traffic interaction at the SOHO edge rather than by an intrinsic weakness of QUIC. A plausible explanation is that a long-lived TCP upload tends to occupy the bottleneck queue more aggressively, while smaller foreground transactions, especially short QUIC bursts, become more exposed to delay variation and occasional application-level failure. Consumer NAT and state-handling behaviour may contribute further pressure, but this study does not directly instrument NAT binding timers or mapping policies, so NAT should be treated here as a contributing factor rather than as the sole proven cause (see Figs. 6–7).

Wired (quantitative results). Table 4 summarises RTT behaviour when a sustained background upload keeps the SOHO uplink busy. Under a background HTTP/2 upload, the HTTP/3 probe retains a lower median RTT than the HTTP/2 probe (7.24 ms versus 8.05 ms), but its spread is larger (IQR 0.78 ms versus 0.55 ms), indicating increased burstiness rather than a worse centre. When the background upload instead uses HTTP/3, both medians and tails move closer to baseline and to each other. This suggests that the main penalty in the wired busy-uplink case is not congestion alone, but mixed TCP/UDP contention at the bottleneck queue.

Wireless (quantitative results). Table 5 extends the busy-uplink experiment to Wi-Fi. When a background HTTP/2 upload is active, both probe types show severe inflation in spread: the H2 probe reaches a mean of 38.53 ms, median of 15.95 ms, and IQR of 46.92 ms, while the H3 probe reaches a mean of 31.67 ms, median of 11.53 ms, and IQR of 37.70 ms. The fact that medians remain far below the means indicates intermittent queue blow-ups and bursty service rather than a uniform shift upward. When the background flow uses HTTP/3 instead, both probe types return close to baseline medians and sub-millisecond IQRs. This supports the interpretation that the most damaging condition in these experiments is mixed contention between a long-lived TCP upload and foreground probe traffic on a stressed Wi-Fi uplink.

4.4. Delay + loss combination

We combine a fixed base delay with small random loss to examine how recovery cost interacts with queueing and the access medium. Each panel uses 10,000 probes, trims the first 200 samples, and groups the remainder in blocks of 50, as in earlier sections. We show four representative settings: ($D = 0$ ms, $p = 0\%$), (25 ms, 0.5%), (50 ms, 1%), and (100 ms, 1%).

Wired. With fixed delay applied, both protocols shift upward as expected, but their recovery behaviour differs once loss is added. Across the tested settings, HTTP/3 generally preserves a slightly lower median RTT than HTTP/2. At the stronger delay-plus-loss settings, however, the largest extreme tail inflation appears in HTTP/2, whose 99th-percentile values rise much more sharply.

Wi-Fi. On Wi-Fi, fixed delay compounds radio contention and MAC retries. Both protocols shift upward and show more short bursts, but the access medium narrows practical differences relative to wired Ethernet. HTTP/3 still tends to keep a slightly lower centre, while overall variability is increasingly dominated by the wireless channel.

