

Шпаргалка по пайплайну стриминга

Пять задач, протоколы на стыках, бюджеты задержки

Пайплайн по стадиям

1. Захват	Камера или экран — сырые пиксели и звук	5-20 ms
2. Кодирование	Кодек сжимает (H.264 / H.265 / AV1)	50-500 ms
3. Упаковка	Нарезка на сегменты и манифест (CMAF / fMP4)	200 ms - 4 s
4. CDN	Origin → shield → edge, кэш ближе к зрителю	5-80 ms
5. Плеер	Буфер, декод, отрисовка (hls.js / Shaka / native)	0,5 - 30 s

Семейства протоколов на стыках

- Контрибьюшн (ingest)**
RTMP, SRT, RIST, WHIP, Zixi — стадии 1-3
- Дистрибуция (delivery)**
HLS, LL-HLS, DASH, CMAF, WebRTC, WHEP, HESP, MoQ — стадии 3-5
- Адаптивный битрейт (ABR)**
Плеер выбирает ступень лестницы — основа стабильного качества

Типичный glass-to-glass latency по стекам

Классический HLS / DASH	20-30 s
Low-Latency HLS (LL-HLS)	2-5 s
Low-Latency DASH (CMAF chunked)	2-5 s
HESP	0.4-1 s
WebRTC доставка	0.2-0.5 s
Media over QUIC (MoQ)	0.3-1 s

Где каждая стадия чаще всего ломается первой

- Захват — отвалился кабель, тихий микрофон, не тот формат сигнала
- Кодирование — CPU энкодера упёрся, выбран слишком высокий битрейт
- Упаковка — несогласованная длина сегмента, слишком долгий TTL манифеста
- CDN — промахи кэша, лавина запросов на «горячий» сегмент, всплески fill
- Плеер — underrun буфера на слабом Wi-Fi, осцилляция ABR, нет поддержки кодека

Правило: буфер плеера доминирует в общей задержке. Его уменьшение — самый сильный рычаг для низкой задержки и самый быстрый путь к подвисаниям, если ошибиться в оценке сети.