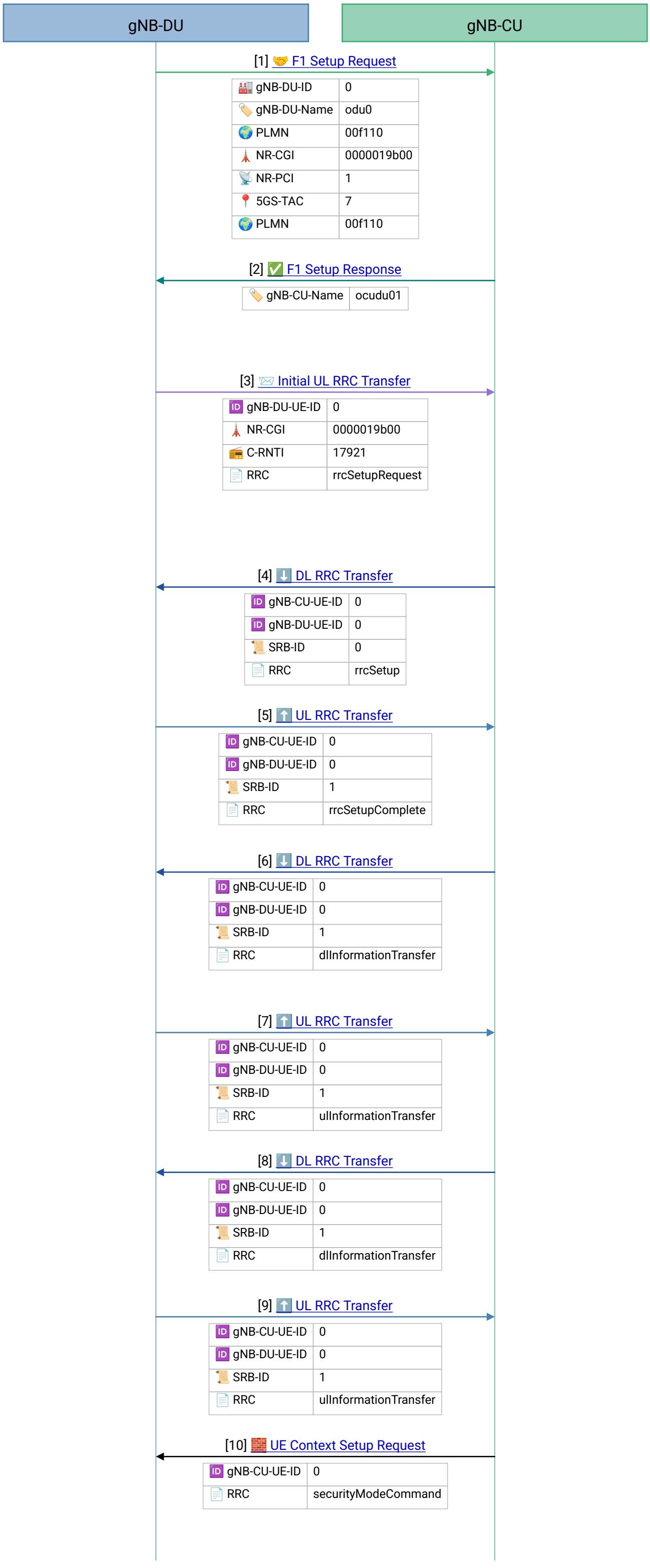


[Frame 1]

Phase 1 — F1 interface bring-up

F1 Setup Request (gNB-DU `odu0` -> gNB-CU). Before any UE can be served, the DU must register with the CU. The DU advertises its identity (`gNB-DU-ID` 0) and its full served-cell configuration: PLMN `00f110`, cell `NR-CGI` `0000019b00`, `PCI` `1`, `5GS-TAC` `7`, plus the frequency, SSB and system-information layout. No UE context exists yet — this is pure interface management, and every UE-associated procedure below depends on it succeeding.

[Frame 2] F1 Setup Response (gNB-CU `ocudu01` -> gNB-DU). The CU accepts the association and returns its name plus the Cells-to-be-Activated list (the served cells it authorises the DU to switch on). F1 is now up. Everything from here to frame 30 rides on this single association; it is torn down at the very end (frames 31-32).

[Frame 3]

Phase 2 — RRC connection setup

Initial UL RRC Transfer (DU -> CU), carrying the UE's **RRCSetupRequest.** The UE sent this on the radio CCCH; the DU received it, allocated a `gNB-DU-UE-F1AP-ID` (0) and a `C-RNTI` (17921), and forwards it — with the serving `NR-CGI` — up to the CU. This is the first UE-associated message and it opens the UE's context on the CU side. Note the DU UE id: it is the one identifier present on **every** UE message through release, which is why the per-UE session keys on it.

