TSG-RAN Meeting #13 ***RP-010542***
Beijing, China, 18 - 21 September 2001

Title: **Agreed CRs (Release ’99 and Rel-4 category A) to TS 25.322**

Source: **TSG-RAN WG2**

Agenda item: **8.2.3**

Doc-1st-	Status-	Spec	CR	Rev	Phase	Subject	Cat	Version	Versio
R2-012187	agreed	25.322	141	1	R99	General clarifications	F	3.7.0	3.8.0
R2-012193	agreed	25.322	142		Rel-4	General clarifications	A	4.1.0	4.2.0
R2-012157	agreed	25.322	149	1	R99	Correction to RLC state variables	F	3.7.0	3.8.0
R2-012158	agreed	25.322	150		Rel-4	Correction to RLC state variables	A	4.1.0	4.2.0

3GPP TSG-RAN WG2 Meeting #23 **_Tdoc R2-012187_**
Finland, Helsinki, 27-31 August 2001

CR-Form-v4

CHANGE REQUEST

⌘ **25.322** **CR** **141** ⌘ rev **r1** ⌘ Current version: **3.7.0** ⌘

*For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: ⌘ (U)SIM ☐ ME/UE ☒ Radio Access Network ☒ Core Network ☐

Title:	⌘	General clarifications		
Source:	⌘	TSG-RAN WG2		
Work item code:	⌘	TEI	***Date:*** ⌘	31 August 2001
Category:	⌘	**F**	***Release:*** ⌘	R99

Use one of the following categories:
F *(correction)*
A *(corresponds to a correction in an earlier release)*
B *(addition of feature),*
C *(functional modification of feature)*
D *(editorial modification)*
Detailed explanations of the above categories can be found in 3GPP TR 21.900.

Use one of the following releases:

2	*(GSM Phase 2)*
R96	*(Release 1996)*
R97	*(Release 1997)*
R98	*(Release 1998)*
R99	*(Release 1999)*
REL-4	*(Release 4)*
REL-5	*(Release 5)*

Reason for change: ⌘ Clarify the whole specification.

Summary of change: ⌘ All clauses and subclauses have been modified.

Isolated impact analysis:

- The CR contains only editorial modifications and clarifications.

Consequences if not approved: ⌘ .

Clauses affected: ⌘ 1, 2, 3, 3.1, 3.2, 4.1, 4.2, 4.2.1, 4.2.1.1, 4.2.1.1.1 (new), 4.2.1.1.2 (new), 4.2.1.2, 4.2.1.2.1 (new), 4.2.1.2.2 (new), 4.2.1.3, 4.2.1.3.1 (new), 4.2.1.3.2 (new), 5, 6, 6.1, 6.2, 7, 8, 8.1, 8.2, 9.1, 9.1.1, 9.1.2, 9.2, 9.2.1.1, 9.2.1.2, 9.2.1.3, 9.2.1.4, 9.2.1.5, 9.2.1.6, 9.2.1.7, 9.2.2.1, 9.2.2.3, 9.2.24, 9.2.25, 9.2.2.6, 9.2.2.7, 9.2.2.8, 9.2.2.9, 9.2.2.10, 9.2.2.11, 9.2.2.11.2, 9.2.2.11.3, 9.2.2.11.4, 9.2.2.11.5, 9.2.2.11.6, 9.2.2.11.7, 9.2.2.11.8, 9.2.2.12, 9.2.2.13, 9.3, 9.3.1, 9.3.1.1, 9.3.1.2, 9.3.2, 9.3.2.1, 9.3.2.2, 9.3.2.3, 9.3.3, 9.3.3.1, 9.3.3.2, 9.3.3.3, 9.3.3.4, 9.3.3.5, 9.4, 9.5, 9.6, 9.7, 9.7.1, 9.7.2, 9.7.3, 9.7.3.1, 9.7.3.2, 9.7.3.3, 9.7.3.4, 9.7.3.5, 9.7.4, 9.7.5, 9.7.6, 9.7.7, 9.7.8, 10, 10.1 (new), 10.2 (new), 10.3 (new), 11, 11.1.1, 11.1.2, 11.1.2.1, 11.1.2.2 (new), 11.1.3, 11.1.4.2, 11.2.1, 11.2.2, 11.2.2.1, 11.2.2.2 (new), 11.2.3, 11.2.4.1, 11.2.4.2, 11.2.4.3, 11.3.1, 11.3.2, 11.3.2.1, 11.3.2.1.1, 11.3.2.1.2, 11.3.2.2 (new), 11.3.3, 11.3.4.1, 11.3.4.2, 11.3.4.3.1, 11.3.4.4, 11.3.4.5, 11.3.4.6, 11.4.1, 11.4.2, 11.4.2.1, 11.4.3, 11.4.3.1, 11.4.4, 11.4.5.1, 11.4.5.2, 11.4.5.3, 11.5.1, 11.5.2, 11.5.2.1, 11.5.2.2, 11.5.2.3 (new), 11.5.3, 11.5.4.1, 11.6.1, 11.6.2, 11.6.2.1, 11.6.2.2, 11.6.3, 11.6.3.1 (new), 11.6.4, 11.6.5, 11.6.6.1, 11.6.6.2, 11.6.6.3

Other specs affected: ⌘ ☒ Other core specifications ⌘ 25.322 v4.1.0, CR 142
☐ Test specifications

	□ O&M Specifications	
Other comments: ⌘		

How to create CRs using this form:

Comprehensive information and tips about how to create CRs can be found at: http://www.3gpp.org/3G_Specs/CRs.htm. Below is a brief summary:

1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.

2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be downloaded from the 3GPP server under ftp://ftp.3gpp.org/specs/ For the latest version, look for the directory name with the latest date e.g. 2001-03 contains the specifications resulting from the March 2001 TSG meetings.

3) With "track changes" disabled, paste the entire CR form (use CTRL-A to select it) into the specification just in front of the clause containing the first piece of changed text. Delete those parts of the specification which are not relevant to the change request.

Foreword

This Technical Specification (TS) 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.

1 Scope

The present document specifies the Radio Link Control protocol for the UE-UTRAN radio interface.

Features for the current Release:

- Transparent mode.
- Unacknowledged mode.
- Acknowledged mode.

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 TS 25.401: "UTRAN Overall Description".

[2] 3GPP TR 25.990: "Vocabulary for the UTRAN".

[3] 3GPP TS 25.301: "Radio Interface Protocol Architecture".

[4] 3GPP TS 25.302: "Services Provided by the Physical Layer".

[5] 3GPP TS 25.303: "Interlayer Procedures in Connected Mode".

[6] 3GPP TS 25.304: "UE Procedures in Idle Mode and Procedures for Cell Reselection in Connected Mode".

[7] 3GPP TS 25.321: "MAC Protocol Specification".

[8] 3GPP TS 25.331: "RRC Protocol Specification".

[9] 3GPP TS 33.102: "Security Architecture".

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the terms and definitions given in [2] apply.

3.2 Abbreviations

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

AM	Acknowledged Mode
AMD	Acknowledged Mode Data
ARQ	Automatic Repeat Request

BCCH	Broadcast Control Channel
BCH	Broadcast Channel
C-	Control-
CCCH	Common Control Channel
CCH	Control Channel
CCTrCH	Coded Composite Transport Channel
CRC	Cyclic Redundancy Check
CTCH	Common Traffic Channel
DCCH	Dedicated Control Channel
DCH	Dedicated Channel
DL	Downlink
DSCH	Downlink Shared Channel
DTCH	Dedicated Traffic Channel
FACH	Forward Link Access Channel
FDD	Frequency Division Duplex
L1	Layer 1 (physical layer)
L2	Layer 2 (data link layer)
L3	Layer 3 (network layer)
LI	Length Indicator
LSB	Least Significant Bit
MAC	Medium Access Control
MRW	Move Receiving Window
MSB	Most Significant Bit
PCCH	Paging Control Channel
PCH	Paging Channel
PDU	Protocol Data Unit
PHY	Physical layer
PhyCH	Physical Channels
RACH	Random Access Channel
RLC	Radio Link Control
RRC	Radio Resource Control
SAP	Service Access Point
SDU	Service Data Unit
SHCCH	Shared Channel Control Channel
SN	Sequence Number
SUFI	Super Field
TCH	Traffic Channel
TDD	Time Division Duplex
TFI	Transport Format Indicator
TM	Transparent Mode
TMD	Transparent Mode Data
TTI	Transmission Time Interval
U-	User-
UE	User Equipment
UL	Uplink
UM	Unacknowledged Mode
UMD	Unacknowledged Mode Data
UMTS	Universal Mobile Telecommunications System
UTRA	UMTS Terrestrial Radio Access
UTRAN	UMTS Terrestrial Radio Access Network

4 General

4.1 ~~Void~~Objective

This subclause describes the architecture of the RLC sublayer.

4.2 Overview of~~on~~ the RLC sublayer architecture

The model presented in this subclause is intended to support the definition of the RLC sublayer only, and is not meant to specify or constrain the~~for~~ implementation ~~purposes~~ of the protocol. The RLC sublayer consists of RLC entities, of which there are three types: Transparent Mode (TM), Unacknowledged Mode (UM), and Acknowledged Mode (AM) RLC entities.

4.2.1 Model of the RLC sublayer

Figure 4.1~~ gives an overview model of the RLC layer. The figure~~ illustrates~~ the~~ different RLC~~ peer~~ entities in the RLC model.

An UM and a TM RLC entity can be configured to be a ~~sending~~transmitting RLC entity or a receiving RLC entity. The ~~sending~~transmitting RLC entity transmits RLC PDUs and the receiving RLC entity receives RLC PDUs. An AM RLC entity consists of a transmitting side, and a receiving side, where the transmitting side of the AM RLC entity transmits RLC PDUs and the receiving side of the AM RLC entity receives RLC PDUs.

Elementary procedures (see clause 11) are defined between a "Sender" and a "Receiver". In UM and TM, the ~~sending~~transmitting RLC entity acts as a Sender and the peer RLC entity acts as a Receiver. An AM RLC entity acts either as a Sender or as a Receiver depending on the elementary procedure. The Sender is the transmitter of AMD PDUs and the Receiver is the receiver of AMD PDUs.– A Sender or a Receiver can reside at either the UE or the UTRAN.

There is ~~a ~~one ~~sendingtransmitting~~transmitting and ~~a ~~one receiving RLC entity for each~~the~~ transparent mode (TM) ~~service~~ and ~~the ~~unacknowledged mode (UM) ~~unidirectional ~~service. ~~And~~There is ~~a ~~one combined, ~~sendingtransmitting~~transmitting and receiving entity for the acknowledged mode (AM) ~~bi-directional ~~service.~~.~~

In this specification, ~~the word “~~"transmitted~~”~~" is equivalent to "submitted to the lower layer" unless otherwise explicitly stated. Each RLC UM, and TM entity uses one logical channel to send or receive data PDUs. An AM RLC entity can be configured to use one or two logical channels to send or receive data and control PDUs. If two logical channels are configured, they are of the same type (DCCH or DTCH). In figure 4.1, ~~T~~the dashed lines between the AM-Entities illustrate the possibility to send ~~the~~and receive RLC PDUs on separate logical channels, e.g. control PDUs on one, and data PDUs on the other. More detailed description~~s~~ of the different entities is~~are~~ given in subclauses 4.2.1.1, 4.2.1.2 and 4.2.1.3.

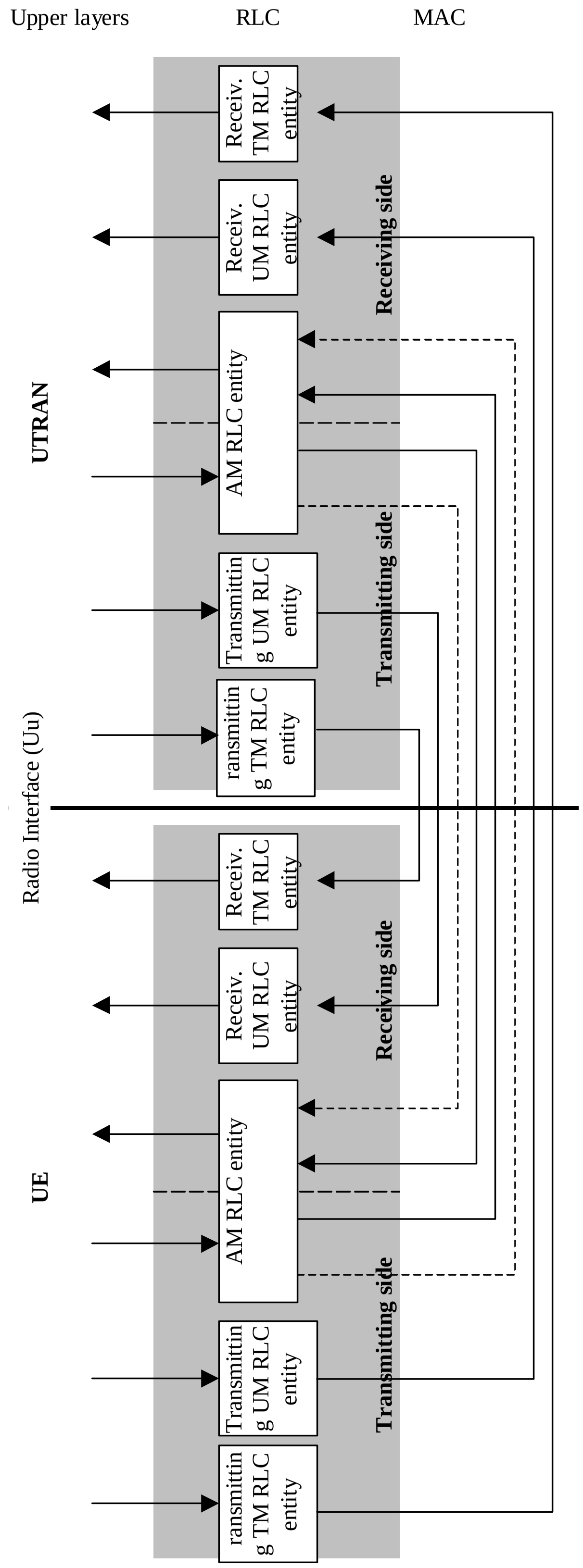


Figure 4.1: Overview model of the RLC sublayer

4.2.1.1 Transparent mode (TM) RLC entities

Figure 4.2 below shows the model of two transparent mode peer RLC entities. The logical channels used to communicate with the lower layer are different depending on the placement of the RLC entity (at the UE or the UTRAN)~~, and are specified~~described in the ~~diagram~~figure below.

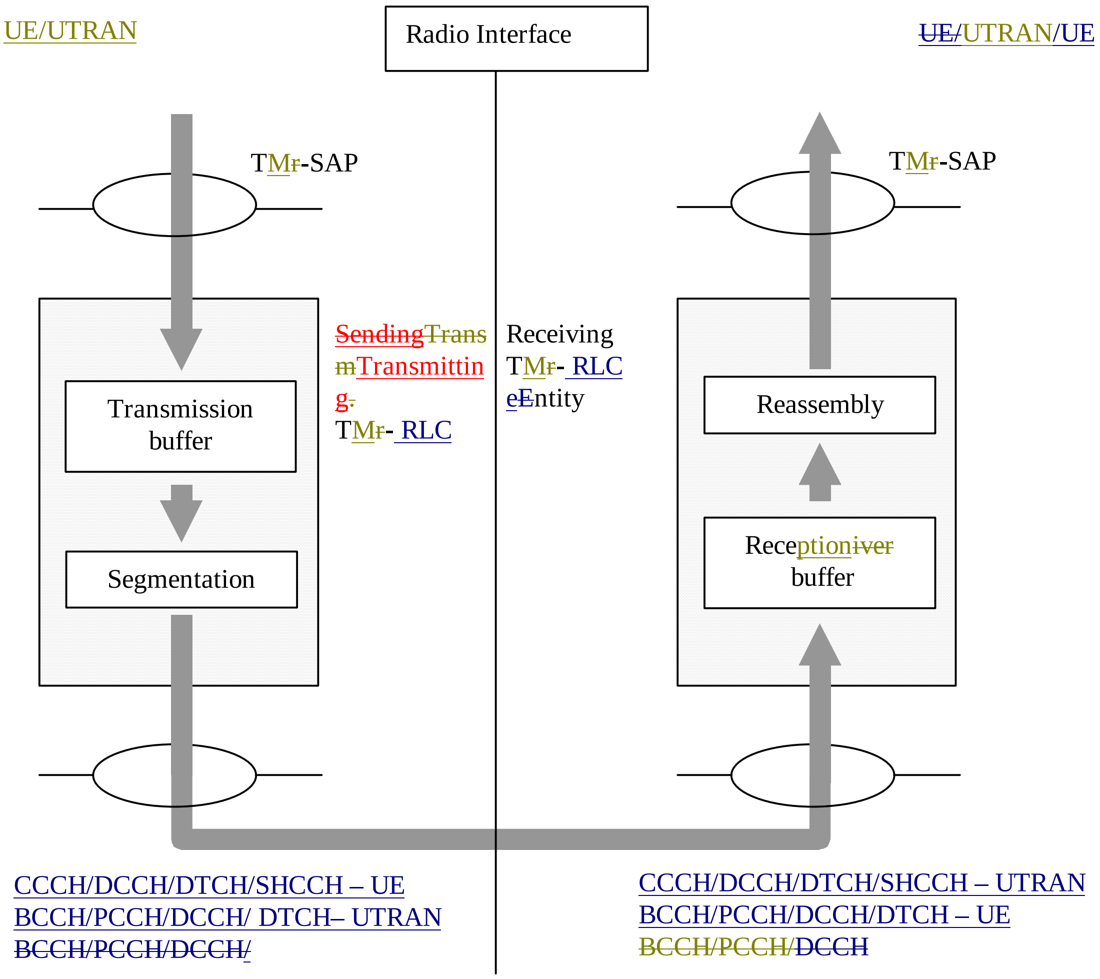


Figure 4.2: Model of two transparent mode peer entities

4.2.1.1.1 ~~Sending~~Transmitting TM RLC entity

The ~~sending~~transmitting~~transmitting~~ TM~~r~~-RLC entity receives ~~SDU~~RLC SDUs from ~~the upper layer~~upper layers through the TM~~r~~-SAP.

All received RLC SDUs must be of a length that is a multiple of one of the ~~configured~~valid TMD PDU lengths.

If segmentation has been configured by upper layers and a RLC SDU is larger than the TMD PDU size used by the lower layer for thata TTI, t~~The~~ ~~sending~~transmitting TM RLC entity ~~maymight~~ segments ~~the~~ ~~SDU~~RLC SDUs to fit the ~~into appropriately sized~~ ~~RLC PDU~~TMD PDUs size without adding ~~any~~ RLC headersoverhead. ~~The segmentation is performed if an RLC SDU is larger thatthan the largest TMD PDU size (configured at the sending TM RLC entity) that is supported by the lower layer for that TTI. If segmentation is performedhas been configured by upper layer, a~~All the TMD PDUs carrying one RLC SDU are~~will be~~ sent in ~~one~~the same TTI, and no segment from another RLC SDU are~~will be~~ sent in this TTI~~only the segments from one RLC SDU will be sent in one TTI~~.

If ~~the~~ segmentation ~~is not used by the TM RLC entity~~has not been configured by upper layers, then more than one RLC SDU can be sent in one TTI by placing one RLC SDU in one TMD PDU. All TMD PDUs in one TTI must be of equal length.

When the processing of ~~an~~ a RLC SDU is complete, the resulting one or more TMD PDU(s) are/is ~~deliver~~submitted to the~~How to perform the segmentation is decided upon when the service is established. RLC delivers the RLC PDUs to~~

lower layer through either a BCCH, DCCH, PCCH, CCCH, SHCCH or a DTCH logical channel~~. The CCCH and SHCCH uses transparent mode only for the uplink. Which type of logical channel depends on if the upper layer is located in the control plane (BCCH, DCCH, PCCH, CCCH, SHCCH) or user plane (DTCH).~~

4.2.1.1.2 Receiving TM RLC entity

The receiving TM~~r~~-RLC entity receives ~~PDU~~TMD PDUs through the configured ~~one of the~~ logical channels from the lower layer.

If segmentation is configured by upper layers, all TMD PDUs received within one TTI are reassembled to form the RLC SDU~~The receiving TM RLC receiving entity reassembles (if segmentation at the sendersending TM RLC peer entity has been performedconfigured by upper layer) the RLC PSDUs from TMD PDUs into RLC SDUs. How to perform the reassembling is decided upon when the service is established. The reassembly of TMD PDUs into an RLC SDU is facilitated through the use of the information passed from the lower layer along with the delivered TMD PDUs in one TTI~~.

If segmentation is not configured by upper layers, each TMD PDU is treated as a RLC SDU.

The receiving TM RLC entity delivers~~ the~~ RLC SDUs to ~~the upper layer upper layer~~upper layers through the TM~~r~~-SAP.

4.2.1.2 Unacknowledged mode (UM) RLC entities

Figure 4.3 below shows the model of two unacknowledged mode peer RLC entities.

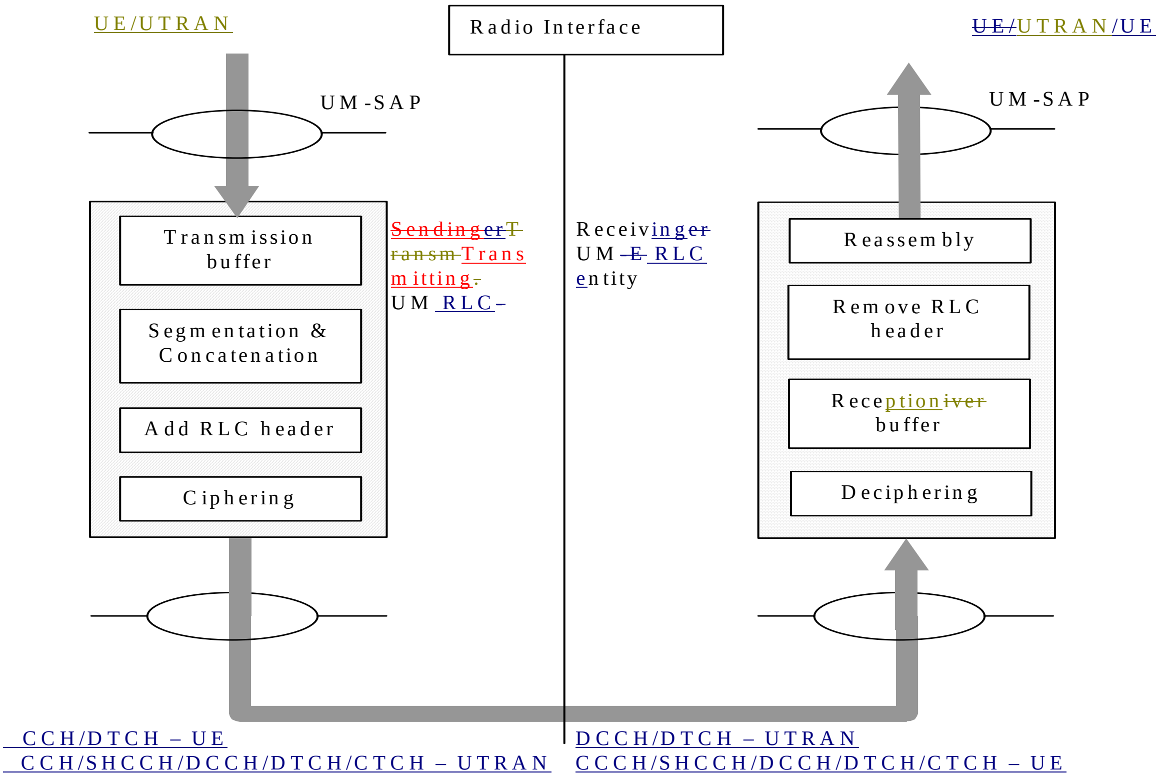


Figure 4.3: Model of two unacknowledged mode peer entities

4.2.1.2.1 ~~Sending~~Transmitting UM RLC entity

The ~~sending~~transmitting~~transmitting~~ UM RLC -entity receives ~~SDU~~RLC SDUs from ~~the upper layer~~upper layers through the UM-SAP.

The ~~sending~~transmitting UM RLC entity ~~maymight~~ segments the RLC SDUs into ~~RLC PDU~~UMD PDUs of appropriate size,~~. The segmentation is performed~~ if the RLC SDU is larger than the length of available space in the UMD PDU~~ of size supported by the lower layer for that TTI~~. ~~The SDU might also be concatenated with other SDUs.~~ The UMD PDU may contain segmented and/or concatenated RLC SDUs. UMD PDU may also contain padding to ensure that it is of a valid length. Length Indicators are used to define boundaries between RLC SDUs within UMD PDUs. Length Indicators are also used to define whether Padding is included in the UMD PDU.

If ~~the~~ ciphering is configured and started, an UMD PDU is ciphered (except for the UMD PDU header) before it is ~~delivered~~submitted to the lower layer.

The ~~sending~~transmitting UM RLC entity ~~deliver~~submits ~~the~~ ~~RLC PDU~~UMD PDUs to the lower layer through either a CCCH, SHCCH, DCCH, CTCH or a DTCH logical channel.~~. The CCCH and SHCCH uses unacknowledged mode only for the downlink. Which type of logical channel depends on if the upper layer is located in the control plane (CCCH, DCCH, SHCCH) or user plane (CTCH, DTCH).~~

4.2.1.2.2 Receiving UM RLC entity

The receiving UM- RLC entity receives ~~PDU~~UMD PDUs through the configured ~~one of the~~ logical channels from the lower layer.~~logical channels from the MAC sublayer~~.

The receiving UM RLC ~~receiving~~ entity deciphers (if ciphering is configured and started) the received UMD PDUs (except for the UMD PDU header). It removes RLC headers from ~~the~~ received ~~PDU~~UMD PDUs, and ~~uses the information placed by the sender in the fields of UMD PDU headers and to~~ reassembles RLC~~the~~ S~~P~~DUs (if segmentation and/or concatenation has been performed by the ~~senderSender~~transmitting UM RLC entity)~~ into RLC SDU~~s.

RLC SDUs are delivered by the receiving UM RLC entity to ~~the upper layer~~upper layers through the UM-SAP.

4.2.1.3 Acknowledged mode (AM) RLC entity

Figure 4.4 below shows the model of an acknowledged mode RLC entity.

The AM RLC entity can be configured to utilize one or two logical channels. The figure 4.4 shows the model of the AM RLC entity when one logical channel (shown as a solid line), and when two logical channels (shown as dashed lines) are used.

If one logical channel is configured, the transmitting side of the AM RLC entity submits AMD and Control PDUs to the lower layer on that logical channel. And the RLC PDU size shall be the same for AMD PDUs and control PDUs.

In case two logical channels are configured in the uplink, AMD PDUs are transmitted on the first logical channel, and control PDUs are transmitted on the second logical channel. In case two logical channels are configured in the downlink, AMD and Control PDUs can be transmitted on any of the two logical channels.

~~In case two logical channels are configured,used in the uplink, the first logical channel shallwillis be used for AMDdata PDUs, and the second logical channel shallwillis be used for control PDUs. In case one logical channel is usedconfigured, the RLC PDU size isshall be the same for AMD PDUs and control PDUs.~~

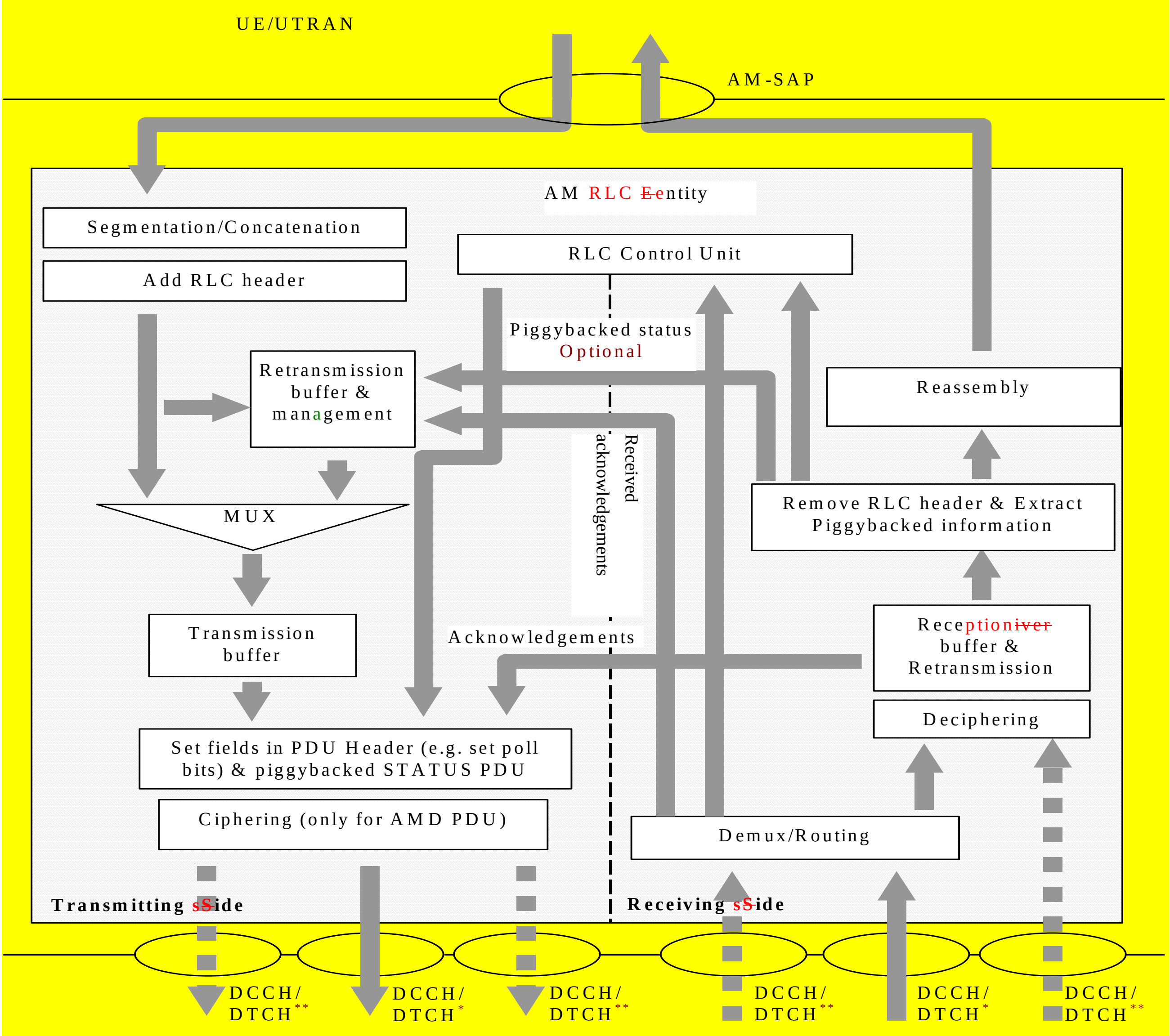

UE/UTRAN
AM-SAP
AM RLC ~~E~~entity
Segmentation/Concatenation
Add RLC header
RLC Control Unit
Piggybacked status
Optional
Retransmission buffer & management
Reassembly
Received acknowledgements
MUX
Remove RLC header & Extract Piggybacked information
Transmission buffer
Acknowledgements
Recep~~tioniver~~ buffer & Retransmission
Deciphering
Set fields in PDU Header (e.g. set poll bits) & piggybacked STATUS PDU
Ciphering (only for AMD PDU)
Demux/Routing
Transmitting s~~S~~ide
Receiving s~~S~~ide
DCCH/DTCH**
DCCH/DTCH*
DCCH/DTCH**
DCCH/DTCH**
DCCH/DTCH*
DCCH/DTCH**

Figure 4.4: Model of an acknowledged mode entity

4.2.1.3.1 Transmitting side

The transmitting side of the AM RLC -entity receives RLC SDUs from ~~the upper layer~~upper layers through the AM-SAP.

~~The~~ RLC SDUs are segmented and/or concatenated into ~~PDU~~AMD PDUs of a fixed length. The segmentation is performed if the received RLC SDU is larger than the length of available space in the AMD PDU. ~~An~~The AMD PDU ~~length~~ size is a semi-static value that is configured by ~~the upper layer~~upper layers~~decided in bearer setup~~, and can ~~only~~ be changed through the~~bearer~~ modification~~reconfiguration~~ of the AM RLC entity by ~~the upper layer~~upper layers. The AMD PDU may contain segmented and/or concatenated RLC SDUs. The AMD PDU may also contain Padding to ensure that it is of a valid ~~length~~size. Length Indicators are used to define boundaries between RLC SDUs within AMD PDUs. Length Indicators are also used to define whether Padding or Piggybacked STATUS PDU is included in the ~~UMD~~AMD PDU.

~~For concatenation or padding purposes, bits of information on the length and extension are inserted into the beginning of the last PDU where data from an SDU is included. Padding can be replaced by piggybacked status information. This includes setting the poll bit.~~

~~If several SDUs fit into one PDU, they are concatenated and the appropriate length indicators are inserted into the beginning of the PDU.~~ After~~ that~~ the segmentation and/or concatenation are performed, the ~~PDU~~AMD PDUs are placed in~~ the retransmission buffer and~~ the Retransmission buffer, and at the MUX.

~~The Retransmission buffer deletes buffered AMD PDUs, or indicates which AMD PDUs are to be retransmitted, based on the input from the receiving side of the RLC AM entity. The input consists of positive and negative acknowledgments of individual AMD PDUs received from the peer AM RLC entity in the status report (Piggybacked STATUS or STATUS PDU).~~AMD PDUs buffered in the Retransmission buffer ~~may be~~are deleted or retransmitted based on the status report found within a STATUS PDU or Piggybacked STATUS PDU sent ~~from~~by the peer AM RLC entity. This status report may contain positive or negative acknowledgements of individual AMD PDUs received ~~from~~by the peer AM RLC entity.

The MUX multiplexes AMD PDUs from the Retransmission buffer that need to be retransmitted, and the newly generated AMD PDUs delivered from the Segmentation/Concatenation function. ~~then decides which PDUs and when the PDUs are submitted to the lower layer.~~

The PDUs are ~~submitted via~~ delivered to the~~a~~ function that completes the AMD PDU header and potentially replaces padding with piggybacked status information. ~~The RLC entity shall assume a PDU to be transmitted when the PDU is submitted to lower layer. When Piggybacking mechanism is applied, t~~The padding in the AMD PDU may be replaced by the Piggybacked STATUS PDU that may be delivered from the Rece~~iver~~ption buffer at the receiving side of the AM RLC entity. ~~The~~A Piggybacked STATUS PDUs can be of variable size in order to match the amount of free space in the AMD PDU. The AMD PDU header is completed based on the input from the RLC Control ~~function~~Unit that indicates the values to set in various fields (e.g. Polling Bit). The function also multiplexes, if required, Control PDUs received from the RLC Control ~~u~~Unit (RESET and RESET ACK PDUs), and from the Rece~~iver~~ption buffer (Piggybacked STATUS and STATUS PDUs), with AMD PDUs~~, if one logical channel is configured. The multiplexing of control and data PDUs is not performed if two logical channels are configured~~.

The ciphering (if ~~it is~~ configured) is then applied~~ only~~ to~~for~~ the AMD PDUs. The ~~fixed 2 octets~~ AMD PDU header is not ciphered. Piggybacked STATUS PDU and Padding~~ parts~~ in~~of~~ AMD PDU (when present~~existing~~) are ciphered. ~~The other~~ Control PDUs (i.e. STATUS PDU, RESET PDU, and RESET ACK PDU) ~~shall~~~~will~~are not ~~be~~ ciphered.

The transmitting side of the AM RLC entity submits AMD PDUs to the lower layer through either one or two DCCH or DTCH logical channels.

~~The transmitting side of the AM RLC entity deliversubmits AMD and Control PDUs to the lower layer through either a DCCH or a DTCH logical channel, if one logical channel is configured. If two logical channels are configured (either of DCCH or DTCH type), AMD and Control PDUs are deliversubmitted to the lower layer on their individual logical channels on the uplink (UE as a Sender). On the downlink (UTRAN as a Sender), AMD and Control PDUs can be submitted to the lower layer on either of the logical channels.~~

~~When Piggybacking mechanism is applied the padding is replaced by control information, in order to increase the transmission efficiency and making possible a faster message exchange between the peer-to-peer RLC entities. The piggybacked control information is not saved in any retransmission buffer. The piggybacked control information is contained in the piggybacked STATUS PDU, which is in turn included into the AMD-PDU. The piggybacked STATUS PDUs will be of variable size in order to match with the amount of free space in the AMD PDU.~~

~~The retransmission buffer also receives acknowledgements from the receiving side, which are used to indicate retransmissions of PDUs and when to delete a PDU from the retransmission buffer.~~

4.2.1.3.2 Receiving side

The receiving side of the AM- RLC entity receives AMD and Control PDUs through ~~one of~~ the configured logical channels from the lower layer~~, if one logical channel is configured~~. ~~If two logical channels are configured, AMD and Control PDUs are received on their individual logical channels from the lower layers~~.

AMD PDUs are routed to the Deciphering U~~u~~nit, where AMD PDUs (minus the ~~mandatory~~ AMD PDU header) are deciphered (if ciphering is configured and started), and then delivered to the Rece~~iver~~ption buffer.

The AMD PDUs are placed in the Reception ~~Receiver~~ buffer until a complete RLC SDU has been received. The Receiver ~~will~~ acknowledges successful reception or requests retransmission of the missing AMD PDUs by sending one or more STATUS PDUs to the AM RLC peer entity, through its transmitting side. If a Piggybacked STATUS PDU is found in an AMD PDU, it is delivered to the Retransmission buffer & Management Unit at the ~~sending~~transmitting side of the AM RLC entity, in order to purge the buffer of positively acknowledged AMD PDUs, and to indicate which AMD PDUs need to be retransmitted.

Once a complete RLC SDU has been received, the associated AMD PDUs are reassembled by the Reassembly ~~function~~Unit and delivered to ~~the upper layer~~upper layers through the AM-SAP.~~ If a Piggybacked STATUS PDU is found in an AMD PDU, it is delivered to the Retransmission buffer & management at the sending side of the AM RLC~~

~~entity, in order to purge the buffer of positively acknowledged AMD PDUs, and to indicate which AMD PDUs need to be retransmitted.~~

RESET and RESET ACK PDUs are ~~routed~~delivered to the RLC Control U~~u~~nit for processing. If a response to the peer AM RLC entity is needed, an appropriate Control PDU is delivered, by the RLC Control~~,~~ U~~u~~nit to the transmitting side of the AM RLC entity. The received STATUS PDUs are ~~routed~~ delivered to the Retransmission buffer and Management Unit at the transmitting side of the AM RLC entity, in order to purge the buffer of positively acknowledged AMD PDUs, and to indicate which AMD PDUs need to be retransmitted.

~~If two logical channels are configured, the Router function is transparent as Control PDUs are delivered directly to the RLC Control function, as are AMD PDUs to the Receiver buffer.~~

~~Piggybacked status information is extracted, if present. The PDUs are placed in the receiver buffer until a complete SDU has been received. The receiver buffer requests retransmissions of PDUs by sending negative acknowledgements to the peer entity. After that the RLC headers are removed from the PDUs and the PDUs are reassembled into an SDU. Finally the SDU is delivered to the upper layer. The receiving side also receives acknowledgements from the peer entity. The acknowledgements are passed to the retransmission buffer on the transmitting side.~~

5 Functions

The following functions are supported by RLC sublayer. For ~~a detailed~~an overall description of the following functions see [3]:

- Segmentation and reassembly.
- Concatenation.
- Padding.
- Transfer of user data.
- Error correction.
- In-sequence delivery of upper layer PDUs.
- Duplicate detection.
- Flow control.
- Sequence number check.
- Protocol error detection and recovery.
- Ciphering.
- ~~Polling.~~
- ~~Status transmission.~~
- SDU discard.
- ~~Estimated PDU Counter (EPC) mechanism.~~
- ~~Suspend/resume function.~~
- ~~Stop/continue function.~~
- ~~Re-establishment function.~~

6 Services provided to upper layers

This clause describes the different services provided by RLC sublayer to upper layers. It also includes the mapping of RLC functions to different RLC services. For a detailed description of the RLC services~~following functions~~ see [3].

- **Transparent data transfer Service:**

The following functions are needed to support~~may be used~~are~~configured needed to support when~~ transparent data transfer~~ is supported~~:

- Segmentation and reassembly.
- Transfer of user data.
- SDU discard.

- **Unacknowledged data transfer Service:**

The following functions are needed to support~~may be used~~are~~configured needed to support when~~ unacknowledged data transfer~~ is supported~~:

- Segmentation and reassembly.
- Concatenation.
- Padding.
- Transfer of user data.
- Ciphering.
- Sequence number check.
- SDU discard.

~~- Suspend/resume function.~~

~~- Stop/continue function.~~

~~- Re-establishment function.~~

- **Acknowledged data transfer Service:**

The following functions are needed to support~~may be used~~are~~configured needed to support when~~ acknowledged data transfer~~ is supported~~:

- Segmentation and reassembly.
- Concatenation.
- Padding.
- Transfer of user data.
- Error correction~~ ()~~.
- In-sequence delivery of ~~upper layer~~ upper layers PDUs.
- Duplicate detection.
- Flow Control.
- Protocol error detection and recovery.
- Ciphering.

~~- Polling.~~

~~- Status transmission.~~

- SDU discard.

~~- Estimated PDU Counter (EPC) mechanism.~~

- ~~Suspend/resume function.~~
- ~~Stop/continue function.~~
- ~~Re-establishment function.~~

- **Maintenance of QoS as defined by upper layers.**
- **Notification of unrecoverable errors.**

6.1 Mapping of services/functions onto logical channels

The following tables show the applicability of services and functions to the logical channels in UL/DL and UE/UTRAN. A '+' in a column denotes that the service/function is applicable for the logical channel in question whereas a '-' denotes that the service/function is not applicable.

Table 6.1: RLC modes and functions in UE uplink side

Service	Functions	CCCH	SHCCH	DCCH	DTCH
Transparent Service	Applicability	+	+	+	+
	Segmentation	-	-	+	+
	Transfer of user data	+	+	+	+
	SDU Discard	-	-	+	+
Unacknowledged Service	Applicability	-	-	+	+
	Segmentation	-	-	+	+
	Concatenation	-	-	+	+
	Padding	-	-	+	+
	Transfer of user data	-	-	+	+
	Ciphering	-	-	+	+
	SDU Discard	-	-	+	+
	~~Suspend/Resume Function~~	~~-~~	~~-~~	~~+~~	~~+~~
	~~Stop/Continue Function~~	~~-~~	~~-~~	~~+~~	~~+~~
	~~Re-establishment FunctionSDU Discard~~	~~-~~	~~-~~	~~+~~	~~+~~
Acknowledged Service	Applicability	-	-	+	+
	Segmentation	-	-	+	+
	Concatenation	-	-	+	+
	Padding	-	-	+	+
	Transfer of user data	-	-	+	+
	Flow Control	-	-	+	+
	Error Correction	-	-	+	+
	Protocol error correction & recovery	-	-	+	+
	Ciphering	-	-	+	+
	~~Polling~~	~~-~~	~~-~~	~~+~~	~~+~~
	SDU Discard	-	-	+	+
	~~Suspend/Resume Function~~	~~-~~	~~-~~	~~+~~	~~+~~
	~~Stop/Continue Function~~	~~-~~	~~-~~	~~+~~	~~+~~
	~~Re-establishment Function~~	~~-~~	~~-~~	~~+~~	~~+~~

Table 6.2: RLC modes and functions in UE downlink side

Service	Functions	BCCH	PCCH	SHCCH	CCCH	DCCH	DTCH	CTCH
Transparent Service	Applicability	+	+	-	-	+	+	-
	Reassembly	-	-	-	-	+	+	-
	Transfer of user data	+	+	-	-	+	+	-
Unacknowledged Service	Applicability	-	-	+	+	+	+	+
	Reassembly	-	-	+	+	+	+	+
	Deciphering	-	-	-	-	+	+	-
	Sequence number check	-	-	+	+	+	+	+
	Transfer of user data	-	-	+	+	+	+	+
	~~Suspend/Resume Function~~	~~-~~	~~-~~	~~-~~	~~-~~	~~+~~	~~+~~	~~-~~
	~~Stop/Continue Function~~	~~-~~	~~-~~	~~-~~	~~-~~	~~+~~	~~+~~	~~-~~
	~~Re-establishment FunctionTransfer of user data~~	~~--~~	~~--~~	~~-+~~	~~-+~~	~~++~~	~~++~~	~~-+~~
Acknowledged Service	Applicability	-	-	-	-	+	+	-
	Reassembly	-	-	-	-	+	+	-
	Error correction	-	-	-	-	+	+	-
	Flow Control	-	-	-	-	+	+	-
	In sequence delivery	-	-	-	-	+	+	-
	Duplicate detection	-	-	-	-	+	+	-
	Protocol error correction & recovery	-	-	-	-	+	+	-
	Deciphering	-	-	-	-	+	+	-
	Transfer of user data	-	-	-	-	+	+	-
	~~Status Transmission~~	~~-~~	~~-~~	~~-~~	~~-~~	~~+~~	~~+~~	~~-~~
	SDU DIscard	-	-	-	-	+	+	-
	~~Suspend/Resume Function~~	~~-~~	~~-~~	~~-~~	~~-~~	~~+~~	~~+~~	~~-~~
	~~Stop/Continue Function~~	~~-~~	~~-~~	~~-~~	~~-~~	~~+~~	~~+~~	~~-~~
	~~Re-establishment Function~~	~~-~~	~~-~~	~~-~~	~~-~~	~~+~~	~~+~~	~~-~~

Table 6.3: RLC modes and functions in UTRAN downlink side

Service	Functions	BCCH	PCCH	CCCH	SHCCH	DCCH	DTCH	CTCH
Transparent Service	Applicability	+	+	-	-	+	+	-
	Segmentation	-	-	-	-	+	+	-
	Transfer of user data	+	+	-	-	+	+	-
	SDU Discard	-	-	-	-	+	+	-
Unacknowledged Service	Applicability	-	-	+	+	+	+	+
	Segmentation	-	-	+	+	+	+	+
	Concatenation	-	-	+	+	+	+	+
	Padding	-	-	+	+	+	+	+
	Ciphering	-	-	-	-	+	+	-
	Transfer of user data	-	-	+	+	+	+	+
	SDU Discard	-	-	-	-	+	+	-
	~~Suspend/Resume Function~~	~~-~~	~~-~~	~~-~~	~~-~~	~~+~~	~~+~~	~~-~~
	~~Stop/Continue Function~~	~~-~~	~~-~~	~~-~~	~~-~~	~~+~~	~~+~~	~~-~~
	~~Re-establishment FunctionSDU Discard~~	~~--~~	~~--~~	~~--~~	~~--~~	~~++~~	~~++~~	~~--~~
Acknowledged Service	Applicability	-	-	-	-	+	+	-
	Segmentation	-	-	-	-	+	+	-
	Concatenation	-	-	-	-	+	+	-
	Padding	-	-	-	-	+	+	-
	Transfer of user data	-	-	-	-	+	+	-
	Flow Control	-	-	-	-	+	+	-
	Error Correction	-	-	-	-	+	+	-
	Protocol error correction & recovery	-	-	-	-	+	+	-
	Ciphering	-	-	-	-	+	+	-
	~~Polling~~	~~-~~	~~-~~	~~-~~	~~-~~	~~+~~	~~+~~	~~-~~
	SDU Discard					+	+	-
	~~Suspend/Resume Function~~	~~-~~	~~-~~	~~-~~	~~-~~	~~+~~	~~+~~	~~-~~
	~~Stop/Continue Function~~	~~-~~	~~-~~	~~-~~	~~-~~	~~+~~	~~+~~	~~-~~
	~~Re-establishment Function~~	~~-~~	~~-~~	~~-~~	~~-~~	~~+~~	~~+~~	~~-~~

Table 6.4: RLC modes and functions in UTRAN uplink side

Service	Functions	CCCH	SHCCH	DCCH	DTCH
Transparent Service	Applicability	+	+	+	+
	Reassembly	-	-	+	+
	Transfer of user data	+	+	+	+
Unacknowledged Service	Applicability	-	-	+	+
	Reassembly	-	-	+	+
	Deciphering	-	-	+	+
	Sequence number check	-	-	+	+
	Transfer of user data	-	-	+	+
	~~Suspend/Resume Function~~	~~-~~	~~-~~	~~+~~	~~+~~
	~~Stop/Continue Function~~	~~-~~	~~-~~	~~+~~	~~+~~
	~~Re-establishment FunctionTransfer of user data~~	~~--~~	~~--~~	~~++~~	~~++~~
Acknowledged Service	Applicability	-	-	+	+
	Reassembly	-	-	+	+
	Error correction	-	-	+	+
	Flow Control	-	-	+	+
	In sequence delivery	-	-	+	+
	Duplicate detection	-	-	+	+
	Protocol error correction & recovery	-	-	+	+
	Deciphering	-	-	+	+
	Transfer of user data	-	-	+	+
	~~Status Transmission~~	~~-~~	~~-~~	~~+~~	~~+~~
	SDU DIscard	-	-	+	+
	~~Suspend/Resume Function~~	~~-~~	~~-~~	~~+~~	~~+~~
	~~Stop/Continue Function~~	~~-~~	~~-~~	~~+~~	~~+~~
	~~Re-establishment Function~~	~~-~~	~~-~~	~~+~~	~~+~~

7 Services expected from MAC

For a detailed description of the service provided by the MAC sublayer to upper layers~~following functions~~ see [3].

- Data transfer.

8 Elements for layer-to-layer communication

The interaction between the RLC sublayer and other layers are described in terms of primitives where the primitives represent the logical exchange of information and control between the RLC sublayer and other layers. The primitives shall not specify or constrain the implementation~~s~~.

8.1 Primitives between RLC and upper layers

The primitives between RLC and upper layers are shown in Table 8.1.

Table 8.1: Primitives between RLC and upper layers

Generic Name	Parameters			
	Req.	Ind.	Resp.	Conf.
RLC-AM-DATA	Data, CNF, MUI	Data, DiscardInfo	Not Defined	MUI
RLC-UM-DATA	Data, Use special LI	Data	Not Defined	Not Defined
RLC-~~TR~~TM-DATA	Data	Data	Not Defined	Not Defined
CRLC-CONFIG	E/R, Stop, Continue, Ciphering Elements (UM/AM only), TM_parameters (TM only), UM_parameters (UM only), AM_parameters (AM only)	Not Defined	Not Defined	Not Defined
CRLC-SUSPEND (UM/AM only)	N	Not Defined	Not Defined	VT(US) (UM only), VT(S) (AM only)
CRLC-RESUME (UM/AM only)	No Parameter	Not Defined	Not Defined	Not Defined
CRLC-STATUS	Not Defined	EVC	Not Defined	Not Defined

Each Primitive is defined as follows:

RLC-AM-DATA-Req/Ind/Conf

- RLC-AM-DATA-Req is used by upper layers to request transmission of~~ an~~ a RLC SDU~~upper layer PDU~~ in acknowledged mode.

- RLC-AM-DATA-Ind is used by the AM RLC entity to deliver to upper layers~~ ~~ a RLC SDU~~s~~ that has~~ve~~ been transmitted in acknowledged mode, and to indicate to upper layers of the discarded RLC SDU in the peer RLC AM entity.

- RLC-AM-DATA-Conf is used by the AM RLC entity to confirm to upper layers the reception of~~ an~~ a RLC SDU by the peer-RLC AM entity.

RLC-UM-DATA-Req/Ind

- RLC-UM-DATA-Req is used by upper layers to request transmission of~~ an~~ a RLC SDU ~~upper layer PDU~~ in unacknowledged mode~~, and to request the use of the special procedure when handling the RLC SDU~~.

- RLC-UM-DATA-Ind is used by the UM RLC entity to deliver to upper layers~~ ~~ a RLC SDU~~s~~ that has~~ve~~ been transmitted in unacknowledged mode.

RLC-~~TR~~TM-DATA-Req/Ind

- RLC-~~TR~~TM-DATA-Req is used by upper layers to request transmission of~~ an~~ a RLC SDU~~upper layer PDU~~ in transparent mode.

- RLC-~~TR~~TM-DATA-Ind is used by the TM RLC entity to deliver to upper layers a RLC SDU~~s~~ that has~~ve~~ been transmitted in transparent mode.

CRLC-CONFIG-Req

This primitive is used by upper layers to establish, re-establish, release, stop, continue or ~~reconfigure~~ modify the RLC. Ciphering elements are included for UM and AM operation.

CRLC-SUSPEND-Req/Conf

~~This primitive~~

- CRLC-SUSPEND-Req is used by upper layers to suspend the UM or AM RLC entity~~, after a specified number of SNsstarting at a specified time~~.

- CRLC-SUSPEND-Conf is used by the UM or AM RLC entity to ~~inform upper layers of its state variable~~confirm that the entity is suspended.~~ The N parameter indicates that RLC shall not send a PDU with SN>=VT(S)+N for AM and SN>=VT(US)+N for UM, where N is an integer. RLC informs upper layers of the VT(S) for AM and VT(US) for UM in the confirm primitive.~~

CRLC-RESUME-Req

This primitive is used by upper layers to resume the UM or AM RLC entity after~~when~~ the UM or AM RLC entity has been suspended.

CRLC-STATUS-Ind

It is used by an~~the~~ RLC entity to send status information to upper layers.

8.2 Primitive parameters

Following parameters are used in the primitives:

1) The parameter Data is the RLC SDU that is ~~processed by an RLC sending entity (via segmentation/concatenation) and~~ mapped onto the Data field in RLC PDUs.~~ The Data parameter may be divided over several RLC PDUs. In~~ When AM or UM RLC entities are used,~~,case of an RLC-AM-DATA or an RLC-UM-DATA primitive~~ the length of the Data parameter ~~shallwill be~~is a multiple of 8 bits~~octet-aligned~~, otherwise (TM RLC entity) the length of Data parameter is a bit-string whose length may not be a multiple of 8 bits.~~.~~

2) The parameter Confirmation ~~r~~Request (CNF) indicates whether the ~~sending~~transmitting side of the AM RLC entity needs to confirm the reception of the RLC SDU by the peer-RLC AM entity. If required, once all AMD PDUs that make up the RLC SDU are positively acknowledged by the receiving AM RLC entity, the ~~sending~~transmitting AM RLC entity~~ will~~ notifies~~y the upper layer~~upper layers.

3) The parameter Message Unit Identifier (MUI) is an identity of the RLC SDU, which is used to indicate which RLC SDU that is confirmed with the RLC-AM-~~DATA~~ DATA-~~conf~~Conf. primitive.

4) The parameter E/R indicates ~~the~~ establishment, ~~(~~re~~)~~-establishment, release or modification of~~ an~~ a RLC entity, where re-establishment is applicable to the~~for~~ AM and ~~the~~ UM RLC entities only. If ~~the it indicates~~ re-establishment is requested, the state variables and configurable parameters are initialised according to subclause 9.7.7. If ~~it indicatesthe~~ release is requested, all protocol parameters, variables and timers ~~shallwill be~~are released and the RLC entity ~~shallwill exit the DATA_TRANSFER_READYdata transfer ready~~enters the NULL state. If ~~theit indicates~~ modification is requested, the protocol parameters indicated by upper layers (e.g. ciphering parameters) ~~shallwill~~ only are~~be~~ modified, while~~ith~~ keeping the other protocol parameters, such as the protocol variables, ~~the~~ protocol timers and ~~the~~ protocol state unchanged. AM RLC entities ~~shallwill~~are always~~ be~~ re-established if the AMD PDU size is changed. The modification of other protocol parameters does not ~~warrant~~ require a re-establishment.

5) The parameter Event Code (EVC) indicates the reason for the CRLC-STATUS-~~ind~~Ind (e.g., unrecoverable errors such as data link layer loss or recoverable status events such as reset.).

6) The parameter ~~c~~Ciphering ~~e~~Elements are only applicable for UM and AM operations. These parameters are Ciphering Mode, Ciphering Key, Transmitting Activation Time (SN to activate a new ciphering configuration at the ~~sender~~Sender~~transmitter~~), Receiving Activation Time (SN to activate a new ciphering configuration at the ~~receiver~~Receiver) and HFN (Hyper Frame Number).

7) The AM_parameters are only applicable for AM operation. ~~It contains~~These parameters are AMD PDU size, In-sequence Delivery Indication (indicating that RLC SDUs ~~shallwill be~~are delivered to ~~the~~ upper layers in sequence or out of sequence), Timer values (see subclause 9.5), Protocol parameter values (see subclause 9.6), Polling triggers (see subclause 9.7.1), Status triggers (see subclause 9.7.2), Periodical Status blocking configuration (see subclause 9.7.2), SDU discard mode (see subclause 9.7.3), Minimum WSN (see subclause 9.2.2.11.3), and Send MRW. The Minimum WSN ~~shallwill~~is always ~~be~~ greater than or equal to the number of transport blocks in the smallest transport block set. The Send MRW indicates that the information of each discarded ~~SDU~~RLC SDU ~~shallwill be~~is sent to the ~~receiver~~Receiver, and the MRW SUFI ~~shallwill be~~is sent to the ~~receiver~~Receiver even if no segments of the ~~SDU~~RLC SDU to be discarded were submitted to a lower layer.

8) The parameter DiscardInfo indicates to upper layers the discarded RLC SDU in the peer-RLC AM entity. It is applicable only when in-sequence delivery is configured,~~active~~ and it is ~~purposed~~ to be used when ~~the~~ upper layers requires the reliable data transfer~~. and especially the information of the discarded RLC SDU~~.

9) The Stop parameter indicates ~~that~~ to the RLC entity ~~shall will~~to (see subclause 9.7.6):

- discard all ~~not transmit~~ n~~or receive~~ RLC PDUs received from the lower layer~~(see subclause …).~~.

- not submit to lower layer any RLC PDUs.

10) The Continue parameter indicates to~~that~~ the RLC entity ~~shallwill~~ to continue transmission and reception of RLC PDUs.

11) The parameter Use special LI indicates that the LI indicating that ~~an~~ a RLC SDU begins in the beginning of ~~an~~ a RLC PDU ~~(the first data octet of the PDU is the first octet of an SDU) shallwill~~is to be used (see subclause 9.2.2.8)~~. If the RLC SDU does not begin in the beginning of the RLC PDU, or if the LI indicating that an SDU ended exactly in the end or one octet short (only when 15 bit LI is used) of the previous RLC PDU is present, the special LI shallwill not be used~~.

12) The UM_parameters are only applicable for UM operation. It contains Timer_Discard value (see subclause 9.5) and largest UMD PDU size (see subclause 9.2.2.8).

13) The TM_parameters are only applicable for TM operation. It contains e.g. segmentation indication (see subclauses 9.2.2.9 and 11.1.2.1), Timer_Discard value (see subclause 9.5) and delivery of erroneous SDU indication (see subclause 11.1.3).

14) The N parameter indicates that ~~an~~ a RLC entity will not send a PDU with SN>=VT(S)+N for AM and SN>=VT(US)+N for UM, where N is a~~n~~ non-negative integer.

15) The VT(S) parameter indicates ~~that~~ the value of the Send State Variable for the case of the AM.

16) The VT(US) parameter indicates ~~that~~ the value of the UM Data State Variable, for the case of the UM.

9 Elements for peer-to-peer communication

9.1 Protocol data units

The ~~PDUs~~structures defined in this subclause are normative.

9.1.1 Data PDUs

a) TM~~r~~D PDU (Transparent Mode Data PDU).

The TM~~r~~D PDU is used to convey RLC SDU data without adding any RLC overhead. The TM~~r~~D PDU is used by RLC when it is in transparent mode.

b) UMD PDU (Unacknowledged Mode Data PDU).

The UMD PDU is used to convey sequentially numbered PDUs containing RLC SDU data. It is used by RLC when using unacknowledged data transfer.

c) AMD PDU (Acknowledged Mode Data PDU).

The AMD PDU is used to convey sequentially numbered PDUs containing RLC SDU data. The AMD PDU is used by RLC when it is in acknowledged mode.

9.1.2 Control PDUs

a) STATUS PDU and Piggybacked STATUS PDU

The STATUS PDU and the Piggybacked STATUS PDU are used in acknowledged mode:

- by the ~~receiving~~ Receiver ~~entity~~ to inform the ~~transmitting~~ Sender ~~entity~~ about missing and received AMD PDUs in~~at~~ the Receiver~~receiving entity~~;
- by the ~~receiving entity~~Receiver to inform the ~~transmitting entity~~Sender about the size of the allowed transmission window;
- ~~and~~ by the ~~transmitting entity~~Sender to request the ~~receiving entity~~Receiver to move the ~~receiving~~ reception window;~~.~~and
- ~~and~~ by the Receiver to acknowledge the Sender about the reception of the request to move the reception window~~Move Receiving Window SUFI~~.

b) RESET PDU

The RESET PDU is used in acknowledged mode to reset all protocol states, protocol variables and protocol timers of the peer RLC entity in order to synchronise the two peer entities.

c) RESET ACK PDU

The RESET ACK PDU is an acknowledgement to the RESET PDU.

Table 9.1: RLC PDU names and descriptions

Data Transfer Mode	PDU name	Description
Transparent	TM~~r~~D	Transparent mode data
Unacknowledged	UMD	Sequenced unacknowledged mode data
Acknowledged	AMD	Sequenced acknowledged mode data
	STATUS	Solicited or Unsolicited Status Report, Change window size command, SDU discard command, or SDU discard acknowledgement
	Piggybacked STATUS	Piggybacked Solicited or Unsolicited Status Report, Change window size command, SDU discard command, or SDU discard acknowledgement
	RESET	Reset Command
	RESET ACK	Reset Acknowledgement

9.2 Formats and parameters

The ~~data~~ formats of PDUs and their parameters defined in this subclause are normative.

9.2.1 Formats

This subclause specifies the format of the RLC PDUs. The parameters of each PDU are explained in subclause 9.2.2.

9.2.1.1 General

An RLC PDU is a bit string~~, with a length not necessarily a multiple of 8 bits~~. In the ~~drawings ~~figures in subclause 9.2, bit strings are represented by tables in which the first bit is the leftmost one on the first line of the table, the last bit is the rightmost one on 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.

Depending on the provided service, RLC SDUs are bit strings, with any non-null length, or bit strings with a multiple of 8 bits~~an integer number of octets~~ in length.~~ An ~~ A RLC SDU is included into~~ an ~~ a RLC PDU from first bit onward.

9.2.1.2 TM~~r~~D PDU

The TM~~r~~D PDU is used to transfer~~s~~ user data when RLC is operating in transparent mode. No overhead is added to the SDU by RLC. The data length is not constrained to be a multiple of 8 bits~~an integer number of octets~~.

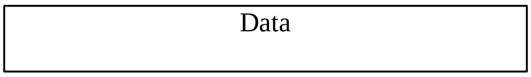


Figure 9.1: TM~~r~~D PDU

9.2.1.3 UMD PDU

The UMD PDU is used to transfer~~s~~ user data when RLC is operating in unacknowledged mode. The length of the data part shall be a multiple of 8 bits~~an integer number of octets~~. The UMD PDU header consists of the first octet, which contains the "S~~s~~equence N~~n~~umber". The RLC header consists of the first octet and all the octets that contain "~~l~~Length ~~i~~Indicators".

