

Opus / Speex / stream protection — validation checklist (task 3)

From OPUS_SPEEX_VALIDATION.md

Table of contents

[Prerequisites](#)

[Automated baseline \(CI / dev machine\)](#)

[User settings \(both peers\)](#)

[Developer settings](#)

[E2E matrix — run on two devices](#)

[A. Handshake and negotiation](#)

[B. Audio codec](#)

[C. Controls \(user vs developer\)](#)

[D. Stress \(optional\)](#)

[What to capture per run](#)

[Pass criteria](#)

[Known limits \(next\)](#)

Source: OPUS_SPEEX_VALIDATION.md — section links work in PDF viewers that support internal anchors.

E2E validation guide for **audio codecs** and **UDP stream protection** (FEC, NACK, negotiation). Code is on next; this doc is the operator checklist.

See also: ndk/README.md, ROADMAP.md, AV_QUALITY_QA_SESSION.md.

Prerequisites

- Two Android devices (or sender + receiver) on **same LAN** for first pass; repeat subset on lossy/WAN if possible.
- Debug builds with NDK codecs built if testing native Opus/Speex probes:

```
./scripts/build-native-codecs.sh arm64-v8a # optional; see ndk/README.md
./gradlew :app:assembleDebug
```

Automated baseline (CI / dev machine)

```
Run before field soak:
./gradlew :app:testDebugUnitTest --tests 'com.foxx.androidcast.network.transport.StreamProtectionNegotiatorTest'
./gradlew :app:testDebugUnitTest --tests 'com.foxx.androidcast.media.AudioNegotiatorTest'
./gradlew :app:testDebugUnitTest --tests 'com.foxx.androidcast.media.codec.*'
```

Or:
bash scripts/validate_opus_speex.sh
Expect **BUILD SUCCESSFUL** and all tests green.

User settings (both peers)

Global settings (main UI):

Control	Location	Notes
Transport	Sender / receiver	Use UDP for FEC/NACK tests
Stream protection (UDP)	Global settings	Default None ; try FEC 3/4, NACK, FEC+NACK
Audio codec	Global settings	AUTO, Opus, Speex , AAC baseline

After changing protection, **start a new session** — negotiation runs at handshake (CastSession / StreamProtectionNegotiator).

Developer settings

Enable **Developer options** on both devices if testing passthrough / debug audio:

Control	Purpose
Passthrough / debug audio	PASSTHROUGH_DEBUG path
Codec overrides	Align with CodecCatalog / PassthroughCodecPolicy
Diagnostics export	Session stats + negotiated mime + protection string

Check **session diagnostics** after connect: protection line should match PassthroughCodecPolicy.protectionDescription().

E2E matrix — run on two devices

Mark each cell **PASS / FAIL / SKIP** in a ticket tagged 20260604.

A. Handshake and negotiation

#	Sender protection	Receiver capability	Audio	Expected
A1	None	Any	AUTO	No FEC/NACK; stable video+audio
A2	FEC 3/4	FEC 3/4	AUTO	Agreed FEC 3/4; loss recovery under artificial drop
A3	NACK	NACK	AUTO	Retransmit under single-packet loss
A4	FEC+NACK	FEC+NACK	AUTO	Combined behavior; no deadlock
A5	FEC 3/4	NONE only	AUTO	Prompt or downgrade per StreamProtectionNegotiator

B. Audio codec

#	Sender audio	Receiver audio	Expected
B1	AUTO	AUTO	AAC or negotiated fallback

B2	Opus	Opus	Opus native if .so present; else stub/fallback per <code>AudioNegotiator</code>
B3	Speex	Speex	Speex native if built; else fallback
B4	Opus	AUTO (no Opus)	Fallback — no silent failure

C. Controls (user vs developer)

#	Check
C1	User can change transport UDP ↔ TCP without crash
C2	User protection change applies on next session
C3	Developer diagnostics show negotiated video mime + audio + protection
C4	Receiver UI shows aligned protection label after handshake

D. Stress (optional)

#	Check
D1	Background/rotate sender — session recovers or fails cleanly
D2	~5 min soak with FEC on — no memory runaway (logcat)
D3	Switch Wi-Fi ↔ mobile data mid-cast — defined behavior (fail or reconnect)

What to capture per run

1. **Diagnostics** export from both devices (or logcat filter `androidcast|CastSession|StreamProtection|AudioNegotiator`).
2. Negotiated lines: `negotiatedVideoMime`, `audioCodec`, `streamProtection`, `wantsNack`.
3. Subjective: audio dropouts, video freezes, recovery after packet loss (FEC/NACK test: weak Wi-Fi or `tc netem` on Linux AP if available).

Pass criteria

- All **automated** tests green (`validate_opus_speex.sh`).
- Matrix **A1–A4** and **B1–B2** pass on reference hardware.
- No unexplained **silent** fallback (user sees toast/prompt when negotiation downgrades).
- Diagnostics string matches actual wire behavior for UDP protection.

Known limits (next)

Item	Status
Opus / Speex native encode	Probe + bridge; full native path per <code>ndk/README.md</code>
FEC 3/4 + RS	Per-shard wire (<code>FecShardWire</code>); v1 monolithic still accepted
TCP transport	Protection spinners disabled / N/A for TCP

File issues under Tickets → tag 20260604, component Opus/Speex.