

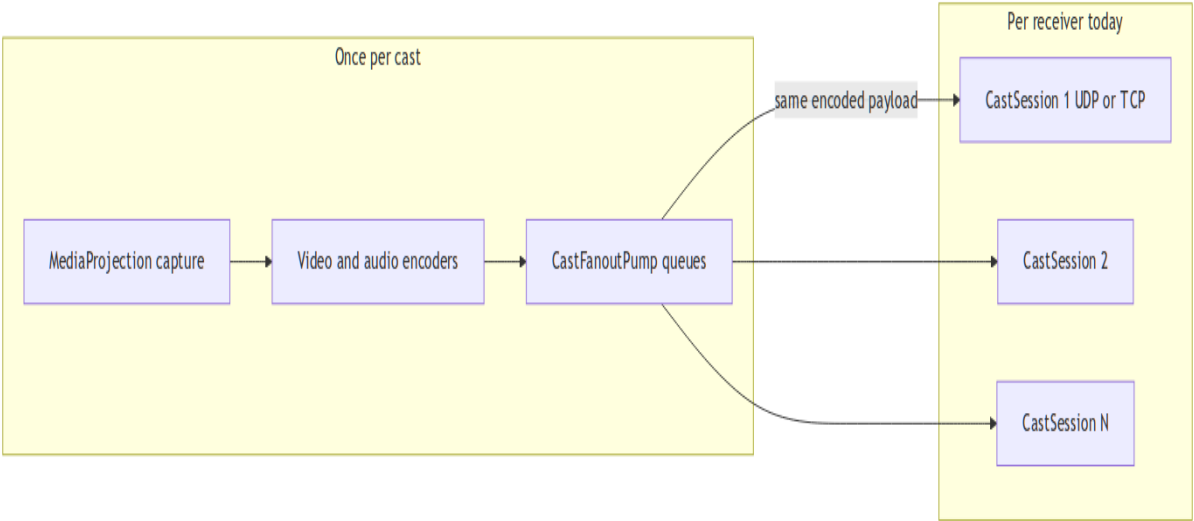
Android Cast — Next-gen egress review

Summary of sender 1:N behavior, bandwidth model, limitations, and feasibility of single-stream multicast plus multi-tier (full / 720p / 480p) renditions. Generated 2026-05-21.

1. Current sender behavior

- How many receivers?** Up to 5 when MULTI_RECEIVER_ENABLED (CastConfig.MAX_RECEIVERS). Sender opens one CastSession per selected device; failed handshakes are skipped if at least one peer connects.
- CPU / GPU / memory:** Capture and encode run once per cast. CastFanoutPump duplicates each encoded frame to every session. Network send and per-peer UDP/FEC state scale with N; encode cost does not.
- Bandwidth:** Per-receiver unicast duplicate of the same bitstream — not IP multicast. Rough LAN egress $\approx$ (video + ~96 kbps audio) $\times$ N, plus per-peer protocol overhead.

Current sender path (1 encode, N unicast sends)



1.1 Current limitations (sender-focused)

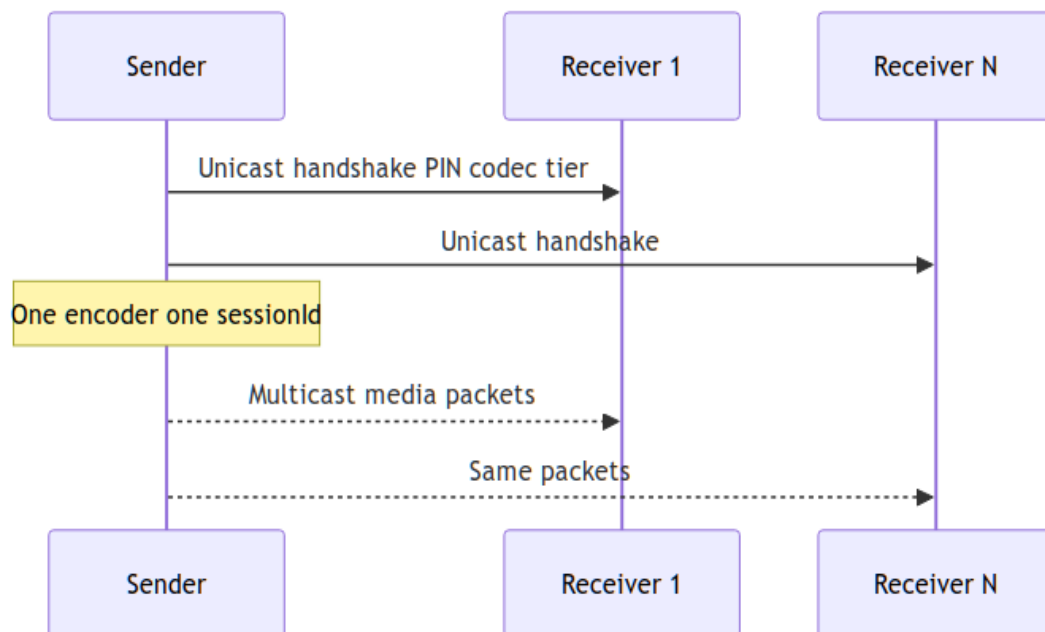
Area	Limitation
Primary peer	First connected peer sets negotiated MIME, adaptive tuning, and network feedback.
Codec	Single encoder; all peers must tolerate the same bitstream.
Send path	One CastFanoutPump thread; serial fan-out to N sessions.
Wi-Fi	Uplink often saturates before CPU on 3–5 receivers.
Topology	Sender dials out; no shared LAN broadcast stream today.

