3GPP TS 36.322 V11.0.0 (2012-09)

Technical Specification

3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA); Radio Link Control (RLC) protocol specification (Release 11)

The present document has been developed within the 3rd Generation Partnership Project (3GPP TM) and may be further elaborated for the purposes of 3GPP.
The present document has not been subject to any approval process by the 3GPP Organisational Partners and shall not be implemented.
This Specification is provided for future development work within 3GPP only. The Organisational Partners accept no liability for any use of this Specification.
Specifications and reports for implementation of the 3GPP TM system should be obtained via the 3GPP Organisational Partners' Publications Offices.

ONEPLUS - EX. 1013
IPR2025-00762

Keywords
UMTS, radio

3GPP

Postal address

3GPP support office address
650 Route des Lucioles - Sophia Antipolis
Valbonne - FRANCE
Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Internet
http://www.3gpp.org

Copyright Notification

No part may be reproduced except as authorized by written permission.
The copyright and the foregoing restriction extend to reproduction in all media.

© 2012, 3GPP Organizational Partners (ARIB, ATIS, CCSA, ETSI, TTA, TTC).
All rights reserved.

UMTS™ is a Trade Mark of ETSI registered for the benefit of its members
3GPP™ is a Trade Mark of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners
LTE™ is a Trade Mark of ETSI currently being registered for the benefit of its Members and of the 3GPP Organizational Partners
GSM® and the GSM logo are registered and owned by the GSM Association

ONEPLUS - EX. 1013

Contents

Foreword....5
1 Scope....6
2 References....6
3 Definitions, symbols and abbreviations....6
3.1 Definitions....6
3.2 Abbreviations....6
4 General....7
4.1 Introduction....7
4.2 RLC architecture....7
4.2.1 RLC entities....7
4.2.1.1 TM RLC entity....9
4.2.1.1.1 General....9
4.2.1.1.2 Transmitting TM RLC entity....9
4.2.1.1.3 Receiving TM RLC entity....9
4.2.1.2 UM RLC entity....9
4.2.1.2.1 General....9
4.2.1.2.2 Transmitting UM RLC entity....10
4.2.1.2.3 Receiving UM RLC entity....10
4.2.1.3 AM RLC entity....11
4.2.1.3.1 General....11
4.2.1.3.2 Transmitting side....11
4.2.1.3.3 Receiving side....12
4.3 Services....12
4.3.1 Services provided to upper layers....12
4.3.2 Services expected from lower layers....12
4.4 Functions....12
4.5 Data available for transmission....13
5 Procedures....13
5.1 Data transfer procedures....13
5.1.1 TM data transfer....13
5.1.1.1 Transmit operations....13
5.1.1.1.1 General....13
5.1.1.2 Receive operations....13
5.1.1.2.1 General....13
5.1.2 UM data transfer....13
5.1.2.1 Transmit operations....13
5.1.2.1.1 General....13
5.1.2.2 Receive operations....13
5.1.2.2.1 General....13
5.1.2.2.2 Actions when an UMD PDU is received from lower layer....14
5.1.2.2.3 Actions when an UMD PDU is placed in the reception buffer....14
5.1.2.2.4 Actions when *t-Reordering* expires....15
5.1.3 AM data transfer....15
5.1.3.1 Transmit operations....15
5.1.3.1.1 General....15
5.1.3.2 Receive operations....15
5.1.3.2.1 General....15
5.1.3.2.2 Actions when a RLC data PDU is received from lower layer....16
5.1.3.2.3 Actions when a RLC data PDU is placed in the reception buffer....16
5.1.3.2.4 Actions when *t-Reordering* expires....17
5.2 ARQ procedures....17
5.2.1 Retransmission....17
5.2.2 Polling....18
5.2.2.1 Transmission of a AMD PDU or AMD PDU segment....18
5.2.2.2 Reception of a STATUS report....18

ONEPLUS - EX. 1013

5.2.2.3 Expiry of *t-PollRetransmit* ... 19
5.2.3 Status reporting ... 19
5.3 SDU discard procedures ... 20
5.4 Re-establishment procedure ... 20
5.5 Handling of unknown, unforeseen and erroneous protocol data ... 21
5.5.1 Reception of PDU with reserved or invalid values ... 21

6 Protocol data units, formats and parameters ... 21
6.1 Protocol data units ... 21
6.1.1 RLC data PDU ... 21
6.1.2 RLC control PDU ... 21
6.2 Formats and parameters ... 21
6.2.1 Formats ... 21
6.2.1.1 General ... 21
6.2.1.2 TMD PDU ... 22
6.2.1.3 UMD PDU ... 22
6.2.1.4 AMD PDU ... 24
6.2.1.5 AMD PDU segment ... 25
6.2.1.6 STATUS PDU ... 27
6.2.2 Parameters ... 27
6.2.2.1 General ... 27
6.2.2.2 Data field ... 28
6.2.2.3 Sequence Number (SN) field ... 28
6.2.2.4 Extension bit (E) field ... 28
6.2.2.5 Length Indicator (LI) field ... 28
6.2.2.6 Framing Info (FI) field ... 29
6.2.2.7 Segment Offset (SO) field ... 29
6.2.2.8 Last Segment Flag (LSF) field ... 29
6.2.2.9 Data/Control (D/C) field ... 29
6.2.2.10 Re-segmentation Flag (RF) field ... 29
6.2.2.11 Polling bit (P) field ... 30
6.2.2.12 Reserved 1 (R1) field ... 30
6.2.2.13 Control PDU Type (CPT) field ... 30
6.2.2.14 Acknowledgement SN (ACK_SN) field ... 30
6.2.2.15 Extension bit 1 (E1) field ... 30
6.2.2.16 Negative Acknowledgement SN (NACK_SN) field ... 31
6.2.2.17 Extension bit 2 (E2) field ... 31
6.2.2.18 SO start (SOstart) field ... 31
6.2.2.19 SO end (SOend) field ... 31

7 Variables, constants and timers ... 31
7.1 State variables ... 31
7.2 Constants ... 33
7.3 Timers ... 33
7.4 Configurable parameters ... 34

Annex A (informative): Change history ... 35

ONEPLUS - EX. 1013

Foreword

This Technical Specification has been produced by the 3rd Generation Partnership Project (3GPP).

The contents of the present document are subject to continuing work within the TSG and may change following formal TSG approval. Should the TSG modify the contents of the present document, it will be re-released by the TSG with an identifying change of release date and an increase in version number as follows:

Version x.y.z

where:

- x the first digit:
  - 1 presented to TSG for information;
  - 2 presented to TSG for approval;
  - 3 or greater indicates TSG approved document under change control.
- y the second digit is incremented for all changes of substance, i.e. technical enhancements, corrections, updates, etc.
- z the third digit is incremented when editorial only changes have been incorporated in the document.

ONEPLUS - EX. 1013

1 Scope

The present document specifies the E-UTRA Radio Link Control (RLC) protocol for the UE – E-UTRAN radio interface.

The specification describes:

- E-UTRA RLC sublayer architecture;
- E-UTRA RLC entities;
- services expected from lower layers by E-UTRA RLC;
- services provided to upper layers by E-UTRA RLC;
- E-UTRA RLC functions;
- elements for peer-to-peer E-UTRA RLC communication including protocol data units, formats and parameters;
- handling of unknown, unforeseen and erroneous protocol data at E-UTRA RLC.

2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication, edition number, version number, etc.) or non-specific.
- For a specific reference, subsequent revisions do not apply.
- For a non-specific reference, the latest version applies. In the case of a reference to a 3GPP document (including a GSM document), a non-specific reference implicitly refers to the latest version of that document *in the same Release as the present document*.

[1] 3GPP TR 21.905: "Vocabulary for 3GPP Specifications".

[2] 3GPP TS 36.300: "E-UTRA and E-UTRAN Overall Description; Stage 2".

[3] 3GPP TS 36.321: "E-UTRA MAC protocol specification".

[4] 3GPP TS 36.323: "E-UTRA PDCP specification".

[5] 3GPP TS 36.331: "E-UTRA RRC Protocol specification".

3 Definitions, symbols and abbreviations

3.1 Definitions

For the purposes of the present document, the terms and definitions given in TR 21.905 [1] and the following apply. A term defined in the present document takes precedence over the definition of the same term, if any, in TR 21.905 [1].

byte segment: A byte of the Data field of an AMD PDU. Specifically, byte segment number 0 corresponds to the first byte of the Data field of an AMD PDU.

Data field element: An RLC SDU or an RLC SDU segment that is mapped to the Data field.

RLC SDU segment: A segment of an RLC SDU.

3.2 Abbreviations

For the purposes of the present document, the following abbreviations apply:

ONEPLUS - EX. 1013

AM	Acknowledged Mode
AMD	AM Data
ARQ	Automatic Repeat reQuest
BCCH	Broadcast Control CHannel
BCH	Broadcast CHannel
CCCH	Common Control CHannel
DCCH	Dedicated Control CHannel
DL	DownLink
DL-SCH	DL-Shared CHannel
DTCH	Dedicated Traffic CHannel
E	Extension bit
eNB	E-UTRAN Node B
E-UTRA	Evolved UMTS Terrestrial Radio Access
E-UTRAN	Evolved UMTS Terrestrial Radio Access Network
FI	Framing Info
HARQ	Hybrid ARQ
LI	Length Indicator
LSF	Last Segment Flag
MAC	Medium Access Control
MCCH	Multicast Control Channel
MTCH	Multicast Traffic Channel
PCCH	Paging Control CHannel
PDU	Protocol Data Unit
RLC	Radio Link Control
RRC	Radio Resource Control
SAP	Service Access Point
SDU	Service Data Unit
SN	Sequence Number
SO	Segment Offset
TB	Transport Block
TM	Transparent Mode
TMD	TM Data
UE	User Equipment
UL	UpLink
UM	Unacknowledged Mode
UMD	UM Data

4 General

4.1 Introduction

The objective is to describe the RLC architecture and the RLC entities from a functional point of view.

4.2 RLC architecture

4.2.1 RLC entities

The description in this sub clause is a model and does not specify or restrict implementations.

RRC is generally in control of the RLC configuration.

Functions of the RLC sub layer are performed by RLC entities. For a RLC entity configured at the eNB, there is a peer RLC entity configured at the UE and vice versa.

An RLC entity receives/delivers RLC SDUs from/to upper layer and sends/receives RLC PDUs to/from its peer RLC entity via lower layers. An RLC PDU can either be a RLC data PDU (see sub clause 6.1.1) or a RLC control PDU (see sub clause 6.1.2). If an RLC entity receives RLC SDUs from upper layer, it receives them through a single SAP between RLC and upper layer, and after forming RLC data PDUs from the received RLC SDUs, the RLC entity delivers the RLC data PDUs to lower layer through a single logical channel. If an RLC entity receives RLC data PDUs from lower layer, it receives them through a single logical channel, and after forming RLC SDUs from the received RLC data PDUs, the RLC entity delivers the RLC SDUs to upper layer through a single SAP between RLC and upper

ONEPLUS - EX. 1013

layer. If an RLC entity delivers/receives RLC control PDUs to/from lower layer, it delivers/receives them through the same logical channel it delivers/receives the RLC data PDUs through.

An RLC entity can be configured to perform data transfer in one of the following three modes: Transparent Mode (TM), Unacknowledged Mode (UM) or Acknowledged Mode (AM). Consequently, an RLC entity is categorized as a TM RLC entity, an UM RLC entity or an AM RLC entity depending on the mode of data transfer that the RLC entity is configured to provide.

A TM RLC entity is configured either as a transmitting TM RLC entity or a receiving TM RLC entity. The transmitting TM RLC entity receives RLC SDUs from upper layer and sends RLC PDUs to its peer receiving TM RLC entity via lower layers. The receiving TM RLC entity delivers RLC SDUs to upper layer and receives RLC PDUs from its peer transmitting TM RLC entity via lower layers.

An UM RLC entity is configured either as a transmitting UM RLC entity or a receiving UM RLC entity. The transmitting UM RLC entity receives RLC SDUs from upper layer and sends RLC PDUs to its peer receiving UM RLC entity via lower layers. The receiving UM RLC entity delivers RLC SDUs to upper layer and receives RLC PDUs from its peer transmitting UM RLC entity via lower layers.

An AM RLC entity consists of a transmitting side and a receiving side. The transmitting side of an AM RLC entity receives RLC SDUs from upper layer and sends RLC PDUs to its peer AM RLC entity via lower layers. The receiving side of an AM RLC entity delivers RLC SDUs to upper layer and receives RLC PDUs from its peer AM RLC entity via lower layers.

Figure 1 illustrates the overview model of the RLC sub layer.

