App Meets Transport
Rethinking Video Streaming over QUIC

Abdelhak Bentaleb
Founder & Director of IN2GM Lab
Concordia University

IETF Media OPerationS (MOPS)

Nov 3, 2025 | IETF 124, MONTREAL, CANADA

Why Rethink Video Streaming?

Motivation

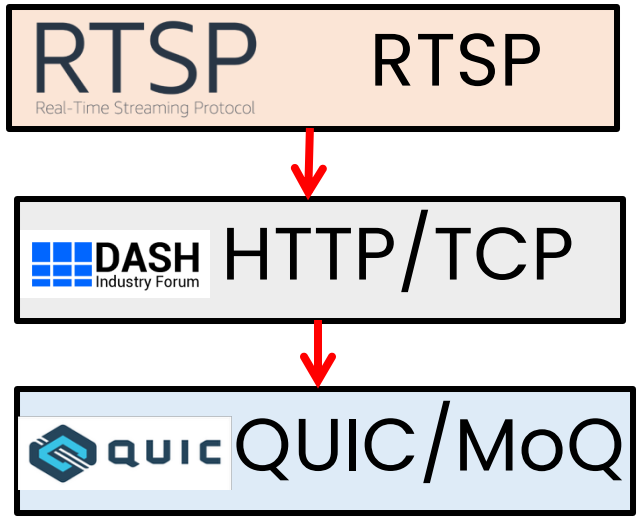
- Video streaming =**>65% of Internet traffic**[1]
- HAS streaming = critical for viewer QoE.
- Historically tuned for **HTTP/TCP** stack.

Limitations of TCP for video:

- Head-of-Line (HoL) blocking.
- Slow start latency issues.
- Ossified transport → hard to innovate.

QUIC = new transport standard (HTTP/3 backbone)

- Transport + Security + Multiplexing + More in userspace.



Timeline of streaming evolution

[1*] **Sandvine's** 2024 Global Internet Phenomena

Two Critical Questions for Modern Streaming

Does a simple migration to QUIC **guarantee** superior QoE?

Is a **transport-centric optimization** of QUIC sufficient to solve the QoE challenges in today's diverse networks?

Why QUIC Matters for Streaming (Theoretically)

From the protocol spec:

- Lower handshake latency = faster startup
- Stream multiplexing = better parallel requests (video + audio + manifests + ads)
- Custom congestion control = opportunity for tuning video traffic
- Richer feedback signals available to apps

Does a simple migration to QUIC **guarantee** superior QoE?

Internet Engineering Task Force (IETF)
Request for Comments: 9000
Category: Standards Track
Published: May 2021
ISSN: 2070-1721

J. Iyengar, Ed.
Fastly
M. Thomson, Ed.
Mozilla

QUIC: A UDP-Based Multiplexed and Secure Transport

Abstract

This document defines the core of the QUIC transport protocol. QUIC provides applications with flow-controlled streams for structured communication, low-latency connection establishment, and network path migration. QUIC includes security measures that ensure confidentiality, integrity, and availability in a range of deployment circumstances. Accompanying documents describe the integration of TLS for key negotiation, loss detection, and an exemplary congestion control algorithm.

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at https://www.rfc-editor.org/info/rfc9000.

Initial Findings

QUIC vs TCP (in general):

- *Startup Delay*: QUIC wins (0-RTT, faster handshake)
- *Rebuffering*: mixed results—depends on CC and AQM
- *Throughput stability*: often similar to TCP
- Less HoL blocking improves resilience for segment delivery
- Conclusion: QUIC helps, but it's not a silver bullet

