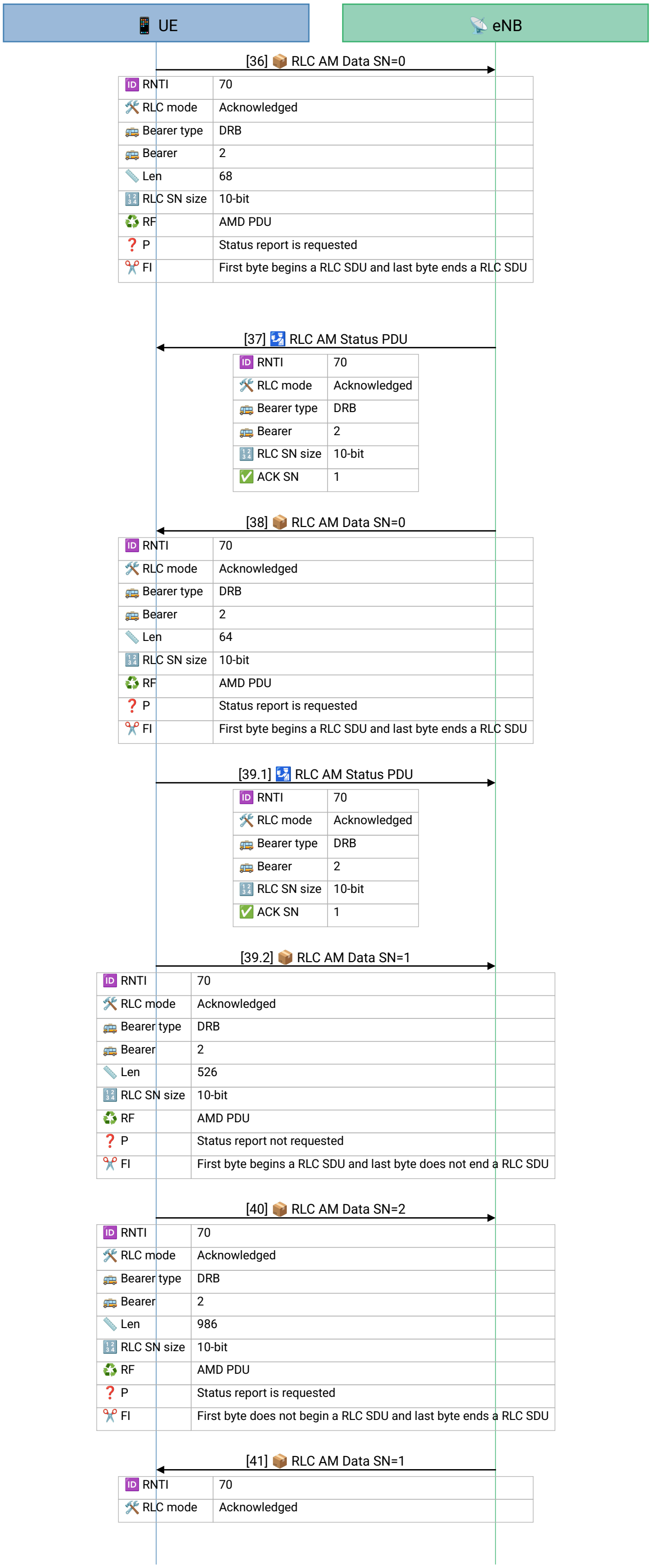
lte-rlc-am-ims-bearer.annotated



[Frame 36] Act 1 - IMS registration. The bearer's first PDU. DRB 2 was added by the RRC Connection Reconfiguration in frame 32 (EPS bearer 6, QCI 5, LCID 4) with an RLC AM configuration:

- UE transmitter (`ul-AM-RLC`): `t-PollRetransmit` 80 ms, `pollPDU` 128, `pollByte` 125 kB, `maxRetxThreshold` 4
- UE receiver (`dl-AM-RLC`): `t-Reordering` 80 ms, `t-StatusProhibit` 60 ms

This is uplink `SN=0`: a 68-byte PDU carrying the TCP SYN that opens the UE's SIP connection to the P-CSCF (port 5060). `FI` = 00, so the PDU holds exactly one whole SDU (one PDCP packet). The poll bit P is set: the UE wants a Status PDU back. The LTE AM header is 2 bytes: D/C, RF, P, two FI bits, E and a 10-bit SN.

[Frame 37] The eNB's Status PDU, 0.7 ms later. `ACK_SN=1` means "the next SN I am waiting for is 1", so SN 0 is in. With no NACK it is 2 bytes: D/C, a 3-bit CPT, the 10-bit `ACK_SN` and E1 = 0.

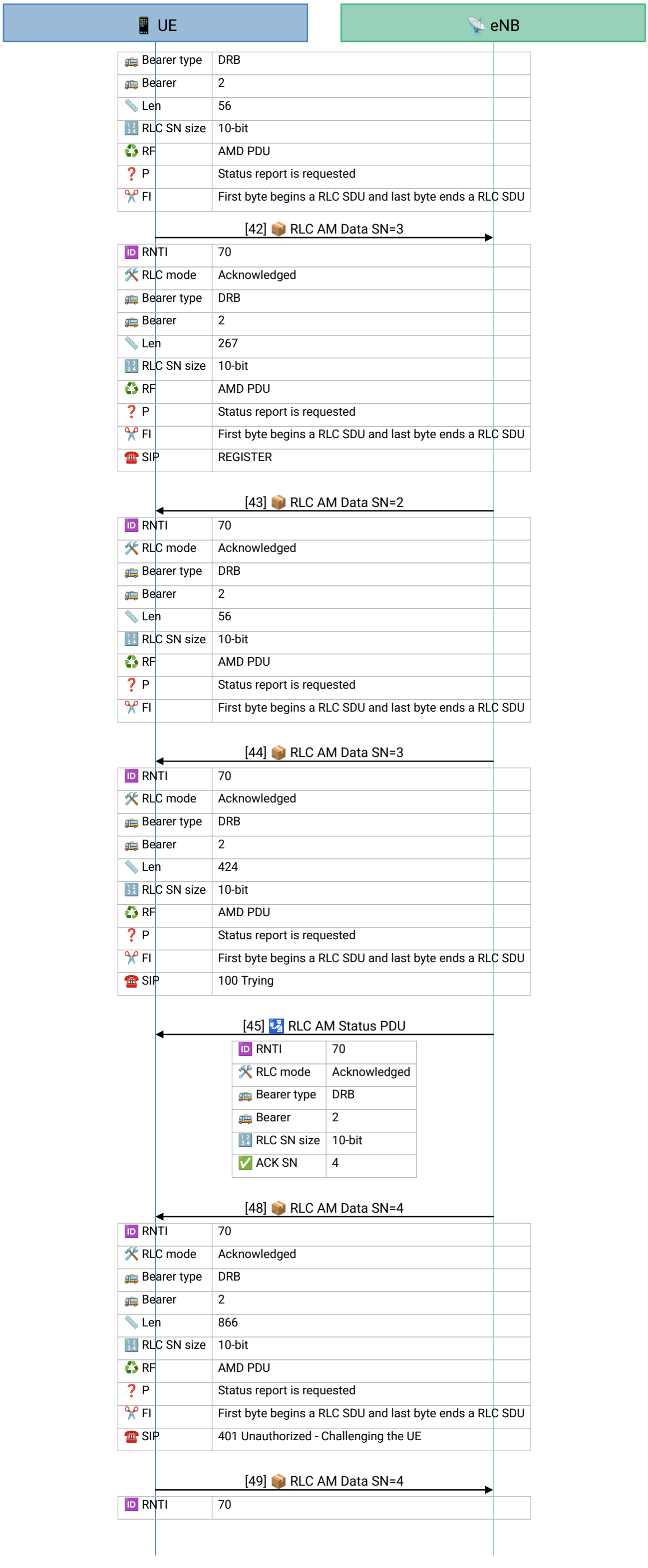
[Frame 38] Downlink `SN=0`: the P-CSCF's SYN-ACK. The downlink has its own transmitting entity and its own count, so this SN 0 has nothing to do with frame 36's. The eNB polls here, and it will poll on every one of its 128 data PDUs on this bearer.

[Frame 39.1] The UE acknowledges downlink SN 0 (`ACK_SN=1`), 38 ms after the poll: the UE can only send once the eNB grants it uplink resources.

[Frame 39.2] Concatenation and segmentation in one PDU. Uplink `SN=1`, 526 bytes, `FI` = 01 (starts with the start of an SDU, ends mid-SDU). A length indicator (`LI=54`) marks the end of the first SDU, a 54-byte PDCP packet carrying the handshake's last TCP ACK. The remaining 468 bytes are the first piece of the next SDU: a 1452-byte packet with the first TCP segment of the SIP REGISTER. No poll: more data is waiting.

[Frame 40] Uplink `SN=2`, `FI` = 10 (starts mid-SDU, ends an SDU): the last 984 bytes of that 1452-byte SDU. Wireshark reassembles it from 2 segments, SN 1 and SN 2. In LTE the SN numbers PDUs, not SDUs, so the two pieces carry different SNs and `FI` says how they join. (5G NR is different: there all segments of one SDU share an SN.) P = 1 because this PDU empties the UE's buffer.

[Frame 41] Downlink `SN=1`, a TCP ACK from the P-CSCF.



[Frame 42] Uplink SN=3 : the second TCP segment (211 bytes) completes the `REGISTER`. It travels in the clear, on port 5060, because no IPsec security exists yet.