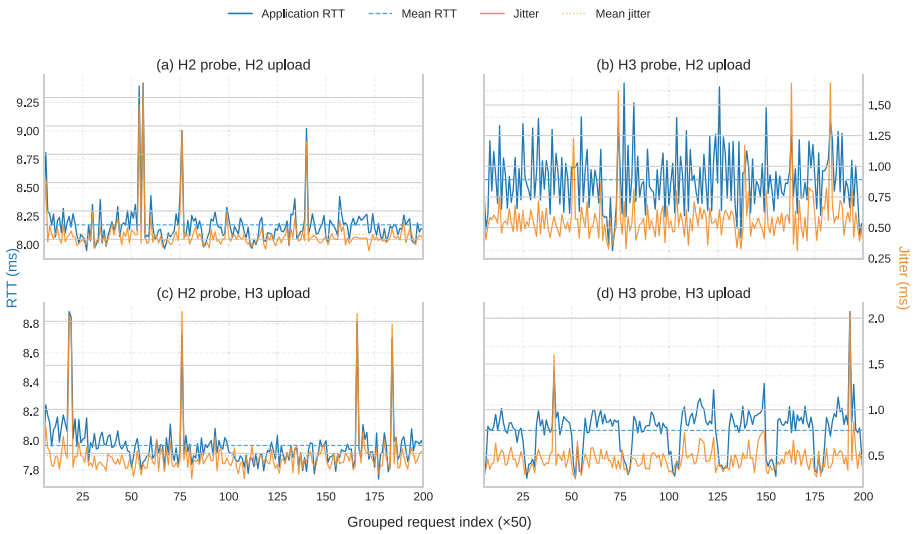

Fig. 6. Concurrent uplink load on **wired Ethernet**. Each panel shows 10,000 probes (first 200 trimmed; grouped ×50). Top row: background HTTP/2 upload. Bottom row: background HTTP/3 upload. Blue: application RTT. Orange (right axis): jitter. Dashed lines show mean values. Under background HTTP/2 upload, the foreground traces become more bursty, with the HTTP/3 probe showing greater short-term variability and occasional application-level timeouts. Under background HTTP/3 upload, degradation is milder and more symmetric. Exact summary statistics are reported in the corresponding table; the plots are intended to show temporal variation and burst behaviour. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

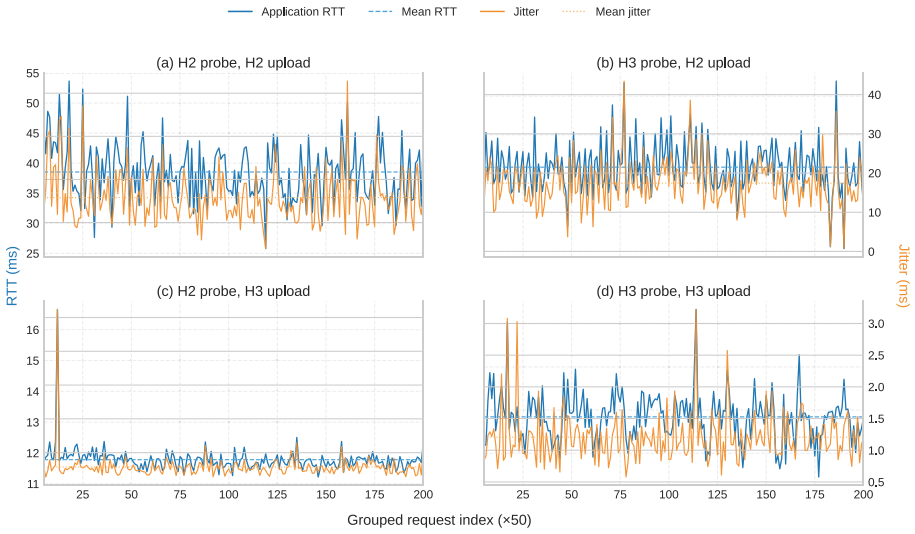

Fig. 7. Concurrent uplink load on **Wi-Fi**. Each panel shows 10,000 probes (first 200 trimmed; grouped ×50). A background HTTP/2 upload produces severe burstiness and tail inflation for both probe types, whereas a background HTTP/3 upload leaves both traces close to baseline. The contrast suggests that mixed-protocol contention, rather than background load alone, dominates instability on the stressed Wi-Fi uplink. Exact summary statistics are reported in the corresponding table; the plots are intended to show temporal variation and burst behaviour.

Table 5
Wi-Fi concurrent uplink load results (10 000 probes per condition).

Probe	Background upload	Mean [ms]	Median [ms]	IQR [ms]	95th [ms]	99th [ms]
H2	H2 upload	38.53	15.95	46.92	110.84	157.02
H2	H3 upload	11.78	11.47	0.53	14.62	16.95
H3	H2 upload	31.67	11.53	37.70	97.63	141.65
H3	H3 upload	11.42	11.11	0.64	14.50	16.67

Summary. Base delay amplifies the impact of loss. On wired links, HTTP/3 usually retains a slightly lower centre, while HTTP/2 exhibits the heavier extreme tails at higher impairment levels. On Wi-Fi, medium dynamics dominate and narrow protocol differences, although HTTP/3 still keeps a modestly lower centre in most settings (see Figs. 8–11).