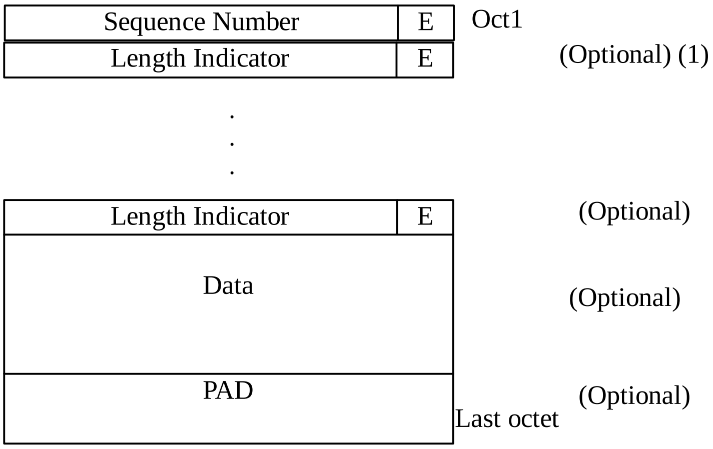
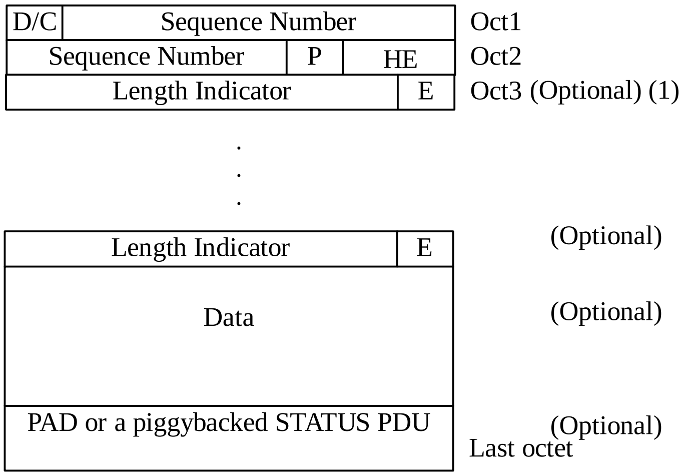


Figure 9.2: UMD PDU

NOTE (1): The~~ Length Indicator ~~ "Length Indicator" may be 15 bits.

9.2.1.4 AMD PDU

The AMD PDU is used to transfer~~s~~ user data~~, and~~ piggybacked status information and ~~requests status report by setting~~the Polling bit when RLC is operating in acknowledged mode. The length of the data part shall be a multiple of 8 bits~~an integer number of octets~~. The AMD PDU header consists of the first two octets, which contain the "S~~s~~equence N~~n~~umber". The RLC header consists of the first two octets and all the octets that contain "L~~l~~ength I~~i~~ndicators".



NOTE (1): The~~ Length Indicator ~~ "Length Indicator" may be 15 bits.

Figure 9.3: AMD PDU

9.2.1.5 STATUS PDU

The STATUS PDU is used to report the status between two RLC AM entities~~. Both receiver and transmitter status information may be included in the same STATUS PDU~~.

The format of the STATUS PDU is given in Figure 9.4 below. The Figure shows an example of STATUS PDU and the length of each SUFI is dependent on the SUFI type.

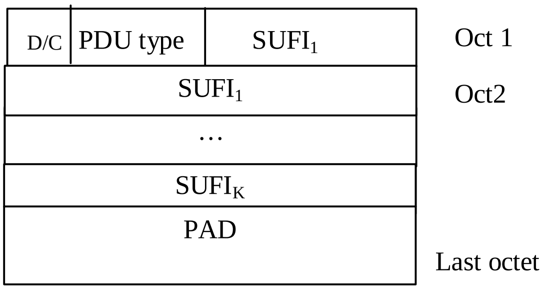


Figure 9.4: ~~Status Information Control PDU (~~STATUS PDU~~)~~

Up to K super-fields (SUFI$_1$-SUFI$_K$) can be included into one STATUS PDU, in which each super-field can be of different type. The size of a STATUS PDU is variable and upper bounded by the maximum RLC PDU size used by the logical channel on which the control PDUs are sent. Padding shall be included to exactly fit one of the PDU sizes used by the logical channel on which the control PDUs are sent. The length of the STATUS PDU shall be a multiple of 8 bits~~an integer number of octets~~.

9.2.1.6 Piggybacked STATUS PDU

The format of the piggybacked STATUS PDU is the same as for ~~the ordinary Control~~STATUS PDU except that the D/C field is replaced by a reserved bit (R2). This PDU can be ~~used to ~~piggybacked ~~STATUS PDU ~~in an AMD PDU if the data does not fill the complete AMD PDU. The PDU Type field is set to ~~zero ~~“000” and all other values are invalid for this version of the protocol and the PDU is discarded.

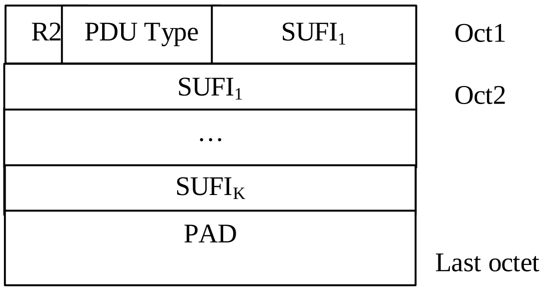


Figure 9.5: Piggybacked STATUS PDU

9.2.1.7 RESET, RESET ACK PDU

The RESET PDU (~~and~~ the RESET ACK PDU) have a one-bit sequence number field (RSN)~~. With the aid of this field the Receiver can define whether the received RESET PDU is transmitted by the Sender for the first time or~~ in order to know whether or not it is ~~or not~~ a retransmission of a previous RESET PDU (of a previous RESET ACK PDU).

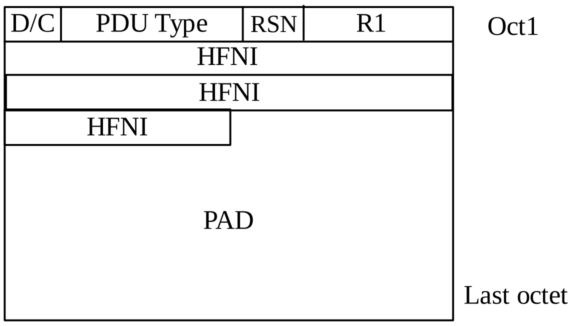


Figure 9.6: RESET, RESET ACK PDU

The size of a RESET or RESET ACK PDU is variable and upper bounded by the maximum RLC PDU size used by the logical channel on which the control PDUs are sent. Padding shall be included to exactly fit one of the PDU sizes used by the logical channel on which the control PDUs are sent. The length of the RESET or RESET ACK PDU shall be a multiple of 8 bits~~an integer number of octets~~.

9.2.2 Parameters

If not otherwise mentioned in the definition of each field then the bits in the parameters shall be interpreted as follows: the left-most bit string is the first and most significant and the right most bit is the last and least significant bit.

Unless otherwise mentioned, integers are encoded in standard binary encoding for unsigned integers. In all cases, including when a value extends over more than one octet as shown in the tables, the bits appear ordered from MSB to LSB when read in the PDU.

9.2.2.1 D/C field

Length: 1bit.

The D/C field indicates the type of an ~~acknowledged mode~~AM PDU. It can be either data or control PDU.

Bit	Description
0	Control PDU
1	~~Acknowledged mode~~ Ddata PDU

9.2.2.2 PDU Type

Length: 3 bit.

The PDU type field indicates the Control PDU type.

Bit	PDU Type
000	STATUS
001	RESET
010	RESET ACK
011-111	Reserved (PDUs with this coding will be discarded by this version of the protocol).

9.2.2.3 Sequence Number (SN)

This field indicates the "S~~s~~equence N~~n~~umber" of the PDU, encoded in binary.

PDU type	Length	Notes
AMD PDU	12 bits	Used for retransmission and reassembly
UMD PDU	7 bits	Used for reassembly

9.2.2.4 Polling bit (P)

Length: 1bit.

This field is used to request a status report (one or several STATUS PDUs) from the ~~receiver RLC~~Receiver.

Bit	Description
0	Status report not requested
1	Request a status report

9.2.2.5 Extension bit (E)

Length: 1bit.

This bit indicates if the next octet will be a "~~l~~Length I~~i~~ndicator" and E bit.

Bit	Description
0	The next field is data, piggybacked STATUS PDU or padding
1	The next field is Length Indicator and E bit

9.2.2.6 Reserved 1 (R1)

Length: 3 bits.

This field in the RESET PDU and RESET ACK PDU is used to ~~achieve octet alignment~~have a multiple of 8 bits in length and for this purpose it is coded as "000". Other functions of it are left for future releases.

9.2.2.7 Header Extension Type (HE)

Length: 2 bits.

This two-bit field indicates if the next octet will be data or a "L~~l~~ength I~~i~~ndicator" and E bit.

Value	Description
00	The succeeding octet contains data
01	The succeeding octet contains a length indicator and E bit
10-11	Reserved (PDUs with this coding will be discarded by this version of the protocol).

9.2.2.8 Length Indicator (LI)

The "Length Indicator" is used to indicate, each time, the end of~~ an~~ a ~~SDU~~RLC SDU occurs in the PDU.

Unless for the predefined values reserved for special purposes and listed in the tables below, ~~The~~the "Length Indicator" shall:

- be set to ~~points out~~ the number of octets between the end of the ~~last Length Indicator field~~RLC header and up to and including the octet at the end of~~ an~~ a ~~SDU~~RLC SDU segment;
- be ~~. Length Indicators are~~ included in the PDUs that they refer to.

The size of the~~ Length Indicator~~ "Length Indicator" may be either 7 bits or 15 bits. The value of a~~ Length Indicator~~ "Length Indicator" shall not exceed the values specified in subclauses 11.2.4.2 and 11.3.4.5.

~~A Length Indicator group is a set of Length Indicators that refer to a PDU.~~ The "Length Indicators" which refer to the same PDU ~~that are part of a Length Indicator group must never~~shall:

- not be reordered ~~within the Length Indicator group or~~nor removed ~~from the Length Indicator group~~in case of retransmission;
- be in the same order as the ~~SDU~~RLC SDUs that they refer to.~~.~~

~~If there can be more than one Length Indicator, each specifying the end of an SDU in a PDU, the order of these Length Indicators must be in the same order as the SDUs that they refer to.~~

For AM:

- 7-bit "Length Indicators" shall be used if the "AMD PDU size" is ≤ l26 octets;
- else, 15-bit "Length Indicators" shall be used;
- the size of the "Length Indicator" is always the same for all AMD PDUs, for one RLC entity.

For UM:

- 7-bit "Length Indicators" shall be used if the "largest UMD PDU size" is ≤ l25 octets;
- else 15-bit "Length Indicators" shall be used;
- between modifications of the "largest UMD PDU size", the size of the "Length Indicator" is the same for all UMD PDUs;
- the "Length Indicator" with value "111 1100" if 7-bit "Length Indicator" is used or "111 1111 1111 1100" if 15-bit "Length Indicator" is used shall be used:
    - if the parameter Use special LI is configured; and
    - if the RLC SDU begins in the beginning of the RLC PDU; and
    - if the "Length Indicators" indicating that~~ an~~ a ~~SDU~~RLC SDU ended exactly in the end or one octet short (only when 15-bit "Length Indicators" is used) of the previous RLC PDU is not present.

In the case where the end of the last segment of~~ an~~ a ~~SDU~~RLC SDU exactly ends at the end of a PDU and there is no ~~LI~~"Length Indicator" that indicates the end of the ~~SDU~~RLC SDU:~~,~~

- ~~the next~~ a "Length Indicator" with value "000 0000" if 7-bit "Length Indicator" is used or "000 0000 0000 0000" if 15-bit "Length Indicator" is used, shall be placed as the first ~~Length Indicator~~ "Length Indicator" in the following PDU ~~and have value LI=0~~.

In case this ~~SDU~~RLC SDU was the last one to be transmitted, a RLC PDU may be transmitted, this RLC PDU consists of:~~,~~

- ~~a PDU consisting of an~~ a RLC Header;
- ~~with Li~~a "Length Indicator" with value ~~=0~~ "000 0000" if 7-bit "Length Indicator" is used or "000 0000 0000 0000" if 15-bit "Length Indicator" is used;
- ~~followed by~~ a padding ~~Length Indicator~~ "Length Indicator";
- ~~and~~ padding ~~may be transmitted~~.

In the case where a PDU contains a 15-bit ~~LI~~"Length Indicator" indicating that ~~an~~ a ~~SDU~~RLC SDU ends with one octet left in the PDU, the last octet of this PDU shall:

- be padded by the Sender and ignored by the Receiver though there is no "Length Indicator" indicating the existence of Padding ~~and~~;
- ~~shall~~ not be filled with the first octet of the next ~~SDU~~RLC SDU data.

In the case where 15-bit "Length Indicators" are used for the previous PDU and the last segment of ~~an~~ a RLC SDU is one octet short of exactly filling the PDU~~, and~~:

- if a 15-bit ~~Length Indicator~~ "Length Indicator" is used for the following PDU: ~~then~~
  - the ~~LI~~"Length Indicator" with value ~~LI~~="111 1111 1111 1011" shall be placed as the first "Length Indicator" in the following PDU~~.~~ ;
  - t~~T~~he remaining one octet in the previous PDU shall be padded by the Sender and ignored at the ~~receiver~~Receiver though there is no "Length Indicator" indicating the existence of Padding~~.~~ ;
  - i~~I~~n case this ~~SDU~~RLC SDU was the last one to be transmitted, a PDU consisting of ~~an~~ a RLC Header with ~~LI~~"Length Indicator" with value ="111 1111 1111 1011" followed by a padding ~~Length Indicator~~ "Length Indicator" and padding may be transmitted.
- if a 7-bit ~~Length Indicator~~ "Length Indicator" is used for the following PDU; ~~then~~
  - if RLC is configured for UM mode, the ~~LI~~"Length Indicator" with value ~~LI~~="000 0000" shall be placed as the first ~~Length indicator~~ "Length Indicator" in the following PDU and its SN shall be incremented by 2 before it is transmitted ~~(this can only occur in UM)~~.

Predefined values of the "Length Indicator" are used to indicate padding. The values that are reserved for special purposes are listed in the tables below depending on the size of the "Length Indicator". Only predefined "Length Indicator" values can refer to the padding space~~A PDU that has unused space, to be referred to as padding, shall use a Length Indicator to indicate that this space is used as padding unless the padding size is one octet for PDUs with 15-bit LIs~~. ~~A~~ These values~~padding Length Indicator~~ ~~"Length Indicator"must~~ shall only be placed after ~~any~~ all other "Length Indicators" for a PDU.

~~All unused space in a PDU must be located at the end of the PDU, be a homogeneous space and is referred to as padding. Predefined values of the Length Indicator are used to indicate this. The values that are reserved for special purposes are listed in the tables below depending on the size of the Length Indicator. Only predefined Length Indicator values can refer to the padding space.~~

STATUS PDUs can be piggybacked on the AMD PDU by using part or all of the padding space. A ~~Length Indicator~~ "Length Indicator" ~~must~~shall be used to indicate the piggybacked STATUS PDU. This ~~Length Indicator~~ "Length Indicator" takes space from the padding space or piggybacked STATUS PDU and not the PDU data and will always be the last "Length Indicator". Where only part of the padding space is used by a piggybacked STATUS PDU~~, then~~ the end of the piggybacked STATUS PDU is determined by one of the SUFI fields NO_MORE or ACK, thus no additional ~~Length Indicator~~ "Length Indicator" is required to show that there is still padding in the PDU. The padding/piggybacked STATUS PDU predefined "Length Indicators" shall be added after the very last ~~(i.e. there could be more than one SDU that end within a PDU) Length Indicator~~ "Length Indicator" that indicates the end of the last ~~SDU~~RLC SDU segment in the PDU.

If "SDU discard with explicit signalling" is ~~used~~ configured:

- an AMD PDU can contain a maximum number of 15 ~~LI~~"Length Indicators" indicating the end of ~~an~~ 15 corresponding SDUs ~~and~~;
- the rest of the AMD PDU space shall be used as padding~~/~~ or as piggybacked STATUS PDU.

~~For AM, 7bit indicators shall be used if the AMD PDU size is ≤ l26 octets. Otherwise 15bit indicators shall be used. For UM, 7bit indicators shall be used if the largest UMD PDU size is ≤ l25 octets. Otherwise 15bit indicators shall be used.~~

~~The length of the Length Indicator only depends on the size of the largest RLC PDU. Between RLC modifications the length of the Length Indicator is the same for all UMD PDUs. The length of the Length Indicator is always the same for all AMD PDUs, for one RLC entity.~~

Length: 7 bits

Bit	Description
0000000	The previous RLC PDU was exactly filled with the last segment of~~ an~~ a RLC SDU and there is no ~~LI~~"Length Indicator" that indicates the end of the ~~SDU~~RLC SDU in the previous RLC PDU.
1111100	UMD PDU: The first data octet in this RLC PDU is the first octet of~~ an~~ a RLC SDU. AMD PDU: Reserved (PDUs with this coding will be discarded by this version of the protocol).
1111101	Reserved (PDUs with this coding will be discarded by this version of the protocol).
1111110	AMD PDU: The rest of the RLC PDU includes a piggybacked STATUS PDU. UMD PDU: Reserved (PDUs with this coding will be discarded by this version of the protocol).
1111111	The rest of the RLC PDU is padding. The padding length can be zero.

Length: 15 bits

Bit	Description
000000000000000	The previous RLC PDU was exactly filled with the last segment of~~ an~~ a RLC SDU and there is no ~~LI~~"Length Indicator" that indicates the end of the ~~SDU~~RLC SDU in the previous RLC PDU.
111111111111011	The last segment of~~ an~~ a RLC SDU was one octet short of exactly filling the previous RLC PDU and there is no ~~LI~~"Length Indicator" that indicates the end of the ~~SDU~~RLC SDU in the previous RLC PDU. The remaining one octet in the previous RLC PDU is ignored.
111111111111100	UMD PDU: The first data octet in this RLC PDU is the first octet of~~ an~~ a RLC SDU. AMD PDU: Reserved (PDUs with this coding will be discarded by this version of the protocol).
111111111111101	Reserved (PDUs with this coding will be discarded by this version of the protocol).
111111111111110	AMD PDU: The rest of the RLC PDU includes a piggybacked STATUS PDU. UMD PDU: Reserved (PDUs with this coding will be discarded by this version of the protocol).
111111111111111	The rest of the RLC PDU is padding. The padding length can be zero.

9.2.2.9 Data field

RLC SDUs or segments of RLC SDUs are mapped to this field in transparent, unacknowledged and acknowledged modes.

Transparent mode data:

- ~~T~~the length of RLC SDUs is not constrained to a multiple of 8 bits,~~.~~
- ~~The RLC SDUs might be segmented.~~if "Segmentation" is configured:

~~The allowed size for the segments shall be determined from the transport formats of the transport channel [4, 8].~~

  - ~~A~~all the RLC PDUs carrying one RLC SDU shall be sent in one ~~transmission time interval~~TTI;~~.~~

- o~~O~~nly segments from one RLC SDU shall be sent in one ~~transmission time interval~~TTI.

- otherwise (~~if~~ "Segmentation~~"~~ is not configured):

  - TMD PDU size is fixed within a single TTI and is equal to the RLC SDU size.

~~NOTE: If segmentation is not used for the transparent mode RLC entity then more than one RLC SDU can be sent in one transmission time interval using one RLC PDU per RLC SDU. The RLC PDUs need, however, to be of the same size due to L1 limitations.~~

Unacknowledged mode data and Acknowledged mode data:

- t~~T~~he length of RLC SDUs is constrained to a multiple of 8 bits;~~.~~
- if a RLC SDU~~s~~ ~~might be~~is segmented, its~~. If possible, the~~ last segment ~~of an SDU~~ shall be concatenated with the first segment of the next ~~SDU~~RLC SDU in order to fill the data field completely and avoid unnecessary padding. The~~ length indicator~~ "Length Indicator" field is used to point the borders between ~~SDU~~RLC SDUs;~~.~~
- f~~F~~or PDUs with 15-bit ~~LI~~"Length Indicators", if~~ an~~ a ~~SDU~~RLC SDU ends with one octet left in a PDU whether the ~~LI~~"Length Indicator" indicating the end of the ~~SDU~~RLC SDU is contained in this PDU or in the next PDU, padding for the last octet of this PDU is necessary and the next ~~SDU~~RLC SDU shall not be concatenated in this PDU. No ~~LI~~"Length Indicator" shall be needed to indicate this kind of one-octet padding.

9.2.2.10 Padding (PAD)

All unused space in a PDU ~~must~~shall be located at the end of the PDUand is referred to as padding. Padding has a length such that the PDU has the required predefined total length.

Padding may have any value and the ~~receiving entity~~Receiver shall disregard it.

9.2.2.11 SUFI

Which SUFI fields to use is implementation dependent, but when a STATUS PDU includes information about which PDUs have been received and which are detected as missing, information shall not be included about PDUs with SN≥VR(H) i.e. PDUs that have not yet reached the ~~receiver~~Receiver. Information about PDUs with SN<VR(R) shall not be given except when this is necessary in order to use the BITMAP SUFI, see subclause 9.2.2.11.5.

Length: variable number of bits.

The SUFI ~~(Super-Field)~~ ~~may~~can includes three sub-fields: type information (type of super-field, e.g. list, bitmap, acknowledgement, etc), length information (providing the length of a variable length field within the following value field) and a value.

Figure 9.7 shows the structure of the super-field. The size of the type sub-field is non-zero but the size of the other sub-fields may be zero.

Type
Length
Value

Figure 9.7: The Structure of a Super-Field

The length of the type field is 4 bits and it may have any of following values.

Bit	Description
0000	No More Data (**NO_MORE**)
0001	Window Size (**WINDOW**)
0010	Acknowledgement (**ACK**)
0011	List (**LIST**)
0100	Bitmap (**BITMAP**)
0101	Relative list (**Rlist)**
0110	Move Receiving Window **(MRW)**
0111	Move Receiving Window Acknowledgement **(MRW_ACK)**
1000-1111	Reserved (PDUs with this encoding are invalid for this version of the protocol)

The length sub-field gives the length of the variable size part of the following value sub-field and the length of it depends on the super-field type. The value sub-field includes the value of the super-field, e.g. the bitmap in case of a BITMAP super-field, and the length is given by the length of the type sub-field.

9.2.2.11.1 The No More Data super-field

The 'No More Data' super-field indicates the end of the data part of a STATUS PDU and is shown in Figure 9.8 below. It shall always be placed as the last SUFI if it is included in a STATUS PDU. All data after this SUFI shall be regarded as padding and shall be neglected.

Type=**NO_MORE**

Figure 9.8: NO_MORE field in a STATUS PDU

9.2.2.11.2 The Acknowledgement super-field

The 'Acknowledgement' super-field consists of a type identifier field (ACK) and a sequence number (LSN) as shown in Figure 9.9 below. The acknowledgement super-field is also indicating the end of the data part of a STATUS PDU. Thus, no 'NO_MORE' super-field is needed in the STATUS PDU when the 'ACK' super-field is present. The ACK SUFI shall always be placed as the last SUFI if it is included in a STATUS PDU. All data after this SUFI shall be regarded as padding and shall be neglected.

Type = **ACK**
LSN

Figure 9.9: The ACK fields in a STATUS PDU

LSN

Length: 12 bits

Acknowledges the reception of all PDUs with "S~~s~~equence N~~n~~umber"~~s~~ < LSN (Last Sequence Number) that are *not* indicated to be erroneous in earlier parts of the STATUS PDU. This means that if the LSN is set to a value greater than VR(R), all erroneous PDUs ~~must~~shall be included in the same STATUS PDU and if the LSN is set to VR(R), the erroneous PDUs can be split into several STATUS PDUs. At the transmitter, if the value of the LSN =< the value of the first error indicated in the STATUS PDU, VT(A) will be updated according to the LSN, otherwise VT(A) will be updated according to the first error indicated in the STATUS PDU. VT(A) is only updated based on STATUS PDUs where ACK SUFI (or MRW_ACK SUFI) is included. The LSN ~~should~~ shall not be set to a value > VR(H) nor < VR(R).

9.2.2.11.3 The Window Size super-field

The ~~'~~Window Size~~'~~ super-field consists of a type identifier (WINDOW) and a window size number (WSN) as shown in Figure 9.10 below. The ~~receiver~~Receiver is always allowed to change the ~~Tx transmitter~~transmission window size of the peer entity during a connection, but the minimum and the maximum allowed value is given by upper layers configuration. The ~~Rx receiver~~reception window size of the ~~receiver~~Receiver is not changed.

Type = **WINDOW**
WSN

Figure 9.10: The WINDOW fields in a STATUS PDU

WSN

Length: 12 bits

The value of VT(WS) to be used by the transmitter. The range of the WSN is [0, 2^{12}-1]. The minimum value of VT(WS) is 1~~, if~~. If WSN is zero, the SUFI shall be discarded by this version of the protocol. The variable VT(WS) is set equal to WSN upon reception of this SUFI. If WSN is greater than Configured_Tx_Window_Size, VT(WS) shall be set equal to Configured_Tx_Window_Size.

9.2.2.11.4 The List super-field

The List Super-Field consists of a type identifier field (LIST), a list length field (LENGTH) and a list of LENGTH number of pairs as shown in Figure 9.11 below:

Type = **LIST**
LENGTH
SN_1
L_1
SN_2
L_2
…
SN_{LENGTH}
L_{LENGTH}

Figure 9.11: The List fields in a STATUS PDU ~~for a list~~

LENGTH

Length: 4 bits

The number of (SN_i , L_i)-pairs in the super-field of type LIST. The value "0000" is invalid and the STATUS PDU is discarded.

SN*i*

Length: 12 bits

"Sequence ~~number~~Number" of PDU, which was not correctly received.

L*i*

Length: 4 bits

Number of consecutive PDUs not correctly received following PDU with "S~~s~~equence ~~n~~Number" SN_i.

9.2.2.11.5 The Bitmap super-field

The Bitmap Super-Field consists of a type identifier field (BITMAP), a bitmap length field (LENGTH), a first sequence number (FSN) and a bitmap as shown in Figure 9.12 below:

Type = **BITMAP**
LENGTH
FSN
Bitmap

Figure 9.12: The Bitmap fields in a STATUS PDU

LENGTH

Length: 4 bits

The size of the bitmap in octets equals LENGTH+1, i.e. LENGTH="0000" means that the size of the bitmap is one octet and LENGTH="1111" gives the maximum bitmap size of 16 octets.

FSN

Length: 12 bits

The "S~~s~~equence ~~n~~Number" for the first bit in the bitmap. FSN shall not be set to a value lower than VR(R)-7 when the ~~Rx receiver~~reception window size is less than half the maximum RLC AM "S~~s~~equence ~~n~~Number". If the ~~Rx receiver~~reception window size is larger, FSN shall not be set to a value lower than VR(R).

Bitmap

Length: Variable number of octets given by the LENGTH field.

Status of the SNs in the interval [FSN, FSN + (LENGTH+1)*8 - 1] indicated in the bitmap where each position (from left to right) can have two different values (0 and 1) with the following meaning (bit_position$\in$[0,(LENGTH+1)*8 - 1]):

1: SN = (FSN + bit_position) has been correctly received.

0: SN = (FSN + bit_position) has not been correctly received.

9.2.2.11.6 The Relative List super-field

The Relative List super-field consists of a type identifier field (RLIST), a list length field (LENGTH), the first sequence number (FSN) and a list of LENGTH number of codewords (CW) as shown in Figure 9.13 below.

Type = **RLIST**
LENGTH
FSN
CW_1
CW_2
…
CW_{LENGTH}

Figure 9.13: The RList fields in a STATUS PDU

LENGTH

Length: 4 bits

The number of codewords (CW) in the super-field of type RLIST.

FSN

Length: 12 bits

The "S~~s~~equence ~~n~~Number" for the first erroneous PDU in the RLIST, i.e. LENGTH="0000" means that only FSN is present in the SUFI.

CW

Length: 4 bits

The CW consists of 4 bits where the three first bits are part of a number and the last bit is a status indicator and it shall be interpreted as follows:

Code Word	Description
$X_1X_2X_3$ **0**	Next 3 bits of the number are $X_1X_2X_3$ and the number continues in the next CW. The most significant bit within this CW is X_1.
$X_1X_2X_3$ **1**	Next 3 bits of the number are $X_1X_2X_3$ and the number is terminated. The most significant bit within this CW is X_1. This is the most significant CW within the number.

By default, the number given by the CWs represents a distance between the previous indicated erroneous PDU up to and including the next erroneous PDU.

One special value of CW is defined:

000 1 ’Error burst indicator’.

The error burst indicator means that the next CWs will represent the number of subsequent erroneous PDUs (not counting the already indicated error position). After the number of errors in a burst is terminated with XXX 1, the next codeword will again by default be the least significant bits (LSB) of the distance to the next error.

If the last CW, as indicated by the value of the LENGTH field, does not contain a "1" in its rightmost position, or the last CW, as indicated by the value of the LENGTH field does contain a "1" in its rightmost position, but is a special "error burst indicator" CW, the encoding of the RLIST SUFI is invalid, and the STATUS PDU is discarded.

9.2.2.11.7 The Move Receiving Window Acknowledgement super-field

The ’Move Receiving Window Acknowledgement’ super-field acknowledges the reception of a MRW SUFI. The format is given in Figure 9.14 below.

Type = **MRW_ACK**
N
SN_ACK

Figure 9.14: The MRW-ACK fields in a STATUS PDU

N

Length: 4 bits

The N field shall be set equal to the N_{LENGTH} field in the received MRW SUFI if the SN_ACK field is equal to the SN_MRW_{LENGTH} field. Otherwise N shall be set to 0.

With the aid of this field in combination with the SN_ACK field, it can be determined if the MRW_ACK corresponds to a previously transmitted MRW SUFI.

SN_ACK

Length: 12 bits

The SN_ACK field indicates the updated value of VR(R) after the reception of the MRW SUFI. With the aid of this field in combination with the N field, it can be determined if the MRW_ACK corresponds to a previously transmitted MRW SUFI.

9.2.2.11.8 The Move Receiving Window (MRW) super-field

The ’Move Receiving Window’ super-field is used to request the ~~RLC r~~Receiver to move its ~~receiving~~ reception window and optionally to indicate the set of discarded ~~SDU~~RLC SDUs, as a result of~~ an~~ a ~~SDU~~RLC SDU discard in the ~~RLC transmitter~~Sender. The format is given in Figure 9.15 below.

Type = **MRW**
LENGTH
SN_MRW$_1$
SN_MRW$_2$
...
SN_MRW$_{\mathrm{LENGTH}}$
N$_{\mathrm{LENGTH}}$

Figure 9.15: The MRW fields in a STATUS PDU

LENGTH

Length: 4 bits

The number of SN_MRW$_i$ fields in the super-field of type MRW.

The values "0001" through "1111" indicate 1 through 15 SN_MRW$_i$ respectively. The value "0000" indicates that one SN_MRW$_i$ field is present and that the ~~SDU~~RLC SDU to be discarded in the R~~r~~eceiver extends above the configured ~~Tx transmitting~~transmission window in the ~~transmitter~~Sender.

SN_MRW$_i$

Length: 12 bits

When Send MRW is configured,~~ an~~ a SN_MRW$_i$ shall be used to indicate the end of each discarded ~~SDU~~RLC SDU, i.e. the number of SN_MRW$_i$ fields shall equal the number of ~~SDU~~RLC SDUs discarded by that MRW SUFI. When "Send MRW" is not configured, SN_MRW$_i$ shall be used to indicate the end of the last ~~SDU~~RLC SDU to be discarded in the ~~receiver~~ Receiver and they may optionally be used to indicate the end of other discarded ~~SDU~~RLC SDUs. SN_MRW$_i$ is the "S~~s~~equence ~~n~~Number" of the PDU that contains the ~~LI~~"Length Indicator" of the i:th ~~SDU~~RLC SDU to be discarded in the ~~receiver~~ Receiver (except for SN_MRW$_{\mathrm{LENGTH}}$ when N$_{\mathrm{LENGTH}}$ = 0, see definition of N$_{\mathrm{LENGTH}}$). The order of the SN_MRW$_i$ shall be in the same sequential order as the ~~SDU~~RLC SDUs that they refer to.

Additionally SN_MRW$_{\mathrm{LENGTH}}$ requests the ~~RLC r~~Receiver to discard all PDUs with "S~~s~~equence N~~n~~umber" < SN_MRW$_{\mathrm{LENGTH}}$, and to move the ~~receiving~~ reception window accordingly. In addition, when N$_{\mathrm{LENGTH}}$ > 0, the R~~r~~eceiver has to discard the first N$_{\mathrm{LENGTH}}$ ~~LI~~"Length Indicators" and the corresponding data octets in the PDU with "S~~s~~equence N~~n~~umber" SN_MRW$_{\mathrm{LENGTH}}$.

N$_{\mathrm{LENGTH}}$

Length: 4 bits

N$_{\mathrm{LENGTH}}$ is used together with SN_MRW$_{\mathrm{LENGTH}}$ to indicate the end of the last ~~SDU~~RLC SDU to be discarded in the ~~receiver~~Receiver.

N$_{\mathrm{LENGTH}}$ indicates which ~~LI~~"Length Indicator" in the PDU with "S~~s~~equence N~~n~~umber" SN_MRW$_{\mathrm{LENGTH}}$ corresponds to the last ~~SDU~~RLC SDU to be discarded in the ~~receiver~~Receiver. N$_{\mathrm{LENGTH}}$ = 0 indicates that the last ~~SDU~~RLC SDU ended in the PDU with "S~~s~~equence N~~n~~umber" SN_MRW$_{\mathrm{LENGTH}}$ –1 and that the first data octet in the PDU with "S~~s~~equence N~~n~~umber" SN_MRW$_{\mathrm{LENGTH}}$ is the first data octet to be reassembled next.

9.2.2.12 Reserved 2 (R2)

Length: 1 bit

This bit in the Piggybacked STATUS PDU is used to ~~achieve octet alignment~~make the Piggybacked STATUS PDU a multiple of 8 bits in length and for this purpose it is coded as 0. Otherwise the PDU is treated as invalid and hence shall be discarded by this version of the protocol.

9.2.2.13 Reset Sequence Number (RSN)

Length: 1 bit

This field is used to indicate the sequence number of the transmitted RESET PDU. If this RESET PDU is a retransmission of the original RESET PDU then the retransmitted RESET PDU would have the same ~~sequence~~

~~number~~RSN value as the original RESET PDU. Otherwise it will have the next ~~reset sequence number~~RSN value. The initial value of this field is zero. The value of this field shall be reinitialised when the RLC is re-established. It shall not be reinitialised when the RLC is reset.

9.2.2.14 Hyper Frame Number Indicator (HFNI)

Length: 20 bit

This field is used to indicate the hyper frame number (HFN) to the peer entity. With the aid of this field the HFN in UE and UTRAN can be synchronised.

9.3 Protocol states

The content presented in this subclause is intended to support the definition of the RLC protocol states only, and is not meant to specify or constrain the implementation of the protocol.

9.3.1 State model for transparent mode entities

Figure 9.16 illustrates the state model for transparent mode RLC entities (both transmitting ~~sending~~ and receiving). A transparent mode entity can be in one of the following states.

9.3.1.1 ~~Null~~ NULL State

In the ~~null~~ NULL state, the RLC entity does not exist and therefore it is not possible to transfer any data through it.

Upon reception of a CRLC-CONFIG-Req from upper layers indicating establishment, the RLC entity ~~should~~:

- ~~be~~is created; and
- enters the ~~transparent data transfer ready~~DATA_TRANSFER_READY state ~~is entered~~.

9.3.1.2 ~~Transparent Data Transfer Ready~~DATA_TRANSFER_READY State

In the ~~transparent data~~ DATA_TRANSFER_READY~~transfer ready~~ state, transparent mode data can be exchanged between the entities according to subclause 11.1.

Upon reception of a CRLC-CONFIG-Req from upper layer indicating release, the RLC entity ~~should~~:

- enters the NULL state; and
- ~~be~~is considered as being terminated.~~; and~~
- ~~enter the null~~ NULL ~~state is entered.~~

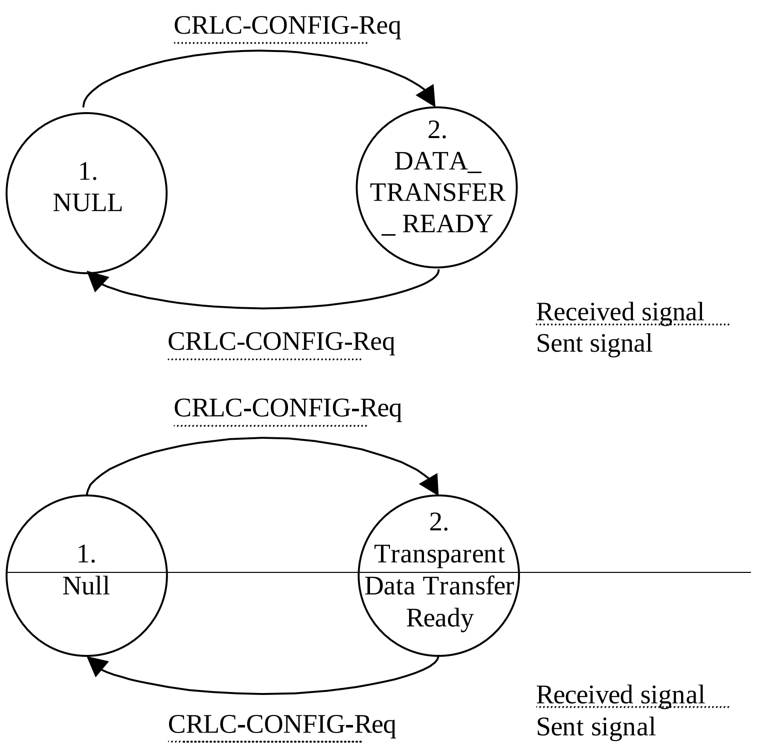


Figure 9.16: The state model for transparent mode entities

9.3.2 State model for unacknowledged mode entities

Figure 9.17 illustrates the state model for unacknowledged mode RLC entities (both transmitting ~~sending~~ and receiving). An unacknowledged mode entity can be in one of the following states.

9.3.2.1 ~~Null~~ NULL State

In the ~~null~~ NULL state, the RLC entity does not exist and therefore it is not possible to transfer any data through it.

Upon reception of a CRLC-CONFIG-Req from upper layer indicating establishment, the RLC entity ~~should~~:

- ~~be~~is created; and
- enters the ~~unacknowledged data transfer ready~~DATA_TRANSFER_READY state ~~is entered~~.

9.3.2.2 ~~Unacknowledged Data Transfer Ready~~DATA_TRANSFER_READY State

In the ~~unacknowledged data transfer ready~~DATA_TRANSFER_READY state, unacknowledged mode data can be exchanged between the entities according to subclause 11.2.

Upon reception of a CRLC-CONFIG-Req from upper layer indicating release, the RLC entity ~~should~~:

- enters the NULL state; and
- ~~be~~is considered as being terminated.~~; and~~

~~- enter and the null NULL state is entered.~~

Upon reception of a CRLC-CONFIG-Req from upper layer indicating modification, the RLC entity ~~should~~:

- stays in the DATA_TRANSFER_READY state;
- modif~~y~~ies only the protocol parameters and timers as indicated by upper layers.~~; and~~

~~- keep other protocol parameters, the protocol variables and the protocol timers unchanged.~~

Upon reception of a CRLC-SUSPEND-Req from upper layers, the RLC entity ~~should~~:

- enters the LOCAL_SUSPEND state.

9.3.2.3 ~~Local Suspend~~LOCAL_SUSPEND State

~~Upon reception of a CRLC-SUSPEND-Req from upper layers, the RLC entity is suspended and the Local Suspend state is entered.~~ In the ~~Local Suspend~~LOCAL_SUSPEND state, the RLC entity ~~shall~~ is suspended, i.e. it does not send ~~RLC-~~UMD PDUs with SN greater than or equal to certain specified value (see subclause 9.7.5).~~≥VT(US)+N.~~

Upon reception of a CRLC-RESUME-Req from upper layers, the RLC entity ~~should~~:

- ~~the RLC entity is resumed and~~enters the ~~Data Transfer Ready~~DATA_TRANSFER_READY state; and ~~is entered.~~
- resumes the data transmission ~~normally~~.

Upon reception of a CRLC-CONFIG-Req from upper layer indicating modification, the RLC entity ~~should~~:

- stays in the LOCAL_SUSPEND state;
- modif~~y~~ies only the protocol parameters and timers as indicated by upper layers.~~; and~~
- ~~keep other protocol parameters, the protocol variables and the protocol timers unchanged.~~

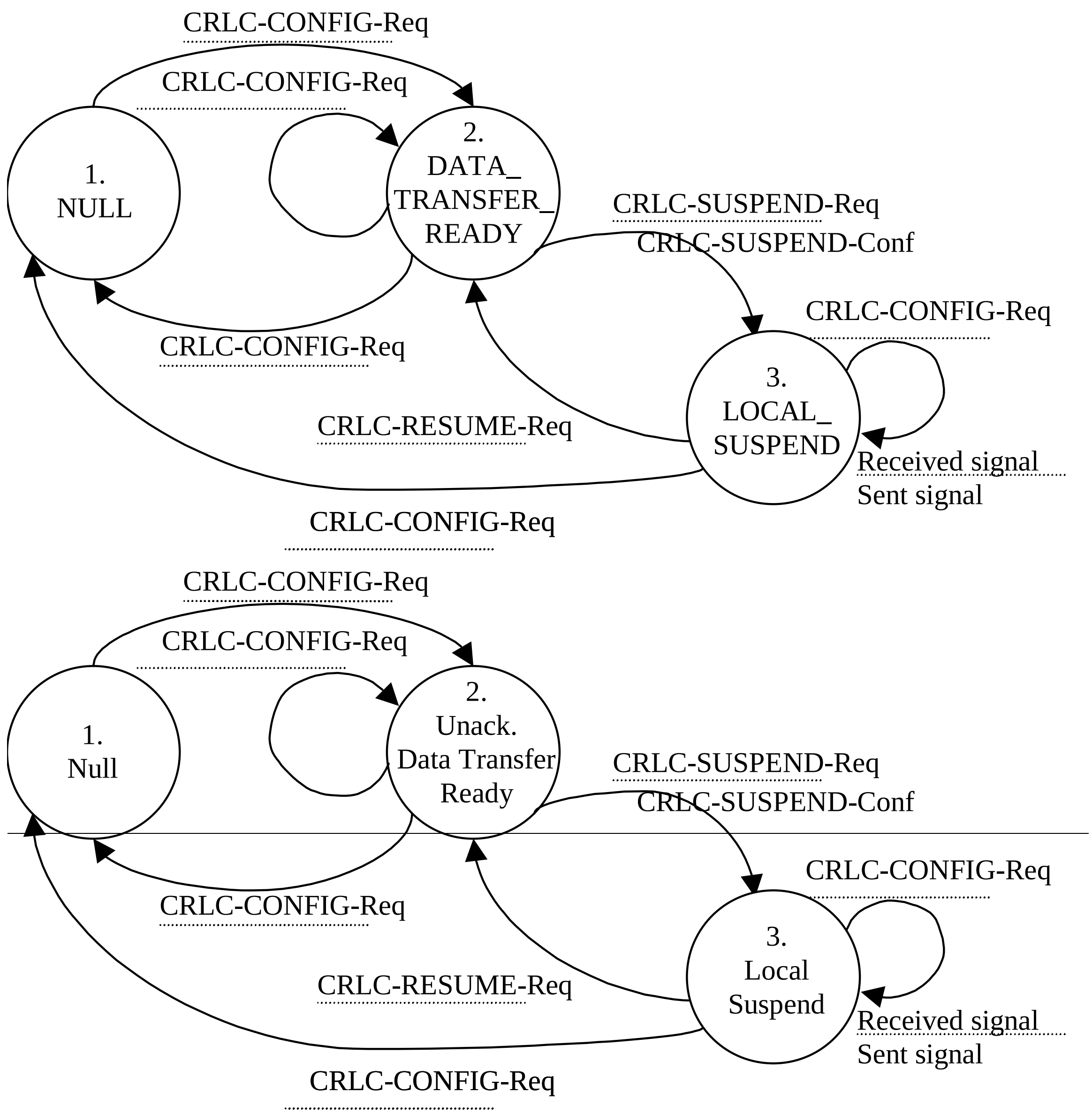


Figure 9.17: The state model for unacknowledged mode entities

9.3.3 State model for acknowledged mode entities

Figure 9.18 illustrates the state model for the acknowledged mode RLC entity (both transmitting and receiving). An acknowledged mode entity can be in one of the following states.

9.3.3.1 ~~Null~~ NULL State

In the ~~null~~ NULL state, the RLC entity does not exist and therefore it is not possible to transfer any data through it.

Upon reception of a CRLC-CONFIG-Req from upper layer indicating ~~(re)~~establishment, the RLC entity ~~should~~:

- ~~be~~is created; and
- enters the ~~acknowledged data transfer ready~~ DATA_TRANSFER_READY state ~~is entered~~.

9.3.3.2 ~~Acknowledged Data Transfer Ready~~DATA_TRANSFER_READY State

In the ~~acknowledged data transfer ready~~DATA_TRANSFER_READY state, acknowledged mode data can be exchanged between the entities according to subclause 11.3.

Upon reception of a CRLC-CONFIG-Req from upper layer indicating release, the RLC entity ~~should~~:

- enters the NULL state; and

- ~~be~~is considered as being terminated.~~; and~~

~~- enter the null NULL state is entered.~~

Upon detection of an initiating condition for the RLC reset procedure described in subclause 11.4.2~~errors in the protocol~~, the RLC entity ~~should~~:

- ~~sends a RESET PDU to its peer~~initiates the RLC reset procedure (see subclause 11.4); and

- enters the ~~reset pending~~RESET_PENDING state.

Upon reception of a RESET PDU, the RLC entity ~~resets the protocol (see~~ responds according to subclause 11.4.3.~~), sets the hyper frame number HFN (DL HFN when the RESET is received in UE or UL HFN when the RESET is received in UTRAN) equal to the HFNI field in the RESET PDU; and responds to the peer entity with a RESET ACK PDU.~~

Upon reception of a RESET ACK PDU, the RLC entity takes no action.

Upon reception of CRLC-SUSPEND-Req from upper layer, the RLC entity ~~should~~is suspended and enters the ~~local suspend~~LOCAL_SUSPEND state ~~is entered~~.

9.3.3.3 ~~*Reset* Pending~~RESET_PENDING State

In the ~~reset pending~~ RESET_PENDING state, the entity waits for a response from its peer entity and no data can be exchanged between the entities.

Upon reception of a CRLC-CONFIG-Req from upper layer indicating release, the RLC entity ~~should~~:

- enters the NULL state; and

- ~~be~~is considered as being terminated.

~~; and~~

~~- enter the nullNULL state is entered~~.

Upon reception of a RESET ACK PDU with the same RSN value as in the corresponding RESET PDU, the RLC entity ~~should~~:

- ~~respond~~acts ~~resets the protocol (see~~according to subclause 11.4.4; and~~), sets the hyper frame number HFN (DL HFN when the RESET ACK is received in UE or UL HFN when the RESET ACK is received in UTRAN) equal to the HFNI field in the RESET ACK PDU and~~

- enters the ~~acknowledged data transfer ready~~DATA_TRANSFER_READY state.

Upon reception of a RESET ACK PDU with a different RSN value as in the corresponding RESET PDU, the RLC entity ~~should~~:

- discards the RESET ACK PDU (see subclause 11.4.4); and

- stays in the RESET_PENDING state. ~~is discarded.~~

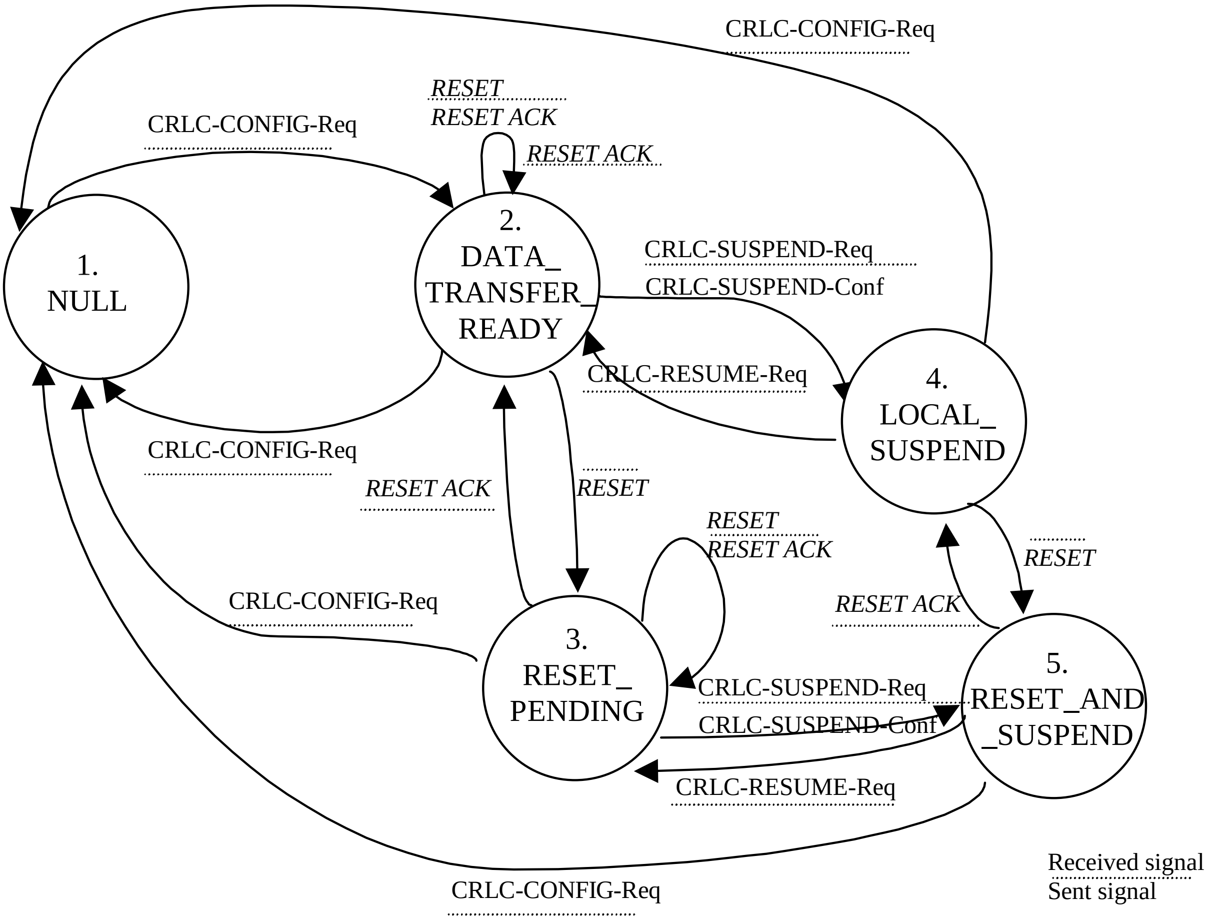
Upon reception of a RESET PDU, the RLC entity ~~should~~:

- responds according to~~resets the protocol (see~~ subclause 11.4.3; and~~), sets the hyper frame number HFN (DL HFN when the RESET is received in UE or UL HFN when the RESET is received in UTRAN) equal to the HFNI field in the RESET PDU, sends a RESET ACK PDU and~~

- stays in the ~~reset pending~~RESET_PENDING state.

Upon reception of CRLC-SUSPEND-Req from ~~the upper layer~~upper layer, the RLC entity ~~should~~:~~is suspended and~~

- enters the ~~reset and suspend~~RESET_AND_SUSPEND state ~~is entered~~.


CRLC-CONFIG-Req
RESET
RESET ACK
CRLC-CONFIG-Req
RESET ACK
2.
DATA_
TRANSFER_
READY
1.
NULL
CRLC-SUSPEND-Req
CRLC-SUSPEND-Conf
CRLC-RESUME-Req
4.
LOCAL_
SUSPEND
CRLC-CONFIG-Req
RESET ACK
RESET
RESET
RESET ACK
RESET
RESET ACK
CRLC-CONFIG-Req
3.
RESET_
PENDING
5.
RESET_AND
_SUSPEND
CRLC-SUSPEND-Req
CRLC-SUSPEND-Conf
CRLC-RESUME-Req
CRLC-CONFIG-Req
Received signal
Sent signal

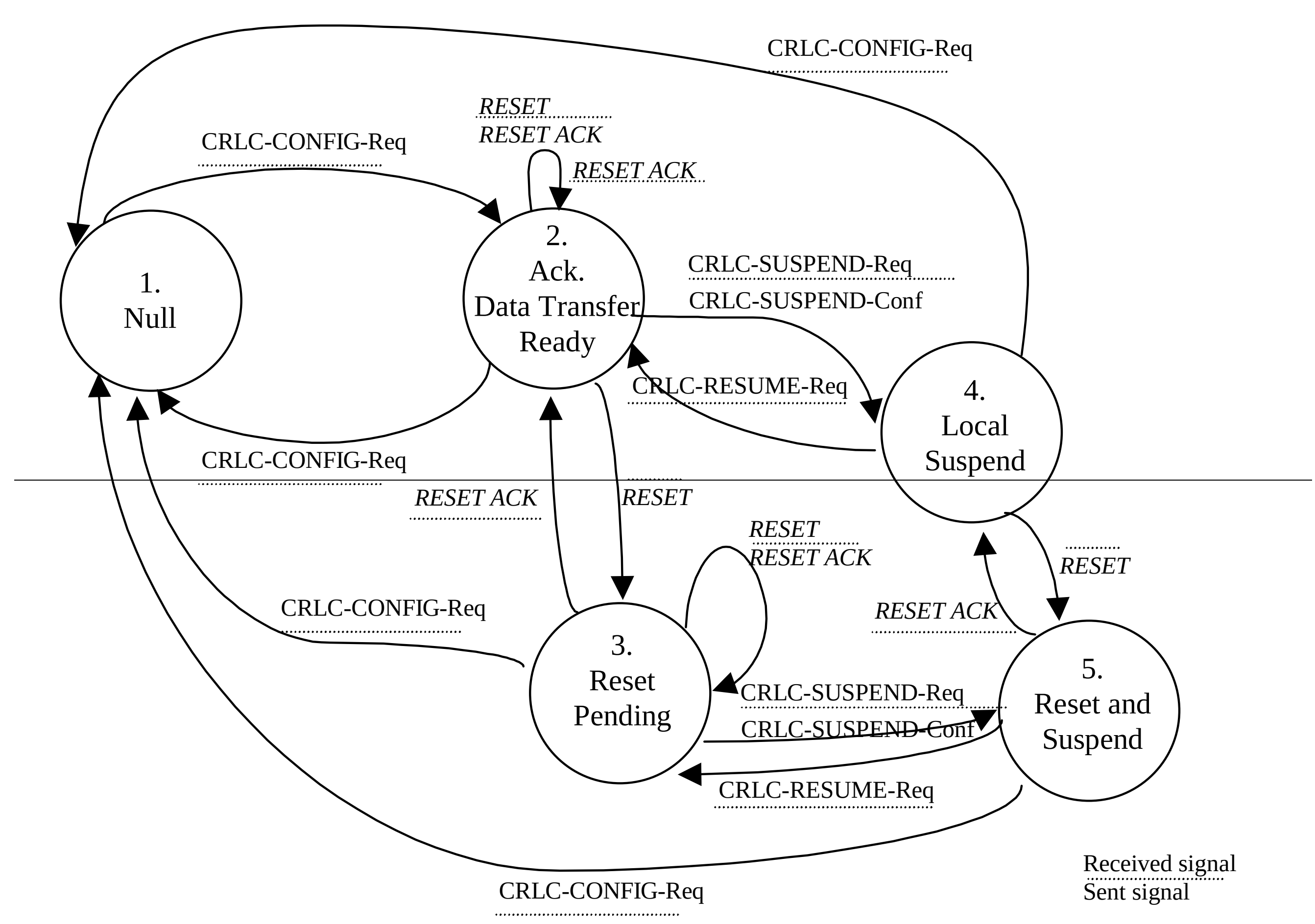


Figure 9.18: The state model for the acknowledged mode entities

9.3.3.4 ~~Local Suspend~~LOCAL_SUSPEND State

In the ~~Local Suspend~~LOCAL_SUSPEND state, the RLC entity is suspended, i.e. it does~~ shall~~ not send ~~an RLC-~~AMD PDUs with SN greater than or equal to certain specified value (see subclause 9.7.5).~~≥VT(S)+N, where VT(S) is the value of the send state variable when the CRLC-SUSPEND-Req with parameter N was received.~~

Upon reception of CRLC-RESUME-Req from upper layers in this state, the RLC entity~~ should~~:

- ~~is ~~resumes the data transmission~~ normally~~; and
- enters the ~~Acknowledged Data Transfer Ready~~DATA_TRANSFER_READY state~~ is entered~~.

Upon reception of CRLC-CONFIG-Req from upper layers indicating release, the RLC entity~~ should~~:

- enters the NULL state; and
- ~~be~~is considered as being terminated.

~~; and~~

~~- enter the null NULL state is entered.~~

Upon detection of an initiating condition for RLC reset procedure described in subclause 11.4.2~~errors in the protocol~~, the RLC entity~~ should~~:

- ~~sends a RESET PDU to its peer~~initiates the RLC reset procedure (see subclause 11.4); and

- enters the ~~reset and suspend~~ RESET_AND_SUSPEND state.

9.3.3.5 ~~Reset and Suspend~~RESET_AND_SUSPEND State

In the ~~reset and suspend~~RESET_AND_SUSPEND state, the entity waits for a response from its peer entity or a primitive (CRLC-RESUME-Req) from its upper layer and no data can be exchanged between the entities.

Upon reception of CRLC-CONFIG-Req from upper layer indicating release, the RLC entity~~ should~~:

- enters the NULL state; and

- ~~be~~is considered as being terminated.~~; and~~

~~- enter the null NULL state is entered~~.

Upon reception of a RESET ACK PDU with the same RSN value as in the corresponding RESET PDU, the RLC entity ~~should~~:

- ~~respond~~acts according to ~~resets the protocol (see~~ subclause 11.4.4; and~~), sets the hyper frame number HFN (DL HFN when the RESET ACK is received in UE or UL HFN when the RESET ACK is received in UTRAN) equal to the HFNI field in the RESET ACK and~~

- enters the ~~local suspend~~LOCAL_SUSPEND state.

Upon reception of CRLC-RESUME-Req from upper layer in this state, the RLC entity~~ should~~:

- ~~be~~is resumed, i.e. releases the suspend constraint; and

- enters the ~~reset pending~~RESET_PENDING state~~ is entered~~.

9.4 State variables

The state variables defined in this subclause are normative.

This sub-clause describes the state variables used in AM and UM in order to ~~the~~ specify~~ication of~~ the peer-to-peer protocol. All state variables are non-negative integers. ~~PDUs are sequentially and independently numbered and may have the value 0 through n minus 1 (where n is the modulus of the sequence numbers). The modulus equals 212 for AM and 27 for UM; the s~~UMD and AMD PDUs are numbered by modulo integer sequence numbers (SN), cycling~~e~~ through the ~~entire range~~field: 0 ~~through~~to $2^{12} – 1$ for AM and 0 ~~through~~ to $2^7 – 1$ for UM. All arithmetic operations contained in this specification on VT(S), VT(A), VT(MS), VR(R), VR(H) and VR(MR)~~the following state variables and sequence numbers contained in this specification~~ are affected by the AM modulus. All arithmetic operations contained in this specification on ~~: VT(S), VT(A), VT(MS), VR(R), VR(H), VR(MR),~~ VT(US) and VR(US) are affected by the UM modulus. When performing arithmetic comparisons of state variables or SN values a modulus base shall be used. This modulus base is subtracted (within the appropriate field) from all the values involved and then an absolute comparison is performed. ~~a~~At the ~~s~~Sender, VT(A) and VT(US) ~~are~~ ~~shallwill~~shall be assumed to be the modulus base in AM and UM respectively. ~~When performing~~At the Receiver, arithmetic comparisons of variables or SN values~~ at the r~~Receiver, VR(R) and VR(US) ~~are~~ ~~shallwill~~shall be assumed to be the modulus base in AM and UM respectively.

The RLC ~~shallwill~~shall maintain~~s~~ the following state variables ~~at~~ in the ~~transmitter~~~~transmitting side~~Sender.

a) VT(S) - Send state variable.

~~The sequence number~~This state variable contains the SN of the next AMD PDU to be transmitted for the first time (i.e. excluding retransmitted PDUs~~ssion~~). It ~~is~~ ~~shallwill~~shall be updated after ~~transmission~~ the aforementioned PDU is ~~of a PDU~~, ~~which includes not earlier~~ transmitted ~~PDUs and~~ or after transmission of a MRW SUFI which includes SN_MRW_{LENGTH} >VT(S) (see subclause 11.6).

The initial value of this variable is 0.

b) VT(A) - Acknowledge state variable.

~~The sequence number~~This state variable contains the SN following ~~of~~ the SN of the ~~next~~last in-sequence acknowledged PDU ~~expected to be acknowledged, which~~ This forms the lower edge of the window of acceptable acknowledgements. VT(A) ~~is~~ shall~~will~~shall be updated based on the receipt of a STATUS PDU including an ACK (see subclause 9.2.2.11.2) and/or a MRW_ACK ~~super-field~~SUFI (see subclause 11.6).

The initial value of this variable is 0. For the purpose of initialising the protocol, this value shall be assumed to be the first SN following the last in-sequence acknowledged PDU.

c) VT(DAT).

This state variable counts the number of times a PDU has been transmitted. There ~~is will~~shall be one VT(DAT) for each PDU and ~~it is~~each ~~will~~shall be incremented ~~each~~ every time the corresponding PDU is transmitted.

The initial value of this variable is 0.

d) VT(MS) - Maximum Send state variable.

This state variable contains the ~~sequence number~~SN of the first PDU ~~that may be rejected by thenot allowed~~that ~~constrained~~can be rejected by~~ not to be sent by the transmitter window set by by~~ the peer R~~r~~eceiver~~ [i.e. the receiver will allow up to VT(MS) – 1]~~, VT(MS) = VT(A) + VT(WS). This value represents the upper edge of the transmit~~ter~~ window. The transmitter ~~shall will~~shall not transmit ~~a~~ PDUs with SN ≥ VT(MS) unless VT(S) ≥ VT(MS). ~~VT(MS) is updated when either VT(A) or VT(WS) is updated. T,~~In that case, the PDU with SN = VT(S) - 1 can also be transmitted~~ also when VT(S) ≥ VT(MS)~~. VT(MS) ~~shallwill~~shall be updated when VT(A) or VT(WS) is updated.

The initial value of this variable is Configured_Tx_Window_size.

e) VT(US) – UM data state variable.

This state variable ~~gives~~ contains the ~~sequence number~~SN of the next UMD PDU to be transmitted. It ~~is~~ shall be ~~updated~~ incremented by 1 each time a UMD PDU is transmitted.

The initial value of this variable is 0.

f) VT(PDU).

This state variable is used when the “poll every Poll_PDU PDU” polling trigger is configured ~~function is used~~. It ~~is~~ shall~~will~~shall be incremented ~~with~~ by 1 for each PDU that is transmitted~~. It should be incremented for~~, including both new and retransmitted PDUs. When it ~~reache~~becomes equal~~s~~ to the value Poll_PDU, a new poll ~~is will~~shall be transmitted and the state variable ~~is will~~shall be set to zero.

The initial value of this variable is 0.

g) VT(SDU).

This state variable is used when the “poll every Poll_SDU SDU~~-~~” polling trigger is configured~~function is used~~. It ~~is~~ shall~~will~~shall be incremented ~~with~~ by 1 for ~~each~~ a given SDU ~~that is~~when all the PDUs carrying a part of this SDU have been transmitted at least once. When it ~~reache~~becomes equal~~s~~ to the value Poll_SDU a new poll ~~is will~~shall be transmitted and the state variable ~~is will~~shall be set to zero. The ~~p~~”Polling bit” ~~should will~~shall be set to “1” in the first transmission of the PDU that contains the last segment of the SDU.

The initial value of this variable is 0.

h) VT(RST) - Reset state variable.

~~It~~ This state variable is used to count the number of times a RESET PDU is transmitted before the reset procedure is completed. VT(RST) ~~is will~~shall be incremented ~~with~~ by 1 each time a RESET PDU is transmitted. VT(RST) ~~is will~~shall only be reset ~~only~~ upon the reception of a RESET ACK PDU, i.e. VT(RST) ~~is will~~shall not be reset when a~~n~~ RLC reset ~~occurs which was~~ initiated ~~from~~ by the peer RLC entity occurs.