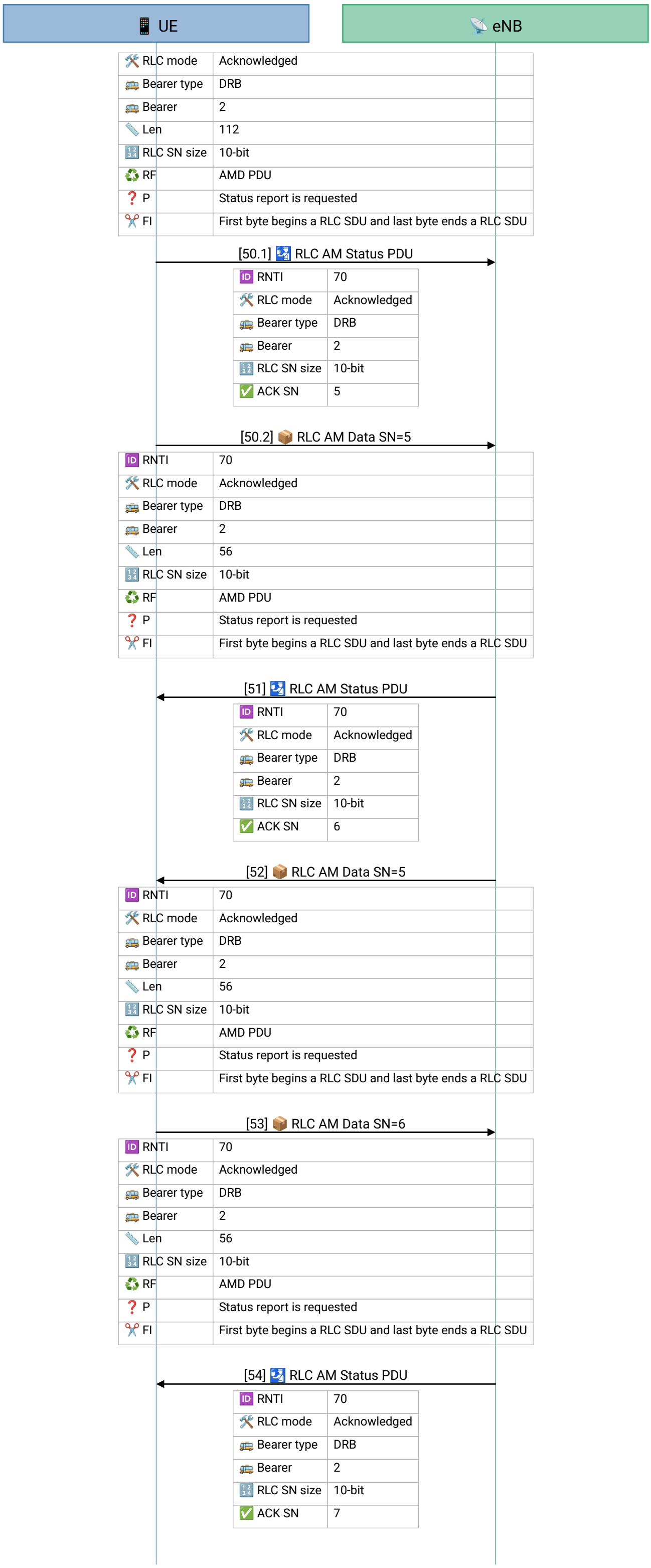
[Frame 43] Downlink SN=2, another TCP ACK.

[Frame 44] Downlink SN=3 carries `100 Trying`.

[Frame 45] ACK_SN=4 : uplink SN 0 to 3 all arrived, including both halves of the REGISTER.

[Frame 48] Downlink SN=4, 866 bytes: `401 Unauthorized`, the IMS-AKA challenge. One whole SIP response in one PDU: downlink grants are large enough here that the eNB never needs to split a small message.

[Frame 49] Uplink SN=4, 112 bytes: two TCP ACKs (54 bytes each), joined by one length indicator. Concatenation saves a header.



[Frame 50.1] ACK_SN=5 covers downlink SN 0 to 4.

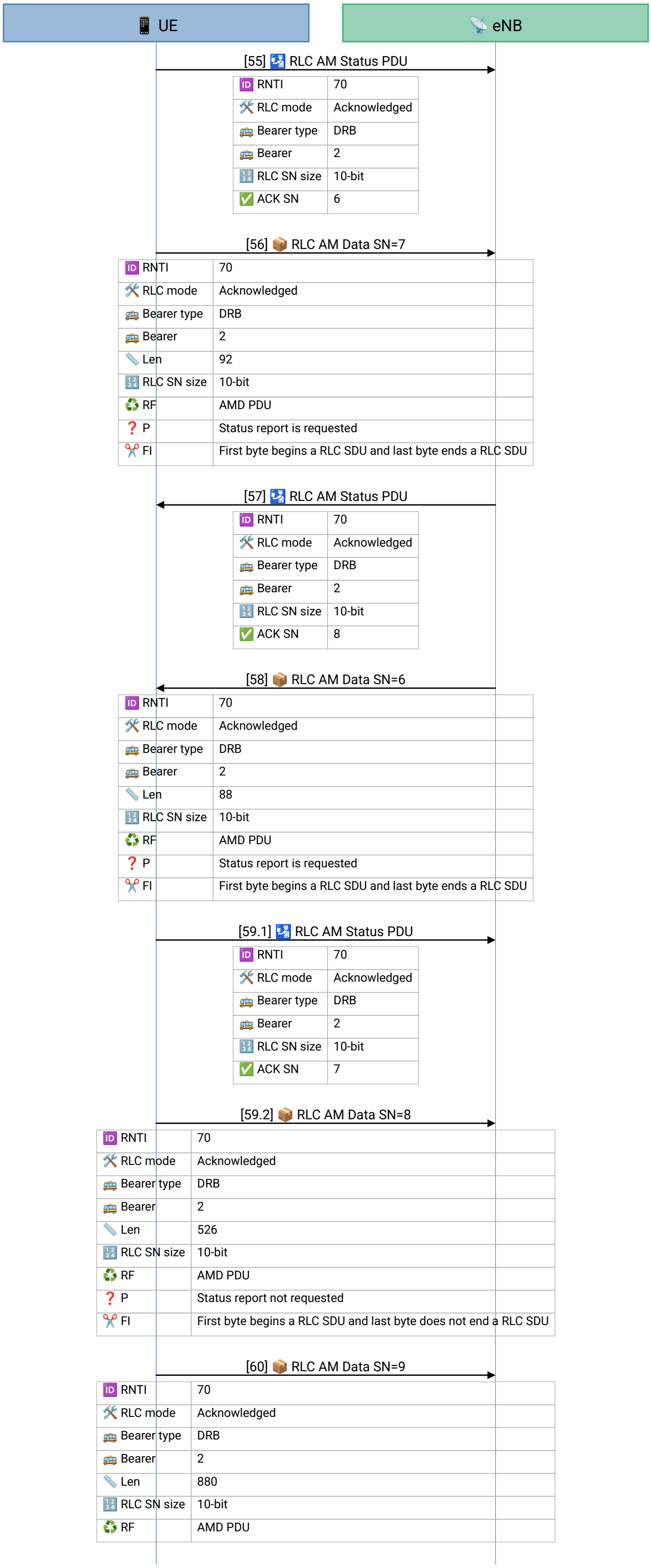
[Frame 50.2] Uplink SN=5 : the UE's TCP FIN. The unprotected connection closes; the second REGISTER will use a new connection inside IPsec.

[Frame 51] ACK_SN=6 .

[Frame 52] Downlink SN=5 , the P-CSCF's FIN, polled.

[Frame 53] Uplink SN=6 , the final TCP ACK.

[Frame 54] ACK_SN=7 : everything the UE has sent so far is in.



[Frame 55] t-StatusProhibit at work. `ACK_SN=6` answers frame 52's poll, but 78 ms after it. The UE sent its previous Status PDU in frame 50, and `t-StatusProhibit` (60 ms, from frame 32) stops it sending another until the timer runs out. 60 ms, plus the wait for an uplink grant, gives the 80 ms between frames 50 and 55. Keep this in mind for Act 3.

[Frame 56] Uplink `SN=7`, 92 bytes: a TCP SYN, now inside ESP (SPI `0x00001001`) to the P-CSCF's protected port 6100. RLC neither knows nor cares that the SDU is now an IPsec packet.

