

# ETSI TS 125 322 V3.6.0 (2001-03)

---

*Technical Specification*

## **Universal Mobile Telecommunications System (UMTS); RLC protocol specification (3GPP TS 25.322 version 3.6.0 Release 1999)**

---



---

**Reference**

RTS/TSGR-0225322UR5

---

**Keywords**

UMTS

**ETSI**

---

650 Route des Lucioles  
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C  
Association à but non lucratif enregistrée à la  
Sous-Préfecture de Grasse (06) N° 7803/88

---

**Important notice**

Individual copies of the present document can be downloaded from:

<http://www.etsi.org>

The present document may be made available in more than one electronic version or in print. In any case of existing or perceived difference in contents between such versions, the reference version is the Portable Document Format (PDF). In case of dispute, the reference shall be the printing on ETSI printers of the PDF version kept on a specific network drive within ETSI Secretariat.

Users of the present document should be aware that the document may be subject to revision or change of status. Information on the current status of this and other ETSI documents is available at <http://www.etsi.org/tb/status/>

If you find errors in the present document, send your comment to:  
editor@etsi.fr

---

**Copyright Notification**

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

© European Telecommunications Standards Institute 2001.

All rights reserved.

---

## Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: "*Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards*", which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (<http://www.etsi.org/ipr>).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

---

## Foreword

This Technical Specification (TS) has been produced by the ETSI 3<sup>rd</sup> Generation Partnership Project (3GPP).

The present document may refer to technical specifications or reports using their 3GPP identities, UMTS identities or GSM identities. These should be interpreted as being references to the corresponding ETSI deliverables.

The cross reference between GSM, UMTS, 3GPP and ETSI identities can be found under [www.etsi.org/key](http://www.etsi.org/key).

# Contents

Foreword.....	6
1 Scope.....	7
2 References.....	7
3 Abbreviations.....	7
4 General.....	8
4.2 Overview on sublayer architecture.....	8
4.2.1 Model of RLC.....	8
4.2.1.1 Transparent mode entities.....	10
4.2.1.2 Unacknowledged mode entities.....	10
4.2.1.3 Acknowledged mode entity.....	11
5 Functions.....	13
6 Services provided to upper layers.....	13
6.1 Mapping of services/functions onto logical channels.....	14
7 Services expected from MAC.....	16
8 Elements for layer-to-layer communication.....	16
8.1 Primitives between RLC and higher layers.....	16
8.2 Primitive parameters.....	17
9 Elements for peer-to-peer communication.....	18
9.1 Protocol data units.....	18
9.1.1 Data PDUs.....	18
9.1.2 Control PDUs.....	18
9.2 Formats and parameters.....	19
9.2.1 Formats.....	19
9.2.1.1 General.....	19
9.2.1.2 TrD PDU.....	19
9.2.1.3 UMD PDU.....	19
9.2.1.4 AMD PDU.....	20
9.2.1.5 STATUS PDU.....	20
9.2.1.6 Piggybacked STATUS PDU.....	21
9.2.1.7 RESET, RESET ACK PDU.....	21
9.2.2 Parameters.....	22
9.2.2.1 D/C field.....	22
9.2.2.2 PDU Type.....	22
9.2.2.3 Sequence Number (SN).....	22
9.2.2.4 Polling bit (P).....	23
9.2.2.5 Extension bit (E).....	23
9.2.2.6 Reserved 1 (R1).....	23
9.2.2.7 Header Extension Type (HE).....	23
9.2.2.8 Length Indicator (LI).....	23
9.2.2.9 Data field.....	25
9.2.2.10 Padding (PAD).....	25
9.2.2.11 SUFI.....	25
9.2.2.11.1 The No More Data super-field.....	26
9.2.2.11.2 The Acknowledgement super-field.....	26
9.2.2.11.3 The Window Size super-field.....	27
9.2.2.11.4 The List super-field.....	27
9.2.2.11.5 The Bitmap super-field.....	28
9.2.2.11.6 The Relative List super-field.....	28
9.2.2.11.7 The Move Receiving Window Acknowledgement super-field.....	29
9.2.2.11.8 The Move Receiving Window (MRW) super-field.....	29
9.2.2.12 Reserved 2 (R2).....	30

9.2.2.13	Reset Sequence Number (RSN) .....	30
9.2.2.14	Hyper Frame Number Indicator (HFNI) .....	30
9.3	Protocol states.....	31
9.3.1	State model for transparent mode entities.....	31
9.3.1.1	Null State .....	31
9.3.1.2	Transparent Data Transfer Ready State .....	31
9.3.2	State model for unacknowledged mode entities.....	31
9.3.2.1	Null State .....	31
9.3.2.2	Unacknowledged Data Transfer Ready State .....	31
9.3.2.3	Local Suspend State.....	31
9.3.3	State model for acknowledged mode entities.....	32
9.3.3.1	Null State .....	32
9.3.3.2	Acknowledged Data Transfer Ready State .....	32
9.3.3.3	Reset Pending State.....	32
9.3.3.4	Local Suspend State.....	33
9.4	State variables.....	34
9.5	Timers.....	36
9.6	Protocol Parameters.....	37
9.7	Specific functions.....	38
9.7.1	Polling function for acknowledged mode .....	38
9.7.2	STATUS transmission for acknowledged mode.....	38
9.7.3	SDU discard function for acknowledged, unacknowledged, and transparent mode .....	39
9.7.3.1	Timer based discard, with explicit signalling .....	40
9.7.3.2	Timer based discard, without explicit signalling .....	40
9.7.3.3	SDU discard after MaxDAT number of retransmissions .....	40
9.7.3.4	No_discard after MaxDAT number of retransmissions .....	40
9.7.4	The Estimated PDU Counter for acknowledged mode .....	40
9.7.5	Local Suspend function for acknowledged and unacknowledged mode.....	41
9.7.6	RLC stop, RLC Continue function .....	41
10	Handling of unknown, unforeseen and erroneous protocol data .....	41
11	Elementary procedures .....	42
11.1	Transparent mode data transfer procedure.....	42
11.1.1	Purpose .....	42
11.1.2	Initiation.....	42
11.1.2.1	TrD PDU contents to set .....	42
11.1.3	Reception of TrD PDU .....	42
11.1.4	Abnormal cases.....	42
11.1.4.1	Undefined SDU size at receiver .....	42
11.1.4.2	SDU discard without explicit signalling.....	43
11.2	Unacknowledged mode data transfer procedure.....	43
11.2.1	Purpose .....	43
11.2.2	Initiation.....	43
11.2.2.1	UMD PDU contents to set.....	43
11.2.3	Reception of UMD PDU.....	43
11.2.4	Abnormal cases.....	44
11.2.4.1	Length Indicator value reserved for UMD PDU .....	44
11.2.4.2	Invalid length indicator value.....	44
11.2.4.3	SDU discard without explicit signalling.....	44
11.3	Acknowledged mode data transfer procedure.....	44
11.3.1	Purpose .....	44
11.3.2	Initiation.....	44
11.3.2.1	AMD PDU contents to set.....	45
11.3.2.1.1	Setting of the Polling bit.....	45
11.3.2.1.2	Segmentation and concatenation of SDUs .....	45
11.3.3	Reception of AMD PDU by the receiver .....	46
11.3.4	Abnormal cases.....	46
11.3.4.1	Timer_Poll timeout .....	46
11.3.4.2	Receiving a PDU outside the receiving window .....	46
11.3.4.3	Timer_Discard timeout .....	46
11.3.4.3.1	SDU discard with explicit signalling.....	46

11.3.4.4	VT(DAT) $\geq$ MaxDAT .....	46
11.3.4.5	Invalid length indicator value.....	46
11.3.4.6	Length Indicator value reserved for AMD PDU .....	46
11.4	RLC reset procedure .....	47
11.4.1	Purpose .....	47
11.4.2	Initiation.....	47
11.4.2.1	RESET PDU contents to set.....	47
11.4.3	Reception of the RESET PDU by the receiver .....	47
11.4.3.1	RESET ACK PDU contents to set .....	48
11.4.4	Reception of the RESET ACK PDU by the sender .....	48
11.4.5	Abnormal cases.....	48
11.4.5.1	Timer_RST timeout .....	48
11.4.5.2	Unrecoverable error (VT(RST) $\geq$ MaxRST).....	48
11.4.5.3	Reception of the RESET PDU by the sender .....	48
11.5	STATUS report transfer procedure.....	48
11.5.1	Purpose .....	48
11.5.2	Initiation.....	49
11.5.2.1	Piggybacked STATUS PDU .....	49
11.5.2.2	STATUS PDU contents to set .....	49
11.5.3	Reception of the STATUS PDU by the sender .....	50
11.5.4	Abnormal cases.....	50
11.5.4.1	VR(EP) reaches zero and the requested PDUs have not been received .....	50
11.6	SDU discard with explicit signalling procedure .....	50
11.6.1	Purpose .....	50
11.6.2	Initiation.....	51
11.6.2.1	Piggybacked STATUS PDU .....	51
11.6.2.2	STATUS PDU contents to set .....	51
11.6.3	Reception of the STATUS PDU by the receiver .....	51
11.6.4	Termination.....	52
11.6.5	Expiration of timer Timer_MRW .....	52
11.6.6	Abnormal cases.....	52
11.6.6.1	Obsolete/corrupted MRW command.....	52
11.6.6.2	VT(MRW) equals MaxMRW .....	52
11.6.6.3	Reception of obsolete MRW_ACK.....	52
11.7	Ciphering .....	53
11.8	RLC re-establishment procedure .....	54
<b>Annex A (informative): Change history.....</b>		<b>55</b>

---

# Foreword

This Technical Specification (TS) has been produced by the 3<sup>rd</sup> 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 RLC protocol.

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".

---

# 3 Abbreviations

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

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
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
PCCCH	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
TTI	Transmission Time Interval
U-	User-
UE	User Equipment
UL	Uplink
UMTS	Universal Mobile Telecommunications System
UTRA	UMTS Terrestrial Radio Access
UTRAN	UMTS Terrestrial Radio Access Network

---

## 4 General

### 4.2 Overview on sublayer architecture

The model presented in this subclause is not for implementation purposes.