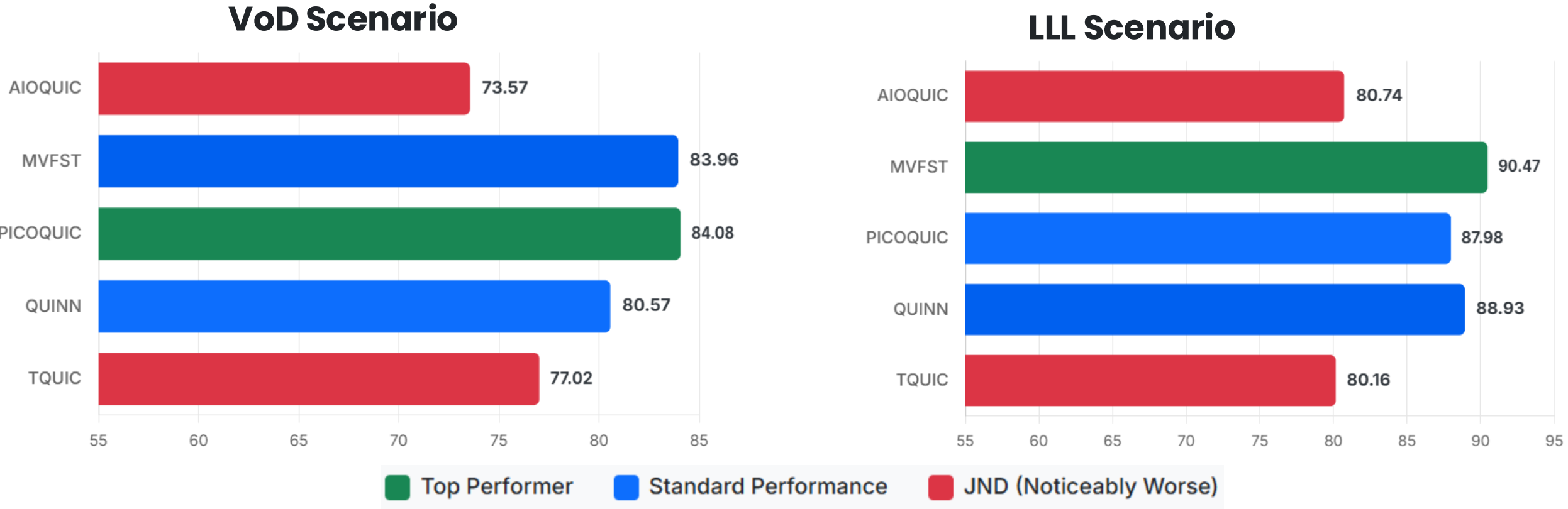
BUT

- Various QUIC heterogeneous implementations (https://interop.seemann.io/)
- Congestion control behavior critical
- Loss recovery not always better than TCP

Reading: Sidhu, J. S., & Bentaleb, A. (2025). *Video streaming over QUIC: A comprehensive study*. arXiv. https://arxiv.org/abs/2505.21769

