Performance of QUIC over Starlink LEO Satellite Networks

Introduction

Low Earth Orbit (LEO) satellite constellations like SpaceX's **Starlink** are redefining global Internet connectivity, offering much lower latency than traditional geostationary satellites. At the same time, the **QUIC** protocol (standardized as HTTP/3) is emerging as a dominant transport for web traffic, promising faster connection setup and improved performance over TCP. This report evaluates QUIC's performance on Starlink-class LEO networks, drawing on both real-world measurements and simulation studies from 2023–2025. We compare QUIC with conventional protocols (TCP/TLS as used in HTTP/2) across key metrics – **latency**, **throughput**, **jitter**, and **packet loss** – and examine application-level impacts for video streaming, voice/video calls, online gaming, and other latency-sensitive uses. Findings are organized into measurements from live Starlink systems and insights from controlled experiments, with tables and figures highlighting protocol performance differences.

QUIC and LEO Satellites: Background

QUIC is a UDP-based transport protocol integrating TLS encryption and multiplexed streams at the transport layer. Compared to TCP (which underpins HTTP/2), QUIC requires fewer round trips to establish a secure connection (often 1-RTT for handshake vs. 3-RTT for TCP+TLS), and it eliminates head-of-line blocking between parallel streams. These traits suggest QUIC can reduce latency and improve throughput utilization, especially on high-latency paths . QUIC also supports connection migration, meaning an ongoing session can survive IP address changes – a potential advantage if a user's satellite link is handed off to a new ground station or satellite. However, QUIC's use of fully encrypted headers prevents the use of traditional Performance-Enhancing Proxies (PEPs) that historically helped TCP in satellite networks .

Starlink LEO network characteristics differ markedly from terrestrial networks. Starlink satellites orbit ~550 km high, yielding base round-trip light-of-flight latency of ~20–40 ms (much lower than ~500+ ms for geostationary satellites) . Each user's dish connects to satellites that hand off to one another in a coordinated schedule – in Starlink's case, a **15-second globally**

synchronized interval for link reconfiguration . At these interval boundaries (occurring at 0:12, 0:27, 0:42, 0:57 of each minute), the network may switch the serving satellite or ground station, causing brief disruptions. As a result, Starlink links see **periodic latency spikes and packet loss bursts** at 15 s intervals . Fig. 1 illustrates how Starlink latency fluctuates with each satellite switch – the minimum latency resets every 15 s, while jitter accumulates in between due to the satellite's motion and queueing . These rapid variations pose challenges for congestion control algorithms that expect more stable paths .

Figure 1: Starlink latency over sequential 15-second intervals (blue). Every 15 s, a network reconfiguration causes latency to shift (red step function), introducing jitter .

Starlink offers high raw capacity – individual users often see 50–200 Mbps of throughput – but it is a **shared medium** with performance depending on network load and coverage. The user terminal (Dishy) and satellite use large buffers to maximize throughput, which can introduce bufferbloat under heavy traffic . Overall, Starlink's "bent-pipe" architecture (satellite simply relays between user and gateway) contributes roughly ~40 ms of latency in the best case . In regions with dense ground station (gateway) infrastructure, Starlink latency and speeds are competitive with terrestrial broadband and 5G . Regions poorly served by ground stations experience higher latency (e.g. >70 ms) due to longer routing paths . Crucially for transport protocols, Starlink's rapid bandwidth and latency fluctuations (particularly the 15 s cycle) can confuse loss-based congestion control – a packet loss during a satellite handover is not due to true congestion, yet standard TCP or QUIC algorithms might needlessly cut their sending rate . This context sets the stage for comparing QUIC vs. TCP behavior on LEO networks.