[Frame 4] DL RRC Transfer (CU -> DU), carrying **RRCSetup,** on `SRB0`. The CU admits the UE and sends the RRCSetup that moves it off the CCCH and establishes signalling radio bearer `SRB1`. From the next message onward all RRC travels on SRB1.

[Frame 5] UL RRC Transfer (DU -> CU), carrying **RRCSetupComplete** on `SRB1` — and piggybacked inside it, the UE's first NAS message: a **5GMM Registration Request** (NAS `0x41`, still sent in the clear). This is the moment the RRC connection becomes useful: it delivers the UE's registration attempt to the AMF behind the CU. Everything in Phase 3-4 below is the AMF authenticating and securing that registration, relayed transparently over F1.

[Frame 6]

Phase 3 — NAS authentication (relayed as DL/UL Information Transfer)

DL RRC Transfer (CU -> DU): RRC **DL Information Transfer** carrying a NAS **Authentication Request** (`0x56`, plaintext). The AMF challenges the UE. The CU is just a NAS pipe here — it wraps the AMF's NAS PDU in RRC and hands it to the DU for radio delivery.

[Frame 7] UL RRC Transfer (DU -> CU): RRC **UL Information Transfer** carrying the NAS **Authentication Response** (`0x57`, plaintext). The UE answers the challenge. Authentication succeeds, which is what lets the network derive keys and turn on security next.

[Frame 8]

Phase 4 — NAS security activation

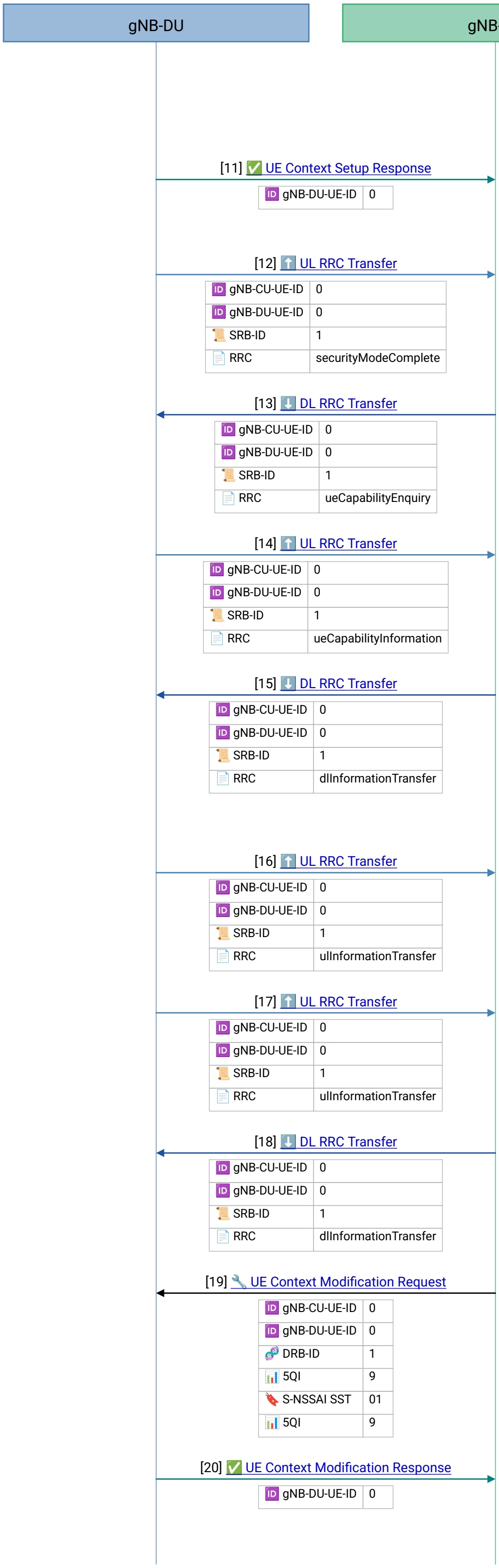
DL RRC Transfer (CU -> DU): NAS **Security Mode Command** (`0x5d`). Its NAS security header is type 3 — *integrity-protected with a new 5G security context* — the hinge of the whole flow: from here the network expects NAS to be protected with the freshly derived keys.

[Frame 9] UL RRC Transfer (DU -> CU): the UE's NAS **Security Mode Complete.** This is the first message under the new context — NAS security header type 4, *integrity-protected and ciphered with the new context* — so Wireshark can no longer decode the inner NAS type, only confirm it is protected. NAS security is now active; every NAS PDU after this (frames 15-18, 27-28) is ciphered.