4.5. Cross-scenario comparison

Taken together, the four regimes show that HTTP/2 and HTTP/3 remain close in median RTT across realistic SOHO conditions, while differing more noticeably in variability and tail behaviour. Table 7 summarises representative *wired* median RTTs from the main scenarios

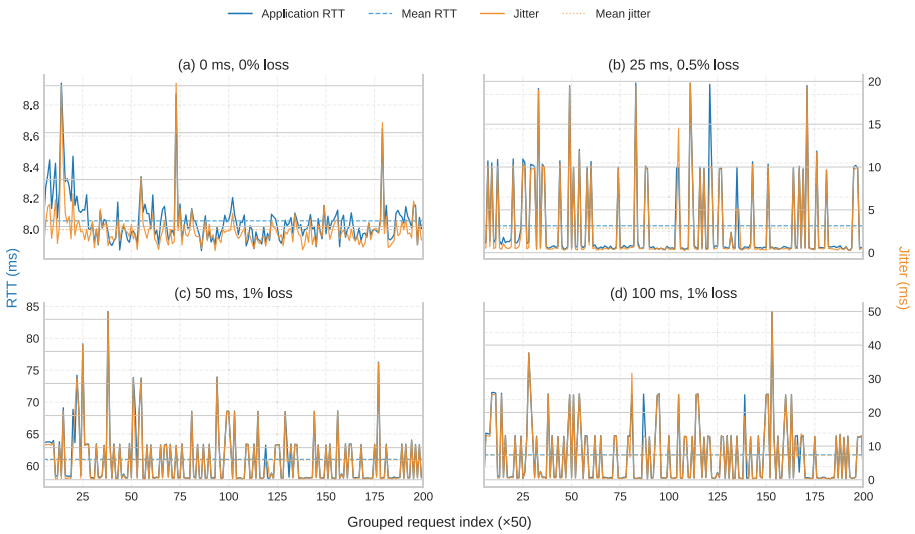

Fig. 8. Delay + loss on wired Ethernet for HTTP/2. Each panel shows 10 000 probes (first 200 trimmed; grouped ×50). Blue: application RTT. Orange (right axis): jitter. Dashed lines show mean values. Exact summary statistics for all delay-plus-loss panels are reported in Table 6; the plots are intended to show temporal variation and burst behaviour. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

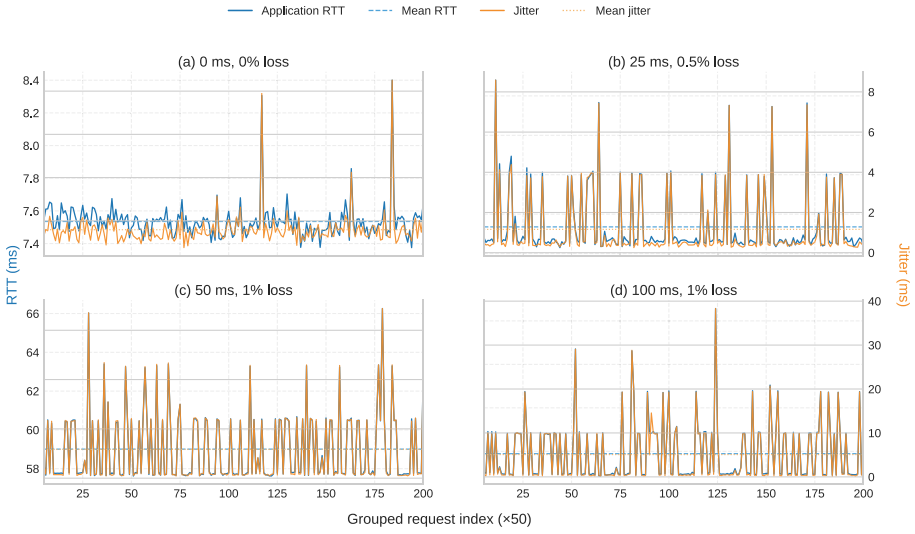

Fig. 9. Delay + loss on wired Ethernet for HTTP/3. Settings and data volume as in Fig. 8. HTTP/3 generally keeps a slightly lower typical RTT than HTTP/2 at the same settings, while the strongest extreme tail inflation is more evident in HTTP/2.

Table 6
Delay + loss combination results (10 000 probes per condition).