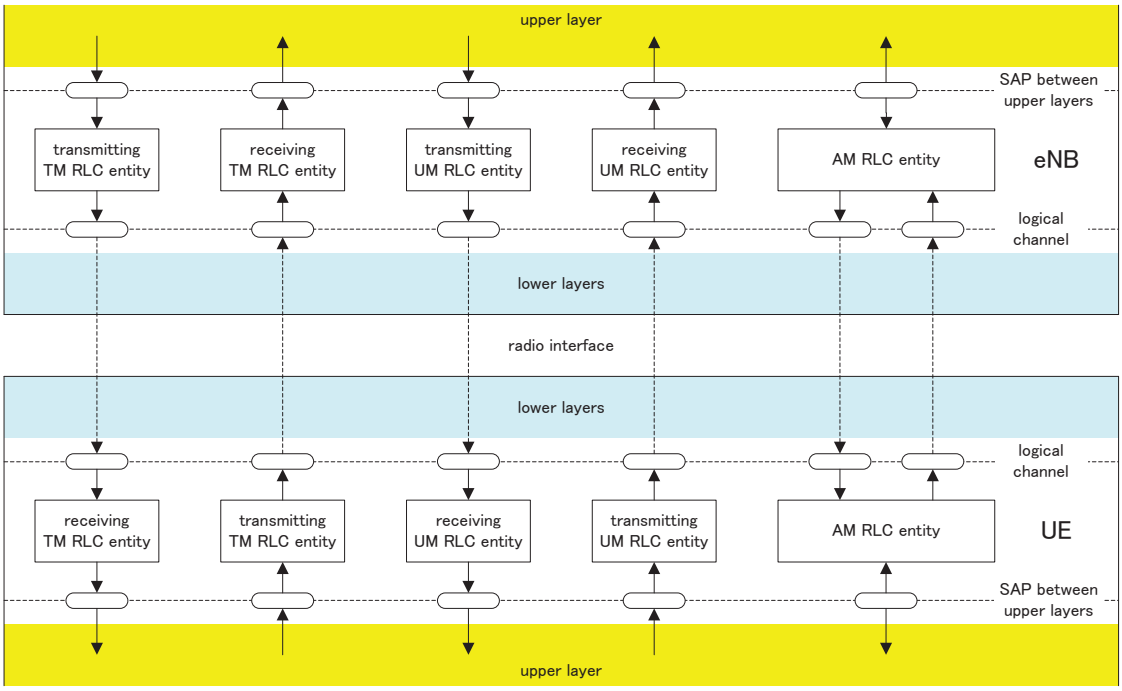

Figure 4.2.1-1: Overview model of the RLC sub layer

The following applies to all RLC entity types (i.e. TM, UM and AM RLC entity):

- RLC SDUs of variable sizes which are byte aligned (i.e. multiple of 8 bits) are supported;
- RLC PDUs are formed only when a transmission opportunity has been notified by lower layer (i.e. by MAC) and are then delivered to lower layer.

Description of different RLC entity types are provided below.

ONEPLUS - EX. 1013

4.2.1.1 TM RLC entity

4.2.1.1.1 General

A TM RLC entity can be configured to deliver/receive RLC PDUs through the following logical channels:

- BCCH, DL/UL CCCH and PCCH.

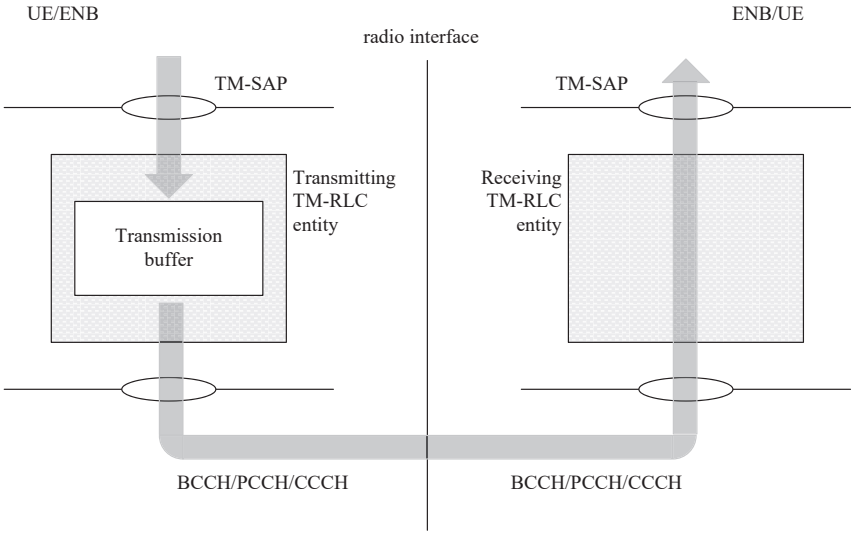

Figure 4.2.1.1.1-1: Model of two transparent mode peer entities

A TM RLC entity delivers/receives the following RLC data PDU:

- TMD PDU.

4.2.1.1.2 Transmitting TM RLC entity

When a transmitting TM RLC entity forms TMD PDUs from RLC SDUs, it shall:

- not segment nor concatenate the RLC SDUs;
- not include any RLC headers in the TMD PDUs.

4.2.1.1.3 Receiving TM RLC entity

When a receiving TM RLC entity receives TMD PDUs, it shall:

- deliver the TMD PDUs (which are just RLC SDUs) to upper layer.

4.2.1.2 UM RLC entity

4.2.1.2.1 General

An UM RLC entity can be configured to deliver/receive RLC PDUs through the following logical channels:

- DL/UL DTCH, MCCH or MTCH.

ONEPLUS - EX. 1013

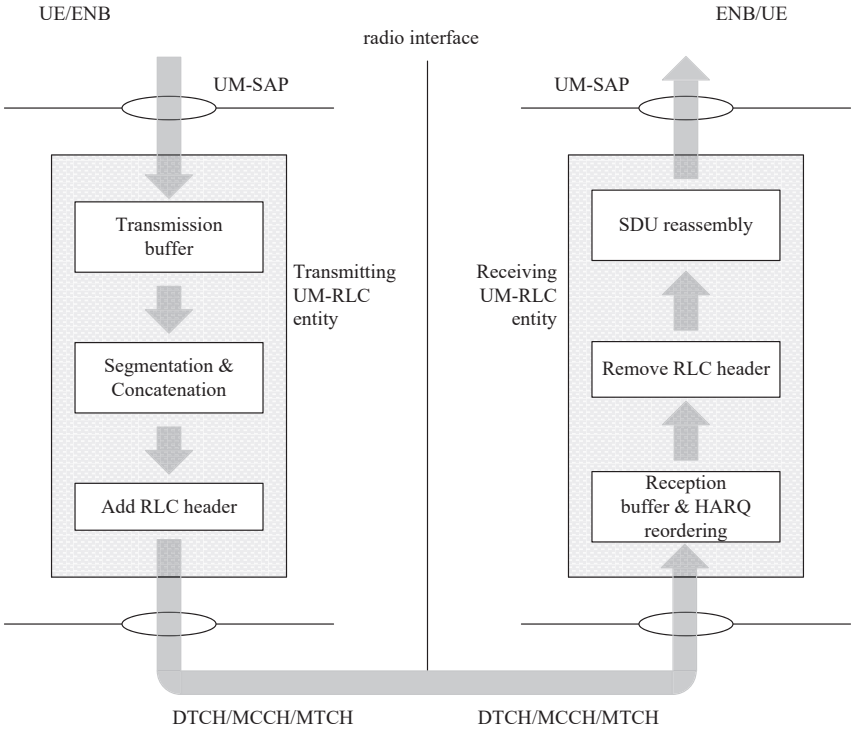

Figure 4.2.1.2.1-1: Model of two unacknowledged mode peer entities

An UM RLC entity delivers/receives the following RLC data PDU:

- UMD PDU.

NOTE: HARQ reordering is not applicable for MCCH or MTCH reception.

4.2.1.2.2 Transmitting UM RLC entity

When a transmitting UM RLC entity forms UMD PDUs from RLC SDUs, it shall:

- segment and/or concatenate the RLC SDUs so that the UMD PDUs fit within the total size of RLC PDU(s) indicated by lower layer at the particular transmission opportunity notified by lower layer;
- include relevant RLC headers in the UMD PDU.

4.2.1.2.3 Receiving UM RLC entity

When a receiving UM RLC entity receives UMD PDUs, it shall:

- detect whether or not the UMD PDUs have been received in duplication, and discard duplicated UMD PDUs;
- reorder the UMD PDUs if they are received out of sequence;
- detect the loss of UMD PDUs at lower layers and avoid excessive reordering delays;
- reassemble RLC SDUs from the reordered UMD PDUs (not accounting for RLC PDUs for which losses have been detected) and deliver the RLC SDUs to upper layer in ascending order of the RLC SN;
- discard received UMD PDUs that cannot be re-assembled into a RLC SDU due to loss at lower layers of an UMD PDU which belonged to the particular RLC SDU.

At the time of RLC re-establishment, the receiving UM RLC entity shall:

- if possible, reassemble RLC SDUs from the UMD PDUs that are received out of sequence and deliver them to upper layer;

ONEPLUS - EX. 1013

- discard any remaining UMD PDUs that could not be reassembled into RLC SDUs;
- initialize relevant state variables and stop relevant timers.

4.2.1.3 AM RLC entity

4.2.1.3.1 General

An AM RLC entity can be configured to deliver/receive RLC PDUs through the following logical channels:

- DL/UL DCCH or DL/UL DTCH.

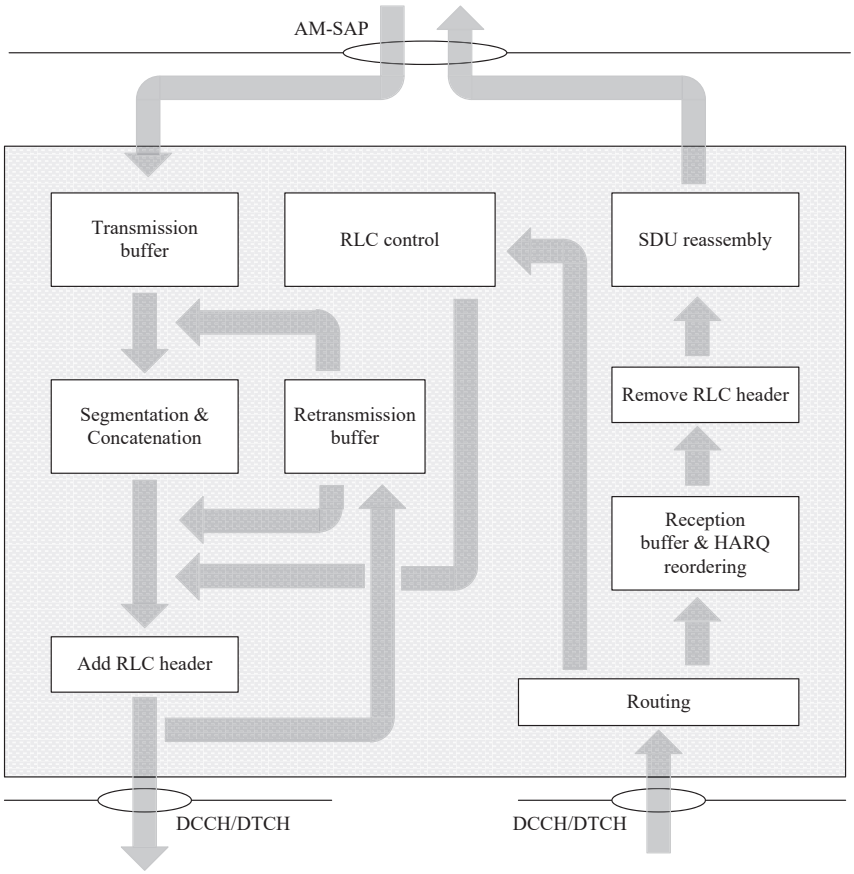

Figure 4.2.1.3.1-1: Model of an acknowledged mode enttiy

An AM RLC entity delivers/receives the following RLC data PDUs:

- AMD PDU;
- AMD PDU segment.

An AM RLC entity delivers/receives the following RLC control PDU:

- STATUS PDU.

4.2.1.3.2 Transmitting side

When the transmitting side of an AM RLC entity forms AMD PDUs from RLC SDUs, it shall:

- segment and/or concatenate the RLC SDUs so that the AMD PDUs fit within the total size of RLC PDU(s) indicated by lower layer at the particular transmission opportunity notified by lower layer.

The transmitting side of an AM RLC entity supports retransmission of RLC data PDUs (ARQ):

ONEPLUS - EX. 1013

- if the RLC data PDU to be retransmitted does not fit within the total size of RLC PDU(s) indicated by lower layer at the particular transmission opportunity notified by lower layer, the AM RLC entity can re-segment the RLC data PDU into AMD PDU segments;
- the number of re-segmentation is not limited.

When the transmitting side of an AM RLC entity forms AMD PDUs from RLC SDUs received from upper layer or AMD PDU segments from RLC data PDUs to be retransmitted, it shall:

- include relevant RLC headers in the RLC data PDU.

4.2.1.3.3 Receiving side

When the receiving side of an AM RLC entity receives RLC data PDUs, it shall:

- detect whether or not the RLC data PDUs have been received in duplication, and discard duplicated RLC data PDUs;
- reorder the RLC data PDUs if they are received out of sequence;
- detect the loss of RLC data PDUs at lower layers and request retransmissions to its peer AM RLC entity;
- reassemble RLC SDUs from the reordered RLC data PDUs and deliver the RLC SDUs to upper layer in sequence.