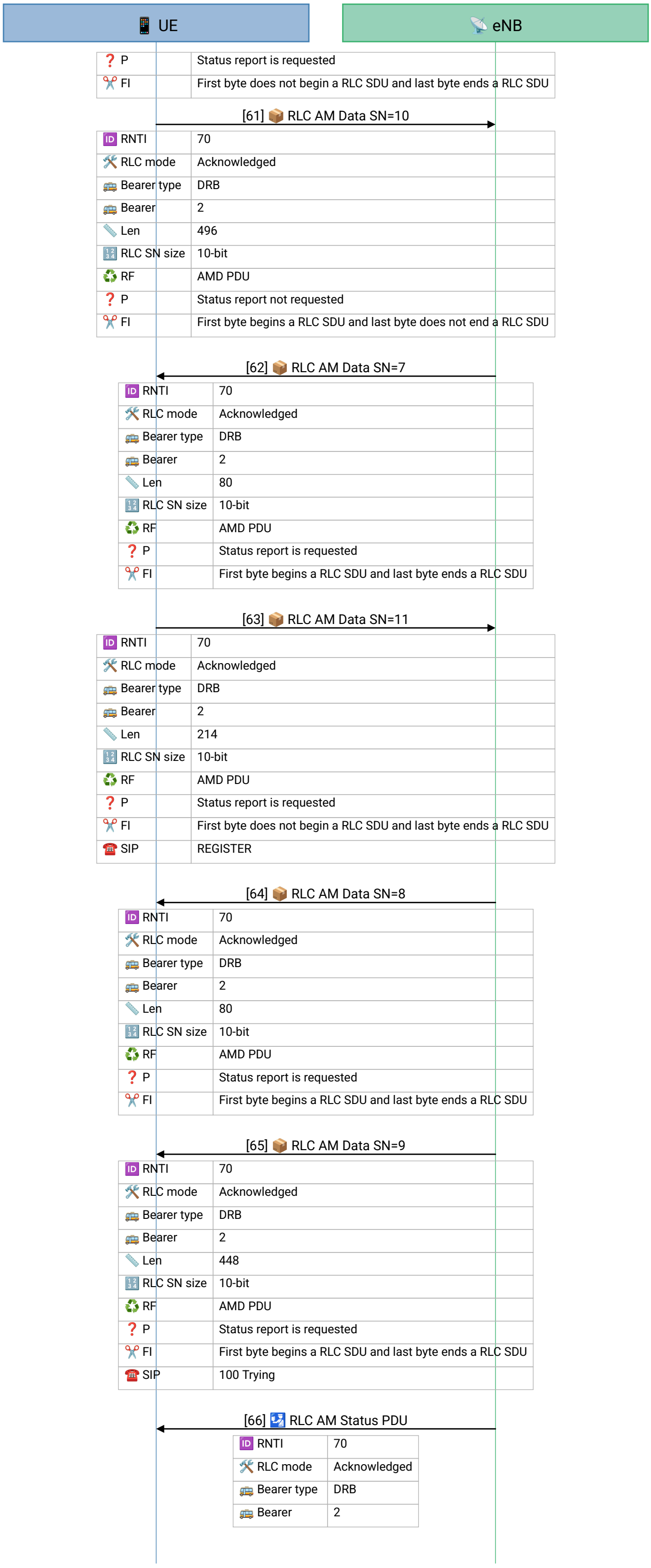
[Frame 57] `ACK_SN=8`.

[Frame 58] Downlink `SN=6`, the SYN-ACK in ESP.

[Frame 59.1] `ACK_SN=7`, 38 ms after the poll.

[Frame 59.2] Uplink `SN=8`, `FI = 01`: a 78-byte TCP ACK, then the first 444 bytes of a 1322-byte SDU (the first TCP segment of the protected REGISTER).

[Frame 60] Uplink `SN=9`, `FI = 10`: the rest of that SDU, polled.



[Frame 61] Uplink `SN=10, FI = 01`: the start of the second TCP segment.

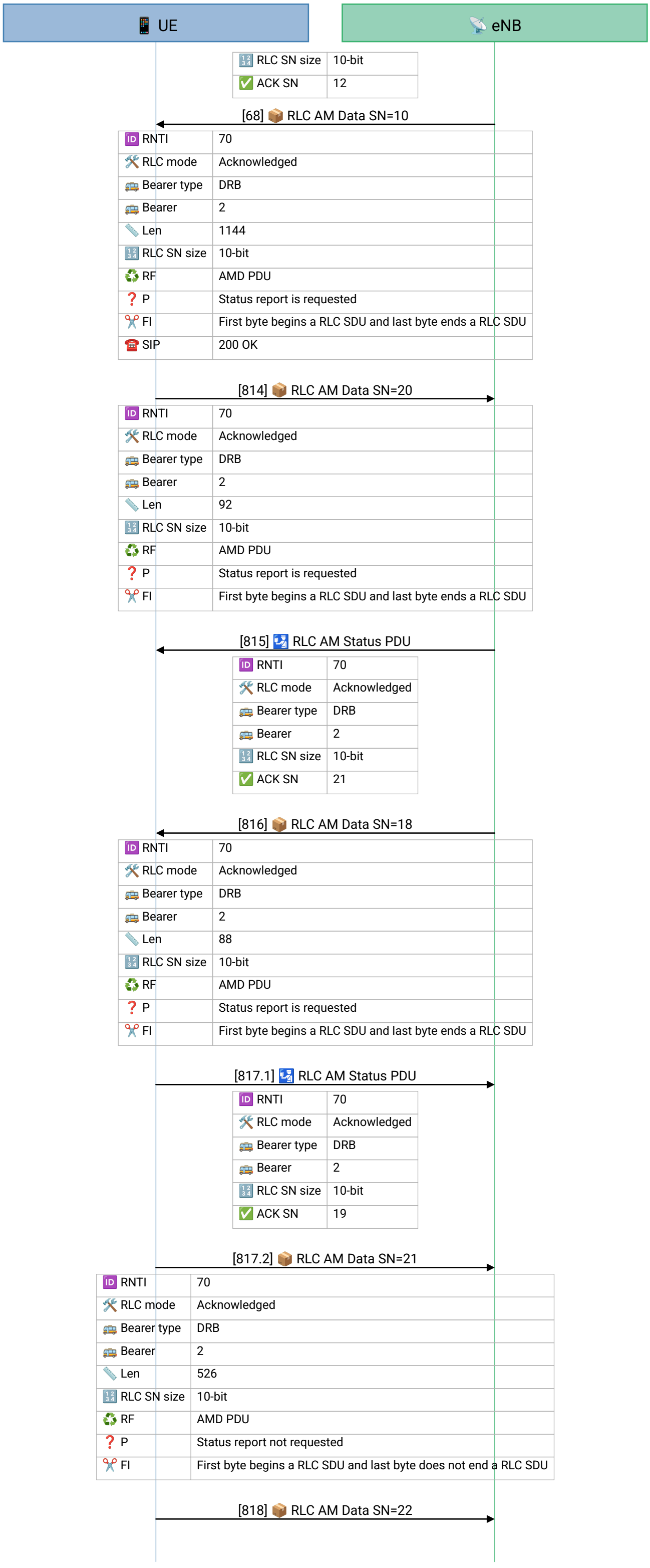
[Frame 62] Downlink `SN=7`, a TCP ACK in ESP.

[Frame 63] Uplink `SN=11, FI = 10`: the protected `REGISTER` is complete, four PDUs (SN 8 to 11) for two TCP segments of 1245 and 629 bytes.

[Frame 64] Downlink `SN=8`, a TCP ACK.

[Frame 65] Downlink `SN=9 : 100 Trying.`

[Frame 66] `ACK_SN=12 :` uplink SN 0 to 11, every PDU of both REGISTERs, delivered without a single NACK.



[Frame 68] Downlink `SN=10`, 1144 bytes: `200 OK`. UE 70 is registered with IMS. Its acknowledgment falls after this window.

[Frame 814] Act 2 - Call 1. 21 seconds later, the same entity. Uplink `SN=20` continues the count the UE kept since frame 63 (SN 12 to 19 went by outside this window). The SDU is a TCP SYN in ESP to port 6100: UE 70 opens a connection to send its INVITE.

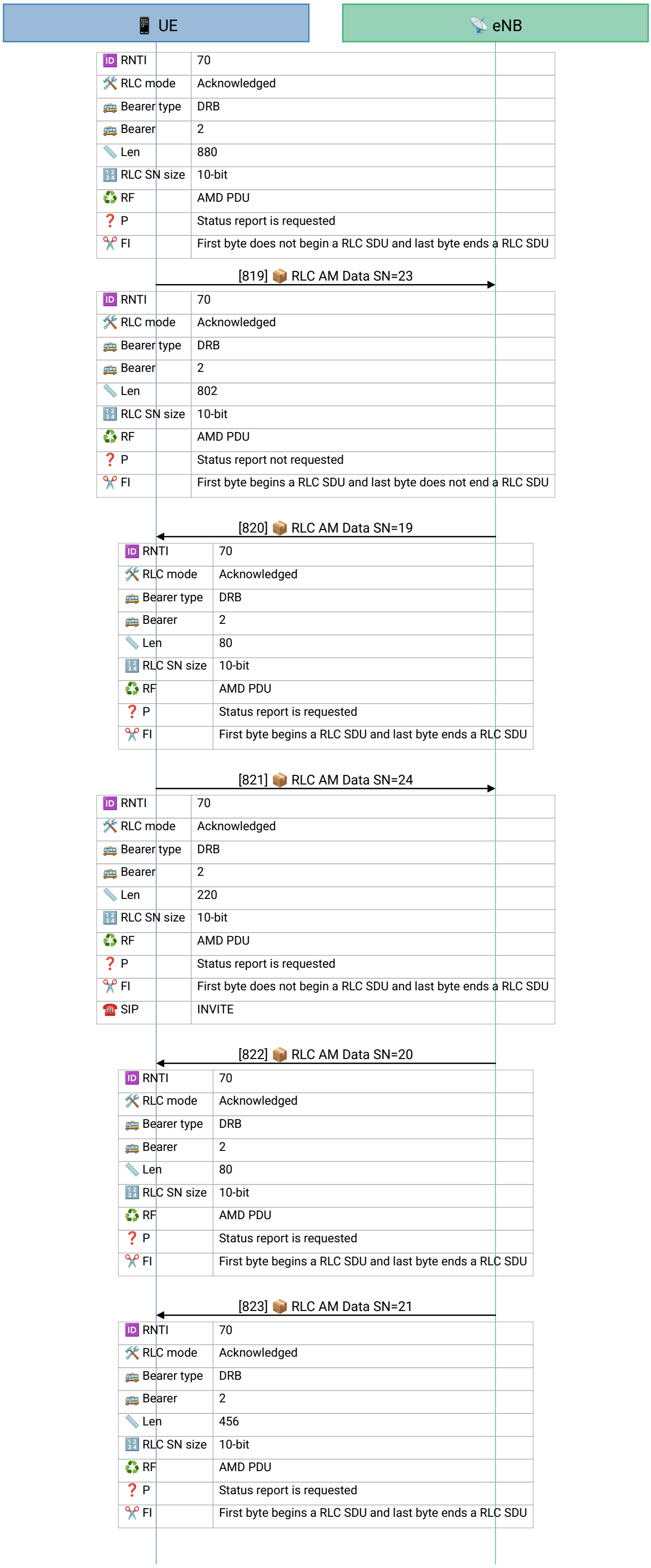
[Frame 815] `ACK_SN=21`, 0.7 ms later.

