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.