NetEQ - Tuning & Diagnostics Cheat Sheet

Fora Soft Learn

WebRTC's adaptive audio jitter buffer: the operations, the delay-manager knobs, and the getStats fields to watch.

The six operations (one per 10 ms tick)

Normal	Packet on time	Decode & play at natural speed
Acceleration	Buffer too full	Speed up slightly (no pitch change)
Preemptive expand	Buffer too low	Slow down slightly to buy time
Expand (PLC)	Packet missing	Manufacture audio - the last resort
Merge	Real packet returns	Stitch concealed audio to real, no click
Comfort noise	DTX silence	Faint matched hiss, not dead air

Delay manager - how it sizes the buffer

Relative delay (2022+)	Each packet vs the FASTEST packet in the window
History window = 2 s	Separates temporary jitter from a permanent shift
Forget factor = 0.983	How fast the delay histogram forgets old values
Target percentile	e.g. 95th: higher = safer + larger + more delay
Underrun histogram	Captures low-frequency delay events
Reorder optimizer	Cost function for late / retransmitted packets

getStats fields to watch

jitterBufferDelay / EmittedCount	Average ms each sample waited - the headline number
jitterBufferTargetDelay	Network-driven target, before any AV-sync padding
concealmentEvents	How often the buffer ran dry and faked audio
concealedSamples	Volume of faked audio (should be a tiny fraction)
insertedSamplesForDeceleration	Audio added by slowing down (underrun pressure)
removedSamplesForAcceleration	Audio dropped by speeding up

Worked example: histogram -> target

Relative-delay histogram over 2 s: 20ms 60%, 40ms 20%, 60ms 10%, 80ms 6%, 100ms 4%.
Cumulative: 60% -> 80% -> 90% -> 96% -> 100%. First bucket past 0.95 is the 80 ms bucket.
Target delay = 80 ms: 96% of packets arrive in time; the worst 4% trigger an Expand.
Raise percentile to 0.99 -> target 100 ms: safer, but +20 ms delay on every word.

Diagnosis rules

High concealmentEvents usually = real packet loss UPSTREAM - fix the network, not the buffer.
jitterBufferDelay high but network fine? Check AV-sync padding (look at jitterBufferTargetDelay).
Insert + remove counters both climbing = an unstable, hunting buffer (noisy network estimate).
Watch it live at chrome://webrtc-internals; replay a captured call offline with neteq_rtpplay.

Remember

- [ ] Jitter = uneven arrival of evenly sent packets; the jitter buffer smooths it back out.
- [ ] NetEQ is adaptive: it resizes itself to match the live network, trading delay against running dry.
- [ ] Every 10 ms it outputs ONE of: Normal, Acceleration, Preemptive expand, Expand, Merge, Comfort noise.
- [ ] Target delay = a high percentile of a forgetting histogram of relative delays.
- [ ] Measure before you tune - the right fix is almost always upstream of NetEQ.