At the time of RLC re-establishment, the receiving side of an AM RLC entity shall:

- if possible, reassemble RLC SDUs from the RLC data PDUs that are received out of sequence and deliver them to upper layer;
- discard any remaining RLC data PDUs that could not be reassembled into RLC SDUs;
- initialize relevant state variables and stop relevant timers.

4.3 Services

4.3.1 Services provided to upper layers

The following services are provided by RLC to upper layer:

- TM data transfer;
- UM data transfer;
- AM data transfer, including indication of successful delivery of upper layers PDUs.

4.3.2 Services expected from lower layers

The following services are expected by RLC from lower layer (i.e. MAC):

- data transfer;
- notification of a transmission opportunity, together with the total size of the RLC PDU(s) to be transmitted in the transmission opportunity.

4.4 Functions

The following functions are supported by the RLC sub layer:

- transfer of upper layer PDUs;
- error correction through ARQ (only for AM data transfer);
- concatenation, segmentation and reassembly of RLC SDUs (only for UM and AM data transfer);
- re-segmentation of RLC data PDUs (only for AM data transfer);

ONEPLUS - EX. 1013

- reordering of RLC data PDUs (only for UM and AM data transfer);
- duplicate detection (only for UM and AM data transfer);
- RLC SDU discard (only for UM and AM data transfer);
- RLC re-establishment;
- Protocol error detection (only for AM data transfer).

4.5 Data available for transmission

For the purpose of MAC buffer status reporting, the UE shall consider the following as data available for transmission in the RLC layer:

- RLC SDUs, or segments thereof, that have not yet been included in an RLC data PDU;
- RLC data PDUs, or portions thereof, that are pending for retransmission (RLC AM).

In addition, if a STATUS PDU has been triggered and *t-StatusProhibit* is not running or has expired, the UE shall estimate the size of the STATUS PDU that will be transmitted in the next transmission opportunity, and consider this as data available for transmission in the RLC layer.

5 Procedures

5.1 Data transfer procedures

5.1.1 TM data transfer

5.1.1.1 Transmit operations

5.1.1.1.1 General

When submitting a new TMD PDU to lower layer, the transmitting TM RLC entity shall:

- submit a RLC SDU without any modification to lower layer.

5.1.1.2 Receive operations

5.1.1.2.1 General

When receiving a new TMD PDU from lower layer, the receiving TM RLC entity shall:

- deliver the TMD PDU without any modification to upper layer.

5.1.2 UM data transfer

5.1.2.1 Transmit operations

5.1.2.1.1 General

When delivering a new UMD PDU to lower layer, the transmitting UM RLC entity shall:

- set the SN of the UMD PDU to VT(US), and then increment VT(US) by one.

5.1.2.2 Receive operations

5.1.2.2.1 General

The receiving UM RLC entity shall maintain a reordering window according to state variable VR(UH) as follows:

- a SN falls within the reordering window if (VR(UH) – UM_Window_Size) <= SN < VR(UH);
- a SN falls outside of the reordering window otherwise.

ONEPLUS - EX. 1013

When receiving an UMD PDU from lower layer, the receiving UM RLC entity shall:

- either discard the received UMD PDU or place it in the reception buffer (see sub clause 5.1.2.2.2);
- if the received UMD PDU was placed in the reception buffer:
  - update state variables, reassemble and deliver RLC SDUs to upper layer and start/stop *t-Reordering* as needed (see sub clause 5.1.2.2.3);

When *t-Reordering* expires, the receiving UM RLC entity shall:

- update state variables, reassemble and deliver RLC SDUs to upper layer and start *t-Reordering* as needed (see sub clause 5.1.2.2.4).

5.1.2.2.2 Actions when an UMD PDU is received from lower layer

When an UMD PDU with SN = x is received from lower layer, the receiving UM RLC entity shall:

- if VR(UR) < x < VR(UH) and the UMD PDU with SN = x has been received before; or
- if (VR(UH) – UM_Window_Size) <= x < VR(UR):
  - discard the received UMD PDU;
- else:
  - place the received UMD PDU in the reception buffer.

5.1.2.2.3 Actions when an UMD PDU is placed in the reception buffer

When an UMD PDU with SN = x is placed in the reception buffer, the receiving UM RLC entity shall:

- if x falls outside of the reordering window:
  - update VR(UH) to x + 1;
  - reassemble RLC SDUs from any UMD PDUs with SN that falls outside of the reordering window, remove RLC headers when doing so and deliver the reassembled RLC SDUs to upper layer in ascending order of the RLC SN if not delivered before;
  - if VR(UR) falls outside of the reordering window:
    - set VR(UR) to (VR(UH) – UM_Window_Size);
- if the reception buffer contains an UMD PDU with SN = VR(UR):
  - update VR(UR) to the SN of the first UMD PDU with SN > current VR(UR) that has not been received;
  - reassemble RLC SDUs from any UMD PDUs with SN < updated VR(UR), remove RLC headers when doing so and deliver the reassembled RLC SDUs to upper layer in ascending order of the RLC SN if not delivered before;
- if *t-Reordering* is running:
  - if VR(UX) <= VR(UR); or
  - if VR(UX) falls outside of the reordering window and VR(UX) is not equal to VR(UH)::
    - stop and reset *t-Reordering*;
- if *t-Reordering* is not running (includes the case when *t-Reordering* is stopped due to actions above):
  - if VR(UH) > VR(UR):
    - start *t-Reordering*;
    - set VR(UX) to VR(UH).

ONEPLUS - EX. 1013

5.1.2.2.4 Actions when *t-Reordering* expires

When *t-Reordering* expires, the receiving UM RLC entity shall:

- update VR(UR) to the SN of the first UMD PDU with SN >= VR(UX) that has not been received;
- reassemble RLC SDUs from any UMD PDUs with SN < updated VR(UR), remove RLC headers when doing so and deliver the reassembled RLC SDUs to upper layer in ascending order of the RLC SN if not delivered before;
- if VR(UH) > VR(UR):
  - start *t-Reordering*;
  - set VR(UX) to VR(UH).

5.1.3 AM data transfer

5.1.3.1 Transmit operations

5.1.3.1.1 General

The transmitting side of an AM RLC entity shall prioritize transmission of RLC control PDUs over RLC data PDUs. The transmitting side of an AM RLC entity shall prioritize retransmission of RLC data PDUs over transmission of new AMD PDUs.

The transmitting side of an AM RLC entity shall maintain a transmitting window according to state variables VT(A) and VT(MS) as follows:

- a SN falls within the transmitting window if VT(A) <= SN < VT(MS);
- a SN falls outside of the transmitting window otherwise.

The transmitting side of an AM RLC entity shall not deliver to lower layer any RLC data PDU whose SN falls outside of the transmitting window.

When delivering a new AMD PDU to lower layer, the transmitting side of an AM RLC entity shall:

- set the SN of the AMD PDU to VT(S), and then increment VT(S) by one.

The transmitting side of an AM RLC entity can receive a positive acknowledgement (confirmation of successful reception by its peer AM RLC entity) for a RLC data PDU by the following:

- STATUS PDU from its peer AM RLC entity.

When receiving a positive acknowledgement for an AMD PDU with SN = VT(A), the transmitting side of an AM RLC entity shall:

- set VT(A) equal to the SN of the AMD PDU with the smallest SN, whose SN falls within the range VT(A) <= SN <= VT(S) and for which a positive acknowledgment has not been received yet.
- if positive acknowledgements have been received for all AMD PDUs associated with a transmitted RLC SDU:
  - send an indication to the upper layers of successful delivery of the RLC SDU.

5.1.3.2 Receive operations

5.1.3.2.1 General

The receiving side of an AM RLC entity shall maintain a receiving window according to state variables VR(R) and VR(MR) as follows:

- a SN falls within the receiving window if VR(R) <= SN < VR(MR);
- a SN falls outside of the receiving window otherwise.

When receiving a RLC data PDU from lower layer, the receiving side of an AM RLC entity shall:

ONEPLUS - EX. 1013

- either discard the received RLC data PDU or place it in the reception buffer (see sub clause 5.1.3.2.2);
- if the received RLC data PDU was placed in the reception buffer:
  - update state variables, reassemble and deliver RLC SDUs to upper layer and start/stop *t-Reordering* as needed (see sub clause 5.1.3.2.3).

When *t-Reordering* expires, the receiving side of an AM RLC entity shall:

- update state variables and start *t-Reordering* as needed (see sub clause 5.1.3.2.4).

5.1.3.2.2 Actions when a RLC data PDU is received from lower layer

When a RLC data PDU is received from lower layer, where the RLC data PDU contains byte segment numbers y to z of an AMD PDU with SN = x, the receiving side of an AM RLC entity shall:

- if x falls outside of the receiving window; or
- if byte segment numbers y to z of the AMD PDU with SN = x have been received before:
  - discard the received RLC data PDU;
- else:
  - place the received RLC data PDU in the reception buffer;
  - if some byte segments of the AMD PDU contained in the RLC data PDU have been received before:
    - discard the duplicate byte segments.

5.1.3.2.3 Actions when a RLC data PDU is placed in the reception buffer

When a RLC data PDU with SN = x is placed in the reception buffer, the receiving side of an AM RLC entity shall:

- if x >= VR(H)
  - update VR(H) to x+ 1;
- if all byte segments of the AMD PDU with SN = VR(MS) are received:
  - update VR(MS) to the SN of the first AMD PDU with SN > current VR(MS) for which not all byte segments have been received;
- if x = VR(R):
  - if all byte segments of the AMD PDU with SN = VR(R) are received:
    - update VR(R) to the SN of the first AMD PDU with SN > current VR(R) for which not all byte segments have been received;
    - update VR(MR) to the updated VR(R) + AM_Window_Size;
  - reassemble RLC SDUs from any byte segments of AMD PDUs with SN that falls outside of the receiving window and in-sequence byte segments of the AMD PDU with SN = VR(R), remove RLC headers when doing so and deliver the reassembled RLC SDUs to upper layer in sequence if not delivered before;
- if *t-Reordering* is running:
  - if VR(X) = VR(R); or
  - if VR(X) falls outside of the receiving window and VR(X) is not equal to VR(MR):
    - stop and reset *t-Reordering*;
- if *t-Reordering* is not running (includes the case *t-Reordering* is stopped due to actions above):
  - if VR (H) > VR(R):

ONEPLUS - EX. 1013

- start *t-Reordering*;
- set VR(X) to VR(H).

5.1.3.2.4 Actions when *t-Reordering* expires

When *t-Reordering* expires, the receiving side of an AM RLC entity shall:

- update VR(MS) to the SN of the first AMD PDU with SN >= VR(X) for which not all byte segments have been received;
- if VR(H) > VR(MS):
  - start *t-Reordering*;
  - set VR(X) to VR(H).

5.2 ARQ procedures

ARQ procedures are only performed by an AM RLC entity.

5.2.1 Retransmission

The transmitting side of an AM RLC entity can receive a negative acknowledgement (notification of reception failure by its peer AM RLC entity) for an AMD PDU or a portion of an AMD PDU by the following:

- STATUS PDU from its peer AM RLC entity.

When receiving a negative acknowledgement for an AMD PDU or a portion of an AMD PDU by a STATUS PDU from its peer AM RLC entity, the transmitting side of the AM RLC entity shall:

- if the SN of the corresponding AMD PDU falls within the range VT(A) <= SN < VT(S):
  - consider the AMD PDU or the portion of the AMD PDU for which a negative acknowledgement was received for retransmission.

When an AMD PDU or a portion of an AMD PDU is considered for retransmission, the transmitting side of the AM RLC entity shall:

- if the AMD PDU is considered for retransmission for the first time:
  - set the RETX_COUNT associated with the AMD PDU to zero;
- else, if it (the AMD PDU or the portion of the AMD PDU that is considered for retransmission) is not pending for retransmission already, or a portion of it is not pending for retransmission already:
  - increment the RETX_COUNT;
- if RETX_COUNT = *maxRetxThreshold*:
  - indicate to upper layers that max retransmission has been reached.

When retransmitting an AMD PDU, the transmitting side of an AM RLC entity shall:

- if the AMD PDU can entirely fit within the total size of RLC PDU(s) indicated by lower layer at the particular transmission opportunity:
  - deliver the AMD PDU as it is except for the P field (the P field should be set according to sub clause 5.2.2) to lower layer;
- otherwise:
  - segment the AMD PDU, form a new AMD PDU segment which will fit within the total size of RLC PDU(s) indicated by lower layer at the particular transmission opportunity and deliver the new AMD PDU segment to lower layer.

When retransmitting a portion of an AMD PDU, the transmitting side of an AM RLC entity shall:

ONEPLUS - EX. 1013

- segment the portion of the AMD PDU as necessary, form a new AMD PDU segment which will fit within the total size of RLC PDU(s) indicated by lower layer at the particular transmission opportunity and deliver the new AMD PDU segment to lower layer.

When forming a new AMD PDU segment, the transmitting side of an AM RLC entity shall:

- only map the Data field of the original AMD PDU to the Data field of the new AMD PDU segment;
- set the header of the new AMD PDU segment in accordance with the description in sub clause 6.;
- set the P field according to sub clause 5.2.2.