Flaw #1 — QUIC is a Spec, Not a Solution

Same Streaming Setup + CC: Cubic

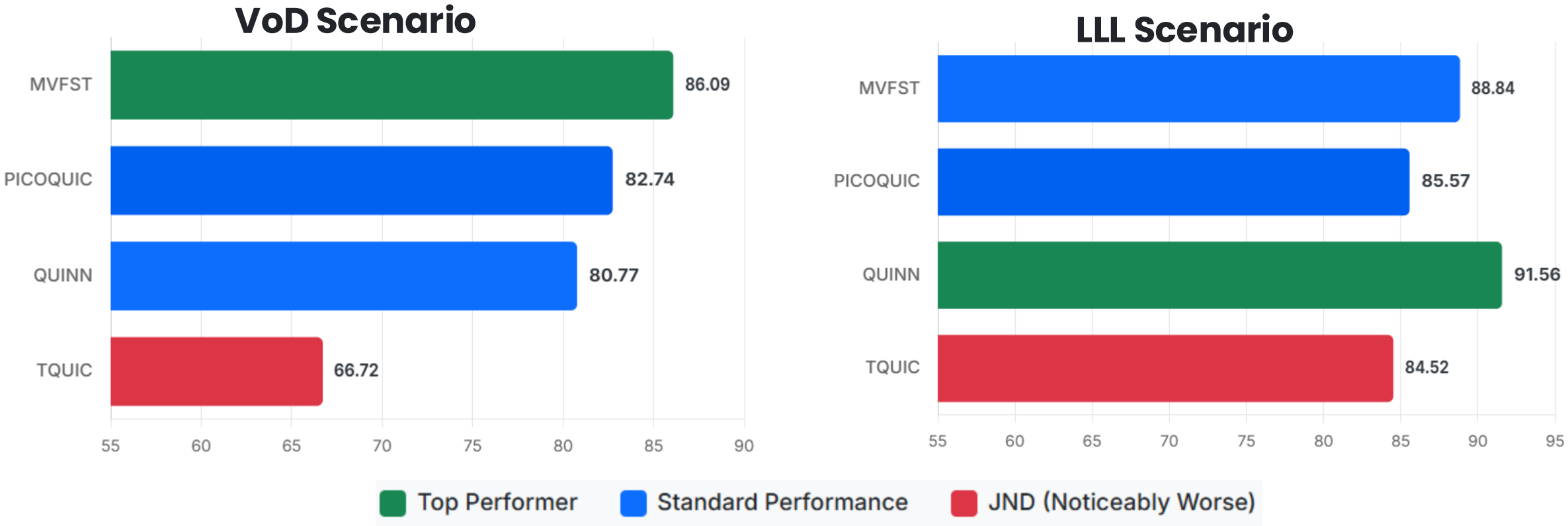


AIOQUIC and TQUIC show a significant quality (high JND) drop in both scenarios

Reading: Sidhu, J. S., & Bentaleb, A. (2025). *Video streaming over QUIC: A comprehensive study*. arXiv. https://arxiv.org/abs/2505.21769

Flaw #1 — QUIC is a Spec, Not a Solution

Same Streaming Setup + CC: BBR



AIOQUIC and TQUIC show a significant quality (high JND) drop in both scenarios

Reading: Sidhu, J. S., & Bentaleb, A. (2025). *Video streaming over QUIC: A comprehensive study*. arXiv. https://arxiv.org/abs/2505.21769

Flaw #1 — QUIC is a Spec, Not a Solution

Main Takeaways

QUIC Implementation Matters

- Video quality depends on CC algorithm + QUIC implementation
- Same CC -> very different results across implementations

MVFST is a Consistent Winner (across tested CCs and scenarios)

Not All QUIC Implementation Are Equal

- TQUIC and AIOQUIC show critical weaknesses with specific CCs → Low QoE

Use Case Sensitivity (VoD vs. LLL)

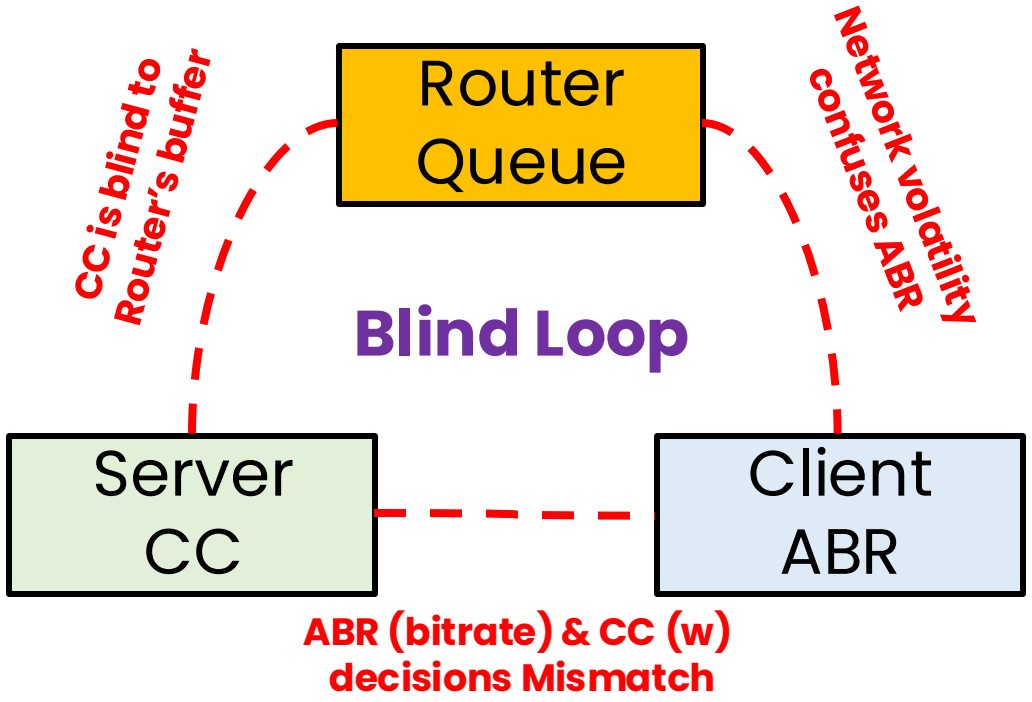
- No single “best” implementation across all scenarios.
- VoD (throughput-driven): MVFST excels.
- LLL (latency-sensitive): QUINN outperforms with better pacing/loss recovery.