2. Proposal: single outgoing stream (multicast)

Goal: One encoded stream on the air; only handshake-authorized receivers consume it; save $\sim(N-1) \times$ bitrate vs today's unicast fan-out.

Feasible with UDP IP multicast + unicast handshake (group address, port, sessionId, PIN). Handshake-only targeting without multicast does **not** reduce bandwidth. TCP/QUIC cannot deliver true $1 \times$ Wi-Fi²Fi² airtime.

Proposed multicast setup (handshake then one media flow)



2.1 Approach comparison

Approach	1× sender airtime?	Notes
Unicast fan-out (today)	No	Same encode; $N \times \text{send}()$.
UDP multicast	Yes on good LAN	MulticastLock; AP client isolation risk.
Sender server, N TCP clients	No	Flip connect direction; still N wire copies.
LAN relay receiver	Sender 1×	Relay pays duplicate bandwidth.

Protocol notes: CastFramingContext.sessionId and CastSessionGate already support session binding. FEC/NACK must become group-scoped (one retransmit cache, resend to multicast). Existing code paths: MultiCastCoordinator, CastFanoutPump, UdpCastTransport.

3. Proposal: multi-tier streams (e.g. full + 720p + 480p)

Goal: Several encoded renditions from one capture; each receiver picks a tier at handshake (e.g. full quality vs 480p for bandwidth-limited devices).

Feasible via multiple encoders fed from one capture (EGL / scaled Surfaces into separate MediaCodec instances). Cost scales with **active** tiers (~2–3 HW encoders typical phone limit; 3× software VP9 is heavy). Audio usually stays one AAC stream.

Bandwidth: Unicast per tier → egress ≈ sum of each peer's tier bitrate. Multicast per tier → one group per tier; airtime ≈ sum of tiers with ≥1 subscriber.

3.1 Tier architecture options

Option	Verdict
Multiple HW encoders from one capture	Recommended v1.
One encode + CPU re-encode	Too expensive for live cast.
SVC / simulcast in one bitstream	Poor Android HW support; defer.
Receiver-only downscale	Saves sender encode, not downlink bandwidth.

4. Recommended rollout

Phase	Deliverable	Benefit
A — Tier selection (unicast)	Handshake tier; 720p + 480p encoders; send chosen tier per device	Per-device quality without multicast; no AP surprises
B — Multicast one tier	UDP multicast publish after unicast handshake	~(N-1) egress for same-quality 1:N
C — Multicast per tier	Separate multicast group per tier; lazy encoder start	Combines (1) and (2) on good LANs

4.1 Summary judgment

Idea	Feasible?	Main win	Main cost
(1) Multicast + handshake gate	Yes (UDP)	~(N-1) uplink on good LAN	Transport refactor; FEC/NACK; AP testing
(1) Unicast only	Trivial	Auth only	No bandwidth win
(2) Multi-tier encode	Yes	Per-receiver quality	2-3× encode resources
(1) + (2) combined	Yes	Multicast per tier	Highest complexity

Key code paths today: CastConfig.MAX_RECEIVERS, MultiCastCoordinator, CastFanoutPump, ScreenCastService (primary peer), UdpCastTransport, CastFramingContext, CastSessionGate.