5.2.2 Polling

An AM RLC entity can poll its peer AM RLC entity in order to trigger STATUS reporting at the peer AM RLC entity.

5.2.2.1 Transmission of a AMD PDU or AMD PDU segment

Upon assembly of a new AMD PDU, the transmitting side of an AM RLC entity shall:

- increment PDU_WITHOUT_POLL by one;
- increment BYTE_WITHOUT_POLL by every new byte of Data field element that it maps to the Data field of the RLC data PDU;
- if PDU_WITHOUT_POLL >= *pollPDU*; or
- if BYTE_WITHOUT_POLL >= *pollByte*;
  - include a poll in the RLC data PDU as described below.

Upon assembly of an AMD PDU or AMD PDU segment, the transmitting side of an AM RLC entity shall:

- if both the transmission buffer and the retransmission buffer becomes empty (excluding transmitted RLC data PDU awaiting for acknowledgements) after the transmission of the RLC data PDU; or
- if no new RLC data PDU can be transmitted after the transmission of the RLC data PDU (e.g. due to window stalling);
  - include a poll in the RLC data PDU as described below.

To include a poll in a RLC data PDU, the transmitting side of an AM RLC entity shall:

- set the P field of the RLC data PDU to "1";
- set PDU_WITHOUT_POLL to 0;
- set BYTE_WITHOUT_POLL to 0;

After delivering a RLC data PDU including a poll to lower layer and after incrementing of VT(S) if necessary, the transmitting side of an AM RLC entity shall:

- set POLL_SN to VT(S) – 1;
- if *t-PollRetransmit* is not running:
  - start *t-PollRetransmit*;
- else:
  - restart *t-PollRetransmit*;

5.2.2.2 Reception of a STATUS report

Upon reception of a STATUS report from the receiving RLC AM entity the transmitting side of an AM RLC entity shall:

ONEPLUS - EX. 1013

- if the STATUS report comprises a positive or negative acknowledgement for the RLC data PDU with sequence number equal to POLL_SN:
  - if *t-PollRetransmit* is running:
    - stop and reset *t-PollRetransmit*.

5.2.2.3 Expiry of *t-PollRetransmit*

Upon expiry of *t-PollRetransmit*, the transmitting side of an AM RLC entity shall:

- if both the transmission buffer and the retransmission buffer are empty (excluding transmitted RLC data PDU awaiting for acknowledgements); or
- if no new RLC data PDU can be transmitted (e.g. due to window stalling):
  - consider the AMD PDU with SN = VT(S) – 1 for retransmission; or
  - consider any AMD PDU which has not been positively acknowledged for retransmission;
- include a poll in a RLC data PDU as described in section 5.2.2.1.

5.2.3 Status reporting

An AM RLC entity sends STATUS PDUs to its peer AM RLC entity in order to provide positive and/or negative acknowledgements of RLC PDUs (or portions of them).

RRC configures whether or not the status prohibit function is to be used for an AM RLC entity.

Triggers to initiate STATUS reporting include:

- Polling from its peer AM RLC entity:
  - When a RLC data PDU with SN = x and the P field set to “1” is received from lower layer, the receiving side of an AM RLC entity shall:
    - if the PDU is to be discarded as specified in subclause 5.1.3.2.2; or
    - if x < VR(MS) or x >= VR(MR):
      - trigger a STATUS report;
    - else:
      - delay triggering the STATUS report until x < VR(MS) or x >= VR(MR).

NOTE 1: This ensures that the RLC Status report is transmitted after HARQ reordering.

- Detection of reception failure of an RLC data PDU:
  - The receiving side of an AM RLC entity shall trigger a STATUS report when *t-Reordering* expires.

NOTE 2: The expiry of *t-Reordering* triggers both VR(MS) to be updated and a STATUS report to be triggered, but the STATUS report shall be triggered after VR(MS) is updated.

When STATUS reporting has been triggered, the receiving side of an AM RLC entity shall:

- if *t-StatusProhibit* is not running:
  - at the first transmission opportunity indicated by lower layer, construct a STATUS PDU and deliver it to lower layer;
- else:
  - at the first transmission opportunity indicated by lower layer after *t-StatusProhibit* expires, construct a single STATUS PDU even if status reporting was triggered several times while *t-StatusProhibit* was running and deliver it to lower layer;

ONEPLUS - EX. 1013

When a STATUS PDU has been delivered to lower layer, the receiving side of an AM RLC entity shall:

- start *t-StatusProhibit*.

When constructing a STATUS PDU, the AM RLC entity shall:

- for the AMD PDUs with SN such that VR(R) <= SN < VR(MS) that has not been completely received yet, in increasing SN order of PDUs and increasing byte segment order within PDUs, starting with SN = VR(R) up to the point where the resulting STATUS PDU still fits to the total size of RLC PDU(s) indicated by lower layer:
  - for an AMD PDU for which no byte segments have been received yet::
    - include in the STATUS PDU a NACK_SN which is set to the SN of the AMD PDU;
  - for a continuous sequence of byte segments of a partly received AMD PDU that have not been received yet:
    - include in the STATUS PDU a set of NACK_SN, SOstart and SOend
- set the ACK_SN to the SN of the next not received RLC Data PDU which is not indicated as missing in the resulting STATUS PDU.

5.3 SDU discard procedures

When indicated from upper layer (i.e. PDCP) to discard a particular RLC SDU, the transmitting side of an AM RLC entity or the transmitting UM RLC entity shall discard the indicated RLC SDU if no segment of the RLC SDU has been mapped to a RLC data PDU yet.

5.4 Re-establishment procedure

RLC re-establishment is performed upon request by RRC, and the function is applicable for AM, UM and TM RLC entities.

When RRC indicates that an RLC entity should be re-established, the RLC entity shall:

- if it is a transmitting TM RLC entity:
  - discard all RLC SDUs;
- if it is a receiving UM RLC entity:
  - when possible, reassemble RLC SDUs from UMD PDUs with SN < VR(UH), remove RLC headers when doing so and deliver all reassembled RLC SDUs to upper layer in ascending order of the RLC SN, if not delivered before;
  - discard all remaining UMD PDUs;
- if it is a transmitting UM RLC entity:
  - discard all RLC SDUs;
- if it is an AM RLC entity:
  - when possible, reassemble RLC SDUs from any byte segments of AMD PDUs with SN < VR(MR) in the receiving side, remove RLC headers when doing so and deliver all reassembled RLC SDUs to upper layer in ascending order of the RLC SN, if not delivered before;
  - discard the remaining AMD PDUs and byte segments of AMD PDUs in the receiving side;
  - discard all RLC SDUs and AMD PDUs in the transmitting side;
  - discard all RLC control PDUs.
- stop and reset all timers;
- reset all state variables to their initial values.

ONEPLUS - EX. 1013

5.5 Handling of unknown, unforeseen and erroneous protocol data

5.5.1 Reception of PDU with reserved or invalid values

When an RLC entity receives an RLC PDU that contains reserved or invalid values, the RLC entity shall:

- discard the received PDU.

6 Protocol data units, formats and parameters

6.1 Protocol data units

RLC PDUs can be categorized into RLC data PDUs and RLC control PDUs. RLC data PDUs in sub clause 6.1.1 are used by TM, UM and AM RLC entities to transfer upper layer PDUs (i.e. RLC SDUs). RLC control PDUs in sub clause 6.1.2 are used by AM RLC entity to perform ARQ procedures.

6.1.1 RLC data PDU

a) TMD PDU

TMD PDU is used to transfer upper layer PDUs by a TM RLC entity.

b) UMD PDU

UMD PDU is used to transfer upper layer PDUs by an UM RLC entity.

c) AMD PDU

AMD PDU is used to transfer upper layer PDUs by an AM RLC entity. It is used when the AM RLC entity transmits (part of) the RLC SDU for the first time, or when the AM RLC entity retransmits an AMD PDU without having to perform re-segmentation.

d) AMD PDU segment

AMD PDU segment is used to transfer upper layer PDUs by an AM RLC entity. It is used when the AM RLC entity needs to retransmit a portion of an AMD PDU.

6.1.2 RLC control PDU

a) STATUS PDU

STATUS PDU is used by the receiving side of an AM RLC entity to inform the peer AM RLC entity about RLC data PDUs that are received successfully, and RLC data PDUs that are detected to be lost by the receiving side of an AM RLC entity.

6.2 Formats and parameters

The formats of RLC PDUs are described in sub clause 6.2.1 and their parameters are described in sub clause 6.2.2.

6.2.1 Formats

6.2.1.1 General

RLC PDU is a bit string. In the figures in sub clause 6.2.1.2 to 6.2.1.6, bit strings are represented by tables in which the first and most significant bit is the left most bit of the first line of the table, the last and least significant bit is the rightmost bit of the last line of the table, and more generally the bit string is to be read from left to right and then in the reading order of the lines.

RLC SDUs are bit strings that are byte aligned (i.e. multiple of 8 bits) in length. An RLC SDU is included into an RLC PDU from first bit onward.

ONEPLUS - EX. 1013

6.2.1.2 TMD PDU

TMD PDU consists only of a Data field and does not consist of any RLC headers.

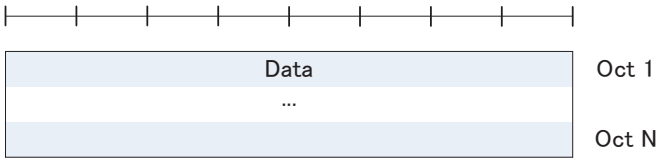

Figure 6.2.1.2-1: TMD PDU

6.2.1.3 UMD PDU

UMD PDU consists of a Data field and an UMD PDU header.

UMD PDU header consists of a fixed part (fields that are present for every UMD PDU) and an extension part (fields that are present for an UMD PDU when necessary). The fixed part of the UMD PDU header itself is byte aligned and consists of a FI, an E and a SN. The extension part of the UMD PDU header itself is byte aligned and consists of E(s) and LI(s).

An UM RLC entity is configured by RRC to use either a 5 bit SN or a 10 bit SN. When the 5 bit SN is configured, the length of the fixed part of the UMD PDU header is one byte. When the 10 bit SN is configured, the fixed part of the UMD PDU header is identical to the fixed part of the AMD PDU header, except for D/C, RF and P fields all being replaced with R1 fields. The extension part of the UMD PDU header is identical to the extension part of the AMD PDU header (regardless of the configured SN size).

An UMD PDU header consists of an extension part only when more than one Data field elements are present in the UMD PDU, in which case an E and a LI are present for every Data field element except the last. Furthermore, when an UMD PDU header consists of an odd number of LI(s), four padding bits follow after the last LI.

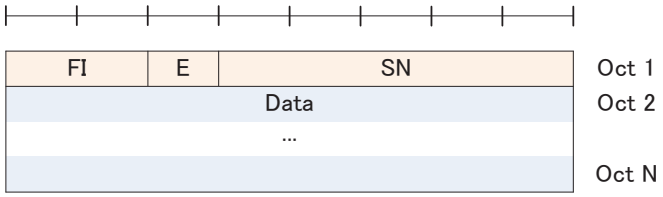

Figure 6.2.1.3-1: UMD PDU with 5 bit SN (No LI)

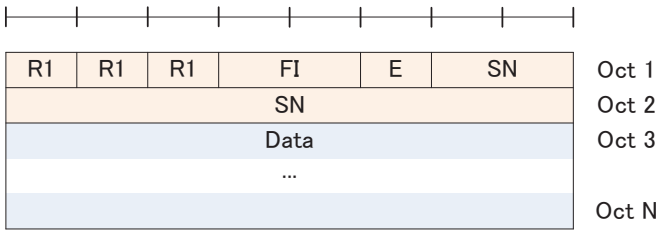

Figure 6.2.1.3-2: UMD PDU with 10 bit SN (No LI)

ONEPLUS - EX. 1013

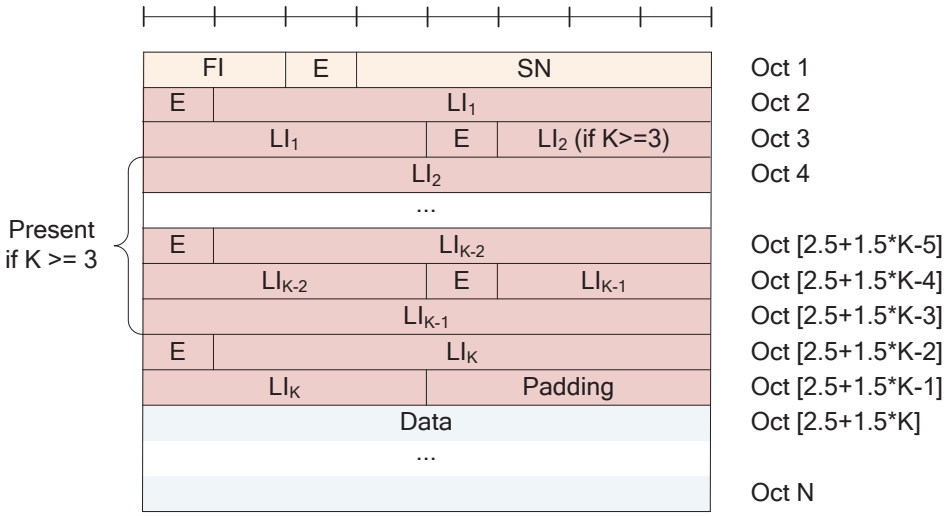

Figure 6.2.1.3-3: UMD PDU with 5 bit SN (Odd number of LIs, i.e. K = 1, 3, 5, …)