#### 4.2.1 Model of RLC

Figure 4.1 gives an overview model of the RLC layer. The figure illustrates the different RLC peer entities. There is one transmitting and one receiving entity for the transparent mode service and the unacknowledged mode service and one combined transmitting and receiving entity for the acknowledged mode service. In this specification the word transmitted is equivalent to "submitted to lower layer" unless otherwise explicitly stated. The dashed lines between the AM-Entities illustrate the possibility to send the RLC PDUs on separate logical channels, e.g. control PDUs on one and data PDUs on the other. More detailed descriptions of the different entities are given in subclauses 4.2.1.1, 4.2.1.2 and 4.2.1.3.

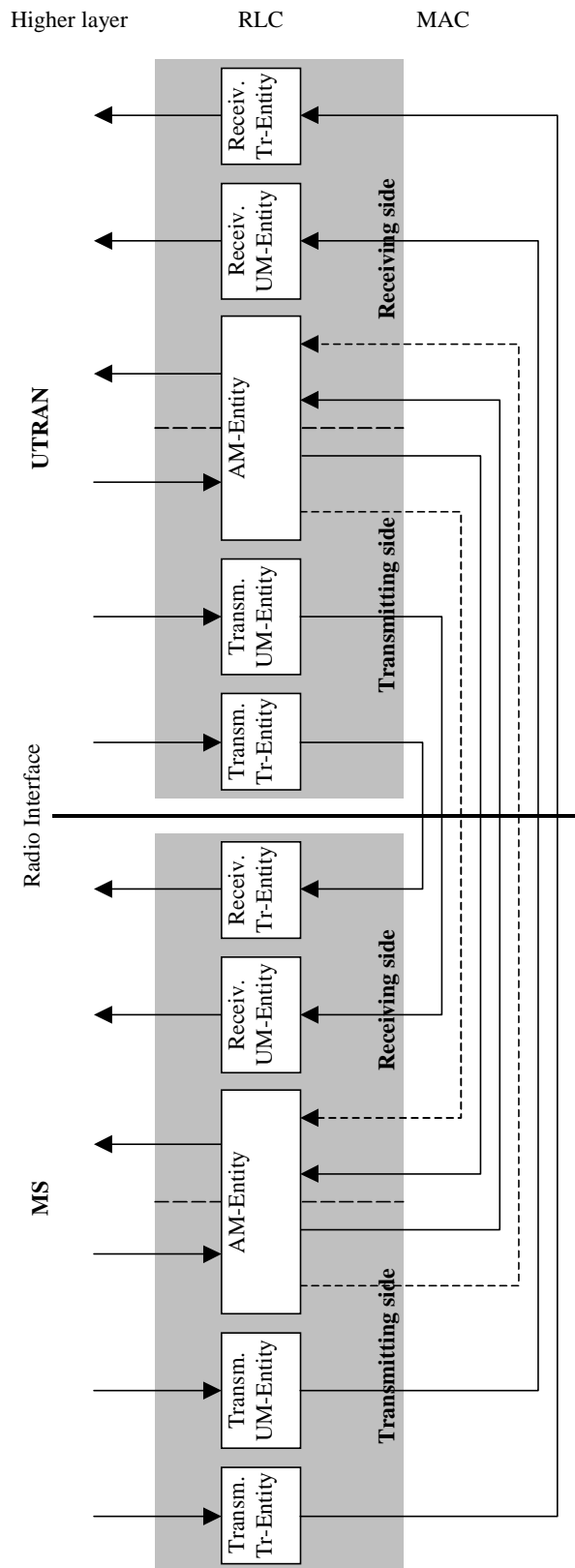
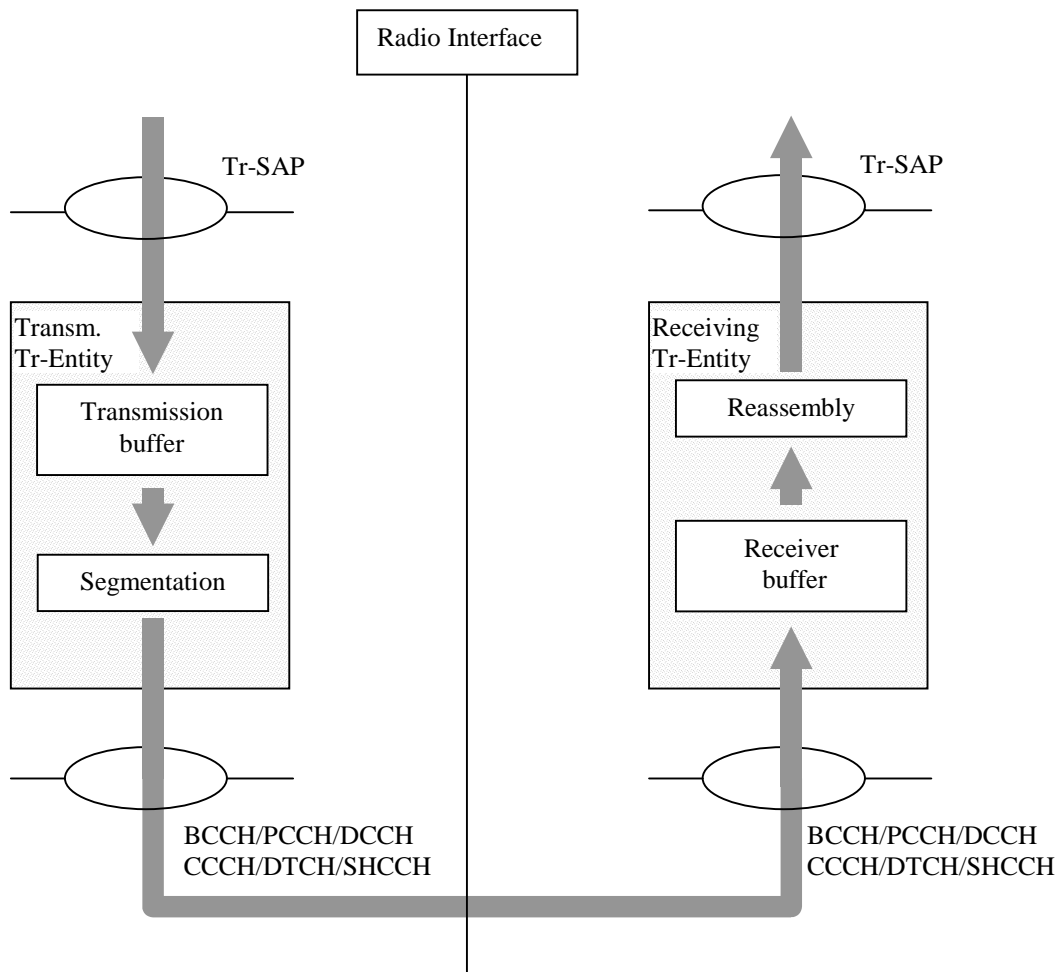


Figure 4.1: Overview model of RLC

#### 4.2.1.1 Transparent mode entities

Figure 4.2 below shows the model of two transparent mode peer entities.



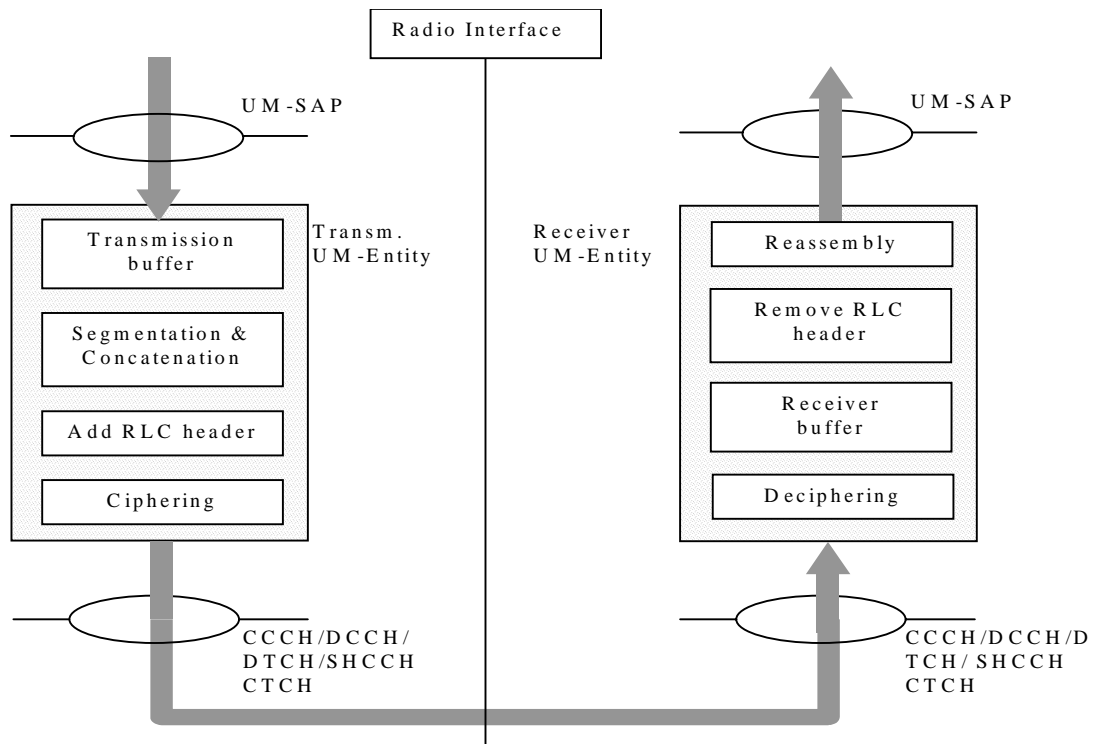
**Figure 4.2: Model of two transparent mode peer entities**

The transmitting Tr-entity receives SDUs from the higher layers through the Tr-SAP. RLC might segment the SDUs into appropriate RLC PDUs without adding any overhead. 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. The CCCH and SHCCH uses transparent mode only for the uplink. Which type of logical channel depends on if the higher layer is located in the control plane (BCCH, DCCH, PCCH, CCCH, SHCCH) or user plane (DTCH).

