

WebRTC Bandwidth-Estimation Tuning Checklist

30 items to verify before shipping a WebRTC product to a network you don't control

1. The control signal

- transport-wide-cc-02 RTP header extension is negotiated in SDP on every media section (audio and video).
- Receiver emits Transport-Wide CC Feedback every ≈ 50 ms; verify via Wireshark on a sample call.
- REMB feedback is supported as a fallback path for legacy receivers.
- Plan a migration window for RFC 8888 CCFB once libwebrtc ships it as default.

2. The encoder side

- min-bitrate set to a value that matches the product's quality floor, not to zero.
- max-bitrate set to a sane ceiling (1500-2500 kbps for 720p conferencing). Raising it past genuine product need invites trouble.
- Simulcast is publishing 3 layers (720p + 360p + 180p) so the SFU has room to switch.
- VP8/VP9/H.264 encoder rate-control mode is set for low-latency (CBR or capped-CRF with low buffer).
- Pacer is enabled (libwebrtc default); confirm no custom code bypasses it.

3. Probing and start-up

- Initial probing bursts are enabled in the publishing SDK.
- Initial target bitrate raised above the 300 kbps default if call traffic justifies it.
- On reconnects, the SDK does not start from zero — it resumes from a recent target.
- Verify start-up time: target should hit 80% of ceiling within 5 seconds of a clean call.

4. SFU downlink

- Per-subscriber BWE is running on every SFU subscriber's downlink, not just the publisher.
- Simulcast layer selection logic is verified on slow subscribers (force a 500 kbps cap, watch the layer chosen).
- If subscribers are heterogeneous, consider SVC over simulcast for finer granularity.
- TURN relay paths do not break the feedback direction (verify with a TURN-forced ICE config).

5. Failure-mode tests

- Simulate Wi-Fi → cellular hand-off (toss the phone on cellular mid-call). Freeze ≤ 2 s.
- Simulate bufferbloat (use a tc qdisc with 500 ms FIFO). Loop should drop target, not freeze audio.
- Simulate parallel TCP download (run iperf on the same uplink). Document the quality floor.
- Simulate radio loss (5% random drop). Loss-based estimator should halve, not collapse.
- Verify recovery: after each failure, target should rebuild within 10 s of clean conditions.

6. Observability

- getStats() sampled every 1 s, stored per-session for at least 30 days.
- targetBitrate, fractionLost, roundTripTime, availableOutgoingBitrate all in dashboards.
- Alert on targetBitrate hovering below 30% of ceiling for ≥ 30 s in a session.
- Alert on fractionLost $> 5\%$ sustained for ≥ 10 s.
- Per-session post-mortems wired into support escalation.