[Frame 10]

Phase 5 — AS security and UE context at the DU

UE Context Setup Request (CU -> DU). NAS security is up; now the CU builds the UE's **radio** context on the DU. This F1



message establishes the SRBs and embeds, in its CU-to-DU RRC container, the RRC **SecurityModeCommand** that activates *access-stratum* (PDCP-layer) ciphering/integrity on the radio. (The data radio bearer is **not** set up here, and neither is the UE's slice (`S-NSSAI`) — both arrive later at Modification, frame 19.)

[Frame 11] UE Context Setup Response (DU -> CU). The DU has created the SRBs and returns its DU-to-CU RRC container — the `CellGroupConfig` (RLC/MAC/PHY radio parameters) that the CU folds into RRC toward the UE. The UE's F1 context now exists on both sides.

[Frame 12] UL RRC Transfer (DU -> CU): RRC **SecurityModeComplete.** The UE confirms access-stratum security is active. Both NAS (frame 9) and AS (here) security are now on — the radio link is fully protected.

[Frame 13]

Phase 6 — UE radio capabilities

DL RRC Transfer (CU -> DU): RRC **UECapabilityEnquiry.** The CU asks the UE what it can do (bands, carrier aggregation, MIMO, features) so it can tailor the reconfiguration that adds the DRB in Phase 8.

[Frame 14] UL RRC Transfer (DU -> CU): RRC **UECapabilityInformation** — the UE's capability report. The CU now knows the radio feature set it may configure.

[Frame 15]

Phase 7 — ciphered NAS exchange (registration completion)

DL RRC Transfer (CU -> DU): a **ciphered DL NAS** PDU (security header type 2). The inner type is not decodable, but at this point in a registration this is the AMF's **Registration Accept** path — the network confirming the UE is registered. Treat the specific type as inferred from position, not decoded.

[Frame 16] UL RRC Transfer (DU -> CU): a ciphered UL NAS PDU — the UE's side of the post-security NAS dialogue (e.g. Registration Complete / a UL NAS transport). Ciphered, so role-only.

[Frame 17] UL RRC Transfer (DU -> CU): a second ciphered UL NAS PDU. This is where the UE would drive **PDU-session establishment** signalling toward the SMF — the NAS that motivates the data radio bearer added next. Ciphered; inferred from position.

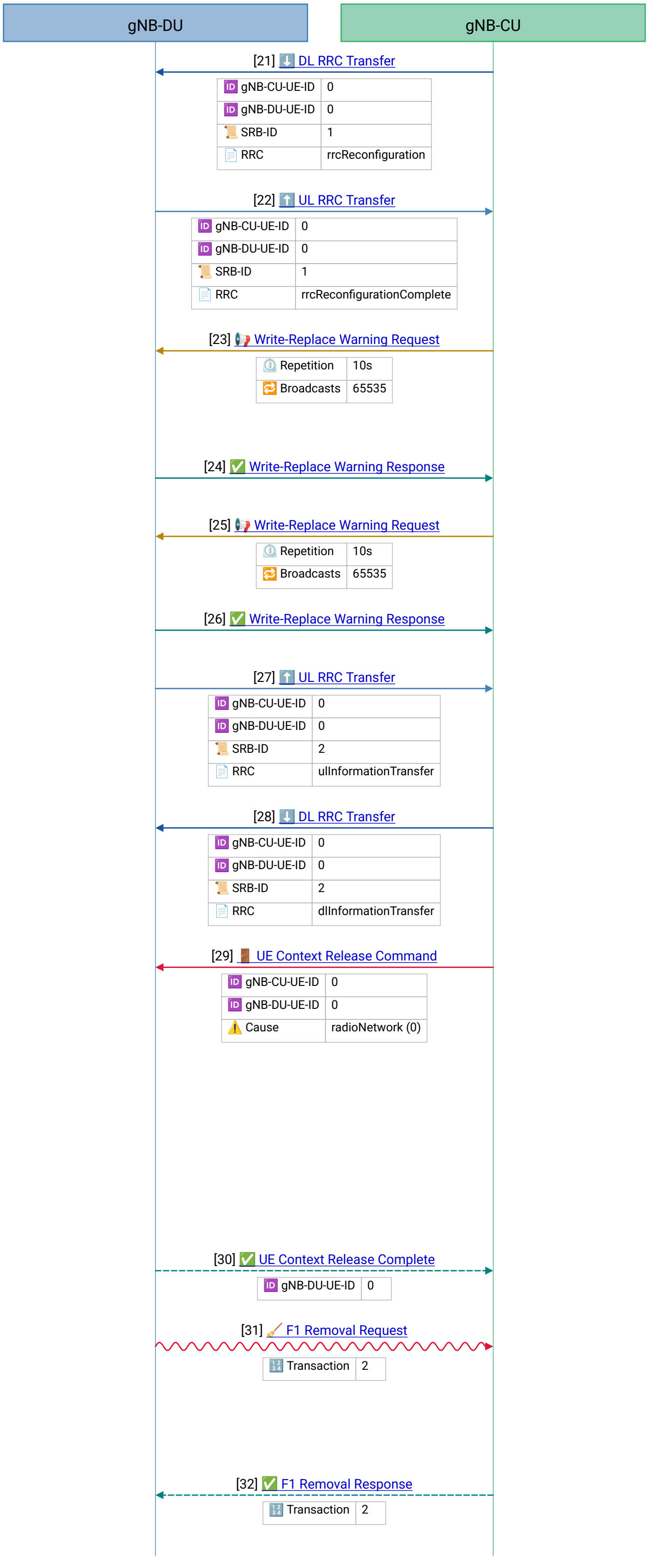
[Frame 18] DL RRC Transfer (CU -> DU): a ciphered DL NAS PDU answering the uplink NAS above. The NAS negotiation that a data bearer is needed completes here, which sets up Phase 8.