Reading: Sidhu, J. S., & Bentaleb, A. (2025). *Video streaming over QUIC: A comprehensive study*. arXiv. https://arxiv.org/abs/2505.21769

Flaw #2 — The Network is an Adversary, Not a Pipe

QUIC is blind to network-internal states like buffer dynamics

- Routers buffers mask the true network state from end-to-end protocols.
- The server's CC sees a stable RTT while a massive queue silently builds in the router.
- When the buffer overflows, the congestion signal is abrupt and sever → Massive, unpredictable bufferbloat that kills live QoE.

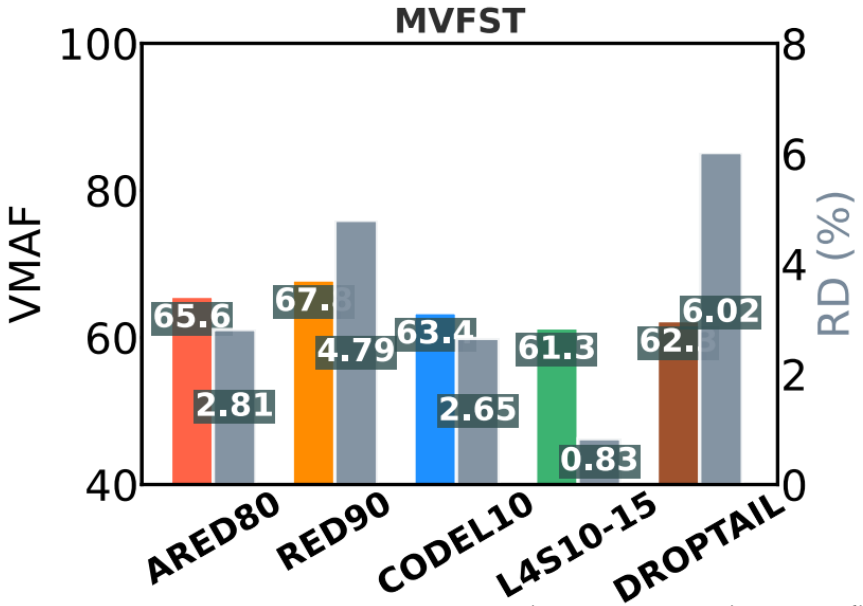


Reading: Sidhu, J. S.,Sandoval, JI., Bentaleb, A., & S Cespedes (2025). *From 5G RAN Queue Dynamics to Playback: A Performance Analysis for QUIC Video Streaming*. arXiv. https://arxiv.org/abs/2508.15087

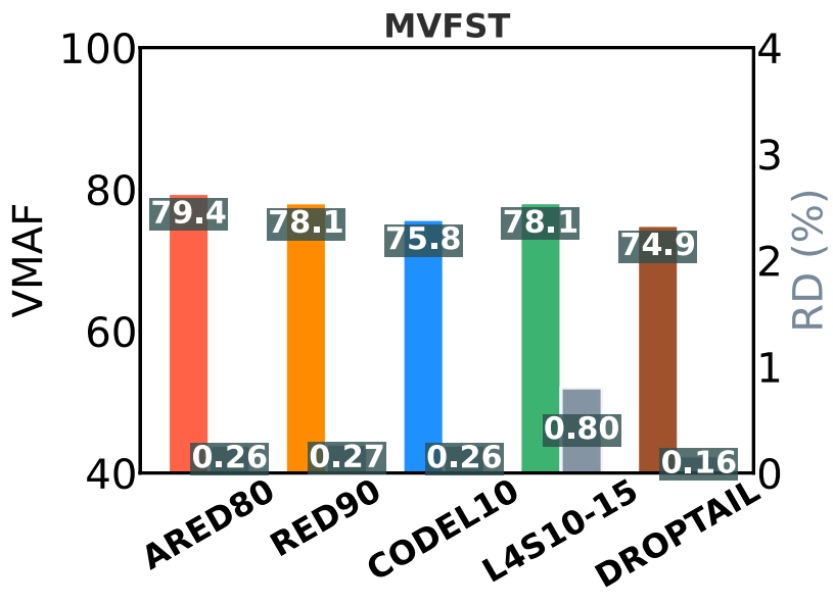
Flaw #2 — The Network is an Adversary, Not a Pipe