Real-World Performance Measurements on Starlink

Recent measurement studies provide insight into QUIC and TCP performance over live Starlink connections. **Geoff Huston (APNIC)** conducted controlled file transfer tests over Starlink to evaluate TCP versus QUIC under different congestion controllers. His findings indicate that for bulk data transfer, **QUIC achieves similar throughput and behavior as TCP when using the same congestion control algorithm** . In other words, QUIC's transport-layer improvements do not magically increase link capacity – a QUIC flow using the standard CUBIC congestion control will "behave like any other CUBIC session" on Starlink, and a QUIC flow using BBR will behave like BBR, much as TCP does . This was evidenced by throughput traces: a TCP+CUBIC flow averaged about ~45 Mbps on a loaded Starlink cell (peaking near 240 Mbps before packet loss forced a window reduction) , while a QUIC+CUBIC flow showed a nearly identical throughput curve . Table 1 summarizes a comparison from APNIC's experiments, showing that protocol differences are minor relative to the choice of congestion control:

Protocol & CC	Average Throughput	Loss Events	Notes
TCP (CUBIC)	~45 Mbps	Periodic (15 s)	Saw ~240 Mbps burst then saw periodic loss causing window reset to ~50 Mbps .
QUIC (CUBIC)	~40–50 Mbps (similar)	Periodic (15 s)	Similar sending pattern as TCP Cubic .
TCP (BBR)	100–200+ Mbps (highly variable)	Frequent loss	Quickly fills pipe to 150–200+ Mbps, but induces more loss; throughput fluctuates widely .
QUIC (BBR)	~150+ Mbps (similar to TCP BBR)	Frequent loss	Behaves like TCP BBR; attempts to maximize rate, achieving higher goodput but with jitter .

Table 1: Observed Starlink downlink throughput for TCP vs QUIC with common congestion algorithms. QUIC's performance largely mirrors TCP when using the same CUBIC or BBR controller . BBR can attain higher throughput than CUBIC on Starlink but at the cost of more loss and variability.

These results illustrate that **congestion control, not the base protocol, is the dominant factor** for bulk transfer performance over LEO links. Notably, BBR (which tries to estimate bandwidth and maximize pipe utilization) was able to push a Starlink link closer to its 200+ Mbps capacity, but caused frequent oscillations and packet loss bursts . CUBIC, in contrast, ramped up more cautiously and plateaued around 50–70 Mbps in a loaded scenario, avoiding heavy loss but under-utilizing the available capacity . This underlines a key challenge: the fast-changing latency and periodic interruptions on Starlink can fool CUBIC into cutting its congestion window too often, impairing throughput efficiency . Indeed, Huston observes that standard CUBIC "is still prone to reduced efficiency" during Starlink's 15 s reconfiguration events . He suggests that explicit signals (such as using ECN or scheduling awareness) could help distinguish losses due to handovers from true congestion .

Beyond synthetic iperf tests, a **large-scale empirical study (Mohan et al. 2024)** collected **19.2 million speed tests** from Starlink users worldwide . They report Starlink's median performance to be **"competitive to terrestrial cellular networks"**, with typical user download speeds in the tens of Mbps and median latencies around 40–50 ms . In fact, at the 75th percentile, Starlink users saw **≈50–100 Mbps download and 4–12 Mbps upload** throughput – comparable to 4G/5G broadband. However, consistency varies: Starlink latency can **spike substantially under load**, e.g. if a user saturates their link, latency can climb (due to bufferbloat) from a baseline of ~40 ms to well over 100 ms . This was confirmed by Starlink's own telemetry: the user terminal maintains a large send buffer, which can drive up delay when filled . For QUIC traffic, which often involves multiplexed streams and possibly continuous media, such latency variability can affect quality of experience if not managed (e.g. by application jitter buffers).

Importantly, Mohan et al. also measured fine-grained Starlink performance using instrumented clients. They found that every 15 s interval, there were **noticeable throughput and RTT degradations at sub-second granularity**, aligning exactly with Starlink's global reconfiguration schedule . In controlled tests with Starlink dishes in Germany and Scotland, they observed that within each 15 s epoch, latency and throughput stayed relatively stable, but **at the moment of handover, RTT could jump by tens of milliseconds and throughput would dip briefly** . For example, Figure 2 (from their paper) shows Starlink's throughput over several intervals: throughput is high and steady for ~10–12 seconds, then drops at the reconfiguration boundary (vertical lines), recovering shortly after . These periodic performance hiccups were **synchronized globally** (the two distant Starlink terminals showed aligned events) . Interestingly, the study concluded that **these effects are not strictly "satellite handoffs"** (since even with only one satellite in view, similar 15 s oscillations occurred); rather, they appear to be a result of the network's scheduled routing updates or resource shuffling in the bent-pipe architecture .

