

TCP vs UDP for Streaming

Which transport every streaming protocol uses, and why.

1. The six-dimension comparison

Dimension	TCP — RFC 9293	UDP — RFC 768
Connection setup	3-way handshake + TLS (1-3 RTTs)	None — 0 RTT
Reliability	Guaranteed via ACK + retransmit	Best-effort — app's problem
Ordering	Strict in-order (HoL blocking)	None — any order
Congestion control	Mandatory (CUBIC, BBR, Reno)	App must implement (RFC 8085)
Flow control	Receive window	None
Header overhead	20+ bytes	8 bytes
Latency floor under loss	1 RTT per lost packet, blocks	0 — next packet plays now

2. Which protocol sits on which transport

TCP — file-shaped, buffered

- HLS — RFC 8216 (2017)
- MPEG-DASH — ISO/IEC 23009-1:2022
- CMAF · LL-HLS · LL-DASH
- RTMP delivery (legacy)
- HTTP · HTTPS · file downloads
- WHIP signalling plane

UDP — real-time, loss-tolerant

- WebRTC media — RFC 8825 family
- SRT — draft-sharabayko-srt-01
- RIST — SMPTE TR-06-1/2/3
- RTP — RFC 3550
- Media over QUIC — draft-ietf-moq
- HTTP/3 over QUIC — RFC 9114

3. Head-of-line blocking — the one fact that decides everything

TCP: lose frame 100 → buffer F101, F102, F103, F104 for ~160 ms → viewer freezes 4-6 frames

Strict in-order delivery is the rule; HoL blocking is the consequence; it cannot be turned off. Why it's fine for OTT: the player's 3-24 s buffer hides the stall. Why it kills conferencing: budget < 200 ms.

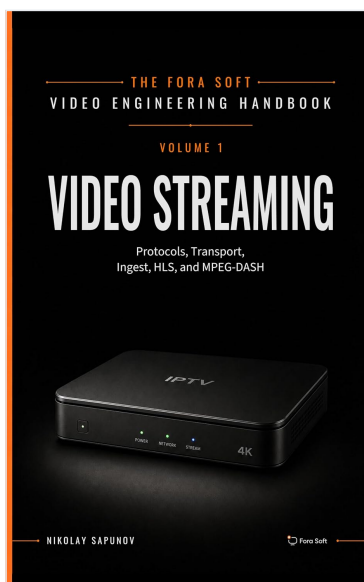
UDP: lose frame 100 → F101 plays immediately → decoder conceals the missing region → no freeze

App-level NACK / RTX / FEC recover what they can inside the latency budget; the rest is concealed. Next I-frame fully resyncs the picture within ~2 s. Glass-to-glass holds at 100-500 ms.

4. The decision rule

Reliability is a latency budget, not a property. Pick the transport that fits the budget:

- Latency budget > 3 s →** TCP via HLS / DASH / LL-HLS. Buffer hides retransmits; CDN caches the segments.
- Latency budget < 1 s →** UDP via WebRTC / SRT / RIST / RTP. App-level recovery beats kernel retransmit.
- New build, 2026 →** Reach for QUIC. Per-stream multiplexing kills connection-wide HoL blocking.



GO DEEPER · VOLUME 2 OF 7

Video Streaming: A Complete Guide to Video Delivery

From Network Protocols and Transport to Ingest, HLS, and MPEG-DASH

By Nikolay Sapunov, CEO at Fora Soft · 1,124 pages · Kindle & paperback on Amazon

Read it on Amazon →

amazon.com/dp/B0HH8F4R82

This sheet tells you which transport each streaming protocol rides on and why head-of-line blocking settles the choice. Volume 2 goes deep on the same ground: TCP versus UDP, congestion control, QUIC and HTTP/3, NAT traversal, and the SRT, RIST and RTMP ingest paths that sit on top of them.

WHAT YOU WILL FIND INSIDE

- TCP vs UDP and head-of-line blocking in practice
- Congestion control on live media paths
- QUIC and HTTP/3 for new streaming builds
- SRT, RIST and RTMP contribution and ingest
- Glass-to-glass latency budgets for live and VOD

THE FULL HANDBOOK · 7 VOLUMES · 9,000+ PAGES · ALL ON AMAZON

Vol. 1 · Video Encoding — pixels and color to H.264, HEVC, AV1, quality metrics and AI

Vol. 2 · Video Streaming I — network transport, contribution and ingest, HLS, MPEG-DASH, CMAF

Vol. 3 · Video Streaming II — ABR, CDNs, players, WebRTC, DRM and the economics of delivery

Vol. 4 · Audio for Video I — PCM, codecs, loudness, accessibility, real-time voice processing

Vol. 5 · Audio for Video II — jitter buffers, lip sync, Atmos, quality metrics, neural codecs

Vol. 6 · AI for Video I — models and inference, computer vision, speech, search, generative video

Vol. 7 · AI for Video II — real-time pipelines, agents, serving, cost, compliance, verticals

Choosing a transport for a live or real-time video product?

Tell us your latency budget and audience; we will map it to a protocol stack (HLS, WebRTC, SRT or QUIC) and estimate delivery cost per viewer-hour. 250+ video products shipped since 2005.

Book a 30-min call →

WhatsApp →

Email us →