The receiving Tr-entity receives PDUs through one of the logical channels from lower layer. RLC reassembles (if segmentation has been performed) the PDUs into RLC SDUs. How to perform the reassembling is decided upon when the service is established. RLC delivers the RLC SDUs to the higher layer through the Tr-SAP.

#### 4.2.1.2 Unacknowledged mode entities

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



**Figure 4.3: Model of two unacknowledged mode peer entities**

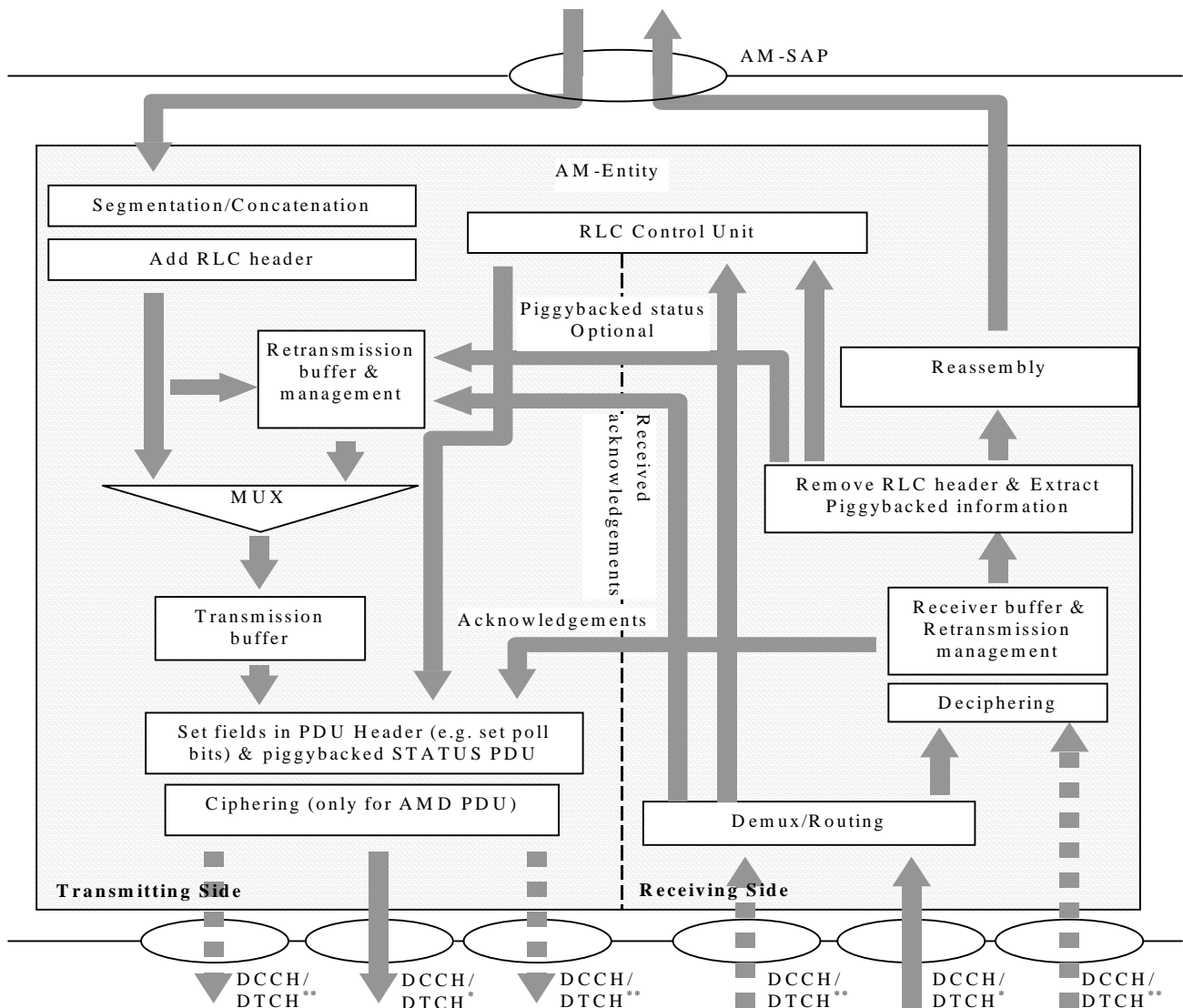
The transmitting UM-entity receives SDUs from the higher layers. RLC might segment the SDUs into RLC PDUs of appropriate size. The SDU might also be concatenated with other SDUs. RLC delivers the RLC PDUs to lower layer through either a CCCH, SHCCH, DCCH, CTCH or a DTCH. The CCCH and SHCCH uses unacknowledged mode only for the downlink. Which type of logical channel depends on if the higher layer is located in the control plane (CCCH, DCCH, SHCCH) or user plane (CTCH, DTCH).

The receiving UM-entity receives PDUs through one of the logical channels from the MAC sublayer. RLC removes header from the PDUs and reassembles the PDUs (if segmentation has been performed) into RLC SDUs. The RLC SDUs are delivered to the higher layer.

#### 4.2.1.3 Acknowledged mode entity

Figure 4.4 below shows the model of an acknowledged mode entity, when one logical channel (shown as a solid line) and when two logical channels (shown as dashed lines) are used.

In case two logical channels are used in the uplink the first logical channel shall be used for data PDUs and the second logical channel shall be used for control PDUs. In case one logical channel is used, the RLC PDU size shall be the same for AMD PDUs and control PDUs.



**Figure 4.4: Model of an acknowledged mode entity**

The transmitting side of the AM-entity receives SDUs from the higher layers. The SDUs are segmented and/or concatenated to PDUs of fixed length. PDU length is a semi-static value that is decided in bearer setup and can only be changed through bearer reconfiguration by RRC.

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 PDUs are placed in the retransmission buffer and the transmission buffer.

The MUX then decides which PDUs and when the PDUs are submitted to the lower layer. The PDUs are submitted via 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.

The ciphering is applied only for AMD PDUs. The fixed 2 octets AMD PDU header is not ciphered. Piggybacked STATUS PDU and Padding parts of AMD PDU when existing are ciphered. The other Control PDUs (i.e. STATUS PDU, RESET PDU, and RESET ACK PDU) shall not be ciphered.

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.

The receiving side of the AM-entity receives PDUs through one of the logical channels from lower layer. 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 higher 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. For a detailed description of the following functions see [3]:

- Segmentation and reassembly.
- Concatenation.
- Padding.
- Transfer of user data.
- Error correction.
- In-sequence delivery of higher layer PDUs.
- Duplicate detection.
- Flow control.
- Sequence number check.
- Protocol error detection and recovery.
- Ciphering.
- Suspend/resume function.

---

## 6 Services provided to upper layers

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

- **Transparent data transfer Service:**

The following functions are needed to support transparent data transfer:

- Segmentation and reassembly.
- Transfer of user data.

- **Unacknowledged data transfer Service:**

The following functions are needed to support unacknowledged data transfer:

- Segmentation and reassembly.
- Concatenation.
- Padding.

- Transfer of user data.
- Ciphering.
- Sequence number check.
- **Acknowledged data transfer Service:**

The following functions are needed to support acknowledged data transfer:

- Segmentation and reassembly.
- Concatenation.
- Padding.
- Transfer of user data.
- Error correction.
- In-sequence delivery of higher layer PDUs.
- Duplicate detection.
- Flow Control.
- Protocol error detection and recovery.
- Ciphering.
- **QoS setting.**
- **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
<b>Transparent Service</b>	Applicability	+	+	+	+
	Segmentation	-	-	+	+
	Transfer of user data	+	+	+	+
<b>Unacknowledged Service</b>	Applicability	-	-	+	+
	Segmentation	-	-	+	+
	Concatenation	-	-	+	+
	Padding	-	-	+	+
	Transfer of user data	-	-	+	+
	Ciphering	-	-	+	+
<b>Acknowledged Service</b>	Applicability	-	-	+	+
	Segmentation	-	-	+	+
	Concatenation	-	-	+	+
	Padding	-	-	+	+
	Transfer of user data	-	-	+	+
	Flow Control	-	-	+	+
	Error Correction	-	-	+	+
	Protocol error correction & recovery	-	-	+	+
Ciphering	-	-	+	+	

Table 6.2: RLC modes and functions in UE downlink side

Service	Functions	BCCH	PCCH	SHCCH	CCCH	DCCH	DTCH	CTCH
Transparent Service	Applicability	+	+	-	-	+	+	-
	Reassembly	-	-	-	-	+	+	-
Unacknowledged Service	Applicability	-	-	+	+	+	+	+
	Reassembly	-	-	+	+	+	+	+
	Deciphering	-	-	-	-	+	+	-
	Sequence number check	-	-	+	+	+	+	+
Acknowledged Service	Applicability	-	-	-	-	+	+	-
	Reassembly	-	-	-	-	+	+	-
	Error correction	-	-	-	-	+	+	-
	Flow Control	-	-	-	-	+	+	-
	In sequence delivery	-	-	-	-	+	+	-
	Duplicate detection	-	-	-	-	+	+	-
	Protocol error correction & recovery	-	-	-	-	+	+	-
	Deciphering	-	-	-	-	+	+	-

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	+	+	-	-	+	+	-
Unacknowledged Service	Applicability	-	-	+	+	+	+	+
	Segmentation	-	-	+	+	+	+	+
	Concatenation	-	-	+	+	+	+	+
	Padding	-	-	+	+	+	+	+
	Ciphering	-	-	-	-	+	+	-
	Transfer of user data	-	-	+	+	+	+	+
Acknowledged Service	Applicability	-	-	-	-	+	+	-
	Segmentation	-	-	-	-	+	+	-
	Concatenation	-	-	-	-	+	+	-
	Padding	-	-	-	-	+	+	-
	Transfer of user data	-	-	-	-	+	+	-
	Flow Control	-	-	-	-	+	+	-
	Error Correction	-	-	-	-	+	+	-
	Protocol error correction & recovery	-	-	-	-	+	+	-
Ciphering	-	-	-	-	+	+	-	

Table 6.4: RLC modes and functions in UTRAN uplink side

Service	Functions	CCCH	SHCCH	DCCH	DTCH
Transparent Service	Applicability	+	+	+	+
	Reassembly	-	-	+	+
Unacknowledged Service	Applicability	-	-	+	+
	Reassembly	-	-	+	+
	Deciphering	-	-	+	+
	Sequence number check	-	-	+	+
Acknowledged Service	Applicability	-	-	+	+
	Reassembly	-	-	+	+
	Error correction	-	-	+	+
	Flow Control	-	-	+	+
	In sequence delivery	-	-	+	+
	Duplicate detection	-	-	+	+
	Protocol error correction & recovery	-	-	+	+
	Deciphering	-	-	+	+



## 7 Services expected from MAC

For a detailed description of the following functions see [3].