Such findings have direct implications for transport protocols. A QUIC or TCP flow that encounters a spike in RTT or brief outage every 15 seconds might falsely infer network congestion. Traditional loss-based congestion control will cut its send rate upon detecting packet loss or large delay, thereby hurting throughput until it ramps up again . In essence, **jitter and loss induced by LEO reconfigurations can degrade transport performance if the protocol isn't LEO-aware**. We next examine how QUIC compares to TCP in coping with these dynamics, and what enhancements have been proposed.

QUIC vs. TCP vs. HTTP/2: Protocol Comparisons

While QUIC and TCP exhibit similar raw throughput on Starlink when using comparable congestion algorithms, there are several areas where QUIC can outperform or improve upon TCP/HTTP2 in LEO conditions:

- **Connection Establishment Latency:** QUIC combines the transport and cryptographic handshake into one step. Over a high-latency path, this yields a tangible speedup in time-to-first-byte. For example, establishing an HTTP/3 connection can be ~30% faster than HTTP/2 (which requires a TCP 3-way handshake plus TLS handshake) . On Starlink, with RTTs in the 40–50 ms range, QUIC's 1-RTT handshake saves about one full round trip. This benefit is most pronounced for short-lived connections and request bursts – web pages with many small objects or API calls see reduced setup overhead. In one emulated study, QUIC's faster slow-start allowed it to reach its throughput plateau *up to 2 seconds sooner* than TCP on high-delay (GEO-like) links . Even in LEO scenarios, the handshake savings translate to snappier page loads and less waiting for initial data.

- **Multiplexing and Head-of-Line Blocking:** HTTP/2 uses multiplexing over a single TCP connection, which avoids the overhead of many parallel TCP connections. However, because TCP delivers bytes in order, a lost packet in one HTTP/2 stream will block *all* streams behind that packet until it's retransmitted. This **head-of-line (HOL) blocking** can be detrimental on lossy or high-delay links – e.g. one small lost packet on a 50 ms link could stall all streams for that 50+ ms repair time, multiplying delay if it happens frequently. QUIC, in contrast, multiplexes streams at the transport layer: lost packets only stall the affected stream, while other streams continue unhindered. This is a big advantage for satellite networks where packet loss (though infrequent on Starlink, aside from handovers) does occur. In practical terms, QUIC's design prevents an isolated loss from, say, a video segment download from halting an unrelated API response. **Web performance tests confirm that HTTP/3 (QUIC) outperforms HTTP/1.1 and HTTP/2 in high-latency scenarios**, especially when multiple objects are in flight . Kosek et al. (2022) showed that in a LEO/GEO testbed, HTTP/3 reduced page load times by several seconds compared to HTTP/1.1, thanks to parallelism and fewer round trips . Even without extreme GEO latency, we can expect HTTP/3 to recover from Starlink's brief packet losses more gracefully than HTTP/2, yielding smoother browsing and streaming.
- **Connection Migration and Reliability:** A unique strength of QUIC is that connections are identified by an arbitrary ID, not the 5-tuple of IP/port. This means if the client's network address changes, the QUIC connection can continue (as long as the QUIC endpoint is aware of the migration) without restarting. In a LEO context, this could be beneficial if a user's traffic is moved to a different gateway or satellite mid-connection, potentially resulting in an IP change or reroute. In practice, Starlink's network tries to preserve session continuity through its NAT and routing, but QUIC provides an extra layer of resilience. For example, if a brief outage occurred during a satellite handover, a TCP connection might time out and need to be re-established, whereas QUIC could resume transmission immediately when connectivity returns (and even employ 0-RTT resumption for quick reconnect if needed). Real-world Starlink users have noted fewer noticeable disruptions on QUIC-based services (like modern web apps) compared to some older VPNs or streams over TCP, partly because QUIC can better tolerate short interruptions. One study pointed out that QUIC's encrypted nature means classic TCP splitting proxies can't be used, but on the flip side, QUIC's agility could reduce reliance on such proxies by handling the variability end-to-end .
- **Congestion Control Customization:** Since QUIC is implemented in user space for many stacks, it is relatively easy to experiment with new congestion control or recovery algorithms tailored to LEO conditions. Researchers have proposed QUIC-specific tweaks, such as **LEO-aware congestion control that anticipates periodic handover losses**. For instance, *StarQUIC* (Kamel et al. 2024) introduced a mechanism to detect the 15 s interval boundaries and avoid halving the congestion window during those moments . By pausing congestion window reduction when a loss is expected due to a handover (and not due to true queue overflow), StarQUIC was able to improve QUIC file transfer completion times by **up to 35%** over Starlink links . This kind of innovation is simpler to deploy at the application layer with QUIC than in kernel-space TCP. Such improvements mean QUIC can more quickly recover after the brief packet loss burst that happens every reconfiguration, whereas unmodified TCP would enter slow start or otherwise back off