Medium	Protocol	Setting (D, p)	Mean [ms]	Median [ms]	IQR [ms]	95th [ms]	99th [ms]
Wired	H2	0 ms, 0%	8.05	7.94	0.40	8.88	10.10
	H2	25 ms, 0.5%	34.33	33.01	0.44	34.05	36.12
	H2	50 ms, 1%	61.03	58.09	0.41	59.16	315.39
	H2	100 ms, 1%	111.76	108.25	0.42	109.58	415.70
	H3	0 ms, 0%	7.54	7.44	0.32	8.13	9.31
	H3	25 ms, 0.5%	32.68	32.23	0.37	33.05	34.46
	H3	50 ms, 1%	59.01	57.67	0.35	58.53	62.55
	H3	100 ms, 1%	109.73	107.30	0.33	108.44	116.64
Wi-Fi	H2	0 ms, 0%	11.97	11.59	0.55	14.86	16.86
	H2	25 ms, 0.5%	38.15	36.75	0.56	40.17	42.49
	H2	50 ms, 1%	65.18	61.86	0.51	65.33	323.24
	H2	100 ms, 1%	115.14	112.04	0.63	116.69	129.07
	H3	0 ms, 0%	11.52	11.13	0.47	14.62	16.96
	H3	25 ms, 0.5%	37.25	36.30	0.49	39.81	42.15
	H3	50 ms, 1%	63.20	61.40	0.50	64.87	86.24
	H3	100 ms, 1%	113.93	111.04	0.49	115.15	143.27

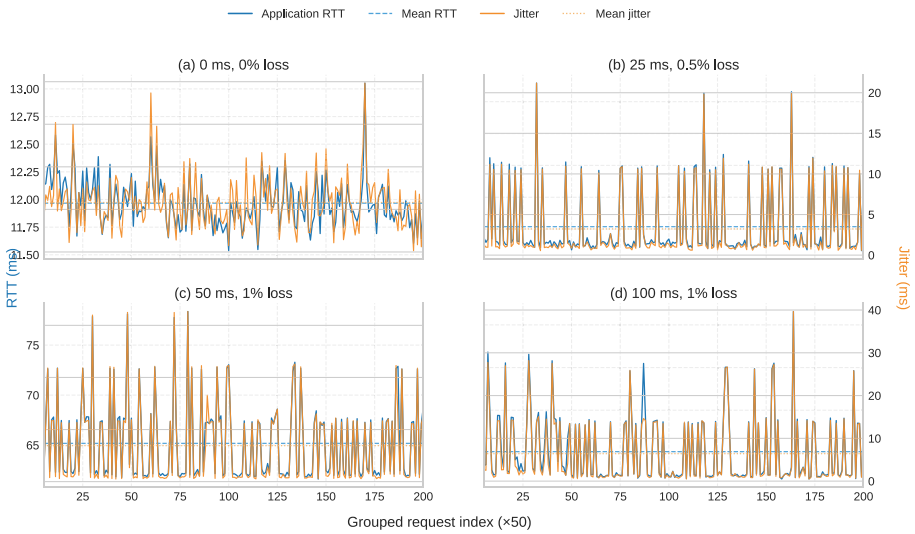

Fig. 10. Delay + loss on Wi-Fi for HTTP/2. Fixed delay stacks with radio contention and MAC retries; medians rise and short spikes are frequent across settings.