- Data transfer.

## 8 Elements for layer-to-layer communication

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

### 8.1 Primitives between RLC and higher 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	Parameter			
	Req.	Ind.	Resp.	Conf.
<b>RLC-AM-DATA</b>	Data, CNF, MUI	Data, DiscardInfo	Not Defined	MUI
<b>RLC-UM-DATA</b>	Data, Use special LI	Data	Not Defined	Not Defined
<b>RLC-TR-DATA</b>	Data	Data	Not Defined	Not Defined
<b>CRLC-CONFIG</b>	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
<b>CRLC-SUSPEND (UM/AM only)</b>	N	Not Defined	Not Defined	VT(US) (UM only), VT(S) (AM only)
<b>CRLC-RESUME (UM/AM only)</b>	No Parameter	Not Defined	Not Defined	Not Defined
<b>CRLC-STATUS</b>	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 higher layers to request transmission of a higher layer PDU in acknowledged mode.
- RLC-AM-DATA-Ind is used by RLC to deliver to higher layers RLC SDUs that have been transmitted in acknowledged mode and to indicate higher layers of the discarded RLC SDU in the peer RLC AM entity.
- RLC-AM-DATA-Conf is used by RLC to confirm to higher layers reception of an RLC SDU by the peer-RLC AM entity .

#### **RLC-UM-DATA-Req/Ind**

- RLC-UM-DATA-Req is used by higher layers to request transmission of a higher layer PDU in unacknowledged mode.
- RLC-UM-DATA-Ind is used by RLC to deliver to higher layers RLC SDUs that have been transmitted in unacknowledged mode.

#### **RLC-TR-DATA-Req/Ind**

- RLC-TR-DATA-Req is used by higher layers to request transmission of a higher layer PDU in transparent mode.

- RLC-TR-DATA-Ind is used by RLC to deliver to higher layers RLC SDUs that have been transmitted in transparent mode.

### **CRLC-CONFIG-Req**

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

### **CRLC-SUSPEND-Req/Conf**

This primitive is used by RRC to suspend the RLC. The N parameter indicates that RLC shall not send a PDU with  $SN \geq VT(S) + N$  for AM and  $SN \geq VT(US) + N$  for UM, where N is an integer. RLC informs RRC of the VT(S) for AM and VT(US) for UM in the confirm primitive.

### **CRLC-RESUME-Req**

This primitive is used by RRC to resume RLC when RLC has been suspended.

### **CRLC-STATUS-Ind**

It is used by the RLC to send status information to RRC.

## **8.2 Primitive parameters**

Following parameters are used in the primitives:

- 1) The parameter Data is the RLC SDU that is mapped onto the Data field in RLC PDUs. The Data parameter may be divided over several RLC PDUs. In case of an RLC-AM-DATA or an RLC-UM-DATA primitive the length of the Data parameter shall be octet-aligned.
- 2) The parameter Confirmation request (CNF) indicates whether the RLC needs to confirm the reception of the RLC SDU by the peer-RLC AM entity.
- 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 conf. primitive.
- 4) The parameter E/R indicates (re)establishment, release or modification of RLC. If it indicates (re-)establishment, the state variables and configurable parameters are initialised according to subclause 11.8. If it indicates release, all protocol parameters, variables and timers shall be released and RLC shall exit the data transfer ready state. If it indicates modification, the protocol parameters indicated by RRC (e.g. ciphering parameters) shall only be modified with keeping the other protocol parameters, the protocol variables, the protocol timers and the protocol state. RLC shall always be re-established if the PDU size is changed.
- 5) The parameter Event Code (EVC) indicates the reason for the CRLC-STATUS-ind (e.g., unrecoverable errors such as data link layer loss or recoverable status events such as reset.).
- 6) The parameter ciphering elements are only applicable for UM and AM operation. These parameters are Ciphering Mode, Ciphering Key, Transmitting Activation Time (SN to activate a new ciphering configuration at the transmitter), Receiving Activation Time (SN to activate a new ciphering configuration at the receiver) and HFN (Hyper Frame Number).
- 7) The AM\_parameters are only applicable for AM operation. It contains PDU size, In-sequence Delivery Indication (indicating that SDUs shall be 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 shall always be greater than or equal to the number of transport blocks in the smallest transport block set. The Send MRW indicates that the MRW SUFI shall be sent to the receiver even if no segments of the expired SDU were submitted to a lower layer.
- 8) The parameter DiscardInfo indicates to upper layer the discarded RLC SDU in the peer-RLC AM entity. It is applicable only when in-sequence delivery is active and it is purposed to be used when the upper layer requires the reliable data transfer and especially the information of the discarded RLC SDU.

- 9) The Stop parameter indicates that the RLC entity shall not transmit or receive RLC PDUs. The Continue parameter indicates that the RLC entity shall continue transmission and reception of RLC PDUs.
- 10) The parameter Use special LI indicates that the LI indicating that an RLC SDU begins in the beginning of an RLC PDU (the first data octet of the PDU is the first octet of an SDU) shall be used. 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 LI shall not be used.
- 11) The UM\_parameters are only applicable for UM operation. It contains Timer\_Discard value (see subclause 9.5).
- 12) The TM\_parameters are only applicable for TM operation. It contains Segmentation indication (see subclauses 9.2.2.9 and 11.1.2.1) and Timer\_Discard value (see subclause 9.5).

---

## 9 Elements for peer-to-peer communication

### 9.1 Protocol data units

#### 9.1.1 Data PDUs

- a) TrD PDU (Transparent Mode Data PDU).

The TrD PDU is used to convey RLC SDU data without adding any RLC overhead. The TrD 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 entity to inform the transmitting entity about missing PDUs at the receiving entity;
- by the receiving entity to inform the transmitting entity about the size of the allowed transmission window;
- and by the transmitting entity to request the receiving entity to move the receiving window.

- 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
<b>Transparent</b>	TrD	Transparent mode data
<b>Unacknowledged</b>	UMD	Sequenced unacknowledged mode data
<b>Acknowledged</b>	AMD	Sequenced acknowledged mode data
	STATUS	Solicited or Unsolicited Status Report
	Piggybacked STATUS	Piggybacked Solicited or Unsolicited Status Report
	RESET	Reset Command
	RESET ACK	Reset Acknowledgement

## 9.2 Formats and parameters

### 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.

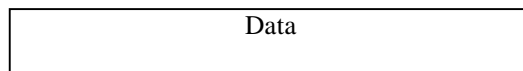
#### 9.2.1.1 General

An RLC PDU is a bit string, with a length not necessarily a multiple of 8 bits. In the drawings in clause 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 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 an integer number of octets in length. An SDU is included into an RLC PDU from first bit onward.

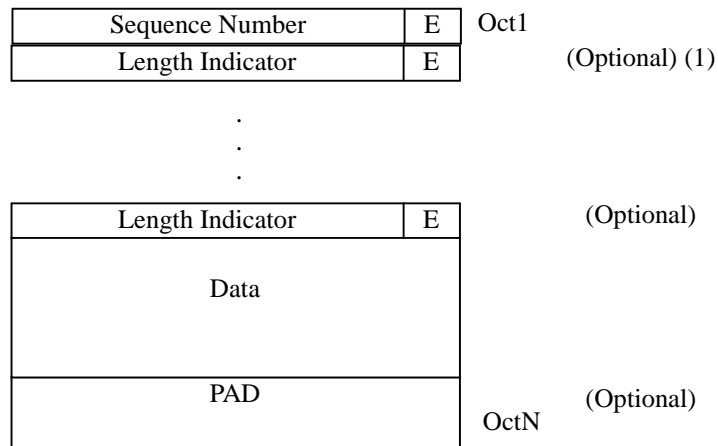
#### 9.2.1.2 TrD PDU

The TrD PDU transfers 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 an integer number of octets.

**Figure 9.1: TrD PDU**

#### 9.2.1.3 UMD PDU

The UMD PDU transfers user data when RLC is operating in unacknowledged mode. The length of the data part shall be an integer number of octets. The UMD PDU header consists of the first octet, which contains the sequence number. The RLC header consists of the first octet and all the octets that contain length indicators.

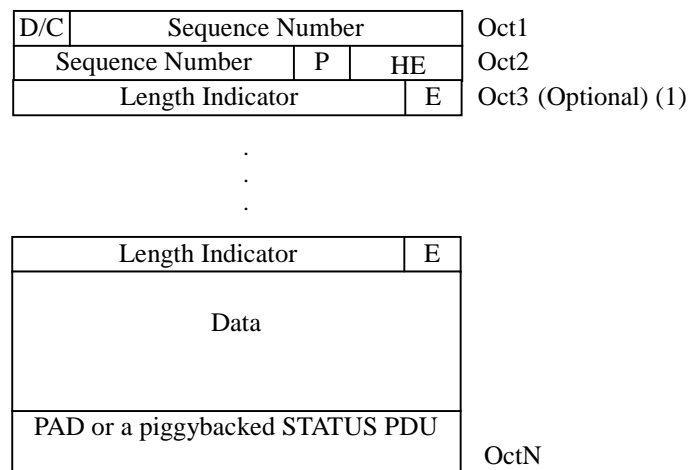


**Figure 9.2: UMD PDU**

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

### 9.2.1.4 AMD PDU

The AMD PDU transfers user data and piggybacked status information and requests status report by setting Poll bit when RLC is operating in acknowledged mode. The length of the data part shall be an integer number of octets. The AMD PDU header consists of the first two octets, which contain the sequence number. The RLC header consists of the first two octets and all the octets that contain length indicators.