unnecessarily. In essence, QUIC offers a flexible platform to incorporate LEO-specific optimizations (or even satellite-specific congestion controllers) without waiting for OS TCP stacks to catch up.

In summary, **QUIC/HTTP3 holds several performance advantages over TCP/HTTP2 for LEO satellite networks**: faster session startup, better multi-stream robustness, and adaptability. However, it's important to note that in steady-state bulk throughput, QUIC is not inherently higher throughput than TCP – the bottleneck is the network capacity and the congestion control algorithm, not the protocol framing. As Huston quips, "in transport performance terms QUIC is just TCP" when it comes to how it fills a Starlink pipe . The true benefits of QUIC emerge in scenarios with many short flows or interactive streams, which are common in modern web and real-time applications.

Latency, Throughput, and Jitter Metrics

Starlink's performance can be quantified by several key metrics relevant to application QoS:

- **Latency:** Under good conditions, Starlink users experience one-way latency around 20–30 ms and RTT around **40–50 ms** to the nearest point-of-presence . This is only slightly higher than terrestrial broadband in many cases. However, latency is not constant; it **drifts and jumps** in a sawtooth pattern due to satellite movement and handovers (Figure 1). Within each 15 s window, latency gradually increases by a few milliseconds as the satellite moves toward the horizon, then drops when the next satellite takes over a closer position . Mohan et al. observed RTT variations of ~10–20 ms between intervals . Under heavy load, latency increases significantly – their data shows Starlink's RTT can **double under uplink traffic load** (e.g. Zoom uplink pushed one-way delay to ~52±14 ms vs 27±7 ms on an idle terrestrial line) . Such bufferbloat can be mitigated by pacing or active queue management, but currently the user terminal appears to prioritize throughput, allowing latency to spike . Compared to 5G cellular (which had ~39 ms RTT in the same tests), Starlink's base latency was similar or better, but the variance was higher .
- **Throughput:** Starlink downlink speeds (to the user) typically range from 50 Mbps up to 200 Mbps+ in uncongested cells . The **median download throughput** as of late 2023 was on the order of tens of Mbps, with top quartile users exceeding 100 Mbps . Uplink (user-to-internet) speeds are lower, usually 5–20 Mbps, sufficient for HD video streaming or video calls. Notably, throughput can drop during the brief handover events; Mohan et al. saw both uplink and downlink throughput **dip at each 15 s boundary** before recovering . Over a longer timescale, throughput also varies with time of day: Starlink is a shared satellite-cell system, so **evening peaks can see reduced speeds** if many users in a cell are active . Huston documented instances where a Starlink link delivered ~200

Mbps in off-peak hours (minimal competing traffic), but only ~50 Mbps at peak usage times due to capacity sharing . In either case, QUIC and TCP have shown they can utilize whatever bandwidth is available, though TCP-CUBIC may ramp more slowly. Packet loss on Starlink is generally low (well under 1%) except during the reconfiguration instants where a flurry of losses can occur . These periodic losses are not indicative of overall link quality, but they do reset congestion windows for loss-sensitive protocols.