[Frame 816] Downlink `SN=18`, the SYN-ACK.

[Frame 817.1] `ACK_SN=19`.

[Frame 817.2] The INVITE, part 1 of 4. Uplink `SN=21`, 526 bytes, `FI` = 01: a 78-byte TCP ACK (ended by `LI=78`), then the first 444 bytes of a 1322-byte SDU, the INVITE's first TCP segment.

[Frame 818] Part 2. `SN=22`, `FI` = 10: the remaining 878 bytes of that SDU. P = 1, although more of the INVITE



follows. Of the TS 36.322 poll triggers, only "this PDU empties the transmit buffer" can fire here (`pollPDU` 128 and `pollByte` 125 kB are far off), so the INVITE's second TCP segment had most likely not reached RLC yet.

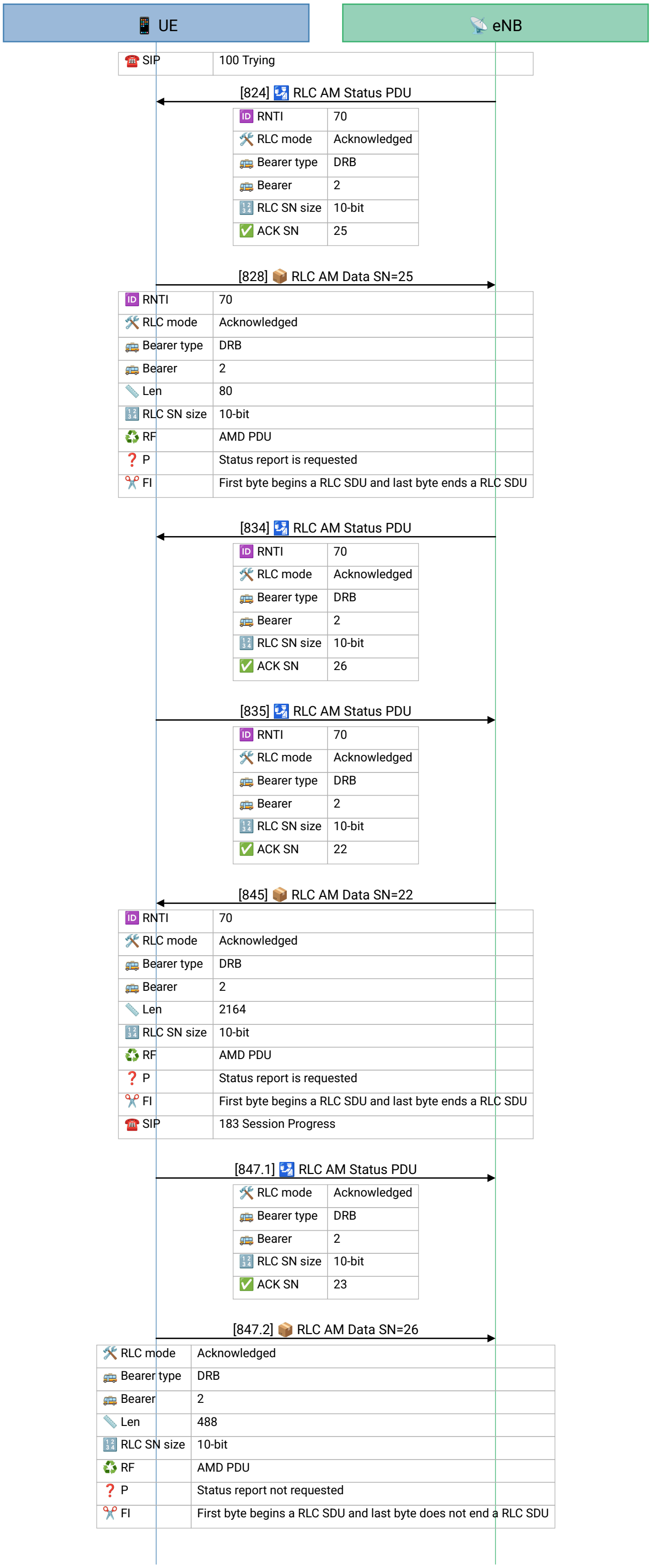
[Frame 819] Part 3. SN=23 , `FI` = 01, no poll: the first 800 bytes of the second SDU (1018 bytes).

[Frame 820] Downlink SN=19 , a TCP ACK, in the middle of the INVITE.

[Frame 821] Part 4. SN=24 , `FI` = 10, P = 1: the last 218 bytes. Two SDUs, four PDUs, four SNs, and the `INVITE` (2187 bytes of TCP) is complete. The PDUs left the UE over 17 ms because each uplink grant was smaller than an SDU.

[Frame 822] Downlink SN=20 , a TCP ACK.

[Frame 823] Downlink SN=21 : `100 Trying`.



[Frame 824] ACK_SN=25 : all four INVITE PDUs arrived. One cumulative ACK covers them.

[Frame 828] Uplink SN=25, a TCP ACK.

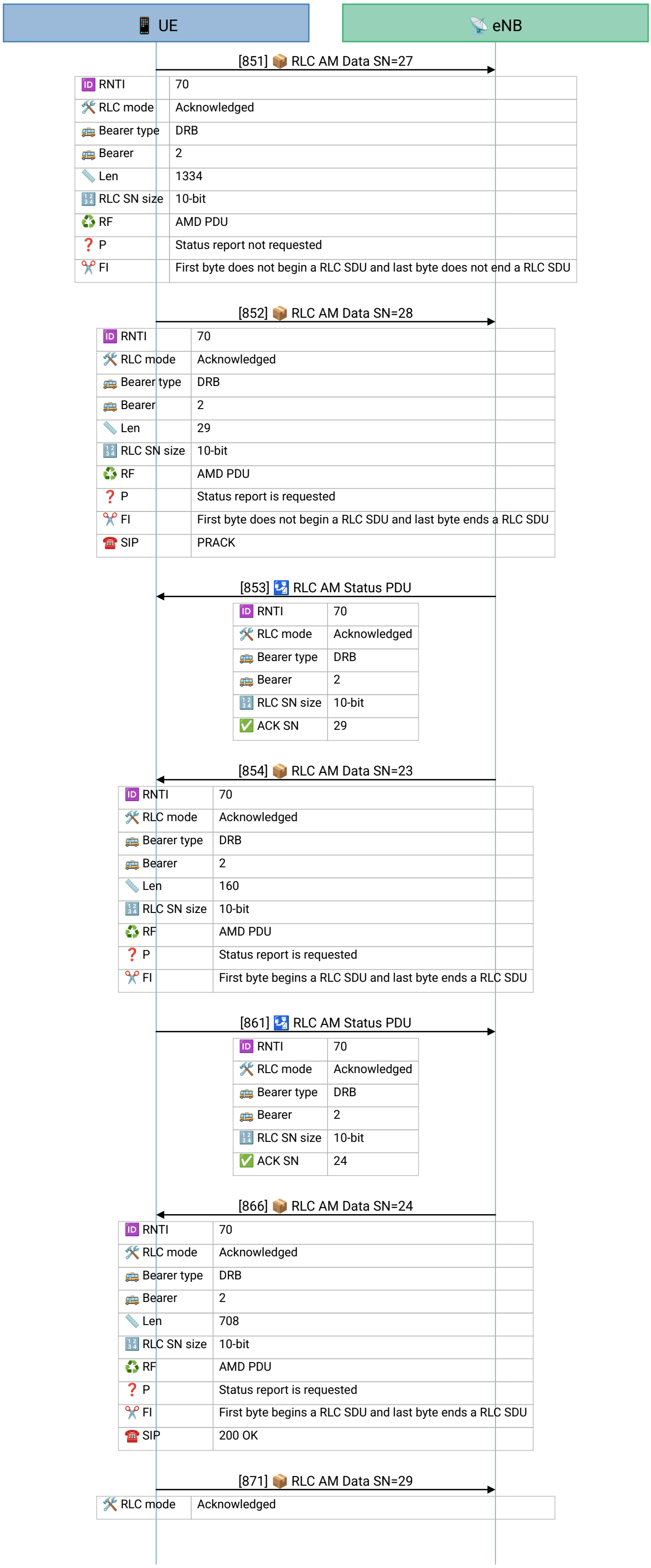
[Frame 834] ACK_SN=26 .

[Frame 835] ACK_SN=22 for downlink SN 0 to 21.

[Frame 845] The downlink concatenates instead. Downlink SN=22, 2164 bytes, FI = 00: the whole 183 Session Progress in one PDU, both TCP segments (SDUs of 1322 and 838 bytes, split by LI=1322). The uplink cut the INVITE into four PDUs; the downlink packs the 183 into one. Same protocol, different grant sizes.

[Frame 847.1] ACK_SN=23 .

[Frame 847.2] PRACK, part 1 of 3. Uplink SN=26, FI = 01: two 78-byte TCP ACKs, then the first 327 bytes of the next SDU.