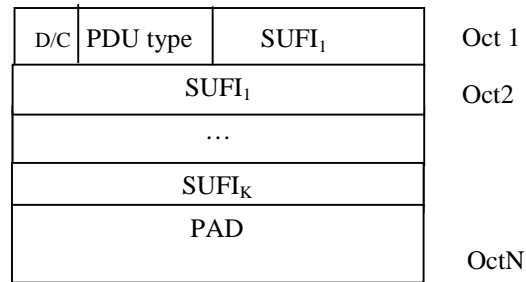
NOTE (1): The Length Indicator may be 15 bits.

**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 and the length of each SUFI is dependent on the SUFI type.

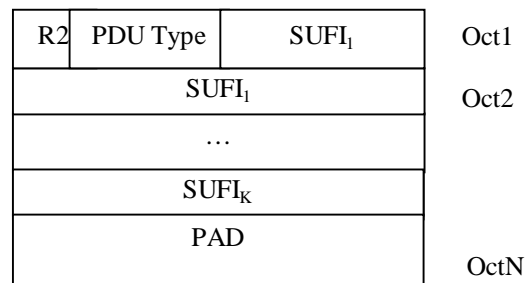


**Figure 9.4: Status Information Control PDU (STATUS PDU)**

Up to K super-fields (SUFI<sub>1</sub>-SUFI<sub>k</sub>) 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 an integer number of octets.

#### 9.2.1.6 Piggybacked STATUS PDU

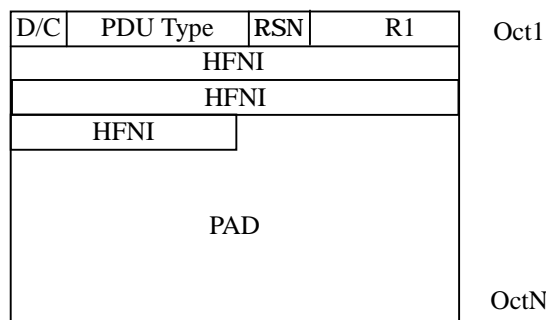
The format of the piggybacked STATUS PDU is the same as the ordinary Control PDU except that the D/C field is replaced by a reserved bit (R2). This PDU can be used to piggyback STATUS PDU in an AMD PDU if the data does not fill the complete AMD PDU. The PDU Type field is set to zero and all other values 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 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 whether it is a retransmission of a previous RESET 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 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 PDU. It can be either data or control PDU.

Bit	Description
0	Control PDU
1	Acknowledged mode data 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 sequence number 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.

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 length indicator and E bit.

Bit	Description
0	The next field is data
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 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 length indicator 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 SDU occurs in the PDU. The Length Indicator points out the number of octets between the end of the last Length Indicator field and up to and including the octet at the end of an SDU segment. Length Indicators are included in the PDUs that they refer to. The size of the Length Indicator may be either 7bits or 15bits. The value of a 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. Length Indicators that are part of a Length Indicator group must never be reordered within the Length Indicator group or removed from the Length Indicator group.

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.



In the case where the end of the last segment of an SDU exactly ends at the end of a PDU and there is no LI that indicates the end of the SDU, the next Length Indicator, shall be placed as the first Length Indicator in the following PDU and have value LI=0.

In the case where a PDU contains a 15-bit LI indicating that an SDU ends with one octet left in the PDU, the last octet of this PDU shall be ignored and shall not be filled with the first octet of the next SDU data.

In the case where the last segment of an RLC SDU is one octet short of exactly filling the previous RLC PDU, and 15-bit Length Indicators are used, the Length Indicator shall be placed as the first Length Indicator in the following PDU and have value LI=111 1111 1111 1011. The remaining one octet in the previous RLC PDU shall be ignored.

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 padding Length Indicator must be placed after any 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 must be used to indicate the piggybacked STATUS PDU. This 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 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 that indicates the end of the last SDU segment in the PDU.

If SDU discard with explicit signalling is used an AMD PDU can contain a maximum number of 15 LIs indicating the end of an SDU and the rest of the AMD PDU space shall be used as padding/piggybacked STATUS PDU.

For AM, 7bit indicators shall be used if the AMD PDU size is  $\leq 126$  octets. Otherwise 15bit indicators shall be used. For UM, 7bit indicators shall be used if the UMD PDU size is  $\leq 125$  octets. Otherwise 15bit indicators shall be used.

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

If the maximum RLC PDU size for an RLC entity is not explicitly configured (e.g. on FACH), the length of the Length Indicator is determined by the maximum configured TB size for the transport channel on which the logical channel is mapped.

Length: 7bits

Bit	Description
0000000	The previous RLC PDU was exactly filled with the last segment of an RLC SDU and there is no LI that indicates the end of the SDU in the previous RLC PDU.
1111100	UMD PDU: The first data octet in this RLC PDU is the first octet of an 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: 15bits

Bit	Description
0000000000000000	The previous RLC PDU was exactly filled with the last segment of an RLC SDU and there is no LI that indicates the end of the SDU in the previous RLC PDU.
111111111111011	The last segment of an RLC SDU was one octet short of exactly filling the previous RLC PDU and there is no LI that indicates the end of the 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 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 mode.

Transparent mode data:

The length of RLC SDUs is not constrained to a multiple of 8 bits.

The RLC SDUs might be segmented. The allowed size for the segments shall be determined from the transport formats of the transport channel [4, 8]. All the RLC PDUs carrying one RLC SDU shall be sent in one transmission time interval. Only segments from one RLC SDU shall be sent in one transmission time interval.

**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:

The length of RLC SDUs is constrained to a multiple of 8 bits.

RLC SDUs might be segmented. If possible, the last segment of an SDU shall be concatenated with the first segment of the next SDU in order to fill the data field completely and avoid unnecessary padding. The length indicator field is used to point the borders between SDUs.

For PDUs with 15-bit LIs, if an SDU ends with one octet left in a PDU whether the LI indicating the end of the 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 shall not be concatenated in this PDU. No LI shall be needed to indicate this kind of one-octet padding.

### 9.2.2.10 Padding (PAD)

Padding has a length such that the PDU has the required predefined total length.

Padding may have any value and the receiving entity 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  $SN \geq VR(H)$  i.e. PDUs that have not yet reached the 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 9.2.2.11.5.

Length: variable number of bits.

The SUFI (Super-Field) 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 ( <b>NO_MORE</b> )
0001	Window Size ( <b>WINDOW</b> )
0010	Acknowledgement ( <b>ACK</b> )
0011	List ( <b>LIST</b> )
0100	Bitmap ( <b>BITMAP</b> )
0101	Relative list ( <b>Rlist</b> )
0110	Move Receiving Window ( <b>MRW</b> )
0111	Move Receiving Window Acknowledgement ( <b>MRW_ACK</b> )
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 sequence numbers < 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 different value than VR(R) all erroneous PDUs must 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 not be set to a value > VR(H).

### 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 is always allowed to change the Tx window size of the peer entity during a connection, but the minimum and the maximum allowed value is given by RRC configuration. The Rx window of the receiver is not changed.

Type = <b>WINDOW</b>
WSN

Figure 9.10: The WINDOW fields in a STATUS PDU

#### WSN

Length: 12 bits

The value of VT(W) to be used by the transmitter. The range of the WSN is  $[0, 2^{12}-1]$ . The minimum value of VT(W) is 1, if WSN is zero the SUFI shall be discarded by this version of the protocol. The variable VT(W) is set equal to WSN upon reception of this SUFI. If WSN is greater than Configured\_Tx\_Window\_Size, VT(W) 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 = <b>LIST</b>
LENGTH
SN <sub>1</sub>
L <sub>1</sub>
SN <sub>2</sub>
L <sub>2</sub>
...
SN <sub>LENGTH</sub>
L <sub>LENGTH</sub>

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

#### LENGTH

Length: 4 bits

The number of (SN<sub>*i*</sub>, L<sub>*i*</sub>)-pairs in the super-field of type LIST. The value "0000" is invalid and the list is discarded.

#### SN<sub>*i*</sub>

Length: 12 bits

Sequence number of PDU, which was not correctly received.

#### L<sub>*i*</sub>

Length: 4 bits

Number of consecutive PDUs not correctly received following PDU with sequence number SN<sub>*i*</sub>.

### 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 = <b>BITMAP</b>
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 sequence number for the first bit in the bitmap. FSN shall not be set to a value lower than VR(R)-7 when the Rx window size is less than half the maximum RLC AM sequence number. If the Rx 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 ∈ [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 = <b>RLIST</b>
LENGTH
FSN
CW <sub>1</sub>
CW <sub>2</sub>
...
CW <sub>LENGTH</sub>

**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 sequence 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.

### 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 = <b>MRW_ACK</b>
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 receiver to move its receiving window and to indicate the amount of discarded SDUs, as a result of an SDU discard in the RLC transmitter. The format is given in Figure 9.15 below.

Type = <b>MRW</b>
LENGTH
SN_MRW <sub>1</sub>
SN_MRW <sub>2</sub>
...
SN_MRW <sub>LENGTH</sub>
N <sub>LENGTH</sub>

**Figure 9.15: The MRW fields in a STATUS PDU**

### LENGTH

Length: 4 bits

The number of SN\_MRW<sub>i</sub> fields in the super-field of type MRW. The values "0001" through "1111" indicate 1 through 15 SN\_MRW<sub>i</sub> respectively. The value "0000" indicates that one SN\_MRW<sub>i</sub> field is present and that the discarded SDU extends above the configured Tx window in the transmitter.