Same Setup (QUIC implementation + ABR + streaming scenario/setup)

CC: CUBIC



CC: BBR



CUBIC: aggressive + loss-driven -> fills those network buffers -> high rebuffering & bad QoE
BBR: model the network and avoid filling queues-> 15-point VMAF improvement and near-zero rebuffering

A sophisticated network (with AQMs like CODEL/L4S) cannot fix the problems created by a transport-layer CC that is fundamentally mismatched with the application's needs.

Reading: Sidhu, J. S.,Sandoval, JI., Bentaleb, A., & S Cespedes (2025). *From 5G RAN Queue Dynamics to Playback: A Performance Analysis for QUIC Video Streaming*. arXiv. https://arxiv.org/abs/2508.15087

Key Insight: Q1

Does a simple migration to QUIC **guarantee** superior QoE?

QUIC alone does not guarantee better QoE

Two Critical Questions for Modern Streaming

Is a **transport-centric optimization** of QUIC sufficient to solve the QoE challenges in today's diverse networks?

Is a transport-centric optimization of QUIC sufficient?

Optimizing QUIC Alone (based on transport layer information)

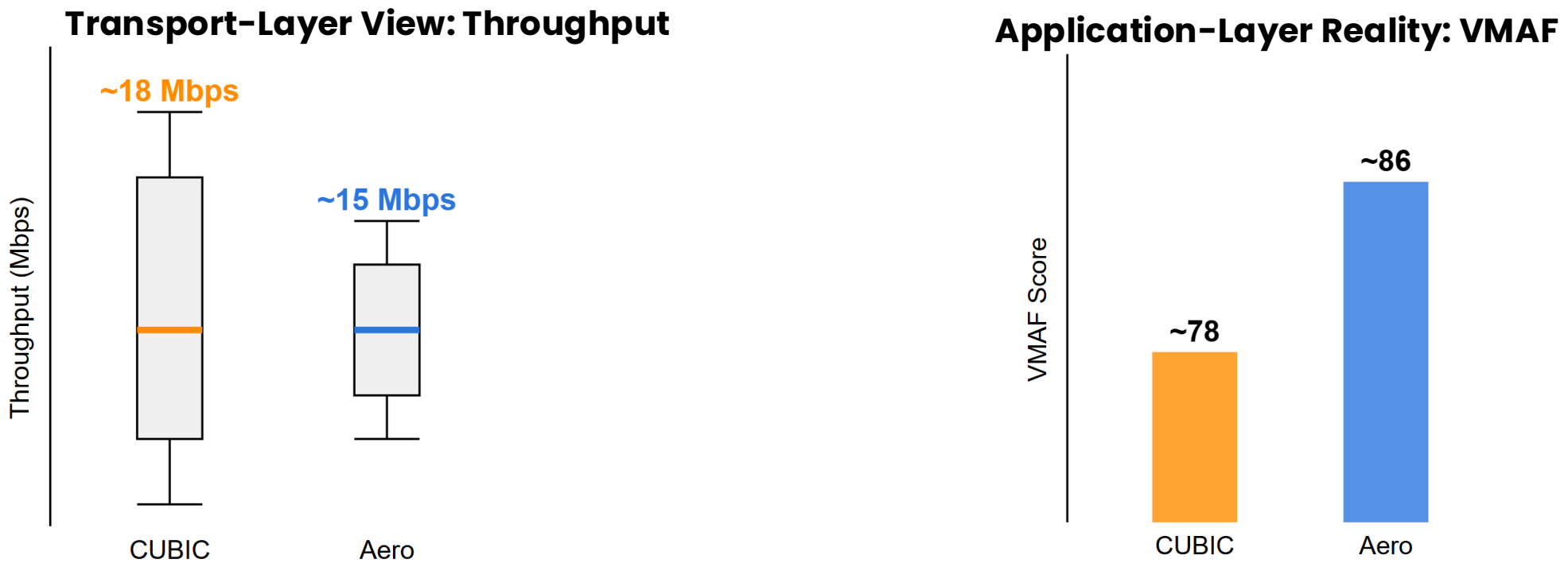
- Tuning existing CCs (BBR, CUBIC, Copa, etc.)
- Loss recovery improvements
- Etc.