The initial value of this variable is 0.

i) VT(MRW) – MRW command send state variable.

~~It~~ This state variable is used to count the number of times a MRW command is transmitted. VT(MRW) is incremented ~~with~~ by 1 each time a~~n~~ MRW ~~command~~ SUFI is transmitted. VT(MRW) ~~is will~~shall be reset when the SDU discard with explicit signalling procedure is terminated.

The initial value of this variable is 0.

j) VT(WS) – Transmitter window size state variable.

This state variable contains the size that ~~shall will~~shall be used for the transmitter window. VT(WS) ~~is~~ shall be set equal to the WSN field when the transmitter receives a STATUS PDU including a ~~Window Size~~WINDOW ~~super-field~~SUFI.

The initial value of this variable is Configured_Tx_Window_size.

The RLC ~~shallwill~~shall maintain~~s~~ the following state variables ~~at~~ in the ~~receiverreceiving side~~Receiver:

a) VR(R) - Receive state variable.

This state variable contains the ~~sequence number~~SN following that of the last ~~of the next~~ in-sequence PDU ~~expected to be~~ received. It ~~is will~~shall be ~~set equal to SNmax+1~~ updated upon the receipt of the ~~next in-sequence~~ PDU with SN equal to VR(R)~~, where SNmax is the sequence number of the highest received in-sequence PDU~~.

The initial value of this variable is 0. For the purpose of initialising the protocol, this value shall be assumed to be the first SN following the last in-sequence received PDU.

b) VR(H) - Highest expected state variable.

This state variable contains the ~~sequence number~~SN following ~~of~~ the highest ~~expected~~ SN of any received PDU. When a PDU is received with SN x such that VR(H)≤x<VR(MR), ~~T~~this state variable ~~is shallwill~~shall be set equal to x~~SN~~+1 ~~only when a new PDU is received with VR(MR)>SN≥VR(H)~~.

The initial value of this variable is 0.

c) VR(MR) - Maximum acceptable Receive state variable.

This state variable contains the ~~sequence number~~SN of the first PDU ~~not allowed~~that ~~shallwill~~shall be rejected by the ~~receiver~~Receiver ~~[i.e. the receiver will allow up to VR(MR) – 1]~~, VR(MR) = VR(R) + Configured_Rx_Window_Size. ~~The receiver shall discard PDUs with SN ≥ VR(MR).~~

d) VR(US) - Receiver Send Sequence state variable.

This state variable contains the ~~sequence number~~SN following that of the ~~next~~last PDU ~~to be~~ received. When a PDU with SN equal to x is received, ~~It~~the state variable shall set equal to ~~SN~~ x + 1 ~~upon reception of a PDU~~.

The initial value of this variable is 0.

e) VR(EP) - Estimated PDU Counter state variable.

This state variable contains the number of PDUs that ~~should yet be received yet~~whose re-transmission is still expected as a consequence of the transmission of the latest status report. ~~In acknowledged mode, this state variable is updated at the end of each transmission time interval. I~~At the end of each ~~transmission time interval~~TTI it is decremented by the total number of PDUs that ~~should have been~~were received during that time~~e transmission time interval~~. ~~If VR(EP) is equal to zero, then check if all PDUs requested for retransmission in the latest status report have been received.~~

9.5 Timers

The timers defined in this subclause are normative. The ~~T~~timers ~~will~~shall be considered active from the time they are started until the time they either expire or are stopped.

a) Timer_Poll.

This timer ~~is will~~shall only be used when ~~the poll timer trigger is used~~so configured by ~~higher layersthe upper layer~~upper layers. The ~~initial value~~value of the timer is signalled by ~~higher layersthe upper layer~~upper layers. In the UE ~~It is~~ this timer ~~will~~shall be started when the successful or unsuccessful transmission of a PDU containing a poll is indicated by ~~lower layers~~lower layer ~~(in UE)~~. In UTRAN ~~Or~~it ~~willshall~~should be started when a PDU containing a poll is submitted to ~~lower layers~~lower layer~~lower layer (in UTRAN)~~. If x is the value of the state variable VT(S) at the time the poll was submitted to ~~lower layers~~lower layer, ~~T~~the timer ~~is will~~shall be ~~stoppedreset~~stopped ~~when~~upon receiving ~~a STATUS PDU that~~ :

- ~~contains an~~ acknowledgements ~~(positive or negative) of~~for all the AMD PDUs with SN up to and including ~~VT(S)~~x-1; or

- ~~at the time the poll was submitted to lower layer, or when~~ a negative acknowledgement ~~of~~ for the ~~same~~ PDU with SN = x-1~~carrying the polling bit~~. ~~is received.~~

~~The value of the timer is signalled by upper layers.~~

If the timer expires and no STATUS PDU fulfilling the criteria above has been received: ~~or if a new poll is triggered while the timer is active,~~

- the ~~receiver~~Receiver ~~is will~~shall be polled once more; ~~(either by the transmission of a PDU which was not yet sent, or by a retransmission), and~~

- the timer ~~is will~~shall be restarted; ~~at the time specified above, with a~~ and

- the new value of VT(S)~~-1~~ ~~will~~shall be saved.

If a new poll is sent when the timer is ~~running~~ active the timer ~~is~~ shall be restarted at the time specified above, ~~with~~ and ~~a~~the ~~new~~ value of VT(S)~~-1~~ shall be saved.

b) Timer_Poll_Prohibit.

This timer ~~is will~~shall only be used when so configured by ~~higher layersthe upper layer~~upper layers. ~~the poll prohibit function is used~~. It is used to prohibit transmission of polls within a certain period. The ~~initial value~~value of the timer is signalled by ~~higher layersthe upper layer~~upper layers. ~~The~~

In the UE ~~timer~~this timer shall be started when the successful or unsuccessful transmission of a PDU containing a poll is indicated by ~~lower layers~~lower layer~~lower layer. (in UE) or~~ In UTRAN it ~~willshall~~should be started when a PDU containing a poll is submitted to ~~lower layers~~lower layer~~lower layer (in UTRAN)~~. ~~The prohibit time is calculated from the time a PDU containing a poll is submitted to lower layer until the timer has expired. If anotherA~~

From the time a poll is triggered ~~whilst the timer is active~~until the timer expires ~~before the timer expires~~, polling is prohibited. If another poll is triggered while polling is prohibited, its transmission shall be delayed until the timer ~~prohibit time~~ expires (see subclause 9.7.1) ~~if a poll is triggered during the prohibit time~~. Only one poll shall be transmitted when Timer_Poll_Prohibit ~~the prohibit time~~ expires even if several polls were triggered in the meantime. ~~during the prohibit time.~~ This timer ~~will~~shall not be ~~stopped~~ affected by ~~a~~ the reception~~ived~~ of STATUS PDUs. ~~The value of the timer is signalled by upper layers.~~

When Timer_Poll_Prohibit is not configured by ~~the upper layer~~upper layers, polling is never prohibited.

c) Timer_EPC.

This timer ~~should will~~shall only be used when the EPC function is ~~used~~ configured by ~~higher layersthe upper layer~~upper layers. ~~and i~~It is meant to account~~s~~ for the roundtrip delay, i.e. the time between the transmission of a status report and the reception of ~~when~~ the first retransmitted PDU. ~~should be received after a status report has been sent.~~ The ~~initial value~~value of the timer is signalled by ~~higher layersthe upper layer~~upper layers.

~~The timer~~ In the UE ~~is~~this timer ~~will~~shall be started when the successful or unsuccessful transmission of the first STATUS PDU of a status report is indicated by ~~lower layers~~lower layer~~lower layer (in UE)~~. In UTRAN it ~~willshall~~should be started when ~~or~~ the first STATUS PDU of a status report is submitted to ~~lower layers~~lower layer~~lower layer (in UTRAN)~~. ~~and w~~Only after Timer_EPC ~~hen it~~ expires ~~will~~shall VR(EP) be decremented as described in clause 9.7.4.~~can start its counting-down process (see subclause 9.7.4). The value of the timer is signalled by upper layers.~~

d) Timer_Discard.

This timer ~~is will~~shall be used ~~for the~~when timer based SDU discard is configured by ~~higher layersthe upper layer~~upper layers~~function~~. The ~~initial value~~value of the timer is signalled by ~~higher layersthe upper layer~~upper layers. In the transmitter, ~~the~~ a new timer is ~~activated~~ started upon reception of a~~n~~ SDU from ~~upper higher layersthe upper layer~~upper layers. ~~One timer is used for each SDU that is received from upper layer. For~~

In UM/TM~~r~~, if ~~the~~ a timer expires before the corresponding SDU is submitted to ~~a lower layers~~slower layer, "SDU discard without explicit signalling" specified in subclauses 11.2.4.3 and ~~/~~11.1.4.2 shall be ~~started~~initiated. ~~For~~ In AM, if ~~the~~ a timer expires before the corresponding SDU is acknowledged, "SDU discard with explicit signalling" specified in subclause 11.6 shall be ~~started~~initiated.

e) Timer_Poll_Periodic.

This timer ~~is will~~shall only be used when ~~the~~ “timer based polling” is ~~used~~configured by ~~higher layersthe upper layer~~upper layers. The ~~initial value~~value of the timer is signalled by ~~higher layersthe upper layer~~upper layers. The timer ~~is will~~shall be started when the RLC entity is created. ~~Each~~ When~~time~~ the timer expires, the RLC entity ~~will~~shall:

- ~~-~~ restart the timer~~; is restarted and a poll is triggered (either by the transmission of a PDU which was not yet sent, or by a retransmission)~~.

- ~~I~~if ~~there is no~~ PDUs are available for ~~to be~~ transmission~~tted~~ or retransmission (not yet acknowledged):

  - trigger a poll.~~and all PDUs have already been acknowledged, a poll shall not be triggered and the timer shall only be restarted. The value of the timer is signalled by upper layers.~~

f) Timer_Status_Prohibit.

This timer ~~is will~~shall only be used when ~~the STATUS prohibit function is~~so ~~used~~configured by ~~higher layersthe upper layer~~upper layers. It is meant to prohibit~~s~~ the ~~receiving side~~Receiver from sending consecutive acknowledgment status reports. A status report is an acknowledgement status report if it contains~~ing~~ .any of the SUFIs LIST, BITMAP, RLIST or ACK. The ~~initial value~~value of the timer is signalled by ~~higher layersthe upper layer~~upper layers.

In the UE ~~T~~thise timer ~~is will~~shall be started when the successful or unsuccessful transmission of the last STATUS PDU ~~in~~ of an acknowledgment -status report is indicated by ~~lower layers~~slower layer~~lower layer. (in UE) or~~ In UTRAN it ~~willshall~~should be started when the last STATUS PDU ~~in~~ of an acknowledgment status report is submitted to ~~lower layers~~slower layer~~lower layer (in UTRAN)~~. ~~The prohibit time is calculated f~~

From the time an ~~If another~~ acknowledgment status report ~~containing the above-mentioned SUFIs~~ is triggered until ~~before rom the time the last STATUS PDU of a status report is submitted to lower layer until~~ the Timer_Status_Prohibit timer ~~has~~ expires~~d~~, acknowledgment is prohibited. If another such status report is triggered while acknowledgment is prohibited, its transmission ~~will~~shall be delayed until the timer expires (see subclause 9.7.2). The status report may be updated during this time. ~~and no new status report containing the mentioned SUFIs can be transmitted during the prohibit time.~~ The ~~timer does not prohibit~~ transmission of ~~the~~ SUFIs MRW, MRW_ACK, WINDOW or NO_MORE is not restricted. ~~The value of the timer is signalled by upper layers.~~

When Timer_Status_Prohibit is not configured by ~~the upper layer~~upper layers, acknowledgment is never prohibited.

g) Timer_Status_Periodic.

This timer ~~is will~~shall only be used when timer based status reporting ~~sending~~ is ~~used~~configured by ~~higher layersthe upper layer~~upper layers.

Thise timer ~~is will~~shall be started when the RLC entity is created. ~~Each time~~When the timer expires the transmission of a status report ~~is will~~shall be triggered and the timer ~~is will~~shall be restarted. ~~The value of the timer is signalled by upper layers.~~ This timer can be blocked by ~~upper higher layersthe upper layer~~upper layers. ~~In this case, the timer shall not be active.~~ The timer shall be ~~reset and~~ restarted when ~~it ishigher layersthe upper layer~~upper layers indicates that it is no longer ~~un~~blocked~~ by upper layers~~.

h) Timer_RST.

This timer is ~~used~~ meant to ~~detect~~handle the loss of a RESET PDU by the peer entity, or the loss of a RESET ACK PDU from the peer ~~RLC~~ entity. The ~~initial value~~value of the timer is signalled by ~~higher layersthe upper layer~~upper layers. ~~This~~

In the UE this timer ~~is will~~shall be started when the successful or unsuccessful transmission of a RESET PDU is indicated by ~~lower layers~~slower layer~~lower layer.~~ ~~(in UE) or~~ In UTRAN it ~~willshall~~should be started when a RESET PDU is submitted to ~~lower layers~~slower layer ~~(in UTRAN)~~.

Timer_RST ~~It will~~shall only be stopped upon reception of a RESET ACK PDU (with same RSN as RESET PDU), i.e. this timer ~~is will~~shall not be stopped when a~~n~~ RLC reset ~~occurs which was~~ initiated ~~from~~ by the peer RLC entity occurs. If ~~it~~ this timer expires, the RESET PDU ~~will~~shall be retransmitted. ~~The value of the timer is signalled by upper layers.~~

i) Timer_MRW.

~~This timer is used as part of the Move Receiving Window protocol. It~~This timer is used to trigger the retransmission of a status report containing a~~n~~ MRW SUFI field. The ~~initial value~~value of the timer is signalled by ~~higher layersthe upper layer~~upper layers.

In the UE ~~The~~this timer ~~is will~~shall be started when the successful or unsuccessful transmission of a STATUS PDU containing the MRW SUFI is indicated by ~~lower layers~~slower layer.~~ (in UE) or~~ In UTRAN, it ~~willshall~~should be started when a STATUS PDU containing the MRW SUFI is submitted to ~~lower layers~~slower layer ~~(in UTRAN)~~.

Each time the timer expires the MRW SUFI is retransmitted and the timer is restarted~~ (at the time specified above)~~. It ~~shall will~~shall be ~~stopped reset~~stopped when one of the termination criteria for the SDU discard with explicit signalling procedure is fulfilled (see subclause 11.6.4). ~~The value of the timer is signalled by upper layers.~~

9.6 Protocol Parameters

The behavior defined in this subclause is normative. The values of the protocol parameters defined in this subclause are signalled by ~~upper higher layersthe upper layer~~upper layers.

a) MaxDAT.

~~It~~ This protocol parameter indicates~~is~~ the maximum value for the number of retransmissions of a PDU. ~~This parameter is an~~ It represents the upper limit ~~of counter~~for state variable VT(DAT). When ~~the value of~~ VT(DAT) ~~comes toreaches~~equals the value MaxDAT, either RLC RESET procedure or SDU discard procedure ~~shall will~~shall be initiated according to the configuration by ~~upper higher layersthe upper layer~~upper layers.

b) Poll_PDU.

This protocol parameter indicates how often the transmitter ~~should will~~shall poll the ~~receiver~~Receiver in the case ~~of~~ where “polling every Poll_PDU PDU” is configured by ~~higher layersthe upper layer~~upper layers. ~~This is an~~It represents -the upper limit for ~~the~~ state variable VT(PDU).~~ state variable,~~ ~~w~~When VT(PDU) ~~reaches~~equals the value Poll_PDU a poll ~~is will~~shall be transmitted to the peer entity.

c) Poll_SDU.

This protocol parameter indicates how often the transmitter ~~will~~shall~~should~~ poll the ~~receiver~~Receiver in the case ~~of~~where “polling every Poll_SDU SDU” is configured by ~~higher layersthe upper layer~~upper layers. ~~This is an~~ It represents the upper limit for ~~the~~ state variable VT(SDU).~~ state variable,~~ ~~w~~When VT(SDU) ~~reaches~~equals the value Poll_SDU a poll ~~is will~~shall be transmitted to the peer entity.

d) Poll_Window.

This protocol parameter indicates when the transmitter ~~will~~shall~~should~~ poll the ~~receiver~~Receiver in the case ~~of performing~~where “window-based polling” is configured by ~~higher layersthe upper layer~~upper layers. The range of values of this parameter ~~shall will~~shall be 0 ≤ Poll_Window ≤ 100. A poll is triggered for each PDU when J ≥ Poll_Window, where J is the window transmission percentage defined ~~by~~as:

$$J = \frac{(4096+VT(S) - VT(A)) \ \mathbf{mod}\ 4096}{VT(WS)} * 100 ,$$

where the constant 4096 is the modulus for AM described in ~~S~~subclause 9.4.

e) MaxRST.

~~It~~ This protocol parameter ~~is~~ indicates the maximum ~~value for the~~ number of retransmissions of a RESET PDU. ~~This parameter is an~~ It represents the upper limit ~~of counter~~for state variable VT(RST). When ~~the value of~~ VT(RST) ~~comes reaches~~equals the value~~to~~ MaxRST, unrecoverable error ~~shall will~~shall be indicated to ~~upper higher layers~~~~the upper layer~~upper layers.

f) Configured_Tx_Window_Size.

This protocol parameter indicates ~~T~~both the maximum allowed transmitter window size and the ~~initial value~~value for the state variable VT(WS).

g) Configured_Rx_Window_Size.

This protocol parameter indicates the ~~allowed~~ receiver window size.

h) MaxMRW.

~~It~~ This protocol parameter indicates~~is~~ the maximum value for the number of retransmissions of a MRW SUFI~~command~~. ~~This parameter is an~~ It represents the upper limit ~~of counter~~for state variable VT(MRW). When ~~the value of~~ VT(MRW) ~~comes reaches~~equals the value~~to~~ MaxMRW, the RLC RESET procedure ~~shall will~~shall be initiated.

9.7 Specific functions

The functions defined in this subclause are normative.

9.7.1 Polling function for acknowledged mode

The Polling function is used by the Sender~~transmitter of AMD PDUs may poll~~ to request the ~~receiver~~ peer RLC entity for a status report ~~(consisting of one or several STATUS PDUs)~~. The "Polling bit" in the AMD PDU indicates the poll request. ~~If there is no PDU to be transmitted and all PDUs have already been acknowledged, the receiver shall not be polled.~~ There are several triggers for initiating the Polling function. ~~setting the polling bit. Upper layers control,~~ ~~w~~Which of the triggers ~~should~~ shall be used is configured by upper layers for each RLC entity. ~~If a poll has been triggered while a Timer_Poll is active a new Polling function is initiated after Timer_Poll has expired.~~ The ~~F~~following triggers ~~are possible~~can be configured:

1) Last PDU in buffer.

The ~~s~~Sender triggers ~~a~~ the ~~p~~Polling function when the last PDU available for transmission is submitted to lower layer~~transmitted~~.

2) Last PDU in R~~r~~etransmission buffer.

The ~~s~~Sender triggers ~~a~~ the ~~p~~Polling function when the last PDU to be retransmitted is submitted to lower layer~~transmitted~~.

3) Poll timer.

The timer Timer_Poll is started and stopped according to subclause 9.5 a). ~~when the successful or unsuccessful transmission of a PDU containing a poll is indicated by lower layer (in UE) or a PDU containing a poll is submitted to lower layer (in UTRAN) and if the criterion for stopping the timer has not occurred before~~ When the timer Timer_Poll expires ~~a new~~ the Sender triggers the ~~p~~Polling function ~~is triggered~~.

4) Every Poll_PDU PDU.

The ~~s~~Sender triggers ~~a~~ the ~~p~~Polling function every Poll_PDU PDU. Both retransmitted and new PDUs shall be counted.

5) Every Poll_SDU SDU.

The ~~s~~Sender triggers ~~a~~ the ~~p~~Polling function every Poll_SDU SDU.

6) Window based.

The ~~s~~Sender triggers ~~a~~ the ~~p~~Polling function when the condition described in subclause 9.6 d) ("Poll_Window") is fullfilled.~~it has reached Poll_Window% of the transmission window.~~

7) Timer based.

The ~~s~~Sender triggers ~~a~~ the ~~p~~Polling function periodically.

Either the triggers 1) and 2) or the trigger 7) should be configured for every RLC entity to avoid deadlock situations.~~Either the trigger "Last PDU in buffer" and "Last PDU in retransmission buffer" or "Timer based" can be chosen to avoid deadlock for every RLC entity. The network also controls if the poll prohibit function shall be used. The poll bit shall be set to 0 if the poll prohibit function is used and the timer Timer_Poll_Prohibit is active. If a poll was triggered during the prohibit time defined in subclause 9.5 b) (Timer_Poll_Prohibit), the poll shall be delayed until the timer expires. Only one poll shall be transmitted when the timer expires even if several polls were triggered during the prohibit time. This function has higher priority than any of the above-mentioned triggers.~~

The Poll Prohibit function is used by the Sender to delay the initiation of the Polling function. Usage of the Poll Prohibit function is configured by upper layers. The Poll Prohibit function consists of starting the timer Timer_Poll_Prohibit according to subclause 9.5 b) and delaying the Polling function according to the following rules:

When the Polling function is triggered the Sender shall:

- if ~~the timer Timer_Poll_Prohibit is not active~~polling is not prohibited (see suclause 9.5 b)); and
- if there is one or more AMD PDUs to be transmitted or there are AMD PDUs not acknowledged by the Receiver:
    - initiate the Polling function by setting the polling bit according to subclause 11.3.2.1.1.

Upon expiry of the timer Timer_Poll_Prohibit, the Sender shall:

- if the Polling function was triggered at least once while the timer Timer_Poll_Prohibit was active; and
- if there is one or more AMD PDUs to be transmitted or there are AMD PDUs not acknowledged by the Receiver:
    - initiate the Polling function once by setting the polling bit according to subclause 11.3.2.1.1.

9.7.2 STATUS transmission for acknowledged mode

The ~~r~~Receiver ~~of AMD PDUs~~ transmits status reports to the Sender in order to inform the Sender about which AMD PDUs have been received and not received. ~~(e~~Each status report consists of one or several STATUS PDUs.~~) to the sender in order to inform about which PDUs that have been received and not received. There are several triggers for sending a status report. Upper layers control which triggers should be used for each RLC entity, except for one, which is always present.~~ The ~~r~~Receiver shall always send a status report when receiving a poll request. ~~Except for that trigger~~ Additionally the following triggers for transmission of status reports are configurable by upper layers:

1) Detection of missing PDU(s).

If the ~~r~~Receiver detects one or several missing AMD PDUs it shall trigger the transmission of a status report to the ~~s~~Sender.

2) Timer based ~~STATUS~~ status report transfer.

The ~~r~~Receiver triggers the transmission of a status report ~~periodically~~ to the ~~s~~Sender periodically. The timer Timer_Status_Periodic controls the time period according to subclause 9.5 g). When “Periodical Status blocking” is configured by upper layers, the trigger shall not be active.

3) The EPC mechanism.

The timer Timer_EPC is started according to subclause 9.5 c) and the state variable VR(EP) is set and decreased according to subclause 9.7.4.~~when the successful or unsuccessful transmission of the first STATUS PDU of a~~

~~status report is indicated by lower layer (in UE) or the first STATUS PDU of a status report is submitted to lower layer (in UTRAN).~~ If not all AMD PDUs requested for retransmission have been received before the variable VR(EP) ~~has reached~~equalled zero, a new status report is ~~transmitted to the peer entity~~triggered by the Receiver. A more detailed description of the EPC mechanism is given in subclause 9.7.4.

There are two functions that can prohibit the ~~r~~Receiver from sending a status report containing any of the SUFIs LIST, BITMAP, RLIST or ACK. Status reports containing other SUFIs are not prohibited. Upper layers control which functions should be used for each RLC entity. If any of the following functions is used the ~~sending~~ transmission of the status report shall be delayed, even if any of the triggering conditions above are fulfilled:

1) STATUS prohibit.

   The timer Timer_Status_Prohibit is started accoring to subclause 9.5 f).~~when the successful or unsuccessful transmission of the last STATUS PDU of a status report is indicated by lower layer (in UE) or the last STATUS PDU of a status report is submitted to lower layer (in UTRAN). The prohibit time is calculated from the time the last STATUS PDU of a status report is submitted to lower layer until the timer has expired.~~ The ~~receiving side~~ Receiver is not allowed to transmit a status report ~~during~~ while ~~the prohibit time~~~~timer Timer_Status_Prohibit is active~~acknowledgement is prohibited (see subclause 9.5 f)). If a status report was triggered during thise ~~prohibit~~ time, the status report is transmitted after the timer Timer_Status_Prohibit ~~prohibit time~~ has expired, as described below. ~~The receiver shall only send one status report, even if there are several triggers during the prohibit time. This timer only prohibits the transmission of status reports containing any of the SUFIs LIST, BITMAP, RLIST or ACK. Status reports containing other SUFIs are not prohibited.~~

2) The EPC mechanism.

   If the "EPC mechanism" is active and the ~~sending~~ transmission of a status report is triggered it shall be delayed until the "EPC mechanism" has ended, as described below. ~~The receiver shall only send one status report, even if there are several triggers when the timer is active or the counter is counting down. This mechanism only prohibits the transmission of status reports containing any of the SUFIs LIST, BITMAP, RLIST or ACK. Status reports containing other SUFIs are not prohibited.~~

When a status report is triggered the Receiver shall:

- if transmission of status reports is not prohibited by any of the functions "STATUS prohibit" or "EPC mechanism":
  - assemble and transmit the status report to the Sender, as specified in subclause 11.5.2.2 and 11.5.2.3~~X~~;
- otherwise (if the status report is prohibited by at least one of the functions "STATUS prohibit" or "EPC mechanism")
  - if MRW, MRW_ACK or WINDOW SUFIs are required in the status report:
    - send a status report immediately excluding ACK, LIST, BITMAP, and RLIST SUFIs;
  - if ACK, LIST, BITMAP, or RLIST SUFIs are required in the status report:
    - delay sending these SUFIs until the prohibit function terminates.

Upon expiry of the timer Timer_Status_Prohibit or termination of the "EPC mechanism", the Receiver shall:

- if at least one status report was triggered during the time the transmission of a status reports was prohibited that could not be transmitted due to prohibition; and
- if transmission of a status reports is no longer prohibited by any of the functions "STATUS prohibit" or "EPC mechanism":
  - transmit one status report to the Sender, using the procedure described in subclause 11.5.2.3~~X~~.

9.7.3 SDU discard function for acknowledged, unacknowledged, and transparent mode

The SDU discard function is used by the Sender ~~allows~~ to discharge RLC PDUs from the RLC PDU buffer ~~on the transmitter side~~, when the transmission of the RLC PDUs does not succeed~~ss~~ for a period of ~~long~~ time or for a number of retransmissions. The SDU discard function allows to avoid buffer overflow. There ~~will be~~ are several alternative operation modes of the RLC SDU discard function. Upper layers control, which discard function shall be used for each RLC entity.

The following is a list of operation modes for the RLC SDU discard function, which are described in detail in the subsequent subclauses.

Table 9.2: List of criteria that control when to perform SDU discard

Operation mode	Presence
Timer based discard, with explicit signalling	Network controlled
Timer based discard, without explicit signalling	Network controlled
SDU discard after MaxDAT number of retransmissions	Network controlled
No_discard after MaxDAT number of retransmissions	Network controlled

9.7.3.1 Timer based discard, with explicit signalling

This alternative is only applicable to RLC entities operating in acknowledged mode. ~~This alternative~~ It uses a timer based triggering of SDU discard (Timer_Discard). This makes the SDU discard function insensitive to variations in the channel rate and provides means for exact definition of maximum delay. However, the SDU loss rate of the connection is increased as SDUs are discarded.

~~For every SDU received from a upper layer, timer monitoring of the transmission time of the SDU is started. If the transmission time exceeds a predefined value for an SDU in acknowledged mode RLC, this SDU is discarded in the transmitter. Following which, if one or more segments of the SDU have been submitted to a lower layer, a Move Receiving Window (MRW) command is sent to the receiver so that AMD PDUs carrying that SDU are discarded in the receiver and the receiver window is updated accordingly. If Send MRW is configured, an expired SDU whose segments were not submitted to a lower layer is also informed to the receiver by a MRW command.~~
For every SDU received from upper layers, the Sender shall:

- start a timer Timer_Discard ~~monitoring of the transmission time of the SDU~~.

When the ~~transmission time~~timer Timer_Discard of a SDU expires~~exceeds the configured value for a SDU~~, the Sender shall:

- discard the SDU;
- ~~if "Send MRW" is not configured and no segments of the discarded SDU were submitted to the lower layer:~~
  - ~~not utilise explicit signalling;~~
- ~~otherwise (~~if "Send MRW" is configured, or one or more segments of the discarded SDU were submitted to the lower layer~~)~~:
  - utilise ~~explicite~~explicit signalling to inform the Receiver according to subclause 11.6.

~~NOTE: When the concatenation function is active, PDUs carrying segments of other SDUs that have not timed out shall not be discarded.~~

~~The MRW command is defined as a super-field in the RLC STATUS PDU (see subclause 9.2), and piggybacked to status information of transmissions in the opposite direction. If the MRW command has not been acknowledged by receiver, it will be retransmitted. Therefore, SDU discard variants requiring peer-to-peer signalling are only possible for full duplex connections.~~

9.7.3.2 Timer based discard, without explicit signalling

This alternative is only applicable to RLC entities operating in unacknowledged or transparent mode. ~~This alternative~~ It uses the same timer based trigger for SDU discard (Timer_Discard) as the one described in the subclause 9.7.3.1. The difference is that this discard method does not use any peer-to-peer signalling. ~~This function is applied only for unacknowledged and transparent mode RLC and peer-to-peer signalling is never needed. The SDUs are simply discarded in the transmitter, once the transmission time is exceeded. For UM RLC, how to update the sequence number is specified in subclause 11.2.4.3.~~

For every SDU received from upper layers, the Sender shall:

- start timer monitoring of the transmission time of the SDU.

When the transmission time exceeds the configured value for a SDU, the Sender shall:

- discard the SDU without ~~explicite~~explicit signalling (for RLC entities operating in unacknowledged mode apply subclause 11.2.4.3 for updateing the state variables).

9.7.3.3 SDU discard after MaxDAT number of retransmissions

This alternative uses the number of retransmissions as a trigger for SDU discard, and is therefore only applicable for acknowledged mode RLC. This makes the SDU discard function dependent on~~f~~ the channel rate. Also, this variant of the SDU discard function strives to keep the SDU loss rate constant for the connection, on the cost of a variable delay. ~~SDU discard is triggered at the transmitter, and a MRW command is necessary to convey the discard information to the receiver, like in the timer-based discard with explicit signalling.~~

If MaxDAT number of retransmissions is reached for a AMD PDU, the Sender shall:

- discard all SDUs segments of which are contained in the AMD PDU and utilise ~~explicite~~explicit signalling to inform the Receiver according to clause 11.6.

9.7.3.4 No_discard after MaxDAT number of retransmissions

This alternative uses the number of retransmissions, and is therefore only applicable for acknowledged mode RLC. ~~Reset procedure shall be initiated after MaxDAT number of retransmissions of an AMD PDU (see subclause 11.3.4.4).~~

If MaxDAT number of retransmissions is reached for an AMD PDU, the Sender shall:

- initiate the RLC Reset procedure (see subclause 11.3.4.4).

9.7.3.5 SDU discard not configured

If SDU discard has not been configured for an unacknowledged mode RLC entity, SDUs in the transmitter shall not be discarded unless the T~~t~~ransmission buffer is full. ~~If the transmission buffer is full, SDUs may be discarded using SDU discard without explicit signalling. If no segments of the SDU has been transmitted, the SDU may be removed from the buffer without using any of the SDU discard procedures.~~

When the T~~t~~ransmission buffer in an unacknowledged mode RLC entity is full, the Sender may:

- if segments of the SDU to be discarded have been submitted to lower layer:
  - discard the SDU without ~~explicite~~explicit signalling according to subclause 11.2.4.3;
- otherwise, if no segments of the SDU to be discarded have been submitted to lower layer:
  - remove the SDU from the T~~t~~ransmission buffer without utilising any of the discard procedures.

~~If SDU discard has not been configured for a transparent mode RLC entity, SDUs in the transmitter shall be transmitted in the first possible TTI and other not yet transmitted SDUs received from upper layer in previous TTIs shall be discarded upon reception of a new SDU from upper layer.~~
If SDU discard has not been configured for a transparent mode RLC entity, the Sender shall upon reception of new SDUs from upper layer:

- discard all SDUs received from upper layer in previous TTIs that are not yet submitted to lower layer;
- submit the new SDUs in the first possible TTI.

For an acknowledged mode RLC entity, an SDU discard mode is always configured.

9.7.4 The Estimated PDU Counter for acknowledged mode

The Estimated PDU Counter (EPC) is only applicable for RLC entities operating in acknowledged mode. The EPC is a mechanism configured by ~~higher layer~~upper layers used for scheduling the retransmission of status reports in the

~~r~~Receiver~~ side~~. With this mechanism, the ~~r~~Receiver will send a new status report in which it requests for AMD PDUs not yet received. The time between two subsequent status report retransmissions is not fixed, but it is controlled by both the timer Timer_EPC and the state variable VR(EP), which adapt this time to the round trip delay and the current bit rate, indicated in the TFI, in order to minimise the delay of the status report retransmission.

When a ~~STATUS~~ status report is triggered by some mechanisms and it is submitted to lower layer (in UTRAN) or the successful or unsuccessful transmission of it is indicated by lower layer (in UE) to request for retransmitting one or more missing AMD PDUs, the variable VR(EP) is set equal to the number of requested AMD PDUs. At least one requested AMD PDU is needed to activate the EPC mechanism. The variable VR(EP) is a counter, which is decremented every transmission time interval with the estimated number of AMD PDUs that should have been ~~transmitted~~ received during that transmission time interval on the corresponding logical channel.

~~A special~~ The timer~~, called~~ Timer_EPC~~,~~ controls the maximum time that the variable VR(EP) needs to wait before it will start counting down. This timer starts immediately after a transmission of a retransmission request from the ~~r~~Receiver (when the first STATUS PDU of the status report is submitted to lower layer (in UTRAN) or the successful or unsuccessful transmission of it is indicated by lower layer(in UE)). The initial value of the timer Timer_EPC is configured by upper layers. It typically depends on the roundtrip delay, which consists of the propagation delay, processing time in the transmitter and ~~receiver~~Receiver and the frame structure. This timer can also be implemented as a counter, which counts the number of 10 ms radio frames that could be expected to elapse before the first requested AMD PDU is received.

If not all of these requested AMD PDUs have been received correctly when VR(EP) is equal to zero, a new status report will be transmitted and the EPC mechanism will be reset accordingly. The timer Timer_EPC will be started once more when the first STATUS PDU of the status report is submitted to lower layer (in UTRAN) or the successful or unsuccessful transmission of it is indicated by lower layer (in UE). If all of the requested AMD PDUs have been received correctly, the EPC mechanism ends.

9.7.5 Local Suspend function for acknowledged and unacknowledged mode

The upper layers may suspend a~~the~~ RLC entity. ~~The CRLC-SUSPEND-Req indicates this request. The RLC entity shall, when receiving this request, not send RLC PDUs with SN≥VT(S)+N for AM and SN≥VT(US)+N for UM ,where N is given by the CRLC_SUSPEND-Req primitive. The RLC entity shall acknowledge the CRLC-SUSPEND-Req ordering a suspend with a CRLC-SUSPEND-Conf with the current value of VT(S) for AM and VT(US) for UM. When a CRLC-RESUME-Req primitive indicating resume is received, the AM RLC entity enters the acknowledged data transfer ready state if it is in the local suspend state and enters the reset pending state if it is in the reset and suspend state.~~

When ~~the~~ a RLC entity operating in unacknowledged mode is suspended by upper layers with the parameter N, the RLC entity shall:

- acknowledge the suspend request~~order~~ with a confirmation containing the current value of VT(US);
- not send UMD PDUs with sequence number SN≥VT(US)+N.

When a~~the~~ RLC entity operating in acknowledged mode is suspended by upper layers with the parameter N, the RLC entity shall:

- acknowledge the suspend request~~order~~ with a confirmation containing the current value of VT(S);
- not send AMD PDUs with sequence number SN≥VT(S)+N.

When a~~the~~ RLC entity operating in unacknowledged mode is resumed by upper layers, the RLC entity ~~should~~shall:

- resume ~~normal~~ data transfer procedure.~~enter UNACKNOWLEDGED_DATA_TRANSFER_READY state.~~.

When a~~the~~ RLC entity operating in acknowledged mode is resumed by upper layers, the RLC entity ~~should~~shall:

- if the RLC entity is suspended and ~~the~~ a RLC Reset procedure is not ongoing~~activated is in LOCAL_SUSPEND state~~:
  - resume ~~normal~~ data transfer procedure~~enter ACKNOWLEDGED_DATA_TRANSFER_READY state~~.

- otherwise if the RLC entity is suspended and a~~the~~ RLC Reset procedure is ongoing~~activatedin RESET_AND_SUSPEND state~~:
    - remove the suspend constraint;
    - resume the RLC reset procedure according to subclause 11.4~~enter RESET_PENDING state~~.

9.7.6 RLC Stop, RLC Continue function for acknowledged and unacknowledged mode

The upper layer may stop a~~the~~ RLC entity. ~~The stop parameter in the CRLC-CONFIG-Req primitive indicates this request. The RLC entity shall, when receiving this request, not submit any RLC PDUs to lower layer or receive any RLC PDUs. The data transmission and reception is continued when the continue parameter in the CRLC-CONFIG-Req primitive is received. If the continue parameter is received when the RLC entity is not stopped, no action shall be taken.~~

When a~~the~~ RLC entity is stopped, the RLC timers are not affected. ~~Triggered polls and status transmissions are delayed until the RLC entity is continued.~~

When a~~the~~ RLC entity is stopped by upper layers, the RLC entity shall:

- not ~~not~~ submit any RLC PDUs to lower layer or receive any RLC PDUs;
- delay triggered Polling functions or status transmissions until the RLC entity is continued.

When a~~the~~ RLC entity is continued by upper layers, the RLC entity shall:

- if the RLC entity is stopped:
    - continue the data transmission and reception;
    - process the triggered Polling functions and status transmissions.
- otherwise, if the RLC is not stopped:
    - take no action.

9.7.7 RLC re-establishment function for acknowledged and unacknowledged mode

The RLC re-establishment function is applicable for AM and UM and is used when upper layers request a~~the~~ RLC entity to be re-established.

~~When an RLC entity is re-established, the state variables in the RLC entity (see 9.4) shall be reset to their initial value and the configurable parameters shall be set to their configured value. In AM, all RLC PDUs in the RLC receiver and transmitter shall be discarded. In UM, the RLC SDU for which one or more segments have been submitted to a lower layer in the transmitter shall be discarded. The hyper frame number (HFN) in UL and DL shall be set to the value configured by upper layers. After the re-establishment, RLC shall enter the data transfer ready state.~~

When a~~the~~ RLC entity is re-established by upper layers, the RLC entity shall:

- reset the state variables to their initial value;
- set the configurable parameters to their configured value;
- set the hyper frame number (HFN) in UL and DL to the value configured by upper layers;
- if the RLC entity is operating in unacknowledged mode:
    - if it is a receiving UM RLC entity:
        - discard all UMD PDUs;
    - if it is a ~~sending~~transmitting UM RLC entity:

- discard the RLC SDUs for which one or more segments have been submitted to a lower layer~~ in the sending UM RLC entity~~Sender;

~~- discard all UMD PDUs at the receiving UM RLC entity;~~

~~- enter UNACKNOWLEDGED_DATA_TRANSFER_READY state;~~

otherwise if the RLC entity is operating in acknowledged mode:

- discard all AMD PDUs in the Receiver and Sender;

~~- enter ACKNOWLEDGED_DATA_TRANSFER_READY state.~~

9.7.8 Ciphering for acknowledged and unacknowledged mode

The ciphering function is performed in RLC, according to the following rules if a radio bearer is using a non-transparent RLC mode (AM or UM). The data unit that is ciphered, depends on the transmission mode as described below.

- For RLC UM mode, the ciphering unit is the UMD PDU excluding the first octet, i.e. excluding the RLC UMD PDU header. This is shown below in Figure 9.19.

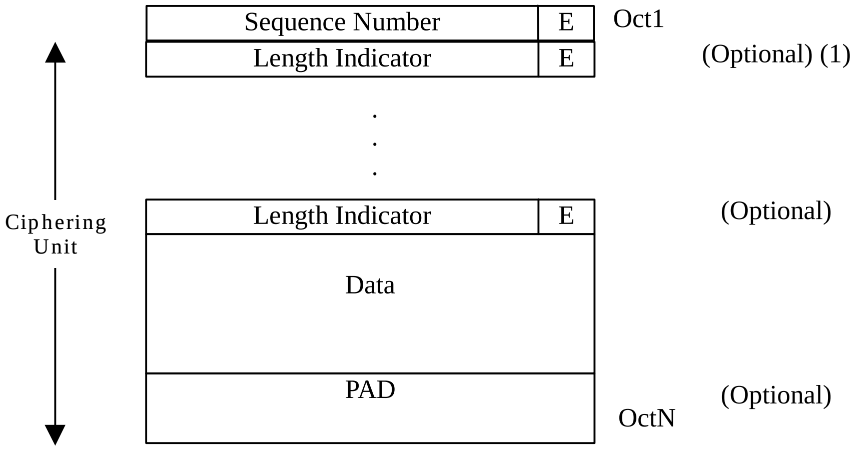


Figure 9.19: Ciphering unit for a UMD PDU

- For RLC AM mode, the ciphering unit is the AMD PDU excluding the first two octets, i.e. excluding the RLC AMD PDU header. This is shown below in Figure 9.20.

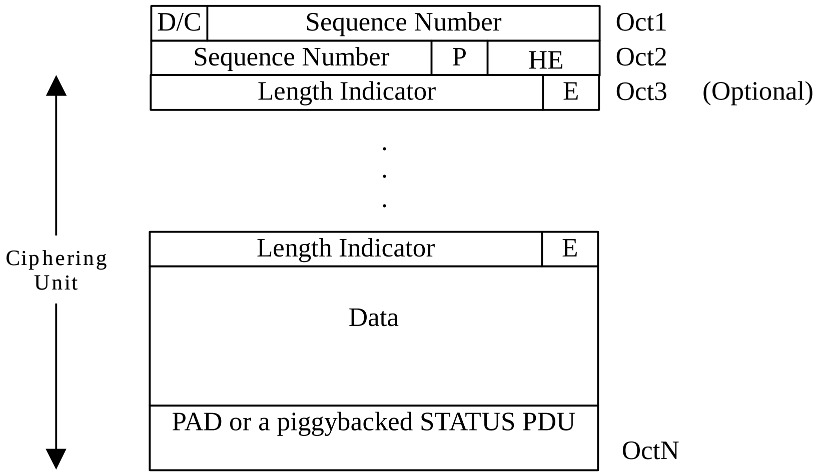


Figure 9.20: Ciphering unit for an AMD PDU

The ciphering algorithm and key to be used are configured by upper layers [8] and the ciphering method shall be applied as specified in [9~~10~~].

The parameters that are required by RLC for ciphering are defined in [9~~10~~] and are input to the ciphering algorithm. The parameters required by RLC which are provided by upper layers [8] are listed below:

- RLC AM HFN (Hyper frame number for radio bearers that are mapped onto RLC AM)
- RLC UM HFN (Hyper frame number for radio bearers that are mapped onto RLC UM)
- BEARER (Radio Bearer ID)
- CK (Ciphering Key)

10 Handling of unknown, unforeseen and erroneous protocol data

Errors and the handling of errors defined in this clause are normative.

~~In case of error situations the following actions are foreseen:~~

~~1) RLC entity shall initiate RESET procedure in case of a protocol error.~~

~~2) RLC entity shall discard invalid PDUs.~~

~~3) RLC entity shall notify upper layer of unrecoverable error occurrence (see subclause 11.4.5.2).~~

~~The list of protocol error cases is reported below:~~

~~Inconsistent state variables;~~

~~If the RLC entity receives a PDU including "erroneous Sequence Number", state variables between peer entities may be inconsistent. Following shows "erroneous Sequence Number" examples:~~

~~- Each Sequence Number of missing PDU informed by SUFI LIST, BITMAP or RLIST is not within the value between "Acknowledge state variable(VT(A))" and "Send state variable(VT(S)) – 1", and~~

~~- LSN of SUFI ACK is not within the value between "Acknowledge state variable(VT(A))" and "Send state variable(VT(S))".~~

~~Inconsistent status indication of a PDU;~~

~~- If a received STATUS PDU indicates different status for the same PDU, then the transmitter shall discard the STATUS PDU.~~

~~Invalid PDU format;~~

~~- If the RLC PDU format contains reserved or invalid values, the RLC PDU shall be discarded.~~

10.1 Erroneous Sequence Number

A STATUS PDU or Piggybacked STATUS PDU including "erroneous Sequence Number" is a STATUS PDU or Piggybacked STATUS PDU that contains:

- a LIST, BITMAP or RLIST SUFI in which the "Sequence Number" of at least one PDU ~~indicated as missing~~that is negatively acknowledged is outside the interval VT(A)≤"Sequence Number"≤ VT(S)-1, or
- an ACK SUFI in which "LSN" is outside the interval VT(A)≤"LSN"≤ VT(S).

If an AM RLC entity receives a STATUS PDU or a Piggybacked STATUS PDU including "erroneous Sequence Number", it shall:

- discard the STATUS PDU or the Piggybacked STATUS PDU;
- initiate the RLC reset procedure (see subclause 11.4).

10.2 Inconsistent status indication

If an AM RLC entity receives a STATUS PDU or a Piggybacked STATUS PDU that indicates different status for the same AMD PDU, it shall:

- discard the STATUS PDU or the Piggybacked STATUS PDU.

10.3 Invalid PDU format

If an UM or AM RLC entity receives a RLC PDU that contains reserved or invalid values (see subclause 9.2), it shall:

- discard the RLC PDU.

11 Elementary procedures

Procedures defined in this clause are normative.

~~In each procedure, RLC entities are functionally identified as Sender or Receiver. This functional distinction is summarised in the message sequence diagram at the beginning of each procedure. Sender and Receiver RLC entities may be either in the UE or the UTRAN. This convention will be used across the procedures.~~

~~The Sender is the RLC entity that transmits data PDUswhich initiates the procedure. The Receiver is the peer RLC entity of the Sender.~~ This description assumes elementary procedures. Interactions between procedures are not described.

11.1 Transparent mode data transfer procedure

11.1.1 ~~Purpose~~General

The transparent mode data transfer procedure is used for transferring ~~of~~ data between two RLC peer entities, which are operating in transparent mode. Data is transferred from Sender to Receiver. This procedure ~~shall~~should only apply to entities in DATA_TRANSFER_READY state. Figure 11.1 below illustrates the elementary procedure for transparent mode data transfer. ~~The sender can be either the UE or the network and the receiver is either the network or the UE.~~

Channels that can be used are DTCH, CCCH (uplink only), SHCCH (uplink only), BCCH and PCCH. The type of logical channel depends on if the RLC entity is located in the user plane (DTCH) or in the control plane (CCCH/BCCH/SHCCH/PCCH).

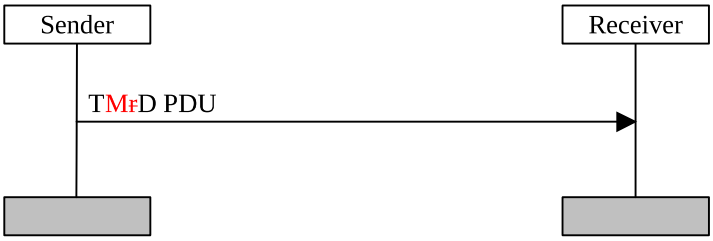


Figure 11.1: Transparent mode data transfer procedure

11.1.2 ~~Initiation~~Transmission of ~~Tr~~TMD PDU

~~The sender initiates this procedure u~~Upon a request of transparent mode data transfer from upper layer, the Sender shall:

- i~~I~~f no SDU discard configuration ~~is configured~~has been made by upper layers:
    - discard ~~all buffered~~ SDUs received in previous TTIs upon reception of new SDUs from ~~the upper layer~~upper layers (see subclause 9.7.3.5);~~.~~

- o~~O~~therwise~~,~~ (if "Timer Based SDU Discard without explicit signalling" is configured):
  - start a timer Timer_Discard for each SDU received from ~~the upper layer~~upper layers (see subclause 9.7.3);
- schedule the RLC SDUs that have been received from upper layer for transmission;
- if one or more RLC SDUs have been scheduled for transmission, the Sender shall;
  - notify the lower layer of reception of data from ~~the upper layer~~upper layers;~~.~~
  - perform the actions specified in subclause 11.1.2.2.

~~. When the sender is in data transfer ready state it shall put the data received from the upper layer into TrD PDUs. If required RLC shall perform segmentation.~~

~~Channels that can be used are DTCH, CCCH (uplink only), SHCCH (uplink only), BCCH and PCCH. The type of logical channel depends on if the RLC entity is located in the user plane (DTCH) or in the control plane (CCCH/BCCH/SHCCH/PCCH). One or several PDUs may be transmitted in each transmission time interval (TTI). For each TTI, MAC decides which PDU size shall be used (applicable when segmentation is used) and how many PDUs shall be transmitted. The SDUs that cannot be transmitted in a TTI shall be buffered according to the discard configuration set by upper layers.~~

~~If timer based SDU discard is used, the timer Timer_Discard shall be started when the RLC entity receives an SDU from upper layer. One timer is used for each SDU that is received from upper layer.~~

11.1.2.1 ~~Tr~~TMD PDU contents to set

The Sender shall set the data field of the ~~Tr~~TMD PDU to all or a subset of the data contained in the SDU as described in subclause 11.1.2.2.~~The TrD PDU includes a complete SDU or a segment of an SDU. How to perform the segmentation is decided upon when the service is established. No overhead or header is added, instead segmentation is done based on which of the transport formats of the transport channel that will be used. A particular transport format informs the receiver how the segmentation was performed.~~

11.1.2.~~X~~2 Submission of TMD PDUs to the lower layer

If one or more RLC SDUs have been scheduled for transmission, according to Subclause 11.1.2, the Sender shall:

- if it is configured for segmented operation:
  - inform the lower layer of the size of the next SDU to be sent;
  - segment the SDU according to the PDU size indicated by the lower layer;
- otherwise~~,~~ (the Sender is configured for non-segmented operation):
  - inform the lower layer of the number and size of SDUs available for transmission;
- submit to the lower layer, the requested number of TMD PDUs;
- buffer the SDUs that are not submitted to the lower layer according to the discard configuration (see subclause 9.7.3).

11.1.3 Reception of ~~Tr~~TMD PDU

Upon ~~reception~~ delivery by the lower layer of a set of ~~Tr~~TMD PDUs (received within one TTI), the Receiver shall:

- if it is configured for segmented operation:
  - ~~receiving entity~~ reassembles ~~(if segmentation was performed)~~ the TMD PDUs received in one TTI into one RLC SDU~~s~~;~~.~~
- otherwise (it is configured for non-segmented operation):
  - treat each received TMD PDU as a SDU;

- if "Delivery of Erroneous SDUs" is configured as "no":
    - ~~RLC delivers~~submit only ~~all~~the RLC SDUs received without error to ~~the upper layer~~upper layers through the ~~Tr~~TM-SAP;~~.~~
- else if "Delivery of Erroneous SDUs" is configured as "yes":
    - submit all RLC SDUs to ~~the upper layer~~upper layers through the TM~~R~~-SAP;~~.~~
    - provide an error indication for each SDU received in error;
- otherwise if "Delivery of Erroneous SDUs" is configured as "No detect":
    - submit all RLC SDUs to ~~the upper layer~~upper layers through the TM-SAP.

~~If delivery of erroneous SDUs is configured as 'yes' by an upper layer, the receiver shall deliver an erroneous SDU to upper layer with an error indication. If delivery of erroneous SDUs is configured as 'no' by an upper layer the receiver shall discard the erroneous SDU. If delivery of erroneous SDUs is configured as 'No detect' by an upper layer, all SDUs shall be delivered to upper layer without error indication.~~

If segmentation is performed in transparent mode RLC, an SDU is erroneous if one or more of the ~~Tr~~TMD PDUs received in a TTI contains an error. If segmentation is not performed, an SDU is erroneous if the corresponding ~~Tr~~TMD PDU is erroneous.

11.1.4 Abnormal cases

11.1.4.1 Void

11.1.4.2 SDU discard without explicit signalling

Upon expiry of the timer Timer_Discard in~~on~~ the S~~s~~ender~~-side~~, the S~~s~~ender shall:

- ~~-~~discard the associated SDU.~~;~~

~~.~~In the case where the TFC selection exchange has been initiated by sending the RLC Entity Info parameter to MAC, the UE may wait until after it provides MAC with the requested set of PDUs before discarding the afore-mentioned SDU.

11.2 Unacknowledged mode data transfer procedure

11.2.1 ~~Purpose~~General

The unacknowledged mode data transfer procedure is used for transferring data between two RLC peer entities, which are operating in unacknowledged mode. Data is transferred from Sender to Receiver. This procedure ~~shall~~should only apply to RLC entities in DATA_TRANSFER_READY state or LOCAL_SUSPEND state. Figure 11.2 below illustrates the elementary procedure for unacknowledged mode data transfer~~. The sender can be either the UE or the network and the receiver is either the network or the UE~~.

Channels that can be used are DTCH, DCCH, CCCH (downlink only), CTCH, SHCCH (downlink only). The type of logical channel depends on if the RLC entity is located in the user plane (DTCH, CTCH) or in the control plane (DCCH/CCCH(downlink only)/SHCCH(downlink only)). One or several PDUs may be transmitted in each transmission time interval (TTI). For each TTI, MAC decides which PDU size shall be used and how many PDUs shall be transmitted.

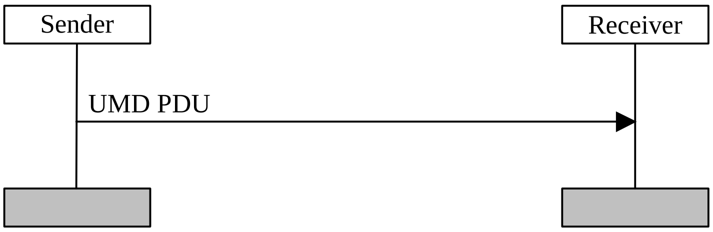


Figure 11.2: Unacknowledged mode data transfer procedure

11.2.2 ~~Initiation~~Transmission of UMD PDU

~~The sender initiates this procedure u~~Upon a request of unacknowledged mode data transfer from upper layer, the Sender shall:~~.~~

- if no SDU discard configuration has been made by upper layers~~configuration is configured~~:
  - ~~not~~only discard ~~any~~ SDUs when ~~in the transmitter unless~~ the ~~transmitter~~Transmission buffer is full (see subclause 9.7.3)
- if "Timer based SDU Discard without explicit signalling" is configured:
  - start a timer Timer_Discard for each SDU received from upper layer (see subclause 9.7.3);
- schedule the RLC SDUs received from upper layer for transmission;
- if one or more RLC SDUs have been scheduled for transmission~~, the Sender shall~~;
  - notify the lower layer of reception of data from ~~the upper layer~~upper layers.
  - perform the actions specified in subclause 11.2~~1~~.2.2.

~~When the sender is in data transfer ready state it shall segment and, if possible, concatenate the data received from the upper layer into PDUs.~~

~~Channels that can be used are DTCH, DCCH, CCCH (downlink only), CTCH, SHCCH (downlink only). The type of logical channel depends on if the RLC entity is located in the user plane (DTCH, CTCH) or in the control plane (DCCH/CCCH(downlink only)/SHCCH(downlink only)). One or several PDUs may be transmitted in each transmission time interval (TTI). For each TTI, MAC decides which PDU size shall be used and how many PDUs shall be transmitted.~~

~~The VT(US) state variable shall be updated for each UMD PDU that is transmitted.~~

~~If timer based SDU discard is used, the timer Timer_Discard shall be started when the RLC entity receives an SDU from upper layers. One timer is used for each SDU that is received from upper layers.~~

A UMD PDU ~~will~~ shall be considered to be a padding PDU if it consists only of an RLC Header with one length indicator (indicating that the rest of the PDU is padding) and padding.

11.2.2.1 UMD PDU contents to set

The Sender shall:

- set the field "Sequence Number" equal to VT(US);
- set a "Length Indicator" field for each SDU that ends in the UMD PDU according to subclause 9.2.2.8;

For each "Extension bit" field in the RLC header, the ~~sender~~Sender shall:

- if the next field in the UMD PDU is a "Length Indicator";
  - set the "Extension bit" to "1";
- otherwise if the next field in the UMD PDU is data:

  - set the "Extension bit" to "0";

~~The Sequence Number field shall be set equal to VT(US).~~

~~The Extension bit shall be set to 1 if the next field is a length indicator field, otherwise it shall be set to zero.~~

~~One length indicator field shall be included for each end of an SDU that the PDU includes. The LI fields shall be set as specified in subclause 9.2.2.8.~~

11.2.2.~~X~~2 Submission of UMD PDUs to the lower layer

If one or more SDUs have been scheduled for transmission according to subclause 11.2.2, the Sender shall:

- inform the lower layer of the number and size of SDUs scheduled for transmission;
- segment, and if possible concatenate the SDUs according to the PDU sizes indicated by the lower layer;
- submit to the lower layer, the requested number of UMD PDUs;
- update VT(US) for each UMD PDU submitted to the lower layer (see subclause 9.4);
- buffer the SDUs that are not submitted to the lower layer according to the discard configuration (see subclause 9.7.3).

11.2.3 Reception of UMD PDU

Upon ~~reception~~delivery of a set of UMD PDUs from the lower layer, the ~~receiver~~Receiver shall:

- update VR(US)~~ state variable~~ according to ~~the ~~each received UMD PDU~~(s)~~ (see subclause 9.4);
- if the updating step of VR(US) is not equal to one (i.e. one or more UMD PDUs are missing);
  - discard the SDUs that have segments in the missing UMD PDUs;
- reassemble the received UMD PDUs into RLC SDUs~~.~~;
- submit the RLC SDUs to ~~the upper layer~~upper layers through the UM-SAP.

~~The PDUs are reassembled into RLC SDUs. If the updating step of the variable VR(US) is greater than one, one or more PDUs are missing. The SDUs that have segments in these missing PDUs shall be discarded. RLC delivers the RLC SDUs to the upper layers through the UM-SAP.~~

11.2.4 Abnormal cases

11.2.4.1 Length Indicator value reserved for UMD PDU

Upon ~~reception ~~delivery by the lower layer of an UMD PDU that contains a "Length Indicator" value specified to be reserved for UMD PDUs in this version of the protocol, the ~~receiver~~Receiver shall:

- discard that UMD PDU and treat the UMD PDU as missing~~.~~

11.2.4.2 Invalid length indicator value

If the "L~~l~~ength I~~i~~ndicator" of an UMD PDU has a value that is larger than the PDU size – RLC header size and is not one of the predefined values listed in the table of subclause 9.2.2.8, the ~~receiver~~Receiver shall:

- discard the UMD PDU and treat the UMD PDU as missing~~PDU shall be discarded and treated as a missing PDU.~~

11.2.4.3 SDU discard without explicit signalling

Upon expiry of the timer Timer_Discard ~~on the~~ in the S~~s~~ender~~-side~~ the S~~s~~ender shall:

- ~~-~~discard the associated SDU;

- for the first UMD PDU to be transmitted after the discard operation the Sender shall:

  - increment VT(US) so that the "Sequence Number" field in this UMD PDU is incremented with two compared with the previous UMD PDU;

  - fill the first data octet in this UMD PDU with the first octet of an RLC SDU;

  - set the first "Length Indicator" in this UMD PDU to indicate that the previous RLC PDU was exactly filled with the last segment of an RLC SDU (to avoid that the ~~receiver~~Receiver unnecessarily discards an extra SDU).

~~-~~In the case where the TFC selection exchange has been initiated by sending the RLC Entity Info parameter to MAC, the UE may wait until after it provides MAC with the requested set of PDUs before discarding the afore-mentioned SDU~~. For UM RLC, SN of the UMD PDUs shall be incremented by a step of 2 for the first PDU transmitted after a Discard Operation to indicate that there were some RLC SDUs discarded before this RLC PDU. The first data octet in this RLC PDU shall be the first octet of an RLC SDU.~~. ~~To prevent the receiver from discarding one extra SDU, an LI field shall be added in the first PDU transmitted after a Discard Operation. The value of the LI field shall be the value indicating that the previous SDU filled exactly the previous RLC PDU.~~

11.3 Acknowledged mode data transfer procedure

11.3.1 General~~Purpose~~

The acknowledged mode data transfer procedure is used for transferring~~ of~~ data between two RLC peer entities, which are operating in acknowledged mode. Data is transferred from Sender to Receiver. This procedure ~~shall~~should only apply to RLC entities in DATA_TRANSFER_READY state or LOCAL_SUSPEND state. Figure 11.3 below illustrates the elementary procedure for acknowledged mode data transfer~~. The sender can be either the UE or the network and the receiver is either the network or the UE~~.

The AMD PDUs shall be transmitted on the DCCH logical channel if the Sender is located in the control plane and on the DTCH if it is located in the user plane. One or several PDUs may be transmitted in each transmission time interval (TTI) and MAC decides how many PDUs shall be transmitted in each TTI.

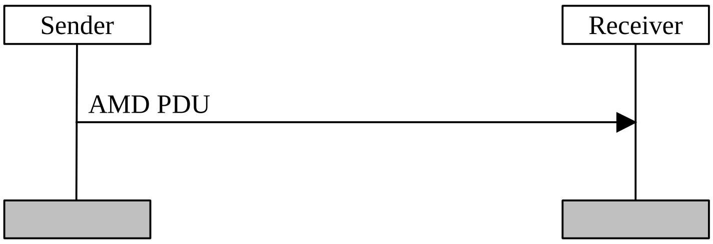


Figure 11.3: Acknowledged mode data transfer procedure

11.3.2 ~~Initiation~~Transmission of AMD PDU

~~The sender initiates this procedure u~~Upon a request of acknowledged mode data transfer from upper layers or upon retransmission of AMD PDUs, the Sender shall:

- when RLC SDUs are received from ~~the upper layer~~upper layers:~~;~~

  - segment the RLC SDUs into AMD PDUs where the fixed PDU size is configured by ~~higher~~upper layer;

  - set a "Length Indicator" field for each SDU that ends in the AMD PDU according to subclause 9.2.2.8;

  - if "Timer based SDU Discard with explicit signalling" is configured:

    - start a timer Timer_Discard for each SDU received from upper layer (see subclause 9.7.3);

  - s~~S~~chedule the AMD PDUs for transmission;

- if one or several AMD PDUs have~~s~~ been negatively acknowledged (see subclause 11.5.3):~~;~~
    - schedule the AMD PDUs ~~indicated as missing~~that were negatively acknowledged for retransmission;
- if a poll has been triggered by either ~~any of~~ the poll triggers "Poll timer" or "Timer based"~~,~~ (see subclause 9.7.1);and
- if no AMD PDU is scheduled for transmission or retransmission:~~;~~
    - if the value of "Configured_Tx_Window_Size" is larger than or equal to "2048":~~;~~
        - select the AMD PDU with "Sequence Number" equal to VT(S)-1;
    - otherwise if the "Configured_Tx_Window_Size" is less than "2048";~~;~~
        - select the AMD PDU with "Sequence Number" equal to VT(S)-1; or
        - select an AMD PDU that has not yet been acknowledged by the peer entity;
    - schedule the selected AMD PDU for retransmission (in order to transmit a poll).