### SN\_MRW<sub>i</sub>

Length: 12 bits

SN\_MRW<sub>i</sub> is used to indicate the end of each discarded SDU. SN\_MRW<sub>i</sub> is the sequence number of the PDU that contains the LI of the i:th discarded SDU (except when N<sub>LENGTH</sub> = 0, see definition of N<sub>LENGTH</sub>). The order of the SN\_MRW<sub>i</sub> shall be in the same sequential order as the SDUs that they refer to.

Additionally SN\_MRW<sub>LENGTH</sub> requests the RLC receiver to discard all PDUs with sequence number < SN\_MRW<sub>LENGTH</sub>, and to move the receiving window accordingly. In addition, the receiver has to discard the first N<sub>LENGTH</sub> LIs and the corresponding data octets in the PDU with sequence number SN\_MRW<sub>LENGTH</sub>.

### N<sub>LENGTH</sub>

Length: 4 bits

N<sub>LENGTH</sub> is used together with SN\_MRW<sub>LENGTH</sub> to indicate the end of the last discarded SDU.

N<sub>LENGTH</sub> indicates which LI in the PDU with sequence number SN\_MRW<sub>LENGTH</sub> corresponds to the last discarded SDU. N<sub>LENGTH</sub> = 0 indicates that the last SDU ended in the PDU with sequence number SN\_MRW<sub>LENGTH</sub> - 1 and that the first data octet in the PDU with sequence number SN\_MRW<sub>LENGTH</sub> 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 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 value as the original RESET PDU. Otherwise it will have the next reset sequence number. 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

### 9.3.1 State model for transparent mode entities

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

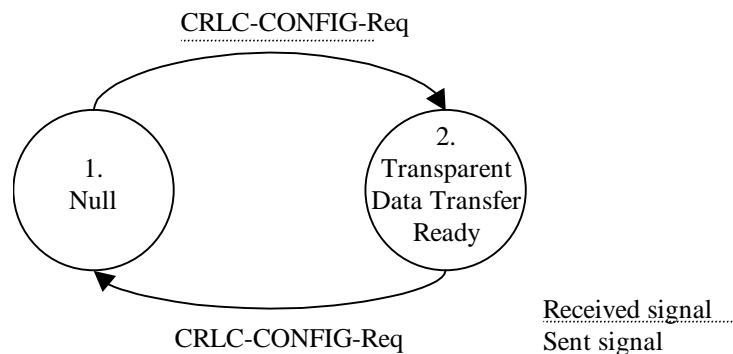
#### 9.3.1.1 Null State

In the 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 higher layer the RLC entity is created and transparent data transfer ready state is entered.

#### 9.3.1.2 Transparent Data Transfer Ready State

In the transparent data transfer ready, transparent mode data can be exchanged between the entities. Upon reception of a CRLC-CONFIG-Req from higher layer the RLC entity is terminated and the 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 and receiving). An unacknowledged mode entity can be in one of following states.

#### 9.3.2.1 Null State

In the 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 higher layer the RLC entity is created and unacknowledged data transfer ready state is entered.

#### 9.3.2.2 Unacknowledged Data Transfer Ready State

In the unacknowledged data transfer ready, unacknowledged mode data can be exchanged between the entities. Upon reception of a CRLC-CONFIG-Req from higher layer the RLC entity is terminated and the null state is entered.

#### 9.3.2.3 Local Suspend State

Upon reception of a CRLC-SUSPEND-Req from higher layer (RRC) the RLC entity is suspended and the Local Suspend state is entered. In the Local Suspend state RLC shall not send RLC-PDUs with  $SN \geq VT(US) + N$ . Upon reception of a CRLC-RESUME-Req from higher layer (RRC) the RLC entity is resumed and the Data Transfer Ready state is entered.



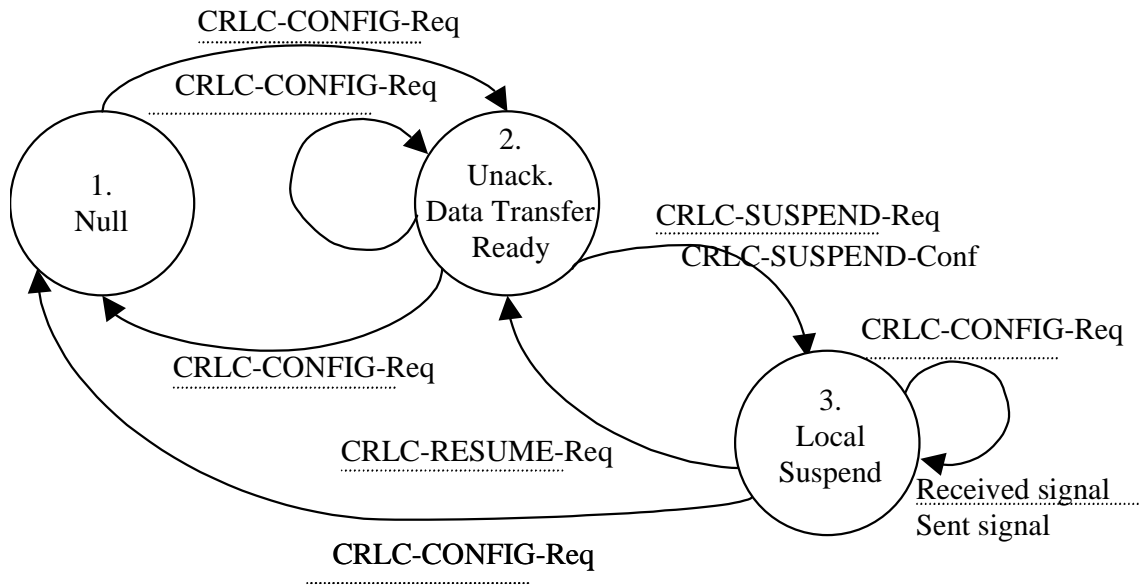


Figure 9.17: The state model for unacknowledged mode entities

### 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 following states.

#### 9.3.3.1 Null State

In the 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 higher layer the RLC entity is created and acknowledged data transfer ready state is entered.

#### 9.3.3.2 Acknowledged Data Transfer Ready State

In the acknowledged data transfer ready state, acknowledged mode data can be exchanged between the entities. Upon reception of a CRLC-CONFIG-Req from higher layer the RLC entity is terminated and the null state is entered.

Upon errors in the protocol, the RLC entity sends a RESET PDU to its peer and enters the reset pending state.

Upon reception of a RESET PDU, the RLC entity resets the protocol (see 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 takes no action.

#### 9.3.3.3 Reset Pending State

In the 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 higher layer the RLC entity is terminated and the null state is entered.

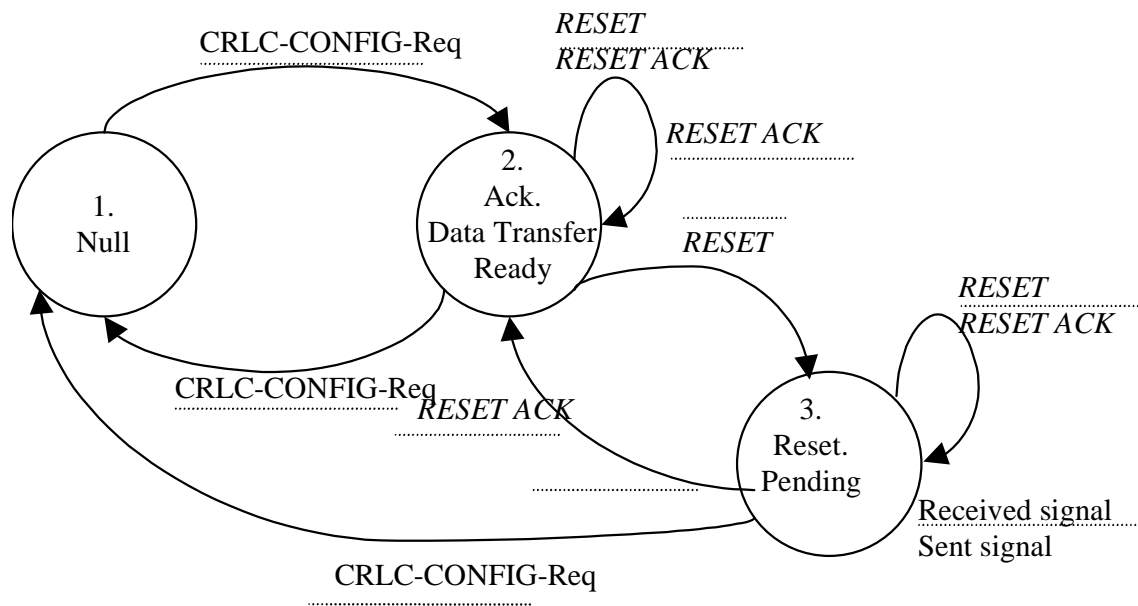
Upon reception of a RESET ACK PDU with the same RSN value as in the corresponding RESET PDU, the RLC entity resets the protocol (see subclause 11.4.4), sets the hyper frame number HFN (DL HFN when the RESET ACK is received in UE or UL HFN when the RESET ACK is received in UTRAN) equal to the HFNI field in the RESET ACK PDU and one of the following state transitions take place.

The RLC entity enters the acknowledged data transfer ready state if Reset Pending State was entered from Acknowledged Data Transfer Ready State or if Reset Pending State was entered from Local Suspend State and a CRLC-RESUME-Req was received in Reset Pending State.

The RLC entity enters into Local Suspend State if Reset Pending State was entered from Local Suspend State or if Reset Pending State was entered from Acknowledged Data Transfer Ready State and a CRLC-SUSPEND-Req was received in Reset Pending 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 PDU, the RLC entity resets the protocol (see 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, sends a RESET ACK PDU and stays in the reset pending state.



**Figure 9.18: The state model for the acknowledged mode entities when reset is performed**

### 9.3.3.4 Local Suspend State

Upon reception of a CRLC-SUSPEND-Req from higher layer (RRC) in Acknowledge Data Transfer Ready State the RLC entity is suspended and the Local Suspend state is entered. In the Local Suspend state RLC shall not send an RLC-PDUs with  $SN \geq VT(S) + N$ . Upon reception of CRLC-RESUME-Req from higher layer (RRC) in this state, the RLC entity is resumed and the Data Transfer Ready state is entered.