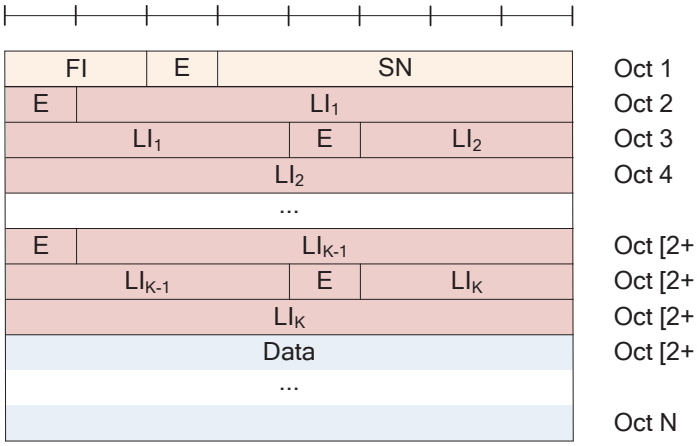

Figure 6.2.1.3-4: UMD PDU with 5 bit SN (Even number of LIs, i.e. K = 2, 4, 6, …)

ONEPLUS - EX. 1013

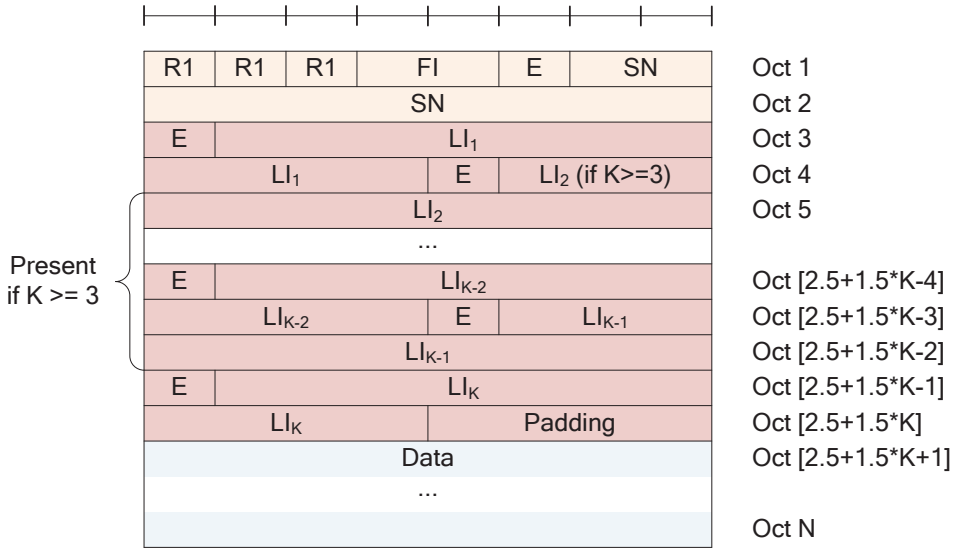

Figure 6.2.1.3-5: UMD PDU with 10 bit SN (Odd number of LIs, i.e. K = 1, 3, 5, …)

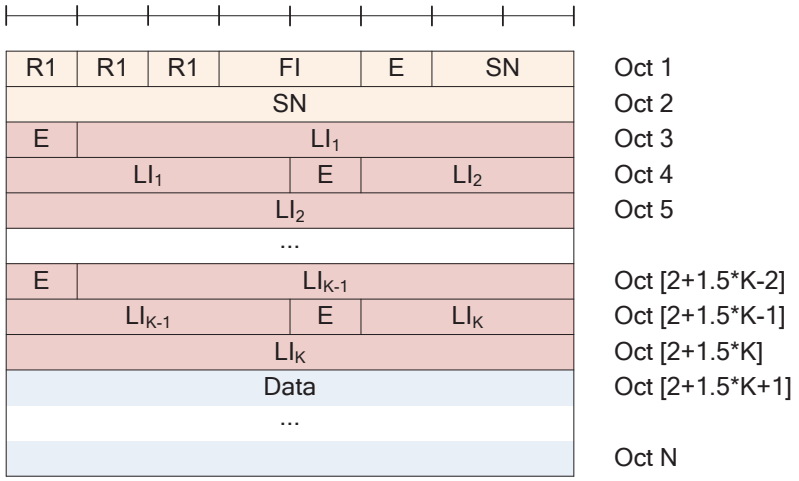

Figure 6.2.1.3-6: UMD PDU with 10 bit SN (Even number of LIs, i.e. K = 2, 4, 6, …)

6.2.1.4 AMD PDU

AMD PDU consists of a Data field and an AMD PDU header.

AMD PDU header consists of a fixed part (fields that are present for every AMD PDU) and an extension part (fields that are present for an AMD PDU when necessary). The fixed part of the AMD PDU header itself is byte aligned and consists of a D/C, a RF, a P, a FI, an E and a SN. The extension part of the AMD PDU header itself is byte aligned and consists of E(s) and LI(s).

An AMD PDU header consists of an extension part only when more than one Data field elements are present in the AMD PDU, in which case an E and a LI are present for every Data field element except the last. Furthermore, when an AMD PDU header consists of an odd number of LI(s), four padding bits follow after the last LI.

ONEPLUS - EX. 1013

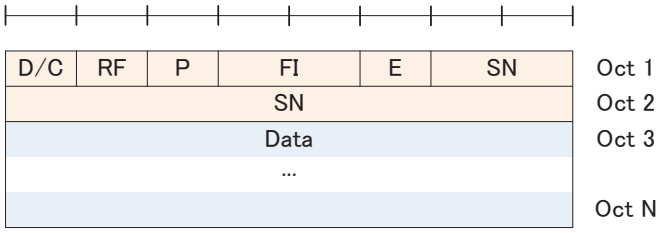

Figure 6.2.1.4-1: AMD PDU (No LI)

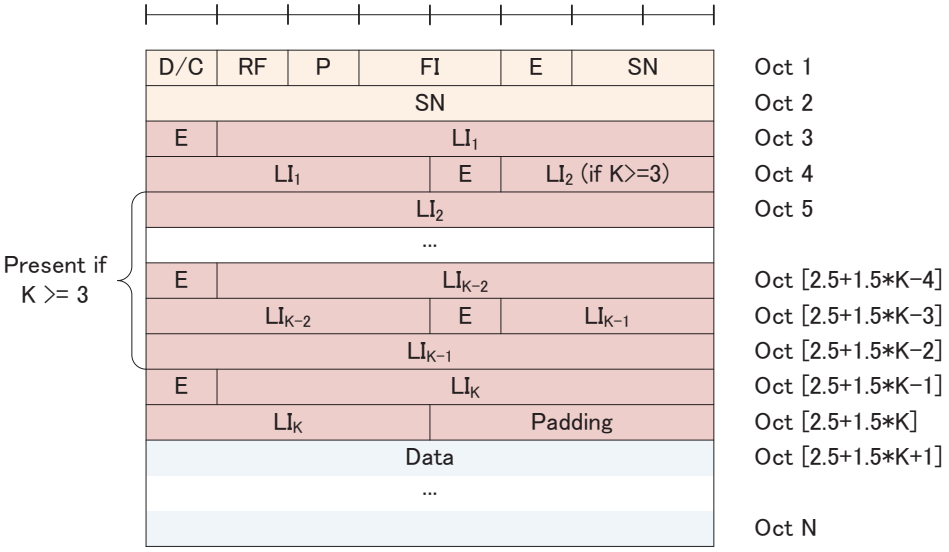

Figure 6.2.1.4-2: AMD PDU (Odd number of LIs, i.e. K = 1, 3, 5, …)

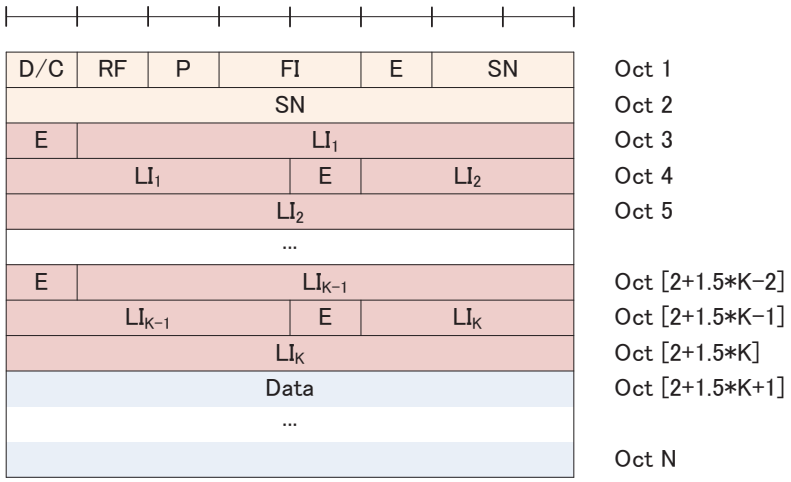

Figure 6.2.1.4-3: AMD PDU (Even number of LIs, i.e. K = 2, 4, 6, …)

6.2.1.5 AMD PDU segment

AMD PDU segment consists of a Data field and an AMD PDU segment header.

ONEPLUS - EX. 1013

AMD PDU segment header consists of a fixed part (fields that are present for every AMD PDU segment) and an extension part (fields that are present for an AMD PDU segment when necessary). The fixed part of the AMD PDU segment header itself is byte aligned and consists of a D/C, a RF, a P, a FI, an E, a SN, a LSF and a SO. The extension part of the AMD PDU segment header itself is byte aligned and consists of E(s) and LI(s).

An AMD PDU segment header consists of an extension part only when more than one Data field elements are present in the AMD PDU segment, in which case an E and a LI are present for every Data field element except the last. Furthermore, when an AMD PDU segment header consists of an odd number of LI(s), four padding bits follow after the last LI.

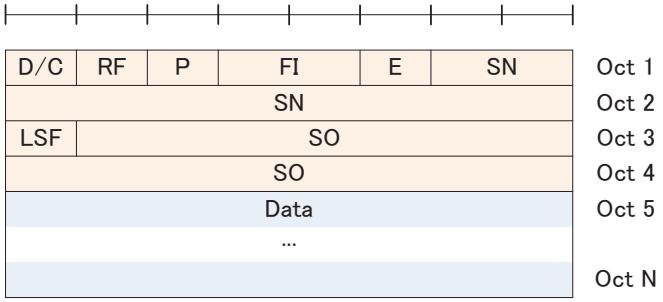

Figure 6.2.1.5-1: AMD PDU segment (No LI)

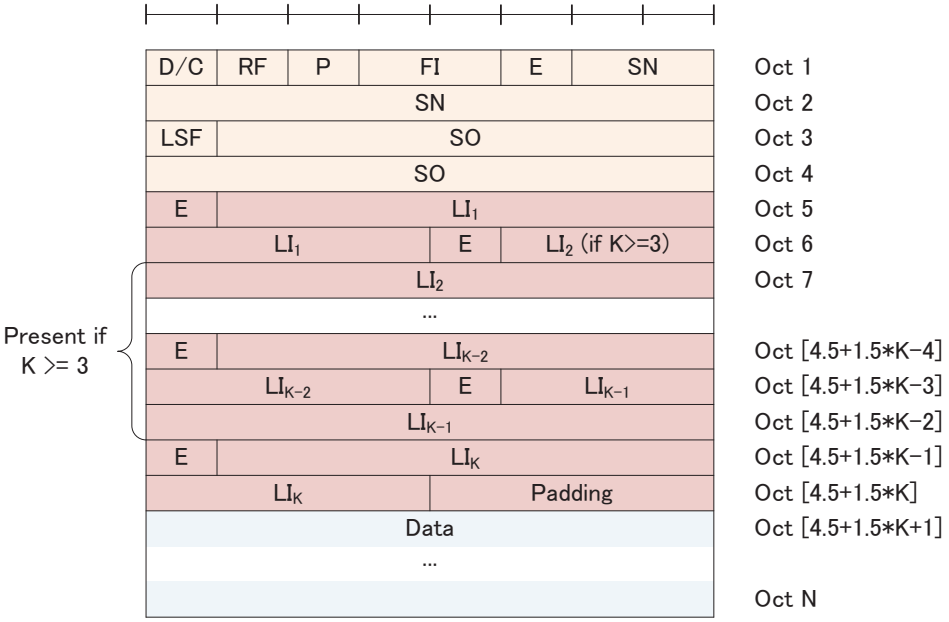

Figure 6.2.1.5-2: AMD PDU segment (Odd number of LIs, i.e. K = 1, 3, 5, …)

ONEPLUS - EX. 1013

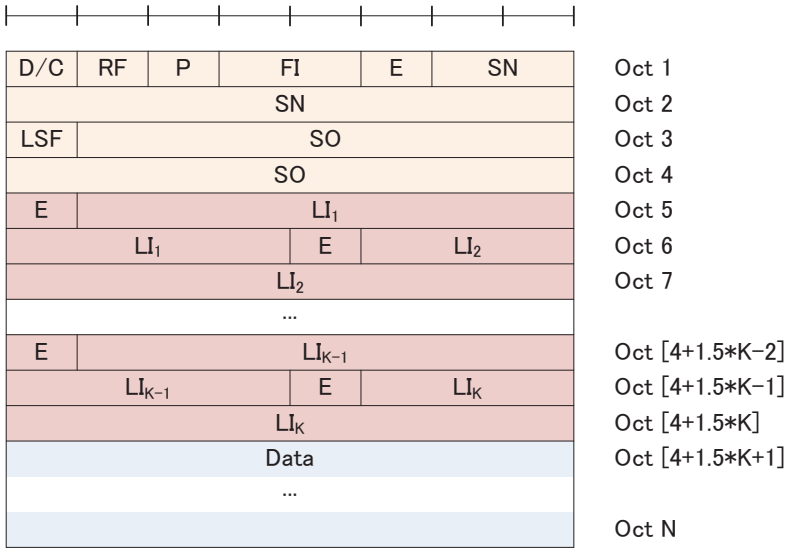

Figure 6.2.1.5-3: AMD PDU segment (Even number of LIs, i.e. K = 2, 4, 6, …)

6.2.1.6 STATUS PDU

STATUS PDU consists of a STATUS PDU payload and a RLC control PDU header.