The Sender may also schedule an AMD PDU for retransmission even if none of the criteria above is fulfilled. In this case, the Sender may:

- if the value of "Configured_Tx_Window_Size" is larger than or equal to "2048":~~;~~
    - select the AMD PDU with "Sequence Number" equal to VT(S)-1;
- otherwise if the "Configured_Tx_Window_Size" is less than "2048":~~;~~
    - select the AMD PDU with "Sequence Number" equal to VT(S)-1; or
    - select an AMD PDU that has not yet been acknowledged by the peer entity;
- schedule the selected AMD PDU for retransmission

Each time an AMD PDU is scheduled for transmission or retransmission ~~by one of the criteria listed in this subclause~~, the Sender shall:

- notify the lower layer that data is available for transmission;
- perform the actions specified in subclause 11.3.2.2.~~. Retransmitted PDUs have higher priority than PDUs transmitted for the first time.~~

~~The sender is only allowed to retransmit PDUs that have been indicated missing by the receiver. An exception is the PDU with SN VT(S)-1, which can be retransmitted. In addition, a PDU that has not yet been acknowledged, may be retransmitted if Configured_Tx_Window_Size is less than 2048.~~

~~RLC shall segment the data received from the upper layers into AMD PDUs. The~~ AMD ~~PDUs shall be transmitted on the DCCH logical channel if the sender~~Sender ~~is located in the control plane and on the DTCH if it is located in the user plane. One or several PDUs may be transmitted in each transmission time interval (TTI) and MAC decides how many PDUs shall be transmitted in each TTI. In the UE, the PDUs that cannot be transmitted in a TTI (i.e. MAC has indicated that some of the available PDUs can not be transmitted) shall be buffered according to the discard configuration set by upper layers.~~

~~The VT(DAT) state variables shall be updated for each AMD PDU that is transmitted. The PDU shall not have a Sequence Number ≥ VT(MS), except for the sequence number VT(S)-1; a PDU with this sequence number may be sent also when VT(S) ≥ VT(MS).~~

~~If the poll bit is set in any of the AMD PDUs and the timer Timer_Poll shall be used, the sender shall start the timer Timer_Poll when the successful or unsuccessful transmission of a PDU with the set poll bit is indicated by lower layer (in UE) or submitted to lower layer (in UTRAN).~~

~~If timer based SDU discard is used, the timer Timer_Discard shall be started when the RLC entity receives an SDU from upper layers. One timer is used for each SDU that is received from upper layers.~~

~~If the trigger for polling, "Every Poll_PDU PDU", is used, the VT(PDU) shall be increased by 1 for each PDU that is transmitted.~~

~~If the trigger for polling, "Every Poll_SDU SDU", is used, the VT(SDU) shall be increased by 1 for each SDU that is transmitted.~~

In AM, a PDU ~~will~~ shall be considered to be a padding PDU if it is:

- a~~A~~n AMD PDU consisting only of an RLC Header with one "L~~l~~ength I~~i~~ndicator" (indicating that the rest of the PDU is padding) and padding; or~~.~~
- a~~A~~ Status PDU consisting only of a NO_MORE SUFI.

11.3.2.1 AMD PDU contents to set

- i~~I~~f the AMD PDU is transmitted for the first time, the Sender shall:
  - set the "Sequence Number" field~~ shall be set~~ equal to VT(S)~~; and VT(S) shall be updated.~~
  - set a "Length Indicator" field for each SDU that ends in the AMD PDU according to clause 9.2.2.8;
- otherwise if the AMD PDU is retransmitted~~, the Sender shall~~:
  - use the same value of the "Sequence Number" field as in the original transmission of the AMD PDU
  - if the "Length Indicator" fields needed in the AMD PDU according to subclause 9.2.2.8 has changed due to that a piggybacked STATUS PDU is included in the AMD PDU or a piggybacked STATUS PDU was included in the previous transmission of the AMD PDU;
    - update the "Length Indicator" fields according to 9.2.2.8;
- set ~~The setting of~~ the "Polling bit" to the value~~bit is~~ specified in ~~sub~~clause 11.3.2.1.1.

~~One length indicator field shall be included for each end of an SDU that the PDU includes. The LI fields shall be set as specified in subclause 9.2.2.8.~~

~~How to perform the segmentation and concatenation of SDUs is specified in subclause 11.3.2.1.2.~~

11.3.2.1.1 Setting of the Polling bit

The Sender shall:

- ~~if the "poll prohibit" function is configured; and~~
- ~~if the timer Timer_Poll_Prohibit is active~~
  - ~~consider polling to be prohibited;~~
- if a poll has been triggered by one or several poll triggers (see clause 9.7.1); and
  - if polling is not prohibited, see subclause 9.5;
    - set the "Polling bit" in the AMD PDU header to "1";
- otherwise:
  - set the "Polling bit" in the AMD PDU header to "0";~~The Polling bit shall be set to 1 if any of following conditions are fulfilled except when the poll prohibit function is used and the timer Timer_Poll_Prohibit is active (the different triggers are described in 9.7.1):~~

~~1) Last PDU in buffer is used and the last PDU available for transmission is transmitted.~~

~~2) Last PDU in retransmission buffer is used and the last PDU to be retransmitted is transmitted.~~

~~3) Poll timer is used and timer Timer_Poll has expired.~~

~~4) Every Poll_PDU PDU is used and when VT(PDU)=Poll_PDU.~~

~~5) Every Poll_SDU is used and VT(SDU)=Poll_SDU and the PDU contains the last segment of that SDU.~~

~~6) Window based polling is used, and J≥Poll_Window, where J is defined in subclause 9.6.~~

~~7) Timer based polling is used and Timer_Poll_Periodic has expired.~~

~~8) Poll prohibit shall be used, the timer Timer_Poll_Prohibit has expired and one or several polls were prohibited during the time Timer_Poll_Prohibit was active.~~

11.3.2.1.2 ~~Segmentation and concatenation of SDUs~~Void

~~Upon reception of an SDU, RLC shall segment the SDU to fit into the fixed size of a PDU. The segments are inserted in the data field of a PDU. A length indicator shall be added to each PDU that includes a border of an SDU, i.e. if a PDU does not contain an LI, the SDU continues in the next PDU. The length indicator indicates where the border occurs in the PDU. The data after the indicated border can be either a new SDU, padding or piggybacked information. If padding or piggybacking is added another LI shall be added unless the padding size is one octet for PDUs with 15-bit LIs, see subclauses 9.2.2.8 and 9.2.2.9.~~

11.3.2.~~X~~2 Submission of AMD PDUs to lower layer

If one or more AMD PDUs have been scheduled for transmission or retransmission according to Subclause 11.3.2 the Sender shall:

- not submit any AMD PDUs to lower layer that is not allowed to transmit. AMD PDUs are only allowed to transmit if: ~~according to the following definition. An AMD PDUs is allowed to transmit if:~~
  - The AMD PDU has a "Sequence Number" < VT(MS); or
- ~~-~~The AMD PDU has a "Sequence Number" equal to VT(S)-1
- The AMD PDU is not restricted to be transmitted by the local suspend function, see subclause 9.7.5
- inform the lower layer of the number of AMD PDUs scheduled for transmission or retransmission;
- submit to the lower layer, the requested number of AMD PDUs;
- set the AMD PDU contents according to clause 11.3.2.1
- treat retransmissions with higher priority than AMD PDUs transmitted for the first time;
- update the state variables in clause 9.4 for each AMD PDU submitted to lower layer;
- if the "Polling bit" is set to "1" in any of the AMD PDUs; and
- if the timer Timer_Poll is configured;
  - start the timer Timer_Poll according to subclause 9.5;
- buffer the AMD PDUs that are not submitted to the lower layer according to the discard configuration (see subclause 9.7.3)

11.3.3 Reception of AMD PDU by the ~~receiver~~Receiver

Upon reception of an AMD PDU, the R~~r~~eceiver shall:

- ~~-~~update VR(R), VR(H) and VR(MR) state variables ~~according to the~~for each received AMD PDU (see clause 9.4);~~.~~

~~-~~

i~~I~~f a received AMD PDU includes a "Polling bit" set to "1",~~;~~ or "Missing PDU Indicator" is configured and the Receiver detects that a PDU is missing;

- ~~, if the "detection of missing PDU(s)" is configured and the Receiver detects that a PDU is missing;~~

  - initiate the STATUS PDU transfer procedure~~ shall be initiated.~~;
- reassemble the received AMD PDUs into RLC SDUs;
- if "In-Sequence Delivery" is configured;
  - submit the RLC SDUs in-sequence (i.e. in the same order as the RLC SDUs where originally transmitted by the peer entity) to ~~the upper layer~~upper layers through the AM-SAP
- otherwise if "In-Sequence Delivery" is not configured;
  - submit the RLC SDUs in arbitrary order to ~~the upper layer~~upper layers through the AM-SAP.

~~If the detection of missing PDU(s) shall be used and the receiver detects that a PDU is missing, the receiver shall initiate the STATUS PDU transfer procedure.~~

11.3.4 Abnormal cases

11.3.4.1 Timer_Poll timeout

Upon expiry of the timer Timer_Poll, the ~~sender~~Sender shall:

- if an AMD PDU ~~not previously transmitted has been received from the upper layer~~is available for transmission;
  - transmit an AMD PDU with the "Polling bit" set to "1";
- otherwise if no AMD PDU is available for transmission;
  - retransmit an AMD PDU even if that AMD PDU is not ~~indicated as missing~~negatively acknowledged, with the "Polling bit" set to "1";
- Initiate the acknowledged mode data transfer procedure for the AMD PDU to be transmitted or retransmitted. ~~retransmit the poll. The poll can be retransmitted in either a new PDU or a retransmitted PDU.~~

11.3.4.2 Receiving an AMD PDU outside the receiving window

Upon reception of an AMD PDU with sequence number outside the interval VR(R)≤SN<VR(MR), the ~~receiver~~Receiver shall:

- ~~-~~discard the AMD PDU;

~~. The poll bit shall be~~ - if the "polling bit" in the discarded AMD PDU is set to "1";

  - initiate the STATUS PDU transfer procedure. ~~considered even if a complete PDU is discarded.~~

11.3.4.3 Timer_Discard timeout

11.3.4.3.1 SDU discard with explicit signalling

Upon expiry of the timer Timer_Discard, the ~~sender~~Sender shall:

- ~~-~~initiate the SDU discard with explicit signalling procedure, see subclause 11.6.2. ~~to discard all SDUs up to and including the SDU for which the timer expired.~~

~~.~~In the case where the TFC selection exchange has been initiated by sending the RLC Entity Info parameter to MAC, the UE may wait until after it provides MAC with the requested set of PDUs before discarding the afore-mentioned SDUs.

11.3.4.4 VT(DAT) ≥ MaxDAT

The Sender shall:

- if VT(DAT) ≥ MaxDAT for any AMD PDU;

- if "No_discard after MaxDAT number of retransmissions" is configured:
    - initiate the RLC reset procedure, see subclause 11.4;
- i~~I~~f "SDU discard after MaxDAT number of retransmissions" is configured:~~;used and VT(DAT) ≥ MaxDAT for any PDU~~
    - ~~, the sender shall ~~initiate the SDU discard with explicit signalling procedure, see subclause 11.6.~~. All the SDUs that~~

~~when the SDU discard procedure is initiated, all SDUs that have segments in AMD PDUs with "Ssequence Nnumber"s inside the interval VT(A) ≤ "Sequence Nnumber" SN ≤ X, where X is the value of the "Sequence Nnumber"SN of the AMD PDU with VT(DAT) ≥ MaxDAT shall be discarded.~~

~~If No_discard after MaxDAT number of retransmissions is used, the sender shall initiate the RLC reset procedure when VT(DAT) ≥ MaxDAT.~~

11.3.4.5 Invalid length indicator value

If the "L~~l~~ength I~~i~~ndicator" of an AMD PDU has a value that is larger than the PDU size – RLC header size and is not one of the predefined values listed in the table of subclause 9.2.2.8, the Sender shall:

- discard that AMD PDU;
- ~~shall be discarded and ~~treated the discarded AMD PDU as ~~a ~~missing.~~ PDU.~~

11.3.4.6 Length Indicator value reserved for AMD PDU

Upon ~~reception ~~delivery by the lower layer of an AMD PDU that contains a "Length Indicator" value specified to be reserved for AMD PDUs in this version of the protocol, the ~~receiver~~Receiver shall:

- discard that AMD PDU;
- treat the discarded AMD PDU as missing.~~.~~

11.4 RLC reset procedure

11.4.1 General~~Purpose~~

The RLC reset procedure is used to reset two RLC peer entities, which are operating in acknowledged mode. Figure 11.4 below illustrates the elementary procedure for an RLC reset. ~~The sender can be either the UE or the network and the receiver is either the network or the UE. ~~During the reset procedure the hyper frame numbers (HFN) in UTRAN and UE are synchronised. Two HFNs used for ciphering needs to be synchronised, DL HFN in downlink and UL HFN in uplink. In the reset procedure, the highest UL HFN and DL HFN used by the RLC entity in the transmitting sides, i.e. the HFNs associated with PDUs of SN=VT(S)-1 if at least one data PDU had been transmitted or of SN=0 if no data PDU had been transmitted, are exchanged between UE and UTRAN~~. After the reset procedure is terminated, the UL HFN and DL HFN shall be increased with one in both UE and UTRAN, and the updated HFN values shall be used for the first transmitted and received PDUs after the reset procedure~~.

The RESET PDUs and the RESET ACK PDUs have higher priority than AMD PDUs.

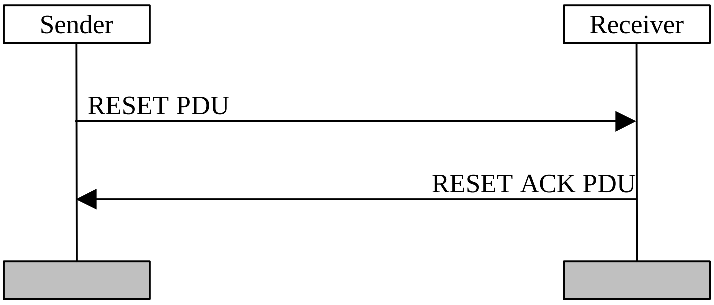


Figure 11.4: RLC reset procedure

11.4.2 Initiation

~~The procedure shall be initiated when a protocol error occurs.~~ The Sender shall:

- if one of the following triggers is detected:
    1) "No_Discard after MaxDAT number of retransmissions" is configured and ~~the number of retransmissions of an AMD PDU~~VT(DAT) ~~comes to~~equals the value MaxDAT (see subclause 9.7.3.4);
    2) ~~The value of~~ VT(MRW) ~~comes to~~equals the value MaxMRW;
    3) A STATUS PDU including "erroneous Sequence Number" is received (see clause 10);
    - ~~The sender sends~~ submit a~~the~~ RESET PDU to the lower layer;~~,~~
    - ~~enters reset pending state when it was in data transfer ready state, and enters reset and suspend state when it was in local suspend state~~
    - ~~. The sender shall~~ start the timer Timer_RST and increase VT(RST) with 1. ~~The RESET PDU shall be transmitted on the DCCH logical channel if the sender is located in the control plane and on the DTCH if it is located in the user plane.~~

~~The RESET PDU has higher priority than data PDUs.~~

When a reset procedure has been initiated it can only be ended upon reception of a RESET ACK PDU with the same RSN value as in the corresponding RESET PDU, ~~i.e.~~or upon request of re-establishment or release from upper layer, a reset procedure is not interrupted by the reception of a RESET PDU from the peer entity.

11.4.2.1 RESET PDU contents to set

The Sender shall:

- set the HFNI field to the currently highest used HFN (DL HFN when the RESET PDU is sent by UTRAN or UL HFN when the RESET PDU is sent by the UE);
- set the RSN field to the sequence number of the RESET PDU. This sequence number is incremented every time a new RESET PDU is transmitted, but not when a RESET PDU is retransmitted.

~~The size of the RESET PDU shall be equal to one of the allowed PDU sizes. The hyper frame number indicator field (HFNI) shall be set equal to the currently highest used HFN (DL HFN when the RESET is sent by UTRAN or UL HFN when the RESET is sent by the UE). The RSN field shall indicate the sequence number of the RESET PDU. This sequence number is incremented every time a new RESET PDU is transmitted, but not when a RESET PDU is retransmitted.~~

11.4.3 Reception of the RESET PDU by the R~~r~~eceiver

Upon reception of a RESET PDU the R~~r~~eceiver shall:

- ~~respond with a RESET ACK PDU~~ submit a RESET ACK PDU to the lower layer;
- ~~. The receiver~~ resets the state variables ~~in~~ described in subclause 9.4 except VT(RST) to their initial values~~,~~;

- ~~-~~stops all the timers ~~in~~described in subclause 9.5 except Timer_RST~~, and ~~;
- resets configurable parameters to their configured values~~. Both the transmitting and receiving sides of the AM RLC entity are reset. ~~;
- discard ~~A~~all RLC PDUs in the receiving side of the AM RLC entity~~ receiver shall be discarded.~~;
- discard all~~The~~ RLC SDUs ~~in the AM RLC transmitter~~ that were transmitted before the reset in the transmitting side of the AM RLC entity~~ shall be discarded~~;~~.~~
- ~~When a RESET PDU is received, the receiver shall~~ set the HFN (DL HFN when the RESET PDU is received in UE or UL HFN when the RESET PDU is received in UTRAN) equal to the HFNI field in the received RESET PDU.
- increase with one the UL HFN and DL HFN, and the updated HFN values shall be used for the first transmitted and received PDUs after the reset procedure

~~The RESET ACK PDU shall be transmitted on the DCCH logical channel if the sender is located in the control plane and on the DTCH if it is located in the user plane.~~

~~The RESET ACK PDU has higher priority than data PDUs.~~

11.4.3.1 RESET ACK PDU contents to set

~~The size of the RESET ACK PDU shall be equal to one of the allowed PDU sizes.~~ The Receiver shall:

- ~~The RSN field shall always be set to the same value as in the corresponding RESET PDU.~~ set ~~T~~the hyper frame number indicator field (HFNI) ~~shall be set equal~~ to the currently highest used HFN (DL HFN when the RESET ACK PDU is sent by UTRAN or UL HFN when the RESET ACK PDU is sent by the UE);~~.~~
- set the RSN field to the same value as in the corresponding received RESET PDU.

11.4.4 Reception of the RESET ACK PDU by the S~~s~~ender

Upon reception of a RESET ACK PDU the Sender shall:

- if the Sender has already transmitted a RESET PDU which has not been yet acknowledged by a RESET ACK PDU:
    - if the received RSN value is the same ~~than~~as the one in the corresponding RESET PDU:
        - set the HFN value (DL HFN when the RESET ACK PDU is received in UE or UL HFN when the RESET ACK PDU is received in UTRAN) to the HFNI field of the received RESET ACK PDU;
        - reset the state variables described in subclause 9.4 to their initial values;
        - stop all the timers described in subclause 9.5;
        - reset configurable parameters to their configured values;
        - discard all RLC PDUs in the receiving side of the AM RLC entity;
        - discard all RLC SDUs that were transmitted before the reset in the transmitting side of the AM RLC entity;
        - increase with one the UL HFN and DL HFN, and the updated HFN values shall be used for the first transmitted and received PDUs after the reset procedure;
    - otherwise (if the received RSN value is not the same ~~than~~as the one in the corresponding RESET PDU):
        - discard the RESET ACK PDU;
- otherwise (if the Sender has not transmitted a RESET PDU which has not been yet acknowledged by a RESET ACK PDU):
    - discard the RESET ACK PDU.

~~When the sender is in reset pending state or reset and suspend state and receives a RESET ACK PDU with the same RSN value as in the corresponding RESET PDU, the Timer_RST shall be stopped and the value of the HFN (DL HFN when the RESET ACK PDU is received in UE or UL HFN when the RESET ACK PDU is received in UTRAN) shall be set equal to the HFNI field in the received RESET ACK PDU. The sender resets the state variables in 9.4 to their initial value and resets configurable parameters to their configured value. Both the transmitting and receiving sides of the AM RLC entity are reset. All RLC PDUs in the AM RLC receiver shall be discarded. The RLC SDUs in the AM RLC transmitter that were transmitted before the reset shall be discarded.~~

~~The sender shall enter data transfer ready state if it was in reset pending state and enter local suspend state if it was in reset and suspend state.~~

~~Upon reception of a RESET ACK PDU with a different RSN value as in the corresponding RESET PDU the RESET ACK PDU is discarded.~~

~~Upon reception of a RESET ACK PDU in data transfer ready state or local suspend state, the RESET ACK PDU is discarded.~~

11.4.5 Abnormal cases

11.4.5.1 Timer_RST timeout

Upon expiry of Timer_RST the S~~s~~ender shall:

- ~~-~~ retransmit the RESET PDU and increase VT(RST) with 1. In the retransmitted RESET PDU the value of the RSN field shall not be incremented.

11.4.5.2 Unrecoverable error (VT(RST) ≥ MaxRST)

The Sender shall:

- ~~I~~if VT(RST) becomes larger than or equal to MaxRST:
    - ~~,~~ indicate unrecoverable error ~~shall be indicated~~ to upper layers.

11.4.5.3 Reception of the RESET PDU by the S~~s~~ender

Upon reception of a RESET PDU ~~in acknowledged data READY state, RESET PENDING state, LOCAL SUSPEND state or RESET AND SUSPEND state~~, the Ssender shall:

- submit a RESET ACK PDU to the lower layer;
- reset the state variables described in subclause 9.4 except VT(RST) to their initial values;
- stop all the timers described in subclause 9.5 except Timer_RST;
- reset configurable parameters to their configured values;
- discard all RLC PDUs in the receiving side of the AM RLC entity;
- discard all RLC SDUs that were transmitted before the reset in the transmitting side of the AM RLC entity;
- set the HFN (DL HFN when the RESET PDU is received in UE or UL HFN when the RESET PDU is received in UTRAN) equal to the HFNI field in the received RESET PDU;
- increase with one the UL HFN and DL HFN, and the updated HFN values shall be used for the first transmitted and received PDUs after the reset procedure.
- ~~stay in its current state.~~

~~respond with a RESET ACK PDU. The sender resets the state variables in 9.4 to their initial value, resets configurable parameters to their configured value. However, VT(RST) and Timer_RST are not reset. Both the transmitting and receiving sides of the AM RLC entity are reset. All RLC PDUs in the AM RLC receiver shall be discarded. The RLC SDUs in the AM RLC transmitter that were transmitted before the reset shall be discarded. The hyper frame number,~~

~~HFN (DL HFN when the RESET is received in UE or UL HFN when the RESET is received in UTRAN) is set equal to the HFNI field in the received RESET PDU. The sender shall stay in its current state. The sender shall enter data transfer ready state or local suspend state only upon reception of a RESET ACK PDU with the same RSN value as in the corresponding RESET PDU when it is in reset pending state or reset and suspend state respectively.~~

11.5 STATUS report transfer procedure

11.5.1 ~~Purpose~~General

The status report transfer procedure is used for transferring of status information between two RLC peer entities, which are operating in acknowledged mode. Figure 11.5 below illustrates the elementary procedure for status report transfer. A status report consists of one or several STATUS PDUs.~~ The receiver is the receiver of AMD PDUs and it is either the UE or the network and the sender is~~~~Information included in a status report is (with the exception of MRW SUFI) sent from the~~~~ the sender of AMD PDUs and it is either the network or the UE.~~~~Receiver's receiving side to the Sender's transmitting side (see figure 4.4).~~

~~If the RLC AM entity is assigned two logical channels on configuration, one logical channel shall be for AMD PDU transfer and the other for Control PDU transfer.~~In case two logical channels are configured in the uplink, control PDUs are transmitted on the second logical channel. In case two logical channels are configured in the downlink, Control PDUs can be transmitted on any of the two logical channels.

The STATUS PDUs have higher priority than AMD PDUs.

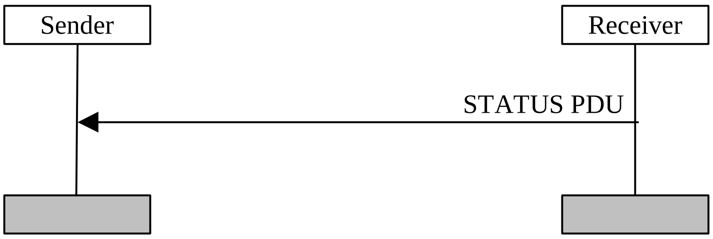


Figure 11.5: Status report transfer procedure

11.5.2 Initiation

The Receiver shall:

~~The receiver in any~~- if one of the following ~~cases initiates this procedure:~~triggers is detected:

1) The ~~poll bit~~"Polling Bit" in a received AMD PDU is set to ~~1.~~"1";

2) ~~Detection of missing PDUs is used and a missing PDU is detected.~~"~~Detection of m~~Missing PDU~~(s)~~ Indicator" is configured and a missing AMD PDU is detected;

3) The ~~timer based STATUS transfer is used~~"Timer based STATUS transfer" is configured and the timer Timer_Status_Periodic has ~~expired.~~expired~~ and periodical status blocking is not configured~~;

- act on the trigger as specified in subclause 9.7.2.~~;~~

~~The receiver shall transmit a status report on the DCCH logical channel if the receiver is located in the control plane and on the DTCH if it is located in the user plane. Separate logical channels can be assigned for AMD PDU transfer and for Control PDU transfer.~~

~~The STATUS PDUs have higher priority than data PDUs.~~

~~There are two functions that can prohibit the receiver from sending a status report. If any of following conditions are fulfilled the sending of the status report shall be delayed, even if any of the conditions above are fulfilled:~~

~~1) STATUS prohibit is used and the timer Timer_Status_Prohibit is active.~~

- ~~The status report shall be transmitted after the Timer_Status_Prohibit has expired. The receiver shall send only one status report, even if there are several triggers when the timer is active. The rules for when the timer Timer_status_Prohibit is active are defined in subclause 9.5.~~

~~2) The EPC mechanism is used and the timer Timer_EPC is active or VR(EP) is counting down.~~

- ~~The status report shall be transmitted after the VR(EP) has reached 0. The receiver send only one status report, even if there are several triggers when the timer is active or the counter is counting down. The rules for when the timer Timer_EPC is active are defined in subclause 9.5.~~

~~If the timer based STATUS transfer shall be used and the Timer_Status_Periodic has expired it shall be restarted.~~

~~-If the EPC mechanism shall be used the timer Timer_EPC shall be started and the VR(EP) shall be set equal to the number PDUs requested to be retransmitted.~~

11.5.2.1 Piggybacked STATUS PDU

The Receiver may:

- if STATUS PDU(s) to be sent fit into padding octets in AMD PDU(s) to be sent:
    - ~~It is possible to~~ piggyback a STATUS PDU on the AMD PDU to be sent.

~~an AMD PDU. If a PDU includes padding, a piggybacked STATUS PDU can be inserted instead of the padding. The sending~~Submission of a piggybacked STATUS PDU in an AMD PDU to the lower layer follows the same rules as~~the sending of~~ an ordinary STATUS PDU.

11.5.2.2 STATUS PDU contents to set

On triggering of a status report ~~T~~the Receiver shall:

- if neither the "STATUS prohibit" nor "EPC mechanism" are active:
    - include negative acknowledgements for all AMD PDUs detected as missing;
    - include acknowledgements for all AMD PDUs received up to at least VR(R);
- if "SDU discard with explicit signalling" procedure has been initiated since the last status report was sent:
    - optionally include one MRW SUFI as specified in subclause 11.6.2.2;
- if the MRW SUFI was received in the last status report received:
    - include one MRW_ACK SUFI as specified in subclause 11.6.3.2;
- if the Sender’s transmission window is to be updated:
    - include the WINDOW SUFI;
- if all SUFIs can be accommodated in one STATUS PDU:
    - construct the status report using one STATUS PDU, using one of the allowed PDU sizes;
    - if the SUFIs included do not fill the entire STATUS PDU:

        ~~The size of~~- terminate the STATUS PDU with the ACK or NO_MORE SUFI;

        - use padding in the remainder of the STATUS PDU;
    - otherwise (SUFIs included fill the entire STATUS PDU):
        - ACK or NO_MORE SUFIs need not be included in that STATUS PDU;
- otherwise (the status report is segmented):

~~shall be equal to~~- construct STATUS PDUs including only complete SUFIs using one of the allowed PDU sizes. The ~~information that needs to be transmitted in a status report can be split into several~~set of STATUS PDUs ~~if one STATUS PDU does not~~shall accommodate all the ~~information. A SUFI cannot be split into several STATUS PDUs.~~SUFIs to form the complete status report. Indication of the same AMD PDU shall not be given in more than one STATUS PDU of a ~~STATUS~~status report, but the ACK SUFI can be present in more than one STATUS PDU of a status ~~report.~~report;

- if any STATUS PDU constructed is not entirely filled with SUFIs:
    - terminate that STATUS PDU with the ACK or NO_MORE SUFI;
    - use padding in the remainder of that STATUS PDU;
- otherwise (SUFIs included fill the entire STATUS PDU):
    - ACK or NO_MORE SUFIs ~~need~~should not be included in that STATUS PDU.

Which SUFI fields to use is implementation dependent~~, but the status report shall include information about PDUs that have been received and information about all PDUs detected as missing~~. Bitmap SUFI is used to indicate both received and/or missing AMD PDUs. List SUFI and/or Relative List SUFI are used to indicate missing AMD PDUs only. Acknowledgement SUFI is used to indicate the received AMD PDUs. (For SUFI details see 9.2.2.11.) No information shall be given for AMD PDUs with SN≥VR(H), i.e. AMD PDUs that have not yet reached the ~~receiver.~~Receiver.

11.5.2.3~~X~~ Submission of STATUS PDUs to the lower layer

The Receiver shall:

- inform the lower layer of the STATUS PDUs scheduled for transmission;
- submit to the lower layer, the requested number of PDUs (STATUS PDUs, piggybacked AMD / STATUS PDUs and optionally AMD PDUs, see also subclause 11.3.2.2~~X~~).
- if "Timer based STATUS transfer" is configured and the timer Timer_Status_Periodic has expired:
    - restart the timer Timer_Status_Periodic according to subclause 9.5 f);
- if the "EPC mechanism" is configured:
    - start the timer Timer_EPC according to subclause 9.5 c), and set VR(EP) equal to the number of AMD PDUs requested to be retransmitted;
- if the STATUS PDU includes the MRW SUFI:
    - start the timer Timer_MRW according to subclause 9.5 i).~~;~~

~~Padding shall be inserted if the SUFI fields do not fill an entire STATUS PDU. If the PDU contains padding the last SUFI field shall be either an ACK SUFI or a NO_MORE SUFI. If there is no padding in the STATUS PDU, NO_MORE SUFI or ACK SUFI does not need to be included in the STATUS PDU.~~

11.5.3 Reception of the STATUS PDU by the ~~sender~~Sender

~~The sender shall upon~~Upon reception of the STATUS PDU / piggybacked STATUS PDU, the Sender shall:

~~PDU/piggybacked STATUS PDU update~~- update the state variables VT(A) and VT(MS) according to the received STATUS PDU/piggybacked STATUS ~~PDU.~~PDU;

~~If~~- if the STATUS PDU includes negatively acknowledged AMD PDUs:

~~PDUs, the~~- the acknowledged data transfer procedure shall be initiated and these AMD PDUs shall be retransmitted. Retransmitted AMD PDUs shall have higher priority than AMD PDUs to be transmitted for the first time;

~~If a~~- if an AMD PDU is ~~indicated as missing~~negatively acknowledged more than once in a STATUS PDU:

~~PDU, the~~- the AMD PDU shall be retransmitted only once. ~~Retransmitted PDUs have higher priority than new PDUs.~~

- if the STATUS PDU includes the MRW SUFI:
  - take the actions specified in subclause 11.6.3;
- if the STATUS PDU includes the MRW_ACK SUFI:
  - take the actions specified in subclause 11.6.4;
- if the STATUS PDU includes the WINDOW SUFI:
  - update the current transmitter window size, VT(WS).

11.5.4 Abnormal cases

11.5.4.1 VR(EP) ~~reache~~equals zero and the requested AMD PDUs have not been received

If the EPC mechanism is ~~used~~configured and VR(EP) ~~has reached~~equals zero and not all AMD PDUs requested for retransmission have been received, the ~~receiver~~Receiver shall:

- Retransmit the status report. The retransmitted status report may contain new or different SUFI fields in order to indicate that some previously lost AMD PDUs have been received and that some ~~new~~additional AMD PDUs have been lost.

11.6 SDU discard with explicit signalling procedure

11.6.1 General~~Purpose~~

The SDU discard with explicit signalling procedure is used for discarding SDUs and transferring the discard information between two peer entities, which are operating in acknowledged mode. The Sender shall discard an SDU that has not been successfully transmitted for a period of time or for a number of retransmissions, and send a Move Receiving Window (MRW) SUFI to the Receiver. According to the MRW SUFI, the Receiver shall discard AMD PDUs carrying that SDU and update the receiving window. Figure 11.6 below illustrates the elementary procedure for SDU discard with explicit signalling. ~~An SDU can be discarded with explicit signalling when MaxDAT number of retransmissions is reached or the transmission time exceeds a predefined value (Timer_Discard) for an SDU in acknowledged mode RLC. Move Receiving Window (MRW) command is sent to the receiver so that AMD PDUs carrying that SDU are discarded in the receiver and the receiver window is updated accordingly. Note that when the concatenation function is active, PDUs carrying segments of other SDUs that have not timed out shall not be discarded. If Send MRW is not configured and no segments of an SDU were submitted to a lower layer, the SDU is simply discarded in the transmitter without notification to the receiver. If Send MRW is configured, a Move Receiving Window request shall be sent to the receiver even if no segments of the SDU were submitted to a lower layer. The Send MRW is used when the AM RLC entity is connected to a PDCP layer which supports lossless SRNS relocation.~~

~~The MRW command is defined as a super-field in the RLC STATUS PDU, and can be piggybacked to status information of transmissions in the opposite direction.~~

~~Figure 11.6 below illustrates the elementary procedure for SDU discard with explicit signalling. The sender is the sender of AMD PDUs and it is either the UE or the network and the receiver is the receiver of AMD PDUs and it is either the network or the UE.~~

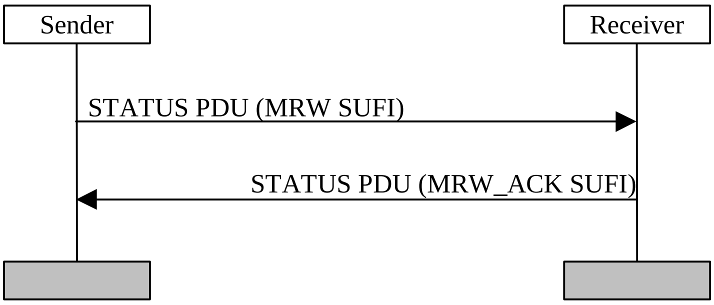


Figure 11.6: SDU discard with explicit signalling

11.6.2 Initiation

The Sender shall initiate the SDU discard with explicit signalling procedure if one of the following triggers is detected: ~~This procedure is initiated by the sender when any of the following conditions is fulfilled:~~

- ~~1)~~ "Timer based SDU discard with explicit signalling" is configured~~used~~, Timer_Discard expires for an SDU, and one or more segments of the SDU have been submitted to a lower layer;~~.~~
- ~~2)~~ "Timer based SDU discard with explicit signalling" is configured~~used~~, Timer_Discard expires for an SDU, and "Send MRW" is configured;~~.~~
- ~~3)~~ "SDU discard after MaxDAT number of retransmissions" is configured~~used~~, and MaxDAT number of retransmissions is reached (i.e. VT(DAT) ≥ MaxDAT) for an AMD ~~S~~PDU.

Upon initiation of the SDU discard with explicit signalling procedure, the Sender shall:

- if "Timer based SDU discard with explicit signalling" is configured:
    - discard all SDUs up to and including the SDU for which the timer Timer_Discard expired;
    - not discard AMD PDUs carrying segments of other SDUs whose timers have not expired;
- if "SDU discard after MaxDAT number of retransmissions" is configured:
    - discard all SDUs that have segments in AMD PDUs with SN inside the interval VT(A) ≤ SN ≤ X, where X is the value of the SN of the AMD PDU with VT(DAT) ≥ MaxDAT;
- if more than 15 discarded SDUs are to be informed to the Receiver:
    - assemble an MRW SUFI with the discard information of the first 15 SDUs. The discard information of the rest SDUs shall be included in another MRW SUFI which shall be sent by the next SDU discard with explicit signalling procedure (after the current SDU discard with explicit signalling procedure is terminated);
- otherwise (less than or equal to 15 discarded SDUs are to be informed to the Receiver):
    - assemble an MRW SUFI with the discard information of the SDUs;
- include the MRW SUFI in the next STATUS PDU/piggybacked STATUS PDU to be transmitted, according to subclause 11.5.2;
- if SN_MRW$_{LENGTH}$ in the MRW SUFI >VT(S):
    - update VT(S) to SN_MRW$_{LENGTH}$;
- start a timer Timer_MRW according to subclause 9.5.

~~The sender shall discard all PDUs that contain segments of the associated SDUs. If the concatenation function is active, PDUs carrying segments of other SDUs that have not timed out shall not be discarded. VT(A) shall be updated when the procedure is terminated, and VT(S) shall be updated when a new MRW SUFI which includes SN_MRW$_{LENGTH}$ >VT(S) is transmitted.~~

~~The sender shall transmit a status report on the DCCH logical channel if the sender is located in the control plane and on the DTCH if it is located in the user plane.~~

~~This status report is sent even if the 'STATUS prohibit' is used and the timer 'Timer_Status_Prohibit' is active, or if the 'EPC mechanism' is used and the timer 'Timer_EPC' is active or 'VR(EP)' is counting down.~~

~~The STATUS PDUs have higher priority than data PDUs.~~

~~The sender shall start timer Timer_MRW.~~ If a new SDU discard with explicit signalling procedure is triggered when the timer Timer_MRW is active~~running~~, no new MRW SUFIs shall be sent before the current SDU discard with explicit signalling procedure is terminated by one of the termination criteria specified in subclause 11.6.4.

11.6.2.1 ~~Piggybacked STATUS PDU~~Void

~~It is possible to piggyback a STATUS PDU on an AMD PDU. If a PDU includes padding a piggybacked STATUS PDU can be inserted instead of the padding.~~

11.6.2.2 STATUS PDU contents to set

~~The size of the STATUS PDU shall be equal to one of the allowed PDU sizes. The discard information shall not be split into several MRW SUFIs. If the discard information cannot be fit into one MRW SUFI, another SDU discard with explicit signalling procedure shall be initiated after the current procedure is terminated.~~

~~The status report shall include the MRW SUFI, other SUFI fields can be used additionally. MRW SUFI shall convey information about the discarded SDU(s) to the receiver.~~The Sender shall:

- if "Send MRW" is configured:
  - if the last discarded SDU ended in an AMD PDU, and its LI is present in the same AMD PDU, and no new SDU is present inside this AMD PDU:
    - set the last SN_MRW$_i$ field in the MRW SUFI to 1 + SN of the AMD PDU which contains the LI of the last discarded SDU;
    - set the N$_{LENGTH}$ field in the MRW SUFI to "0000";
  - otherwise:
    - set the last SN_MRW$_i$ field in the MRW SUFI to the SN of the AMD PDU which contains the LI of the last discarded SDU;
    - set the N$_{LENGTH}$ field in the MRW SUFI so that the last data octet to be discarded in the Receiver shall be the octet indicated by the N$_{LENGTH}$:th LI field of the AMD PDU which contains the LI of the last discarded SDU;
  - set each of the other SN_MRW$_i$ fields in the MRW SUFI to the SN of the AMD PDU which contains the LI of the i:th discarded SDU;
- otherwise ("Send MRW" is not configured):
  - if the last SDU to be discarded in the Receiver ended in an AMD PDU, and its LI is present in the same AMD PDU, and no new SDU is present inside this AMD PDU:
    - set the last SN_MRW$_i$ field in the MRW SUFI to 1 + SN of the AMD PDU which contains the LI of the last SDU to be discarded in the Receiver;
    - set the N$_{LENGTH}$ field in the MRW SUFI to "0000";
  - otherwise:
    - set the last SN_MRW$_i$ field in the MRW SUFI to the SN of the AMD PDU which contains the LI of the last SDU to be discarded in the Receiver;

        - set the N$_{LENGTH}$ field in the MRW SUFI so that the last data octet to be discarded in the Receiver shall be the octet indicated by the N$_{LENGTH}$:th LI field of the AMD PDU which contains the LI of the last SDU to be discarded in the Receiver;

    - optionally set each of the other SN_MRW$_{i}$ fields in the MRW SUFI to the SN of the AMD PDU which contains the LI of the i:th SDU to be discarded in the Receiver;

- if the MRW SUFI contains only one SN_MRW$_{i}$ field and the value of SN_MRW$_{i}$ field ≥ VT(A)+Configured_Tx_Window_Size:

    - set the LENGTH field in the MRW SUFI to "0000";

- otherwise:

    - set the LENGTH field in the MRW SUFI to the number of SN_MRW$_{i}$ fields in the same MRW SUFI. In this case, SN_MRW$_{1}$ shall be in the interval VT(A) ≤ SN_MRW$_{1}$ < VT(A)+Configured_Tx_Window_Size;

- include the MRW SUFI in the next STATUS PDU/piggybacked STATUS PDU to be transmitted, according to subclause 11.5.2;~~When Send MRW is configured, the MRW SUFI shall contain the information about each discarded SDU (see subclause 9.2.2.11.8). In order to discard a single SDU that ends in a PDU with SN≥ VT(A)+Configured_Tx_Window_Size, the LENGTH field in the MRW SUFI shall be set to "0000". If more than one SDU are discarded with the same MRW SUFI, at least the first discarded SDU must end (i.e. the LI must be located) in a PDU with SN in the interval VT(A)≤ SN <VT(A)+Configured_Tx_Window_Size. Otherwise, multiple SDU discard with explicit signalling procedures need to be performed in order to signal the set of discarded SDUs.~~

~~When Send MRW is not configured, the MRW SUFI shall contain the information about the last SDU to be discarded in the receiver. The information about the other SDUs to be discarded in the receiver may optionally be contained in the MRW SUFI (see subclause 9.2.2.11.8). If the MRW SUFI contains only the information about the last SDU to be discarded in the receiver and if this SDU ends in a PDU with SN≥ VT(A)+Configured_Tx_Window_Size, the LENGTH field in the MRW SUFI shall be set to "0000". If the MRW SUFI contains information about more than one discarded SDU, at least the first discarded SDU must end (i.e. the LI must be located) in a PDU with SN in the interval VT(A)≤ SN <VT(A)+Configured_Tx_Window_Size.~~

~~Padding shall be inserted if the SUFI fields do not fill the entire STATUS PDU. If the STATUS PDU contains padding the last SUFI field shall be either an ACK SUFI or a NO_MORE SUFI. If there is no padding in the STATUS PDU, NO_MORE SUFI or ACK SUFI does not need to be included in the STATUS PDU.~~

11.6.3 Reception of the STATUS PDU by the ~~receiver~~Receiver

~~The receiver shall u~~Upon reception of the STATUS PDU/piggybacked STATUS PDU containing an MRW SUFI, the Receiver shall:~~discard PDUs and update the state variables VR(R), VR(H) and VR(MR) according to the received STATUS PDU/piggybacked STATUS PDU.~~

- if the LENGTH field in the received MRW SUFI is "0000":

    - consider SN_MRW$_{1}$ to be above or equal to VR(R);

- otherwise:

    - consider SN_MRW$_{1}$ to be less than VR(MR);

- consider all the SN_MRW$_{i}$s other than SN_MRW$_{1}$ to be in sequential order within the list and sequentially above or equal to SN_MRW$_{i-1}$.

- discard AMD PDUs up to and including the PDU with sequence number SN_MRW$_{LENGTH}$–1;

- if the N$_{LENGTH}$ field in the received MRW SUFI is "0000":

    - reassemble from the first data octet of the AMD PDU with sequence number SN_MRW$_{LENGTH}$ after the discard;

- otherwise:

  - discard further the data octets in the AMD PDU with sequence number SN_MRW$_{LENGTH}$ up to and including the octet indicated by the N$_{LENGTH}$:th LI field of the PDU with sequence number SN_MRW$_{LENGTH}$;

  - reassemble from the succeeding data octet in the AMD PDU with sequence number SN_MRW$_{LENGTH}$ after the discard;

- ~~Additionally, when~~if "Send MRW" is configured~~,~~:

  - ~~the receiver shall~~ inform ~~the upper layer~~upper layers about all of the discarded SDUs that were not previously delivered to upper layer or discarded by other MRW SUFIs~~.~~;

- update the state variables VR(R), VR(H) and VR(MR) according to the received STATUS PDU/piggybacked STATUS PDU;

- include an MRW_ACK SUFI in the next STATUS PDU/piggybacked STATUS PDU to be transmitted, according to subclause 11.5.2; ~~The receiver shall initiate the transmission of a status report containing an MRW_ACK SUFI.~~

11.6.3.1~~X~~ STATUS PDU contents to set

The Receiver shall:

- ~~In the MRW_ACK SUFI,~~set the SN_ACK field in the MRW_ACK SUFI ~~shall be set~~ to the new value of VR(R), updated after reception of the MRW SUFI~~.~~;

- if the SN_ACK field in the MRW_ACK SUFI is set equal to the SN_MRW$_{LENGTH}$ field in the received MRW SUFI:

  - set ~~T~~the N field in the MRW_ACK SUFI ~~shall be set~~ to the NLENGTH field in the received MRW SUFI;

- otherwise:

  - ~~if the SN_ACK field is equal to SN_MRWLENGTH. Otherwise~~set the N field in the MRW_ACK SUFI ~~shall be set~~ to "0000"~~0.~~;

  - include the MRW_ACK SUFI in the next STATUS PDU/piggybacked STATUS PDU to be transmitted, according to subclause 11.5.2;~~The last discarded data octet is the octet indicated by the N$_{LENGTH}$:th LI field of the PDU with sequence number SN_MRW$_{LENGTH}$ and the succeeding data octet is the first data octet to be reassembled after the discard. When N$_{LENGTH}$ = 0, the first data octet of the PDU with sequence number SN_MRW$_{LENGTH}$ is the first data octet to be reassembled after the discard.~~

~~If LENGTH="0000", the sequence number SN_MRW$_1$ is considered to be above or equal to VR(R). Else, the sequence number SN_MRW$_1$ is considered to be less than VR(MR). All the SN_MRW$_i$s other than SN_MRW$_1$ are considered to be in sequential order within the list and sequentially above or equal to SN_MRW$_{i-1}$.~~

11.6.4 Termination

The Sender shall terminate ~~T~~the SDU discard with explicit signalling procedure ~~is terminated in the sender~~ if one of the following criteria is fulfilled~~in the following cases~~:

- ~~1. On the reception of a~~A STATUS PDU/piggybacked STATUS PDU ~~which~~ containings an MRW_ACK SUFI is received, and~~with~~the SN_ACK field in the received MRW_ACK SUFI > the SN_MRW$_{LENGTH}$ field in the transmitted MRW_SUFI, and ~~with~~ the N field in the received MRW_ACK SUFI is set equal to "0000"~~zero.~~;

~~2. On the reception of a STATUS PDU which contains an ACK SUFI indicating VR(R) > SN_MRW$_{LENGTH}$~~

- ~~3. On reception of a~~A STATUS PDU/piggybacked STATUS PDU ~~which~~ containings an MRW_ACK SUFI is received, and~~with~~the SN_ACK field in the received MRW_ACK SUFI = the SN_MRW$_{LENGTH}$ field in the transmitted MRW_SUFI, and ~~with~~ the N field in the received MRW_ACK SUFI is set equal to the N$_{LENGTH}$ field~~indicated~~ in the transmitted MRW SUFI~~.~~;

- A STATUS PDU/piggybacked STATUS PDU containing an ACK SUFI is received, and the LSN field in the received ACK SUFI > the SN_MRW$_{LENGTH}$ field in the transmitted MRW SUFI.

~~If one of the termination criteria above is fulfilled,~~ Upon termination of the SDU discard with explicit signalling procedure, the Sender shall:

- stop the timer Timer_MRW;
- update VT(A) and VT(MS) according to the received STATUS PDU/piggybacked STATUS PDU.~~ shall be stopped and the discard procedure is terminated.~~

The Sender shall not confirm to upper layers ~~T~~the SDUs that are requested to be discarded~~ shall not be confirmed to upper layers~~.

~~When VT(MRW) reaches MaxMRW, the procedure is terminated and an RLC reset shall be performed.~~

11.6.5 Expiration of timer Timer_MRW

If Timer_MRW expires before the SDU discard with explicit signalling procedure is terminated, the Sender shall:

- retransmit the MRW SUFI. The retransmitted ~~shall be retransmitted, VT(MRW) is incremented by one and Timer_MRW restarted.~~ MRW SUFI shall be exactly the same as previously transmitted even though some new SDUs would have been discarded during the running of the Timer_MRW~~.~~;
- increment VT(MRW) by one;
- restart the timer Timer_MRW.~~ If the retransmitted STATUS PDU contains other SUFIs than the MRW SUFI, the status information indicated by these SUFIs shall be updated.~~

11.6.6 Abnormal cases

11.6.6.1 Reception of ~~O~~obsolete/corrupted MRW SUFI~~command~~ by the Receiver

If the received MRW SUFI~~command~~ contains outdated information about the receiving~~er~~ window (receiving~~er~~ window already moved further than MRW SUFI~~command~~ is indicating), the Receiver shall:

- discard the MRW SUFI;~~command shall be discarded and~~
- set the SN_ACK field in the MRW_ACK SUFI to the current value of VR(R);
- set the N field in the MRW_ACK SUFI to "0000";
- include the MRW_ACK SUFI in the next STATUS PDU/piggybacked STATUS PDU to be transmitted, according to subclause 11.5.2; ~~a status report containing SUFI MRW_ACK shall be transmitted indicating the value of VR(R) and the N field shall be set to zero.~~

11.6.6.2 VT(MRW) equals MaxMRW

If the number of retransmission of an MRW SUFI~~command~~ (i.e. VT(MRW)) ~~reache~~equals MaxMRW, the Sender shall:

- terminate the SDU discard with explicit signalling procedure;
- stop the timer Timer_MRW;
- deliver an error indication ~~shall be passed~~ to upper layers;~~ and~~
- initiate the RLC RESET procedure (see clause 11.4).~~shall be performed.~~

11.6.6.3 Reception of obsolete/corrupted MRW_ACK SUFI by the Sender

The Sender shall discard ~~T~~the received MRW_ACK SUFI if one of the following cases occurs:~~shall be discarded in the following cases.~~

- ~~1. If~~The timer Timer_MRW is not active~~.~~;

- ~~2. If t~~The SN_ACK field in the received MRW_ACK SUFI < the SN_MRW$_{LENGTH}$ field in the transmitted MRW SUFI~~.~~;

- ~~3. If t~~The SN_ACK field in the received MRW_ACK SUFI = ~~is equal to~~ the SN_MRW$_{LENGTH}$ field in the transmitted MRW SUFI, and the N field in the received MRW_ACK SUFI is not equal to the N$_{LENGTH}$ field in the transmitted MRW SUFI;

- ~~4. If t~~The SN_ACK field in the received MRW_ACK SUFI > the SN_MRW$_{LENGTH}$ field in the transmitted MRW SUFI, and the N field in the received MRW_ACK SUFI is not equal to "0000"~~zero~~.

11.7 Void

11.8 Void

3GPP TSG-RAN WG2 Meeting #23
Finland, Helsinki, 27-31 August 2001

Tdoc R2-012193

CR-Form-v4

CHANGE REQUEST

⌘	**25.322**	**CR**	**142**	⌘	rev		⌘	Current version: **4.1.0** ⌘

For ***HELP*** *on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: ⌘ (U)SIM □ ME/UE ☒ Radio Access Network ☒ Core Network □

Title:	⌘	General clarifications		
Source:	⌘	TSG-RAN WG2		
Work item code:	⌘	TEI	***Date:*** ⌘	31 August 2001
Category:	⌘	**F**	***Release:*** ⌘	REL-4
		Use one of the following categories: ***F*** *(correction)* ***A*** *(corresponds to a correction in an earlier release)* ***B*** *(addition of feature),* ***C*** *(functional modification of feature)* ***D*** *(editorial modification)* Detailed explanations of the above categories can be found in 3GPP TR 21.900.		*Use one of the following releases:* *2 (GSM Phase 2)* *R96 (Release 1996)* *R97 (Release 1997)* *R98 (Release 1998)* *R99 (Release 1999)* *REL-4 (Release 4)* *REL-5 (Release 5)*

Reason for change:	⌘	Clarify the whole specification.
Summary of change:	⌘	All clauses and subclauses have been modified. Isolated impact analysis: • The CR contains only editorial modifications and clarifications.
Consequences if not approved:	⌘	.

Clauses affected:	⌘	1, 2, 3, 3.1, 3.2, 4.1, 4.2, 4.2.1, 4.2.1.1, 4.2.1.1.1 (new), 4.2.1.1.2 (new), 4.2.1.2, 4.2.1.2.1 (new), 4.2.1.2.2 (new), 4.2.1.3, 4.2.1.3.1 (new), 4.2.1.3.2 (new), 5, 6, 6.1, 6.2, 7, 8, 8.1, 8.2, 9.1, 9.1.1, 9.1.2, 9.2, 9.2.1.1, 9.2.1.2, 9.2.1.3, 9.2.1.4, 9.2.1.5, 9.2.1.6, 9.2.1.7, 9.2.2.1, 9.2.2.3, 9.2.24, 9.2.25, 9.2.2.6, 9.2.2.7, 9.2.2.8, 9.2.2.9, 9.2.2.10, 9.2.2.11, 9.2.2.11.2, 9.2.2.11.3, 9.2.2.11.4, 9.2.2.11.5, 9.2.2.11.6, 9.2.2.11.7, 9.2.2.11.8, 9.2.2.12, 9.2.2.13, 9.3, 9.3.1, 9.3.1.1, 9.3.1.2, 9.3.2, 9.3.2.1, 9.3.2.2, 9.3.2.3, 9.3.3, 9.3.3.1, 9.3.3.2, 9.3.3.3, 9.3.3.4, 9.3.3.5, 9.4, 9.5, 9.6, 9.7, 9.7.1, 9.7.2, 9.7.3, 9.7.3.1, 9.7.3.2, 9.7.3.3, 9.7.3.4, 9.7.3.5, 9.7.4, 9.7.5, 9.7.6, 9.7.7, 9.7.8, 10, 10.1 (new), 10.2 (new), 10.3 (new), 11, 11.1.1, 11.1.2, 11.1.2.1, 11.1.2.2 (new), 11.1.3, 11.1.4.2, 11.2.1, 11.2.2, 11.2.2.1, 11.2.2.2 (new), 11.2.3, 11.2.4.1, 11.2.4.2, 11.2.4.3, 11.3.1, 11.3.2, 11.3.2.1, 11.3.2.1.1, 11.3.2.1.2, 11.3.2.2 (new), 11.3.3, 11.3.4.1, 11.3.4.2, 11.3.4.3.1, 11.3.4.4, 11.3.4.5, 11.3.4.6, 11.4.1, 11.4.2, 11.4.2.1, 11.4.3, 11.4.3.1, 11.4.4, 11.4.5.1, 11.4.5.2, 11.4.5.3, 11.5.1, 11.5.2, 11.5.2.1, 11.5.2.2, 11.5.2.3 (new), 11.5.3, 11.5.4.1, 11.6.1, 11.6.2, 11.6.2.1, 11.6.2.2, 11.6.3, 11.6.3.1 (new), 11.6.4, 11.6.5, 11.6.6.1, 11.6.6.2, 11.6.6.3		
Other specs affected:	⌘	☒ Other core specifications □ Test specifications	⌘	25.322 v3.7.0, CR 141r1

	□	O&M Specifications	
Other comments: ⌘			

How to create CRs using this form:

Comprehensive information and tips about how to create CRs can be found at: http://www.3gpp.org/3G_Specs/CRs.htm. Below is a brief summary:

1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.

2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be downloaded from the 3GPP server under ftp://ftp.3gpp.org/specs/ For the latest version, look for the directory name with the latest date e.g. 2001-03 contains the specifications resulting from the March 2001 TSG meetings.

3) With "track changes" disabled, paste the entire CR form (use CTRL-A to select it) into the specification just in front of the clause containing the first piece of changed text. Delete those parts of the specification which are not relevant to the change request.

Foreword

This Technical Specification (TS) 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.

1 Scope

The present document specifies the Radio Link Control protocol for the UE-UTRAN radio interface.

Features for the current Release:

- Transparent mode.
- Unacknowledged mode.
- Acknowledged mode.

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 TS 25.401: "UTRAN Overall Description".

[2] 3GPP TR 25.990: "Vocabulary for the UTRAN".

[3] 3GPP TS 25.301: "Radio Interface Protocol Architecture".

[4] 3GPP TS 25.302: "Services Provided by the Physical Layer".

[5] 3GPP TS 25.303: "Interlayer Procedures in Connected Mode".

[6] 3GPP TS 25.304: "UE Procedures in Idle Mode and Procedures for Cell Reselection in Connected Mode".

[7] 3GPP TS 25.321: "MAC Protocol Specification".

[8] 3GPP TS 25.331: "RRC Protocol Specification".

[9] 3GPP TS 33.102: "Security Architecture".

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the terms and definitions given in [2] apply.

3.2 Abbreviations

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

AM	Acknowledged Mode
AMD	Acknowledged Mode Data
ARQ	Automatic Repeat Request

BCCH	Broadcast Control Channel
BCH	Broadcast Channel
C-	Control-
CCCH	Common Control Channel
CCH	Control Channel
CCTrCH	Coded Composite Transport Channel
CRC	Cyclic Redundancy Check
CTCH	Common Traffic Channel
DCCH	Dedicated Control Channel
DCH	Dedicated Channel
DL	Downlink
DSCH	Downlink Shared Channel
DTCH	Dedicated Traffic Channel
FACH	Forward Link Access Channel
FDD	Frequency Division Duplex
L1	Layer 1 (physical layer)
L2	Layer 2 (data link layer)
L3	Layer 3 (network layer)
LI	Length Indicator
LSB	Least Significant Bit
MAC	Medium Access Control
MRW	Move Receiving Window
MSB	Most Significant Bit
PCCH	Paging Control Channel
PCH	Paging Channel
PDU	Protocol Data Unit
PHY	Physical layer
PhyCH	Physical Channels
RACH	Random Access Channel
RLC	Radio Link Control
RRC	Radio Resource Control
SAP	Service Access Point
SDU	Service Data Unit
SHCCH	Shared Channel Control Channel
SN	Sequence Number
SUFI	Super Field
TCH	Traffic Channel
TDD	Time Division Duplex
TFI	Transport Format Indicator
TM	Transparent Mode
TMD	Transparent Mode Data
TTI	Transmission Time Interval
U-	User-
UE	User Equipment
UL	Uplink
UM	Unacknowledged Mode
UMD	Unacknowledged Mode Data
UMTS	Universal Mobile Telecommunications System
UTRA	UMTS Terrestrial Radio Access
UTRAN	UMTS Terrestrial Radio Access Network

4 General

4.1 ~~Void~~Objective

This subclause describes the architecture of the RLC sublayer.

4.2 Overview of~~on~~ the RLC sublayer architecture

The model presented in this subclause is intended to support the definition of the RLC sublayer only, and is not meant to specify or constrain the~~for~~ implementation ~~purposes~~ of the protocol. The RLC sublayer consists of RLC entities, of which there are three types: Transparent Mode (TM), Unacknowledged Mode (UM), and Acknowledged Mode (AM) RLC entities.

4.2.1 Model of the RLC sublayer

Figure 4.1~~ gives an overview model of the RLC layer. The figure~~ illustrates~~ the~~ different RLC~~ peer~~ entities in the RLC model.

An UM and a TM RLC entity can be configured to be a ~~sending~~transmitting RLC entity or a receiving RLC entity. The ~~sending~~transmitting RLC entity transmits RLC PDUs and the receiving RLC entity receives RLC PDUs. An AM RLC entity consists of a transmitting side, and a receiving side, where the transmitting side of the AM RLC entity transmits RLC PDUs and the receiving side of the AM RLC entity receives RLC PDUs.

Elementary procedures (see clause 11) are defined between a "Sender" and a "Receiver". In UM and TM, the ~~sending~~transmitting RLC entity acts as a Sender and the peer RLC entity acts as a Receiver. An AM RLC entity acts either as a Sender or as a Receiver depending on the elementary procedure. The Sender is the transmitter of AMD PDUs and the Receiver is the receiver of AMD PDUs.– A Sender or a Receiver can reside at either the UE or the UTRAN.

There is ~~a ~~one ~~sendingtransmitting~~transmitting and ~~a ~~one receiving RLC entity for each~~the~~ transparent mode (TM) ~~service~~ and ~~the ~~unacknowledged mode (UM) ~~unidirectional ~~service. ~~And~~There is ~~a ~~one combined, ~~sendingtransmitting~~transmitting and receiving entity for the acknowledged mode (AM) ~~bi-directional ~~service.~~.~~

In this specification, ~~the word “~~"transmitted~~”~~" is equivalent to "submitted to the lower layer" unless otherwise explicitly stated. -Each RLC UM, and TM entity uses one logical channel to send or receive data PDUs. An AM RLC entity can be configured to use one or two logical channels to send or receive data and control PDUs. If two logical channels are configured, they are of the same type (DCCH or DTCH). In figure 4.1, ~~T~~the dashed lines between the AM-Entities illustrate the possibility to send ~~the~~and receive RLC PDUs on separate logical channels, e.g. control PDUs on one, and data PDUs on the other. More detailed description~~s~~ of the different entities is~~are~~ given in subclauses 4.2.1.1, 4.2.1.2 and 4.2.1.3.

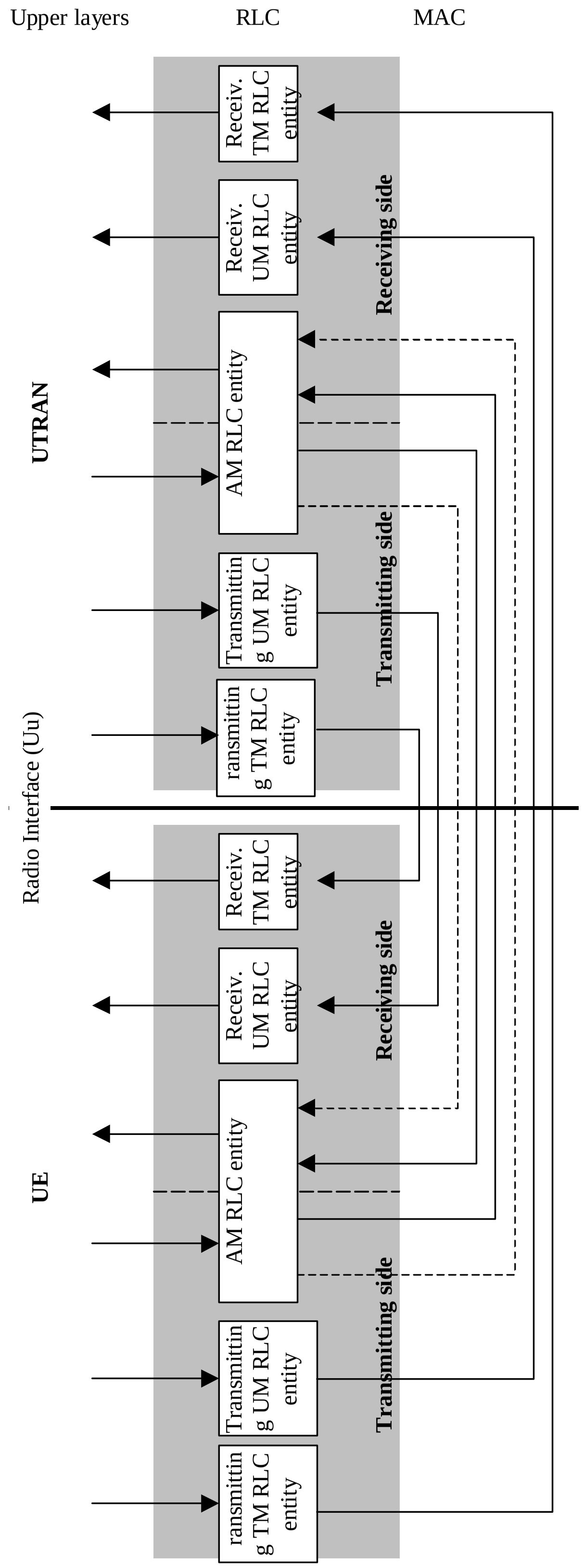


Figure 4.1: Overview model of the RLC sublayer

4.2.1.1 Transparent mode (TM) RLC entities

Figure 4.2 below shows the model of two transparent mode peer RLC entities. The logical channels used to communicate with the lower layer are different depending on the placement of the RLC entity (at the UE or the UTRAN)~~, and are~~ ~~specified~~described in the ~~diagram~~figure below.