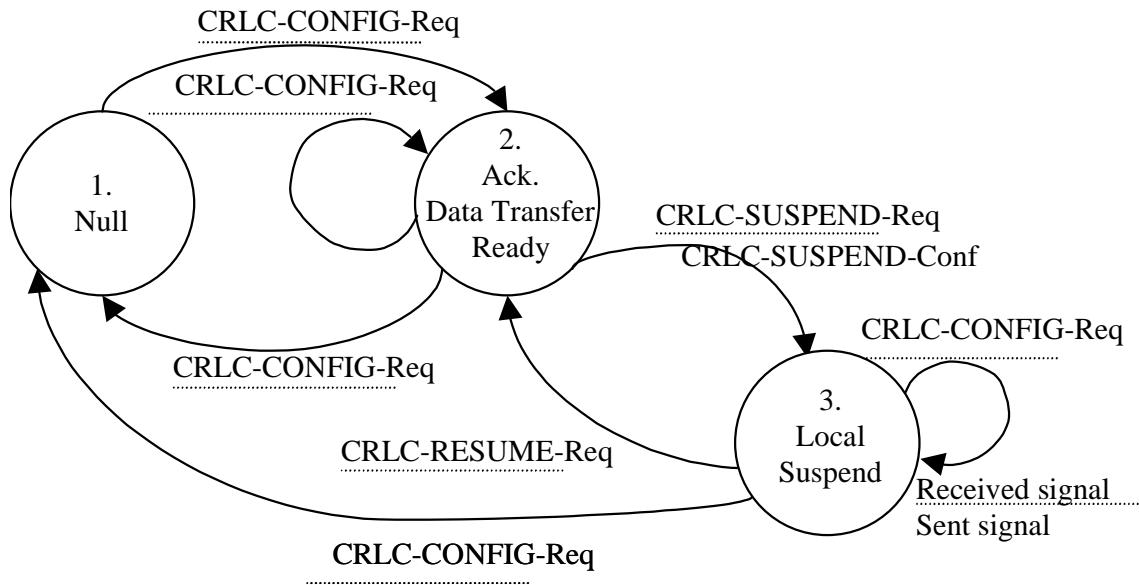


Figure 9.19: The state model for the acknowledged mode entities when local suspend is performed

## 9.4 State variables

This sub-clause describes the state variables used in the specification 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  $2^{12}$  for AM and  $2^7$  for UM; the sequence numbers cycle through the entire range: 0 through  $2^{12} - 1$  for AM and 0 through  $2^7 - 1$  for UM. All arithmetic operations on the following state variables and sequence numbers contained in this specification are affected by the modulus: VT(S), VT(A), VT(MS), VR(R), VR(H), VR(MR), VT(US) and VR(US). When performing arithmetic comparisons of variables or SN values at the sender, VT(A) and VT(US) are assumed to be the base in AM and UM respectively. When performing arithmetic comparisons of variables or SN values at the receiver, VR(R) and VR(US) are assumed to be the base in AM and UM respectively.

The RLC maintains the following state variables at the transmitter.

- a) VT(S) - Send state variable.

The sequence number of the next PDU to be transmitted for the first time (i.e. excluding retransmission). It is updated after transmission of a PDU, which includes not earlier transmitted PDUs and after transmission of a MRW SUFI which includes  $SN\_MRW_{LENGTH} > VT(S)$ . The initial value of this variable is 0.

- b) VT(A) - Acknowledge state variable.

The sequence number of the next in-sequence PDU expected to be acknowledged, which forms the lower edge of the window of acceptable acknowledgements. VT(A) is updated based on receipt of a STATUS PDU including an ACK and/or MRW\_ACK super-field. The initial value of this variable is 0.

- c) VT(DAT).

This state variable counts the number of times a PDU has been transmitted. There is one VT(DAT) for each PDU and it is incremented each time the PDU is transmitted. The initial value of this variable is 0.

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

The sequence number of the first PDU not allowed by the peer receiver [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 window. The transmitter shall not transmit a PDU with  $SN \geq VT(MS)$ . VT(MS) is updated when either VT(A) or VT(WS) is updated. The PDU with  $SN$   $VT(S)-1$  can be transmitted also when  $VT(S) \geq VT(MS)$ .

- e) VT(US) – UM data state variable.

This state variable gives the sequence number of the next UMD PDU to be transmitted. It is updated 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 function is used. It is incremented with 1 for each PDU that is transmitted. It should be incremented for both new and retransmitted PDUs. When it reaches Poll\_PDU a new poll is transmitted and the state variable is 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 function is used. It is incremented with 1 for each SDU that is transmitted. When it reaches Poll\_SDU a new poll is transmitted and the state variable is set to zero. The poll bit should be set in 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 is used to count the number of times a RESET PDU is transmitted. VT(RST) is incremented with 1 each time a RESET PDU is transmitted. VT(RST) is reset only upon the reception of a RESET ACK PDU, i.e. VT(RST) is not reset when an RLC reset occurs which was initiated from the peer RLC entity. The initial value of this variable is 0.

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

It is used to count the number of times a MRW command is transmitted. VT(MRW) is incremented with 1 each time an MRW command is transmitted. VT(MRW) is reset when the discard procedure is terminated. The initial value of this variable is 0.

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

The size that shall be used for the transmitter window. VT(WS) is set equal to the WSN field when the transmitter receives a STATUS PDU including a Window Size super-field. The initial value of this variable is Configured\_Tx\_Window\_size.

The RLC maintains the following state variables at the receiver:

a) VR(R) - Receive state variable.

The sequence number of the next in-sequence PDU expected to be received. It is set equal to SN<sub>max</sub>+1 upon receipt of the next in-sequence PDU, where SN<sub>max</sub> is the sequence number of the highest received in-sequence PDU. The initial value of this variable is 0.

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

The sequence number of the highest expected PDU. This state variable is set equal to 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.

The sequence number of the first PDU not allowed by the 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.

The sequence number of the next PDU to be received. It shall set equal to SN + 1 upon reception of a PDU. The initial value of this variable is 0.

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

The number of PDUs that should be received yet 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. It is decremented by the number of PDUs that should have been received during the 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

### a) Timer\_Poll.

This timer is only used when the poll timer trigger is used. It is started 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). The timer is stopped when receiving a STATUS PDU that contains an acknowledgement of all AMD PDUs with SN up to and including VT(S)-1 at the time the poll was submitted to lower layer, or when a negative acknowledgement of the same PDU is received. The value of the timer is signalled by RRC.

If the timer expires and no STATUS PDU fulfilling the criteria above has been received, the receiver is polled once more (either by the transmission of a PDU which was not yet sent, or by a retransmission) and the timer is restarted at the time specified above, with a new value of VT(S)-1.

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

### b) Timer\_Poll\_Prohibit.

This timer is only used when the poll prohibit function is used. It is used to prohibit transmission of polls within a certain period. The timer shall be started 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). The prohibit time is calculated from the time a PDU containing a poll is submitted to lower layer until the timer has expired. A poll shall be delayed until the prohibit time expires if a poll is triggered during the prohibit time. Only one poll shall be transmitted when the prohibit time expires even if several polls were triggered during the prohibit time. This timer will not be stopped by a received STATUS PDU. The value of the timer is signalled by RRC.

### c) Timer\_EPC.

This timer is only used when the EPC function is used and it accounts for the roundtrip delay, i.e. the time when the first retransmitted PDU should be received after a status report has been sent. The timer is started 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) and when it expires VR(EP) can start its counting-down process (see subclause 9.7.4). The value of the timer is signalled by RRC.

### d) Timer\_Discard.

This timer is used for the SDU discard function. In the transmitter, the timer is activated upon reception of an SDU from higher layer. One timer is used for each SDU that is received from higher layer. For UM/Tr, if the timer expires before the SDU is submitted to a lower layer, "SDU discard without explicit signalling" specified in subclauses 11.2.4.3/11.1.4.2 shall be started. For AM, if the timer expires before the SDU is acknowledged, "SDU discard with explicit signalling" specified in subclause 11.6 shall be started.

### e) Timer\_Poll\_Periodic.

This timer is only used when the timer based polling is used. The timer is started when the RLC entity is created. Each time the timer expires, 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). If there is no PDU to be transmitted 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 RRC.

### f) Timer\_Status\_Prohibit.

This timer is only used when the STATUS prohibit function is used. It prohibits the receiving side from sending status reports containing any of the SUFIs LIST, BITMAP, RLIST or ACK. The timer is started when the successful or unsuccessful transmission of the last STATUS PDU in a status report is indicated by lower layer (in UE) or the last STATUS PDU in 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 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. The value of the timer is signalled by RRC.

## g) Timer\_Status\_Periodic.

This timer is only used when timer based status report sending is used. The timer is started when the RLC entity is created. Each time the timer expires the transmission of a status report is triggered and the timer is restarted. The value of the timer is signalled by RRC. This timer can be blocked by RRC. In this case, the timer shall not be active. The timer shall be reset and restarted when it is unblocked by RRC.

## h) Timer\_RST.

This timer is used to detect the loss of RESET ACK PDU from the peer RLC entity. This timer is started when the successful or unsuccessful transmission of a RESET PDU is indicated by lower layer (in UE) or a RESET PDU is submitted to lower layer (in UTRAN). It will only be stopped upon reception of RESET ACK PDU, i.e. this timer is not stopped when an RLC reset occurs which was initiated from the peer RLC entity. If it expires, RESET PDU will be retransmitted. The value of the timer is signalled by RRC.

## i) Timer\_MRW.

This timer is used as part of the Move Receiving Window protocol. It is used to trigger the retransmission of a status report containing an MRW SUFI field. The timer is started when the successful or unsuccessful transmission of a STATUS PDU containing the MRW SUFI is indicated by lower layer (in UE) or a STATUS PDU containing the MRW SUFI is submitted to 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 be stopped when one of the termination criteria for the SDU discard is fulfilled. The value of the timer is signalled by RRC.