RLC control PDU header consists of a D/C and a CPT field.

The STATUS PDU payload starts from the first bit following the RLC control PDU header, and it consists of one ACK_SN and one E1, zero or more sets of a NACK_SN, an E1 and an E2, and possibly a set of a SOstart and a SOend for each NACK_SN. When necessary one to seven padding bits are included in the end of the STATUS PDU to achieve octet alignment.

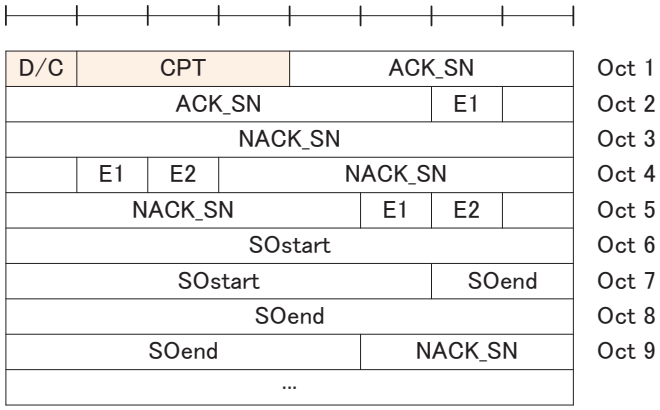

Figure 6.2.1.6-1: STATUS PDU

6.2.2 Parameters

6.2.2.1 General

In the definition of each field in sub clauses 6.2.2.2 to 6.2.2.19, the bits in the parameters are represented in which the first and most significant bit is the left most bit and the last and least significant bit is the rightmost bit. Unless mentioned otherwise, integers are encoded in standard binary encoding for unsigned integers.

ONEPLUS - EX. 1013

6.2.2.2 Data field

Data field elements are mapped to the Data field in the order which they arrive to the RLC entity at the transmitter.

For TMD PDU, UMD PDU and AMD PDU:

- The granularity of the Data field size is one byte;
- The maximum Data field size is the maximum TB size minus the sum of minimum MAC PDU header size and minimum RLC PDU header size.

For TMD PDU:

- Only one RLC SDU can be mapped to the Data field of one TMD PDU.

For UMD PDU, AMD PDU and AMD PDU segment:

- Either of the following can be mapped to the Data field of one UMD PDU, AMD PDU or AMD PDU segment:
  - Zero RLC SDU segments and one or more RLC SDUs;
  - One or two RLC SDU segments and zero or more RLC SDUs;
- RLC SDU segments are either mapped to the beginning or the end of the Data field;
- A RLC SDU or RLC SDU segment larger than 2047 octets can only be mapped to the end of the Data field;
- When there are two RLC SDU segments, they belong to different RLC SDUs.

6.2.2.3 Sequence Number (SN) field

Length: 10 bits for AMD PDU and AMD PDU segments. 5 bits or 10 bits (configurable) for UMD PDU.

The SN field indicates the sequence number of the corresponding UMD or AMD PDU. For an AMD PDU segment, the SN field indicates the sequence number of the original AMD PDU from which the AMD PDU segment was constructed from. The sequence number is incremented by one for every UMD or AMD PDU.

6.2.2.4 Extension bit (E) field

Length: 1 bit.

The E field indicates whether Data field follows or a set of E field and LI field follows. The interpretation of the E field is provided in Table 6.2.2.4-1 and Table 6.2.2.4-2.

Table 6.2.2.4-1: E field interpretation (for E field in the fixed part of the header)

Value	Description
0	Data field follows from the octet following the fixed part of the header
1	A set of E field and LI field follows from the octet following the fixed part of the header

Table 6.2.2.4-2: E field interpretation (for E field in the extension part of the header)

Value	Description
0	Data field follows from the octet following the LI field following this E field
1	A set of E field and LI field follows from the bit following the LI field following this E field

6.2.2.5 Length Indicator (LI) field

Length: 11 bits.

The LI field indicates the length in bytes of the corresponding Data field element present in the RLC data PDU delivered/received by an UM or an AM RLC entity. The first LI present in the RLC data PDU header corresponds to the first Data field element present in the Data field of the RLC data PDU, the second LI present in the RLC data PDU header corresponds to the second Data field element present in the Data field of the RLC data PDU, and so on. The value 0 is reserved.

ONEPLUS - EX. 1013

6.2.2.6 Framing Info (FI) field

Length: 2 bits.

The FI field indicates whether a RLC SDU is segmented at the beginning and/or at the end of the Data field. Specifically, the FI field indicates whether the first byte of the Data field corresponds to the first byte of a RLC SDU, and whether the last byte of the Data field corresponds to the last byte of a RLC SDU. The interpretation of the FI field is provided in Table 6.2.2.6-1.

Table 6.2.2.6-1: FI field interpretation

Value	Description
00	First byte of the Data field corresponds to the first byte of a RLC SDU. Last byte of the Data field corresponds to the last byte of a RLC SDU.
01	First byte of the Data field corresponds to the first byte of a RLC SDU. Last byte of the Data field does not correspond to the last byte of a RLC SDU.
10	First byte of the Data field does not correspond to the first byte of a RLC SDU. Last byte of the Data field corresponds to the last byte of a RLC SDU.
11	First byte of the Data field does not correspond to the first byte of a RLC SDU. Last byte of the Data field does not correspond to the last byte of a RLC SDU.

6.2.2.7 Segment Offset (SO) field

Length: 15 bits.

The SO field indicates the position of the AMD PDU segment in bytes within the original AMD PDU. Specifically, the SO field indicates the position within the Data field of the original AMD PDU to which the first byte of the Data field of the AMD PDU segment corresponds to. The first byte in the Data field of the original AMD PDU is referred by the SO field value "000000000000000", i.e., numbering starts at zero.

6.2.2.8 Last Segment Flag (LSF) field

Length: 1 bit.

The LSF field indicates whether or not the last byte of the AMD PDU segment corresponds to the last byte of an AMD PDU. The interpretation of the LSF field is provided in Table 6.2.2.8-1.

Table 6.2.2.8-1: LSF field interpretation

Value	Description
0	Last byte of the AMD PDU segment does not correspond to the last byte of an AMD PDU.
1	Last byte of the AMD PDU segment corresponds to the last byte of an AMD PDU.

6.2.2.9 Data/Control (D/C) field

Length: 1 bit.

The D/C field indicates whether the RLC PDU is a RLC data PDU or RLC control PDU. The interpretation of the D/C field is provided in Table 6.2.2.9-1.

Table 6.2.2.9-1: D/C field interpretation

Value	Description
0	Control PDU
1	Data PDU

6.2.2.10 Re-segmentation Flag (RF) field

Length: 1 bit.

The RF field indicates whether the RLC PDU is an AMD PDU or AMD PDU segment. The interpretation of the RF field is provided in Table 6.2.2.10-1.

ONEPLUS - EX. 1013

Table 6.2.2.10-1: RF field interpretation

Value	Description
0	AMD PDU
1	AMD PDU segment

6.2.2.11 Polling bit (P) field

Length: 1 bit.

The P field indicates whether or not the transmitting side of an AM RLC entity requests a STATUS report from its peer AM RLC entity. The interpretation of the P field is provided in Table 6.2.2.11-1.

Table 6.2.2.11-1: P field interpretation

Value	Description
0	Status report not requested
1	Status report is requested

6.2.2.12 Reserved 1 (R1) field

Length: 1 bit.

The R1 field is a reserved field for this release of the protocol. The transmitting entity shall set the R1 field to "0". The receiving entity shall ignore this field.

6.2.2.13 Control PDU Type (CPT) field

Length: 3 bits.

The CPT field indicates the type of the RLC control PDU. The interpretation of the CPT field is provided in Table 6.2.2.13-1.

Table 6.2.2.13-1: CPT field interpretation

Value	Description
000	STATUS PDU
001-111	Reserved (PDUs with this coding will be discarded by the receiving entity for this release of the protocol)

6.2.2.14 Acknowledgement SN (ACK_SN) field

Length: 10 bits.

The ACK_SN field indicates the SN of the next not received RLC Data PDU which is not reported as missing in the STATUS PDU. When the transmitting side of an AM RLC entity receives a STATUS PDU, it interprets that all AMD PDUs up to but not including the AMD PDU with SN = ACK_SN have been received by its peer AM RLC entity, excluding those AMD PDUs indicated in the STATUS PDU with NACK_SN and portions of AMD PDUs indicated in the STATUS PDU with NACK_SN, SOstart and SOend.

6.2.2.15 Extension bit 1 (E1) field

Length: 1 bit.

The E1 field indicates whether or not a set of NACK_SN, E1 and E2 follows. The interpretation of the E1 field is provided in Table 6.2.2.15-1.

Table 6.2.2.15-1: E1 field interpretation

Value	Description
0	A set of NACK_SN, E1 and E2 does not follow.
1	A set of NACK_SN, E1 and E2 follows.

ONEPLUS - EX. 1013

6.2.2.16 Negative Acknowledgement SN (NACK_SN) field

Length: 10 bits.

The NACK_SN field indicates the SN of the AMD PDU (or portions of it) that has been detected as lost at the receiving side of the AM RLC entity.

6.2.2.17 Extension bit 2 (E2) field

Length: 1 bit.

The E2 field indicates whether or not a set of SOstart and SOend follows. The interpretation of the E2 field is provided in Table 6.2.2.17-1.

Table 6.2.2.17-1: E2 field interpretation

Value	Description
0	A set of SOstart and SOend does not follow for this NACK_SN.
1	A set of SOstart and SOend follows for this NACK_SN.

6.2.2.18 SO start (SOstart) field

Length: 15 bits.

The SOstart field (together with the SOend field) indicates the portion of the AMD PDU with SN = NACK_SN (the NACK_SN for which the SOstart is related to) that has been detected as lost at the receiving side of the AM RLC entity. Specifically, the SOstart field indicates the position of the first byte of the portion of the AMD PDU in bytes within the Data field of the AMD PDU. The first byte in the Data field of the original AMD PDU is referred by the SOstart field value "000000000000000", i.e., numbering starts at zero.

6.2.2.19 SO end (SOend) field

Length: 15 bits.

The SOend field (together with the SOstart field) indicates the portion of the AMD PDU with SN = NACK_SN (the NACK_SN for which the SOend is related to) that has been detected as lost at the receiving side of the AM RLC entity. Specifically, the SOend field indicates the position of the last byte of the portion of the AMD PDU in bytes within the Data field of the AMD PDU. The first byte in the Data field of the original AMD PDU is referred by the SOend field value "000000000000000", i.e., numbering starts at zero. The special SOend value "111111111111111" is used to indicate that the missing portion of the AMD PDU includes all bytes to the last byte of the AMD PDU.

7 Variables, constants and timers

7.1 State variables

This sub clause describes the state variables used in AM and UM entities in order to specify the RLC protocol. The state variables defined in this subclause are normative.

All state variables and all counters are non-negative integers.

All state variables related to AM data transfer can take values from 0 to 1023. All arithmetic operations contained in the present document on state variables related to AM data transfer are affected by the AM modulus (i.e. final value = [value from arithmetic operation] modulo 1024).

All state variables related to UM data transfer can take values from 0 to $[2^{[sn\text{-}FieldLength]} - 1]$. All arithmetic operations contained in the present document on state variables related to UM data transfer are affected by the UM modulus (i.e. final value = [value from arithmetic operation] modulo $2^{[sn\text{-}FieldLength]}$).

AMD PDUs and UMD PDUs are numbered integer sequence numbers (SN) cycling through the field: 0 to 1023 for AMD PDU and 0 to $[2^{[sn\text{-}FieldLength]} - 1]$ for UMD PDU.

