

BBR / CUBIC / Copa decision sheet

One-page reference for picking the right TCP/QUIC congestion controller for video streaming.

1. The comparison

Dimension	CUBIC	BBR	Copa
Family	Loss-based	Model-based (BtBw + RTprop)	Delay-based (queueing delay)
Standard	RFC 9438 (Aug 2023, Standards Track)	draft-ietf-ccwg-bbr-05 (Mar 2026, Exp.)	NSDI 2018, MIT — no IETF document
Default in	Linux, Windows, macOS, iOS	Google services, Cloudflare, YouTube	Meta live video upload (mvfst)
On 1 % loss, 50 ms RTT	~3 Mbps cap	~80 % of link rate	~70-80 % of link rate, low delay
On deep-buffer router	Fills the buffer (bufferbloat)	Targets BDP, queue near empty	Targets queue-delay budget
RTT sensitivity	Slow recovery on long RTT	Designed for long RTT	Sensitive to RTT-min jitter
App-tunable knob	None typically	None typically	delta (throughput vs delay)
Fairness with CUBIC	n/a	Approx. fair in v2/v3	Fair only in competitive mode
Best for	LAN, data centre, clean links	Public internet, lossy WAN, satellite	Real-time video upload, RT links
Worst on	Lossy mobile / Wi-Fi (Mathis cap)	Shallow-buffer overshoot	Aggressive loss-based neighbours

2. Mathis cap — worked example

CUBIC: phone on 4G in Seoul, RTT 60 ms, loss 4 %, MSS 1460 B.
 $\text{throughput} \leq \text{MSS} / (\text{RTT} \times \sqrt{\text{loss}})$
 $= 1460 / (0.06 \times 0.2) = 121\,667 \text{ B/s}$
 $\approx 0.97 \text{ Mbps}$ — cannot serve 6 Mbps 1080p.

BBR: same link, same loss. BBR treats packet loss as data, not a stop signal, and paces to its measured bottleneck bandwidth.
 Measured throughput: $\approx 18\text{-}22 \text{ Mbps}$.
 Same physical link, +18x over CUBIC.

3. The Linux recipe (BBRv3 on kernel 6.x)

```
# Switch the algorithm
sysctl -w net.ipv4.tcp_congestion_control=bbr
# Pair with fair-queue qdisc (required, BBR's pacing depends on it)
sysctl -w net.core.default_qdisc=fq
# Persist in /etc/sysctl.conf so it survives reboots
echo 'net.ipv4.tcp_congestion_control=bbr' >> /etc/sysctl.conf
echo 'net.core.default_qdisc=fq' >> /etc/sysctl.conf
```

4. Three-step rollout test (do not skip)

1. Pick a non-customer-facing edge node. Switch the TCP controller (and Linux qdisc) on it alone. Verify with **sysctl net.ipv4.tcp_congestion_control**.
2. Drive traffic from three representative paths: a wired client (baseline), a 4G/5G mobile (lossy), and a transcontinental client (high-RTT). Capture throughput p50/p90/p99, rebuffer rate, time-to-first-frame, for ≥ 24 hours.
3. Compare against a matched CUBIC cohort. $\geq 5\,000$ sessions per arm to trust a 5 % delta. Roll out only after the comparison is clean. Re-evaluate after every BBR draft revision.