- **Jitter:** The **jitter** on Starlink – meaning short-term variation in packet delay – is higher than on fiber and slightly higher than cellular. For instance, Zoom video call measurements showed Starlink's uplink one-way delay varying with a standard deviation of ~14 ms, versus ~7 ms on a terrestrial link . The downlink jitter was smaller (~11 ms std) since the Starlink "bent-pipe" hop mostly impacts the upstream leg in that setup . The most salient jitter comes from the 15-second periodic events: essentially an abrupt step in delay. If an application's packets straddle a reconfiguration, one might arrive ~20–30 ms later than the previous one. This kind of periodic jitter (as opposed to random network jitter) can be predicted and smoothed if known. Indeed, Starlink's synchronization means every user experiences these blips at the same instants. For QUIC, which can record RTT samples, such jitter might trigger spurious timeout or fast-retransmit decisions if not accounted for. Current QUIC implementations using CUBIC or BBR don't inherently know about the 15 s cycle, but a **LEO-tuned QUIC** could ignore delay spikes exactly at those intervals. Outside of the handover-induced jitter, Starlink's jitter is fairly modest – many users report steady gaming pings with only occasional spike events. Table 2 summarizes typical Starlink latency and jitter compared to a terrestrial link, based on data from Mohan et al. (2024):

Metric	Starlink (LEO)	Terrestrial (Fiber/5G)
RTT (idle)	~40–50 ms (global median)	~5–30 ms (depending on distance)
RTT under load (bufferbloat)	100 ms or more	50–80 ms (with heavy load)
Jitter (std dev of one-way delay)	~10–15 ms (uplink) ; periodic 15–30 ms jumps	~5–10 ms (typical)
Packet loss (steady state)	<0.1% (very low)	<0.1% (very low)
Packet loss (handover event)	Burst of losses over ~100–200 ms	N/A (no equivalent events)

Table 2: Latency and jitter characteristics of Starlink LEO vs. a typical terrestrial network. Starlink has low baseline latency but experiences periodic spikes and higher variability due to satellite reconfigurations. Data synthesized from .

These metrics show that Starlink is fundamentally capable of supporting low-latency applications, but the devil is in the details of how protocols handle the jitter and occasional loss. Next, we discuss how this plays out for specific use cases and applications over QUIC.

Application-Level Performance on Starlink LEO

Video Streaming: Modern video streaming (e.g. YouTube, Netflix) often runs over QUIC (HTTP/3). Such services buffer content to mask network variability, so they are generally tolerant of Starlink's 40–50 ms base latency. The high throughput of Starlink allows even 4K streams to be delivered, as long as the link remains stable. Real-world data shows Starlink can sustain the needed bitrate for HD/UHD streaming; for example, cloud gaming tests (which are essentially real-time video streams) sustained ~23 Mbps video at 1080p 60fps over Starlink with >99% of frames delivered at full resolution . Streaming applications mainly need to worry about **jitter and brief outages**: a 100–200 ms pause every 15 s could cause a video buffer to deplete if it's very shallow. In practice, streaming clients buffer several seconds, so a 0.1–0.2 s dip is negligible. Users report that services like Netflix run without interruption on Starlink. QUIC's contribution here is quick recovery from lost packets (preventing a minor loss from halting all streams) and the ability to ramp up the video bitrate quickly after startup or a resolution switch (thanks to faster handshake and no head-of-line blocking). A 2019 study of QUIC vs TCP for video streaming found that users often didn't notice a difference, as both protocols can ensure smooth playback, but QUIC tended to reduce the incidence of playback stalls under challenging network conditions . We can infer that on Starlink, QUIC's resilience to sporadic packet loss helps maintain video quality during those handover blips. Overall, Starlink's bandwidth is more than sufficient for streaming; latency is only a factor in live streaming (where viewer comments or interactions have a slight delay). For on-demand streaming, Starlink+QUIC provides an experience comparable to terrestrial links, with fast startup and stable throughput.