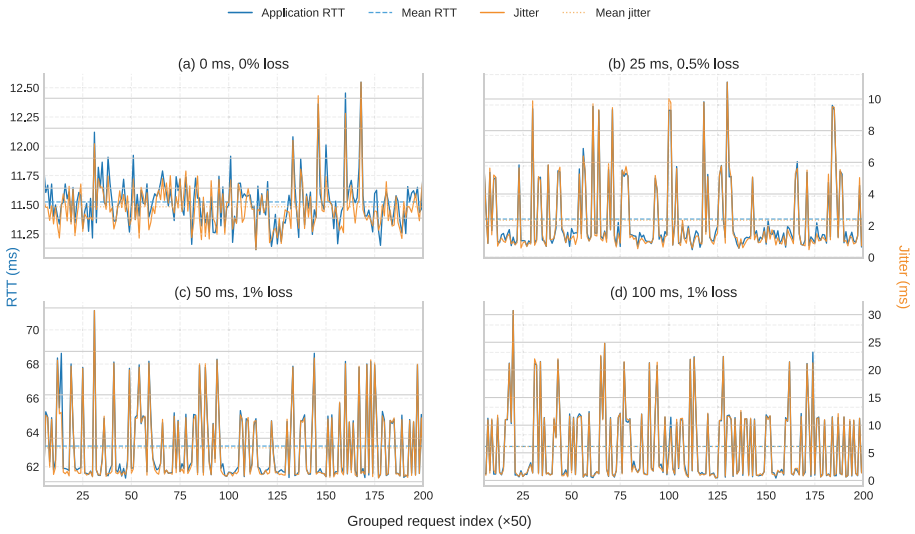

Fig. 11. Delay + loss on Wi-Fi for HTTP/3. The wireless medium narrows protocol differences relative to Ethernet; HTTP/3 generally keeps a slightly lower centre than HTTP/2 at the same settings, while overall variability is increasingly dominated by the wireless channel.

in order to compare the centre of each distribution under clean conditions, random loss, sustained uplink contention, and combined delay plus loss.

Across all four scenarios, HTTP/3 delivers either a slightly lower median RTT than HTTP/2 or an effectively equal one. The absolute difference is small in clean baseline conditions (0.91 ms) and remains small under packet loss and under combined delay plus loss. This indicates that, for the median case, transport selection alone does not radically change application RTT in the tested SOHO environment.

The main divergence appears under sustained uplink load. When a long-lived TCP upload keeps the uplink busy, median RTTs for HTTP/2 and HTTP/3 still remain relatively close, but the higher-percentile delays and short-term variability become much worse, especially for foreground HTTP/3 probes. By contrast, when the background upload also uses HTTP/3, the spread contracts and behaviour returns much closer to baseline. This pattern suggests that the most damaging condition is not load alone, but mixed contention between a long-lived TCP upload and foreground probe traffic at the access bottleneck.

Under random loss and under combined delay plus loss, both protocols follow the expected upward shift in RTT as impairment severity increases. HTTP/3 typically preserves a slightly lower median and a tighter bulk distribution, while HTTP/2 more often shows broader inflation of the centre or heavier tails. However, these differences remain modest compared with the influence of the access medium, queue occupancy, and contention pattern.

Interpretation. Taken together, the results show that HTTP/3 does not fundamentally change median latency in SOHO networks: in most of the tested conditions, HTTP/2 and HTTP/3 remain within about 1 ms of each other at the median. The more important differences appear in distribution shape, especially in the tails and under contention. HTTP/3 is generally competitive under clean and mildly impaired conditions, but it becomes more vulnerable to burstiness and occasional failure when mixed TCP/UDP uplink contention stresses the consumer access link.

The broader implication is that last-mile queueing and access-link configuration matter more than protocol version alone. In the tested environment, fair queueing, active queue management, or modest uplink rate shaping would likely yield larger and more reliable latency improvements than simply preferring HTTP/2 or HTTP/3 in isolation. The practical lesson is therefore not that HTTP/3 is universally better or worse, but that its realised performance depends strongly on how the access network handles mixed traffic under load.

Table 7
Representative wired median RTTs across major scenarios.

Scenario	H2 median [ms]	H3 Median [ms]	Difference [ms]	H3 Advantage
Baseline (clean)	7.90	6.99	0.91	11.5% lower
Packet loss 1%	7.67	7.66	0.01	≈ equal
Busy uplink (TCP)	8.05	7.24	0.81	10% lower, higher variance
Delay + loss (100 ms + 1%)	108.25	107.30	0.95	0.9% lower

Although this study focuses on RTT-oriented measurements rather than full browser workloads, the observed tail effects are still relevant to user experience. In practice, elevated upper-percentile RTTs can slow connection setup, delay the completion of small object fetches, and increase responsiveness problems during interactive web activity, especially when many short exchanges share a stressed access link. The results therefore suggest that queueing-induced tail inflation in the last mile may affect real user experience even when median latency remains relatively close across protocols.