Assumption

- These optimizations can help throughput and fairness but don’t always translate to better streaming QoE

Is a transport-centric optimization of QUIC sufficient?

Same Setup (QUIC implementation + ABR + streaming scenario/setup)



CUBIC achieves a higher median throughput but with high variance.

Aero's stability yields a significantly higher and more consistent QoE.

Takeaway: **A transport-centric "win" can be an application-layer "loss".**

Reading: Sidhu, J. S., & Bentaleb, A., (2025). Aero: A Pluggable Congestion Control for QUIC. ACM NOSSDAV. https://doi.org/10.1145/3712678.3721878

Key Insight: Q2

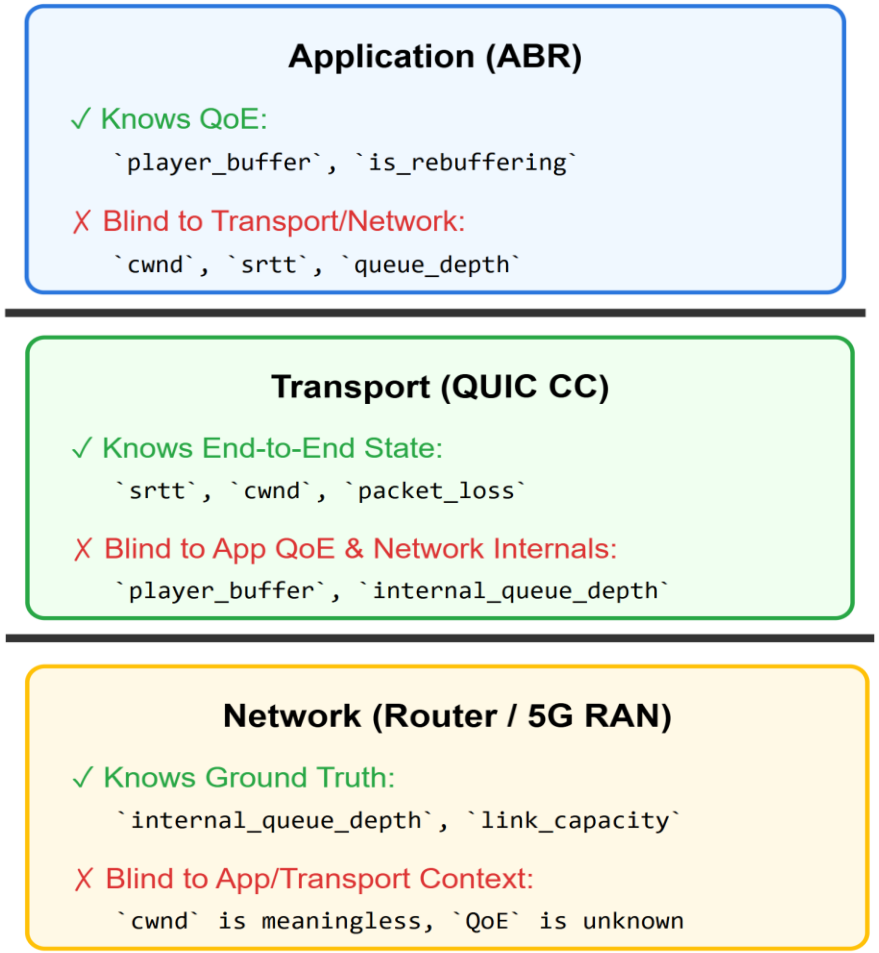
Is a **transport-centric optimization** of QUIC sufficient to solve the QoE challenges in today's diverse networks?

Optimizing QUIC alone does not guarantee better QoE

The Root Cause

Each layer makes local optimizations with incomplete information, leading to global QoE failure.

