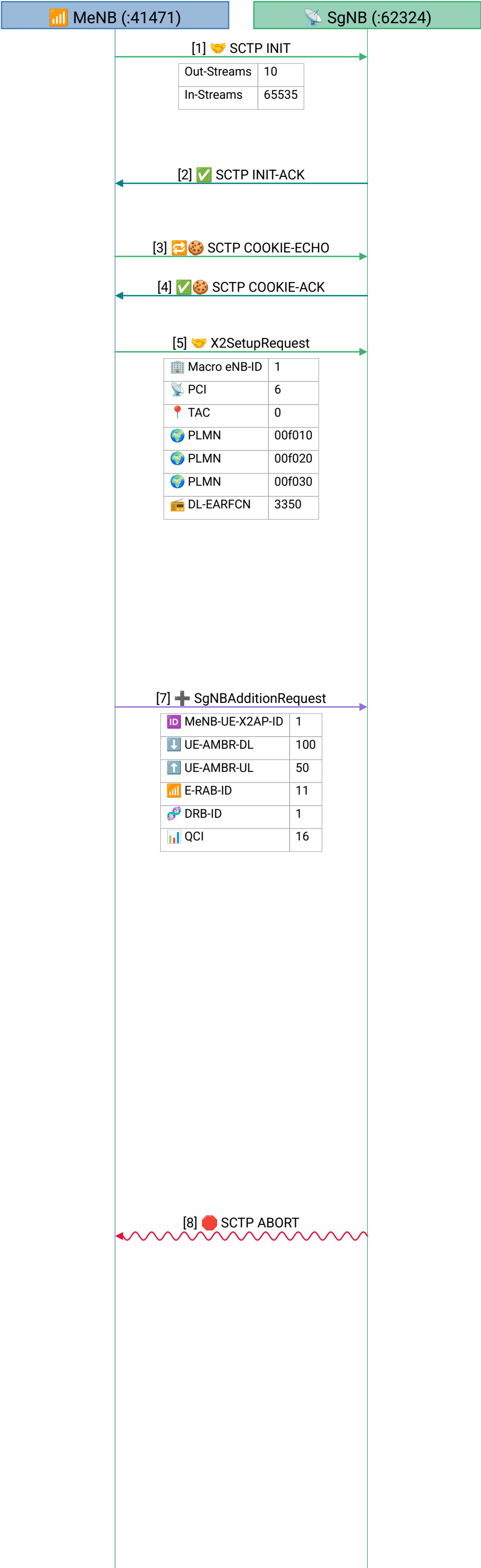




[Frame 1] **SCTP INIT — the transport comes up first.** Before any X2AP can flow, the two nodes build an SCTP association (X2AP is carried over SCTP, not TCP). The initiator on port **41471** — the master eNB (MeNB) side — offers **10 outbound streams** and accepts up to 65535 inbound. X2AP uses separate SCTP streams to keep non-UE-associated signalling (like the X2 Setup in frame 5) off the same stream as UE-associated procedures (the SgNB Addition in frame 7). Watch the 41471 <-> 62324 port pair: it is what distinguishes MeNB from the secondary gNB throughout this loopback capture.

[Frame 2] **INIT-ACK.** The responder on port **62324** — the secondary gNB (SgNB) side — accepts and returns its own parameters plus a state cookie. No association state is committed yet; that is deliberate SCTP anti-DoS design (the cookie carries the state so the responder holds nothing until frame 3 echoes it back).

[Frame 3] **COOKIE-ECHO.** The MeNB echoes the cookie back, proving the round-trip. This is the third leg of SCTP's 4-way handshake.

[Frame 4] **COOKIE-ACK — association established.** The gNB acknowledges; the SCTP association is now open in both directions and ready to carry X2AP PDUs. The next two frames are the actual X2 application signalling.