5. Discussion of key findings

The measurements in this paper show that HTTP/3 behaviour in SOHO networks is more nuanced than a simple "faster or slower than HTTP/2" narrative. Under clean conditions, both protocols achieve very similar median RTTs, with HTTP/3 often showing only a small advantage. Once packet loss, Wi-Fi contention, and sustained uplink load are introduced, however, observed performance becomes dominated by last-mile queueing and access-link conditions rather than by transport choice alone.

A first key finding is that median latency is only weakly affected by protocol choice in the tested environment. Across wired and wireless baselines, and even under moderate packet loss and combined delay-plus-loss impairment, HTTP/2 and HTTP/3 usually remain within about 1 ms of each other at the median. This suggests that on lightly loaded access links, transport-level differences are often overshadowed by access-medium characteristics, path conditions, and normal network variability. For many users, simply enabling HTTP/3 should therefore not be expected to produce dramatic per-request latency gains by itself.

A second key finding is that the main protocol difference appears in the tails rather than at the centre of the distribution. Under sustained uplink contention, particularly when a long-lived TCP upload is present, foreground RTT becomes markedly more bursty and higher-percentile delays increase sharply. In our experiments, this effect is especially visible for HTTP/3 probes under mixed TCP/UDP contention. The results are consistent with the interpretation that access-link queueing under mixed traffic can disadvantage small QUIC bursts, even when median latency remains relatively close to HTTP/2. At the same time, the study does not directly instrument scheduler internals, NAT mapping policies, or binding timers, so these should be treated as plausible contributing mechanisms rather than as individually proven causes.

A third finding is that the disadvantage observed under busy uplink conditions is not a general property of QUIC itself. When the background upload also uses HTTP/3, both median RTT and spread return much closer to baseline, especially compared with the more disruptive mixed TCP/UDP case. Similarly, under combined delay and loss, both protocols broadly follow the configured base delay, while HTTP/3 often retains a slightly lower median and a somewhat tighter bulk distribution. These observations support the view that the main penalty arises from access-link contention patterns and queueing interactions rather than from an intrinsic weakness in HTTP/3.

From a practical perspective, the results suggest that operators and home users may gain more by improving access-link behaviour than by focusing only on protocol version. Fair queueing, active queue management, or modest uplink rate shaping are likely to reduce burstiness and tail inflation more effectively than simply preferring HTTP/2 or HTTP/3 in isolation. In other words, HTTP/3 performance in the last mile depends heavily on whether the surrounding access network handles mixed traffic fairly under load. Where such queue management is absent, HTTP/3 benefits may be reduced or even reversed during heavy upstream contention.

This also has a user-facing interpretation. Web browsing and application responsiveness are often influenced less by average delay than by whether short request–response exchanges complete promptly and consistently, particularly when pages depend on many small transfers or interactive control messages. While this paper does not directly measure page load time, buffering events, or browser object prioritisation, the reported RTT tail inflation under stressed uplink conditions is consistent with the type of latency instability that can degrade perceived responsiveness at the application layer.

Although each scenario used long runs with large probe counts, the present analysis is primarily descriptive rather than inferential. We report means, medians, interquartile ranges, and high-percentile values to characterise central tendency and tail behaviour, but we do not yet present repeated independent trial sets with formal significance tests or confidence intervals across runs. The large sample sizes help stabilise within-run estimates, especially for tail behaviour, but future work should strengthen the statistical treatment by adding repeated runs across time, uncertainty intervals, and explicit significance analysis.

To provide a modest robustness check without overstating the inferential strength of the study, 95% bootstrap confidence intervals were computed from the trimmed RTT samples for selected key conditions, including baseline wired, baseline Wi-Fi, and the concurrent-uplink-load Wi-Fi case with a background HTTP/2 upload. These intervals were used to characterise within-run stability of the median and upper-tail metrics (95th and 99th percentiles), rather than to claim full cross-run statistical significance across repeated independent experiments. The resulting intervals were narrow for baseline medians and remained clearly elevated for the high-percentile RTTs observed under stressed uplink conditions, supporting the conclusion that the reported tail effects are not driven by a few isolated samples within a run.