[Frame 851] Part 2, FI = 11. Uplink SN=27, 1334 bytes, starts mid-SDU and ends mid-SDU: 995 bytes finish the first SDU (Wireshark: 2 segments, 1322 bytes), and LI=995 marks where the next one begins. One PDU, the end of one SDU and the start of another. No poll.

[Frame 852] Part 3. SN=28, 29 bytes, FI = 10, P = 1: the last 27 bytes complete the PRACK.

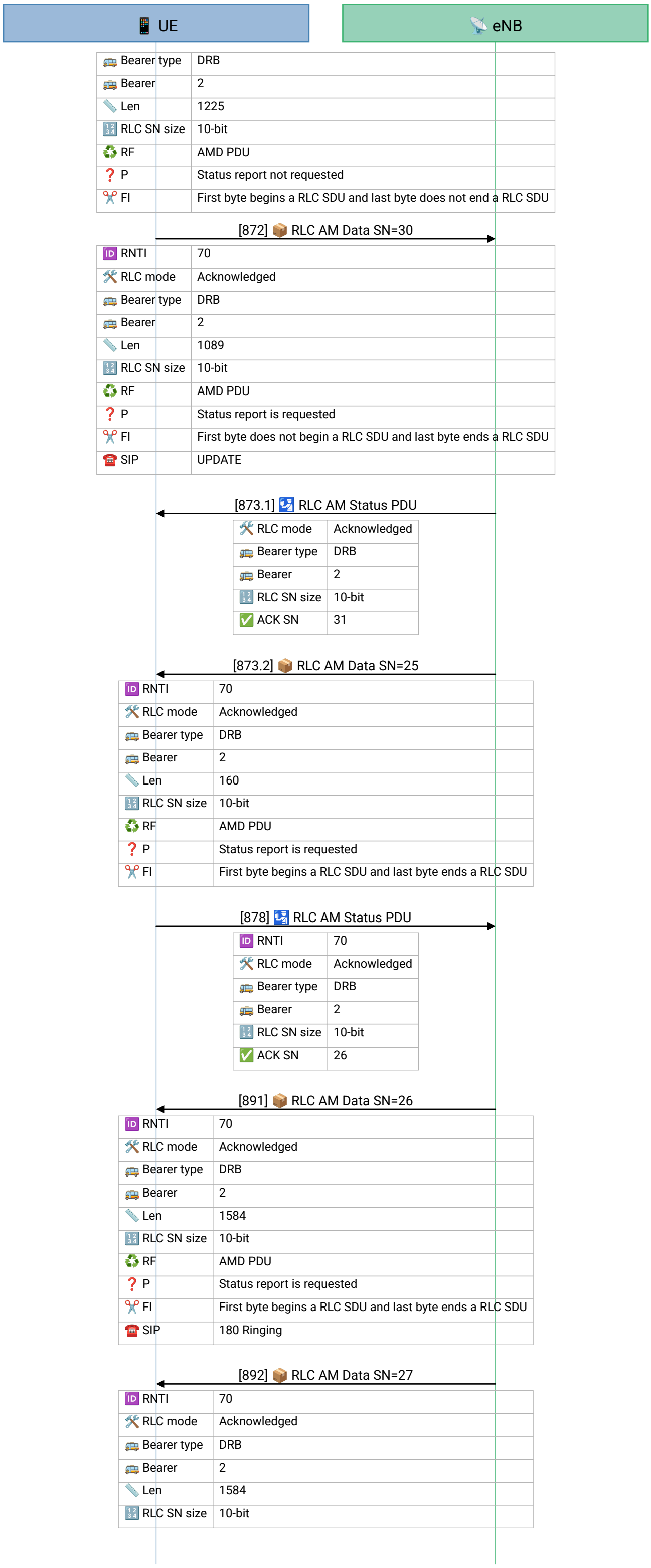
[Frame 853] ACK_SN=29, 0.06 ms after the poll.

[Frame 854] Downlink SN=23: two TCP ACKs in one 160-byte PDU.

[Frame 861] ACK_SN=24.

[Frame 866] Downlink SN=24: 200 OK to the PRACK.

[Frame 871] Uplink SN=29, FI = 01: the start of the UPDATE, which tells the far end UE 70's voice bearer is ready.



[Frame 872] Uplink `SN=30, FI = 10, P = 1`: the UPDATE is complete.

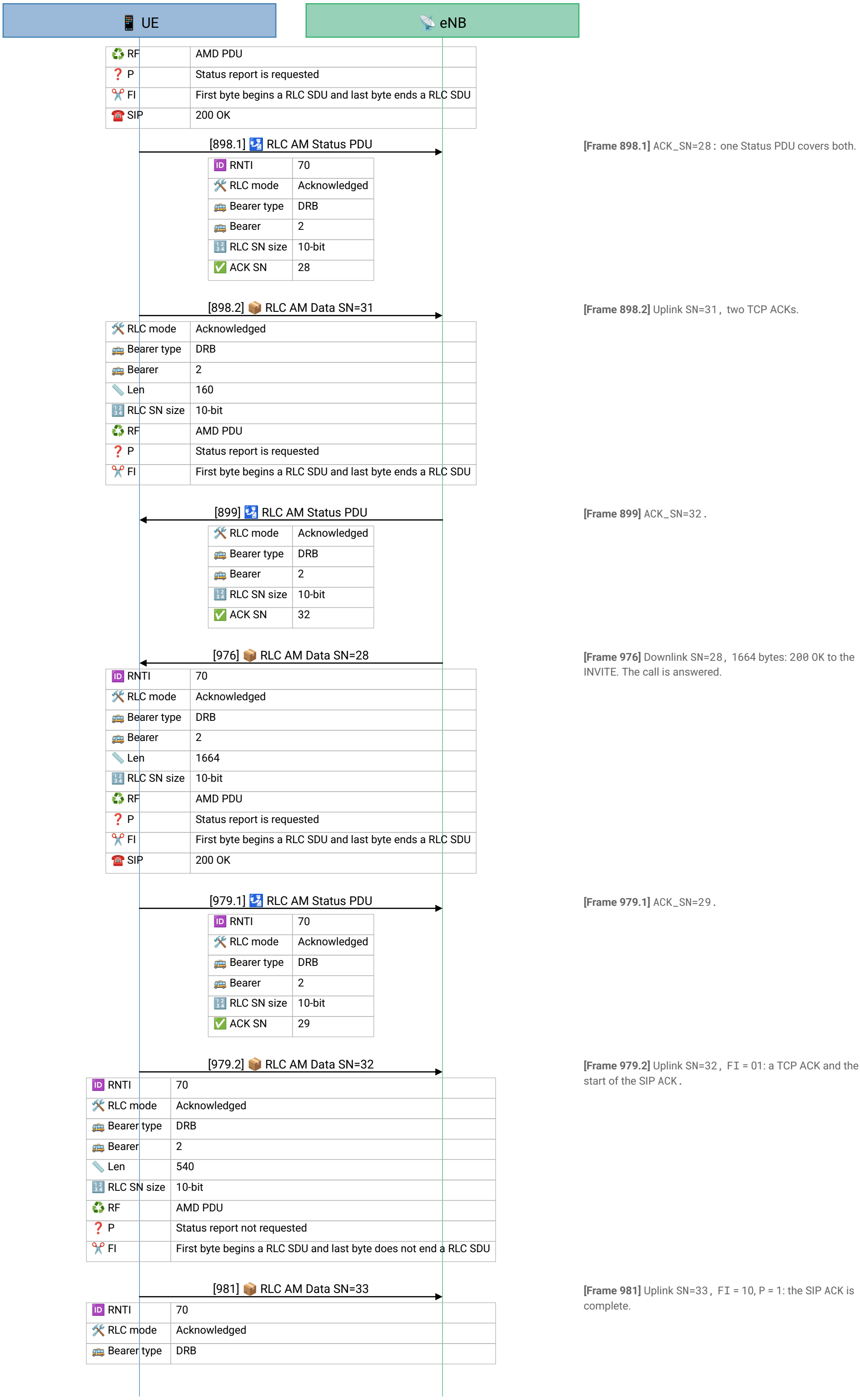
[Frame 873.1] `ACK_SN=31.`

[Frame 873.2] Downlink `SN=25`, two TCP ACKs.

[Frame 878] `ACK_SN=26.`

[Frame 891] Downlink `SN=26`, 1584 bytes: `180 Ringing`, both TCP segments in one PDU.

[Frame 892] Downlink `SN=27`, 1584 bytes: `200 OK` to the UPDATE, 1.2 ms later.



[Frame 898.1] ACK_SN=28: one Status PDU covers both.

[Frame 898.2] Uplink SN=31, two TCP ACKs.