- **Application (ABR):** Knows the user's QoE but is blind to the underlying network conditions.
- **Transport (QUIC CC):** Sees the end-to-end network state but is blind to both the user's QoE and the internal state of network devices.
- **Network (Router/5G RAN):** Knows the ground truth of its own resources but has zero awareness of the application's goals.

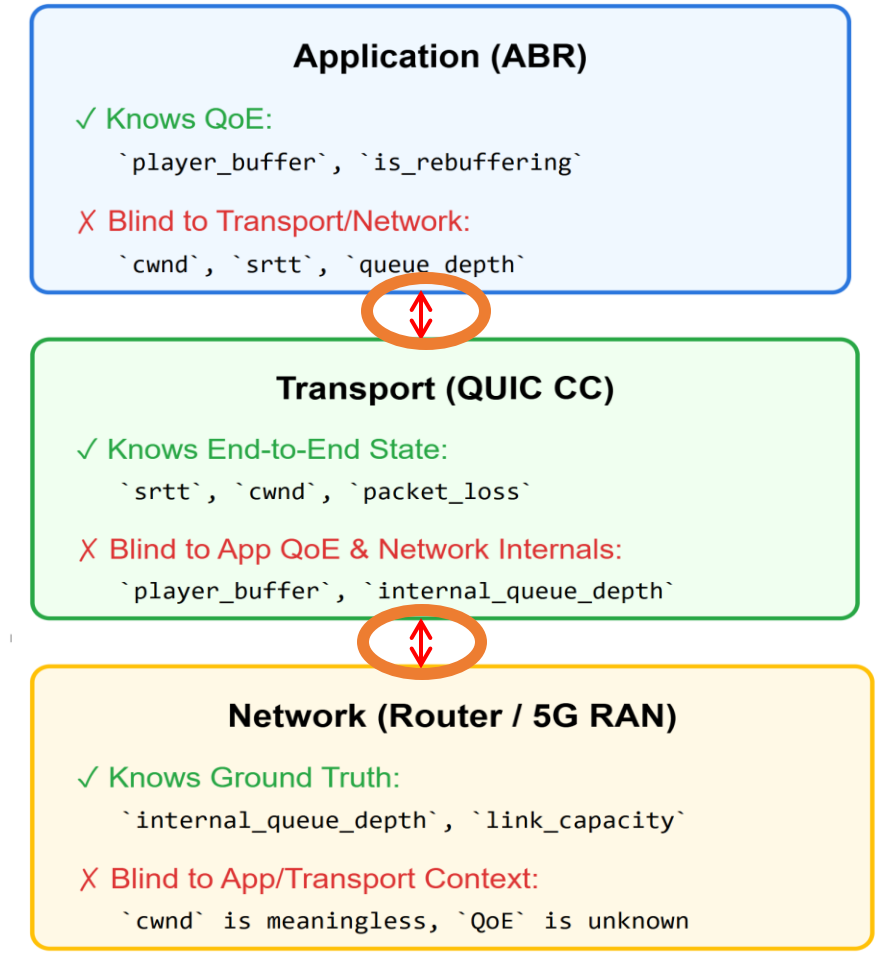


The Vision: A Symbiotic (cooperative) Streaming Stack

Tear Down the Silo Walls

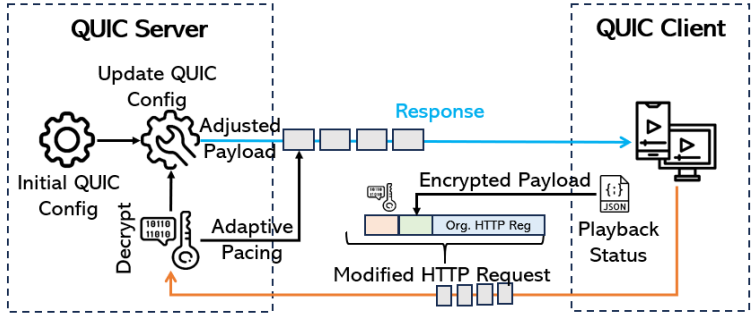
We must create new, explicit communication channels between the layers, guided by three core principles.