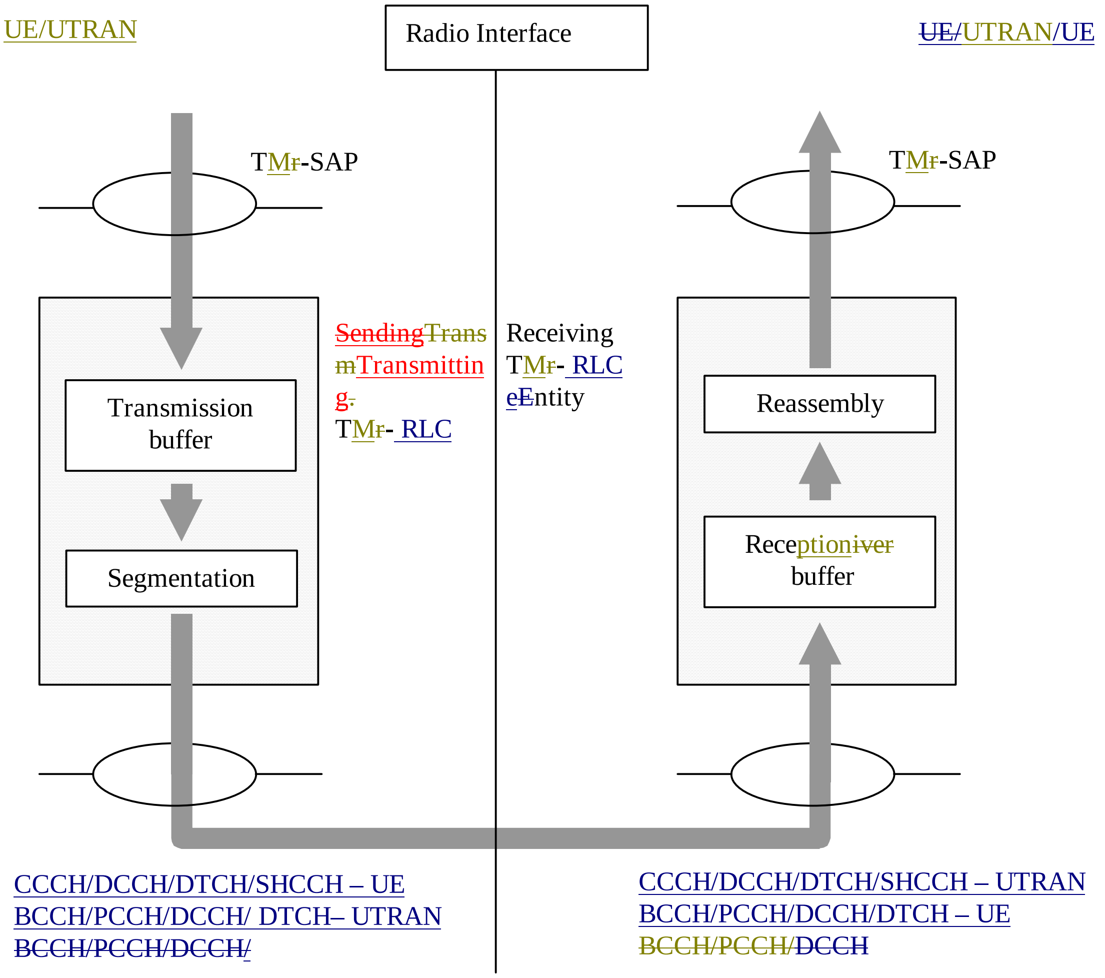


Figure 4.2: Model of two transparent mode peer entities

4.2.1.1.1 ~~Sending~~Transmitting TM RLC entity

The ~~sending~~transmitting~~transmitting~~ TM~~r~~-RLC entity receives ~~SDU~~RLC SDUs from ~~the upper layer~~upper layers through the TM~~r~~-SAP.

All received RLC SDUs must be of a length that is a multiple of one of the ~~configured~~valid TMD PDU lengths.

If segmentation has been configured by upper layers and a RLC SDU is larger than the TMD PDU size used by the lower layer for thata TTI, t~~The~~ ~~sending~~transmitting TM RLC entity ~~maymight~~ segments ~~the~~ ~~SDU~~RLC SDUs to fit the ~~into appropriately sized~~ ~~RLC PDU~~TMD PDUs size without adding ~~any~~ RLC headers~~overhead~~. ~~The segmentation is performed if an RLC SDU is larger thatthan the largest TMD PDU size (configured at the sending TM RLC entity) that is supported by the lower layer for that TTI. If segmentation is performedhas been configured by upper layer, a~~All the TMD PDUs carrying one RLC SDU are~~will be~~ sent in ~~one~~the same TTI, and no segment from another RLC SDU are~~will be~~ sent in this TTI~~only the segments from one RLC SDU will be sent in one TTI~~.

If ~~the~~ segmentation ~~is not used by the TM RLC entity~~has not been configured by upper layers, then more than one RLC SDU can be sent in one TTI by placing one RLC SDU in one TMD PDU. All TMD PDUs in one TTI must be of equal length.

When the processing of ~~an~~ a RLC SDU is complete, the resulting one or more TMD PDU(s) are/is ~~deliver~~submitted to the~~How to perform the segmentation is decided upon when the service is established. RLC delivers the RLC PDUs to~~

lower layer through either a BCCH, DCCH, PCCH, CCCH, SHCCH or a DTCH logical channel~~. The CCCH and SHCCH uses transparent mode only for the uplink. Which type of logical channel depends on if the upper layer is located in the control plane (BCCH, DCCH, PCCH, CCCH, SHCCH) or user plane (DTCH).~~

4.2.1.1.2 Receiving TM RLC entity

The receiving TM~~r~~-RLC entity receives ~~PDU~~TMD PDUs through the configured ~~one of the~~ logical channels from the lower layer.

If segmentation is configured by upper layers, all TMD PDUs received within one TTI are reassembled to form the RLC SDU~~The receiving TM~~ ~~RLC~~ ~~receiving entity~~ ~~reassembles (if segmentation~~ ~~at the sendersending TM RLC peer entity~~ ~~has been performed~~~~configured by upper layer~~~~) the~~ ~~RLC~~ ~~PSDUs~~ ~~from TMD PDUs~~ ~~into RLC SDUs.~~ ~~How to perform the reassembling is decided upon when the service is established.~~ ~~The reassembly of TMD PDUs into an RLC SDU is facilitated through the use of the information passed from the lower layer along with the delivered TMD PDUs in one TTI~~.

If segmentation is not configured by upper layers, each TMD PDU is treated as a RLC SDU.

The receiving TM RLC entity delivers~~ the~~ RLC SDUs to ~~the upper layer~~ ~~upper layer~~upper layers through the TM~~r~~-SAP.

4.2.1.2 Unacknowledged mode (UM) RLC entities

Figure 4.3 below shows the model of two unacknowledged mode peer RLC entities.

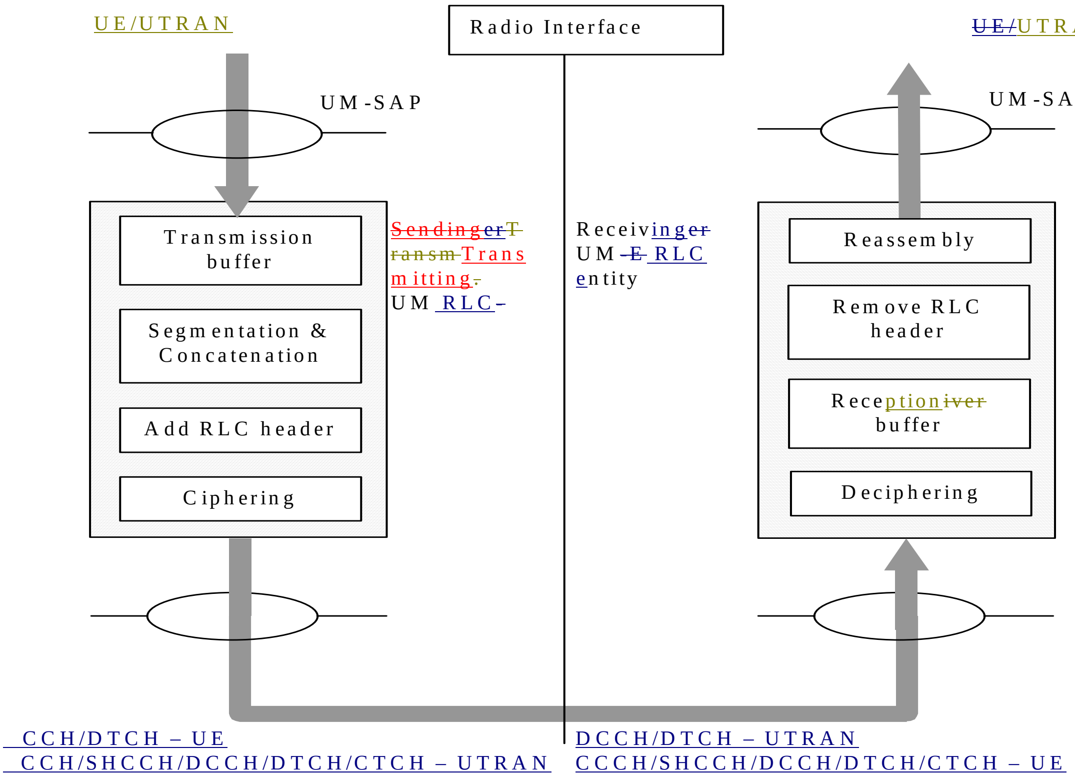


Figure 4.3: Model of two unacknowledged mode peer entities

4.2.1.2.1 ~~Sending~~Transmitting UM RLC entity

The ~~sending~~transmitting~~transmitting~~ UM RLC -entity receives ~~SDU~~RLC SDUs from ~~the upper layer~~upper layers through the UM-SAP.

The ~~sending~~transmitting UM RLC entity ~~maymight~~ segments the RLC SDUs into ~~RLC PDU~~UMD PDUs of appropriate size,~~. The segmentation is performed~~ if the RLC SDU is larger than the length of available space in the UMD PDU~~ of size supported by the lower layer for that TTI~~. ~~The SDU might also be concatenated with other SDUs.~~ The UMD PDU may contain segmented and/or concatenated RLC SDUs. UMD PDU may also contain padding to ensure that it is of a valid length. Length Indicators are used to define boundaries between RLC SDUs within UMD PDUs. Length Indicators are also used to define whether Padding is included in the UMD PDU.

If ~~the~~ ciphering is configured and started, an UMD PDU is ciphered (except for the UMD PDU header) before it is ~~delivered~~submitted to the lower layer.

The ~~sending~~transmitting UM RLC entity ~~deliver~~submits ~~the~~ ~~RLC PDU~~UMD PDUs to the lower layer through either a CCCH, SHCCH, DCCH, CTCH or a DTCH logical channel.~~. The CCCH and SHCCH uses unacknowledged mode only for the downlink. Which type of logical channel depends on if the upper layer is located in the control plane (CCCH, DCCH, SHCCH) or user plane (CTCH, DTCH).~~

4.2.1.2.2 Receiving UM RLC entity

The receiving UM- RLC entity receives ~~PDU~~UMD PDUs through the configured ~~one of the~~ logical channels from the lower layer.~~logical channels from the MAC sublayer~~.

The receiving UM RLC ~~receiving~~ entity deciphers (if ciphering is configured and started) the received UMD PDUs (except for the UMD PDU header). It removes RLC headers from ~~the~~ received ~~PDU~~UMD PDUs, and ~~uses the information placed by the sender in the fields of UMD PDU headers and to~~ reassembles RLC~~the~~ S~~P~~DUs (if segmentation and/or concatenation has been performed by the ~~senderSender~~transmitting UM RLC entity)~~ into RLC SDUs~~.

RLC SDUs are delivered by the receiving UM RLC entity to ~~the upper layer~~upper layers through the UM-SAP.

4.2.1.3 Acknowledged mode (AM) RLC entity

Figure 4.4 below shows the model of an acknowledged mode RLC entity.

The AM RLC entity can be configured to utilize one or two logical channels. The figure 4.4 shows the model of the AM RLC entity when one logical channel (shown as a solid line), and when two logical channels (shown as dashed lines) are used.

If one logical channel is configured, the transmitting side of the AM RLC entity submits AMD and Control PDUs to the lower layer on that logical channel. And the RLC PDU size shall be the same for AMD PDUs and control PDUs.

In case two logical channels are configured in the uplink, AMD PDUs are transmitted on the first logical channel, and control PDUs are transmitted on the second logical channel. In case two logical channels are configured in the downlink, AMD and Control PDUs can be transmitted on any of the two logical channels.

~~In case two logical channels are configured,used in the uplink, the first logical channel shallwillis be used for AMDdata PDUs, and the second logical channel shallwillis be used for control PDUs. In case one logical channel is usedconfigured, the RLC PDU size isshall be the same for AMD PDUs and control PDUs.~~

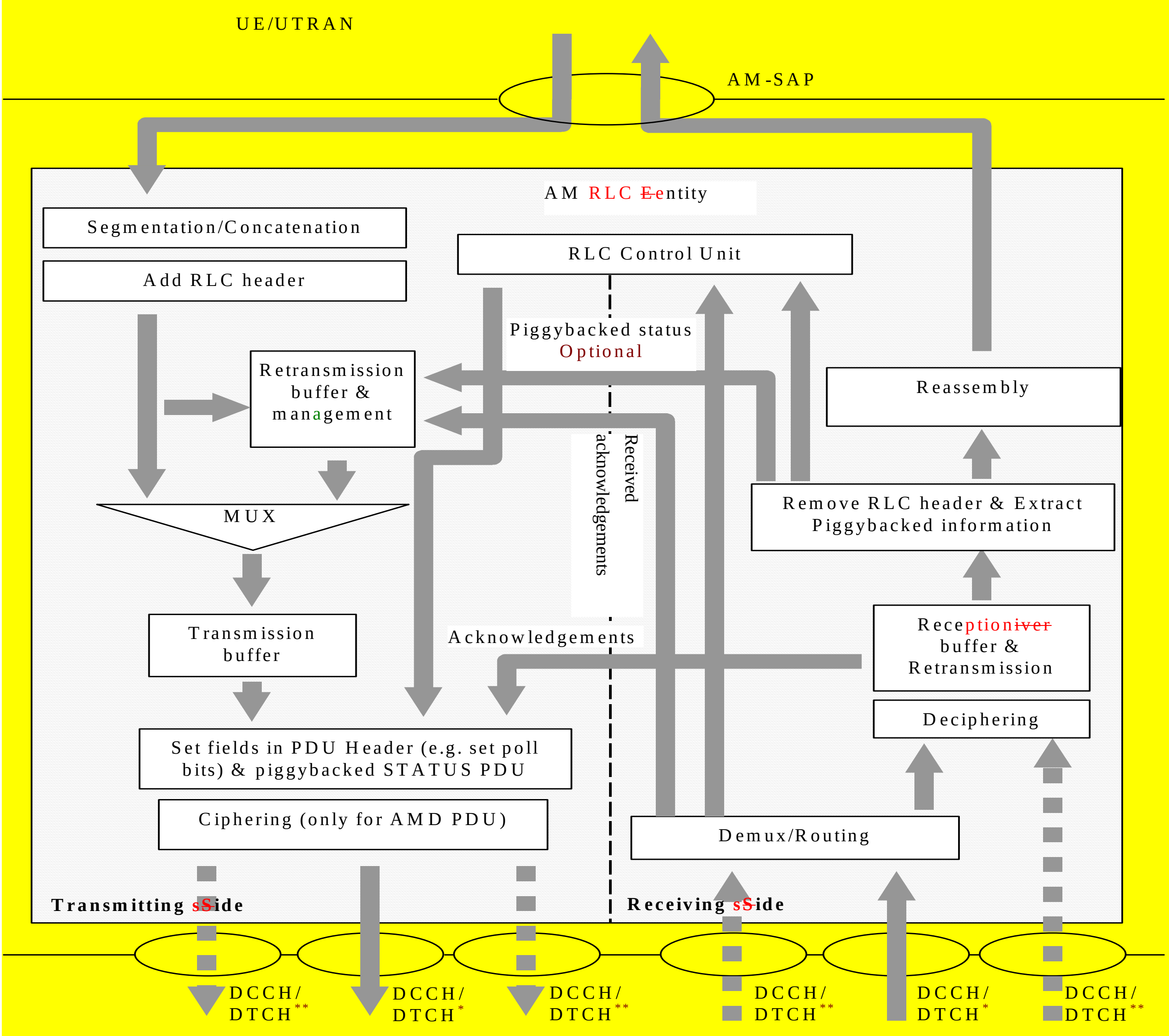

UE/UTRAN
AM-SAP
AM RLC ~~E~~entity
Segmentation/Concatenation
Add RLC header
RLC Control Unit
Piggybacked status
Optional
Retransmission buffer & management
Reassembly
Received acknowledgements
MUX
Remove RLC header & Extract Piggybacked information
Transmission buffer
Acknowledgements
Reception~~iver~~ buffer & Retransmission
Deciphering
Set fields in PDU Header (e.g. set poll bits) & piggybacked STATUS PDU
Ciphering (only for AMD PDU)
Demux/Routing
Transmitting s~~S~~ide
Receiving s~~S~~ide
DCCH/ DTCH**
DCCH/ DTCH*
DCCH/ DTCH**
DCCH/ DTCH**
DCCH/ DTCH*
DCCH/ DTCH**

Figure 4.4: Model of an acknowledged mode entity

4.2.1.3.1 Transmitting side

The transmitting side of the AM RLC -entity receives RLC SDUs from ~~the upper layer~~upper layers through the AM-SAP.

~~The~~ RLC SDUs are segmented and/or concatenated into ~~PDU~~AMD PDUs of a fixed length. The segmentation is performed if the received RLC SDU is larger than the length of available space in the AMD PDU. ~~An~~The AMD PDU ~~length~~ size is a semi-static value that is configured by ~~the upper layer~~upper layers~~decided in bearer setup~~, and can ~~only~~ be changed through the~~bearer~~ modification~~reconfiguration~~ of the AM RLC entity by ~~the upper layer~~upper layers. The AMD PDU may contain segmented and/or concatenated RLC SDUs. The AMD PDU may also contain Padding to ensure that it is of a valid ~~length~~size. Length Indicators are used to define boundaries between RLC SDUs within AMD PDUs. Length Indicators are also used to define whether Padding or Piggybacked STATUS PDU is included in the ~~UMD~~AMD PDU.

~~For concatenation or padding purposes, bits of information on the length and extension are inserted into the beginning of the last PDU where data from an SDU is included. Padding can be replaced by piggybacked status information. This includes setting the poll bit.~~

~~If several SDUs fit into one PDU, they are concatenated and the appropriate length indicators are inserted into the beginning of the PDU.~~ After ~~that~~ the segmentation and/or concatenation are performed, the ~~PDU~~AMD PDUs are placed in ~~the retransmission buffer and~~ the Retransmission buffer, and at the MUX.

~~The Retransmission buffer deletes buffered AMD PDUs, or indicates which AMD PDUs are to be retransmitted, based on the input from the receiving side of the RLC AM entity. The input consists of positive and negative acknowledgments of individual AMD PDUs received from the peer AM RLC entity in the status report (Piggybacked STATUS or STATUS PDU).~~AMD PDUs buffered in the Retransmission buffer ~~may be~~are deleted or retransmitted based on the status report found within a STATUS PDU or Piggybacked STATUS PDU sent ~~from~~by the peer AM RLC entity. This status report may contain positive or negative acknowledgements of individual AMD PDUs received ~~from~~by the peer AM RLC entity.

The MUX multiplexes AMD PDUs from the Retransmission buffer that need to be retransmitted, and the newly generated AMD PDUs delivered from the Segmentation/Concatenation function. ~~then decides which PDUs and when the PDUs are submitted to the lower layer.~~

The PDUs are ~~submitted via~~ delivered to the~~a~~ function that completes the AMD PDU header and potentially replaces padding with piggybacked status information. ~~The RLC entity shall assume a PDU to be transmitted when the PDU is submitted to lower layer. When Piggybacking mechanism is applied, t~~The padding in the AMD PDU may be replaced by the Piggybacked STATUS PDU that may be delivered from the Rece~~iver~~ption buffer at the receiving side of the AM RLC entity. ~~The~~A Piggybacked STATUS PDUs can be of variable size in order to match the amount of free space in the AMD PDU. The AMD PDU header is completed based on the input from the RLC Control ~~function~~Unit that indicates the values to set in various fields (e.g. Polling Bit). The function also multiplexes, if required, Control PDUs received from the RLC Control ~~u~~Unit (RESET and RESET ACK PDUs), and from the Rece~~iver~~ption buffer (Piggybacked STATUS and STATUS PDUs), with AMD PDUs~~, if one logical channel is configured. The multiplexing of control and data PDUs is not performed if two logical channels are configured~~.

The ciphering (if ~~it is~~ configured) is then applied ~~only~~ to~~for~~ the AMD PDUs. The ~~fixed 2 octets~~ AMD PDU header is not ciphered. Piggybacked STATUS PDU and Padding ~~parts~~ in~~of~~ AMD PDU (when present~~existing~~) are ciphered. ~~The other~~ Control PDUs (i.e. STATUS PDU, RESET PDU, and RESET ACK PDU) ~~shall~~~~will~~are not ~~be~~ ciphered.

The transmitting side of the AM RLC entity submits AMD PDUs to the lower layer through either one or two DCCH or DTCH logical channels.

~~The transmitting side of the AM RLC entity deliversubmits AMD and Control PDUs to the lower layer through either a DCCH or a DTCH logical channel, if one logical channel is configured. If two logical channels are configured (either of DCCH or DTCH type), AMD and Control PDUs are deliversubmitted to the lower layer on their individual logical channels on the uplink (UE as a Sender).   On the downlink (UTRAN as a Sender), AMD and Control PDUs can be submitted to the lower layer on either of the logical channels.~~

~~When Piggybacking mechanism is applied the padding is replaced by control information, in order to increase the transmission efficiency and making possible a faster message exchange between the peer-to-peer RLC entities. The piggybacked control information is not saved in any retransmission buffer. The piggybacked control information is contained in the piggybacked STATUS PDU, which is in turn included into the AMD-PDU. The piggybacked STATUS PDUs will be of variable size in order to match with the amount of free space in the AMD PDU.~~

~~The retransmission buffer also receives acknowledgements from the receiving side, which are used to indicate retransmissions of PDUs and when to delete a PDU from the retransmission buffer.~~

4.2.1.3.2 Receiving side

The receiving side of the AM- RLC entity receives AMD and Control PDUs through ~~one of~~ the configured logical channels from the lower layer~~, if one logical channel is configured~~. ~~If two logical channels are configured, AMD and Control PDUs are received on their individual logical channels from the lower layers~~.

AMD PDUs are routed to the Deciphering U~~u~~nit, where AMD PDUs (minus the ~~mandatory~~ AMD PDU header) are deciphered (if ciphering is configured and started), and then delivered to the Rece~~iver~~ption buffer.

The AMD PDUs are placed in the Reception ~~Receiver~~ buffer until a complete RLC SDU has been received. The Receiver ~~will~~ acknowledges successful reception or requests retransmission of the missing AMD PDUs by sending one or more STATUS PDUs to the AM RLC peer entity, through its transmitting side. If a Piggybacked STATUS PDU is found in an AMD PDU, it is delivered to the Retransmission buffer & Management Unit at the ~~sending~~transmitting side of the AM RLC entity, in order to purge the buffer of positively acknowledged AMD PDUs, and to indicate which AMD PDUs need to be retransmitted.

Once a complete RLC SDU has been received, the associated AMD PDUs are reassembled by the Reassembly ~~function~~Unit and delivered to ~~the upper layer~~upper layers through the AM-SAP. ~~If a Piggybacked STATUS PDU is found in an AMD PDU, it is delivered to the Retransmission buffer & management at the sending side of the AM RLC~~

~~entity, in order to purge the buffer of positively acknowledged AMD PDUs, and to indicate which AMD PDUs need to be retransmitted.~~

RESET and RESET ACK PDUs are ~~routed~~delivered to the RLC Control U~~u~~nit for processing. If a response to the peer AM RLC entity is needed, an appropriate Control PDU is delivered, by the RLC Control~~,~~ U~~u~~nit to the transmitting side of the AM RLC entity. The received STATUS PDUs are ~~routed~~ delivered to the Retransmission buffer and Management Unit at the transmitting side of the AM RLC entity, in order to purge the buffer of positively acknowledged AMD PDUs, and to indicate which AMD PDUs need to be retransmitted.

~~If two logical channels are configured, the Router function is transparent as Control PDUs are delivered directly to the RLC Control function, as are AMD PDUs to the Receiver buffer.~~

~~Piggybacked status information is extracted, if present. The PDUs are placed in the receiver buffer until a complete SDU has been received. The receiver buffer requests retransmissions of PDUs by sending negative acknowledgements to the peer entity. After that the RLC headers are removed from the PDUs and the PDUs are reassembled into an SDU. Finally the SDU is delivered to the upper layer. The receiving side also receives acknowledgements from the peer entity. The acknowledgements are passed to the retransmission buffer on the transmitting side.~~

5 Functions

The following functions are supported by RLC sublayer. For ~~a detailed~~an overall description of the following functions see [3]:

- Segmentation and reassembly.
- Concatenation.
- Padding.
- Transfer of user data.
- Error correction.
- In-sequence delivery of upper layer PDUs.
- Duplicate detection.
- Flow control.
- Sequence number check.
- Protocol error detection and recovery.
- Ciphering.
- ~~Polling.~~
- ~~Status transmission.~~
- SDU discard.
- ~~Estimated PDU Counter (EPC) mechanism.~~
- ~~Suspend/resume function.~~
- ~~Stop/continue function.~~
- ~~Re-establishment function.~~

6 Services provided to upper layers

This clause describes the different services provided by RLC sublayer to upper layers. It also includes the mapping of RLC functions to different RLC services. For a detailed description of the RLC services~~following functions~~ see [3].

- **Transparent data transfer Service:**

The following functions are needed to support~~may be used~~are~~configured needed to supportwhen~~ transparent data transfer~~ is supported~~:

- Segmentation and reassembly.
- Transfer of user data.
- SDU discard.

- **Unacknowledged data transfer Service:**

The following functions are needed to support~~may be used~~are~~configured needed to supportwhen~~ unacknowledged data transfer~~ is supported~~:

- Segmentation and reassembly.
- Concatenation.
- Padding.
- Transfer of user data.
- Ciphering.
- Sequence number check.
- SDU discard.
- ~~Suspend/resume function.~~
- ~~Stop/continue function.~~
- ~~Re-establishment function.~~

- **Acknowledged data transfer Service:**

The following functions are needed to support~~may be used~~are~~configured needed to supportwhen~~ acknowledged data transfer~~ is supported~~:

- Segmentation and reassembly.
- Concatenation.
- Padding.
- Transfer of user data.
- Error correction~~ ()~~.
- In-sequence delivery of ~~upper layer ~~upper layers PDUs.
- Duplicate detection.
- Flow Control.
- Protocol error detection and recovery.
- Ciphering.
- ~~Polling.~~
- ~~Status transmission.~~
- SDU discard.
- ~~Estimated PDU Counter (EPC) mechanism.~~

~~- Suspend/resume function.~~

~~- Stop/continue function.~~

~~- Re-establishment function.~~

- **Maintenance of QoS as defined by upper layers.**

- **Notification of unrecoverable errors.**

6.1 Mapping of services/functions onto logical channels

The following tables show the applicability of services and functions to the logical channels in UL/DL and UE/UTRAN. A '+' in a column denotes that the service/function is applicable for the logical channel in question whereas a '-' denotes that the service/function is not applicable.

Table 6.1: RLC modes and functions in UE uplink side

Service	Functions	CCCH	SHCCH	DCCH	DTCH
Transparent Service	Applicability	+	+	+	+
	Segmentation	-	-	+	+
	Transfer of user data	+	+	+	+
	SDU Discard	-	-	+	+
Unacknowledged Service	Applicability	-	-	+	+
	Segmentation	-	-	+	+
	Concatenation	-	-	+	+
	Padding	-	-	+	+
	Transfer of user data	-	-	+	+
	Ciphering	-	-	+	+
	SDU Discard	-	-	+	+
	~~Suspend/Resume Function~~	~~-~~	~~-~~	~~+~~	~~+~~
	~~Stop/Continue Function~~	~~-~~	~~-~~	~~+~~	~~+~~
	~~Re-establishment FunctionSDU Discard~~	~~-~~	~~-~~	~~+~~	~~+~~
Acknowledged Service	Applicability	-	-	+	+
	Segmentation	-	-	+	+
	Concatenation	-	-	+	+
	Padding	-	-	+	+
	Transfer of user data	-	-	+	+
	Flow Control	-	-	+	+
	Error Correction	-	-	+	+
	Protocol error correction & recovery	-	-	+	+
	Ciphering	-	-	+	+
	~~Polling~~	~~-~~	~~-~~	~~+~~	~~+~~
	SDU Discard	-	-	+	+
	~~Suspend/Resume Function~~	~~-~~	~~-~~	~~+~~	~~+~~
	~~Stop/Continue Function~~	~~-~~	~~-~~	~~+~~	~~+~~
	~~Re-establishment Function~~	~~-~~	~~-~~	~~+~~	~~+~~

Table 6.2: RLC modes and functions in UE downlink side

Service	Functions	BCCH	PCCH	SHCCH	CCCH	DCCH	DTCH	CTCH
Transparent Service	Applicability	+	+	-	-	+	+	-
	Reassembly	-	-	-	-	+	+	-
	Transfer of user data	+	+	-	-	+	+	-
Unacknowledged Service	Applicability	-	-	+	+	+	+	+
	Reassembly	-	-	+	+	+	+	+
	Deciphering	-	-	-	-	+	+	-
	Sequence number check	-	-	+	+	+	+	+
	Transfer of user data	-	-	+	+	+	+	+
	~~Suspend/Resume Function~~	~~-~~	~~-~~	~~-~~	~~-~~	~~+~~	~~+~~	~~-~~
	~~Stop/Continue Function~~	~~-~~	~~-~~	~~-~~	~~-~~	~~+~~	~~+~~	~~-~~
	~~Re-establishment FunctionTransfer of user data~~	~~--~~	~~--~~	~~-+~~	~~-+~~	~~++~~	~~++~~	~~-+~~
Acknowledged Service	Applicability	-	-	-	-	+	+	-
	Reassembly	-	-	-	-	+	+	-
	Error correction	-	-	-	-	+	+	-
	Flow Control	-	-	-	-	+	+	-
	In sequence delivery	-	-	-	-	+	+	-
	Duplicate detection	-	-	-	-	+	+	-
	Protocol error correction & recovery	-	-	-	-	+	+	-
	Deciphering	-	-	-	-	+	+	-
	Transfer of user data	-	-	-	-	+	+	-
	~~Status Transmission~~	~~-~~	~~-~~	~~-~~	~~-~~	~~+~~	~~+~~	~~-~~
	SDU DIscard	-	-	-	-	+	+	-
	~~Suspend/Resume Function~~	~~-~~	~~-~~	~~-~~	~~-~~	~~+~~	~~+~~	~~-~~
	~~Stop/Continue Function~~	~~-~~	~~-~~	~~-~~	~~-~~	~~+~~	~~+~~	~~-~~
	~~Re-establishment Function~~	~~-~~	~~-~~	~~-~~	~~-~~	~~+~~	~~+~~	~~-~~

Table 6.3: RLC modes and functions in UTRAN downlink side

Service	Functions	BCCH	PCCH	CCCH	SHCCH	DCCH	DTCH	CTCH
Transparent Service	Applicability	+	+	-	-	+	+	-
	Segmentation	-	-	-	-	+	+	-
	Transfer of user data	+	+	-	-	+	+	-
	SDU Discard	-	-	-	-	+	+	-
Unacknowledged Service	Applicability	-	-	+	+	+	+	+
	Segmentation	-	-	+	+	+	+	+
	Concatenation	-	-	+	+	+	+	+
	Padding	-	-	+	+	+	+	+
	Ciphering	-	-	-	-	+	+	-
	Transfer of user data	-	-	+	+	+	+	+
	SDU Discard	-	-	-	-	+	+	-
	~~Suspend/Resume Function~~	~~-~~	~~-~~	~~-~~	~~-~~	~~+~~	~~+~~	~~-~~
	~~Stop/Continue Function~~	~~-~~	~~-~~	~~-~~	~~-~~	~~+~~	~~+~~	~~-~~
	~~Re-establishment FunctionSDU Discard~~	~~--~~	~~--~~	~~--~~	~~--~~	~~++~~	~~++~~	~~--~~
Acknowledged Service	Applicability	-	-	-	-	+	+	-
	Segmentation	-	-	-	-	+	+	-
	Concatenation	-	-	-	-	+	+	-
	Padding	-	-	-	-	+	+	-
	Transfer of user data	-	-	-	-	+	+	-
	Flow Control	-	-	-	-	+	+	-
	Error Correction	-	-	-	-	+	+	-
	Protocol error correction & recovery	-	-	-	-	+	+	-
	Ciphering	-	-	-	-	+	+	-
	~~Polling~~	~~-~~	~~-~~	~~-~~	~~-~~	~~+~~	~~+~~	~~-~~
	SDU Discard					+	+	-
	~~Suspend/Resume Function~~	~~-~~	~~-~~	~~-~~	~~-~~	~~+~~	~~+~~	~~-~~
	~~Stop/Continue Function~~	~~-~~	~~-~~	~~-~~	~~-~~	~~+~~	~~+~~	~~-~~
	~~Re-establishment Function~~	~~-~~	~~-~~	~~-~~	~~-~~	~~+~~	~~+~~	~~-~~

Table 6.4: RLC modes and functions in UTRAN uplink side

Service	Functions	CCCH	SHCCH	DCCH	DTCH
Transparent Service	Applicability	+	+	+	+
	Reassembly	-	-	+	+
	Transfer of user data	+	+	+	+
Unacknowledged Service	Applicability	-	-	+	+
	Reassembly	-	-	+	+
	Deciphering	-	-	+	+
	Sequence number check	-	-	+	+
	Transfer of user data	-	-	+	+
	~~Suspend/Resume Function~~	~~-~~	~~-~~	~~+~~	~~+~~
	~~Stop/Continue Function~~	~~-~~	~~-~~	~~+~~	~~+~~
	~~Re-establishment FunctionTransfer of user data~~	~~--~~	~~--~~	~~++~~	~~++~~
Acknowledged Service	Applicability	-	-	+	+
	Reassembly	-	-	+	+
	Error correction	-	-	+	+
	Flow Control	-	-	+	+
	In sequence delivery	-	-	+	+
	Duplicate detection	-	-	+	+
	Protocol error correction & recovery	-	-	+	+
	Deciphering	-	-	+	+
	Transfer of user data	-	-	+	+
	~~Status Transmission~~	~~-~~	~~-~~	~~+~~	~~+~~
	SDU DIscard	-	-	+	+
	~~Suspend/Resume Function~~	~~-~~	~~-~~	~~+~~	~~+~~
	~~Stop/Continue Function~~	~~-~~	~~-~~	~~+~~	~~+~~
	~~Re-establishment Function~~	~~-~~	~~-~~	~~+~~	~~+~~

7 Services expected from MAC

For a detailed description of the service provided by the MAC sublayer to upper layers~~following functions~~ see [3].

- Data transfer.

8 Elements for layer-to-layer communication

The interaction between the RLC sublayer and other layers are described in terms of primitives where the primitives represent the logical exchange of information and control between the RLC sublayer and other layers. The primitives shall not specify or constrain the implementation~~s~~.

8.1 Primitives between RLC and upper layers

The primitives between RLC and upper layers are shown in Table 8.1.

Table 8.1: Primitives between RLC and upper layers

Generic Name	Parameters			
	Req.	Ind.	Resp.	Conf.
RLC-AM-DATA	Data, CNF, MUI	Data, DiscardInfo	Not Defined	MUI
RLC-UM-DATA	Data, Use special LI	Data	Not Defined	Not Defined
RLC-~~TR~~TM-DATA	Data	Data	Not Defined	Not Defined
CRLC-CONFIG	E/R, Stop, Continue, Ciphering Elements (UM/AM only), TM_parameters (TM only), UM_parameters (UM only), AM_parameters (AM only)	Not Defined	Not Defined	Not Defined
CRLC-SUSPEND (UM/AM only)	N	Not Defined	Not Defined	VT(US) (UM only), VT(S) (AM only)
CRLC-RESUME (UM/AM only)	No Parameter	Not Defined	Not Defined	Not Defined
CRLC-STATUS	Not Defined	EVC	Not Defined	Not Defined

Each Primitive is defined as follows:

RLC-AM-DATA-Req/Ind/Conf

- RLC-AM-DATA-Req is used by upper layers to request transmission of~~ an~~ a RLC SDU~~upper layer PDU~~ in acknowledged mode.

- RLC-AM-DATA-Ind is used by the AM RLC entity to deliver to upper layers a RLC SDU~~s~~ that has~~ve~~ been transmitted in acknowledged mode, and to indicate to upper layers of the discarded RLC SDU in the peer RLC AM entity.

- RLC-AM-DATA-Conf is used by the AM RLC entity to confirm to upper layers the reception of~~ an~~ a RLC SDU by the peer-RLC AM entity.

RLC-UM-DATA-Req/Ind

- RLC-UM-DATA-Req is used by upper layers to request transmission of~~ an~~ a RLC SDU ~~upper layer PDU~~ in unacknowledged mode~~, and to request the use of the special procedure when handling the RLC SDU~~.

- RLC-UM-DATA-Ind is used by the UM RLC entity to deliver to upper layers a RLC SDU~~s~~ that has~~ve~~ been transmitted in unacknowledged mode.

RLC-~~TR~~TM-DATA-Req/Ind

- RLC-~~TR~~TM-DATA-Req is used by upper layers to request transmission of~~ an~~ a RLC SDU~~upper layer PDU~~ in transparent mode.

- RLC-~~TR~~TM-DATA-Ind is used by the TM RLC entity to deliver to upper layers~~ ~~ a RLC SDU~~s~~ that has~~ve~~ been transmitted in transparent mode.

CRLC-CONFIG-Req

This primitive is used by upper layers to establish, re-establish, release, stop, continue or ~~reconfigure~~ modify the RLC. Ciphering elements are included for UM and AM operation.

CRLC-SUSPEND-Req/Conf

~~This primitive~~

- CRLC-SUSPEND-Req is used by upper layers to suspend the UM or AM RLC entity~~, after a specified number of SNsstarting at a specified time~~.

- CRLC-SUSPEND-Conf is used by the UM or AM RLC entity to ~~inform upper layers of its state variable~~confirm that the entity is suspended.~~ The N parameter indicates that RLC shall not send a PDU with SN>=VT(S)+N for AM and SN>=VT(US)+N for UM, where N is an integer. RLC informs upper layers of the VT(S) for AM and VT(US) for UM in the confirm primitive.~~

CRLC-RESUME-Req

This primitive is used by upper layers to resume the UM or AM RLC entity after~~when~~ the UM or AM RLC entity has been suspended.

CRLC-STATUS-Ind

It is used by an~~the~~ RLC entity to send status information to upper layers.

8.2 Primitive parameters

Following parameters are used in the primitives:

1) The parameter Data is the RLC SDU that is ~~processed by an RLC sending entity (via segmentation/concatenation) and ~~mapped onto the Data field in RLC PDUs.~~ The Data parameter may be divided over several RLC PDUs. In~~ When AM or UM RLC entities are used,~~,case of an RLC-AM-DATA or an RLC-UM-DATA primitive~~ the length of the Data parameter ~~shallwill be~~is a multiple of 8 bits~~octet-aligned~~, otherwise (TM RLC entity) the length of Data parameter is a bit-string whose length may not be a multiple of 8 bits.~~.~~

2) The parameter Confirmation ~~r~~Request (CNF) indicates whether the ~~sending~~transmitting side of the AM RLC entity needs to confirm the reception of the RLC SDU by the peer-RLC AM entity. If required, once all AMD PDUs that make up the RLC SDU are positively acknowledged by the receiving AM RLC entity, the ~~sending~~transmitting AM RLC entity~~ will~~ notifies~~y the upper layer~~upper layers.

3) The parameter Message Unit Identifier (MUI) is an identity of the RLC SDU, which is used to indicate which RLC SDU that is confirmed with the RLC-AM-~~DATA ~~DATA-~~conf~~Conf. primitive.

4) The parameter E/R indicates ~~the ~~establishment, ~~(~~re~~)-~~establishment, release or modification of~~ an~~ a RLC entity, where re-establishment is applicable to the~~for~~ AM and ~~the~~ UM RLC entities only. If the ~~it indicates~~ re-establishment is requested, the state variables and configurable parameters are initialised according to subclause 9.7.7. If ~~it indicates~~the release is requested, all protocol parameters, variables and timers ~~shallwill be~~are released and the RLC entity ~~shallwill exit the DATA_TRANSFER_READYdata transfer ready~~enters the NULL state. If ~~theit indicates~~ modification is requested, the protocol parameters indicated by upper layers (e.g. ciphering parameters) ~~shallwill ~~only are~~be~~ modified, while~~ith~~ keeping the other protocol parameters, such as the protocol variables, ~~the ~~protocol timers and ~~the ~~protocol state unchanged. AM RLC entities ~~shallwill~~are always~~ be~~ re-established if the AMD PDU size is changed. The modification of other protocol parameters does not ~~warrant~~ require a re-establishment.

5) The parameter Event Code (EVC) indicates the reason for the CRLC-STATUS-~~ind~~Ind (e.g., unrecoverable errors such as data link layer loss or recoverable status events such as reset.).

6) The parameter ~~c~~Ciphering ~~e~~Elements are only applicable for UM and AM operations. These parameters are Ciphering Mode, Ciphering Key, Transmitting Activation Time (SN to activate a new ciphering configuration at the ~~sender~~Sender~~transmitter~~), Receiving Activation Time (SN to activate a new ciphering configuration at the ~~receiver~~Receiver) and HFN (Hyper Frame Number).

7) The AM_parameters are only applicable for AM operation. ~~It contains~~These parameters are AMD PDU size, In-sequence Delivery Indication (indicating that RLC SDUs ~~shallwill be~~are delivered to ~~the~~ upper layers in sequence or out of sequence), Timer values (see subclause 9.5), Protocol parameter values (see subclause 9.6), Polling triggers (see subclause 9.7.1), Status triggers (see subclause 9.7.2), Periodical Status blocking configuration (see subclause 9.7.2), SDU discard mode (see subclause 9.7.3), Minimum WSN (see subclause 9.2.2.11.3), and Send MRW. The Minimum WSN ~~shallwill~~is always~~ be~~ greater than or equal to the number of transport blocks in the smallest transport block set. The Send MRW indicates that the information of each discarded ~~SDU~~RLC SDU ~~shallwill be~~is sent to the ~~receiver~~Receiver, and the MRW SUFI ~~shallwill be~~is sent to the ~~receiver~~Receiver even if no segments of the ~~SDU~~RLC SDU to be discarded were submitted to a lower layer.

8) The parameter DiscardInfo indicates to upper layers the discarded RLC SDU in the peer-RLC AM entity. It is applicable only when in-sequence delivery is configured,~~active~~ and it is ~~purposed~~ to be used when~~ the~~ upper layers requires the reliable data transfer~~. and especially the information of the discarded RLC SDU~~.

9) The Stop parameter indicates ~~that~~ to the RLC entity ~~shall will~~to (see subclause 9.7.6):

- discard all ~~not transmit nor receive~~ RLC PDUs received from the lower layer~~(see subclause …)~~..

- not submit to lower layer any RLC PDUs.

10) The Continue parameter indicates to~~that~~ the RLC entity ~~shallwill~~ to continue transmission and reception of RLC PDUs.

11) The parameter Use special LI indicates that the LI indicating that~~ an~~ a RLC SDU begins in the beginning of~~ an~~ a RLC PDU ~~(the first data octet of the PDU is the first octet of an SDU) shallwill~~is to be used (see subclause 9.2.2.8)~~. If the RLC SDU does not begin in the beginning of the RLC PDU, or if the LI indicating that an SDU ended exactly in the end or one octet short (only when 15 bit LI is used) of the previous RLC PDU is present, the special LI shallwill not be used~~.

12) The UM_parameters are only applicable for UM operation. It contains Timer_Discard value (see subclause 9.5) and largest UMD PDU size (see subclause 9.2.2.8).

13) The TM_parameters are only applicable for TM operation. It contains e.g. segmentation indication (see subclauses 9.2.2.9 and 11.1.2.1), Timer_Discard value (see subclause 9.5) and delivery of erroneous SDU indication (see subclause 11.1.3).

14) The N parameter indicates that~~ an~~ a RLC entity will not send a PDU with SN>=VT(S)+N for AM and SN>=VT(US)+N for UM, where N is a~~n~~ non-negative integer.

15) The VT(S) parameter indicates ~~that~~ the value of the Send State Variable for the case of the AM.

16) The VT(US) parameter indicates ~~that~~ the value of the UM Data State Variable, for the case of the UM.

9 Elements for peer-to-peer communication

9.1 Protocol data units

The ~~PDUs~~structures defined in this subclause are normative.

9.1.1 Data PDUs

a) TM~~r~~D PDU (Transparent Mode Data PDU).

The TM~~r~~D PDU is used to convey RLC SDU data without adding any RLC overhead. The TM~~r~~D PDU is used by RLC when it is in transparent mode.

b) UMD PDU (Unacknowledged Mode Data PDU).

The UMD PDU is used to convey sequentially numbered PDUs containing RLC SDU data. It is used by RLC when using unacknowledged data transfer.

c) AMD PDU (Acknowledged Mode Data PDU).

The AMD PDU is used to convey sequentially numbered PDUs containing RLC SDU data. The AMD PDU is used by RLC when it is in acknowledged mode.

9.1.2 Control PDUs

a) STATUS PDU and Piggybacked STATUS PDU

The STATUS PDU and the Piggybacked STATUS PDU are used in acknowledged mode:

- by the ~~receiving~~ Receiver ~~entity~~ to inform the ~~transmitting~~ Sender ~~entity~~ about missing and received AMD PDUs in~~at~~ the Receiver~~receiving entity~~;
- by the ~~receiving entity~~Receiver to inform the ~~transmitting entity~~Sender about the size of the allowed transmission window;
- ~~and~~ by the ~~transmitting entity~~Sender to request the ~~receiving entity~~Receiver to move the ~~receiving~~ reception window;~~.~~and
- ~~and~~ by the Receiver to acknowledge the Sender about the reception of the request to move the reception window~~Move Receiving Window SUFI~~.

b) RESET PDU

The RESET PDU is used in acknowledged mode to reset all protocol states, protocol variables and protocol timers of the peer RLC entity in order to synchronise the two peer entities.

c) RESET ACK PDU

The RESET ACK PDU is an acknowledgement to the RESET PDU.

Table 9.1: RLC PDU names and descriptions

Data Transfer Mode	PDU name	Description
Transparent	TM~~r~~D	Transparent mode data
Unacknowledged	UMD	Sequenced unacknowledged mode data
Acknowledged	AMD	Sequenced acknowledged mode data
	STATUS	Solicited or Unsolicited Status Report, Change window size command, SDU discard command, or SDU discard acknowledgement
	Piggybacked STATUS	Piggybacked Solicited or Unsolicited Status Report, Change window size command, SDU discard command, or SDU discard acknowledgement
	RESET	Reset Command
	RESET ACK	Reset Acknowledgement

9.2 Formats and parameters

The ~~data~~ formats of PDUs and their parameters defined in this subclause are normative.

9.2.1 Formats

This subclause specifies the format of the RLC PDUs. The parameters of each PDU are explained in subclause 9.2.2.

9.2.1.1 General

An RLC PDU is a bit string~~, with a length not necessarily a multiple of 8 bits~~. In the ~~drawings ~~figures in subclause 9.2, bit strings are represented by tables in which the first bit is the leftmost one on the first line of the table, the last bit is the rightmost one on 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.

Depending on the provided service, RLC SDUs are bit strings, with any non-null length, or bit strings with a multiple of 8 bits~~an integer number of octets~~ in length.~~ An~~ A RLC SDU is included into~~ an~~ a RLC PDU from first bit onward.

9.2.1.2 TM~~r~~D PDU

The TM~~r~~D PDU is used to transfer~~s~~ user data when RLC is operating in transparent mode. No overhead is added to the SDU by RLC. The data length is not constrained to be a multiple of 8 bits~~an integer number of octets~~.

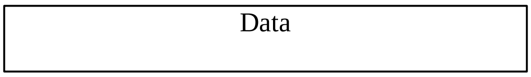


Figure 9.1: TM~~r~~D PDU

9.2.1.3 UMD PDU

The UMD PDU is used to transfer~~s~~ user data when RLC is operating in unacknowledged mode. The length of the data part shall be a multiple of 8 bits~~an integer number of octets~~. The UMD PDU header consists of the first octet, which contains the "S~~s~~equence N~~n~~umber". The RLC header consists of the first octet and all the octets that contain "~~l~~Length ~~i~~Indicators".

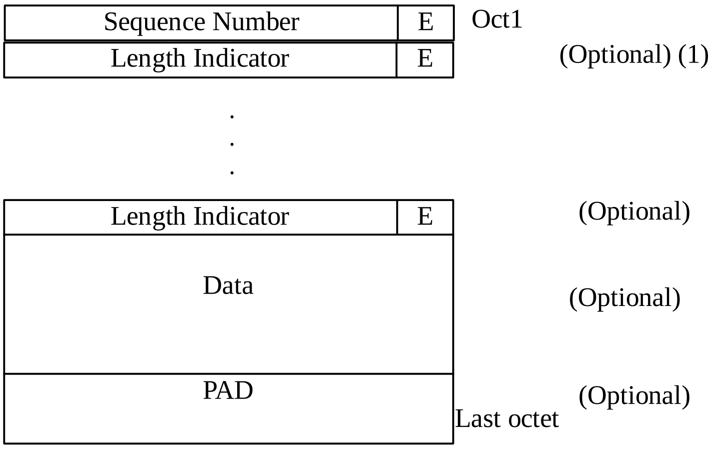


Figure 9.2: UMD PDU

NOTE (1): The~~ Length Indicator~~ "Length Indicator" may be 15 bits.

9.2.1.4 AMD PDU

The AMD PDU is used to transfer~~s~~ user data,~~ and~~ piggybacked status information and ~~requests status report by setting~~the Polling bit when RLC is operating in acknowledged mode. The length of the data part shall be a multiple of 8 bits~~an integer number of octets~~. The AMD PDU header consists of the first two octets, which contain the "S~~s~~equence N~~n~~umber". The RLC header consists of the first two octets and all the octets that contain "L~~l~~ength I~~i~~ndicators".

D/C	Sequence Number		Oct1
Sequence Number	P	HE	Oct2
Length Indicator		E	Oct3 (Optional) (1)
. . .			
Length Indicator		E	(Optional)
Data			(Optional)
PAD or a piggybacked STATUS PDU			(Optional) Last octet

NOTE (1): The~~ Length Indicator ~~ "Length Indicator" may be 15 bits.

Figure 9.3: AMD PDU

9.2.1.5 STATUS PDU

The STATUS PDU is used to report the status between two RLC AM entities~~. Both receiver and transmitter status information may be included in the same STATUS PDU~~.

The format of the STATUS PDU is given in Figure 9.4 below. The Figure shows an example of STATUS PDU and the length of each SUFI is dependent on the SUFI type.

D/C	PDU type	$SUFI_1$	Oct 1
$SUFI_1$			Oct2
…			
$SUFI_K$			
PAD			Last octet

Figure 9.4: ~~Status Information Control PDU (~~STATUS PDU~~)~~

Up to K super-fields ($SUFI_1$-$SUFI_K$) can be included into one STATUS PDU, in which each super-field can be of different type. The size of a STATUS PDU is variable and upper bounded by the maximum RLC PDU size used by the logical channel on which the control PDUs are sent. Padding shall be included to exactly fit one of the PDU sizes used by the logical channel on which the control PDUs are sent. The length of the STATUS PDU shall be a multiple of 8 bits~~an integer number of octets~~.

9.2.1.6 Piggybacked STATUS PDU

The format of the piggybacked STATUS PDU is the same as for ~~the ordinary Control~~STATUS PDU except that the D/C field is replaced by a reserved bit (R2). This PDU can be ~~used to ~~piggybacked ~~STATUS PDU ~~in an AMD PDU if the data does not fill the complete AMD PDU. The PDU Type field is set to ~~zero ~~“000” and all other values are invalid for this version of the protocol and the PDU is discarded.

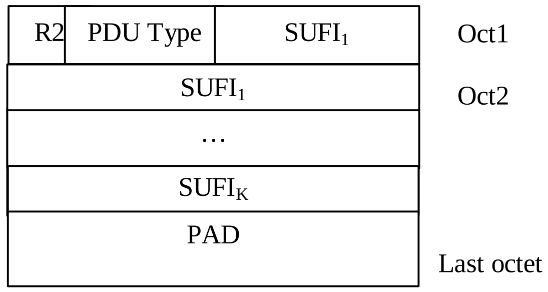


Figure 9.5: Piggybacked STATUS PDU

9.2.1.7 RESET, RESET ACK PDU

The RESET PDU (~~and~~ the RESET ACK PDU) have a one-bit sequence number field (RSN)~~. With the aid of this field the Receiver can define whether the received RESET PDU is transmitted by the Sender for the first time or~~ in order to know whether or not it is ~~or not~~ a retransmission of a previous RESET PDU (of a previous RESET ACK PDU).

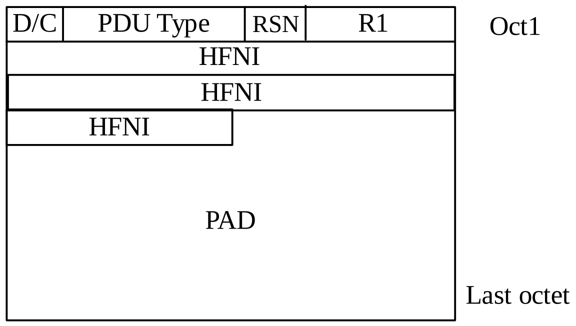


Figure 9.6: RESET, RESET ACK PDU

The size of a RESET or RESET ACK PDU is variable and upper bounded by the maximum RLC PDU size used by the logical channel on which the control PDUs are sent. Padding shall be included to exactly fit one of the PDU sizes used by the logical channel on which the control PDUs are sent. The length of the RESET or RESET ACK PDU shall be a multiple of 8 bits~~an integer number of octets~~.

9.2.2 Parameters

If not otherwise mentioned in the definition of each field then the bits in the parameters shall be interpreted as follows: the left-most bit string is the first and most significant and the right most bit is the last and least significant bit.

Unless otherwise mentioned, integers are encoded in standard binary encoding for unsigned integers. In all cases, including when a value extends over more than one octet as shown in the tables, the bits appear ordered from MSB to LSB when read in the PDU.

9.2.2.1 D/C field

Length: 1bit.

The D/C field indicates the type of an ~~acknowledged mode~~AM PDU. It can be either data or control PDU.

Bit	Description
0	Control PDU
1	~~Acknowledged mode~~ Ddata PDU

9.2.2.2 PDU Type

Length: 3 bit.

The PDU type field indicates the Control PDU type.

Bit	PDU Type
000	STATUS
001	RESET
010	RESET ACK
011-111	Reserved (PDUs with this coding will be discarded by this version of the protocol).

9.2.2.3 Sequence Number (SN)

This field indicates the "S~~s~~equence N~~n~~umber" of the PDU, encoded in binary.

PDU type	Length	Notes
AMD PDU	12 bits	Used for retransmission and reassembly
UMD PDU	7 bits	Used for reassembly

9.2.2.4 Polling bit (P)

Length: 1bit.

This field is used to request a status report (one or several STATUS PDUs) from the ~~receiver RLC~~Receiver.

Bit	Description
0	Status report not requested
1	Request a status report

9.2.2.5 Extension bit (E)

Length: 1bit.

This bit indicates if the next octet will be a "~~l~~Length I~~i~~ndicator" and E bit.

Bit	Description
0	The next field is data, piggybacked STATUS PDU or padding
1	The next field is Length Indicator and E bit

9.2.2.6 Reserved 1 (R1)

Length: 3 bits.

This field in the RESET PDU and RESET ACK PDU is used to ~~achieve octet alignment~~have a multiple of 8 bits in length and for this purpose it is coded as "000". Other functions of it are left for future releases.

9.2.2.7 Header Extension Type (HE)

Length: 2 bits.

This two-bit field indicates if the next octet will be data or a "L~~l~~ength I~~i~~ndicator" and E bit.

Value	Description
00	The succeeding octet contains data
01	The succeeding octet contains a length indicator and E bit
10-11	Reserved (PDUs with this coding will be discarded by this version of the protocol).

9.2.2.8 Length Indicator (LI)

The "Length Indicator" is used to indicate, each time, the end of~~ an~~ a ~~SDU~~RLC SDU occurs in the PDU.

Unless for the predefined values reserved for special purposes and listed in the tables below, ~~The~~the "Length Indicator" shall:

- be set to ~~points out~~ the number of octets between the end of the ~~last Length Indicator field~~RLC header and up to and including the octet at the end of~~ an~~ a ~~SDU~~RLC SDU segment;
- be ~~. Length Indicators are~~ included in the PDUs that they refer to.

The size of the~~ Length Indicator~~ "Length Indicator" may be either 7 bits or 15 bits. The value of a~~ Length Indicator~~ "Length Indicator" shall not exceed the values specified in subclauses 11.2.4.2 and 11.3.4.5.

~~A Length Indicator group is a set of Length Indicators that refer to a PDU.~~ The "Length Indicators" which refer to the same PDU ~~that are part of a Length Indicator group must never~~shall:

- not be reordered ~~within the Length Indicator group or~~nor removed ~~from the Length Indicator group~~in case of retransmission;
- be in the same order as the ~~SDU~~RLC SDUs that they refer to.~~.~~

~~If there can be more than one Length Indicator, each specifying the end of an SDU in a PDU, the order of these Length Indicators must be in the same order as the SDUs that they refer to.~~

For AM:

- 7-bit "Length Indicators" shall be used if the "AMD PDU size" is ≤ l26 octets;
- else, 15-bit "Length Indicators" shall be used;
- the size of the "Length Indicator" is always the same for all AMD PDUs, for one RLC entity.

For UM:

- 7-bit "Length Indicators" shall be used if the "largest UMD PDU size" is ≤ l25 octets;
- else 15-bit "Length Indicators" shall be used;
- between modifications of the "largest UMD PDU size", the size of the "Length Indicator" is the same for all UMD PDUs;
- the "Length Indicator" with value "111 1100" if 7-bit "Length Indicator" is used or "111 1111 1111 1100" if 15-bit "Length Indicator" is used shall be used:

-if the parameter Use special LI is configured; and

-if the RLC SDU begins in the beginning of the RLC PDU; and

-if the "Length Indicators" indicating that~~ an~~ a ~~SDU~~RLC SDU ended exactly in the end or one octet short (only when 15-bit "Length Indicators" is used) of the previous RLC PDU is not present.

In the case where the end of the last segment of~~ an~~ a ~~SDU~~RLC SDU exactly ends at the end of a PDU and there is no ~~LI~~"Length Indicator" that indicates the end of the ~~SDU~~RLC SDU:~~,~~

- ~~the next~~ a "Length Indicator" with value "000 0000" if 7-bit "Length Indicator" is used or "000 0000 0000 0000" if 15-bit "Length Indicator" is used, shall be placed as the first ~~Length Indicator~~ "Length Indicator" in the following PDU ~~and have value LI=0~~.

In case this ~~SDU~~RLC SDU was the last one to be transmitted, a RLC PDU may be transmitted, this RLC PDU consists of:~~,~~

- ~~a PDU consisting of an~~ a RLC Header;

- ~~with Li~~a "Length Indicator" with value ~~=0~~ "000 0000" if 7-bit "Length Indicator" is used or "000 0000 0000 0000" if 15-bit "Length Indicator" is used;

- ~~followed by~~ a padding ~~Length Indicator~~ "Length Indicator";

- ~~and~~ padding ~~may be transmitted~~.

In the case where a PDU contains a 15-bit ~~LI~~"Length Indicator" indicating that ~~an~~ a ~~SDU~~RLC SDU ends with one octet left in the PDU, the last octet of this PDU shall:

- be padded by the Sender and ignored by the Receiver though there is no "Length Indicator" indicating the existence of Padding ~~and~~;

- ~~shall~~ not be filled with the first octet of the next ~~SDU~~RLC SDU data.

In the case where 15-bit "Length Indicators" are used for the previous PDU and the last segment of ~~an~~ a RLC SDU is one octet short of exactly filling the PDU~~, and~~:

- if a 15-bit ~~Length Indicator~~ "Length Indicator" is used for the following PDU: ~~then~~

  - the ~~LI~~"Length Indicator" with value ~~LI~~="111 1111 1111 1011" shall be placed as the first "Length Indicator" in the following PDU~~.~~ ;

  - t~~T~~he remaining one octet in the previous PDU shall be padded by the Sender and ignored at the ~~receiver~~Receiver though there is no "Length Indicator" indicating the existence of Padding~~.~~ ;

  - i~~I~~n case this ~~SDU~~RLC SDU was the last one to be transmitted, a PDU consisting of ~~an~~ a RLC Header with ~~LI~~"Length Indicator" with value ="111 1111 1111 1011" followed by a padding ~~Length Indicator~~ "Length Indicator" and padding may be transmitted.

- if a 7-bit ~~Length Indicator~~ "Length Indicator" is used for the following PDU; ~~then~~

  - if RLC is configured for UM mode, the ~~LI~~"Length Indicator" with value ~~LI~~="000 0000" shall be placed as the first ~~Length indicator~~ "Length Indicator" in the following PDU and its SN shall be incremented by 2 before it is transmitted ~~(this can only occur in UM)~~.

Predefined values of the "Length Indicator" are used to indicate padding. The values that are reserved for special purposes are listed in the tables below depending on the size of the "Length Indicator". Only predefined "Length Indicator" values can refer to the padding space~~A PDU that has unused space, to be referred to as padding, shall use a Length Indicator to indicate that this space is used as padding unless the padding size is one octet for PDUs with 15-bit LIs~~. ~~A~~ These values~~padding Length Indicator~~ "Length Indicator"~~must~~ shall only be placed after ~~any~~ all other "Length Indicators" for a PDU.

~~All unused space in a PDU must be located at the end of the PDU, be a homogeneous space and is referred to as padding. Predefined values of the Length Indicator are used to indicate this. The values that are reserved for special purposes are listed in the tables below depending on the size of the Length Indicator. Only predefined Length Indicator values can refer to the padding space.~~

STATUS PDUs can be piggybacked on the AMD PDU by using part or all of the padding space. A ~~Length Indicator~~ "Length Indicator" ~~must~~shall be used to indicate the piggybacked STATUS PDU. This ~~Length Indicator~~ "Length Indicator" takes space from the padding space or piggybacked STATUS PDU and not the PDU data and will always be the last "Length Indicator". Where only part of the padding space is used by a piggybacked STATUS PDU~~, then~~ the end of the piggybacked STATUS PDU is determined by one of the SUFI fields NO_MORE or ACK, thus no additional ~~Length Indicator~~ "Length Indicator" is required to show that there is still padding in the PDU. The padding/piggybacked STATUS PDU predefined "Length Indicators" shall be added after the very last ~~(i.e. there could be more than one SDU that end within a PDU) Length Indicator~~ "Length Indicator" that indicates the end of the last ~~SDU~~RLC SDU segment in the PDU.