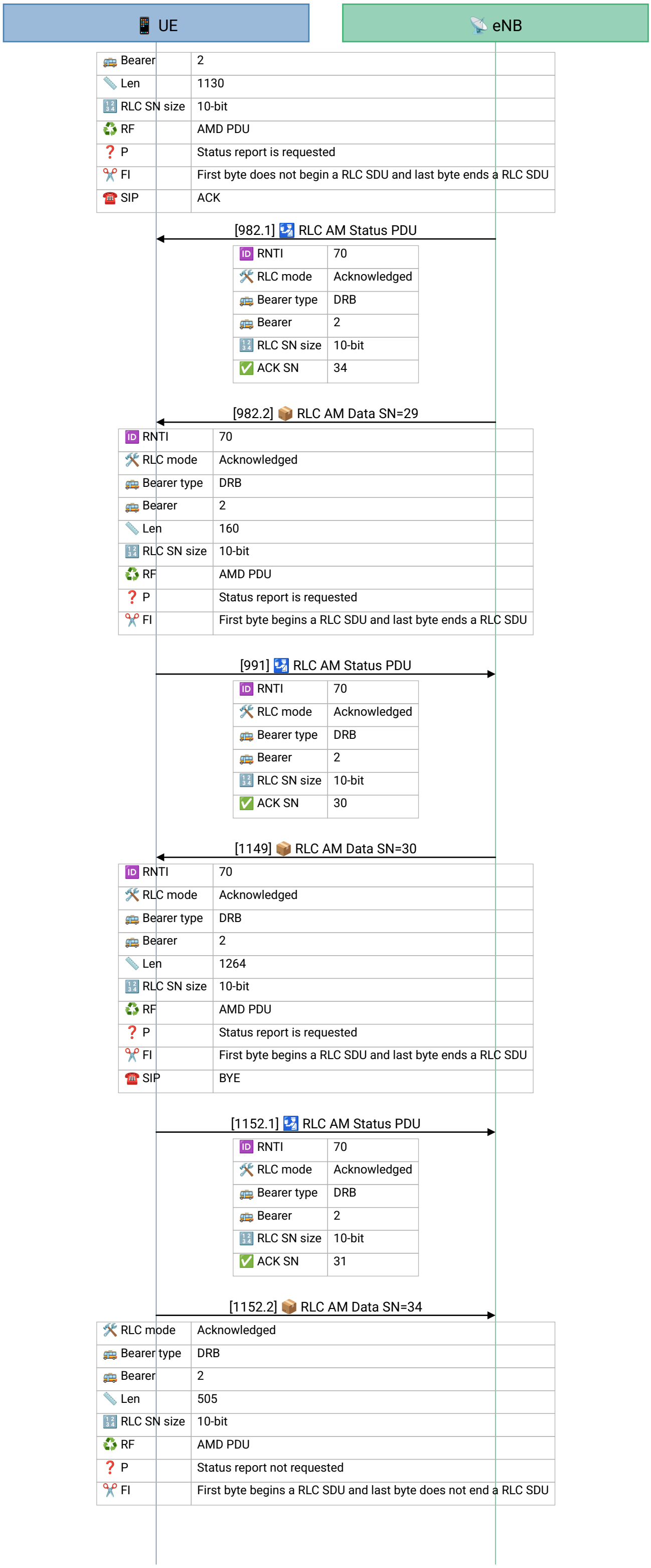
[Frame 899] ACK_SN=32.

[Frame 976] Downlink SN=28, 1664 bytes: 200 OK to the INVITE. The call is answered.

[Frame 979.1] ACK_SN=29.

[Frame 979.2] Uplink SN=32, FI = 01: a TCP ACK and the start of the SIP ACK.

[Frame 981] Uplink SN=33, FI = 10, P = 1: the SIP ACK is complete.



[Frame 982.1] `ACK_SN=34`.

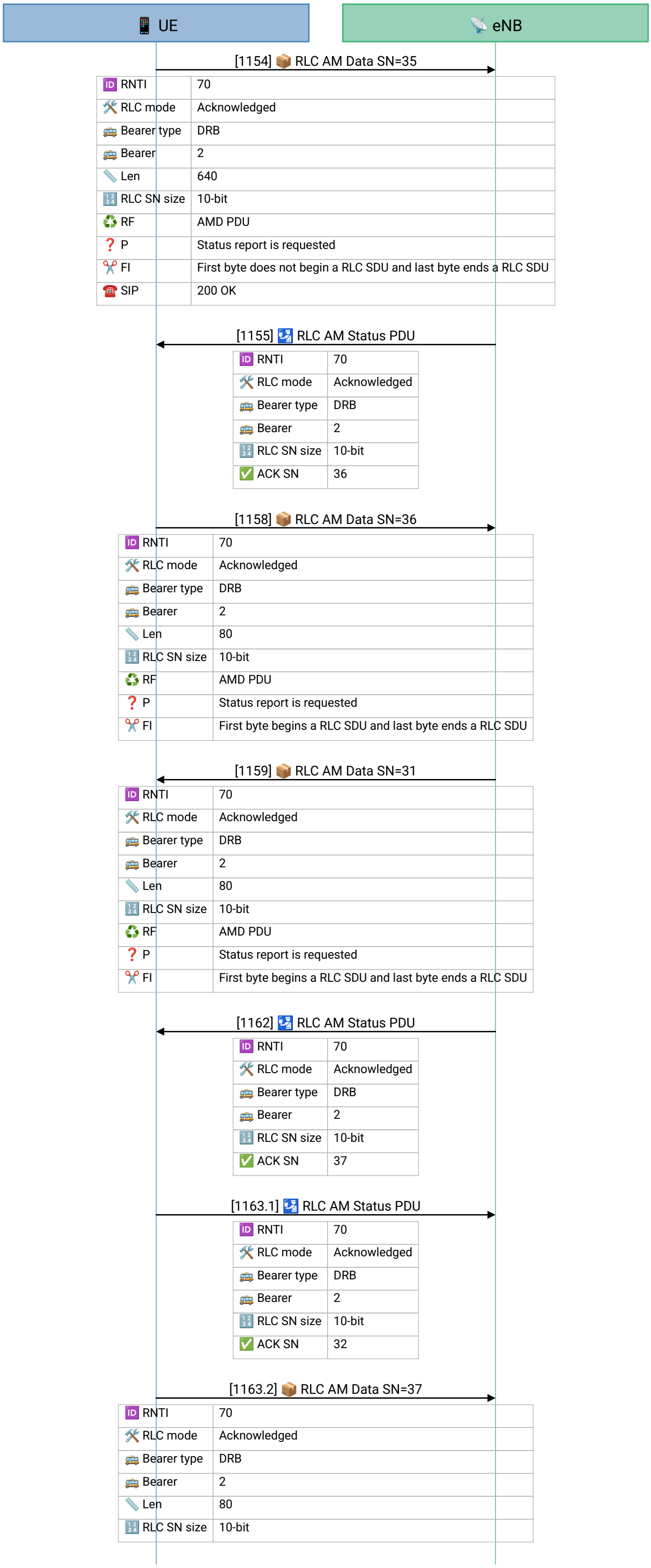
[Frame 982.2] Downlink `SN=29`, two TCP ACKs.

[Frame 991] `ACK_SN=30`. The same transport block carries a voice packet on DRB 3, which runs RLC UM and has no Status PDUs of its own. For the next 4.5 s, voice flows on DRB 3 and this bearer is silent: an AM entity with nothing to send sends nothing.

[Frame 1149] Downlink `SN=30`, 1264 bytes: the `BYE`.

[Frame 1152.1] `ACK_SN=31`.

[Frame 1152.2] A voice packet on the AM bearer. Uplink `SN=34`, `FI` = 01. Its first SDU, ended by `LI=49`, is a 47-byte IP packet carrying RTP sequence 36, a comfort-noise packet from UE 70 to the media relay. DRB 3 was released in frame 1146, 40 ms earlier, so the voice bearer's packet filters are gone and the packet falls through to the IMS PDN's default bearer: this one. The rest of the PDU starts the UE's `200 OK` to the BYE.



[Frame 1154] Uplink `SN=35`, `FI = 10`, `P = 1`: the `200 OK` is complete.

[Frame 1155] `ACK_SN=36`.

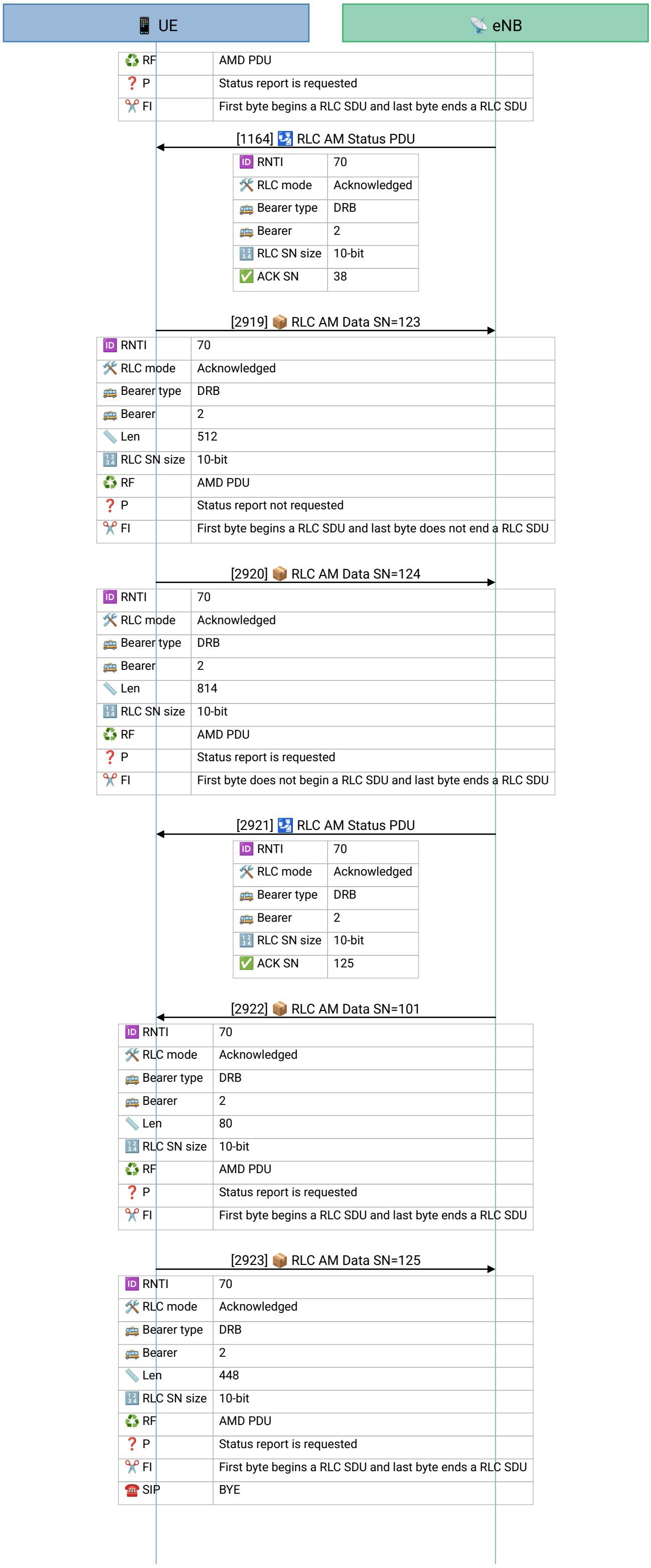
[Frame 1158] Uplink `SN=36` : the UE closes the TCP connection (FIN).