For example, in the concurrent Wi-Fi scenario with a background HTTP/2 upload, the foreground H2 probe showed a median RTT of 15.95 ms with a 95% bootstrap interval of [15.07, 16.86] ms and a 99th percentile of 157.02 ms with interval [151.04, 168.60] ms, while the corresponding H3 probe showed a median of 11.53 ms [11.43, 11.69] ms and a 99th percentile of 141.65 ms [134.33, 147.82] ms.

Additional delay-only, loss-only, and combined-impairment experiments conducted in a separate lab environment showed qualitatively similar trends, but are not pooled with the main SOHO dataset because the network context, path characteristics, and bottleneck conditions differed. They are therefore treated as additional supporting evidence rather than as part of the core comparison presented in this paper.

There are also important limitations. The experiments were carried out using a specific set of consumer routers, Wi-Fi settings, access conditions, and traffic patterns, so the reported absolute values should not be treated as universally representative of all access networks. In particular, router queue-management design, firmware behaviour, ISP uplink policies, and access-medium characteristics may change the magnitude of the observed effects. The contribution of this paper is therefore strongest at the level of mechanism and qualitative trend: namely, that last-mile queueing and mixed-traffic interaction can dominate realised HTTP/3 behaviour under load, even when median RTT

remains close to HTTP/2. In addition, the study focuses mainly on RTT and controlled request–response probing rather than full browser workloads, video delivery, or page-rendering performance. The user-facing implications discussed in this paper should therefore be interpreted as mechanism-based inferences from transport behaviour rather than as direct measurements of page experience. This was a deliberate choice to isolate transport and queueing behaviour in a repeatable manner, but future work should extend the analysis to a broader set of router classes, access technologies, and user-facing metrics such as page load time, buffering events, and object prioritisation.

Overall, the main message is that HTTP/3 is not a universal latency improvement for the last mile, but neither is it inherently worse than HTTP/2. In the tested SOHO scenarios, the two protocols remain close in median RTT across most conditions. The more important factor is how the access link behaves under loss, wireless contention, and especially mixed TCP/UDP uplink load. In practice, the real determinant of whether HTTP/3 performs well is often not the protocol itself, but the queueing and buffering behaviour of the access network that surrounds it.

6. Conclusion and future work

This paper presented a large-scale empirical comparison of HTTP/2 and HTTP/3 performance on real SOHO access networks, using more than half a million RTT samples collected across Ethernet, Wi-Fi, controlled packet loss, uplink contention, and combined delay-plus-loss impairment. The results show that protocol-level differences are modest in clean baseline conditions. HTTP/3 typically achieves slightly lower medians and tighter distributions, but the gains are small relative to the variability introduced by the access medium, especially Wi-Fi.

The most pronounced differences appear under uplink saturation, where a long-lived TCP upload can dominate the upstream buffer and impose significant tail latency on short UDP-based HTTP/3 probes. This behaviour does not appear to arise from an inherent weakness in QUIC, but is instead consistent with queueing unfairness in mixed-protocol traffic at consumer routers. When contention is homogeneous, for example with both flows using QUIC, HTTP/3 performance stabilises and the observed penalties largely disappear.

Overall, the findings demonstrate that the practical performance of HTTP/3 in the last mile is shaped primarily by access-link queueing and buffer policies rather than by the protocol itself. HTTP/3 does not inherently reduce latency in SOHO environments, nor does it consistently underperform relative to HTTP/2. Instead, transport behaviour is dominated by the configuration and behaviour of the bottleneck access link.

Future work

There are several natural extensions to this work. First, future experiments should incorporate a wider range of consumer routers, including devices that already employ FQ-CoDel or other modern active queue management, in order to test how robust the observed queueing and contention patterns remain across different consumer hardware designs. Second, extending the analysis to additional metrics such as page load time, object prioritisation, throughput under browser-driven workloads, and media-session stability would provide a more direct picture of how the observed RTT tail effects translate into real-world user experience. Third, the impact of different QUIC congestion control algorithms, including BBR variants, should be investigated, particularly on Wi-Fi links where airtime fairness is a key constraint. Finally, repeating the study across multiple ISPs, cellular networks, and next-generation Wi-Fi 6/7 environments would help determine how broadly the observed qualitative trends generalise beyond the specific SOHO conditions examined here.

