

BBR / CUBIC / Сора — справочный лист

Страничный референс по выбору TCP/QUIC congestion controller для видео-стриминга.

1. Сравнение

Параметр	CUBIC	BBR	Сора
Семейство	Loss-based	Model-based (BtlBw + RTprop)	Delay-based (queueing delay)
Стандарт	RFC 9438 (авг. 2023, Std Track)	draft-ietf-ccwg-bbr-05 (март 2026, Exp)	NSDI 2018, MIT — без IETF
Дефолт в	Linux, Windows, macOS, iOS	Google services, Cloudflare, YouTube	Meta live upload (mvfst)
При 1 % loss, 50 мс RTT	~3 Mbps кэп	~80 % линка	~70-80 % линка, низкая задержка
На глубоком буфере роутера	Наполняет буфер (bufferbloat)	Таргет BDP, очередь почти пустая	Таргет queueing-delay
Чувствительность к RTT	Медленнее на длинном RTT	Спроектирован под длинный RTT	Чувствителен к jitter RTT-min
Настраиваемая ручка	Обычно нет	Обычно нет	delta (полоса vs задержка)
Справедливость к CUBIC	n/a	Примерно честная в v2/v3	Только в competitive mode
Лучше всего для	LAN, ДЦ, чистые линки	Публичный интернет, lossy WAN, спутник	Real-time видео-upload, RT-линки
Хуже всего на	Lossy mobile / Wi-Fi (Mathis cap)	Очереди с мелким буфером (overshoot)	Агрессивные loss-based соседи

2. Кэп Mathis — разобранный пример

CUBIC: телефон на 4G в Сеуле, RTT 60 мс, loss 4 %, MSS 1460 байт.
 $\text{throughput} \leq \text{MSS} / (\text{RTT} \times \sqrt{\text{loss}})$
 $= 1460 / (0.06 \times 0.2) = 121\,667 \text{ байт/с}$
 $\approx 0.97 \text{ Mbps} — 6 \text{ Mbps } 1080p \text{ не пролезет.}$

BBR: тот же линк, та же доля потерь. BBR трактует потерю как данные, а не как стоп, и расе'ит к измеренной пропускной способности.
 Измерено в поле: $\approx 18\text{-}22 \text{ Mbps}$.
 Тот же физический линк, +18x к CUBIC.

3. Linux-рецепт (BBRv3 на ядре 6.x)

```
# Переключаем алгоритм
sysctl -w net.ipv4.tcp_congestion_control=bbr
# Включаем fair-queue qdisc (обязательно — racing BBR опирается на него)
sysctl -w net.core.default_qdisc=fq
# Сохраняем в /etc/sysctl.conf, чтобы пережить ребут
echo 'net.ipv4.tcp_congestion_control=bbr' >> /etc/sysctl.conf
echo 'net.core.default_qdisc=fq' >> /etc/sysctl.conf
```

4. Трёхшаговый тест перед раскаткой

1. Возьмите edge-ноду, не отдающую пользовательский трафик. Переключите TCP controller (и qdisc на Linux) только на ней. Проверьте: **sysctl net.ipv4.tcp_congestion_control**.
2. Направьте трафик с трёх представительных путей: проводной клиент (baseline), 4G/5G (lossy) и трансконтинентальный (high-RTT). Снимайте throughput p50/p90/p99, rebuffer rate, time-to-first-frame ≥ 24 часа.
3. Сравните с матчейной CUBIC-когортой. $\geq 5\,000$ сессий на ветку, чтобы доверять 5 % дельте. Раскатывайте только когда сравнение чистое. Перепроверяйте после каждой ревизии BBR draft.