- ❑ **Share Application Context (Principle 1):** Create a feedback channel from the App to the Transport
- ❑ **Make Transport QoE-Aware (Principle 2):** Evolve the Transport's logic to optimize for QoE metrics
- ❑ **Leverage Network Signals (Principle 3):** Use explicit signals from the Network to inform the Transport



The Three Principles in Practice

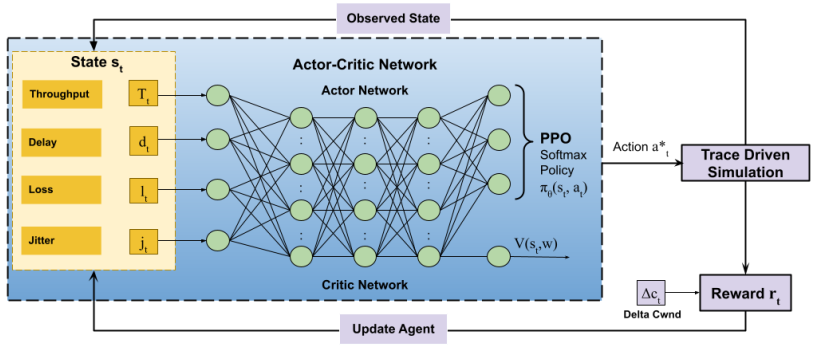
Principle 1



Mechanism: The client encrypts critical playback stats (buffer level, stalls) into QUIC or HTTP headers. The server decrypts this and uses it for real-time, per-user tuning of its transport parameters (e.g., cc, pacing).

Evidence: Increases server throughput by **~33%** compared to only QUIC optimization, enabling higher bitrates and QoE.

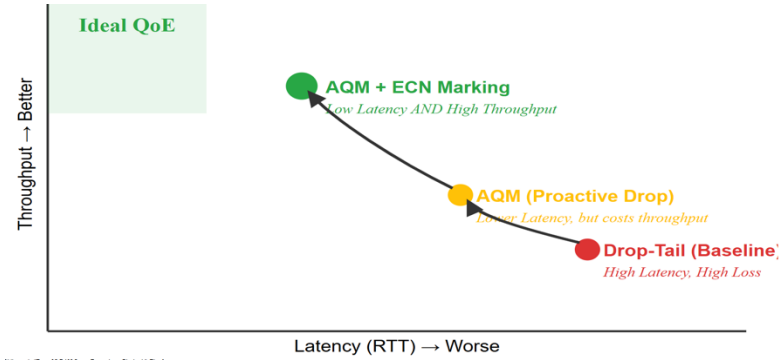
Principle 2



Mechanism: The agent is trained to optimize a reward function based on throughput, delay, loss, and jitter—metrics that directly influence video QoE.

Evidence (LLL): ~82% improvement in VMAF, ~165% increase in delivery rate ~48% lower delay.

Principle 3



Mechanism: Enabling ECN marking with optimizing AQM parameters using cross layer information from transport + application layer

Evidence (LLL): It allows the transport to achieve low latency without the significant throughput penalty of proactive dropping, pushing performance into the ideal QoE zone

Final Notes

🛑 1. Stop Optimizing for the Wrong Metrics

- Transport-level metrics like raw throughput are poor proxies for user QoE.
- A transport-centric “win” can be an application-layer “loss.”

⚠️ Focus on end-user experience, application goals, not just network numbers.

🌉 2. Build Bridges, Not Walls

- The highest QoE gains are achieved by creating **explicit communication channels** between the **Application**, **Transport**, and **Network** layers.

🤝 Enable cross-layer feedback for smarter adaptation.

🧠 3. Embrace Intelligent Control

- Modern streaming complexity demands intelligent agents (DRL, ML) that fuse cross-layer signals to I make holistic decisions .

🚀 Adopt adaptive, learning-based control for sustainable optimization.

Abdelhak Bentaleb, Founder & Director of Intelligent Next Generation Networking and Multimedia (IN2GM) Lab, Concordia University

Thank you!
Special thanks to my students

App Meets Transport Rethinking Video Streaming over QUIC

Abdelhak Bentaleb
abdelhak.bentaleb@concordia.ca
Founder & Director of IN2GM Lab
Concordia University