Video Conferencing and VoIP: Two-way interactive media (Zoom, Skype, voice-over-IP) are more sensitive to latency and jitter. Tests of **Zoom over Starlink** in 2023 show that Starlink can handle high-quality video calls, but with some extra overhead. In one experiment, a Zoom call between a Starlink user and a cloud server achieved the full 27 fps video uplink, similar to fiber, but the Starlink side had a slightly higher packet loss rate which Zoom mitigated by sending about 150 kbps of FEC (forward error correction) data continually . This kept video quality high despite ~0.4% packet loss. The one-way delay on Starlink during the Zoom call averaged ~52 ms (vs ~27 ms on fiber) – roughly double – and had about twice the jitter . Zoom's jitter buffer compensated for this, and no freezes were observed except extremely briefly during occasional handovers. In fact, the measured frame rate and quality on Starlink were effectively the same as on a terrestrial network . This demonstrates that Starlink's latency, while higher, is still low enough for realtime video, and QUIC or UDP-based apps can use techniques like FEC to counteract the small packet losses . For pure VoIP (audio calls), the requirements are even more lenient: audio codecs can handle 100 ms jitter without issue. Many voice apps use Opus codec with built-in packet loss concealment, so a few lost packets per minute (as might happen at reconfigs) are unnoticeable. Thus, Starlink can support VoIP and video calls well. It's worth noting that Zoom and similar apps do not necessarily use QUIC – they often use UDP with their own RTP/RTCP protocols. But if such applications were to use QUIC (e.g. WebRTC over QUIC or future real-time QUIC extensions), the performance should be similar. The key is that **the network is the limiting factor**: ~50 ms latency is slightly high but acceptable for conversation (it's like talking to someone a few continents away on fiber). Users have successfully held Zoom meetings, Microsoft Teams calls, and even conducted remote broadcast interviews over Starlink links.

Online Gaming: Online gaming covers a spectrum from fast-twitch first-person shooters (very latency sensitive) to slower strategy or RPG games. Starlink's ~40–50 ms base RTT to the nearest cloud server is on par with many cable/DSL connections, so **baseline ping** is quite playable for most games. The concern is the **jitter and occasional packet loss**. Competitive shooters typically have algorithms to smooth small jitter, but a sudden 50–100 ms spike every 15 seconds might cause a minor "rubber-banding" or hiccup if it coincides with critical timing. Anecdotally, gamers on Starlink report generally good experiences, with ping times in the 30–60 ms range to regional game servers and only rare lag spikes. The multi-faceted study did not directly measure standard online game traffic, but it did evaluate **cloud gaming (Amazon Luna)** which effectively combines aspects of streaming and input latency. Their results (Table 3) showed that Starlink supported the cloud game **almost as well as fiber** and in some ways better than cellular. Specifically, Starlink maintained a full 1080p60 stream 99.45% of the time (versus 94% on 5G) and had fewer frame freezes . The **end-to-end "game delay"** – the time between a player action and seeing the result – was about 167 ms on Starlink, only ~33 ms slower than on fiber (134 ms) and virtually the same as on 5G (166 ms) . This indicates that even fast reaction games can be played via Starlink, with QUIC or UDP ensuring the network transport isn't a bottleneck.

Network	1080p Video Uptime	Frame Freezes	Game Input Delay	Round-trip Ping
Fiber (wired)	100%	0 ms/min (no freezes)	133 ± 20 ms	11 ± 13 ms RTT
5G Cellular	94.1%	220 ms/min (occasional)	166 ± 24 ms	39 ± 17 ms RTT
Starlink LEO	**99.45%**	**120 ms/min** (rare)	167 ± 23 ms	50 ± 16 ms RTT

Table 3: Cloud gaming performance (Amazon Luna running 1080p60 video stream) over different networks in 2023. Starlink achieved nearly the same low latency and high video quality as a fiber connection, with only slightly higher input latency than 5G .