When performing arithmetic comparisons of state variables or SN values, a modulus base shall be used.

ONEPLUS - EX. 1013

VT(A) and VR(R) shall be assumed as the modulus base at the transmitting side and receiving side of an AM RLC entity, respectively. This modulus base is subtracted from all the values involved, and then an absolute comparison is performed (e.g. VR(R) <= SN < VR(MR) is evaluated as [VR(R) – VR(R)] modulo 1024 <= [SN – VR(R)] modulo 1024 < [VR(MR) – VR(R)] modulo 1024).

VR(UH) – UM_Window_Size shall be assumed as the modulus base at the receiving side of an UM RLC entity. This modulus base is subtracted from all the values involved, and then an absolute comparison is performed (e.g. (VR(UH) – UM_Window_Size) <= SN < VR(UH) is evaluated as [(VR(UH) – UM_Window_Size) – (VR(UH) – UM_Window_Size)] modulo $2^{[sn\text{-}FieldLength]}$ <= [SN – (VR(UH) – UM_Window_Size)] modulo $2^{[sn\text{-}FieldLength]}$ < [VR(UH) – (VR(UH) – UM_Window_Size)] modulo $2^{[sn\text{-}FieldLength]}$).

The transmitting side of each AM RLC entity shall maintain the following state variables:

a) VT(A) – Acknowledgement state variable

This state variable holds the value of the SN of the next AMD PDU for which a positive acknowledgment is to be received in-sequence, and it serves as the lower edge of the transmitting window. It is initially set to 0, and is updated whenever the AM RLC entity receives a positive acknowledgment for an AMD PDU with SN = VT(A).

b) VT(MS) – Maximum send state variable

This state variable equals VT(A) + AM_Window_Size, and it serves as the higher edge of the transmitting window.

c) VT(S) – Send state variable

This state variable holds the value of the SN to be assigned for the next newly generated AMD PDU. It is initially set to 0, and is updated whenever the AM RLC entity delivers an AMD PDU with SN = VT(S).

d) POLL_SN – Poll send state variable

This state variable holds the value of VT(S)-1 upon the most recent transmission of a RLC data PDU with the poll bit set to “1”. It is initially set to 0.

The transmitting side of each AM RLC entity shall maintain the following counters:

a) PDU_WITHOUT_POLL – Counter

This counter is initially set to 0. It counts the number of AMD PDUs sent since the most recent poll bit was transmitted.

b) BYTE_WITHOUT_POLL – Counter

This counter is initially set to 0. It counts the number of data bytes sent since the most recent poll bit was transmitted.

c) RETX_COUNT – Counter

This counter counts the number of retransmissions of an AMD PDU (see subclause 5.2.1). There is one RETX_COUNT counter per PDU that needs to be retransmitted.

The receiving side of each AM RLC entity shall maintain the following state variables:

a) VR(R) – Receive state variable

This state variable holds the value of the SN following the last in-sequence completely received AMD PDU, and it serves as the lower edge of the receiving window. It is initially set to 0, and is updated whenever the AM RLC entity receives an AMD PDU with SN = VR(R).

b) VR(MR) – Maximum acceptable receive state variable

This state variable equals VR(R) + AM_Window_Size, and it holds the value of the SN of the first AMD PDU that is beyond the receiving window and serves as the higher edge of the receiving window.

ONEPLUS - EX. 1013

c) VR(X) – *t-Reordering* state variable

This state variable holds the value of the SN following the SN of the RLC data PDU which triggered *t-Reordering*..

d) VR(MS) – Maximum STATUS transmit state variable

This state variable holds the highest possible value of the SN which can be indicated by "ACK_SN" when a STATUS PDU needs to be constructed. It is initially set to 0.

e) VR(H) – Highest received state variable

This state variable holds the value of the SN following the SN of the RLC data PDU with the highest SN among received RLC data PDUs. It is initially set to 0.

Each transmitting UM RLC entity shall maintain the following state variables:

a) VT(US)

This state variable holds the value of the SN to be assigned for the next newly generated UMD PDU. It is initially set to 0, and is updated whenever the UM RLC entity delivers an UMD PDU with SN = VT(US).

Each receiving UM RLC entity shall maintain the following state variables:

a) VR(UR) – UM receive state variable

This state variable holds the value of the SN of the earliest UMD PDU that is still considered for reordering. It is initially set to 0.

b) VR(UX) – UM *t-Reordering* state variable

This state variable holds the value of the SN following the SN of the UMD PDU which triggered *t-Reordering*.

c) VR(UH) – UM highest received state variable

This state variable holds the value of the SN following the SN of the UMD PDU with the highest SN among received UMD PDUs, and it serves as the higher edge of the reordering window. It is initially set to 0.

7.2 Constants

a) AM_Window_Size

This constant is used by both the transmitting side and the receiving side of each AM RLC entity to calculate VT(MS) from VT(A), and VR(MR) from VR(R). AM_Window_Size = 512.

b) UM_Window_Size

This constant is used by the receiving UM RLC entity to define SNs of those UMD PDUs that can be received without causing an advancement of the receiving window. UM_Window_Size = 16 when a 5 bit SN is configured, UM_Window_Size = 512 when a 10 bit SN is configured and UM_Window_Size = 0 when the receiving UM RLC entity is configured for MCCH or MTCH.

7.3 Timers

The following timers are configured by RRC [5]:

a) *t-PollRetransmit*

This timer is used by the transmitting side of an AM RLC entity in order to retransmit a poll (see sub clause 5.2.2).

b) *t-Reordering*

This timer is used by the receiving side of an AM RLC entity and receiving UM RLC entity in order to detect loss of RLC PDUs at lower layer (see sub clauses 5.1.2.2 and 5.1.3.2). If *t-Reordering* is running, *t-Reordering* shall not be started additionally, i.e. only one *t-Reordering* per RLC entity is running at a given time.

ONEPLUS - EX. 1013

c) *t-StatusProhibit*

This timer is used by the receiving side of an AM RLC entity in order to prohibit transmission of a STATUS PDU (see sub clause 5.2.3).

7.4 Configurable parameters

The following parameters are configured by RRC [5]:

a) *maxRetxThreshold*

This parameter is used by the transmitting side of each AM RLC entity to limit the number of retransmissions of an AMD PDU (see subclause 5.2.1).

b) *pollPDU*

This parameter is used by the transmitting side of each AM RLC entity to trigger a poll for every *pollPDU* PDUs (see subclause 5.2.2).

c) *pollByte*

This parameter is used by the transmitting side of each AM RLC entity to trigger a poll for every *pollByte* bytes (see subclause 5.2.2).

d) *sn-FieldLength*

This parameter gives the UM SN field size in bits (see subclause 7.1).

ONEPLUS - EX. 1013

Annex A (informative): Change history

Change history							
Date	**TSG #**	**TSG Doc.**	**CR**	**Rev**	**Subject/Comment**	**Old**	**New**
2007-06	RAN2#58 bis	R2-072715			First version; Endorsed as v0.1.0.	x.y.z	0.1.0
2007-06	RAN2#58 bis	R2-072910			Added definition for Data field element and RLC SDU segment; Removed Editor's note on non-byte-aligned RLC SDUs; Added description for Data field for AMD PDU; Added Align Info for AMD PDU header element; Added description on extension part of AMD PDU header at concatenation; Added figure for AMD PDU; Added Segment Offset and Length Field for AMD PDU segment header element; Added Editor's note for STATUS PDU; Added general text on parameters; Added description for Extension bit, Length Indicator, Align Info, Segment Offset and Last Segment Flag; Removed Editor's note on Length Indicator.	0.1.0	0.1.1
2007-06	RAN2#58 bis	R2-072995			Moved description of Data field for AMD PDU and TMD PDU to the section dedicated to Data field; Changed terminology for Align Info to Segmentation Info; Added Segmentation Info for UMD PDU header element; Removed figure for AMD PDU; Corrected error for AMD PDU segment header element (replaced Length Field by Last Segment Flag); Added place holders to specify the number of bits for the individual RLC header elements; Modified description for Extension bit, Segmentation Info and Last Segment Flag using tables.	0.1.1	0.1.2
2007-06	RAN2#58 bis	R2-072996			Bracketed terminology for Segmentation Info; Corrected section numbering; Clarified description of Extension bit and Segment Offset.	0.1.2	0.1.3
2007-08	RAN2#59	R2-073554			Added receive operation descriptions for the case AM RLC entity receives AMD PDU segments; Modified general texts regarding retransmissions; Added a general description text for Segmentation Info; Added an Editor's note for Segment Offset.	0.1.3	0.1.4
2007-08	RAN2#59	R2-073712			v0.1.4 was endorsed by RAN WG2 as v0.2.0.	0.1.4	0.2.0
2007-08	RAN2#59	R2-073844			Added some missing abbreviations in section 3.2; Added description and a figure regarding RLC entity configuration in section 4.2.1, and removed Editor's note on this aspect; Added new sub clauses under sections 4.2.1.1-4.2.1.3 (purely editorial modification) Added description on SN, i.e. RLC PDU based SN, and removed Editor's note on the possibility of having the same header structure for AMD PDU and AMD PDU segment due to PDCP SN reuse; Added description of the AM receive window operation; Added description that fixed header part should be byte aligned and extension header part should be byte aligned; Modified description on extension header part (LI and E are not required for the last Data field element) and removed Editor's note on this aspect; Added further description on Data field; LI field size is set to 11bits; Corrected editorial errors (reference number to tables) Added 3 state variables: VT(S), VR(R), VR(MR).	0.2.0	0.2.1
2007-08	RAN2#59	R2-073868			Corrected editorial errors; Added 1 constant: Rx_Wndow_Size; Added description of modulus operation on VT(S), VR(R), VT(MR).	0.2.1	0.2.2
2007-08	RAN2#59	R2-073881			Removed reference to RLC UM for VT(R) and VT(MR).	0.2.2	0.2.3
2007-09	RAN#37	RP-070689			v0.2.3 was endorsed by RAN WG2 as v1.0.0 and presented to RAN plenary for information.	0.2.3	1.0.0
2007-11	RAN2#60	R2-074583			Added description of the AM transmit window operation; Added Editor's note that PDU loss detection should be after HARQ reordering; Added description on AM retransmission and resegmentation; Added description of the polling trigger "transmission of last data in	1.0.0	1.0.1

ONEPLUS - EX. 1013

Change history							
Date	TSG #	TSG Doc.	CR	Rev	Subject/Comment	Old	New
					the buffer"; Added polling trigger "poll retransmit timer" and its description; Added Editor's note that either PDU count based or window based polling trigger should be supported; Added description of the status reporting trigger "polling from its peer AM RLC entity"; Removed old Editor's note in the polling sub clause; Added status reporting trigger "detection of reception failure of an RLC data PDU" and its description; Added status prohibit function and its description; Added Editor's note that at least a 1byte fixed header for the UMD PDU should be supported; Complete AMD PDU and AMD PDU segment headers captured and 3 new figures inserted for each of them; Added Editor's note that STATUS PDU will only have 1 format; Modified description of the Data field so that it also applies to AMD PDU segment; Added description of the SN field in relation to the AMD PDU segment; Corrected description of the E field; Defined SO field length to be 15bits; Completed the description of the SO field and removed the related Editor's note; Added description of the D/C field, RF field and P field; Added 2 state variables: VT(A) and VT(MS); Added to the description of VT(S) and VR(MR); Added 1 constant: Tx_Window_Size; Added 2 timers: T_poll_prohibit and T_status_prohibit.r		
2007-11	RAN2#60	R2-075061			Cleaned up terminology related to PDUs; Editorial corrections (aligned wording, corrected Figure numbering, clarifications, etc.) Added Editor's note that exception cases when a negatively acknowledged RLC data PDU should not be transmitted will be captured when identified; Added description that the use of status prohibit function and particular polling triggers are configurable; Modified receiver operation for RLC-AM regarding AMD PDU segments (aligned with AMD PDU); Added description that DL CCCH is handled by RLC-UM and removed corresponding Editor's note; Added Editor's note that wording "considered" regarding retransmission of AMD PDU / AMD PDU segment in sub clause 5.2.1 should be improved; Added to the description of SN field that it is 10bits for AMD PDU and AMD PDU segment; Removed Editor's note regarding the need for status prohibit function; Added Editor's note that the need for [SI] field for UMD PDU can be challenged.	1.0.1	1.0.2
2007-11	RAN2#60	R2-075154			Added an Editor's note regarding Local NACK; Modified wording in sub clause 5.2.1 on the object of retransmission (RLC data PDU changed to AMD PDU / portion of AMD PDU); Clarified that STATUS PDU is triggered after the PDU containing th poll bit is "HARQ reordered", rather than just "reordered"; Removed incorrect inclusion of a T_status_prohibit and added an Editor's which just says status prohibit function is supported.	1.0.2	1.0.3
2007-11	RAN2#60	R2-075198			Description regarding the modulus operation involving state variables was changed in order to align with TS 25.322; Figures on PDUs were slightly modified (editorial).	1.0.3	1.0.4
2007-11	RAN2#60	R2-075430			Text on receiver operation in sub clause 5.1.3 was revised to align the description regarding modulus operation with TS 25.322; Further cleaned up terminology related to PDUs; V1.0.4 was endorsed by RAN WG2 as v1.1.0 with the above revisions.	1.0.4	1.1.0
2007-11	RAN2#60	R2-075500			Added RLC architecture model figures (Figures 4.2.1.1.1-1, 4.2.1.2.1-1, 4.2.1.3.1-1); Added Editor's note that the SDU discard functionality may not be specified in RLC; Added description of receive operations for RLC-UM in sub clause 5.1.2; Added to the description of receive operations for RLC-AM in sub clause 5.1.3 including: STATUS transmitting window definition, procedures for the case RLC data PDU within receiving window is received, actions at T_reordering expiry and RLC SDU reassembly;	1.1.0	1.1.1