If "SDU discard with explicit signalling" is ~~used~~ configured:

- an AMD PDU can contain a maximum number of 15 ~~LI~~"Length Indicators" indicating the end of ~~an~~ 15 corresponding SDUs ~~and~~ ;
- the rest of the AMD PDU space shall be used as padding~~/~~ or as piggybacked STATUS PDU.

~~For AM, 7bit indicators shall be used if the AMD PDU size is ≤ l26 octets. Otherwise 15bit indicators shall be used. For UM, 7bit indicators shall be used if the largest UMD PDU size is ≤ l25 octets. Otherwise 15bit indicators shall be used.~~

~~The length of the Length Indicator only depends on the size of the largest RLC PDU. Between RLC modifications the length of the Length Indicator is the same for all UMD PDUs. The length of the Length Indicator is always the same for all AMD PDUs, for one RLC entity.~~

Length: 7 bits

Bit	Description
0000000	The previous RLC PDU was exactly filled with the last segment of ~~an~~ a RLC SDU and there is no ~~LI~~"Length Indicator" that indicates the end of the ~~SDU~~RLC SDU in the previous RLC PDU.
1111100	UMD PDU: The first data octet in this RLC PDU is the first octet of ~~an~~ a RLC SDU. AMD PDU: Reserved (PDUs with this coding will be discarded by this version of the protocol).
1111101	Reserved (PDUs with this coding will be discarded by this version of the protocol).
1111110	AMD PDU: The rest of the RLC PDU includes a piggybacked STATUS PDU. UMD PDU: Reserved (PDUs with this coding will be discarded by this version of the protocol).
1111111	The rest of the RLC PDU is padding. The padding length can be zero.

Length: 15 bits

Bit	Description
000000000000000	The previous RLC PDU was exactly filled with the last segment of ~~an~~ a RLC SDU and there is no ~~LI~~"Length Indicator" that indicates the end of the ~~SDU~~RLC SDU in the previous RLC PDU.
111111111111011	The last segment of ~~an~~ a RLC SDU was one octet short of exactly filling the previous RLC PDU and there is no ~~LI~~"Length Indicator" that indicates the end of the ~~SDU~~RLC SDU in the previous RLC PDU. The remaining one octet in the previous RLC PDU is ignored.
111111111111100	UMD PDU: The first data octet in this RLC PDU is the first octet of ~~an~~ a RLC SDU. AMD PDU: Reserved (PDUs with this coding will be discarded by this version of the protocol).
111111111111101	Reserved (PDUs with this coding will be discarded by this version of the protocol).
111111111111110	AMD PDU: The rest of the RLC PDU includes a piggybacked STATUS PDU. UMD PDU: Reserved (PDUs with this coding will be discarded by this version of the protocol).
111111111111111	The rest of the RLC PDU is padding. The padding length can be zero.

9.2.2.9 Data field

RLC SDUs or segments of RLC SDUs are mapped to this field in transparent, unacknowledged and acknowledged modes.

Transparent mode data:

- ~~T~~the length of RLC SDUs is not constrained to a multiple of 8 bits,~~.~~
- ~~The RLC SDUs might be segmented.~~if "Segmentation" is configured:

~~The allowed size for the segments shall be determined from the transport formats of the transport channel [4, 8].~~

  - ~~A~~all the RLC PDUs carrying one RLC SDU shall be sent in one ~~transmission time interval~~TTI;~~.~~

- o~~O~~nly segments from one RLC SDU shall be sent in one ~~transmission time interval~~TTI.

- otherwise (~~if~~ "Segmentation~~"~~ is not configured):

  - TMD PDU size is fixed within a single TTI and is equal to the RLC SDU size.

~~NOTE: If segmentation is not used for the transparent mode RLC entity then more than one RLC SDU can be sent in one transmission time interval using one RLC PDU per RLC SDU. The RLC PDUs need, however, to be of the same size due to L1 limitations.~~

Unacknowledged mode data and Acknowledged mode data:

- t~~T~~he length of RLC SDUs is constrained to a multiple of 8 bits;~~.~~
- if a RLC SDU~~s~~ ~~might be~~is segmented, its~~. If possible, the~~ last segment ~~of an SDU~~ shall be concatenated with the first segment of the next ~~SDU~~RLC SDU in order to fill the data field completely and avoid unnecessary padding. The~~ length indicator~~ "Length Indicator" field is used to point the borders between ~~SDU~~RLC SDUs;~~.~~
- f~~F~~or PDUs with 15-bit ~~LI~~"Length Indicators", if~~ an~~ a ~~SDU~~RLC SDU ends with one octet left in a PDU whether the ~~LI~~"Length Indicator" indicating the end of the ~~SDU~~RLC SDU is contained in this PDU or in the next PDU, padding for the last octet of this PDU is necessary and the next ~~SDU~~RLC SDU shall not be concatenated in this PDU. No ~~LI~~"Length Indicator" shall be needed to indicate this kind of one-octet padding.

9.2.2.10 Padding (PAD)

All unused space in a PDU ~~must~~shall be located at the end of the PDUand is referred to as padding. Padding has a length such that the PDU has the required predefined total length.

Padding may have any value and the ~~receiving entity~~Receiver shall disregard it.

9.2.2.11 SUFI

Which SUFI fields to use is implementation dependent, but when a STATUS PDU includes information about which PDUs have been received and which are detected as missing, information shall not be included about PDUs with SN≥VR(H) i.e. PDUs that have not yet reached the ~~receiver~~Receiver. Information about PDUs with SN<VR(R) shall not be given except when this is necessary in order to use the BITMAP SUFI, see subclause 9.2.2.11.5.

Length: variable number of bits.

The SUFI ~~(Super-Field) may~~can includes three sub-fields: type information (type of super-field, e.g. list, bitmap, acknowledgement, etc), length information (providing the length of a variable length field within the following value field) and a value.

Figure 9.7 shows the structure of the super-field. The size of the type sub-field is non-zero but the size of the other sub-fields may be zero.

Type
Length
Value

Figure 9.7: The Structure of a Super-Field

The length of the type field is 4 bits and it may have any of following values.

Bit	Description
0000	No More Data (**NO_MORE**)
0001	Window Size (**WINDOW**)
0010	Acknowledgement (**ACK**)
0011	List (**LIST**)
0100	Bitmap (**BITMAP**)
0101	Relative list (**Rlist)**
0110	Move Receiving Window **(MRW)**
0111	Move Receiving Window Acknowledgement **(MRW_ACK)**
1000-1111	Reserved (PDUs with this encoding are invalid for this version of the protocol)

The length sub-field gives the length of the variable size part of the following value sub-field and the length of it depends on the super-field type. The value sub-field includes the value of the super-field, e.g. the bitmap in case of a BITMAP super-field, and the length is given by the length of the type sub-field.

9.2.2.11.1 The No More Data super-field

The 'No More Data' super-field indicates the end of the data part of a STATUS PDU and is shown in Figure 9.8 below. It shall always be placed as the last SUFI if it is included in a STATUS PDU. All data after this SUFI shall be regarded as padding and shall be neglected.

Type=**NO_MORE**

Figure 9.8: NO_MORE field in a STATUS PDU

9.2.2.11.2 The Acknowledgement super-field

The 'Acknowledgement' super-field consists of a type identifier field (ACK) and a sequence number (LSN) as shown in Figure 9.9 below. The acknowledgement super-field is also indicating the end of the data part of a STATUS PDU. Thus, no 'NO_MORE' super-field is needed in the STATUS PDU when the 'ACK' super-field is present. The ACK SUFI shall always be placed as the last SUFI if it is included in a STATUS PDU. All data after this SUFI shall be regarded as padding and shall be neglected.

Type = **ACK**
LSN

Figure 9.9: The ACK fields in a STATUS PDU

LSN

Length: 12 bits

Acknowledges the reception of all PDUs with "S~~s~~equence N~~n~~umber"~~s~~ < LSN (Last Sequence Number) that are *not* indicated to be erroneous in earlier parts of the STATUS PDU. This means that if the LSN is set to a value greater than VR(R), all erroneous PDUs ~~must~~shall be included in the same STATUS PDU and if the LSN is set to VR(R), the erroneous PDUs can be split into several STATUS PDUs. At the transmitter, if the value of the LSN =< the value of the first error indicated in the STATUS PDU, VT(A) will be updated according to the LSN, otherwise VT(A) will be updated according to the first error indicated in the STATUS PDU. VT(A) is only updated based on STATUS PDUs where ACK SUFI (or MRW_ACK SUFI) is included. The LSN ~~should~~ shall not be set to a value > VR(H) nor < VR(R).

9.2.2.11.3 The Window Size super-field

The 'Window Size' super-field consists of a type identifier (WINDOW) and a window size number (WSN) as shown in Figure 9.10 below. The ~~receiver~~Receiver is always allowed to change the ~~Tx transmitter~~transmission window size of the peer entity during a connection, but the minimum and the maximum allowed value is given by upper layers configuration. The ~~Rx receiver~~reception window size of the ~~receiver~~Receiver is not changed.

Type = **WINDOW**
WSN

Figure 9.10: The WINDOW fields in a STATUS PDU

WSN

Length: 12 bits

The value of VT(WS) to be used by the transmitter. The range of the WSN is [0, 2^{12}-1]. The minimum value of VT(WS) is 1~~, if~~. If WSN is zero, the SUFI shall be discarded by this version of the protocol. The variable VT(WS) is set equal to WSN upon reception of this SUFI. If WSN is greater than Configured_Tx_Window_Size, VT(WS) shall be set equal to Configured_Tx_Window_Size.

9.2.2.11.4 The List super-field

The List Super-Field consists of a type identifier field (LIST), a list length field (LENGTH) and a list of LENGTH number of pairs as shown in Figure 9.11 below:

Type = **LIST**
LENGTH
SN_1
L_1
SN_2
L_2
…
SN_{LENGTH}
L_{LENGTH}

Figure 9.11: The List fields in a STATUS PDU ~~for a list~~

LENGTH

Length: 4 bits

The number of (SN_i , L_i)-pairs in the super-field of type LIST. The value "0000" is invalid and the STATUS PDU is discarded.

SN_i

Length: 12 bits

"Sequence ~~number~~Number" of PDU, which was not correctly received.

L_i

Length: 4 bits

Number of consecutive PDUs not correctly received following PDU with "S~~s~~equence ~~n~~Number" SN_i.

9.2.2.11.5 The Bitmap super-field

The Bitmap Super-Field consists of a type identifier field (BITMAP), a bitmap length field (LENGTH), a first sequence number (FSN) and a bitmap as shown in Figure 9.12 below:

Type = **BITMAP**
LENGTH
FSN
Bitmap

Figure 9.12: The Bitmap fields in a STATUS PDU

LENGTH

Length: 4 bits

The size of the bitmap in octets equals LENGTH+1, i.e. LENGTH="0000" means that the size of the bitmap is one octet and LENGTH="1111" gives the maximum bitmap size of 16 octets.

FSN

Length: 12 bits

The "S~~s~~equence ~~n~~Number" for the first bit in the bitmap. FSN shall not be set to a value lower than VR(R)-7 when the ~~Rx receiver~~reception window size is less than half the maximum RLC AM "S~~s~~equence ~~n~~Number". If the ~~Rx receiver~~reception window size is larger, FSN shall not be set to a value lower than VR(R).

Bitmap

Length: Variable number of octets given by the LENGTH field.

Status of the SNs in the interval [FSN, FSN + (LENGTH+1)*8 - 1] indicated in the bitmap where each position (from left to right) can have two different values (0 and 1) with the following meaning (bit_position$\in$[0,(LENGTH+1)*8 - 1]):

1: SN = (FSN + bit_position) has been correctly received.

0: SN = (FSN + bit_position) has not been correctly received.

9.2.2.11.6 The Relative List super-field

The Relative List super-field consists of a type identifier field (RLIST), a list length field (LENGTH), the first sequence number (FSN) and a list of LENGTH number of codewords (CW) as shown in Figure 9.13 below.

Type = **RLIST**
LENGTH
FSN
CW_1
CW_2
…
CW_{LENGTH}

Figure 9.13: The RList fields in a STATUS PDU

LENGTH

Length: 4 bits

The number of codewords (CW) in the super-field of type RLIST.

FSN

Length: 12 bits

The "S~~s~~equence ~~n~~Number" for the first erroneous PDU in the RLIST, i.e. LENGTH="0000" means that only FSN is present in the SUFI.

CW

Length: 4 bits

The CW consists of 4 bits where the three first bits are part of a number and the last bit is a status indicator and it shall be interpreted as follows:

Code Word	Description
$X_1X_2X_3$ 0	Next 3 bits of the number are $X_1X_2X_3$ and the number continues in the next CW. The most significant bit within this CW is X_1.
$X_1X_2X_3$ 1	Next 3 bits of the number are $X_1X_2X_3$ and the number is terminated. The most significant bit within this CW is X_1. This is the most significant CW within the number.

By default, the number given by the CWs represents a distance between the previous indicated erroneous PDU up to and including the next erroneous PDU.

One special value of CW is defined:

`000 1` ’Error burst indicator’.

The error burst indicator means that the next CWs will represent the number of subsequent erroneous PDUs (not counting the already indicated error position). After the number of errors in a burst is terminated with XXX 1, the next codeword will again by default be the least significant bits (LSB) of the distance to the next error.

If the last CW, as indicated by the value of the LENGTH field, does not contain a "1" in its rightmost position, or the last CW, as indicated by the value of the LENGTH field does contain a "1" in its rightmost position, but is a special "error burst indicator" CW, the encoding of the RLIST SUFI is invalid, and the STATUS PDU is discarded.

9.2.2.11.7 The Move Receiving Window Acknowledgement super-field

The ’Move Receiving Window Acknowledgement’ super-field acknowledges the reception of a MRW SUFI. The format is given in Figure 9.14 below.

Type = **MRW_ACK**
N
SN_ACK

Figure 9.14: The MRW-ACK fields in a STATUS PDU

N

Length: 4 bits

The N field shall be set equal to the N_{LENGTH} field in the received MRW SUFI if the SN_ACK field is equal to the SN_MRW_{LENGTH} field. Otherwise N shall be set to 0.

With the aid of this field in combination with the SN_ACK field, it can be determined if the MRW_ACK corresponds to a previously transmitted MRW SUFI.

SN_ACK

Length: 12 bits

The SN_ACK field indicates the updated value of VR(R) after the reception of the MRW SUFI. With the aid of this field in combination with the N field, it can be determined if the MRW_ACK corresponds to a previously transmitted MRW SUFI.

9.2.2.11.8 The Move Receiving Window (MRW) super-field

The ’Move Receiving Window’ super-field is used to request the ~~RLC r~~Receiver to move its ~~receiving~~ reception window and optionally to indicate the set of discarded ~~SDU~~RLC SDUs, as a result of~~ an~~ a ~~SDU~~RLC SDU discard in the ~~RLC transmitter~~Sender. The format is given in Figure 9.15 below.

Type = **MRW**
LENGTH
SN_MRW$_1$
SN_MRW$_2$
...
SN_MRW$_{LENGTH}$
N$_{LENGTH}$

Figure 9.15: The MRW fields in a STATUS PDU

LENGTH

Length: 4 bits

The number of SN_MRW$_i$ fields in the super-field of type MRW.

The values "0001" through "1111" indicate 1 through 15 SN_MRW$_i$ respectively. The value "0000" indicates that one SN_MRW$_i$ field is present and that the ~~SDU~~RLC SDU to be discarded in the R~~r~~eceiver extends above the configured ~~Tx transmitting~~transmission window in the ~~transmitter~~Sender.

SN_MRW$_i$

Length: 12 bits

When Send MRW is configured,~~ an~~ a SN_MRW$_i$ shall be used to indicate the end of each discarded ~~SDU~~RLC SDU, i.e. the number of SN_MRW$_i$ fields shall equal the number of ~~SDU~~RLC SDUs discarded by that MRW SUFI. When "Send MRW" is not configured, SN_MRW$_i$ shall be used to indicate the end of the last ~~SDU~~RLC SDU to be discarded in the ~~receiver~~ Receiver and they may optionally be used to indicate the end of other discarded ~~SDU~~RLC SDUs. SN_MRW$_i$ is the "S~~s~~equence ~~n~~Number" of the PDU that contains the ~~LI~~"Length Indicator" of the i:th ~~SDU~~RLC SDU to be discarded in the ~~receiver~~ Receiver (except for SN_MRW$_{LENGTH}$ when N$_{LENGTH}$ = 0, see definition of N$_{LENGTH}$). The order of the SN_MRW$_i$ shall be in the same sequential order as the ~~SDU~~RLC SDUs that they refer to.

Additionally SN_MRW$_{LENGTH}$ requests the ~~RLC r~~Receiver to discard all PDUs with "S~~s~~equence N~~n~~umber" < SN_MRW$_{LENGTH}$, and to move the ~~receiving~~ reception window accordingly. In addition, when N$_{LENGTH}$ > 0, the R~~r~~eceiver has to discard the first N$_{LENGTH}$ ~~LI~~"Length Indicators" and the corresponding data octets in the PDU with "S~~s~~equence N~~n~~umber" SN_MRW$_{LENGTH}$.

N$_{LENGTH}$

Length: 4 bits

N$_{LENGTH}$ is used together with SN_MRW$_{LENGTH}$ to indicate the end of the last ~~SDU~~RLC SDU to be discarded in the ~~receiver~~Receiver.

N$_{LENGTH}$ indicates which ~~LI~~"Length Indicator" in the PDU with "S~~s~~equence N~~n~~umber" SN_MRW$_{LENGTH}$ corresponds to the last ~~SDU~~RLC SDU to be discarded in the ~~receiver~~Receiver. N$_{LENGTH}$ = 0 indicates that the last ~~SDU~~RLC SDU ended in the PDU with "S~~s~~equence N~~n~~umber" SN_MRW$_{LENGTH}$ –1 and that the first data octet in the PDU with "S~~s~~equence N~~n~~umber" SN_MRW$_{LENGTH}$ is the first data octet to be reassembled next.

9.2.2.12 Reserved 2 (R2)

Length: 1 bit

This bit in the Piggybacked STATUS PDU is used to ~~achieve octet alignment~~make the Piggybacked STATUS PDU a multiple of 8 bits in length and for this purpose it is coded as 0. Otherwise the PDU is treated as invalid and hence shall be discarded by this version of the protocol.

9.2.2.13 Reset Sequence Number (RSN)

Length: 1 bit

This field is used to indicate the sequence number of the transmitted RESET PDU. If this RESET PDU is a retransmission of the original RESET PDU then the retransmitted RESET PDU would have the same ~~sequence~~

~~number~~RSN value as the original RESET PDU. Otherwise it will have the next ~~reset sequence number~~RSN value. The initial value of this field is zero. The value of this field shall be reinitialised when the RLC is re-established. It shall not be reinitialised when the RLC is reset.

9.2.2.14 Hyper Frame Number Indicator (HFNI)

Length: 20 bit

This field is used to indicate the hyper frame number (HFN) to the peer entity. With the aid of this field the HFN in UE and UTRAN can be synchronised.

9.3 Protocol states

The content presented in this subclause is intended to support the definition of the RLC protocol states only, and is not meant to specify or constrain the implementation of the protocol.

9.3.1 State model for transparent mode entities

Figure 9.16 illustrates the state model for transparent mode RLC entities (both transmitting ~~sending~~ and receiving). A transparent mode entity can be in one of the following states.

9.3.1.1 ~~Null~~ NULL State

In the ~~null~~ NULL state, the RLC entity does not exist and therefore it is not possible to transfer any data through it.

Upon reception of a CRLC-CONFIG-Req from upper layers indicating establishment, the RLC entity ~~should~~:

- ~~be~~is created; and
- enters the ~~transparent data transfer ready~~DATA_TRANSFER_READY state ~~is entered~~.

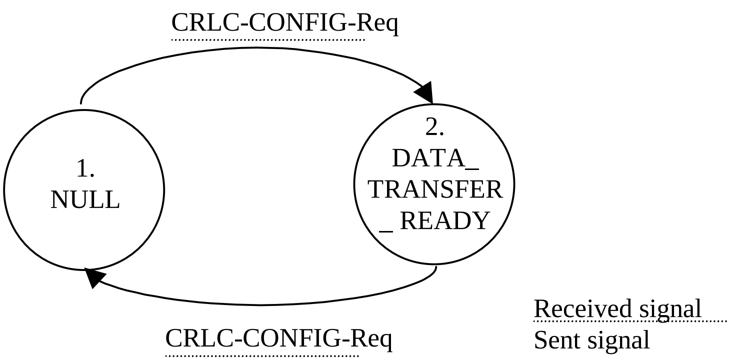
9.3.1.2 ~~Transparent Data Transfer Ready~~DATA_TRANSFER_READY State

In the ~~transparent data~~ DATA_TRANSFER_READY~~transfer ready~~ state, transparent mode data can be exchanged between the entities according to subclause 11.1.

Upon reception of a CRLC-CONFIG-Req from upper layer indicating release, the RLC entity ~~should~~:

- enters the NULL state; and
- ~~be~~is considered as being terminated.~~; and~~

~~-	enter the null NULL state is entered.~~



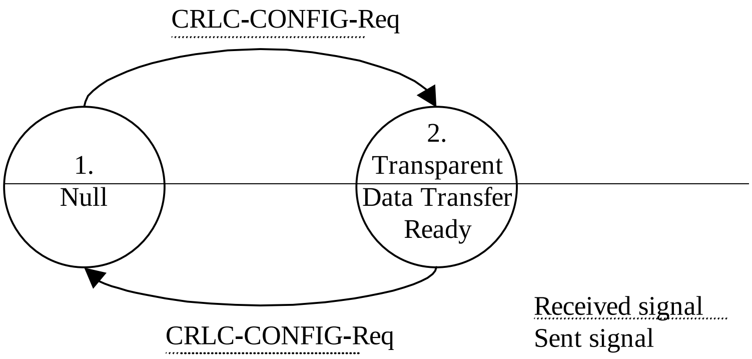


Figure 9.16: The state model for transparent mode entities

9.3.2 State model for unacknowledged mode entities

Figure 9.17 illustrates the state model for unacknowledged mode RLC entities (both transmitting ~~sending~~ and receiving). An unacknowledged mode entity can be in one of the following states.

9.3.2.1 ~~Null~~ NULL State

In the ~~null~~ NULL state, the RLC entity does not exist and therefore it is not possible to transfer any data through it.

Upon reception of a CRLC-CONFIG-Req from upper layer indicating establishment, the RLC entity ~~should~~:

- ~~be~~is created; and
- enters the ~~unacknowledged data transfer ready~~DATA_TRANSFER_READY state ~~is entered~~.

9.3.2.2 ~~Unacknowledged Data Transfer Ready~~DATA_TRANSFER_READY State

In the ~~unacknowledged data transfer ready~~DATA_TRANSFER_READY state, unacknowledged mode data can be exchanged between the entities according to subclause 11.2.

Upon reception of a CRLC-CONFIG-Req from upper layer indicating release, the RLC entity ~~should~~:

- enters the NULL state; and
- ~~be~~is considered as being terminated.~~; and~~

~~- enter and the null NULL state is entered.~~

Upon reception of a CRLC-CONFIG-Req from upper layer indicating modification, the RLC entity ~~should~~:

- stays in the DATA_TRANSFER_READY state;
- modif~~y~~ies only the protocol parameters and timers as indicated by upper layers.~~; and~~

~~- keep other protocol parameters, the protocol variables and the protocol timers unchanged.~~

Upon reception of a CRLC-SUSPEND-Req from upper layers, the RLC entity ~~should~~:

- enters the LOCAL_SUSPEND state.

9.3.2.3 ~~Local Suspend~~LOCAL_SUSPEND State

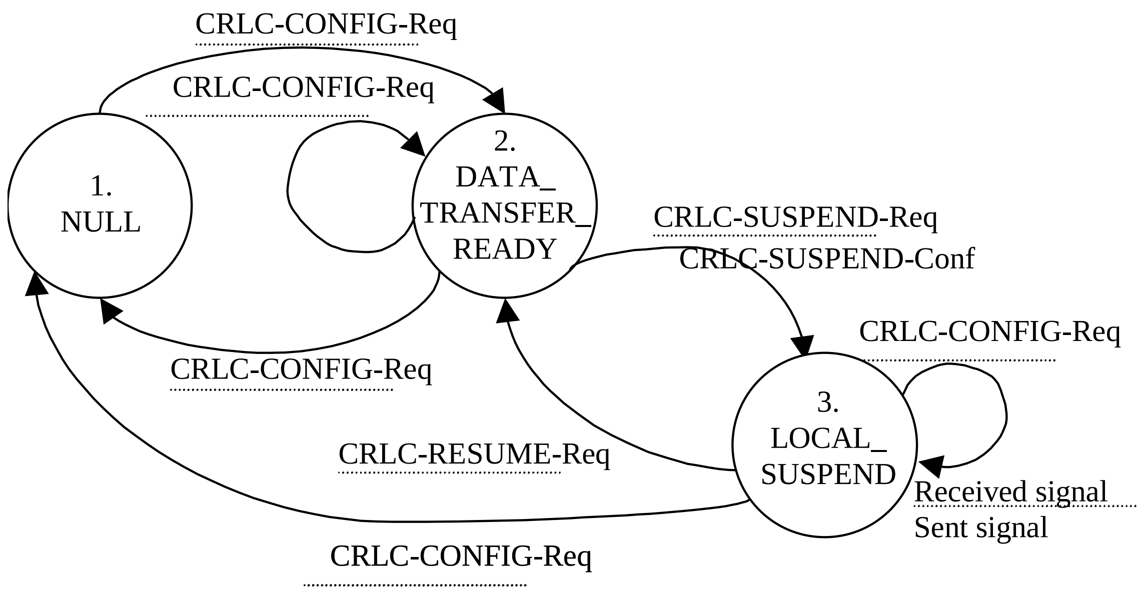
~~Upon reception of a CRLC-SUSPEND-Req from upper layers, the RLC entity is suspended and the Local Suspend state is entered.~~ In the ~~Local Suspend~~LOCAL_SUSPEND state, the RLC entity ~~shall~~ is suspended, i.e. it does not send ~~RLC-UMD~~ PDUs with SN greater than or equal to certain specified value (see subclause 9.7.5).~~≥VT(US)+N.~~

Upon reception of a CRLC-RESUME-Req from upper layers, the RLC entity ~~should~~:

- ~~the RLC entity is resumed and~~enters the ~~Data Transfer Ready~~DATA_TRANSFER_READY state; and ~~is entered.~~
- resumes the data transmission ~~normally~~.

Upon reception of a CRLC-CONFIG-Req from upper layer indicating modification, the RLC entity ~~should~~:

- stays in the LOCAL_SUSPEND state;
- modif~~y~~ies only the protocol parameters and timers as indicated by upper layers.~~; and~~
- ~~keep other protocol parameters, the protocol variables and the protocol timers unchanged.~~



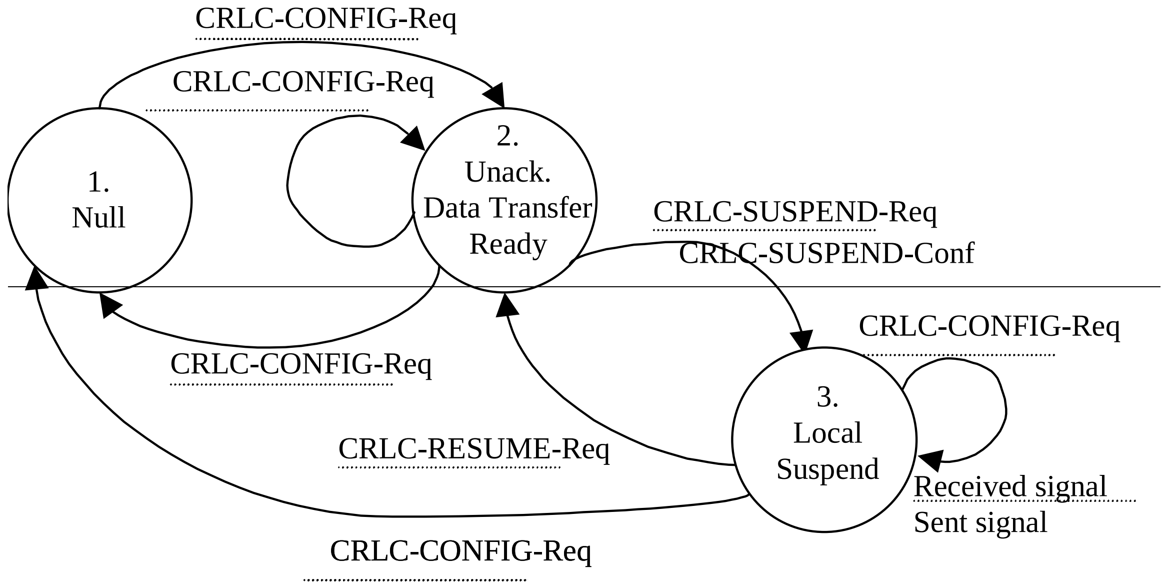


Figure 9.17: The state model for unacknowledged mode entities

9.3.3 State model for acknowledged mode entities

Figure 9.18 illustrates the state model for the acknowledged mode RLC entity (both transmitting and receiving). An acknowledged mode entity can be in one of the following states.

9.3.3.1 ~~Null~~ NULL State

In the ~~null~~ NULL state, the RLC entity does not exist and therefore it is not possible to transfer any data through it.

Upon reception of a CRLC-CONFIG-Req from upper layer indicating ~~(re)~~establishment, the RLC entity~~ should~~:

- ~~be~~is created; and
- enters the ~~acknowledged data transfer ready~~ DATA_TRANSFER_READY state~~ is entered~~.

9.3.3.2 ~~Acknowledged Data Transfer Ready~~DATA_TRANSFER_READY State

In the ~~acknowledged data transfer ready~~DATA_TRANSFER_READY state, acknowledged mode data can be exchanged between the entities according to subclause 11.3.

Upon reception of a CRLC-CONFIG-Req from upper layer indicating release, the RLC entity~~ should~~:

- enters the NULL state; and
- ~~be~~is considered as being terminated.~~; and~~
- ~~enter the null NULL state is entered.~~

Upon detection of an initiating condition for the RLC reset procedure described in subclause 11.4.2~~errors in the protocol~~, the RLC entity~~ should~~:

- ~~sends a RESET PDU to its peer~~initiates the RLC reset procedure (see subclause 11.4); and
- enters the ~~reset pending~~RESET_PENDING state.

Upon reception of a RESET PDU, the RLC entity ~~resets the protocol (see~~ responds according to subclause 11.4.3.~~), sets the hyper frame number HFN (DL HFN when the RESET is received in UE or UL HFN when the RESET is received in UTRAN) equal to the HFNI field in the RESET PDU; and responds to the peer entity with a RESET ACK PDU.~~

Upon reception of a RESET ACK PDU, the RLC entity takes no action.

Upon reception of CRLC-SUSPEND-Req from upper layer, the RLC entity ~~should~~is suspended and enters the ~~local suspend~~LOCAL_SUSPEND state~~ is entered~~.

9.3.3.3 ~~*Reset* Pending~~RESET_PENDING State

In the ~~reset pending~~ RESET_PENDING state, the entity waits for a response from its peer entity and no data can be exchanged between the entities.

Upon reception of a CRLC-CONFIG-Req from upper layer indicating release, the RLC entity~~ should~~:

- enters the NULL state; and
- ~~be~~is considered as being terminated.

~~; and~~

- ~~enter the null~~NULL ~~state is entered~~.

Upon reception of a RESET ACK PDU with the same RSN value as in the corresponding RESET PDU, the RLC entity ~~should~~:

- ~~respond~~acts ~~resets the protocol (see~~according to subclause 11.4.4; and~~), sets the hyper frame number HFN (DL HFN when the RESET ACK is received in UE or UL HFN when the RESET ACK is received in UTRAN) equal to the HFNI field in the RESET ACK PDU and~~
- enters the ~~acknowledged data transfer ready~~DATA_TRANSFER_READY state.

Upon reception of a RESET ACK PDU with a different RSN value as in the corresponding RESET PDU, the RLC entity~~ should~~:

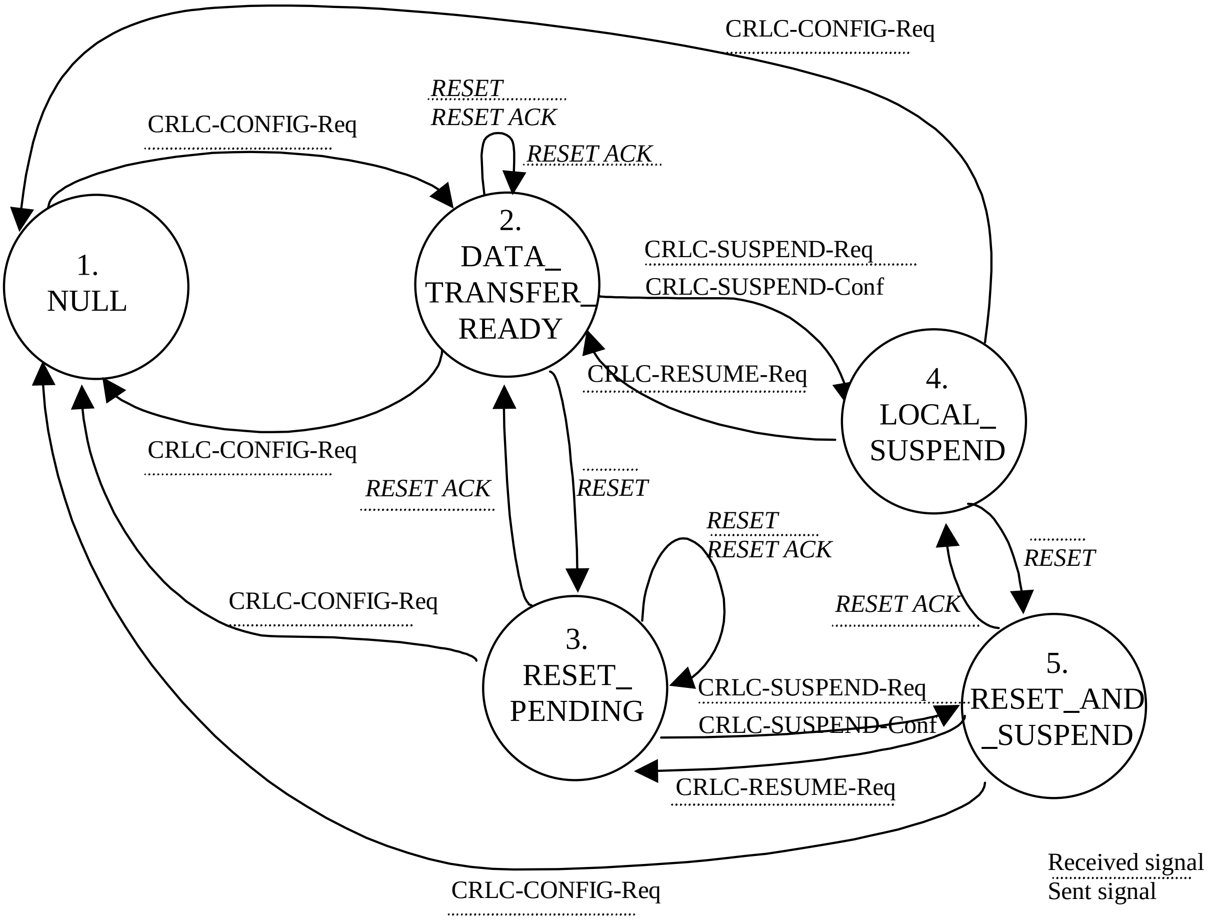
- discards the RESET ACK PDU (see subclause 11.4.4); and
- stays in the RESET_PENDING state.~~ is discarded.~~

Upon reception of a RESET PDU, the RLC entity~~ should~~:

- responds according to~~resets the protocol (see~~ subclause 11.4.3; and~~), sets the hyper frame number HFN (DL HFN when the RESET is received in UE or UL HFN when the RESET is received in UTRAN) equal to the HFNI field in the RESET PDU, sends a RESET ACK PDU and~~
- stays in the ~~reset pending~~RESET_PENDING state.

Upon reception of CRLC-SUSPEND-Req from ~~the upper layer~~upper layer, the RLC entity~~ should~~:~~is suspended and~~

- enters the ~~reset and suspend~~RESET_AND_SUSPEND state~~ is entered~~.


CRLC-CONFIG-Req
RESET
RESET ACK
CRLC-CONFIG-Req
RESET ACK
1. NULL
2. DATA_ TRANSFER_ READY
CRLC-SUSPEND-Req
CRLC-SUSPEND-Conf
CRLC-RESUME-Req
4. LOCAL_ SUSPEND
CRLC-CONFIG-Req
RESET ACK
RESET
RESET
RESET ACK
RESET
RESET ACK
CRLC-CONFIG-Req
3. RESET_ PENDING
CRLC-SUSPEND-Req
CRLC-SUSPEND-Conf
5. RESET_AND _SUSPEND
CRLC-RESUME-Req
CRLC-CONFIG-Req
Received signal
Sent signal

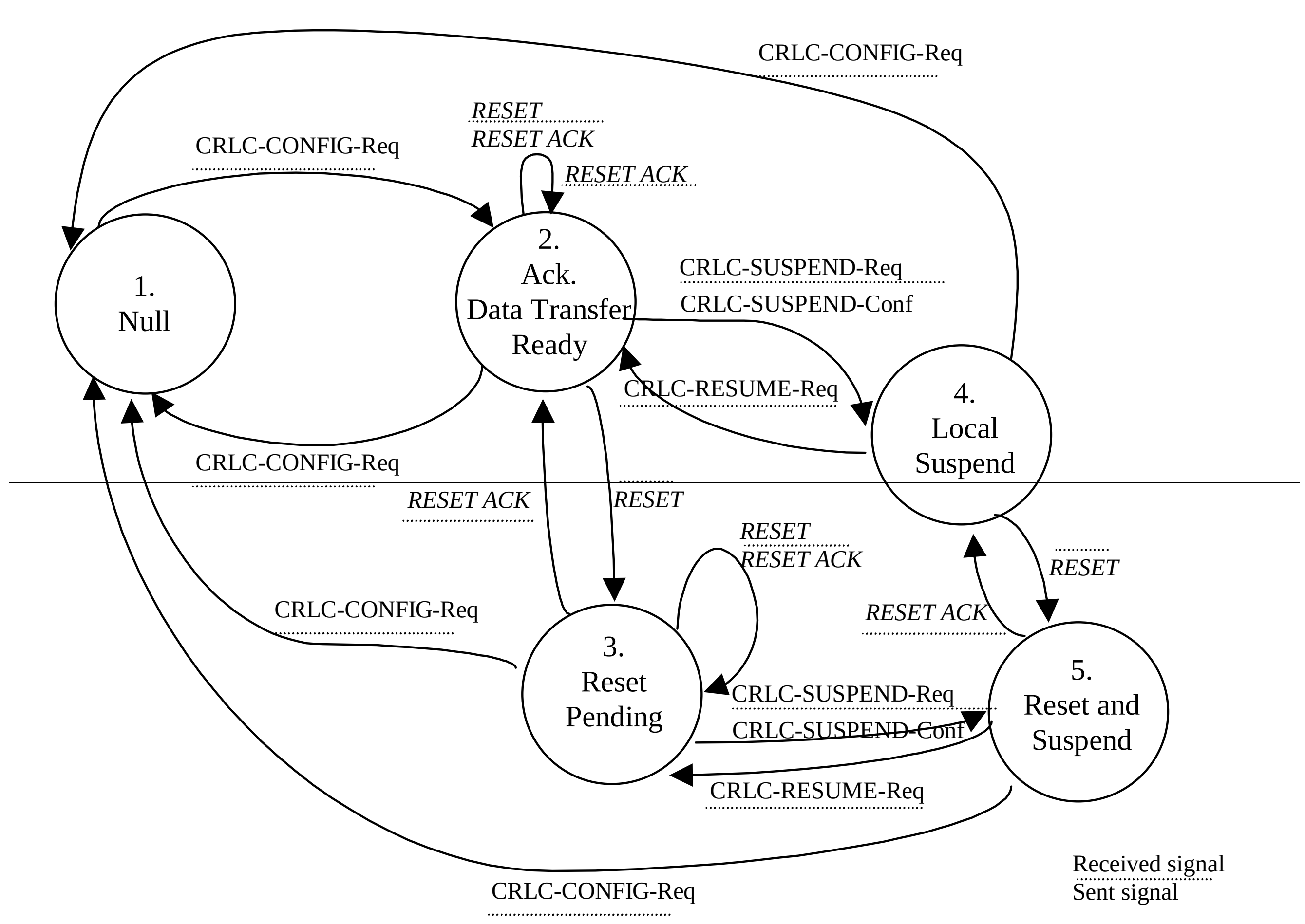


Figure 9.18: The state model for the acknowledged mode entities

9.3.3.4 ~~Local Suspend~~LOCAL_SUSPEND State

In the ~~Local Suspend~~LOCAL_SUSPEND state, the RLC entity is suspended, i.e. it does ~~shall~~ not send ~~an RLC~~-AMD PDUs with SN greater than or equal to certain specified value (see subclause 9.7.5).~~≥VT(S)+N, where VT(S) is the value of the send state variable when the CRLC-SUSPEND-Req with parameter N was received.~~

Upon reception of CRLC-RESUME-Req from upper layers in this state, the RLC entity ~~should~~:

- ~~is~~ resumes the data transmission ~~normally~~; and
- enters the ~~Acknowledged Data Transfer Ready~~DATA_TRANSFER_READY state ~~is entered~~.

Upon reception of CRLC-CONFIG-Req from upper layers indicating release, the RLC entity ~~should~~:

- enters the NULL state; and
- ~~be~~is considered as being terminated.

~~; and~~

~~- enter the null NULL state is entered.~~

Upon detection of an initiating condition for RLC reset procedure described in subclause 11.4.2~~errors in the protocol~~, the RLC entity ~~should~~:

- ~~sends a RESET PDU to its peer~~initiates the RLC reset procedure (see subclause 11.4); and

- enters the ~~reset and suspend~~ RESET_AND_SUSPEND state.

9.3.3.5 ~~Reset and Suspend~~RESET_AND_SUSPEND State

In the ~~reset and suspend~~RESET_AND_SUSPEND state, the entity waits for a response from its peer entity or a primitive (CRLC-RESUME-Req) from its upper layer and no data can be exchanged between the entities.

Upon reception of CRLC-CONFIG-Req from upper layer indicating release, the RLC entity ~~should~~:

- enters the NULL state; and

- ~~be~~is considered as being terminated.~~; and~~

~~- enter the null NULL state is entered~~.

Upon reception of a RESET ACK PDU with the same RSN value as in the corresponding RESET PDU, the RLC entity ~~should~~:

- ~~respond~~acts according to ~~resets the protocol (see~~ subclause 11.4.4; and~~), sets the hyper frame number HFN (DL HFN when the RESET ACK is received in UE or UL HFN when the RESET ACK is received in UTRAN) equal to the HFNI field in the RESET ACK and~~

- enters the ~~local suspend~~LOCAL_SUSPEND state.

Upon reception of CRLC-RESUME-Req from upper layer in this state, the RLC entity ~~should~~:

- ~~be~~is resumed, i.e. releases the suspend constraint; and

- enters the ~~reset pending~~RESET_PENDING state~~ is entered~~.

9.4 State variables

The state variables defined in this subclause are normative.

This sub-clause describes the state variables used in AM and UM in order to ~~the~~ specify~~ication of~~ the peer-to-peer protocol. All state variables are non-negative integers. ~~PDUs are sequentially and independently numbered and may have the value 0 through n minus 1 (where n is the modulus of the sequence numbers). The modulus equals 212 for AM and 27 for UM; the s~~UMD and AMD PDUs are numbered by modulo integer sequence numbers (SN), cycling~~e~~ through the ~~entire range~~field: 0 ~~through~~to 2^{12} – 1 for AM and 0 ~~through~~ to 2^{7} – 1 for UM. All arithmetic operations contained in this specification on VT(S), VT(A), VT(MS), VR(R), VR(H) and VR(MR)~~the following state variables and sequence numbers contained in this specification~~ are affected by the AM modulus. All arithmetic operations contained in this specification on ~~: VT(S), VT(A), VT(MS), VR(R), VR(H), VR(MR),~~ VT(US) and VR(US) are affected by the UM modulus. When performing arithmetic comparisons of state variables or SN values a modulus base shall be used. This modulus base is subtracted (within the appropriate field) from all the values involved and then an absolute comparison is performed. ~~a~~At the ~~s~~Sender, VT(A) and VT(US) ~~are shallwill~~shall be assumed to be the modulus base in AM and UM respectively. ~~When performing~~At the Receiver, arithmetic comparisons of variables or SN values~~ at the r~~R~~eceiver~~, VR(R) and VR(US) ~~are shallwill~~shall be assumed to be the modulus base in AM and UM respectively.

The RLC ~~shallwill~~shall maintain~~s~~ the following state variables ~~at~~ in the ~~transmittertransmitting side~~Sender.

a) VT(S) - Send state variable.

~~The sequence number~~This state variable contains the SN of the next AMD PDU to be transmitted for the first time (i.e. excluding retransmitted PDUs~~ssion~~). It ~~is shallwill~~shall be updated after ~~transmission~~ the aforementioned PDU is ~~of a PDU~~, ~~which includes not earlier~~ transmitted ~~PDUs and~~ or after transmission of a MRW SUFI which includes SN_MRW_{LENGTH} >VT(S) (see subclause 11.6).

The initial value of this variable is 0.

b) VT(A) - Acknowledge state variable.

~~The sequence number~~This state variable contains the SN following ~~of~~ the SN of the ~~next~~last in-sequence acknowledged PDU ~~expected to be acknowledged, which~~ This forms the lower edge of the window of acceptable acknowledgements. VT(A) ~~is shallwill~~shall be updated based on the receipt of a STATUS PDU including an ACK (see subclause 9.2.2.11.2) and/or a MRW_ACK ~~super-field~~SUFI (see subclause 11.6).

The initial value of this variable is 0. For the purpose of initialising the protocol, this value shall be assumed to be the first SN following the last in-sequence acknowledged PDU.

c) VT(DAT).

This state variable counts the number of times a PDU has been transmitted. There ~~is will~~shall be one VT(DAT) for each PDU and ~~it is~~each ~~will~~shall be incremented ~~each~~ every time the corresponding PDU is transmitted.

The initial value of this variable is 0.

d) VT(MS) - Maximum Send state variable.

This state variable contains the ~~sequence number~~SN of the first PDU ~~that may be rejected by thenot allowed~~that ~~constrained~~can be rejected by ~~not to be sent by the transmitter window set by by~~ the peer R~~r~~eceiver ~~[i.e. the receiver will allow up to VT(MS) – 1]~~, VT(MS) = VT(A) + VT(WS). This value represents the upper edge of the transmit~~ter~~ window. The transmitter ~~shall will~~shall not transmit ~~a~~ PDUs with SN ≥ VT(MS) unless VT(S) ≥ VT(MS). ~~VT(MS) is updated when either VT(A) or VT(WS) is updated. T,~~In that case, the PDU with SN = VT(S) - 1 can also be transmitted ~~also when VT(S) ≥ VT(MS)~~. VT(MS) ~~shallwill~~shall be updated when VT(A) or VT(WS) is updated.

The initial value of this variable is Configured_Tx_Window_size.

e) VT(US) – UM data state variable.

This state variable ~~gives~~ contains the ~~sequence number~~SN of the next UMD PDU to be transmitted. It ~~is~~ shall be ~~updated~~ incremented by 1 each time a UMD PDU is transmitted.

The initial value of this variable is 0.

f) VT(PDU).

This state variable is used when the "poll every Poll_PDU PDU" polling trigger is configured ~~function is used~~. It ~~is shallwill~~shall be incremented ~~with~~ by 1 for each PDU that is transmitted~~. It should be incremented for~~, including both new and retransmitted PDUs. When it ~~reache~~becomes equal~~s~~ to the value Poll_PDU, a new poll ~~is will~~shall be transmitted and the state variable ~~is will~~shall be set to zero.

The initial value of this variable is 0.

g) VT(SDU).

This state variable is used when the "poll every Poll_SDU SDU~~-~~" polling trigger is configured~~function is used~~. It ~~is shallwill~~shall be incremented ~~with~~ by 1 for ~~each~~ a given SDU ~~that is~~when all the PDUs carrying a part of this SDU have been transmitted at least once. When it ~~reache~~becomes equal~~s~~ to the value Poll_SDU a new poll ~~is will~~shall be transmitted and the state variable ~~is will~~shall be set to zero. The ~~p~~"Polling bit" ~~should will~~shall be set to "1" in the first transmission of the PDU that contains the last segment of the SDU.

The initial value of this variable is 0.

h) VT(RST) - Reset state variable.

~~It~~ This state variable is used to count the number of times a RESET PDU is transmitted before the reset procedure is completed. VT(RST) ~~is will~~shall be incremented ~~with~~ by 1 each time a RESET PDU is transmitted. VT(RST) ~~is will~~shall only be reset ~~only~~ upon the reception of a RESET ACK PDU, i.e. VT(RST) ~~is will~~shall not be reset when a~~n~~ RLC reset ~~occurs which was~~ initiated ~~from~~ by the peer RLC entity occurs.

The initial value of this variable is 0.

i) VT(MRW) – MRW command send state variable.

~~It~~ This state variable is used to count the number of times a MRW command is transmitted. VT(MRW) is incremented ~~with~~ by 1 each time a~~n~~ MRW ~~command~~ SUFI is transmitted. VT(MRW) ~~is will~~shall be reset when the SDU discard with explicit signalling procedure is terminated.

The initial value of this variable is 0.

j) VT(WS) – Transmitter window size state variable.

This state variable contains the size that ~~shall will~~shall be used for the transmitter window. VT(WS) ~~is~~ shall be set equal to the WSN field when the transmitter receives a STATUS PDU including a ~~Window Size~~WINDOW ~~super-field~~SUFI.

The initial value of this variable is Configured_Tx_Window_size.

The RLC ~~shallwill~~shall maintain~~s~~ the following state variables ~~at~~ in the ~~receiverreceiving side~~Receiver:

a) VR(R) - Receive state variable.

This state variable contains the ~~sequence number~~SN following that of the last ~~of the next~~ in-sequence PDU ~~expected to be~~ received. It ~~is will~~shall be ~~set equal to SNmax+1~~ updated upon the receipt of the ~~next in-sequence~~ PDU with SN equal to VR(R)~~, where SNmax is the sequence number of the highest received in-sequence PDU~~.

The initial value of this variable is 0. For the purpose of initialising the protocol, this value shall be assumed to be the first SN following the last in-sequence received PDU.

b) VR(H) - Highest expected state variable.

This state variable contains the ~~sequence number~~SN following~~ of~~ the highest ~~expected~~ SN of any received PDU. When a PDU is received with SN x such that VR(H)≤x<VR(MR), ~~T~~this state variable ~~is shallwill~~shall be set equal to x~~SN~~+1 ~~only when a new PDU is received with VR(MR)>SN≥VR(H)~~.

The initial value of this variable is 0.

c) VR(MR) - Maximum acceptable Receive state variable.

This state variable contains the ~~sequence number~~SN of the first PDU ~~not allowed~~that ~~shallwill~~shall be rejected by the ~~receiver~~Receiver ~~[i.e. the receiver will allow up to VR(MR) – 1]~~, VR(MR) = VR(R) + Configured_Rx_Window_Size. ~~The receiver shall discard PDUs with SN ≥ VR(MR).~~

d) VR(US) - Receiver Send Sequence state variable.

This state variable contains the ~~sequence number~~SN following that of the ~~next~~last PDU ~~to be~~ received. When a PDU with SN equal to x is received, ~~It~~the state variable shall set equal to ~~SN~~ x + 1 ~~upon reception of a PDU~~.

The initial value of this variable is 0.

e) VR(EP) - Estimated PDU Counter state variable.

This state variable contains the number of PDUs that ~~should yet be received yet~~whose re-transmission is still expected as a consequence of the transmission of the latest status report. ~~In acknowledged mode, this state variable is updated at the end of each transmission time interval. I~~At the end of each ~~transmission time interval~~TTI it is decremented by the total number of PDUs that ~~should have been~~were received during that time~~e transmission time interval~~. ~~If VR(EP) is equal to zero, then check if all PDUs requested for retransmission in the latest status report have been received.~~

9.5 Timers

The timers defined in this subclause are normative. The ~~T~~timers ~~will~~shall be considered active from the time they are started until the time they either expire or are stopped.

a) Timer_Poll.

This timer ~~is will~~shall only be used when ~~the poll timer trigger is used~~so configured by ~~higher layersthe upper layer~~upper layers. The ~~initial value~~value of the timer is signalled by ~~higher layersthe upper layer~~upper layers. In the UE ~~It is~~ this timer ~~will~~shall be started when the successful or unsuccessful transmission of a PDU containing a poll is indicated by ~~lower layers~~lower layer ~~(in UE)~~. In UTRAN ~~Or~~it ~~willshall~~should be started when a PDU containing a poll is submitted to ~~lower layers~~lower layer~~lower layer (in UTRAN)~~. If x is the value of the state variable VT(S) at the time the poll was submitted to ~~lower layers~~lower layer, ~~T~~the timer ~~is will~~shall be ~~stopped~~resetstopped ~~when~~upon receiving ~~a STATUS PDU that~~ :

- ~~contains an~~ acknowledgements ~~(positive or negative) of~~for all the AMD PDUs with SN up to and including ~~VT(S)~~x-1; or
- ~~at the time the poll was submitted to lower layer, or when~~ a negative acknowledgement ~~of~~ for the ~~same~~ PDU with SN = x-1~~carrying the polling bit~~. ~~is received.~~

~~The value of the timer is signalled by upper layers.~~

If the timer expires and no STATUS PDU fulfilling the criteria above has been received: ~~or if a new poll is triggered while the timer is active,~~

- the ~~receiver~~Receiver ~~is~~ ~~will~~shall be polled once more; ~~(either by the transmission of a PDU which was not yet sent, or by a retransmission), and~~
- the timer ~~is~~ ~~will~~shall be restarted; ~~at the time specified above, with a~~ and
- the new value of VT(S)~~-1~~ ~~will~~shall be saved.

If a new poll is sent when the timer is ~~running~~ active the timer ~~is~~ shall be restarted at the time specified above, ~~with~~ and ~~a~~the ~~new~~ value of VT(S)~~-1~~ shall be saved.

b) Timer_Poll_Prohibit.

This timer ~~is~~ ~~will~~shall only be used when so configured by ~~higher layersthe upper layer~~upper layers. ~~the poll prohibit function is used~~. It is used to prohibit transmission of polls within a certain period. The ~~initial value~~value of the timer is signalled by ~~higher layersthe upper layer~~upper layers. ~~The~~

In the UE ~~timer~~this timer shall be started when the successful or unsuccessful transmission of a PDU containing a poll is indicated by ~~lower layers~~lower layer~~lower layer~~. ~~(in UE) or~~ In UTRAN it ~~willshall~~should be started when a PDU containing a poll is submitted to ~~lower layers~~lower layer~~lower layer~~ ~~(in UTRAN)~~. ~~The prohibit time is calculated from the time a PDU containing a poll is submitted to lower layer until the timer has expired.~~ If ~~another~~~~A~~

From the time a poll is triggered ~~whilst the timer is active~~until the timer expires ~~before the timer expires~~, polling is prohibited. If another poll is triggered while polling is prohibited, its transmission shall be delayed until the timer ~~prohibit time~~ expires (see subclause 9.7.1) ~~if a poll is triggered during the prohibit time~~. Only one poll shall be transmitted when Timer_Poll_Prohibit ~~the prohibit time~~ expires even if several polls were triggered in the meantime. ~~during the prohibit time.~~ This timer ~~will~~shall not be ~~stopped~~ affected by ~~a~~ the reception~~ived~~ of STATUS PDUs. ~~The value of the timer is signalled by upper layers.~~

When Timer_Poll_Prohibit is not configured by ~~the upper layer~~upper layers, polling is never prohibited.

c) Timer_EPC.

This timer ~~should~~ ~~will~~shall only be used when the EPC function is ~~used~~ configured by ~~higher layersthe upper layer~~upper layers. ~~and i~~It is meant to account~~s~~ for the roundtrip delay, i.e. the time between the transmission of a status report and the reception of ~~when~~ the first retransmitted PDU. ~~should be received after a status report has been sent.~~ The ~~initial value~~value of the timer is signalled by ~~higher layersthe upper layer~~upper layers.

~~The timer~~ In the UE ~~is~~this timer ~~will~~shall be started when the successful or unsuccessful transmission of the first STATUS PDU of a status report is indicated by ~~lower layers~~lower layer~~lower layer~~ ~~(in UE)~~ . In UTRAN it ~~willshall~~should be started when ~~or~~ the first STATUS PDU of a status report is submitted to ~~lower layers~~lower layer~~lower layer~~ ~~(in UTRAN)~~. ~~and w~~Only after Timer_EPC ~~hen it~~ expires ~~will~~shall VR(EP) be decremented as described in clause 9.7.4.~~can start its counting-down process (see subclause 9.7.4).~~ ~~The value of the timer is signalled by upper layers.~~

d) Timer_Discard.

This timer ~~is~~ ~~will~~shall be used ~~for the~~when timer based SDU discard is configured by ~~higher layersthe upper layer~~upper layers~~function~~. The ~~initial value~~value of the timer is signalled by ~~higher layersthe upper layer~~upper layers. In the transmitter, ~~the~~ a new timer is ~~activated~~ started upon reception of a~~n~~ SDU from ~~upper~~ ~~higher layersthe upper layer~~upper layers. ~~One timer is used for each SDU that is received from upper layer. For~~

In UM/TM~~r~~, if ~~the~~ a timer expires before the corresponding SDU is submitted to ~~a lower layers~~lower layer, "SDU discard without explicit signalling" specified in subclauses 11.2.4.3 and ~~/~~11.1.4.2 shall be ~~started~~initiated. ~~For~~ In AM, if ~~the~~ a timer expires before the corresponding SDU is acknowledged, "SDU discard with explicit signalling" specified in subclause 11.6 shall be ~~started~~initiated.

e) Timer_Poll_Periodic.

This timer ~~is will~~shall only be used when ~~the~~ "timer based polling" is ~~used~~configured by ~~higher layersthe upper layer~~upper layers. The ~~initial value~~value of the timer is signalled by ~~higher layersthe upper layer~~upper layers. The timer ~~is will~~shall be started when the RLC entity is created. ~~Each~~ When~~time~~ the timer expires, the RLC entity ~~will~~shall:

- ~~-~~ restart the timer~~; is restarted and a poll is triggered (either by the transmission of a PDU which was not yet sent, or by a retransmission)~~.
- ~~I~~if ~~there is no~~ PDUs are available for ~~to be~~ transmission~~tted~~ or retransmission (not yet acknowledged):
    - trigger a poll.~~and all PDUs have already been acknowledged, a poll shall not be triggered and the timer shall only be restarted. The value of the timer is signalled by upper layers.~~

f) Timer_Status_Prohibit.

This timer ~~is will~~shall only be used when ~~the STATUS prohibit function is~~so ~~used~~configured by ~~higher layersthe upper layer~~upper layers. It is meant to prohibit~~s~~ the ~~receiving side~~Receiver from sending consecutive acknowledgment status reports. A status report is an acknowledgement status report if it contains~~ing~~ .any of the SUFIs LIST, BITMAP, RLIST or ACK. The ~~initial value~~value of the timer is signalled by ~~higher layersthe upper layer~~upper layers.

In the UE ~~T~~thise timer ~~is will~~shall be started when the successful or unsuccessful transmission of the last STATUS PDU ~~in~~ of an acknowledgment ~~-~~status report is indicated by ~~lower layers~~lower layer~~lower layer~~. ~~(in UE) or~~ In UTRAN it ~~willshall~~should be started when the last STATUS PDU ~~in~~ of an acknowledgment status report is submitted to ~~lower layers~~lower layer~~lower layer (in UTRAN)~~. ~~The prohibit time is calculated f~~

From the time an ~~If another~~ acknowledgment status report ~~containing the above-mentioned SUFIs~~ is triggered until ~~before rom the time the last STATUS PDU of a status report is submitted to lower layer until~~ the Timer_Status_Prohibit timer ~~has~~ expires~~d~~, acknowledgment is prohibited. If another such status report is triggered while acknowledgment is prohibited, its transmission ~~will~~shall be delayed until the timer expires (see subclause 9.7.2). The status report may be updated during this time. ~~and no new status report containing the mentioned SUFIs can be transmitted during the prohibit time.~~ The ~~timer does not prohibit~~ transmission of ~~the~~ SUFIs MRW, MRW_ACK, WINDOW or NO_MORE is not restricted. ~~The value of the timer is signalled by upper layers.~~

When Timer_Status_Prohibit is not configured by ~~the upper layer~~upper layers, acknowledgment is never prohibited.

g) Timer_Status_Periodic.

This timer ~~is will~~shall only be used when timer based status reporting ~~sending~~ is ~~used~~configured by ~~higher layersthe upper layer~~upper layers.

Thise timer ~~is will~~shall be started when the RLC entity is created. ~~Each time~~When the timer expires the transmission of a status report ~~is will~~shall be triggered and the timer ~~is will~~shall be restarted. ~~The value of the timer is signalled by upper layers.~~ This timer can be blocked by ~~upper higher layersthe upper layer~~upper layers. ~~In this case, the timer shall not be active.~~ The timer shall be ~~reset and~~ restarted when ~~it ishigher layersthe upper layer~~upper layers indicates that it is no longer ~~un~~blocked~~ by upper layers~~.

h) Timer_RST.

This timer is ~~used~~ meant to ~~detect~~handle the loss of a RESET PDU by the peer entity, or the loss of a RESET ACK PDU from the peer ~~RLC~~ entity. The ~~initial value~~value of the timer is signalled by ~~higher layersthe upper layer~~upper layers. ~~This~~

In the UE this timer ~~is will~~shall be started when the successful or unsuccessful transmission of a RESET PDU is indicated by ~~lower layers~~lower layer~~lower layer~~. ~~(in UE) or~~ In UTRAN it ~~willshall~~should be started when a RESET PDU is submitted to ~~lower layers~~lower layer ~~(in UTRAN)~~.

Timer_RST ~~It will~~shall only be stopped upon reception of a RESET ACK PDU (with same RSN as RESET PDU), i.e. this timer ~~is will~~shall not be stopped when a~~n~~ RLC reset ~~occurs which was~~ initiated ~~from~~ by the peer RLC entity occurs. If ~~it~~ this timer expires, the RESET PDU ~~will~~shall be retransmitted. ~~The value of the timer is signalled by upper layers.~~

i) Timer_MRW.

~~This timer is used as part of the Move Receiving Window protocol. It~~This timer is used to trigger the retransmission of a status report containing a~~n~~ MRW SUFI field. The ~~initial value~~value of the timer is signalled by ~~higher layersthe upper layer~~upper layers.

In the UE ~~The~~this timer ~~is will~~shall be started when the successful or unsuccessful transmission of a STATUS PDU containing the MRW SUFI is indicated by ~~lower layers~~lower layer. ~~(in UE) or~~ In UTRAN, it ~~willshall~~should be started when a STATUS PDU containing the MRW SUFI is submitted to ~~lower layers~~lower layer ~~(in UTRAN)~~.

Each time the timer expires the MRW SUFI is retransmitted and the timer is restarted ~~(at the time specified above)~~. It ~~shall will~~shall be ~~stopped reset~~stopped when one of the termination criteria for the SDU discard with explicit signalling procedure is fulfilled (see subclause 11.6.4). ~~The value of the timer is signalled by upper layers.~~

9.6 Protocol Parameters

The behavior defined in this subclause is normative. The values of the protocol parameters defined in this subclause are signalled by ~~upper~~ higher ~~layersthe upper layer~~upper layers.

a) MaxDAT.