[Frame 1159] Downlink `SN=31` : the P-CSCF's FIN.

[Frame 1162] `ACK_SN=37`.

[Frame 1163.1] `ACK_SN=32` for downlink SN 0 to 31.

[Frame 1163.2] Uplink `SN=37`, the last TCP ACK.



[Frame 1164] ACK_SN=38. Call 1 ends with both directions fully acknowledged and no NACK.

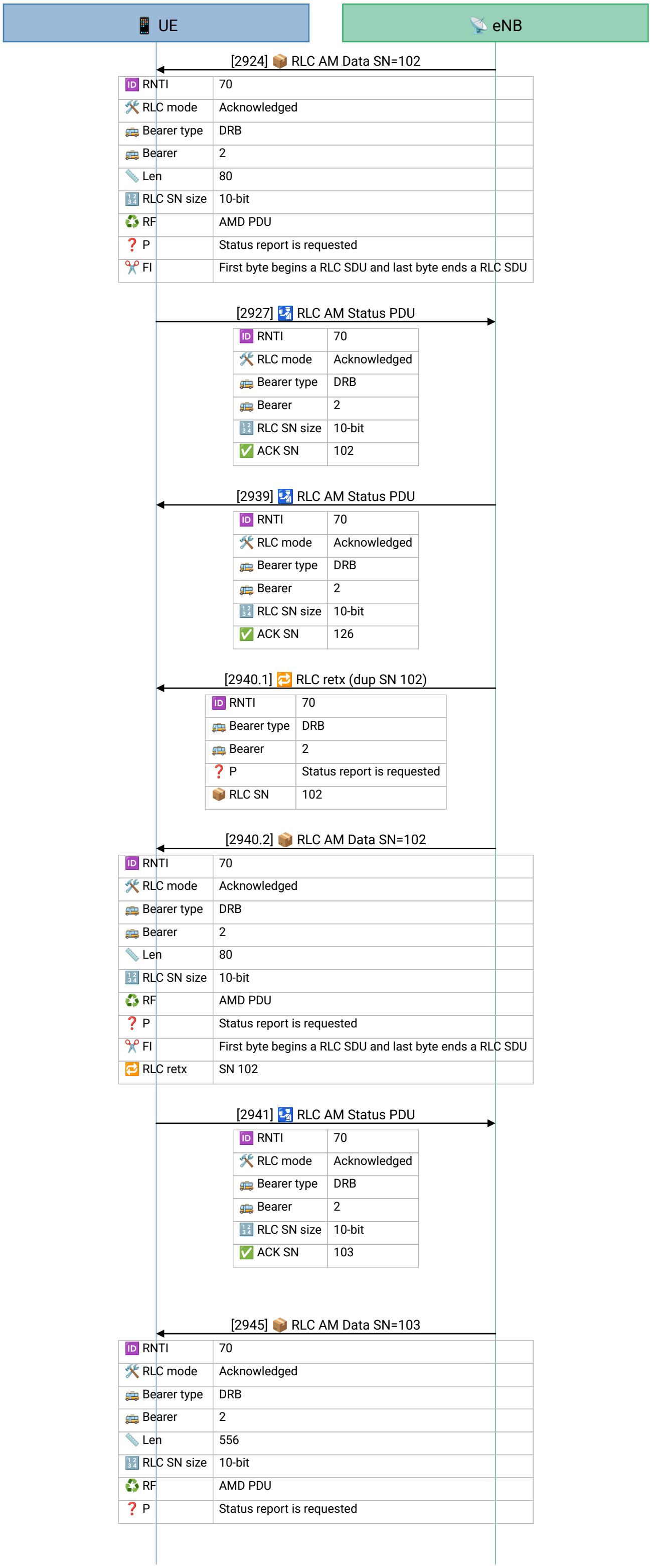
[Frame 2919] Act 3 - The one retransmission. 59 s later, at the end of call 5. Uplink SN=123, FI = 01: UE 70 hangs up, and the BYE goes up as SN 123 to 125.

[Frame 2920] Uplink SN=124, FI = 10, P = 1: the first TCP segment of the BYE (1245 bytes) is complete.

[Frame 2921] ACK_SN=125.

[Frame 2922] Downlink SN=101, polled: a TCP ACK from the P-CSCF.

[Frame 2923] Uplink SN=125: the second segment completes the BYE.



[Frame 2924] The original. Downlink `SN=102`, 80 bytes, polled: another TCP ACK (ESP sequence 78). It goes out in radio frame (SFN) 141, subframe 0. The eNB starts `t-PollRetransmit`.

[Frame 2927] `ACK_SN=102`, in SFN 141 subframe 3. This answers frame 2922's poll: SN 101 is in. It cannot cover SN 102, which reached the UE only 3 subframes earlier; an uplink transmission is built from a grant 4 subframes ahead. Sending this Status PDU starts the UE's 60 ms `t-StatusProhibit`.

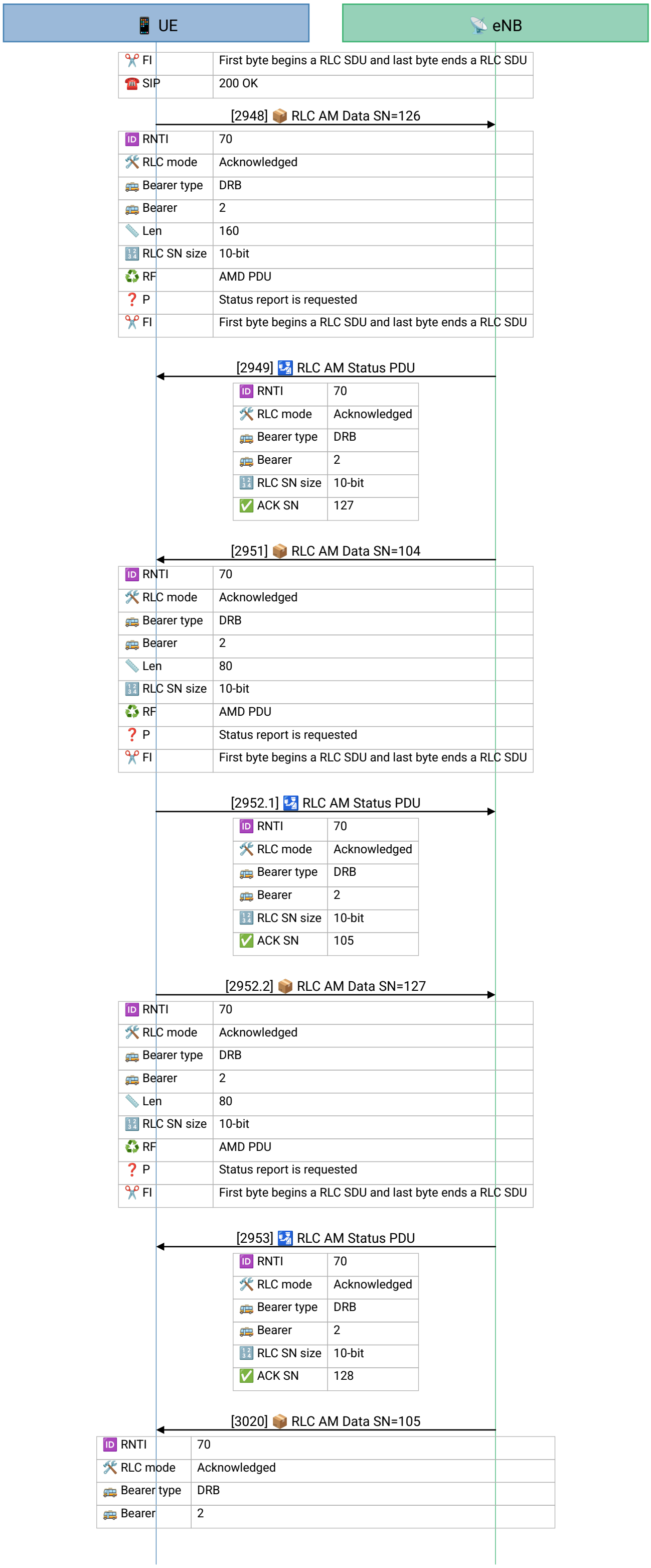
[Frame 2939] `ACK_SN=126` from the eNB.

[Frame 2940.1] Wireshark's "dup SN 102". The eNB sends SN 102 again, in SFN 149 subframe 0: exactly 80 subframes (80 ms) after the original. No Status PDU in the capture carries a NACK, so this is not NACK-driven. The 80 ms matches `t-PollRetransmit`: frame 32 signalled 80 ms for the UE's transmitter; the eNB's own value is not signalled, but this interval fits the same 80 ms.