In summary, HTTP/3 performs well when access queues are well managed, but its potential benefits may be muted or reversed in congested or poorly configured SOHO networks. Improving fairness and queue discipline at the access link remains essential for realising the full performance advantages of modern web transports.

CRediT authorship contribution statement

Astrit Krasniqi: Writing – review & editing, Writing – original draft, Visualization, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Shahram Salekzamankhani:** Writing – review & editing. **Bal Virdee:** Writing – review & editing.

Declaration of generative AI and AI-assisted technologies in the manuscript preparation process

During the preparation of this work the authors used ChatGPT to improve the clarity and readability of some sections. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the content of the published article.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

[1] Gianluca Perna, Martino Trevisan, Danilo Giordano, Idilio Drago, A first look at HTTP/3 adoption and performance, Comput. Commun. 187 (2022) 115–124.
[2] Abhinav Gupta, Radim Bartos, User experience evaluation of HTTP/3 in real-world deployment scenarios, in: 2022 25th Conference on Innovation in Clouds, Internet and Networks and Workshops, ICIN, 2022, pp. 17–23.
[3] Carlo Augusto Grazia, Natale Patriciello, Toke Høiland-Jørgensen, Martin Klapez, Maurizio Casoni, Aggregating without bloating: Hard times for TCP on Wi-Fi, IEEE/ACM Trans. Netw. 30 (5) (2022) 2359–2373.
[4] Xinle Du, Jie Li, Yiyang Shao, Wei Wang, Shuihai Hu, Jingbin Zhou, Kun Tan, Revisiting congestion control for WiFi networks, in: Proceedings of the 8th Asia-Pacific Workshop on Networking, APNet 2024, 2024, pp. 88–94.
[5] Constantin Sander, Ike Kunze, Klaus Wehrle, Analyzing the influence of resource prioritization on HTTP/3 HOL blocking and performance, in: 2022 Network Traffic Measurement and Analysis Conference, TMA, 2022.
[6] Mengying Zhou, Yang Chen, Shihan Lin, Xin Wang, Bingyang Liu, Aaron Yi Ding, Dissecting the applicability of HTTP/3 in content delivery networks, in: 2024 IEEE 44th International Conference on Distributed Computing Systems, ICDCS, 2024, pp. 936–946.
[7] Xumiao Zhang, Shuowei Jin, Yi He, Ahmad Hassan, Z. Morley Mao, Feng Qian, Zhi-Li Zhang, QUIC is not quick enough over fast internet, in: Proceedings of the ACM Web Conference 2024, WWW'24, 2024, pp. 2713–2722.
[8] Jun Sun, Mengzhu Guo, Jian Liu, Preamble slice orderly queue access scheme in cell-free dense communication systems, Digit. Commun. Networks 11 (1) (2025) 126–135.
[9] Weiwei Jiang, Haoyu Han, Miao He, Weixi Gu, ML-based pre-deployment SDN performance prediction with neural network boosting regression, Expert Syst. Appl. 241 (2024) 122774.
[10] Jianchun Liu, Jiaming Yan, Hongli Xu, Lun Wang, Zhiyuan Wang, Jinyang Huang, Chunming Qiao, Accelerating decentralized federated learning with probabilistic communication in heterogeneous edge computing, IEEE Trans. Netw. 34 (2026) 486–501.
[11] Jinbin Hu, Yi He, Wangqing Luo, Jiawei Huang, Jin Wang, Enhancing load balancing with in-network recirculation to prevent packet reordering in lossless data centers, IEEE/ACM Trans. Netw. 32 (5) (2024) 4114–4127.
[12] Jinbin Hu, Chaoliang Zeng, Zilong Wang, Junxue Zhang, Kun Guo, Hong Xu, Jiawei Huang, Kai Chen, Load balancing with multi-level signals for lossless datacenter networks, IEEE/ACM Trans. Netw. 32 (3) (2024) 2736–2748.
[13] Jinbin Hu, Houqiang Shen, Jiawei Huang, R. Simon Sherratt, Jin Wang, A high-performance sketch with dynamic memory allocation for priority-oriented data stream processing, IEEE Trans. Comput. 75 (2) (2026) 720–733.