~~It~~ This protocol parameter indicates~~is~~ the maximum value for the number of retransmissions of a PDU. ~~This parameter is an~~ It represents the upper limit ~~of counter~~for state variable VT(DAT). When ~~the value of~~ VT(DAT) ~~comes to~~~~reaches~~equals the value MaxDAT, either RLC RESET procedure or SDU discard procedure ~~shall will~~shall be initiated according to the configuration by ~~upper~~ higher ~~layersthe upper layer~~upper layers.

b) Poll_PDU.

This protocol parameter indicates how often the transmitter ~~should will~~shall poll the ~~receiver~~Receiver in the case ~~of~~ where "polling every Poll_PDU PDU" is configured by ~~higher layersthe upper layer~~upper layers. ~~This is an~~It represents -the upper limit for ~~the~~ state variable VT(PDU). ~~state variable, w~~When VT(PDU) ~~reaches~~equals the value Poll_PDU a poll ~~is will~~shall be transmitted to the peer entity.

c) Poll_SDU.

This protocol parameter indicates how often the transmitter ~~willshallshould~~ poll the ~~receiver~~Receiver in the case ~~of~~where "polling every Poll_SDU SDU" is configured by ~~higher layersthe upper layer~~upper layers. ~~This is an~~ It represents the upper limit for ~~the~~ state variable VT(SDU). ~~state variable, w~~When VT(SDU) ~~reaches~~equals the value Poll_SDU a poll ~~is will~~shall be transmitted to the peer entity.

d) Poll_Window.

This protocol parameter indicates when the transmitter ~~willshallshould~~ poll the ~~receiver~~Receiver in the case ~~of performing~~where "window-based polling" is configured by ~~higher layersthe upper layer~~upper layers. The range of values of this parameter ~~shall will~~shall be 0 ≤ Poll_Window ≤ 100. A poll is triggered for each PDU when J ≥ Poll_Window, where J is the window transmission percentage defined ~~by~~as:

$$J = \frac{(4096+VT(S) - VT(A))\ \mathbf{mod}\ 4096}{VT(WS)} * 100\ ,$$

where the constant 4096 is the modulus for AM described in ~~S~~subclause 9.4.

e) MaxRST.

~~It~~ This protocol parameter ~~is~~ indicates the maximum ~~value for the~~ number of retransmissions of a RESET PDU. ~~This parameter is an~~ It represents the upper limit ~~of counter~~for state variable VT(RST). When ~~the value of~~ VT(RST) ~~comes reaches~~equals the value~~to~~ MaxRST, unrecoverable error ~~shall will~~shall be indicated to ~~upper higher layersthe upper layer~~upper layers.

f) Configured_Tx_Window_Size.

This protocol parameter indicates ~~T~~both the maximum allowed transmitter window size and the ~~initial value~~value for the state variable VT(WS).

g) Configured_Rx_Window_Size.

This protocol parameter indicates the ~~allowed~~ receiver window size.

h) MaxMRW.

~~It~~ This protocol parameter indicates~~is~~ the maximum value for the number of retransmissions of a MRW SUFI~~command~~. ~~This parameter is an~~ It represents the upper limit ~~of counter~~for state variable VT(MRW). When ~~the value of~~ VT(MRW) ~~comes reaches~~equals the value~~to~~ MaxMRW, the RLC RESET procedure ~~shall will~~shall be initiated.

9.7 Specific functions

The functions defined in this subclause are normative.

9.7.1 Polling function for acknowledged mode

The Polling function is used by the Sender~~transmitter of AMD PDUs may poll~~ to request the ~~receiver~~ peer RLC entity for a status report ~~(consisting of one or several STATUS PDUs)~~. The "Polling bit" in the AMD PDU indicates the poll request. ~~If there is no PDU to be transmitted and all PDUs have already been acknowledged, the receiver shall not be polled.~~ There are several triggers for initiating the Polling function. ~~setting the polling bit. Upper layers control,~~ ~~w~~Which of the triggers ~~should~~ shall be used is configured by upper layers for each RLC entity. ~~If a poll has been triggered while a Timer_Poll is active a new Polling function is initiated after Timer_Poll has expired.~~ The ~~F~~following triggers ~~are possible~~can be configured:

1) Last PDU in buffer.

The ~~s~~Sender triggers ~~a~~ the ~~p~~Polling function when the last PDU available for transmission is submitted to lower layer~~transmitted~~.

2) Last PDU in R~~r~~etransmission buffer.

The ~~s~~Sender triggers ~~a~~ the ~~p~~Polling function when the last PDU to be retransmitted is submitted to lower layer~~transmitted~~.

3) Poll timer.

The timer Timer_Poll is started and stopped according to subclause 9.5 a). ~~when the successful or unsuccessful transmission of a PDU containing a poll is indicated by lower layer (in UE) or a PDU containing a poll is submitted to lower layer (in UTRAN) and if the criterion for stopping the timer has not occurred before~~ When the timer Timer_Poll expires ~~a new~~ the Sender triggers the ~~p~~Polling function ~~is triggered~~.

4) Every Poll_PDU PDU.

The ~~s~~Sender triggers ~~a~~ the ~~p~~Polling function every Poll_PDU PDU. Both retransmitted and new PDUs shall be counted.

5) Every Poll_SDU SDU.

The ~~s~~Sender triggers ~~a~~ the ~~p~~Polling function every Poll_SDU SDU.

6) Window based.

The ~~s~~Sender triggers ~~a~~ the ~~p~~Polling function when the condition described in subclause 9.6 d) ("Poll_Window") is fullfilled.~~it has reached Poll_Window% of the transmission window.~~

7) Timer based.

The ~~s~~Sender triggers ~~a~~ the ~~p~~Polling function periodically.

Either the triggers 1) and 2) or the trigger 7) should be configured for every RLC entity to avoid deadlock situations.~~Either the trigger "Last PDU in buffer" and "Last PDU in retransmission buffer" or "Timer based" can be chosen to avoid deadlock for every RLC entity. The network also controls if the poll prohibit function shall be used. The poll bit shall be set to 0 if the poll prohibit function is used and the timer Timer_Poll_Prohibit is active. If a poll was triggered during the prohibit time defined in subclause 9.5 b) (Timer_Poll_Prohibit), the poll shall be delayed until the timer expires. Only one poll shall be transmitted when the timer expires even if several polls were triggered during the prohibit time. This function has higher priority than any of the above-mentioned triggers.~~

The Poll Prohibit function is used by the Sender to delay the initiation of the Polling function. Usage of the Poll Prohibit function is configured by upper layers. The Poll Prohibit function consists of starting the timer Timer_Poll_Prohibit according to subclause 9.5 b) and delaying the Polling function according to the following rules:

When the Polling function is triggered the Sender shall:

- if ~~the timer Timer_Poll_Prohibit is not active~~polling is not prohibited (see suclause 9.5 b)); and
- if there is one or more AMD PDUs to be transmitted or there are AMD PDUs not acknowledged by the Receiver:
    - initiate the Polling function by setting the polling bit according to subclause 11.3.2.1.1.

Upon expiry of the timer Timer_Poll_Prohibit, the Sender shall:

- if the Polling function was triggered at least once while the timer Timer_Poll_Prohibit was active; and
- if there is one or more AMD PDUs to be transmitted or there are AMD PDUs not acknowledged by the Receiver:
    - initiate the Polling function once by setting the polling bit according to subclause 11.3.2.1.1.

9.7.2 STATUS transmission for acknowledged mode

The ~~r~~Receiver ~~of AMD PDUs~~ transmits status reports to the Sender in order to inform the Sender about which AMD PDUs have been received and not received. ~~(e~~Each status report consists of one or several STATUS PDUs.~~) to the sender in order to inform about which PDUs that have been received and not received. There are several triggers for sending a status report. Upper layers control which triggers should be used for each RLC entity, except for one, which is always present.~~ The ~~r~~Receiver shall always send a status report when receiving a poll request. ~~Except for that trigger~~ Additionally the following triggers for transmission of status reports are configurable by upper layers:

1) Detection of missing PDU(s).

If the ~~r~~Receiver detects one or several missing AMD PDUs it shall trigger the transmission of a status report to the ~~s~~Sender.

2) Timer based ~~STATUS~~ status report transfer.

The ~~r~~Receiver triggers the transmission of a status report ~~periodically~~ to the ~~s~~Sender periodically. The timer Timer_Status_Periodic controls the time period according to subclause 9.5 g). When “Periodical Status blocking” is configured by upper layers, the trigger shall not be active.

3) The EPC mechanism.

The timer Timer_EPC is started according to subclause 9.5 c) and the state variable VR(EP) is set and decreased according to subclause 9.7.4.~~when the successful or unsuccessful transmission of the first STATUS PDU of a~~

~~status report is indicated by lower layer (in UE) or the first STATUS PDU of a status report is submitted to lower layer (in UTRAN).~~ If not all AMD PDUs requested for retransmission have been received before the variable VR(EP) ~~has reached~~equalled zero, a new status report is ~~transmitted to the peer entity~~triggered by the Receiver. A more detailed description of the EPC mechanism is given in subclause 9.7.4.

There are two functions that can prohibit the ~~r~~Receiver from sending a status report containing any of the SUFIs LIST, BITMAP, RLIST or ACK. Status reports containing other SUFIs are not prohibited. Upper layers control which functions should be used for each RLC entity. If any of the following functions is used the ~~sending~~ transmission of the status report shall be delayed, even if any of the triggering conditions above are fulfilled:

1) STATUS prohibit.

   The timer Timer_Status_Prohibit is started accoring to subclause 9.5 f).~~when the successful or unsuccessful transmission of the last STATUS PDU of a status report is indicated by lower layer (in UE) or the last STATUS PDU of a status report is submitted to lower layer (in UTRAN). The prohibit time is calculated from the time the last STATUS PDU of a status report is submitted to lower layer until the timer has expired.~~ The ~~receiving side~~ Receiver is not allowed to transmit a status report ~~during~~ while ~~the prohibit time~~~~timer Timer_Status_Prohibit is active~~acknowledgement is prohibited (see subclause 9.5 f)). If a status report was triggered during thise ~~prohibit~~ time, the status report is transmitted after the timer Timer_Status_Prohibit ~~prohibit time~~ has expired, as described below. ~~The receiver shall only send one status report, even if there are several triggers during the prohibit time. This timer only prohibits the transmission of status reports containing any of the SUFIs LIST, BITMAP, RLIST or ACK. Status reports containing other SUFIs are not prohibited.~~

2) The EPC mechanism.

   If the “EPC mechanism” is active and the ~~sending~~ transmission of a status report is triggered it shall be delayed until the “EPC mechanism” has ended, as described below. ~~The receiver shall only send one status report, even if there are several triggers when the timer is active or the counter is counting down. This mechanism only prohibits the transmission of status reports containing any of the SUFIs LIST, BITMAP, RLIST or ACK. Status reports containing other SUFIs are not prohibited.~~

When a status report is triggered the Receiver shall:

- if transmission of status reports is not prohibited by any of the functions “STATUS prohibit” or “EPC mechanism”:
  - assemble and transmit the status report to the Sender, as specified in subclause 11.5.2.2 and 11.5.2.3~~X~~;
- otherwise (if the status report is prohibited by at least one of the functions "STATUS prohibit" or "EPC mechanism")
  - if MRW, MRW_ACK or WINDOW SUFIs are required in the status report:
    - send a status report immediately excluding ACK, LIST, BITMAP, and RLIST SUFIs;
  - if ACK, LIST, BITMAP, or RLIST SUFIs are required in the status report:
    - delay sending these SUFIs until the prohibit function terminates.

Upon expiry of the timer Timer_Status_Prohibit or termination of the “EPC mechanism”, the Receiver shall:

- if at least one status report was triggered during the time the transmission of a status reports was prohibited that could not be transmitted due to prohibition; and
- if transmission of a status reports is no longer prohibited by any of the functions “STATUS prohibit” or “EPC mechanism”:
  - transmit one status report to the Sender, using the procedure described in subclause 11.5.2.3~~X~~.

9.7.3 SDU discard function for acknowledged, unacknowledged, and transparent mode

The SDU discard function is used by the Sender ~~allows~~ to discharge RLC PDUs from the RLC PDU buffer ~~on the transmitter side~~, when the transmission of the RLC PDUs does not succeed~~s~~s for a period of ~~long~~ time or for a number of retransmissions. The SDU discard function allows to avoid buffer overflow. There ~~will be~~ are several alternative operation modes of the RLC SDU discard function. Upper layers control, which discard function shall be used for each RLC entity.

The following is a list of operation modes for the RLC SDU discard function, which are described in detail in the subsequent subclauses.

Table 9.2: List of criteria that control when to perform SDU discard

Operation mode	Presence
Timer based discard, with explicit signalling	Network controlled
Timer based discard, without explicit signalling	Network controlled
SDU discard after MaxDAT number of retransmissions	Network controlled
No_discard after MaxDAT number of retransmissions	Network controlled

9.7.3.1 Timer based discard, with explicit signalling

This alternative is only applicable to RLC entities operating in acknowledged mode. ~~This alternative~~ It uses a timer based triggering of SDU discard (Timer_Discard). This makes the SDU discard function insensitive to variations in the channel rate and provides means for exact definition of maximum delay. However, the SDU loss rate of the connection is increased as SDUs are discarded.

~~For every SDU received from a upper layer, timer monitoring of the transmission time of the SDU is started. If the transmission time exceeds a predefined value for an SDU in acknowledged mode RLC, this SDU is discarded in the transmitter. Following which, if one or more segments of the SDU have been submitted to a lower layer, a Move Receiving Window (MRW) command is sent to the receiver so that AMD PDUs carrying that SDU are discarded in the receiver and the receiver window is updated accordingly. If Send MRW is configured, an expired SDU whose segments were not submitted to a lower layer is also informed to the receiver by a MRW command.~~
For every SDU received from upper layers, the Sender shall:

- start a timer Timer_Discard~~ monitoring of the transmission time of the SDU~~.

When the ~~transmission time~~timer Timer_Discard of a SDU expires~~exceeds the configured value for a SDU~~, the Sender shall:

- discard the SDU;
- ~~if "Send MRW" is not configured and no segments of the discarded SDU were submitted to the lower layer:~~
  - ~~not utilise explicit signalling;~~
- ~~otherwise (~~if "Send MRW" is configured, or one or more segments of the discarded SDU were submitted to the lower layer~~)~~:
  - utilise ~~explicite~~explicit signalling to inform the Receiver according to subclause 11.6.

~~NOTE: When the concatenation function is active, PDUs carrying segments of other SDUs that have not timed out shall not be discarded.~~

~~The MRW command is defined as a super-field in the RLC STATUS PDU (see subclause 9.2), and piggybacked to status information of transmissions in the opposite direction. If the MRW command has not been acknowledged by receiver, it will be retransmitted. Therefore, SDU discard variants requiring peer-to-peer signalling are only possible for full duplex connections.~~

9.7.3.2 Timer based discard, without explicit signalling

This alternative is only applicable to RLC entities operating in unacknowledged or transparent mode. ~~This alternative~~ It uses the same timer based trigger for SDU discard (Timer_Discard) as the one described in the subclause 9.7.3.1. The difference is that this discard method does not use any peer-to-peer signalling. ~~This function is applied only for unacknowledged and transparent mode RLC and peer-to-peer signalling is never needed. The SDUs are simply discarded in the transmitter, once the transmission time is exceeded. For UM RLC, how to update the sequence number is specified in subclause 11.2.4.3.~~

For every SDU received from upper layers, the Sender shall:

- start timer monitoring of the transmission time of the SDU.

When the transmission time exceeds the configured value for a SDU, the Sender shall:

- discard the SDU without ~~explicite~~explicit signalling (for RLC entities operating in unacknowledged mode apply subclause 11.2.4.3 for updateing the state variables).

9.7.3.3 SDU discard after MaxDAT number of retransmissions

This alternative uses the number of retransmissions as a trigger for SDU discard, and is therefore only applicable for acknowledged mode RLC. This makes the SDU discard function dependent on~~f~~ the channel rate. Also, this variant of the SDU discard function strives to keep the SDU loss rate constant for the connection, on the cost of a variable delay. ~~SDU discard is triggered at the transmitter, and a MRW command is necessary to convey the discard information to the receiver, like in the timer-based discard with explicit signalling.~~

If MaxDAT number of retransmissions is reached for a AMD PDU, the Sender shall:

- discard all SDUs segments of which are contained in the AMD PDU and utilise ~~explicite~~explicit signalling to inform the Receiver according to clause 11.6.

9.7.3.4 No_discard after MaxDAT number of retransmissions

This alternative uses the number of retransmissions, and is therefore only applicable for acknowledged mode RLC. ~~Reset procedure shall be initiated after MaxDAT number of retransmissions of an AMD PDU (see subclause 11.3.4.4).~~

If MaxDAT number of retransmissions is reached for an AMD PDU, the Sender shall:

- initiate the RLC Reset procedure (see subclause 11.3.4.4).

9.7.3.5 SDU discard not configured

If SDU discard has not been configured for an unacknowledged mode RLC entity, SDUs in the transmitter shall not be discarded unless the T~~t~~ransmission buffer is full. ~~If the transmission buffer is full, SDUs may be discarded using SDU discard without explicit signalling. If no segments of the SDU has been transmitted, the SDU may be removed from the buffer without using any of the SDU discard procedures.~~

When the T~~t~~ransmission buffer in an unacknowledged mode RLC entity is full, the Sender may:

- if segments of the SDU to be discarded have been submitted to lower layer:
  - discard the SDU without ~~explicite~~explicit signalling according to subclause 11.2.4.3;
- otherwise, if no segments of the SDU to be discarded have been submitted to lower layer:
  - remove the SDU from the T~~t~~ransmission buffer without utilising any of the discard procedures.

~~If SDU discard has not been configured for a transparent mode RLC entity, SDUs in the transmitter shall be transmitted in the first possible TTI and other not yet transmitted SDUs received from upper layer in previous TTIs shall be discarded upon reception of a new SDU from upper layer.~~
If SDU discard has not been configured for a transparent mode RLC entity, the Sender shall upon reception of new SDUs from upper layer:

- discard all SDUs received from upper layer in previous TTIs that are not yet submitted to lower layer;
- submit the new SDUs in the first possible TTI.

For an acknowledged mode RLC entity, an SDU discard mode is always configured.

9.7.4 The Estimated PDU Counter for acknowledged mode

The Estimated PDU Counter (EPC) is only applicable for RLC entities operating in acknowledged mode. The EPC is a mechanism configured by ~~higher layer~~upper layers used for scheduling the retransmission of status reports in the

~~r~~Receiver~~ side~~. With this mechanism, the ~~r~~Receiver will send a new status report in which it requests for AMD PDUs not yet received. The time between two subsequent status report retransmissions is not fixed, but it is controlled by both the timer Timer_EPC and the state variable VR(EP), which adapt this time to the round trip delay and the current bit rate, indicated in the TFI, in order to minimise the delay of the status report retransmission.

When a ~~STATUS~~ status report is triggered by some mechanisms and it is submitted to lower layer (in UTRAN) or the successful or unsuccessful transmission of it is indicated by lower layer (in UE) to request for retransmitting one or more missing AMD PDUs, the variable VR(EP) is set equal to the number of requested AMD PDUs. At least one requested AMD PDU is needed to activate the EPC mechanism. The variable VR(EP) is a counter, which is decremented every transmission time interval with the estimated number of AMD PDUs that should have been ~~transmitted~~ received during that transmission time interval on the corresponding logical channel.

~~A special~~ The timer~~, called~~ Timer_EPC~~,~~ controls the maximum time that the variable VR(EP) needs to wait before it will start counting down. This timer starts immediately after a transmission of a retransmission request from the ~~r~~Receiver (when the first STATUS PDU of the status report is submitted to lower layer (in UTRAN) or the successful or unsuccessful transmission of it is indicated by lower layer(in UE)). The initial value of the timer Timer_EPC is configured by upper layers. It typically depends on the roundtrip delay, which consists of the propagation delay, processing time in the transmitter and ~~receiver~~Receiver and the frame structure. This timer can also be implemented as a counter, which counts the number of 10 ms radio frames that could be expected to elapse before the first requested AMD PDU is received.

If not all of these requested AMD PDUs have been received correctly when VR(EP) is equal to zero, a new status report will be transmitted and the EPC mechanism will be reset accordingly. The timer Timer_EPC will be started once more when the first STATUS PDU of the status report is submitted to lower layer (in UTRAN) or the successful or unsuccessful transmission of it is indicated by lower layer (in UE). If all of the requested AMD PDUs have been received correctly, the EPC mechanism ends.

9.7.5 Local Suspend function for acknowledged and unacknowledged mode

The upper layers may suspend a~~the~~ RLC entity. ~~The CRLC-SUSPEND-Req indicates this request. The RLC entity shall, when receiving this request, not send RLC PDUs with SN≥VT(S)+N for AM and SN≥VT(US)+N for UM ,where N is given by the CRLC_SUSPEND-Req primitive. The RLC entity shall acknowledge the CRLC-SUSPEND-Req ordering a suspend with a CRLC-SUSPEND-Conf with the current value of VT(S) for AM and VT(US) for UM. When a CRLC-RESUME-Req primitive indicating resume is received, the AM RLC entity enters the acknowledged data transfer ready state if it is in the local suspend state and enters the reset pending state if it is in the reset and suspend state.~~

When ~~the~~ a RLC entity operating in unacknowledged mode is suspended by upper layers with the parameter N, the RLC entity shall:

- acknowledge the suspend request~~order~~ with a confirmation containing the current value of VT(US);
- not send UMD PDUs with sequence number SN≥VT(US)+N.

When a~~the~~ RLC entity operating in acknowledged mode is suspended by upper layers with the parameter N, the RLC entity shall:

- acknowledge the suspend request~~order~~ with a confirmation containing the current value of VT(S);
- not send AMD PDUs with sequence number SN≥VT(S)+N.

When a~~the~~ RLC entity operating in unacknowledged mode is resumed by upper layers, the RLC entity ~~should~~shall:

- resume ~~normal~~ data transfer procedure~~.enter UNACKNOWLEDGED_DATA_TRANSFER_READY state.~~.

When a~~the~~ RLC entity operating in acknowledged mode is resumed by upper layers, the RLC entity ~~should~~shall:

- if the RLC entity is suspended and ~~the~~ a RLC Reset procedure is not ongoing~~activated is in LOCAL_SUSPEND state~~:
  - resume ~~normal~~ data transfer procedure~~enter ACKNOWLEDGED_DATA_TRANSFER_READY state~~.

- otherwise if the RLC entity is suspended and a~~the~~ RLC Reset procedure is ongoing~~activatedin RESET_AND_SUSPEND state~~:
  - remove the suspend constraint;
  - resume the RLC reset procedure according to subclause 11.4~~enter RESET_PENDING state~~.

9.7.6 RLC Stop, RLC Continue function for acknowledged and unacknowledged mode

The upper layer may stop a~~the~~ RLC entity. ~~The stop parameter in the CRLC-CONFIG-Req primitive indicates this request. The RLC entity shall, when receiving this request, not submit any RLC PDUs to lower layer or receive any RLC PDUs. The data transmission and reception is continued when the continue parameter in the CRLC-CONFIG-Req primitive is received. If the continue parameter is received when the RLC entity is not stopped, no action shall be taken.~~

When a~~the~~ RLC entity is stopped, the RLC timers are not affected. ~~Triggered polls and status transmissions are delayed until the RLC entity is continued.~~

When a~~the~~ RLC entity is stopped by upper layers, the RLC entity shall:

- not ~~not~~ submit any RLC PDUs to lower layer or receive any RLC PDUs;
- delay triggered Polling functions or status transmissions until the RLC entity is continued.

When a~~the~~ RLC entity is continued by upper layers, the RLC entity shall:

- if the RLC entity is stopped:
  - continue the data transmission and reception;
  - process the triggered Polling functions and status transmissions.
- otherwise, if the RLC is not stopped:
  - take no action.

9.7.7 RLC re-establishment function for acknowledged and unacknowledged mode

The RLC re-establishment function is applicable for AM and UM and is used when upper layers request a~~the~~ RLC entity to be re-established.

~~When an RLC entity is re-established, the state variables in the RLC entity (see 9.4) shall be reset to their initial value and the configurable parameters shall be set to their configured value. In AM, all RLC PDUs in the RLC receiver and transmitter shall be discarded. In UM, the RLC SDU for which one or more segments have been submitted to a lower layer in the transmitter shall be discarded. The hyper frame number (HFN) in UL and DL shall be set to the value configured by upper layers. After the re-establishment, RLC shall enter the data transfer ready state.~~
When a~~the~~ RLC entity is re-established by upper layers, the RLC entity shall:

- reset the state variables to their initial value;
- set the configurable parameters to their configured value;
- set the hyper frame number (HFN) in UL and DL to the value configured by upper layers;
- if the RLC entity is operating in unacknowledged mode:
  - if it is a receiving UM RLC entity:
    - discard all UMD PDUs;
  - if it is a ~~sending~~transmitting UM RLC entity:

- discard the RLC SDUs for which one or more segments have been submitted to a lower layer~~ in the sending UM RLC entity~~Sender;

~~- discard all UMD PDUs at the receiving UM RLC entity;~~

~~- enter UNACKNOWLEDGED_DATA_TRANSFER_READY state;~~

otherwise if the RLC entity is operating in acknowledged mode:

- discard all AMD PDUs in the Receiver and Sender;

~~- enter ACKNOWLEDGED_DATA_TRANSFER_READY state.~~

9.7.8 Ciphering for acknowledged and unacknowledged mode

The ciphering function is performed in RLC, according to the following rules if a radio bearer is using a non-transparent RLC mode (AM or UM). The data unit that is ciphered, depends on the transmission mode as described below.

- For RLC UM mode, the ciphering unit is the UMD PDU excluding the first octet, i.e. excluding the RLC UMD PDU header. This is shown below in Figure 9.19.

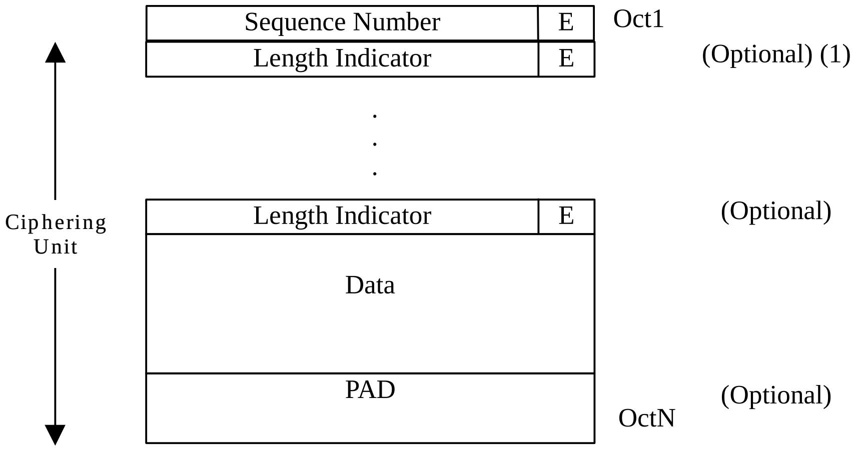


Figure 9.19: Ciphering unit for a UMD PDU

- For RLC AM mode, the ciphering unit is the AMD PDU excluding the first two octets, i.e. excluding the RLC AMD PDU header. This is shown below in Figure 9.20.

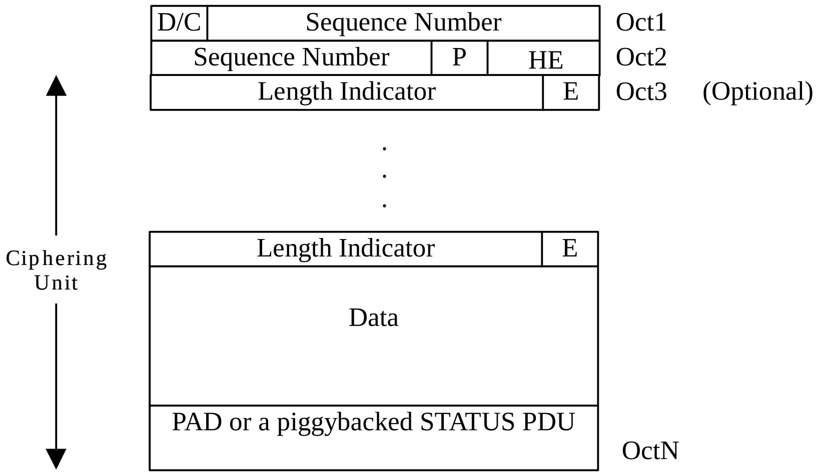


Figure 9.20: Ciphering unit for an AMD PDU

The ciphering algorithm and key to be used are configured by upper layers [8] and the ciphering method shall be applied as specified in [9~~10~~].

The parameters that are required by RLC for ciphering are defined in [9~~10~~] and are input to the ciphering algorithm. The parameters required by RLC which are provided by upper layers [8] are listed below:

- RLC AM HFN (Hyper frame number for radio bearers that are mapped onto RLC AM)
- RLC UM HFN (Hyper frame number for radio bearers that are mapped onto RLC UM)
- BEARER (Radio Bearer ID)
- CK (Ciphering Key)

10 Handling of unknown, unforeseen and erroneous protocol data

Errors and the handling of errors defined in this clause are normative.

~~In case of error situations the following actions are foreseen:~~

~~1) RLC entity shall initiate RESET procedure in case of a protocol error.~~

~~2) RLC entity shall discard invalid PDUs.~~

~~3) RLC entity shall notify upper layer of unrecoverable error occurrence (see subclause 11.4.5.2).~~

~~The list of protocol error cases is reported below:~~

~~Inconsistent state variables;~~

~~If the RLC entity receives a PDU including "erroneous Sequence Number", state variables between peer entities may be inconsistent. Following shows "erroneous Sequence Number" examples:~~

- ~~Each Sequence Number of missing PDU informed by SUFI LIST, BITMAP or RLIST is not within the value between "Acknowledge state variable(VT(A))" and "Send state variable(VT(S)) – 1", and~~
- ~~LSN of SUFI ACK is not within the value between "Acknowledge state variable(VT(A))" and "Send state variable(VT(S))".~~

~~Inconsistent status indication of a PDU;~~

- ~~If a received STATUS PDU indicates different status for the same PDU, then the transmitter shall discard the STATUS PDU.~~

~~Invalid PDU format;~~

- ~~If the RLC PDU format contains reserved or invalid values, the RLC PDU shall be discarded.~~

10.1 Erroneous Sequence Number

A STATUS PDU or Piggybacked STATUS PDU including "erroneous Sequence Number" is a STATUS PDU or Piggybacked STATUS PDU that contains:

- a LIST, BITMAP or RLIST SUFI in which the "Sequence Number" of at least one PDU ~~indicated as missing~~that is negatively acknowledged is outside the interval VT(A)≤"Sequence Number"≤ VT(S)-1, or
- an ACK SUFI in which "LSN" is outside the interval VT(A)≤"LSN"≤ VT(S).

If an AM RLC entity receives a STATUS PDU or a Piggybacked STATUS PDU including "erroneous Sequence Number", it shall:

- discard the STATUS PDU or the Piggybacked STATUS PDU;
- initiate the RLC reset procedure (see subclause 11.4).

10.2 Inconsistent status indication

If an AM RLC entity receives a STATUS PDU or a Piggybacked STATUS PDU that indicates different status for the same AMD PDU, it shall:

- discard the STATUS PDU or the Piggybacked STATUS PDU.

10.3 Invalid PDU format

If an UM or AM RLC entity receives a RLC PDU that contains reserved or invalid values (see subclause 9.2), it shall:

- discard the RLC PDU.

11 Elementary procedures

Procedures defined in this clause are normative.

~~In each procedure, RLC entities are functionally identified as Sender or Receiver. This functional distinction is summarised in the message sequence diagram at the beginning of each procedure. Sender and Receiver RLC entities may be either in the UE or the UTRAN. This convention will be used across the procedures.~~

~~The Sender is the RLC entity that transmits data PDUswhich initiates the procedure. The Receiver is the peer RLC entity of the Sender.~~ This description assumes elementary procedures. Interactions between procedures are not described.

11.1 Transparent mode data transfer procedure

11.1.1 ~~Purpose~~General

The transparent mode data transfer procedure is used for transferring ~~of~~ data between two RLC peer entities, which are operating in transparent mode. Data is transferred from Sender to Receiver. This procedure ~~shall~~should only apply to entities in DATA_TRANSFER_READY state. Figure 11.1 below illustrates the elementary procedure for transparent mode data transfer. ~~The sender can be either the UE or the network and the receiver is either the network or the UE.~~

Channels that can be used are DTCH, CCCH (uplink only), SHCCH (uplink only), BCCH and PCCH. The type of logical channel depends on if the RLC entity is located in the user plane (DTCH) or in the control plane (CCCH/BCCH/SHCCH/PCCH).

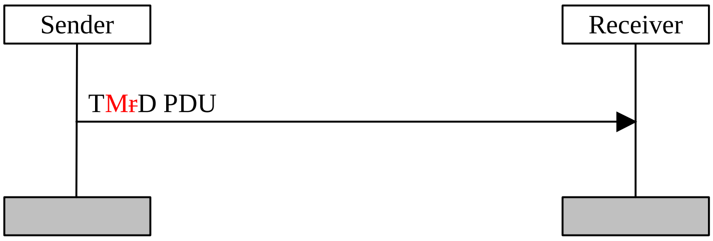


Figure 11.1: Transparent mode data transfer procedure

11.1.2 ~~Initiation~~Transmission of ~~Tr~~TMD PDU

~~The sender initiates this procedure u~~Upon a request of transparent mode data transfer from upper layer, the Sender shall:

- i~~I~~f no SDU discard configuration ~~is configured~~has been made by upper layers:
  - discard ~~all buffered~~ SDUs received in previous TTIs upon reception of new SDUs from ~~the upper layer~~upper layers (see subclause 9.7.3.5);~~.~~

- o~~O~~therwise~~,~~ (if "Timer Based SDU Discard without explicit signalling" is configured):
  - start a timer Timer_Discard for each SDU received from ~~the upper layer~~upper layers (see subclause 9.7.3);
- schedule the RLC SDUs that have been received from upper layer for transmission;
- if one or more RLC SDUs have been scheduled for transmission, the Sender shall;
  - notify the lower layer of reception of data from ~~the upper layer~~upper layers;~~.~~
  - perform the actions specified in subclause 11.1.2.2.

~~. When the sender is in data transfer ready state it shall put the data received from the upper layer into TrD PDUs. If required RLC shall perform segmentation.~~

~~Channels that can be used are DTCH, CCCH (uplink only), SHCCH (uplink only), BCCH and PCCH. The type of logical channel depends on if the RLC entity is located in the user plane (DTCH) or in the control plane (CCCH/BCCH/SHCCH/PCCH). One or several PDUs may be transmitted in each transmission time interval (TTI). For each TTI, MAC decides which PDU size shall be used (applicable when segmentation is used) and how many PDUs shall be transmitted. The SDUs that cannot be transmitted in a TTI shall be buffered according to the discard configuration set by upper layers.~~

~~If timer based SDU discard is used, the timer Timer_Discard shall be started when the RLC entity receives an SDU from upper layer. One timer is used for each SDU that is received from upper layer.~~

11.1.2.1 ~~Tr~~TMD PDU contents to set

The Sender shall set the data field of the ~~Tr~~TMD PDU to all or a subset of the data contained in the SDU as described in subclause 11.1.2.2.~~The TrD PDU includes a complete SDU or a segment of an SDU. How to perform the segmentation is decided upon when the service is established. No overhead or header is added, instead segmentation is done based on which of the transport formats of the transport channel that will be used. A particular transport format informs the receiver how the segmentation was performed.~~

11.1.2.~~X~~2 Submission of TMD PDUs to the lower layer

If one or more RLC SDUs have been scheduled for transmission, according to Subclause 11.1.2, the Sender shall:

- if it is configured for segmented operation:
  - inform the lower layer of the size of the next SDU to be sent;
  - segment the SDU according to the PDU size indicated by the lower layer;
- otherwise~~,~~ (the Sender is configured for non-segmented operation):
  - inform the lower layer of the number and size of SDUs available for transmission;
- submit to the lower layer, the requested number of TMD PDUs;
- buffer the SDUs that are not submitted to the lower layer according to the discard configuration (see subclause 9.7.3).

11.1.3 Reception of ~~Tr~~TMD PDU

Upon ~~reception~~ delivery by the lower layer of a set of ~~Tr~~TMD PDUs (received within one TTI), the Receiver shall:

- if it is configured for segmented operation:
  - ~~receiving entity~~ reassembles ~~(if segmentation was performed)~~ the TMD PDUs received in one TTI into one RLC SDU~~s~~;~~.~~
- otherwise (it is configured for non-segmented operation):
  - treat each received TMD PDU as a SDU;

- if "Delivery of Erroneous SDUs" is configured as "no":
    - ~~RLC delivers~~submit only ~~all~~the RLC SDUs received without error to ~~the upper layer~~upper layers through the ~~Tr~~TM-SAP;~~.~~
- else if "Delivery of Erroneous SDUs" is configured as "yes":
    - submit all RLC SDUs to ~~the upper layer~~upper layers through the TM~~R~~-SAP;~~.~~
    - provide an error indication for each SDU received in error;
- otherwise if "Delivery of Erroneous SDUs" is configured as "No detect":
    - submit all RLC SDUs to ~~the upper layer~~upper layers through the TM-SAP.

~~If delivery of erroneous SDUs is configured as ’yes’ by an upper layer, the receiver shall deliver an erroneous SDU to upper layer with an error indication. If delivery of erroneous SDUs is configured as ’no’ by an upper layer the receiver shall discard the erroneous SDU. If delivery of erroneous SDUs is configured as ’No detect’ by an upper layer, all SDUs shall be delivered to upper layer without error indication.~~

If segmentation is performed in transparent mode RLC, an SDU is erroneous if one or more of the ~~Tr~~TMD PDUs received in a TTI contains an error. If segmentation is not performed, an SDU is erroneous if the corresponding ~~Tr~~TMD PDU is erroneous.

11.1.4 Abnormal cases

11.1.4.1 Void

11.1.4.2 SDU discard without explicit signalling

Upon expiry of the timer Timer_Discard in~~on~~ the S~~s~~ender~~ side~~, the S~~s~~ender shall:

- ~~-~~discard the associated SDU.~~;~~

~~.~~In the case where the TFC selection exchange has been initiated by sending the RLC Entity Info parameter to MAC, the UE may wait until after it provides MAC with the requested set of PDUs before discarding the afore-mentioned SDU.

11.2 Unacknowledged mode data transfer procedure

11.2.1 ~~Purpose~~General

The unacknowledged mode data transfer procedure is used for transferring data between two RLC peer entities, which are operating in unacknowledged mode. Data is transferred from Sender to Receiver. This procedure ~~shall~~should only apply to RLC entities in DATA_TRANSFER_READY state or LOCAL_SUSPEND state. Figure 11.2 below illustrates the elementary procedure for unacknowledged mode data transfer~~. The sender can be either the UE or the network and the receiver is either the network or the UE~~.

Channels that can be used are DTCH, DCCH, CCCH (downlink only), CTCH, SHCCH (downlink only). The type of logical channel depends on if the RLC entity is located in the user plane (DTCH, CTCH) or in the control plane (DCCH/CCCH(downlink only)/SHCCH(downlink only)). One or several PDUs may be transmitted in each transmission time interval (TTI). For each TTI, MAC decides which PDU size shall be used and how many PDUs shall be transmitted.

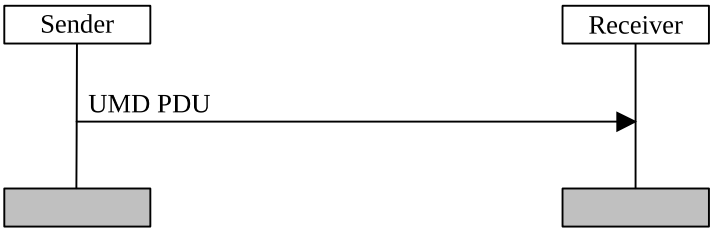


Figure 11.2: Unacknowledged mode data transfer procedure

11.2.2 ~~Initiation~~Transmission of UMD PDU

~~The sender initiates this procedure u~~Upon a request of unacknowledged mode data transfer from upper layer, the Sender shall:~~.~~

- if no SDU discard configuration has been made by upper layers~~configuration is configured~~:
    - ~~not~~only discard ~~any~~ SDUs when ~~in the transmitter unless~~ the ~~transmitter~~Transmission buffer is full (see subclause 9.7.3)
- if "Timer based SDU Discard without explicit signalling" is configured:
    - start a timer Timer_Discard for each SDU received from upper layer (see subclause 9.7.3);
- schedule the RLC SDUs received from upper layer for transmission;
- if one or more RLC SDUs have been scheduled for transmission~~, the Sender shall~~;
    - notify the lower layer of reception of data from ~~the upper layer~~upper layers.
    - perform the actions specified in subclause 11.2~~1~~.2.2.

~~When the sender is in data transfer ready state it shall segment and, if possible, concatenate the data received from the upper layer into PDUs.~~

~~Channels that can be used are DTCH, DCCH, CCCH (downlink only), CTCH, SHCCH (downlink only). The type of logical channel depends on if the RLC entity is located in the user plane (DTCH, CTCH) or in the control plane (DCCH/CCCH(downlink only)/SHCCH(downlink only)). One or several PDUs may be transmitted in each transmission time interval (TTI). For each TTI, MAC decides which PDU size shall be used and how many PDUs shall be transmitted.~~

~~The VT(US) state variable shall be updated for each UMD PDU that is transmitted.~~

~~If timer based SDU discard is used, the timer Timer_Discard shall be started when the RLC entity receives an SDU from upper layers. One timer is used for each SDU that is received from upper layers.~~

A UMD PDU ~~will~~ shall be considered to be a padding PDU if it consists only of an RLC Header with one length indicator (indicating that the rest of the PDU is padding) and padding.

11.2.2.1 UMD PDU contents to set

The Sender shall:

- set the field "Sequence Number" equal to VT(US);
- set a "Length Indicator" field for each SDU that ends in the UMD PDU according to subclause 9.2.2.8;

For each "Extension bit" field in the RLC header, the ~~sender~~Sender shall:

- if the next field in the UMD PDU is a "Length Indicator";
    - set the "Extension bit" to "1";
- otherwise if the next field in the UMD PDU is data:

- set the "Extension bit" to "0";

~~The Sequence Number field shall be set equal to VT(US).~~

~~The Extension bit shall be set to 1 if the next field is a length indicator field, otherwise it shall be set to zero.~~

~~One length indicator field shall be included for each end of an SDU that the PDU includes. The LI fields shall be set as specified in subclause 9.2.2.8.~~

11.2.2.~~X~~2 Submission of UMD PDUs to the lower layer

If one or more SDUs have been scheduled for transmission according to subclause 11.2.2, the Sender shall:

- inform the lower layer of the number and size of SDUs scheduled for transmission;
- segment, and if possible concatenate the SDUs according to the PDU sizes indicated by the lower layer;
- submit to the lower layer, the requested number of UMD PDUs;
- update VT(US) for each UMD PDU submitted to the lower layer (see subclause 9.4);
- buffer the SDUs that are not submitted to the lower layer according to the discard configuration (see subclause 9.7.3).

11.2.3 Reception of UMD PDU

Upon ~~reception~~delivery of a set of UMD PDUs from the lower layer, the ~~receiver~~Receiver shall:

- ~~-~~update VR(US)~~ state variable~~ according to ~~the~~ each received UMD PDU~~(s)~~ (see subclause 9.4);
- if the updating step of VR(US) is not equal to one (i.e. one or more UMD PDUs are missing);
    - discard the SDUs that have segments -in the missing UMD PDUs;
- reassemble the received UMD PDUs into RLC SDUs~~.~~;
- submit the RLC SDUs to ~~the upper layer~~upper layers through the UM-SAP.

~~The PDUs are reassembled into RLC SDUs. If the updating step of the variable VR(US) is greater than one, one or more PDUs are missing. The SDUs that have segments in these missing PDUs shall be discarded. RLC delivers the RLC SDUs to the upper layers through the UM-SAP.~~

11.2.4 Abnormal cases

11.2.4.1 Length Indicator value reserved for UMD PDU

Upon ~~reception~~ delivery by the lower layer of an UMD PDU that contains a "Length Indicator" value specified to be reserved for UMD PDUs in this version of the protocol, the ~~receiver~~Receiver shall:

- ~~-~~discard that UMD PDU and treat the UMD PDU as missing~~.~~

11.2.4.2 Invalid length indicator value

If the "L~~l~~ength I~~i~~ndicator" of an UMD PDU has a value that is larger than the PDU size – RLC header size and is not one of the predefined values listed in the table of subclause 9.2.2.8, the ~~receiver~~Receiver shall:

- discard the UMD PDU and treat the UMD PDU as missing~~PDU shall be discarded and treated as a missing PDU.~~

11.2.4.3 SDU discard without explicit signalling

Upon expiry of the timer Timer_Discard ~~on the~~ in the S~~s~~ender~~-side~~ the S~~s~~ender shall:

- ~~-~~discard the associated SDU;
- for the first UMD PDU to be transmitted after the discard operation the Sender shall:
  - increment VT(US) so that the "Sequence Number" field in this UMD PDU is incremented with two compared with the previous UMD PDU;
  - fill the first data octet in this UMD PDU with the first octet of an RLC SDU;
  - set the first "Length Indicator" in this UMD PDU to indicate that the previous RLC PDU was exactly filled with the last segment of an RLC SDU (to avoid that the ~~receiver~~Receiver unnecessarily discards an extra SDU).

~~-~~In the case where the TFC selection exchange has been initiated by sending the RLC Entity Info parameter to MAC, the UE may wait until after it provides MAC with the requested set of PDUs before discarding the afore-mentioned SDU~~. For UM RLC, SN of the UMD PDUs shall be incremented by a step of 2 for the first PDU transmitted after a Discard Operation to indicate that there were some RLC SDUs discarded before this RLC PDU. The first data octet in this RLC PDU shall be the first octet of an RLC SDU.~~. ~~To prevent the receiver from discarding one extra SDU, an LI field shall be added in the first PDU transmitted after a Discard Operation. The value of the LI field shall be the value indicating that the previous SDU filled exactly the previous RLC PDU.~~

11.3 Acknowledged mode data transfer procedure

11.3.1 General~~Purpose~~

The acknowledged mode data transfer procedure is used for transferring~~ of~~ data between two RLC peer entities, which are operating in acknowledged mode. Data is transferred from Sender to Receiver. This procedure ~~shall~~should only apply to RLC entities in DATA_TRANSFER_READY state or LOCAL_SUSPEND state. Figure 11.3 below illustrates the elementary procedure for acknowledged mode data transfer~~. The sender can be either the UE or the network and the receiver is either the network or the UE~~.

The AMD PDUs shall be transmitted on the DCCH logical channel if the Sender is located in the control plane and on the DTCH if it is located in the user plane. One or several PDUs may be transmitted in each transmission time interval (TTI) and MAC decides how many PDUs shall be transmitted in each TTI.

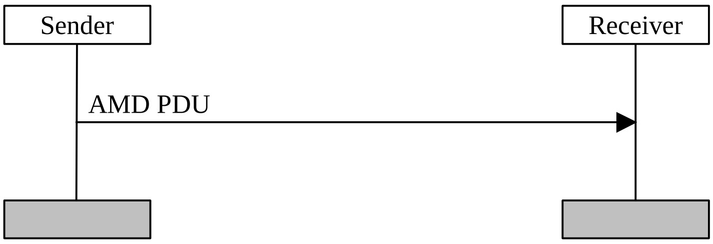


Figure 11.3: Acknowledged mode data transfer procedure

11.3.2 ~~Initiation~~Transmission of AMD PDU

~~The sender initiates this procedure u~~Upon a request of acknowledged mode data transfer from upper layers or upon retransmission of AMD PDUs, the Sender shall:

- when RLC SDUs are received from ~~the upper layer~~upper layers:~~;~~
  - segment the RLC SDUs into AMD PDUs where the fixed PDU size is configured by ~~higher~~upper layer;
  - set a "Length Indicator" field for each SDU that ends in the AMD PDU according to subclause 9.2.2.8;
  - if "Timer based SDU Discard with explicit signalling" is configured:
    - start a timer Timer_Discard for each SDU received from upper layer (see subclause 9.7.3);
  - s~~S~~chedule the AMD PDUs for transmission;

- if one or several AMD PDUs have~~s~~ been negatively acknowledged (see subclause 11.5.3):~~;~~
    - schedule the AMD PDUs ~~indicated as missing~~that were negatively acknowledged for retransmission;
- if a poll has been triggered by either ~~any of~~ the poll triggers "Poll timer" or "Timer based"~~,~~ (see subclause 9.7.1);and
- if no AMD PDU is scheduled for transmission or retransmission:~~;~~
    - if the value of "Configured_Tx_Window_Size" is larger than or equal to "2048":~~;~~
        - select the AMD PDU with "Sequence Number" equal to VT(S)-1;
    - otherwise if the "Configured_Tx_Window_Size" is less than "2048";~~;~~
        - select the AMD PDU with "Sequence Number" equal to VT(S)-1; or
        - select an AMD PDU that has not yet been acknowledged by the peer entity;
    - schedule the selected AMD PDU for retransmission (in order to transmit a poll).

The Sender may also schedule an AMD PDU for retransmission even if none of the criteria above is fulfilled. In this case, the Sender may:

- if the value of "Configured_Tx_Window_Size" is larger than or equal to "2048":~~;~~
    - select the AMD PDU with "Sequence Number" equal to VT(S)-1;
- otherwise if the "Configured_Tx_Window_Size" is less than "2048":~~;~~
    - select the AMD PDU with "Sequence Number" equal to VT(S)-1; or
    - select an AMD PDU that has not yet been acknowledged by the peer entity;
- schedule the selected AMD PDU for retransmission

Each time an AMD PDU is scheduled for transmission or retransmission ~~by one of the criteria listed in this subclause~~, the Sender shall:

- notify the lower layer that data is available for transmission;
- perform the actions specified in subclause 11.3.2.2.~~. Retransmitted PDUs have higher priority than PDUs transmitted for the first time.~~

~~The sender is only allowed to retransmit PDUs that have been indicated missing by the receiver. An exception is the PDU with SN VT(S)-1, which can be retransmitted. In addition, a PDU that has not yet been acknowledged, may be retransmitted if Configured_Tx_Window_Size is less than 2048.~~

~~RLC shall segment the data received from the upper layers into AMD PDUs. The~~ AMD ~~PDUs shall be transmitted on the DCCH logical channel if the sender~~Sender ~~is located in the control plane and on the DTCH if it is located in the user plane. One or several PDUs may be transmitted in each transmission time interval (TTI) and MAC decides how many PDUs shall be transmitted in each TTI. In the UE, the PDUs that cannot be transmitted in a TTI (i.e. MAC has indicated that some of the available PDUs can not be transmitted) shall be buffered according to the discard configuration set by upper layers.~~

~~The VT(DAT) state variables shall be updated for each AMD PDU that is transmitted. The PDU shall not have a Sequence Number ≥ VT(MS), except for the sequence number VT(S)-1; a PDU with this sequence number may be sent also when VT(S) ≥ VT(MS).~~

~~If the poll bit is set in any of the AMD PDUs and the timer Timer_Poll shall be used, the sender shall start the timer Timer_Poll when the successful or unsuccessful transmission of a PDU with the set poll bit is indicated by lower layer (in UE) or submitted to lower layer (in UTRAN).~~

~~If timer based SDU discard is used, the timer Timer_Discard shall be started when the RLC entity receives an SDU from upper layers. One timer is used for each SDU that is received from upper layers.~~

~~If the trigger for polling, "Every Poll_PDU PDU", is used, the VT(PDU) shall be increased by 1 for each PDU that is transmitted.~~

~~If the trigger for polling, "Every Poll_SDU SDU", is used, the VT(SDU) shall be increased by 1 for each SDU that is transmitted.~~

In AM, a PDU ~~will~~ shall be considered to be a padding PDU if it is:

- a~~A~~n AMD PDU consisting only of an RLC Header with one "L~~l~~ength I~~i~~ndicator" (indicating that the rest of the PDU is padding) and padding; or~~.~~
- a~~A~~ Status PDU consisting only of a NO_MORE SUFI.

11.3.2.1 AMD PDU contents to set

- i~~I~~f the AMD PDU is transmitted for the first time, the Sender shall:
    - set the "Sequence Number" field~~ shall be set~~ equal to VT(S)~~; and VT(S) shall be updated.~~
    - set a "Length Indicator" field for each SDU that ends in the AMD PDU according to clause 9.2.2.8;
- otherwise if the AMD PDU is retransmitted~~, the Sender shall~~:
    - use the same value of the "Sequence Number" field as in the original transmission of the AMD PDU
    - if the "Length Indicator" fields needed in the AMD PDU according to subclause 9.2.2.8 has changed due to that a piggybacked STATUS PDU is included in the AMD PDU or a piggybacked STATUS PDU was included in the previous transmission of the AMD PDU;
        - update the "Length Indicator" fields according to 9.2.2.8;
- set ~~The setting of~~ the "Polling bit" to the value~~bit is~~ specified in ~~sub~~clause 11.3.2.1.1.

~~One length indicator field shall be included for each end of an SDU that the PDU includes. The LI fields shall be set as specified in subclause 9.2.2.8.~~

~~How to perform the segmentation and concatenation of SDUs is specified in subclause 11.3.2.1.2.~~

11.3.2.1.1 Setting of the Polling bit

The Sender shall:

- ~~if the "poll prohibit" function is configured; and~~
- ~~if the timer Timer_Poll_Prohibit is active~~
    - ~~consider polling to be prohibited;~~
- if a poll has been triggered by one or several poll triggers (see clause 9.7.1); and
    - if polling is not prohibited, see subclause 9.5;
        - set the "Polling bit" in the AMD PDU header to "1";
- otherwise:
    - set the "Polling bit" in the AMD PDU header to "0";~~The Polling bit shall be set to 1 if any of following conditions are fulfilled except when the poll prohibit function is used and the timer Timer_Poll_Prohibit is active (the different triggers are described in 9.7.1):~~

~~1) Last PDU in buffer is used and the last PDU available for transmission is transmitted.~~

~~2) Last PDU in retransmission buffer is used and the last PDU to be retransmitted is transmitted.~~

~~3) Poll timer is used and timer Timer_Poll has expired.~~

~~4) Every Poll_PDU PDU is used and when VT(PDU)=Poll_PDU.~~

~~5) Every Poll_SDU is used and VT(SDU)=Poll_SDU and the PDU contains the last segment of that SDU.~~

~~6) Window based polling is used, and J≥Poll_Window, where J is defined in subclause 9.6.~~

~~7) Timer based polling is used and Timer_Poll_Periodic has expired.~~

~~8) Poll prohibit shall be used, the timer Timer_Poll_Prohibit has expired and one or several polls were prohibited during the time Timer_Poll_Prohibit was active.~~

11.3.2.1.2 ~~Segmentation and concatenation of SDUs~~Void

~~Upon reception of an SDU, RLC shall segment the SDU to fit into the fixed size of a PDU. The segments are inserted in the data field of a PDU. A length indicator shall be added to each PDU that includes a border of an SDU, i.e. if a PDU does not contain an LI, the SDU continues in the next PDU. The length indicator indicates where the border occurs in the PDU. The data after the indicated border can be either a new SDU, padding or piggybacked information. If padding or piggybacking is added another LI shall be added unless the padding size is one octet for PDUs with 15-bit LIs, see subclauses 9.2.2.8 and 9.2.2.9.~~

11.3.2.~~X~~2 Submission of AMD PDUs to lower layer

If one or more AMD PDUs have been scheduled for transmission or retransmission according to Subclause 11.3.2 the Sender shall:

- not submit any AMD PDUs to lower layer that is not allowed to transmit. AMD PDUs are only allowed to transmit if: ~~according to the following definition. An AMD PDUs is allowed to transmit if:~~
    - The AMD PDU has a "Sequence Number" < VT(MS); or
- ~~-~~The AMD PDU has a "Sequence Number" equal to VT(S)-1
- The AMD PDU is not restricted to be transmitted by the local suspend function, see subclause 9.7.5
- inform the lower layer of the number of AMD PDUs scheduled for transmission or retransmission;
- submit to the lower layer, the requested number of AMD PDUs;
- set the AMD PDU contents according to clause 11.3.2.1
- treat retransmissions with higher priority than AMD PDUs transmitted for the first time;
- update the state variables in clause 9.4 for each AMD PDU submitted to lower layer;
- if the "Polling bit" is set to "1" in any of the AMD PDUs; and
- if the timer Timer_Poll is configured;
    - start the timer Timer_Poll according to subclause 9.5;
- buffer the AMD PDUs that are not submitted to the lower layer according to the discard configuration (see subclause 9.7.3)

11.3.3 Reception of AMD PDU by the ~~receiver~~Receiver

Upon reception of an AMD PDU, the R~~r~~eceiver shall:

- -update VR(R), VR(H) and VR(MR) state variables ~~according to the~~for each received AMD PDU (see clause 9.4);~~.~~

-

i~~I~~f a received AMD PDU includes a "Polling bit" set to "1",~~;~~ or "Missing PDU Indicator" is configured and the Receiver detects that a PDU is missing;

- ~~, if the "detection of missing PDU(s)" is configured and the Receiver detects that a PDU is missing;~~

    - initiate the STATUS PDU transfer procedure~~ shall be initiated.~~;

- reassemble the received AMD PDUs into RLC SDUs;
- if "In-Sequence Delivery" is configured;
    - submit the RLC SDUs in-sequence (i.e. in the same order as the RLC SDUs where originally transmitted by the peer entity) to ~~the upper layer~~upper layers through the AM-SAP
- otherwise if "In-Sequence Delivery" is not configured;
    - submit the RLC SDUs in arbitrary order to ~~the upper layer~~upper layers through the AM-SAP.

~~If the detection of missing PDU(s) shall be used and the receiver detects that a PDU is missing, the receiver shall initiate the STATUS PDU transfer procedure.~~

11.3.4 Abnormal cases

11.3.4.1 Timer_Poll timeout

Upon expiry of the timer Timer_Poll, the ~~sender~~Sender shall:

- if an AMD PDU ~~not previously transmitted has been received from the upper layer~~is available for transmission;
    - transmit an AMD PDU with the "Polling bit" set to "1";
- otherwise if no AMD PDU is available for transmission;
    - retransmit an AMD PDU even if that AMD PDU is not ~~indicated as missing~~negatively acknowledged, with the "Polling bit" set to "1";
- Initiate the acknowledged mode data transfer procedure for the AMD PDU to be transmitted or retransmitted. ~~retransmit the poll. The poll can be retransmitted in either a new PDU or a retransmitted PDU.~~

11.3.4.2 Receiving an AMD PDU outside the receiving window

Upon reception of an AMD PDU with sequence number outside the interval VR(R)≤SN<VR(MR), the ~~receiver~~Receiver shall:

- ~~-~~discard the AMD PDU;

~~. The poll bit shall be~~ - if the "polling bit" in the discarded AMD PDU is set to "1";

    - initiate the STATUS PDU transfer procedure. ~~considered even if a complete PDU is discarded.~~

11.3.4.3 Timer_Discard timeout

11.3.4.3.1 SDU discard with explicit signalling

Upon expiry of the timer Timer_Discard, the ~~sender~~Sender shall:

- ~~-~~initiate the SDU discard with explicit signalling procedure, see subclause 11.6.2. ~~to discard all SDUs up to and including the SDU for which the timer expired.~~

~~.~~In the case where the TFC selection exchange has been initiated by sending the RLC Entity Info parameter to MAC, the UE may wait until after it provides MAC with the requested set of PDUs before discarding the afore-mentioned SDUs.

11.3.4.4 VT(DAT) ≥ MaxDAT

The Sender shall:

- if VT(DAT) ≥ MaxDAT for any AMD PDU;

- if "No_discard after MaxDAT number of retransmissions" is configured:
    - initiate the RLC reset procedure, see subclause 11.4;
- i~~I~~f "SDU discard after MaxDAT number of retransmissions" is configured:~~;used and~~ ~~VT(DAT) ≥ MaxDAT for any PDU~~
    - ~~, the sender shall~~ initiate the SDU discard with explicit signalling procedure, see subclause 11.6.~~. All the SDUs that~~

~~when the SDU discard procedure is initiated, all SDUs that have segments in AMD PDUs with "Ssequence Nnumber"s inside the interval VT(A) ≤ "Sequence Nnumber" SN ≤ X, where X is the value of the "Sequence Nnumber"SN of the AMD PDU with VT(DAT) ≥ MaxDAT shall be discarded.~~

~~If No_discard after MaxDAT number of retransmissions is used, the sender shall initiate the RLC reset procedure when VT(DAT) ≥ MaxDAT.~~

11.3.4.5 Invalid length indicator value

If the "L~~l~~ength I~~i~~ndicator" of an AMD PDU has a value that is larger than the PDU size – RLC header size and is not one of the predefined values listed in the table of subclause 9.2.2.8, the Sender shall:

- discard that AMD PDU;
- ~~shall be discarded and~~ treated the discarded AMD PDU as ~~a~~ missing.~~ PDU.~~

11.3.4.6 Length Indicator value reserved for AMD PDU

Upon ~~reception~~ delivery by the lower layer of an AMD PDU that contains a "Length Indicator" value specified to be reserved for AMD PDUs in this version of the protocol, the ~~receiver~~Receiver shall:

- discard that AMD PDU;
- treat the discarded AMD PDU as missing.~~.~~

11.4 RLC reset procedure

11.4.1 General~~Purpose~~

The RLC reset procedure is used to reset two RLC peer entities, which are operating in acknowledged mode. Figure 11.4 below illustrates the elementary procedure for an RLC reset. ~~The sender can be either the UE or the network and the receiver is either the network or the UE.~~ During the reset procedure the hyper frame numbers (HFN) in UTRAN and UE are synchronised. Two HFNs used for ciphering needs to be synchronised, DL HFN in downlink and UL HFN in uplink. In the reset procedure, the highest UL HFN and DL HFN used by the RLC entity in the transmitting sides, i.e. the HFNs associated with PDUs of SN=VT(S)-1 if at least one data PDU had been transmitted or of SN=0 if no data PDU had been transmitted, are exchanged between UE and UTRAN~~. After the reset procedure is terminated, the UL HFN and DL HFN shall be increased with one in both UE and UTRAN, and the updated HFN values shall be used for the first transmitted and received PDUs after the reset procedure~~.

The RESET PDUs and the RESET ACK PDUs have higher priority than AMD PDUs.

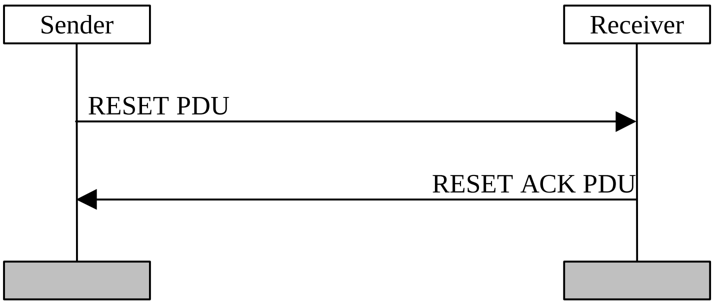


Figure 11.4: RLC reset procedure

11.4.2 Initiation

~~The procedure shall be initiated when a protocol error occurs.~~ The Sender shall:

- if one of the following triggers is detected:
    1) "No_Discard after MaxDAT number of retransmissions" is configured and ~~the number of retransmissions of an AMD PDU~~VT(DAT) ~~comes to~~equals the value MaxDAT (see subclause 9.7.3.4);
    2) ~~The value of~~ VT(MRW) ~~comes to~~equals the value MaxMRW;
    3) A STATUS PDU including "erroneous Sequence Number" is received (see clause 10);
    - ~~The sender sends~~ submit a~~the~~ RESET PDU to the lower layer;~~,~~
    - ~~enters reset pending state when it was in data transfer ready state, and enters reset and suspend state when it was in local suspend state~~
    - ~~. The sender shall~~ start the timer Timer_RST and increase VT(RST) with 1. ~~The RESET PDU shall be transmitted on the DCCH logical channel if the sender is located in the control plane and on the DTCH if it is located in the user plane.~~

~~The RESET PDU has higher priority than data PDUs.~~

When a reset procedure has been initiated it can only be ended upon reception of a RESET ACK PDU with the same RSN value as in the corresponding RESET PDU, ~~i.e.~~or upon request of re-establishment or release from upper layer, a reset procedure is not interrupted by the reception of a RESET PDU from the peer entity.