ONEPLUS - EX. 1013

Change history							
Date	TSG #	TSG Doc.	CR	Rev	Subject/Comment	Old	New
					T_and in sub clause 5.1.3; Removed Editor's note which said that PDU loss detection should be after HARQ reordering; Clarified the description of the polling trigger "transmission of last data in the buffer"; Added description of STATUS PDU construction in sub clause 5.2.3 and removed an Editor's note in this sub clause; Modified trigger for RLC SDU discard to "indication from PDCP"; Removed Editor's note on the type of PDUs to be specified; Removed Editor's notes regarding STATUS PDU piggybacking; Complete UMD PDU headers captured and 2 new figures inserted for them, and removed Editor's note on UMD PDU; Defined one STATUS PDU format with a new figure and an Editor's note, and removed old Editor's note in sub clause 6.2.1.6; Added description of the R1 field, CPT field, ACK_SN field, E1 field, NACK_SN field, E2 field, SOstart field and SOend field. Added 7 state variables: VR(R-SO), VR(X), VR(X-SO), VR(MS), VR(UR), VR(UMR) and VR(UX); Constants Rx_Window_Size and Tx Window size were converged into one constant "Window_Size" of which the value is defined to half the SN space, and constants "AM_Window_Size" and "UM_Window_Size" were newly defined; Added 1 timer: T_reordering.		
2007-11	RAN2#60	R2-075501			Added definition for "byte segment"; Removed Editor's note which said that the SDU discard functionality may not be specified in RLC; Added missing description for RLC-UM receive operation (the case when UMD PDU with SN that falls within the reordering window but not equal to VR(R) is received); Added missing description for RLC-AM receive operation (the case when only part of the received RLC data PDU is received in duplication); Added text related to updating state variable VR(MS); Added Editor's note that it has to be decided whether T_reordering can be triggered by a missing RLC data PDU for which status reporting has already been triggered once; Editorial clarification / corrections were made.	1.1.1	1.1.2
2007-11	RAN2#60	R2-075502			Added text related to updating state variable VR(MS); Editorial clarification / corrections were made.	1.1.2	1.1.3
2007-11	RAN2#60	R2-075503			Modified description of VR(MS) update procedure; Modified description of VR(X) / VR(X-SO) update procedure; Editorial corrections were made.	1.1.3	1.1.4
2007-11	RAN2#60	R2-075504			Added missing description with regards to RLC-AM receive operation; Added Editor's note on the delivery of RLC control PDUs; Editorial corrections were made.	1.1.4	1.1.5
2007-11	RAN2#60	R2-074589			v1.1.5 was endorsed by RAN WG2 as v1.2.0.	1.1.5	1.2.0

ONEPLUS - EX. 1013

Change history							
Date	**TSG #**	**TSG Doc.**	**CR**	**Rev**	**Subject/Comment**	**Old**	**New**
2007-11	RP-38	RP-070918			v1.2.0 was stepped to v2.0.0 and presented to RAN plenary for approval.	1.2.0	2.0.0
2007-12	RP-38	-			Approved at TSG RAN-38 and placed under change control	2.0.0	8.0.0
2008-03	RP-39	RP-080196	0001	-	CR0001 for TS 36.322 E-UTRA RLC: Added reference to TS 36.321; Clarified definition of “byte segment”; Renamed “Segmentation Info” to “Framing Info”; Aligned texts to refer to “upper layer” and “lower layer” instead of RRC/PDCP and MAC; Specified that BCCH and DL CCCH is handled by RLC-TM; Added support for duplicate detection by receiving RLC UM entity;; Clarified that RLC SDUs should be delivered to upper layers in sequence; Modified description so that MAC indicates “total size of RLC PDUs” together with notification of transmission opportunity instead of “TB size”; Specified that RLC SDU discard is applied for RLC-AM and RLC UM, and introduced the detailed RLC SDU discard procedure; Renamed “RLC reset” to “RLC re-establishment”, and introduced the detailed RLC re-establishment procedure; Removed Editor’s note on RLC flow control (flow control will not be supported by RLC); Restructured the texts on RLC AM and RLC UM receive operations, and added/modified the detailed descriptions; Added description on prioritization of data to transmit (control > data; retransmission > new data); Removed the term STATUS transmitting window; Clarified that retransmission of negatively acknowledged data by STATUS PDU is mandatory and that retransmission of negatively acknowledged data by HARQ delivery failure is optional; Removed Editor’s note on retransmission prohibit (there will be no conditions where negatively acknowledged data shall not be retransmitted); Clarified description on polling trigger “Transmission of last data in buffer”; Added new polling triggers “Every Poll_PDU PDUs” and “Every Poll_Byte Bytes”, introduced their descriptions, and added an Editor’s note that their configurability is FFS; Added description on status reporting trigger “detection of reception failure of an RLC data PDU”; Introduced description of the status prohibit function; Removed Editor’s note on the possibility to define more RLC control PDUs (no more RLC control PDUs will be defined); Clarified the “most significant bit” and “least significant bit” in an RLC PDU; Removed reference to bit numbers in RLC PDU; Modified the order of fields in the 1byte UMD PDU header; Removed Editor’s note on the order of fields in the AMD PDU / AMD PDU segment header (they are now confirmed); Modified definition of ACK_SN; Defined the special value of SOend; Added description on the UM modulus operation; Removed state variables VR(R-SO), VR(X-SO) and VR(UMR); Modified description/definition of state variables VR(MR), VR(X), VR(MS), VR(UR) and VR(UX); Introduced new state variables VR(H) and VR(UH) and their descriptions; Introduced new constants Poll_PDU and Poll_Byte and their descriptions; Clarified that only one T_reordering will be running at one time for an RLC entity; Introduced new timer T_status_prohibit and its description; Editorial corrections were made.	8.0.0	8.1.0
2008-05	RP-40	RP-080411	0002	1	Clarification on STATUS PDU size for BSR	8.1.0	8.2.0
	RP-40	RP-080411	0003	-	Removal of Editor’s Note on updating of VR(MS) upon expiry of T_reordering	8.1.0	8.2.0
	RP-40	RP-080411	0004	-	Removal of STATUS receiving window	8.1.0	8.2.0
	RP-40	RP-080411	0005	-	Duplicate detection in UM RLC	8.1.0	8.2.0
	RP-40	RP-080411	0006	-	Correction to Polling Procedure	8.1.0	8.2.0
	RP-40	RP-080411	0007	-	Miscellaneous corrections to TS 36.322	8.1.0	8.2.0
	RP-40	RP-080411	0008	-	Small corrections to RLC	8.1.0	8.2.0
	RP-40	RP-080411	0012	-	CR to 36.322 on correction to RLC PDU reassembly	8.1.0	8.2.0
	RP-40	RP-080411	0015	1	36.322 CR on 'RLC retransmission count and addition of Configurable Parameters'	8.1.0	8.2.0

ONEPLUS - EX. 1013

Change history							
Date	TSG #	TSG Doc.	CR	Rev	Subject/Comment	Old	New
	RP-40	RP-080411	0017	-	Service alignments with TS 36.323 (PDCP)	8.1.0	8.2.0
	RP-40	RP-080411	0018	-	CR on the procedure to construct the STATUS PDU	8.1.0	8.2.0
2008-09	RP-41	RP-080691	0019	1	Clarification of polling	8.2.0	8.3.0
	RP-41	RP-080691	0020	-	Corrections to formatting	8.2.0	8.3.0
	RP-41	RP-080691	0021	2	The value of ACK_SN for partial STATUS PDU	8.2.0	8.3.0
	RP-41	RP-080691	0022	1	Error cases for RLC	8.2.0	8.3.0
	RP-41	RP-080691	0024	-	RLC entity re-establishment	8.2.0	8.3.0
	RP-41	RP-080691	0025	-	Miscellaneous corrections to RLC specification	8.2.0	8.3.0
	RP-41	RP-080691	0029	-	Clarification of the reordering timer	8.2.0	8.3.0
	RP-41	RP-080691	0032	-	Clarification of Triggering Conditions for Status Reports	8.2.0	8.3.0
	RP-41	RP-080691	0033	-	RLC UMD PDU formats with LI	8.2.0	8.3.0
	RP-41	RP-080691	0036	-	Correction on UM Receive Operation	8.2.0	8.3.0
	RP-41	RP-080691	0039	-	Correction for TM RLC entity: 6.1.2.3	8.2.0	8.3.0
	RP-41	RP-080691	0040	-	Removal of MBMS channels: 6.1.2.3	8.2.0	8.3.0
2008-12	RP-42	RP-081019	0043	-	Proposed CR for aligning the construction of partial Status PDUs with intended operation	8.3.0	8.4.0
	RP-42	RP-081019	0046	-	Error Handling in RLC	8.3.0	8.4.0
	RP-42	RP-081019	0047	-	Miscellaneous corrections to 36.322	8.3.0	8.4.0
	RP-42	RP-081019	0048	-	Correction to Segment Offset fields	8.3.0	8.4.0
	RP-42	RP-081019	0049	-	Correction to the description of the delivery of RLC SDU	8.3.0	8.4.0
	RP-42	RP-081019	0050	-	Minor issues on RLC	8.3.0	8.4.0
	RP-42	RP-081019	0054	-	The setting of VR(X)	8.3.0	8.4.0
	RP-42	RP-081019	0055	-	Adding RLC TM operation	8.3.0	8.4.0
	RP-42	RP-081019	0056	-	Removing a redundant text on VT(A) setting	8.3.0	8.4.0
	RP-42	RP-081019	0057	-	Counting RLC Retransmissions	8.3.0	8.4.0
2009-03	RP-43	RP-090129	0058	-	CR to 36.322 on RRC Parameters	8.4.0	8.5.0
	RP-43	RP-090129	0059	-	Local NACKing in UE	8.4.0	8.5.0
	RP-43	RP-090129	0060	-	Supporting RLC SDU larger than 2047 octets	8.4.0	8.5.0
	RP-43	RP-090129	0061	-	CR on the in sequence delivery function for UM	8.4.0	8.5.0
	RP-43	RP-090129	0062	-	Correction to Delivery of PDU	8.4.0	8.5.0
	RP-43	RP-090129	0063	1	Issues with SO, SOstart, and SOend fields	8.4.0	8.5.0
	RP-43	RP-090129	0064	-	Miscellaneous corrections to RLC specification	8.4.0	8.5.0
	RP-43	RP-090129	0065	-	Correction to status reporting triggering condition	8.4.0	8.5.0
	RP-43	RP-090129	0066	-	Alignment of one condition on setting the poll bit	8.4.0	8.5.0
	RP-43	RP-090129	0067	1	Proposed CR to 36.322 on Clarification on Polling procedure	8.4.0	8.5.0
	RP-43	RP-090129	0068	1	Every Poll_PDU PDUs and Every Poll_Byte bytes triggers	8.4.0	8.5.0
	RP-43	RP-090129	0069	-	UE behaviour when T_poll_retransmit expires	8.4.0	8.5.0
	RP-43	RP-090129	0076	-	Definition of RETX_COUNT missing	8.4.0	8.5.0
2009-06	RP-44	RP-090514	0080	-	Reset of T_poll_retransmission	8.5.0	8.6.0
	RP-44	RP-090514	0081	-	RLC functions	8.5.0	8.6.0
	RP-44	RP-090514	0082	1	Correction to handling of reserved field	8.5.0	8.6.0
	RP-44	RP-090514	0083	-	Correction to condition for stopping t-Reordering in AM mode	8.5.0	8.6.0
2009-09	RP-45	RP-090906	0084	-	Possible misinterpretation on incrementing RETX_COUNT	8.6.0	8.7.0
2009-12	RP-46	RP-091341	0087	-	Capturing MBMS agreements in RLC	8.7.0	9.0.0
2010-03	RP-47	RP-100305	0089	-	Correction to RLC entity	9.0.0	9.1.0
2010-06	RP-48	RP-100536	0091	1	Correction of RLC VR(H) update	9.1.0	9.2.0
2010-09	RP-49	RP-100851	0092	1	Miscellaneous corrections to RLC	9.2.0	9.3.0
2010-12	RP-50	-	-	-	Upgrade to Release 10 - no technical change	9.3.0	10.0.0
2012-09	RP-57	-	-	-	Upgrade to Release 11 - no technical change	10.0.0	11.0.0

ONEPLUS - EX. 1013