[Frame 19]

Phase 8 — data radio bearer establishment

UE Context Modification Request (CU -> DU). The payoff of the PDU-session NAS: the CU now adds the **data radio bearer** to the UE's existing context — `DRB-ID 1`, `5QI 9` (the standardised non-GBR default-bearer QoS), under `S-NSSAI SST 01`, with its GTP-U uplink TNL endpoint. The RRCReconfiguration that programs this DRB into the UE is delivered in frame 21.

[Frame 20] UE Context Modification Response (DU -> CU). The DU confirms the DRB is set up and returns its updated TNL / RRC info. The user-plane bearer now exists across F1; only the UE still has to be told.



[Frame 21] DL RRC Transfer (CU -> DU): RRC **RRCReconfiguration** on SRB1 . This is the UE-facing half of the DRB addition from frame 19 — it reconfigures the UE with the new bearer and radio parameters.

[Frame 22] UL RRC Transfer (DU -> CU): RRC **RRCReconfigurationComplete.** The UE has applied the new configuration. Registration plus default-bearer setup is now functionally complete; the UE is connected with a usable DRB.

[Frame 23]

Phase 9 — Public Warning System broadcast

Write-Replace Warning Request (CU -> DU), carrying **SIB6.** This is *non-UE-associated* — it concerns the cell, not our UE. The CU pushes an ETWS primary notification (`SIB6`) for the DU to broadcast, asking for 65535 repeats at a 10s period.

[Frame 24] Write-Replace Warning Response (DU -> CU). The DU acknowledges and reports the cells now broadcasting the warning.

[Frame 25] Write-Replace Warning Request (CU -> DU), carrying **SIB7** — the ETWS *secondary* notification (the actual warning content), companion to the SIB6 primary in frame 23.

[Frame 26] Write-Replace Warning Response (DU -> CU). Second PWS acknowledgement; the ETWS broadcast is fully armed on the cell. The UE-associated flow resumes below.

[Frame 27]

Phase 10 — final ciphered NAS

UL RRC Transfer (DU -> CU): a ciphered UL NAS PDU (security header type 2) — a late uplink NAS transaction from the UE before teardown. Ciphered; role-only.

[Frame 28] DL RRC Transfer (CU -> DU): the ciphered DL NAS reply. This closes the NAS dialogue; what follows is teardown.

[Frame 29]

Phase 11 — UE context release

UE Context Release Command (CU -> DU) with `Cause : radioNetwork normal-release` . The CU tells the DU to release this UE's context and radio resources — an orderly release (not a failure), consistent with a scripted registration-then-teardown capture.

It also embeds the UE's final `RRCRelease` , but as an opaque `RRCContainer ( 00071000ff6c1afc )` rather than a decoded message — Wireshark parses the RRC-Container inside a DL/UL RRC Transfer but not inside a Release Command. Those bytes are a PDCP PDU (SN 7) whose `1000` payload is a DL-DCCH `rrcRelease` plus its 4-byte MAC-I, in the clear because AS ciphering here is `NEA0` (null). So the RRCRelease **is** delivered to the UE — embedded in the release command, not sent as its own message.

[Frame 30] UE Context Release Complete (DU -> CU). The DU confirms the UE's context and resources are freed. The per-UE F1 session that opened back at frame 3 ends here.

[Frame 31]

Phase 12 — F1 interface teardown

F1 Removal Request (`Transaction 2`) . With the UE gone, the F1 *interface* itself is torn down. Note: TS 38.473 lets **either** node initiate F1 Removal, and this exported-PDU capture carries no transport to prove which — it is drawn DU -> CU here (see the FXT header note).

[Frame 32] F1 Removal Response (`Transaction 2`) . The peer confirms removal; the F1 TNL association from frame 1 is gone. A new F1 Setup would be required before any further CU/DU signalling. End of capture.