[Frame 5] **X2 Setup Request — bring the X2 interface up (non-UE-associated).** The MeNB introduces itself so the peer knows who it is and what cells it serves, before any per-UE work. Verified contents of this PDU:

IE	Value	Meaning
Global eNB-ID	1	macro eNB-ID (20-bit), decimal 1
Served cells	1	one E-UTRA cell advertised
PCI	6	physical cell ID
TAC	0	tracking area code
Broadcast PLMNs	001-01, 001-02, 001-03	the three PLMNs the cell broadcasts (00f0x0 encoding)
UL/DL EARFCN	3350	carrier frequency (same up/down => FDD-style pairing in the test data)

Normally the peer answers with an **X2SetupResponse** and only then does UE-associated signalling begin. Here the MeNB does not wait — it fires the SgNB Addition in frame 7 immediately, and no response ever arrives (see frame 8). That "send everything, don't wait" pattern is another tell that this is a message-builder trace rather than a real gNB dialogue.

[Frame 7] **SgNB Addition Request — the point of this capture (EN-DC bearer setup, "with DRB-ID")**. This is X2AP procedure **id-sgNBAdditionPreparation (27)**: the MeNB (LTE anchor) asks the secondary gNB to allocate **NR** radio resources so the UE can run in dual connectivity (EN-DC / NSA Option 3). The MeNB owns the UE here — note the identifier is the **MeNB-UE-X2AP-ID = 1** (the gNB will mint its own SgNB-UE-X2AP-ID in the Acknowledge that never comes).

The bearer being added, decoded from the E-RABs-ToBeAdded list (exactly one item):

- E-RAB-ID 11**, mapped to **DRB-ID 1** — the explicit Data Radio Bearer identity this capture is named for. Carrying the DRB-ID in the request lets the MeNB pin the NR DRB identity rather than leaving it to the SgNB.
- en-DC-ResourceConfiguration: pDCPatSgNB, mCGresources and sCGresources all *present* — i.e. this bearer uses **both** Master and Secondary Cell Group lower-layer resources (a split-bearer shape).
- resource-configuration: sgNBPDcpnotpresent — the PDCP anchor stays at the **MeNB**; the SgNB contributes NR RLC/MAC/PHY only. So this is an **MCG-anchored split bearer**, not an SCG bearer with SgNB-terminated PDCP.
- Requested SCG E-RAB QoS**: QCI 16, ARP priority = highest (1), shall-not-trigger-pre-emption, not-pre-emptable.
- meNB-UL-GTP-TEIDatPDcp: the MeNB's uplink GTP-U endpoint for the split bearer — here **0.0.0.0 / TEID 0x00000000** (placeholder), and r1c-Mode is Unknown . Real values would point the SgNB's uplink X2-U traffic back at the MeNB's PDcp.

Security context IEs are present but carry **placeholder** patterns: the SgNBSecurityKey (S-KgNB) is **all zeros (256-bit)**, and the NR UE security capabilities advertise **no** standard algorithms (nRencryptionAlgorithms=0x0001, nRintegrityProtectionAlgorithms=0x0002 — the 128-NEA1/2/3 and 128-NIA1/2/3 bits all decode as *Not supported*). An opaque **MeNBtoSgNBContainer** (the RRC CG-ConfigInfo) is also included as a short stub (f0f1f2...) . In a live add the gNB would use S-KgNB and these caps to configure NR security and reply with a **SgNBAdditionRequestAcknowledge** carrying the SCG RRCReconfiguration.

[Frame 8] **SCTP ABORT — the association is torn down before the gNB replies.** Instead of an X2AP answer, the gNB-side endpoint (port 62324) sends an SCTP ABORT with cause **"User initiated ABORT" (0x000c)**. That is important to read correctly: it is a *graceful, application-driven socket teardown*, **not** an X2AP SgNBAdditionRequestReject and not a protocol error. The test tool simply closed the association after emitting its request PDUs. Consequently **neither** procedure completes on the wire — the X2 Setup (frame 5) gets no Response and the SgNB Addition (frame 7) gets no Acknowledge/Reject — which is exactly why the session view marks both X2AP sessions incomplete::no-close and the SCTP association failure::aborted.