11.4.2.1 RESET PDU contents to set

The Sender shall:

- set the HFNI field to the currently highest used HFN (DL HFN when the RESET PDU is sent by UTRAN or UL HFN when the RESET PDU is sent by the UE);
- set the RSN field to the sequence number of the RESET PDU. This sequence number is incremented every time a new RESET PDU is transmitted, but not when a RESET PDU is retransmitted.

~~The size of the RESET PDU shall be equal to one of the allowed PDU sizes. The hyper frame number indicator field (HFNI) shall be set equal to the currently highest used HFN (DL HFN when the RESET is sent by UTRAN or UL HFN when the RESET is sent by the UE). The RSN field shall indicate the sequence number of the RESET PDU. This sequence number is incremented every time a new RESET PDU is transmitted, but not when a RESET PDU is retransmitted.~~

11.4.3 Reception of the RESET PDU by the R~~r~~eceiver

Upon reception of a RESET PDU the R~~r~~eceiver shall:

- ~~respond with a RESET ACK PDU~~ submit a RESET ACK PDU to the lower layer;
- ~~. The receiver~~ resets the state variables ~~in~~ described in subclause 9.4 except VT(RST) to their initial values~~,~~;

- ~~-~~stops all the timers ~~in~~described in subclause 9.5 except Timer_RST~~, and~~ ;
- resets configurable parameters to their configured values~~. Both the transmitting and receiving sides of the AM RLC entity are reset.~~ ;
- discard ~~A~~all RLC PDUs in the receiving side of the AM RLC entity~~ receiver shall be discarded.~~;
- discard all~~The~~ RLC SDUs ~~in the AM RLC transmitter~~ that were transmitted before the reset in the transmitting side of the AM RLC entity~~ shall be discarded~~;~~.~~
- ~~When a RESET PDU is received, the receiver shall~~ set the HFN (DL HFN when the RESET PDU is received in UE or UL HFN when the RESET PDU is received in UTRAN) equal to the HFNI field in the received RESET PDU.
- increase with one the UL HFN and DL HFN, and the updated HFN values shall be used for the first transmitted and received PDUs after the reset procedure

~~The RESET ACK PDU shall be transmitted on the DCCH logical channel if the sender is located in the control plane and on the DTCH if it is located in the user plane.~~

~~The RESET ACK PDU has higher priority than data PDUs.~~

11.4.3.1 RESET ACK PDU contents to set

~~The size of the RESET ACK PDU shall be equal to one of the allowed PDU sizes.~~ The Receiver shall:

- ~~The RSN field shall always be set to the same value as in the corresponding RESET PDU.~~ set ~~T~~the hyper frame number indicator field (HFNI) ~~shall be set equal~~ to the currently highest used HFN (DL HFN when the RESET ACK PDU is sent by UTRAN or UL HFN when the RESET ACK PDU is sent by the UE);~~.~~
- set the RSN field to the same value as in the corresponding received RESET PDU.

11.4.4 Reception of the RESET ACK PDU by the S~~s~~ender

Upon reception of a RESET ACK PDU the Sender shall:

- if the Sender has already transmitted a RESET PDU which has not been yet acknowledged by a RESET ACK PDU:
    - if the received RSN value is the same ~~than~~as the one in the corresponding RESET PDU:
        - set the HFN value (DL HFN when the RESET ACK PDU is received in UE or UL HFN when the RESET ACK PDU is received in UTRAN) to the HFNI field of the received RESET ACK PDU;
        - reset the state variables described in subclause 9.4 to their initial values;
        - stop all the timers described in subclause 9.5;
        - reset configurable parameters to their configured values;
        - discard all RLC PDUs in the receiving side of the AM RLC entity;
        - discard all RLC SDUs that were transmitted before the reset in the transmitting side of the AM RLC entity;
        - increase with one the UL HFN and DL HFN, and the updated HFN values shall be used for the first transmitted and received PDUs after the reset procedure;
    - otherwise (if the received RSN value is not the same ~~than~~as the one in the corresponding RESET PDU):
        - discard the RESET ACK PDU;
- otherwise (if the Sender has not transmitted a RESET PDU which has not been yet acknowledged by a RESET ACK PDU):
    - discard the RESET ACK PDU.

~~When the sender is in reset pending state or reset and suspend state and receives a RESET ACK PDU with the same RSN value as in the corresponding RESET PDU, the Timer_RST shall be stopped and the value of the HFN (DL HFN when the RESET ACK PDU is received in UE or UL HFN when the RESET ACK PDU is received in UTRAN) shall be set equal to the HFNI field in the received RESET ACK PDU. The sender resets the state variables in 9.4 to their initial value and resets configurable parameters to their configured value. Both the transmitting and receiving sides of the AM RLC entity are reset. All RLC PDUs in the AM RLC receiver shall be discarded. The RLC SDUs in the AM RLC transmitter that were transmitted before the reset shall be discarded.~~

~~The sender shall enter data transfer ready state if it was in reset pending state and enter local suspend state if it was in reset and suspend state.~~

~~Upon reception of a RESET ACK PDU with a different RSN value as in the corresponding RESET PDU the RESET ACK PDU is discarded.~~

~~Upon reception of a RESET ACK PDU in data transfer ready state or local suspend state, the RESET ACK PDU is discarded.~~

11.4.5 Abnormal cases

11.4.5.1 Timer_RST timeout

Upon expiry of Timer_RST the S~~s~~ender shall:

- ~~-~~ retransmit the RESET PDU and increase VT(RST) with 1. In the retransmitted RESET PDU the value of the RSN field shall not be incremented.

11.4.5.2 Unrecoverable error (VT(RST) ≥ MaxRST)

The Sender shall:

- ~~I~~if VT(RST) becomes larger than or equal to MaxRST:
    - ~~,~~ indicate unrecoverable error ~~shall be indicated~~ to upper layers.

11.4.5.3 Reception of the RESET PDU by the S~~s~~ender

Upon reception of a RESET PDU ~~in acknowledged data READY state, RESET PENDING state, LOCAL SUSPEND state or RESET AND SUSPEND state~~, the S~~s~~ender shall:

- submit a RESET ACK PDU to the lower layer;
- reset the state variables described in subclause 9.4 except VT(RST) to their initial values;
- stop all the timers described in subclause 9.5 except Timer_RST;
- reset configurable parameters to their configured values;
- discard all RLC PDUs in the receiving side of the AM RLC entity;
- discard all RLC SDUs that were transmitted before the reset in the transmitting side of the AM RLC entity;
- set the HFN (DL HFN when the RESET PDU is received in UE or UL HFN when the RESET PDU is received in UTRAN) equal to the HFNI field in the received RESET PDU;
- increase with one the UL HFN and DL HFN, and the updated HFN values shall be used for the first transmitted and received PDUs after the reset procedure.
- ~~stay in its current state.~~

~~respond with a RESET ACK PDU. The sender resets the state variables in 9.4 to their initial value, resets configurable parameters to their configured value. However, VT(RST) and Timer_RST are not reset. Both the transmitting and receiving sides of the AM RLC entity are reset. All RLC PDUs in the AM RLC receiver shall be discarded. The RLC SDUs in the AM RLC transmitter that were transmitted before the reset shall be discarded. The hyper frame number,~~

~~HFN (DL HFN when the RESET is received in UE or UL HFN when the RESET is received in UTRAN) is set equal to the HFNI field in the received RESET PDU. The sender shall stay in its current state. The sender shall enter data transfer ready state or local suspend state only upon reception of a RESET ACK PDU with the same RSN value as in the corresponding RESET PDU when it is in reset pending state or reset and suspend state respectively.~~

11.5 STATUS report transfer procedure

11.5.1 ~~Purpose~~General

The status report transfer procedure is used for transferring of status information between two RLC peer entities, which are operating in acknowledged mode. Figure 11.5 below illustrates the elementary procedure for status report transfer. A status report consists of one or several STATUS PDUs.~~ The receiver is the receiver of AMD PDUs and it is either the UE or the network and the sender is~~~~Information included in a status report is (with the exception of MRW SUFI) sent from the~~ ~~the sender of AMD PDUs and it is either the network or the UE.~~~~Receiver's receiving side to the Sender's transmitting side (see figure 4.4).~~

~~If the RLC AM entity is assigned two logical channels on configuration, one logical channel shall be for AMD PDU transfer and the other for Control PDU transfer.~~In case two logical channels are configured in the uplink, control PDUs are transmitted on the second logical channel. In case two logical channels are configured in the downlink, Control PDUs can be transmitted on any of the two logical channels.

The STATUS PDUs have higher priority than AMD PDUs.

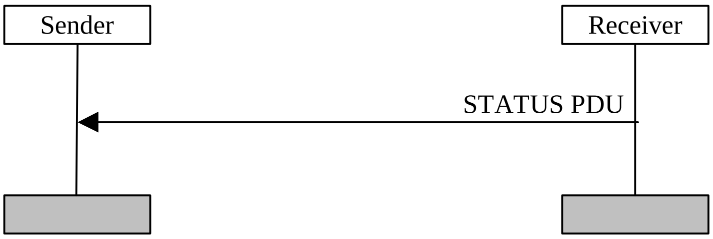


Figure 11.5: Status report transfer procedure

11.5.2 Initiation

The Receiver shall:

~~The receiver in any~~- if one of the following ~~cases initiates this procedure:~~triggers is detected:

1) The ~~poll bit~~"Polling Bit" in a received AMD PDU is set to ~~1.~~"1";

2) ~~Detection of missing PDUs is used and a missing PDU is detected.~~"~~Detection of m~~Missing PDU~~(s)~~ Indicator" is configured and a missing AMD PDU is detected;

3) The ~~timer based STATUS transfer is used~~"Timer based STATUS transfer" is configured and the timer Timer_Status_Periodic has ~~expired.~~expired~~ and periodical status blocking is not configured~~;

- act on the trigger as specified in subclause 9.7.2.~~;~~

~~The receiver shall transmit a status report on the DCCH logical channel if the receiver is located in the control plane and on the DTCH if it is located in the user plane. Separate logical channels can be assigned for AMD PDU transfer and for Control PDU transfer.~~

~~The STATUS PDUs have higher priority than data PDUs.~~

~~There are two functions that can prohibit the receiver from sending a status report. If any of following conditions are fulfilled the sending of the status report shall be delayed, even if any of the conditions above are fulfilled:~~

~~1) STATUS prohibit is used and the timer Timer_Status_Prohibit is active.~~

~~The status report shall be transmitted after the Timer_Status_Prohibit has expired. The receiver shall send only one status report, even if there are several triggers when the timer is active. The rules for when the timer Timer_status_Prohibit is active are defined in subclause 9.5.~~

~~2) The EPC mechanism is used and the timer Timer_EPC is active or VR(EP) is counting down.~~

~~The status report shall be transmitted after the VR(EP) has reached 0. The receiver send only one status report, even if there are several triggers when the timer is active or the counter is counting down. The rules for when the timer Timer_EPC is active are defined in subclause 9.5.~~

~~If the timer based STATUS transfer shall be used and the Timer_Status_Periodic has expired it shall be restarted.~~

~~-If the EPC mechanism shall be used the timer Timer_EPC shall be started and the VR(EP) shall be set equal to the number PDUs requested to be retransmitted.~~

11.5.2.1 Piggybacked STATUS PDU

The Receiver may:

- if STATUS PDU(s) to be sent fit into padding octets in AMD PDU(s) to be sent:
    - ~~It is possible to~~ piggyback a STATUS PDU on the AMD PDU to be sent.

~~an AMD PDU. If a PDU includes padding, a piggybacked STATUS PDU can be inserted instead of the padding. The sending~~Submission of a piggybacked STATUS PDU in an AMD PDU to the lower layer follows the same rules as~~the sending of~~ an ordinary STATUS PDU.

11.5.2.2 STATUS PDU contents to set

On triggering of a status report ~~T~~the Receiver shall:

- if neither the "STATUS prohibit" nor "EPC mechanism" are active:
    - include negative acknowledgements for all AMD PDUs detected as missing;
    - include acknowledgements for all AMD PDUs received up to at least VR(R);
- if "SDU discard with explicit signalling" procedure has been initiated since the last status report was sent:
    - optionally include one MRW SUFI as specified in subclause 11.6.2.2;
- if the MRW SUFI was received in the last status report received:
    - include one MRW_ACK SUFI as specified in subclause 11.6.3.2;
- if the Sender’s transmission window is to be updated:
    - include the WINDOW SUFI;
- if all SUFIs can be accommodated in one STATUS PDU:
    - construct the status report using one STATUS PDU, using one of the allowed PDU sizes;
    - if the SUFIs included do not fill the entire STATUS PDU:
        - ~~The size of~~ terminate the STATUS PDU with the ACK or NO_MORE SUFI;
        - use padding in the remainder of the STATUS PDU;
    - otherwise (SUFIs included fill the entire STATUS PDU):
        - ACK or NO_MORE SUFIs need not be included in that STATUS PDU;
- otherwise (the status report is segmented):

~~shall be equal to~~- construct STATUS PDUs including only complete SUFIs using one of the allowed PDU sizes. The ~~information that needs to be transmitted in a status report can be split into several~~set of STATUS PDUs ~~if one STATUS PDU does not~~shall accommodate all the ~~information. A SUFI cannot be split into several STATUS PDUs.~~SUFIs to form the complete status report. Indication of the same AMD PDU shall not be given in more than one STATUS PDU of a ~~STATUS~~status report, but the ACK SUFI can be present in more than one STATUS PDU of a status ~~report.~~report;

- if any STATUS PDU constructed is not entirely filled with SUFIs:
    - terminate that STATUS PDU with the ACK or NO_MORE SUFI;
    - use padding in the remainder of that STATUS PDU;
- otherwise (SUFIs included fill the entire STATUS PDU):
    - ACK or NO_MORE SUFIs ~~need~~should not be included in that STATUS PDU.

Which SUFI fields to use is implementation dependent~~, but the status report shall include information about PDUs that have been received and information about all PDUs detected as missing~~. Bitmap SUFI is used to indicate both received and/or missing AMD PDUs. List SUFI and/or Relative List SUFI are used to indicate missing AMD PDUs only. Acknowledgement SUFI is used to indicate the received AMD PDUs. (For SUFI details see 9.2.2.11.) No information shall be given for AMD PDUs with SN≥VR(H), i.e. AMD PDUs that have not yet reached the ~~receiver.~~Receiver.

11.5.2.3~~X~~ Submission of STATUS PDUs to the lower layer

The Receiver shall:

- inform the lower layer of the STATUS PDUs scheduled for transmission;
- submit to the lower layer, the requested number of PDUs (STATUS PDUs, piggybacked AMD / STATUS PDUs and optionally AMD PDUs, see also subclause 11.3.2.2~~X~~).
- if "Timer based STATUS transfer" is configured and the timer Timer_Status_Periodic has expired:
    - restart the timer Timer_Status_Periodic according to subclause 9.5 f);
- if the "EPC mechanism" is configured:
    - start the timer Timer_EPC according to subclause 9.5 c), and set VR(EP) equal to the number of AMD PDUs requested to be retransmitted;
- if the STATUS PDU includes the MRW SUFI:
    - start the timer Timer_MRW according to subclause 9.5 i).~~;~~

~~Padding shall be inserted if the SUFI fields do not fill an entire STATUS PDU. If the PDU contains padding the last SUFI field shall be either an ACK SUFI or a NO_MORE SUFI. If there is no padding in the STATUS PDU, NO_MORE SUFI or ACK SUFI does not need to be included in the STATUS PDU.~~

11.5.3 Reception of the STATUS PDU by the ~~sender~~Sender

~~The sender shall upon~~Upon reception of the STATUS PDU / piggybacked STATUS PDU, the Sender shall:

~~PDU/piggybacked STATUS PDU update~~- update the state variables VT(A) and VT(MS) according to the received STATUS PDU/piggybacked STATUS ~~PDU.~~PDU;

~~If~~- if the STATUS PDU includes negatively acknowledged AMD PDUs:

~~PDUs, the~~- the acknowledged data transfer procedure shall be initiated and these AMD PDUs shall be retransmitted. Retransmitted AMD PDUs shall have higher priority than AMD PDUs to be transmitted for the first time;

~~If a~~- if an AMD PDU is ~~indicated as missing~~negatively acknowledged more than once in a STATUS PDU:

~~PDU, the~~- the AMD PDU shall be retransmitted only once. ~~Retransmitted PDUs have higher priority than new PDUs.~~

- if the STATUS PDU includes the MRW SUFI:
  - take the actions specified in subclause 11.6.3;
- if the STATUS PDU includes the MRW_ACK SUFI:
  - take the actions specified in subclause 11.6.4;
- if the STATUS PDU includes the WINDOW SUFI:
  - update the current transmitter window size, VT(WS).

11.5.4 Abnormal cases

11.5.4.1 VR(EP) ~~reache~~equals zero and the requested AMD PDUs have not been received

If the EPC mechanism is ~~used~~configured and VR(EP) ~~has reached~~equals zero and not all AMD PDUs requested for retransmission have been received, the ~~receiver~~Receiver shall:

- Retransmit the status report. The retransmitted status report may contain new or different SUFI fields in order to indicate that some previously lost AMD PDUs have been received and that some ~~new~~additional AMD PDUs have been lost.

11.6 SDU discard with explicit signalling procedure

11.6.1 General~~Purpose~~

The SDU discard with explicit signalling procedure is used for discarding SDUs and transferring the discard information between two peer entities, which are operating in acknowledged mode. The Sender shall discard an SDU that has not been successfully transmitted for a period of time or for a number of retransmissions, and send a Move Receiving Window (MRW) SUFI to the Receiver. According to the MRW SUFI, the Receiver shall discard AMD PDUs carrying that SDU and update the receiving window. Figure 11.6 below illustrates the elementary procedure for SDU discard with explicit signalling. ~~An SDU can be discarded with explicit signalling when MaxDAT number of retransmissions is reached or the transmission time exceeds a predefined value (Timer_Discard) for an SDU in acknowledged mode RLC. Move Receiving Window (MRW) command is sent to the receiver so that AMD PDUs carrying that SDU are discarded in the receiver and the receiver window is updated accordingly. Note that when the concatenation function is active, PDUs carrying segments of other SDUs that have not timed out shall not be discarded. If Send MRW is not configured and no segments of an SDU were submitted to a lower layer, the SDU is simply discarded in the transmitter without notification to the receiver. If Send MRW is configured, a Move Receiving Window request shall be sent to the receiver even if no segments of the SDU were submitted to a lower layer. The Send MRW is used when the AM RLC entity is connected to a PDCP layer which supports lossless SRNS relocation.~~

~~The MRW command is defined as a super-field in the RLC STATUS PDU, and can be piggybacked to status information of transmissions in the opposite direction.~~

~~Figure 11.6 below illustrates the elementary procedure for SDU discard with explicit signalling. The sender is the sender of AMD PDUs and it is either the UE or the network and the receiver is the receiver of AMD PDUs and it is either the network or the UE.~~

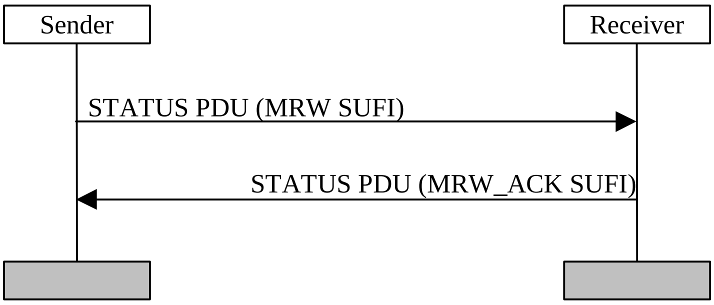


Figure 11.6: SDU discard with explicit signalling

11.6.2 Initiation

The Sender shall initiate the SDU discard with explicit signalling procedure if one of the following triggers is detected: ~~This procedure is initiated by the sender when any of the following conditions is fulfilled:~~

- ~~1)~~ "Timer based SDU discard with explicit signalling" is configured~~used~~, Timer_Discard expires for an SDU, and one or more segments of the SDU have been submitted to a lower layer;~~.~~
- ~~2)~~ "Timer based SDU discard with explicit signalling" is configured~~used~~, Timer_Discard expires for an SDU, and "Send MRW" is configured;~~.~~
- ~~3)~~ "SDU discard after MaxDAT number of retransmissions" is configured~~used~~, and MaxDAT number of retransmissions is reached (i.e. VT(DAT) ≥ MaxDAT) for an AMD ~~S~~PDU.

Upon initiation of the SDU discard with explicit signalling procedure, the Sender shall:

- if "Timer based SDU discard with explicit signalling" is configured:
  - discard all SDUs up to and including the SDU for which the timer Timer_Discard expired;
  - not discard AMD PDUs carrying segments of other SDUs whose timers have not expired;
- if "SDU discard after MaxDAT number of retransmissions" is configured:
  - discard all SDUs that have segments in AMD PDUs with SN inside the interval VT(A) ≤ SN ≤ X, where X is the value of the SN of the AMD PDU with VT(DAT) ≥ MaxDAT;
- if more than 15 discarded SDUs are to be informed to the Receiver:
  - assemble an MRW SUFI with the discard information of the first 15 SDUs. The discard information of the rest SDUs shall be included in another MRW SUFI which shall be sent by the next SDU discard with explicit signalling procedure (after the current SDU discard with explicit signalling procedure is terminated);
- otherwise (less than or equal to 15 discarded SDUs are to be informed to the Receiver):
  - assemble an MRW SUFI with the discard information of the SDUs;
- include the MRW SUFI in the next STATUS PDU/piggybacked STATUS PDU to be transmitted, according to subclause 11.5.2;
- if SN_MRW$_{LENGTH}$ in the MRW SUFI >VT(S):
  - update VT(S) to SN_MRW$_{LENGTH}$;
- start a timer Timer_MRW according to subclause 9.5.

~~The sender shall discard all PDUs that contain segments of the associated SDUs. If the concatenation function is active, PDUs carrying segments of other SDUs that have not timed out shall not be discarded. VT(A) shall be updated when the procedure is terminated, and VT(S) shall be updated when a new MRW SUFI which includes SN_MRW$_{LENGTH}$ >VT(S) is transmitted.~~

~~The sender shall transmit a status report on the DCCH logical channel if the sender is located in the control plane and on the DTCH if it is located in the user plane.~~

~~This status report is sent even if the 'STATUS prohibit' is used and the timer 'Timer_Status_Prohibit' is active, or if the 'EPC mechanism' is used and the timer 'Timer_EPC' is active or 'VR(EP)' is counting down.~~

~~The STATUS PDUs have higher priority than data PDUs.~~

~~The sender shall start timer Timer_MRW.~~ If a new SDU discard with explicit signalling procedure is triggered when the timer Timer_MRW is active~~running~~, no new MRW SUFIs shall be sent before the current SDU discard with explicit signalling procedure is terminated by one of the termination criteria specified in subclause 11.6.4.

11.6.2.1 ~~Piggybacked STATUS PDU~~Void

~~It is possible to piggyback a STATUS PDU on an AMD PDU. If a PDU includes padding a piggybacked STATUS PDU can be inserted instead of the padding.~~

11.6.2.2 STATUS PDU contents to set

~~The size of the STATUS PDU shall be equal to one of the allowed PDU sizes. The discard information shall not be split into several MRW SUFIs. If the discard information cannot be fit into one MRW SUFI, another SDU discard with explicit signalling procedure shall be initiated after the current procedure is terminated.~~

~~The status report shall include the MRW SUFI, other SUFI fields can be used additionally. MRW SUFI shall convey information about the discarded SDU(s) to the receiver.~~The Sender shall:

- if "Send MRW" is configured:
  - if the last discarded SDU ended in an AMD PDU, and its LI is present in the same AMD PDU, and no new SDU is present inside this AMD PDU:
    - set the last SN_MRW$_i$ field in the MRW SUFI to 1 + SN of the AMD PDU which contains the LI of the last discarded SDU;
    - set the N$_{LENGTH}$ field in the MRW SUFI to "0000";
  - otherwise:
    - set the last SN_MRW$_i$ field in the MRW SUFI to the SN of the AMD PDU which contains the LI of the last discarded SDU;
    - set the N$_{LENGTH}$ field in the MRW SUFI so that the last data octet to be discarded in the Receiver shall be the octet indicated by the N$_{LENGTH}$:th LI field of the AMD PDU which contains the LI of the last discarded SDU;
  - set each of the other SN_MRW$_i$ fields in the MRW SUFI to the SN of the AMD PDU which contains the LI of the i:th discarded SDU;
- otherwise ("Send MRW" is not configured):
  - if the last SDU to be discarded in the Receiver ended in an AMD PDU, and its LI is present in the same AMD PDU, and no new SDU is present inside this AMD PDU:
    - set the last SN_MRW$_i$ field in the MRW SUFI to 1 + SN of the AMD PDU which contains the LI of the last SDU to be discarded in the Receiver;
    - set the N$_{LENGTH}$ field in the MRW SUFI to "0000";
  - otherwise:
    - set the last SN_MRW$_i$ field in the MRW SUFI to the SN of the AMD PDU which contains the LI of the last SDU to be discarded in the Receiver;

        - set the N$_{LENGTH}$ field in the MRW SUFI so that the last data octet to be discarded in the Receiver shall be the octet indicated by the N$_{LENGTH}$:th LI field of the AMD PDU which contains the LI of the last SDU to be discarded in the Receiver;

    - optionally set each of the other SN_MRW$_i$ fields in the MRW SUFI to the SN of the AMD PDU which contains the LI of the i:th SDU to be discarded in the Receiver;

- if the MRW SUFI contains only one SN_MRW$_i$ field and the value of SN_MRW$_i$ field ≥ VT(A)+Configured_Tx_Window_Size:

    - set the LENGTH field in the MRW SUFI to "0000";

- otherwise:

    - set the LENGTH field in the MRW SUFI to the number of SN_MRW$_i$ fields in the same MRW SUFI. In this case, SN_MRW$_1$ shall be in the interval VT(A) ≤ SN_MRW$_1$ < VT(A)+Configured_Tx_Window_Size;

- include the MRW SUFI in the next STATUS PDU/piggybacked STATUS PDU to be transmitted, according to subclause 11.5.2;~~When Send MRW is configured, the MRW SUFI shall contain the information about each discarded SDU (see subclause 9.2.2.11.8). In order to discard a single SDU that ends in a PDU with SN≥ VT(A)+Configured_Tx_Window_Size, the LENGTH field in the MRW SUFI shall be set to "0000". If more than one SDU are discarded with the same MRW SUFI, at least the first discarded SDU must end (i.e. the LI must be located) in a PDU with SN in the interval VT(A)≤ SN <VT(A)+Configured_Tx_Window_Size. Otherwise, multiple SDU discard with explicit signalling procedures need to be performed in order to signal the set of discarded SDUs.~~

~~When Send MRW is not configured, the MRW SUFI shall contain the information about the last SDU to be discarded in the receiver. The information about the other SDUs to be discarded in the receiver may optionally be contained in the MRW SUFI (see subclause 9.2.2.11.8). If the MRW SUFI contains only the information about the last SDU to be discarded in the receiver and if this SDU ends in a PDU with SN≥ VT(A)+Configured_Tx_Window_Size, the LENGTH field in the MRW SUFI shall be set to "0000". If the MRW SUFI contains information about more than one discarded SDU, at least the first discarded SDU must end (i.e. the LI must be located) in a PDU with SN in the interval VT(A)≤ SN <VT(A)+Configured_Tx_Window_Size.~~

~~Padding shall be inserted if the SUFI fields do not fill the entire STATUS PDU. If the STATUS PDU contains padding the last SUFI field shall be either an ACK SUFI or a NO_MORE SUFI. If there is no padding in the STATUS PDU, NO_MORE SUFI or ACK SUFI does not need to be included in the STATUS PDU.~~

11.6.3 Reception of the STATUS PDU by the ~~receiver~~Receiver

~~The receiver shall u~~Upon reception of the STATUS PDU/piggybacked STATUS PDU containing an MRW SUFI, the Receiver shall:~~discard PDUs and update the state variables VR(R), VR(H) and VR(MR) according to the received STATUS PDU/piggybacked STATUS PDU.~~

- if the LENGTH field in the received MRW SUFI is "0000":

    - consider SN_MRW$_1$ to be above or equal to VR(R);

- otherwise:

    - consider SN_MRW$_1$ to be less than VR(MR);

- consider all the SN_MRW$_i$s other than SN_MRW$_1$ to be in sequential order within the list and sequentially above or equal to SN_MRW$_{i-1}$.

- discard AMD PDUs up to and including the PDU with sequence number SN_MRW$_{LENGTH}$–1;

- if the N$_{LENGTH}$ field in the received MRW SUFI is "0000":

    - reassemble from the first data octet of the AMD PDU with sequence number SN_MRW$_{LENGTH}$ after the discard;

- otherwise:

  - discard further the data octets in the AMD PDU with sequence number SN_MRW$_{LENGTH}$ up to and including the octet indicated by the N$_{LENGTH}$:th LI field of the PDU with sequence number SN_MRW$_{LENGTH}$;
  - reassemble from the succeeding data octet in the AMD PDU with sequence number SN_MRW$_{LENGTH}$ after the discard;
- ~~Additionally, when~~if "Send MRW" is configured~~,~~:
  - ~~the receiver shall~~ inform ~~the upper layer~~upper layers about all of the discarded SDUs that were not previously delivered to upper layer or discarded by other MRW SUFIs~~.~~;
- update the state variables VR(R), VR(H) and VR(MR) according to the received STATUS PDU/piggybacked STATUS PDU;
- include an MRW_ACK SUFI in the next STATUS PDU/piggybacked STATUS PDU to be transmitted, according to subclause 11.5.2; ~~The receiver shall initiate the transmission of a status report containing an MRW_ACK SUFI.~~

11.6.3.1~~X~~ STATUS PDU contents to set

The Receiver shall:

- ~~In the MRW_ACK SUFI,~~set the SN_ACK field in the MRW_ACK SUFI ~~shall be set~~ to the new value of VR(R), updated after reception of the MRW SUFI~~.~~;
- if the SN_ACK field in the MRW_ACK SUFI is set equal to the SN_MRW$_{LENGTH}$ field in the received MRW SUFI:
  - set ~~T~~the N field in the MRW_ACK SUFI ~~shall be set~~ to the NLENGTH field in the received MRW SUFI;
- otherwise:
  - ~~if the SN_ACK field is equal to SN_MRWLENGTH. Otherwise~~set the N field in the MRW_ACK SUFI ~~shall be set~~ to "0000"~~0.~~;
  - include the MRW_ACK SUFI in the next STATUS PDU/piggybacked STATUS PDU to be transmitted, according to subclause 11.5.2;~~The last discarded data octet is the octet indicated by the N$_{LENGTH}$:th LI field of the PDU with sequence number SN_MRW$_{LENGTH}$ and the succeeding data octet is the first data octet to be reassembled after the discard. When N$_{LENGTH}$ = 0, the first data octet of the PDU with sequence number SN_MRW$_{LENGTH}$ is the first data octet to be reassembled after the discard.~~

~~If LENGTH="0000", the sequence number SN_MRW$_1$ is considered to be above or equal to VR(R). Else, the sequence number SN_MRW$_1$ is considered to be less than VR(MR). All the SN_MRW$_i$s other than SN_MRW$_1$ are considered to be in sequential order within the list and sequentially above or equal to SN_MRW$_{i-1}$.~~

11.6.4 Termination

The Sender shall terminate ~~T~~the SDU discard with explicit signalling procedure ~~is terminated in the sender~~ if one of the following criteria is fulfilled~~in the following cases~~:

- ~~1. On the reception of a~~A STATUS PDU/piggybacked STATUS PDU ~~which~~ containings an MRW_ACK SUFI is received, and~~with~~the SN_ACK field in the received MRW_ACK SUFI > the SN_MRW$_{LENGTH}$ field in the transmitted MRW_SUFI, and ~~with~~ the N field in the received MRW_ACK SUFI is set equal to "0000"~~zero.~~;

~~2. On the reception of a STATUS PDU which contains an ACK SUFI indicating VR(R) > SN_MRW$_{LENGTH}$~~

- ~~3. On reception of a~~A STATUS PDU/piggybacked STATUS PDU ~~which~~ containings an MRW_ACK SUFI is received, and~~with~~the SN_ACK field in the received MRW_ACK SUFI = the SN_MRW$_{LENGTH}$ field in the transmitted MRW_SUFI, and ~~with~~ the N field in the received MRW_ACK SUFI is set equal to the N$_{LENGTH}$ field~~indicated~~ in the transmitted MRW SUFI~~.~~;
- A STATUS PDU/piggybacked STATUS PDU containing an ACK SUFI is received, and the LSN field in the received ACK SUFI > the SN_MRW$_{LENGTH}$ field in the transmitted MRW SUFI.

~~If one of the termination criteria above is fulfilled,~~ Upon termination of the SDU discard with explicit signalling procedure, the Sender shall:

- stop the timer Timer_MRW;
- update VT(A) and VT(MS) according to the received STATUS PDU/piggybacked STATUS PDU.~~ shall be stopped and the discard procedure is terminated.~~

The Sender shall not confirm to upper layers ~~T~~the SDUs that are requested to be discarded~~ shall not be confirmed to upper layers~~.

~~When VT(MRW) reaches MaxMRW, the procedure is terminated and an RLC reset shall be performed.~~

11.6.5 Expiration of timer Timer_MRW

If Timer_MRW expires before the SDU discard with explicit signalling procedure is terminated, the Sender shall:

- retransmit the MRW SUFI. The retransmitted ~~shall be retransmitted, VT(MRW) is incremented by one and Timer_MRW restarted.~~ MRW SUFI shall be exactly the same as previously transmitted even though some new SDUs would have been discarded during the running of the Timer_MRW~~.~~;
- increment VT(MRW) by one;
- restart the timer Timer_MRW.~~ If the retransmitted STATUS PDU contains other SUFIs than the MRW SUFI, the status information indicated by these SUFIs shall be updated.~~

11.6.6 Abnormal cases

11.6.6.1 Reception of ~~O~~obsolete/corrupted MRW SUFI~~command~~ by the Receiver

If the received MRW SUFI~~command~~ contains outdated information about the receiving~~er~~ window (receiving~~er~~ window already moved further than MRW SUFI~~command~~ is indicating), the Receiver shall:

- discard the MRW SUFI;~~command shall be discarded and~~
- set the SN_ACK field in the MRW_ACK SUFI to the current value of VR(R);
- set the N field in the MRW_ACK SUFI to "0000";
- include the MRW_ACK SUFI in the next STATUS PDU/piggybacked STATUS PDU to be transmitted, according to subclause 11.5.2; ~~a status report containing SUFI MRW_ACK shall be transmitted indicating the value of VR(R) and the N field shall be set to zero.~~

11.6.6.2 VT(MRW) equals MaxMRW

If the number of retransmission of an MRW SUFI~~command~~ (i.e. VT(MRW)) ~~reache~~equals MaxMRW, the Sender shall:

- terminate the SDU discard with explicit signalling procedure;
- stop the timer Timer_MRW;
- deliver an error indication ~~shall be passed~~ to upper layers;~~ and~~
- initiate the RLC RESET procedure (see clause 11.4).~~shall be performed.~~

11.6.6.3 Reception of obsolete/corrupted MRW_ACK SUFI by the Sender

The Sender shall discard ~~T~~the received MRW_ACK SUFI if one of the following cases occurs:~~shall be discarded in the following cases.~~

- ~~1. If~~The timer Timer_MRW is not active~~.~~;

- ~~2. If t~~The SN_ACK field in the received MRW_ACK SUFI < the SN_MRW$_{LENGTH}$ field in the transmitted MRW SUFI~~.~~;
- ~~3. If t~~The SN_ACK field in the received MRW_ACK SUFI = ~~is equal to~~ the SN_MRW$_{LENGTH}$ field in the transmitted MRW SUFI, and the N field in the received MRW_ACK SUFI is not equal to the N$_{LENGTH}$ field in the transmitted MRW SUFI;
- ~~4. If t~~The SN_ACK field in the received MRW_ACK SUFI > the SN_MRW$_{LENGTH}$ field in the transmitted MRW SUFI, and the N field in the received MRW_ACK SUFI is not equal to "0000"~~zero~~.

11.7 Void

11.8 Void

TSG-RAN Working Group 2 #23
Helsinky, Finland, 27 - 31 August 2001

Tdoc R2-012157

CR-Form-v3

CHANGE REQUEST

⌘	**25.322**	**CR**	**149**	⌘	rev	**r1**	⌘	Current version:	**3.7.0**	⌘		

*For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: ⌘ (U)SIM □ ME/UE ☒ Radio Access Network □ Core Network □

Title: ⌘	Corrections to RLC State Variables			
Source: ⌘	TSG-RAN WG2			
Work item code: ⌘	TEI		***Date:*** ⌘	2001-07-02
Category: ⌘	**F**		***Release:*** ⌘	R99

Use one of the following categories:
F *(essential correction)*
A *(corresponds to a correction in an earlier release)*
B *(Addition of feature),*
C *(Functional modification of feature)*
D *(Editorial modification)*
Detailed explanations of the above categories can be found in 3GPP TR 21.900.

Use one of the following releases:

2	*(GSM Phase 2)*
R96	*(Release 1996)*
R97	*(Release 1997)*
R98	*(Release 1998)*
R99	*(Release 1999)*
REL-4	*(Release 4)*
REL-5	*(Release 5)*

Reason for change: ⌘ The current text is inaccurate as to the definition of the parameter MaxDAT, MaxMRW and MaxRST. It is not clear best on the specification whether the last transmission of the corresponding PDU will ever be transmitted. The behaviour is ambiguous
It is not clearly specified whether the UE needs to discard the SDUs at the transmitter without explicit signalling when sendMRW is not configured or whether it may.

Summary of change: ⌘ It was clarified that MaxDAT includes both the initial transmission and the subsequent retransmissions.

~~It was clarified that the UE may discard SDUs at the transmitter.~~

It was clarified that the UE shall not make the last transmission (of an AMD PDU, RESET PDU or MRW SUFI depending on the case) before a given counter reaches its maximum value. This implies that the definition of the parameter does not correspond to the functionality. The definitions were changed to align them with the functionality.

The CR has isolated impact and should be seen as a clarification

Consequences if not approved: ⌘ Unclear specification.

Clauses affected: ⌘ 9.6, 9.7.3, 9.7.3.3, 9.7.3.4, 11.3.4.4, 11.3.4.7 (new), 11.4.5.1, 11.6.1, 11.6.2, 11.6.5

Other specs affected:	⌘ □	Other core specifications	⌘	25.322 v4.1.0, CR 150
	□	Test specifications		
	□	O&M Specifications		

Other comments: ⌘	

How to create CRs using this form:

Comprehensive information and tips about how to create CRs can be found at: http://www.3gpp.org/3G_Specs/CRs.htm. Below is a brief summary:

1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.

2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be downloaded from the 3GPP server under ftp://www.3gpp.org/specs/ For the latest version, look for the directory name with the latest date e.g. 2000-09 contains the specifications resulting from the September 2000 TSG meetings.

3) With "track changes" disabled, paste the entire CR form (use CTRL-A to select it) into the specification just in front of the clause containing the first piece of changed text. Delete those parts of the specification which are not relevant to the change request.

9.6 Protocol Parameters

The values of the protocol parameters in this subclause are signalled by upper layers.

a) MaxDAT.

~~It is the~~The maximum number of transmissions of a PDU is equal to MaxDAT – 1.~~value for the number of retransmissions of a PDU plus one.~~ This parameter is an upper limit of counter VT(DAT). When the value of VT(DAT) comes to MaxDAT, either RLC RESET procedure or SDU discard procedure shall be initiated according to configuration by upper layer.

b) Poll_PDU.

This parameter indicates how often the transmitter should poll the receiver in case of polling every Poll_PDU PDU. This is an upper limit for the VT(PDU) state variable, when VT(PDU) reaches Poll_PDU a poll is transmitted to the peer entity.

c) Poll_SDU.

This parameter indicates how often the transmitter should poll the receiver in case of polling every Poll_SDU SDU. This is an upper limit for the VT(SDU) state variable, when VT(SDU) reaches Poll_SDU a poll is transmitted to the peer entity.

d) Poll_Window.

This parameter indicates when the transmitter should poll the receiver in case of performing window-based polling. The range of values of this parameter shall be 0 ≤ Poll_Window ≤ 100. A poll is triggered for each PDU when J ≥ Poll_Window, where J is the window transmission percentage defined by

$$J = \frac{(4096+VT(S) - VT(A))\ \mathbf{mod}\ 4096}{VT(WS)} * 100\ ,$$

where the constant 4096 is the modulus for AM described in Subclause 9.4.

e) MaxRST.

The maximum number of transmissions of a RESET PDU is equal to MaxRST – 1. ~~It is the maximum value for the number of retransmission of RESET PDU plus one.~~ This parameter is an upper limit of counter VT(RST). When the value of VT(RST) comes to MaxRST, unrecoverable error shall be indicated to upper layer.

f) Configured_Tx_Window_Size.

The maximum allowed transmitter window size.

g) Configured_Rx_Window_Size.

The allowed receiver window size.

h) MaxMRW.

The maximum number of transmissions of a MRW command is equal to MaxMRW – 1.~~It is the maximum value for the number of retransmissions of a MRW command plus one.~~ This parameter is an upper limit of counter VT(MRW). When the value of VT(MRW) comes to MaxMRW, RLC RESET procedure shall be initiated.

9.7.3 SDU discard function for acknowledged, unacknowledged, and transparent mode

The SDU discard function allows to discharge RLC PDU from the buffer on the transmitter side, when the transmission of the RLC PDU does not success for a long time. The SDU discard function allows to avoid buffer overflow. There will be several alternative operation modes of the RLC SDU discard function. Upper layers control, which discard function shall be used for each RLC entity.

The following is a list of operation modes for the RLC SDU discard function.

Table 9.2: List of criteria that control when to perform SDU discard

Operation mode	Presence
Timer based discard, with explicit signalling	Network controlled
Timer based discard, without explicit signalling	Network controlled
SDU discard after MaxDAT number of ~~re~~transmissions	Network controlled
No_discard after MaxDAT number of ~~re~~transmissions	Network controlled

9.7.3.3 SDU discard after MaxDAT number of ~~re~~transmissions

This alternative uses the number of ~~re~~transmissions as a trigger for SDU discard, and is therefore only applicable for acknowledged mode RLC. This makes the SDU discard function dependent of the channel rate. Also, this variant of the SDU discard function strives to keep the SDU loss rate constant for the connection, on the cost of a variable delay. SDU discard is triggered at the transmitter, and a MRW command is necessary to convey the discard information to the receiver, like in the timer-based discard with explicit signalling.

9.7.3.4 No_discard after MaxDAT number of ~~re~~transmissions

This alternative uses the number of ~~re~~transmissions, and is therefore only applicable for acknowledged mode RLC. Reset procedure shall be initiated after MaxDAT number of ~~re~~transmissions of an AMD PDU (see subclause 11.3.4.4).

11.3.4 Abnormal cases

11.3.4.1 Timer_Poll timeout

Upon expiry of the Timer_Poll, the sender shall retransmit the poll. The poll can be retransmitted in either a new PDU or a retransmitted PDU.

11.3.4.2 Receiving a PDU outside the receiving window

Upon reception of a PDU with sequence number outside the interval VR(R)≤SN<VR(MR), the receiver shall discard the PDU. The poll bit shall be considered even if a complete PDU is discarded.

11.3.4.3 Timer_Discard timeout

11.3.4.3.1 SDU discard with explicit signalling

Upon expiry of Timer_Discard, the sender shall initiate the SDU discard with explicit signalling procedure. In the case where the TFC selection exchange has been initiated by sending the RLC Entity Info parameter to MAC, the UE may wait until after it provides MAC with the requested set of PDUs before discarding the afore-mentioned SDU.

11.3.4.4 VT(DAT) ≥ MaxDAT

If SDU discard after MaxDAT number of ~~re~~transmission is used and VT(DAT) ≥ MaxDAT for any PDU, the sender shall initiate the SDU discard with explicit signalling procedure. All the SDUs that have segments in PDUs with sequence numbers inside the interval VT(A) ≤ SN ≤ SN of the PDU with VT(DAT) ≥ MaxDAT shall be discarded.

If No_discard after MaxDAT number of ~~re~~transmissions is used, the sender shall initiate the RLC reset procedure when VT(DAT) ≥ MaxDAT.

11.3.4.5 Invalid length indicator value

If the length indicator of a PDU has a value that is larger than the PDU size – RLC header size and is not one of the predefined values listed in the table of subclause 9.2.2.8, the PDU shall be discarded and treated as a missing PDU.

11.3.4.6 Length Indicator value reserved for AMD PDU

Upon reception of an AMD PDU that contains Length Indicator value reserved for AMD PDU, the receiver shall discard that AMD PDU.

11.3.4.7 VT(DAT) ≥ MaxDAT-1

The Sender shall not perform the transmission of the PDU. Instead, it will only increment the corresponding VT(DAT).

11.4.5 Abnormal cases

11.4.5.1 Timer_RST timeout

~~Upon expiry of Timer_RST the sender shall retransmit the RESET PDU and increase VT(RST) with 1. In the retransmitted RESET PDU the value of the RSN field shall not be incremented.~~

If Timer_RST expires before the reset procedure is terminated, the Sender shall:

- if VT(RST)<MaxRST-1:
  - set the RESET PDU as previously transmitted (even if additional SDUs were discarded in the mean-time).
  - transmit RESET PDU;
- increment VT(RST) by one;
- restart Timer_RST.

11.6.1 Purpose

An SDU can be discarded with explicit signalling when MaxDAT number of ~~re~~transmissions is reached or the transmission time exceeds a predefined value (Timer_Discard) for an SDU in acknowledged mode RLC. Move Receiving Window (MRW) command is sent to the receiver so that AMD PDUs carrying that SDU are discarded in the receiver and the receiver window is updated accordingly. Note that when the concatenation function is active, PDUs carrying segments of other SDUs that have not timed out shall not be discarded. If Send MRW is not configured and no segments of an SDU were submitted to a lower layer, the SDU ~~is~~ may ~~simply~~ beis simply discarded in the transmitter without notification to the receiver. If Send MRW is configured, a Move Receiving Window request shall be sent to the receiver even if no segments of the SDU were submitted to a lower layer. The Send MRW is used when the AM RLC entity is connected to a PDCP layer which supports lossless SRNS relocation.

The MRW command is defined as a super-field in the RLC STATUS PDU, and can be piggybacked to status information of transmissions in the opposite direction.

Figure 11.6 below illustrates the elementary procedure for SDU discard with explicit signalling. The sender is the sender of AMD PDUs and it is either the UE or the network and the receiver is the receiver of AMD PDUs and it is either the network or the UE.

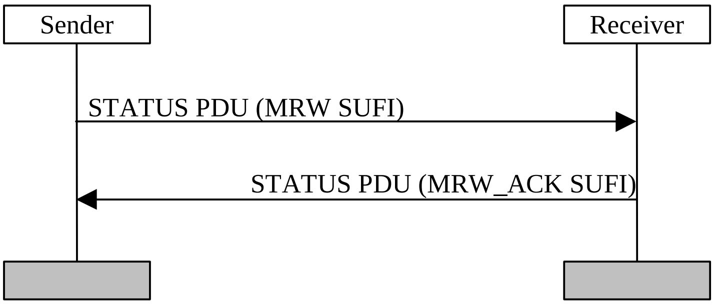


Figure 11.6: SDU discard with explicit signalling

11.6.2 Initiation

This procedure is initiated by the sender when any of the following conditions is fulfilled:

1) Timer based SDU discard with explicit signalling is used, Timer_Discard expires for an SDU, and one or more segments of the SDU have been submitted to a lower layer.

2) Timer based SDU discard with explicit signalling is used, Timer_Discard expires for an SDU, and Send MRW is configured.

3) SDU discard after MaxDAT number of ~~re~~transmissions is used, and MaxDAT number of ~~re~~transmissions is reached for an SDU.

The sender shall discard all PDUs that contain segments of the associated SDUs. If the concatenation function is active, PDUs carrying segments of other SDUs that have not timed out shall not be discarded. VT(A) shall be updated when the procedure is terminated, and VT(S) shall be updated when a new MRW SUFI which includes SN_MRW_{LENGTH} >VT(S) is transmitted.

The sender shall transmit a status report on the DCCH logical channel if the sender is located in the control plane and on the DTCH if it is located in the user plane.

This status report is sent even if the 'STATUS prohibit' is used and the timer 'Timer_Status_Prohibit' is active, or if the 'EPC mechanism' is used and the timer 'Timer_EPC' is active or 'VR(EP)' is counting down.

The STATUS PDUs have higher priority than data PDUs.

The sender shall start timer Timer_MRW. If a new SDU discard procedure is triggered when Timer_MRW is running, no new MRW SUFIs shall be sent before the current SDU discard procedure is terminated by one of the termination criteria.

11.6.5 Expiration of timer Timer_MRW

~~If Timer_MRW expires before the discard procedure is terminated, the MRW SUFI shall be retransmitted, VT(MRW) is incremented by one and Timer_MRW restarted. MRW SUFI shall be exactly the same as previously transmitted even though some new SDUs would have been discarded during the running of the Timer_MRW. If the retransmitted STATUS PDU contains other SUFIs than the MRW SUFI, the status information indicated by these SUFIs shall be updated.~~

If Timer_MRW expires before the discard procedure is terminated, the Sender shall:

- if VT(MRW)<MaxMRW-1:
  - set the MRW SUFI as previously transmitted (even if additional SDUs were discarded in the mean-time).
  - include the MRW SUFI in a new status report (if other SUFIs are included, their contents shall be updated);
  - transmit the status report by either including it in a STATUS PDU or piggy-backed in an AMD PDU;
- increment VT(MRW) by one;
- restart Timer_MRW for this discard procedure.

TSG-RAN Working Group 2 #23
Helsinky, Finland, 27 - 31 August 2001

Tdoc R2-012158

CR-Form-v3

CHANGE REQUEST

⌘ **25.322** **CR** **150** ⌘ rev **-** ⌘ Current version: **4.1.0** ⌘

*For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: ⌘ (U)SIM □ ME/UE ☒ Radio Access Network □ Core Network □

Title:	⌘	Corrections to RLC State Variables		
Source:	⌘	TSG-RAN WG2		
Work item code:	⌘	TEI	***Date:*** ⌘	2001-07-02
Category:	⌘	**A**	***Release:*** ⌘	REL-4

Use one of the following categories:
F *(essential correction)*
A *(corresponds to a correction in an earlier release)*
B *(Addition of feature),*
C *(Functional modification of feature)*
D *(Editorial modification)*
Detailed explanations of the above categories can be found in 3GPP TR 21.900.

Use one of the following releases:

2	*(GSM Phase 2)*
R96	*(Release 1996)*
R97	*(Release 1997)*
R98	*(Release 1998)*
R99	*(Release 1999)*
REL-4	*(Release 4)*
REL-5	*(Release 5)*

Reason for change: ⌘ The current text is inaccurate as to the definition of the parameter MaxDAT, MaxMRW and MaxRST. It is not clear best on the specification whether the last transmission of the corresponding PDU will ever be transmitted. The behaviour is ambiguous
It is not clearly specified whether the UE needs to discard the SDUs at the transmitter without explicit signalling when sendMRW is not configured or whether it may.

Summary of change: ⌘ It was clarified that MaxDAT includes both the initial transmission and the subsequent retransmissions.

~~It was clarified that the UE may discard SDUs at the transmitter.~~

It was clarified that the UE shall not make the last transmission (of an AMD PDU, RESET PDU or MRW SUFI depending on the case) before a given counter reaches its maximum value. This implies that the definition of the parameter does not correspond to the functionality. The definitions were changed to align them with the functionality.

Consequences if not approved: ⌘ Unclear specification.

Clauses affected: ⌘ 9.6, 9.7.3, 9.7.3.3, 9.7.3.4, 11.3.4.4, 11.3.4.7 (new), 11.4.5.1, 11.6.1, 11.6.2, 11.6.5

Other specs affected:	⌘	□ Other core specifications	⌘	25.322 v3.7.0, CR 149r1
		□ Test specifications		
		□ O&M Specifications		

Other comments: ⌘

How to create CRs using this form:

Comprehensive information and tips about how to create CRs can be found at: http://www.3gpp.org/3G_Specs/CRs.htm. Below is a brief summary:

1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.

2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be downloaded from the 3GPP server under ftp://www.3gpp.org/specs/ For the latest version, look for the directory name with the latest date e.g. 2000-09 contains the specifications resulting from the September 2000 TSG meetings.

3) With "track changes" disabled, paste the entire CR form (use CTRL-A to select it) into the specification just in front of the clause containing the first piece of changed text. Delete those parts of the specification which are not relevant to the change request.

9.6 Protocol Parameters

The values of the protocol parameters in this subclause are signalled by upper layers.

a) MaxDAT.

~~It is the~~The maximum number of transmissions of a PDU is equal to MaxDAT – 1.~~value for the number of retransmissions of a PDU plus one.~~ This parameter is an upper limit of counter VT(DAT). When the value of VT(DAT) comes to MaxDAT, either RLC RESET procedure or SDU discard procedure shall be initiated according to configuration by upper layer.

b) Poll_PDU.

This parameter indicates how often the transmitter should poll the receiver in case of polling every Poll_PDU PDU. This is an upper limit for the VT(PDU) state variable, when VT(PDU) reaches Poll_PDU a poll is transmitted to the peer entity.

c) Poll_SDU.

This parameter indicates how often the transmitter should poll the receiver in case of polling every Poll_SDU SDU. This is an upper limit for the VT(SDU) state variable, when VT(SDU) reaches Poll_SDU a poll is transmitted to the peer entity.

d) Poll_Window.

This parameter indicates when the transmitter should poll the receiver in case of performing window-based polling. The range of values of this parameter shall be $0 \le \text{Poll\_Window} \le 100$. A poll is triggered for each PDU when $J \geq \text{Poll\_Window}$, where J is the window transmission percentage defined by

$$J = \frac{(4096 + VT(S) - VT(A)) \ \mathbf{mod}\ 4096}{VT(WS)} * 100 \ ,$$

where the constant 4096 is the modulus for AM described in Subclause 9.4.

e) MaxRST.

The maximum number of transmissions of a RESET PDU is equal to MaxRST – 1. ~~It is the maximum value for the number of retransmission of RESET PDU plus one.~~ This parameter is an upper limit of counter VT(RST). When the value of VT(RST) comes to MaxRST, unrecoverable error shall be indicated to upper layer.

f) Configured_Tx_Window_Size.

The maximum allowed transmitter window size.

g) Configured_Rx_Window_Size.

The allowed receiver window size.

h) MaxMRW.

The maximum number of transmissions of a MRW command is equal to MaxMRW – 1.~~It is the maximum value for the number of retransmissions of a MRW command plus one.~~ This parameter is an upper limit of counter VT(MRW). When the value of VT(MRW) comes to MaxMRW, RLC RESET procedure shall be initiated.

9.7.3 SDU discard function for acknowledged, unacknowledged, and transparent mode

The SDU discard function allows to discharge RLC PDU from the buffer on the transmitter side, when the transmission of the RLC PDU does not success for a long time. The SDU discard function allows to avoid buffer overflow. There will be several alternative operation modes of the RLC SDU discard function. Upper layers control, which discard function shall be used for each RLC entity.

The following is a list of operation modes for the RLC SDU discard function.

Table 9.2: List of criteria that control when to perform SDU discard

Operation mode	Presence
Timer based discard, with explicit signalling	Network controlled
Timer based discard, without explicit signalling	Network controlled
SDU discard after MaxDAT number of ~~re~~transmissions	Network controlled
No_discard after MaxDAT number of ~~re~~transmissions	Network controlled

9.7.3.3 SDU discard after MaxDAT number of ~~re~~transmissions

This alternative uses the number of ~~re~~transmissions as a trigger for SDU discard, and is therefore only applicable for acknowledged mode RLC. This makes the SDU discard function dependent of the channel rate. Also, this variant of the SDU discard function strives to keep the SDU loss rate constant for the connection, on the cost of a variable delay. SDU discard is triggered at the transmitter, and a MRW command is necessary to convey the discard information to the receiver, like in the timer-based discard with explicit signalling.

9.7.3.4 No_discard after MaxDAT number of ~~re~~transmissions

This alternative uses the number of ~~re~~transmissions, and is therefore only applicable for acknowledged mode RLC. Reset procedure shall be initiated after MaxDAT number of ~~re~~transmissions of an AMD PDU (see subclause 11.3.4.4).

11.3.4 Abnormal cases

11.3.4.1 Timer_Poll timeout

Upon expiry of the Timer_Poll, the sender shall retransmit the poll. The poll can be retransmitted in either a new PDU or a retransmitted PDU.

11.3.4.2 Receiving a PDU outside the receiving window

Upon reception of a PDU with sequence number outside the interval VR(R)≤SN<VR(MR), the receiver shall discard the PDU. The poll bit shall be considered even if a complete PDU is discarded.

11.3.4.3 Timer_Discard timeout

11.3.4.3.1 SDU discard with explicit signalling

Upon expiry of Timer_Discard, the sender shall initiate the SDU discard with explicit signalling procedure. In the case where the TFC selection exchange has been initiated by sending the RLC Entity Info parameter to MAC, the UE may wait until after it provides MAC with the requested set of PDUs before discarding the afore-mentioned SDU.

11.3.4.4 VT(DAT) ≥ MaxDAT

If SDU discard after MaxDAT number of ~~re~~transmission is used and VT(DAT) ≥ MaxDAT for any PDU, the sender shall initiate the SDU discard with explicit signalling procedure. All the SDUs that have segments in PDUs with sequence numbers inside the interval VT(A) ≤ SN ≤ SN of the PDU with VT(DAT) ≥ MaxDAT shall be discarded.

If No_discard after MaxDAT number of ~~re~~transmissions is used, the sender shall initiate the RLC reset procedure when VT(DAT) ≥ MaxDAT.

11.3.4.5 Invalid length indicator value

If the length indicator of a PDU has a value that is larger than the PDU size – RLC header size and is not one of the predefined values listed in the table of subclause 9.2.2.8, the PDU shall be discarded and treated as a missing PDU.

11.3.4.6 Length Indicator value reserved for AMD PDU

Upon reception of an AMD PDU that contains Length Indicator value reserved for AMD PDU, the receiver shall discard that AMD PDU.

11.3.4.7 VT(DAT) ≥ MaxDAT-1

The Sender shall not perform the transmission of the PDU. Instead, it will only increment the corresponding VT(DAT).

11.4.5 Abnormal cases

11.4.5.1 Timer_RST timeout

~~Upon expiry of Timer_RST the sender shall retransmit the RESET PDU and increase VT(RST) with 1. In the retransmitted RESET PDU the value of the RSN field shall not be incremented.~~

If Timer_RST expires before the reset procedure is terminated, the Sender shall:

- if VT(RST)<MaxRST-1:
    - set the RESET PDU as previously transmitted (even if additional SDUs were discarded in the mean-time).
    - transmit RESET PDU;
- increment VT(RST) by one;
- restart Timer_RST.

11.6.1 Purpose

An SDU can be discarded with explicit signalling when MaxDAT number of ~~re~~transmissions is reached or the transmission time exceeds a predefined value (Timer_Discard) for an SDU in acknowledged mode RLC. Move Receiving Window (MRW) command is sent to the receiver so that AMD PDUs carrying that SDU are discarded in the receiver and the receiver window is updated accordingly. Note that when the concatenation function is active, PDUs carrying segments of other SDUs that have not timed out shall not be discarded. If Send MRW is not configured and no segments of an SDU were submitted to a lower layer, the SDU ~~is~~ may ~~simply~~ beis simply discarded in the transmitter without notification to the receiver. If Send MRW is configured, a Move Receiving Window request shall be sent to the receiver even if no segments of the SDU were submitted to a lower layer. The Send MRW is used when the AM RLC entity is connected to a PDCP layer which supports lossless SRNS relocation.

The MRW command is defined as a super-field in the RLC STATUS PDU, and can be piggybacked to status information of transmissions in the opposite direction.

Figure 11.6 below illustrates the elementary procedure for SDU discard with explicit signalling. The sender is the sender of AMD PDUs and it is either the UE or the network and the receiver is the receiver of AMD PDUs and it is either the network or the UE.

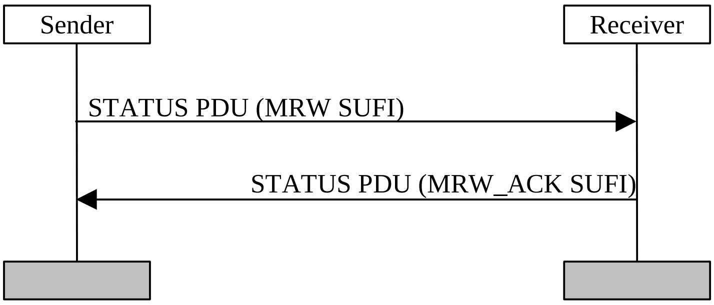


Figure 11.6: SDU discard with explicit signalling

11.6.2 Initiation

This procedure is initiated by the sender when any of the following conditions is fulfilled:

1) Timer based SDU discard with explicit signalling is used, Timer_Discard expires for an SDU, and one or more segments of the SDU have been submitted to a lower layer.

2) Timer based SDU discard with explicit signalling is used, Timer_Discard expires for an SDU, and Send MRW is configured.

3) SDU discard after MaxDAT number of ~~re~~transmissions is used, and MaxDAT number of ~~re~~transmissions is reached for an SDU.

The sender shall discard all PDUs that contain segments of the associated SDUs. If the concatenation function is active, PDUs carrying segments of other SDUs that have not timed out shall not be discarded. VT(A) shall be updated when the procedure is terminated, and VT(S) shall be updated when a new MRW SUFI which includes SN_MRW_{LENGTH} >VT(S) is transmitted.

The sender shall transmit a status report on the DCCH logical channel if the sender is located in the control plane and on the DTCH if it is located in the user plane.

This status report is sent even if the ’STATUS prohibit’ is used and the timer ’Timer_Status_Prohibit’ is active, or if the ’EPC mechanism’ is used and the timer ’Timer_EPC’ is active or ’VR(EP)’ is counting down.

The STATUS PDUs have higher priority than data PDUs.

The sender shall start timer Timer_MRW. If a new SDU discard procedure is triggered when Timer_MRW is running, no new MRW SUFIs shall be sent before the current SDU discard procedure is terminated by one of the termination criteria.

11.6.5 Expiration of timer Timer_MRW

~~If Timer_MRW expires before the discard procedure is terminated, the MRW SUFI shall be retransmitted, VT(MRW) is incremented by one and Timer_MRW restarted. MRW SUFI shall be exactly the same as previously transmitted even though some new SDUs would have been discarded during the running of the Timer_MRW. If the retransmitted STATUS PDU contains other SUFIs than the MRW SUFI, the status information indicated by these SUFIs shall be updated.~~

If Timer_MRW expires before the discard procedure is terminated, the Sender shall:

- if VT(MRW)<MaxMRW-1:
    - set the MRW SUFI as previously transmitted (even if additional SDUs were discarded in the mean-time).
    - include the MRW SUFI in a new status report (if other SUFIs are included, their contents shall be updated);
    - transmit the status report by either including it in a STATUS PDU or piggy-backed in an AMD PDU;
- increment VT(MRW) by one;
- restart Timer_MRW for this discard procedure.