For traditional online multiplayer games (not cloud-based), the player's Starlink link just needs to carry relatively small packets to the game server. QUIC is not commonly used in game networking yet (many games use UDP directly), but if it were, the main benefit might be its connection migration – e.g., if a Starlink user's IP address or route changes mid-game, QUIC could seamlessly continue the session, whereas a game over raw UDP/TCP might need to reconnect. However, as long as Starlink's CGNAT keeps the public endpoint stable, games see no interruption. The periodic reconfig events might introduce a packet or two of loss; most game protocols can handle that through interpolation or a quick retransmit of position data. Given the data, **latency-sensitive workloads are quite feasible on Starlink**. The combination of low base delay and QUIC's efficiencies means even interactive applications (augmented reality, telesurgery, etc.) are within reach, especially as SpaceX introduces laser inter-satellite links to cut long-haul latency. Mohan et al. note that in some remote regions, Starlink (with inter-satellite relays active) actually beat terrestrial networks in latency, enabling users in places like rural Kenya to get better ping to services than via the limited local infrastructure . This suggests LEO networks can even advantageously serve latency-sensitive apps for geographically isolated users.

Simulation and Modeling Studies (2023–2025)

To supplement real-world measurements, researchers have built models and testbeds to simulate QUIC over LEO conditions. These studies help explore "what-if" scenarios and optimize performance:

- **StarQUIC (2024)** – As mentioned, this work focused on adjusting QUIC's congestion control during Starlink's predictable handovers. Using both a Starlink hardware testbed and a network emulator, the authors demonstrated that by simply pausing congestion window reduction around the 15 s mark (when losses are likely due to handover), QUIC's completion time for file transfers improved by up to 35% . Notably, their solution was **algorithm-agnostic** – it worked with Reno/CUBIC as well as newer CCAs – and did not require changing the core QUIC protocol, just the sender's logic . This kind of simulation shows that awareness of LEO satellite dynamics can significantly boost performance. It mirrors earlier ideas developed for TCP (e.g. SatCP which used handover timestamps to avoid slow start) but is applied to QUIC's congestion controller . The implication is that future QUIC implementations could include satellite-specific tweaks (or perhaps endpoints could receive satellite handover signals) to maintain throughput across handoffs.
- **Proxy-enhanced QUIC (2022–2023)** – Kosek et al. built a SATCOM emulator using the OpenSAND framework to compare QUIC vs TCP in both LEO (low RTT, moderate loss) and GEO (high RTT) settings . They also introduced a minimal **Performance Enhancing Proxy (PEP)** for QUIC, which terminates the QUIC connection at the satellite gateway and opens a second QUIC connection to the client (essentially splitting the end-to-end path, but in a QUIC-aware way) . In their experiments, even without any proxy, **HTTP/3 outperformed HTTP/1.1** for web page loads over satellite links due to QUIC's multiplexing and faster recovery . With the QUIC-PEP enabled, slow-start ramp-up was faster (QUIC reached its cruising window ~2 seconds quicker) and worst-case page load times dropped by several seconds . This suggests that while QUIC is good out

of the box, there is room to incorporate satellite link awareness (via proxies or in-protocol signaling) to squeeze out even better performance. It’s a balancing act, however: using proxies can violate end-to-end encryption unless done carefully (e.g., the proxy might need to be a trusted terminator of the QUIC TLS session, which has security implications). The trend nonetheless is clear – whether through endpoints or intelligent middleboxes, QUIC can be tuned to handle long delay and intermittent loss better than standard TCP ever could, due to its flexible design.