[Frame 2940.2] The same PDU through the generic data template: `SN=102`, `P = 1`, byte-identical to frame 2924 (same PDCP SN 126, same ESP sequence 78, same bare TCP ACK). Wireshark also flags the ESP sequence as repeated and the TCP segment as a duplicate ACK; both follow from seeing the same packet twice.

[Frame 2941] Why the retransmission was unnecessary. `ACK_SN=103` acknowledges SN 102, and it went up in SFN 148 subframe 8, two subframes *before* the retransmission went down. The capture lists them the other way round, but the radio timing is clear: the UE already had the first copy. Every other uplink Status PDU on this bearer arrives at least 14 ms after the data it acknowledges; this one is logged 2 ms after the retransmission. The UE was held back by `t-StatusProhibit`: 60 ms from frame 2927, then a wait for a grant. Its report and the eNB's 80 ms poll timer crossed on the air, the report going up 2 subframes before the copy came down. The UE discards the duplicate.

[Frame 2945] Downlink `SN=103 : 200` OK to the BYE.



[Frame 2948] Uplink SN=126 : two TCP segments in one PDU, an ACK and the UE's FIN.

[Frame 2949] ACK_SN=127 .

[Frame 2951] Downlink SN=104 : the P-CSCF's FIN.

[Frame 2952.1] ACK_SN=105 : downlink SN 0 to 104.

[Frame 2952.2] Uplink SN=127 , the last TCP ACK.

[Frame 2953] ACK_SN=128 .

[Frame 3020] Act 4 - Logged out of order, delivered in order. 6.5 s later UE 71 calls (call 6). Downlink SN=105 , 3059 bytes, FI = 01: two length indicators (1322, 1322) mark two whole SDUs, then the start of a third. The INVITE arrives as three SDUs in two PDUs.

📱 UE | 📡 eNB

📏 Len	3059
🔢 RLC SN size	10-bit
♻️ RF	AMD PDU
❓ P	Status report is requested
✂️ FI	First byte begins a RLC SDU and last byte does not end a RLC SDU

[3021] 📦 RLC AM Data SN=106

🆔 RNTI	70
🛠️ RLC mode	Acknowledged
🚌 Bearer type	DRB
🚌 Bearer	2
📏 Len	262
🔢 RLC SN size	10-bit
♻️ RF	AMD PDU
❓ P	Status report is requested
✂️ FI	First byte does not begin a RLC SDU and last byte ends a RLC SDU
☎️ SIP	INVITE

[Frame 3021] Downlink `SN=106`, `FI` = 10: the rest of the `INVITE`.

[3022.1] 🔄 RLC AM Status PDU

🆔 RNTI	70
🛠️ RLC mode	Acknowledged
🚌 Bearer type	DRB
🚌 Bearer	2
🔢 RLC SN size	10-bit
✅ ACK SN	107

[Frame 3022.1] `ACK_SN=107`.

[3022.2] 📦 RLC AM Data SN=128

🆔 RNTI	70
🛠️ RLC mode	Acknowledged
🚌 Bearer type	DRB
🚌 Bearer	2
📏 Len	160
🔢 RLC SN size	10-bit
♻️ RF	AMD PDU
❓ P	Status report is requested
✂️ FI	First byte begins a RLC SDU and last byte ends a RLC SDU

[Frame 3022.2] Uplink `SN=128`, two TCP ACKs.

[3023] 🔄 RLC AM Status PDU

🆔 RNTI	70
🛠️ RLC mode	Acknowledged
🚌 Bearer type	DRB
🚌 Bearer	2
🔢 RLC SN size	10-bit
✅ ACK SN	129

[Frame 3023] `ACK_SN=129`.

[3026] 📦 RLC AM Data SN=129

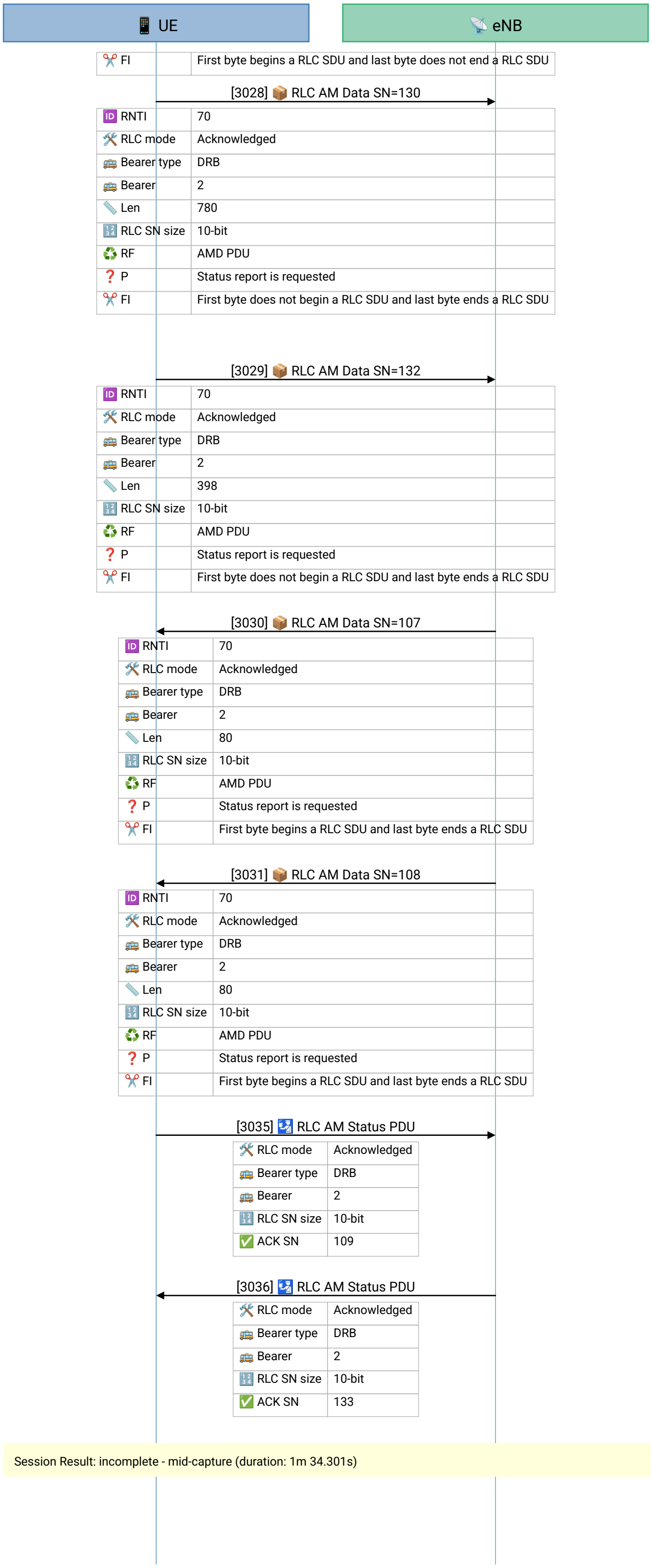
🆔 RNTI	70
🛠️ RLC mode	Acknowledged
🚌 Bearer type	DRB
🚌 Bearer	2
📏 Len	546
🔢 RLC SN size	10-bit
♻️ RF	AMD PDU
❓ P	Status report not requested
✂️ FI	First byte begins a RLC SDU and last byte does not end a RLC SDU

[Frame 3026] Uplink `SN=129`, `FI` = 01: UE 70's reply begins, almost certainly its 183 `Session Progress` (the 183 reaches UE 71 in frame 3039, and UE 70 receives a PRACK for it in frame 3046).

[3027] 📦 RLC AM Data SN=131

🆔 RNTI	70
🛠️ RLC mode	Acknowledged
🚌 Bearer type	DRB
🚌 Bearer	2
📏 Len	804
🔢 RLC SN size	10-bit
♻️ RF	AMD PDU
❓ P	Status report not requested

[Frame 3027] SN 131 before SN 130. Uplink `SN=131`, `FI` = 01, in SFN 811 subframe 8. Wireshark flags "AM SN (130) missing", but nothing is lost.



[Frame 3028] Uplink `SN=130, FI = 10`, in SFN 812 subframe 5, 7 subframes after SN 131. Wireshark now calls it "retransmitted, most likely in response to NACK", yet no NACK exists anywhere in the capture and SN 130 appears only once. The likely cause is MAC HARQ: a first attempt at SN 130 failed and its HARQ retransmission landed after SN 131. LTE's uplink HARQ retransmission comes 8 subframes after the first attempt, which would put that failed attempt in SFN 811 subframe 7, one subframe before SN 131. The eNB's RLC receiver holds SN 131 until SN 130 arrives; out-of-order arrival after HARQ is what its reordering window and `t-Reordering` timer exist for. The capture is FDD (the MAC framing says so), which is where the 8-subframe rule applies. Wireshark's own reassembly does not: it never reassembles these SDUs, so the SIP message stays undecoded here.

[Frame 3029] Uplink `SN=132, FI = 10, P = 1`: the second SDU (SN 131 + 132) is complete.

[Frame 3030] Downlink `SN=107`, a TCP ACK.

[Frame 3031] Downlink `SN=108`, a TCP ACK.

[Frame 3035] `ACK_SN=109.`

[Frame 3036] `ACK_SN=133`, no NACK: the eNB has SN 129 to 132 in order. Across the whole capture this bearer carried 157 uplink and 128 downlink data PDUs and 166 Status PDUs. None of them carried a NACK, and only one PDU was ever sent twice.

Session Result: incomplete - mid-capture (duration: 1m 34.301s)