- **“Realistic LEO transport performance” (Hervella et al. 2023)** – This study (published in **Computer Networks**) modeled a LEO constellation and evaluated TCP vs QUIC throughput and delay using network simulations. The authors found that QUIC and TCP (CUBIC) again showed similar bulk performance when conditions were stable, but QUIC’s advantage emerged in scenarios with high packet reordering or variable delays (thanks to its advanced loss recovery and independent streams). They also noted that enabling Explicit Congestion Notification (ECN) on LEO paths could help both protocols by distinguishing loss from congestion . A takeaway was that *transport protocols function correctly over LEO*, but to approach optimal performance, they must deal with the unique error and delay patterns. The study validated that off-the-shelf QUIC can cope with LEO links without breaking (i.e., its loss recovery tolerates the ~100 ms occasional pause), but there’s potential efficiency gain if QUIC knows about the periodicity of the link.
- **Multipath QUIC for LEO** – An emerging idea is to use multipath capabilities (like MP-QUIC) to bond multiple satellite paths or satellite+terrestrial paths for resilience. A 2024 work by Li et al. proposed a **mobility-aware congestion control for MP-QUIC** in an integrated LEO network, to seamlessly move traffic between paths during satellite handovers . By simulating two subflows (e.g., one Starlink, one 5G), they showed the controller could shift traffic to the better path when one path experiences a handover outage. While not yet practical for consumers (users generally have one Dishy), this line of research may become relevant if end-users or enterprise terminals have access to multiple satellites or networks concurrently for redundancy.

In all these studies, QUIC proves to be a **robust and adaptable protocol for satellite Internet**. The simulations reinforce what real measurements indicated: baseline QUIC performs at least as well as TCP in LEO, and with some tweaks, it can surpass TCP by avoiding legacy behaviors that aren’t optimal for space-based links.

Conclusion

QUIC’s performance over Starlink-class LEO networks is generally excellent and on par with – or better than – traditional TCP-based protocols. Real-world data from 2023–2025 shows that Starlink can deliver low latency (~40–50 ms) and high throughput (50–200 Mbps) globally, and QUIC is well-suited to capitalize on these improvements. QUIC's advantages in connection setup and stream multiplexing translate to faster page loads and more resilient media streams on Starlink compared to HTTP/2. Applications from video streaming to cloud gaming to Zoom calls have been successfully run on Starlink links, with QUIC/HTTP3 often used under the hood to ensure smooth performance. While Starlink's periodic link handovers introduce unique challenges (latency spikes and brief losses every 15 seconds), both TCP and QUIC experience these similarly – and ongoing research indicates QUIC can be enhanced to handle them more gracefully . In bulk transfer terms, QUIC with a standard congestion control (like CUBIC) achieves the same goodput as TCP, but it doesn't inherently "fix" issues like throughput under-utilization during satellite handovers unless modified . Therefore, one key focus is tuning congestion control (whether Cubic, BBR, or novel algorithms) for the highly variable LEO environment. Initiatives like StarQUIC and PEP for QUIC show promising gains by using awareness of satellite dynamics .

Going forward, as LEO networks expand (Amazon's Kuiper, OneWeb, etc.), lessons from Starlink apply broadly. QUIC's encryption will encourage new approaches to performance optimization that don't rely on breaking open the protocol (since the old TCP spoofing tricks by PEPs won't work). Instead, we'll see cooperative approaches – perhaps satellites or ground stations communicating hints to end systems, or machine learning congestion control that identifies the 15-second pattern. The years 2023–2025 have proven that **LEO satellites can support demanding, latency-sensitive workloads**, and that QUIC is up to the task. In fact, QUIC's growing ubiquity (already powering a majority of Internet traffic by some accounts) means most users on Starlink are implicitly benefiting from QUIC's improvements when they use modern web services. As one measurement paper concluded, Starlink's performance is "competitive with the current 5G deployment for supporting demanding real-time applications" – a high praise – and QUIC is a key enabler by mitigating latency and loss at the transport layer. With ongoing enhancements and real-world tuning, QUIC over LEO will only get better, bringing us closer to terrestrial-like Internet experience from anywhere on earth under the satellite constellation.

Sources: This report drew from peer-reviewed studies and measurement reports on LEO networks and QUIC, including Mohan et al. (WWW 2024) , Kamel et al. (MobiCom LEO-Net 2024) , Kosek et al. (IFIP Networking 2022) , and Geoff Huston's Starlink analyses (APNIC, 2024) , among others. These sources provide detailed quantitative evaluations that underpin the comparisons and claims made above.