## 9.6 Protocol Parameters

The values of the protocol parameters in this subclause are signalled by RRC.

## a) MaxDAT.

It is the maximum value for the number of retransmissions of a PDU. 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 higher 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  $0 \leq \text{Poll\_Window} \leq 100$ . A poll is triggered for each PDU when  $J \geq \text{Poll\_Window}$ , where J is the window transmission percentage defined by

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

where the constant 4096 is the modulus for AM described in Subclause 9.4.e) MaxRST.

It is the maximum value for the number of retransmission of RESET PDU. 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 higher layer.

## f) Configured\_Tx\_Window\_Size.

The maximum allowed transmitter window size.

g) Configured\_Rx\_Window\_Size.

The allowed receiver window size.

h) MaxMRW.

It is the maximum value for the number of retransmissions of a MRW command. 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 Specific functions

### 9.7.1 Polling function for acknowledged mode

The transmitter of AMD PDUs may poll the receiver 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 setting the polling bit. The network (RRC) controls, which triggers should be used for each RLC entity. Following triggers are possible:

1) Last PDU in buffer.

The sender triggers a poll when the last PDU available for transmission is transmitted.

2) Last PDU in retransmission buffer.

The sender triggers a poll when the last PDU to be retransmitted is transmitted.

3) Poll timer.

The timer Timer\_Poll is started 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 the timer Timer\_Poll expires a new poll is triggered.

4) Every Poll\_PDU PDU.

The sender triggers a poll every Poll\_PDU PDU. Both retransmitted and new PDUs shall be counted.

5) Every Poll\_SDU SDU.

The sender triggers a poll every Poll\_SDU SDU.

6) Window based.

The sender triggers a poll when it has reached Poll\_Window% of the transmission window.

7) Timer based.

The sender triggers a poll periodically.

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.

### 9.7.2 STATUS transmission for acknowledged mode

The receiver of AMD PDUs transmits status reports (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. The network (RRC) controls which triggers should be used for each RLC entity, except for one, which is always present. The receiver shall always send a status report when receiving a poll request. Except for that trigger following triggers are configurable:

## 1) Detection of missing PDU(s).

If the receiver detects one or several missing PDUs it shall trigger the transmission of a status report to the sender.

## 2) Timer based STATUS transfer.

The receiver triggers the transmission of a status report periodically to the sender. The timer `Timer_Status_Periodic` controls the time period. When Periodical Status blocking is configured by higher layer, the trigger shall not be active.

## 3) The EPC mechanism.

The timer `Timer_EPC` is started and the state variable `VR(EP)` is set 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 PDUs requested for retransmission have been received before the variable `VR(EP)` has reached zero, a new status report is transmitted to the peer entity. A more detailed description of the EPC mechanism is given in subclause 9.7.4.

There are two functions that can prohibit the receiver from sending a status report. The network (RRC) controls which functions should be used for each RLC entity. If any of the following functions is used the sending of the status report shall be delayed, even if any of the triggering conditions above are fulfilled:

## 1) STATUS prohibit.

The `Timer_Status_Prohibit` is started 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 is not allowed to transmit a status report during the prohibit time. If a status report was triggered during the prohibit time, the status report is transmitted after the prohibit time has expired. 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 of a status report is triggered it shall be delayed until the EPC mechanism has ended. 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.

### 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. The network (RRC) controls, 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 retransmissions	Network controlled
No_discard after MaxDAT number of retransmissions	Network controlled



### 9.7.3.1 Timer based discard, with explicit signalling

This alternative 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 higher 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.

**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 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.

### 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 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 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).

## 9.7.4 The Estimated PDU Counter for acknowledged mode

The Estimated PDU Counter is a mechanism used for scheduling the retransmission of status reports in the receiver side. With this mechanism, the receiver will send a new status report in which it requests for 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 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 PDUs, the variable VR(EP) is set equal to the number of requested PDUs. At least one requested 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 PDUs that should have been transmitted during that transmission time interval.

A special 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 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 timer Timer\_EPC typically depends on the

roundtrip delay, which consists of the propagation delay, processing time in the transmitter and 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 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 PDUs have been received correctly, the EPC mechanism ends.

### 9.7.5 Local Suspend function for acknowledged and unacknowledged mode

The higher layer (RRC) may suspend the RLC entity. The `CRLC-SUSPEND-Req` indicates this request. The RLC entity shall, when receiving this request, not send RLC PDUs with  $SN \geq VT(S) + N$  for AM and  $SN \geq VT(US) + N$  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. The suspend state is left when a `CRLC-RESUME-Req` primitive indicating resume is received.

### 9.7.6 RLC stop, RLC Continue function

The higher layer may stop 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 the RLC entity is stopped, the RLC timers are not affected. Triggered polls and status transmissions are delayed until the RLC entity is continued.

---

## 10 Handling of unknown, unforeseen and erroneous protocol data

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.

## 11 Elementary procedures

### 11.1 Transparent mode data transfer procedure

#### 11.1.1 Purpose

The transparent mode data transfer procedure is used for transferring of data between two RLC peer entities, which are operating in transparent mode. 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.



**Figure 11.1: Transparent mode data transfer procedure**

#### 11.1.2 Initiation

The sender initiates this procedure upon a request of transparent mode data transfer from higher layer. When the sender is in data transfer ready state it shall put the data received from the higher 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 RRC.

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

##### 11.1.2.1 TrD PDU contents to set

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.

#### 11.1.3 Reception of TrD PDU

Upon reception of a TrD PDU, the receiving entity reassembles (if segmentation was performed) the PDUs into RLC SDUs. RLC delivers the RLC SDUs to the higher layer through the Tr-SAP.

#### 11.1.4 Abnormal cases

##### 11.1.4.1 Undefined SDU size at receiver

If the TrD PDUs are reassembled to an SDU which has a size that is not allowed the SDU shall be discarded.

### 11.1.4.2 SDU discard without explicit signalling

Upon expiry of the Timer\_Discard on the sender side the sender 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

The unacknowledged mode data transfer procedure is used for transferring data between two RLC peer entities, which are operating in unacknowledged mode. 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.



**Figure 11.2: Unacknowledged mode data transfer procedure**

### 11.2.2 Initiation

The sender initiates this procedure upon a request of unacknowledged mode data transfer from higher layer.

When the sender is in data transfer ready state it shall segment and, if possible, concatenate the data received from the higher 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 higher layer. One timer is used for each SDU that is received from higher layer.

A UMD PDU will 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 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.

### 11.2.3 Reception of UMD PDU

Upon reception of a UMD PDU, the receiver shall update VR(US) state variable according to the received PDU(s).

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 higher layer through the UM-SAP.

## 11.2.4 Abnormal cases

### 11.2.4.1 Length Indicator value reserved for UMD PDU

Upon reception of an UMD PDU that contains Length Indicator value reserved for UMD PDU, the receiver shall discard that UMD PDU.

### 11.2.4.2 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.

### 11.2.4.3 SDU discard without explicit signalling

Upon expiry of the Timer\_Discard on the sender side the sender 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. 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 Purpose

The acknowledged mode data transfer procedure is used for transferring of data between two RLC peer entities, which are operating in acknowledged mode. 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.



**Figure 11.3: Acknowledged mode data transfer procedure**

### 11.3.2 Initiation

The sender initiates this procedure upon a request of acknowledged mode data transfer from higher layer or upon retransmission of PDUs. 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 higher layer into AMD PDUs. The 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. 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 RRC.

The VT(DAT) state variables shall be updated for each AMD PDU that is transmitted. The PDU shall not have a Sequence Number  $\geq$  VT(MS), except for the sequence number VT(S)-1; a PDU with this sequence number may be sent also when VT(S)  $\geq$  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 higher layer. One timer is used for each SDU that is received from higher layer.

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 be considered to be a padding PDU if it is:

- An AMD PDU consisting only of an RLC Header with one length indicator (indicating that the rest of the PDU is padding) and padding.
- A Status PDU consisting only of a NO\_MORE SUFI.

### 11.3.2.1 AMD PDU contents to set

If the PDU is transmitted for the first time, the Sequence Number field shall be set equal to VT(S) and VT(S) shall be updated.

The setting of the Polling bit is specified in subclause 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 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  $J \geq$  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

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.

### 11.3.3 Reception of AMD PDU by the receiver

Upon reception of an AMD PDU, the receiver shall update VR(R), VR(H) and VR(MR) state variables according to the received PDU.

If a received PDU includes a Polling bit set to 1, the STATUS PDU transfer procedure shall be initiated.

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\_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  $VR(R) \leq SN < VR(MR)$ , 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 $VT(DAT) \geq MaxDAT$

If SDU discard after MaxDAT number of retransmission is used and  $VT(DAT) \geq MaxDAT$  for any PDU, the sender shall initiate the SDU discard with explicit signalling procedure for the SDUs to which the PDU with  $VT(DAT) \geq MaxDAT$  belongs.

If No\_discard after MaxDAT number of retransmissions is used, the sender shall initiate the RLC reset procedure when  $VT(DAT) \geq MaxDAT$ .

#### 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.

## 11.4 RLC reset procedure

### 11.4.1 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 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 after the reset procedure.

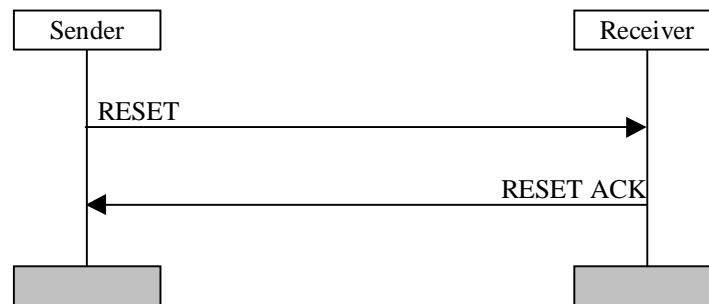


Figure 11.4: RLC reset procedure

### 11.4.2 Initiation

The procedure shall be initiated when a protocol error occurs.

The sender sends the RESET PDU when it is in data transfer ready state and enters reset pending 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., 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 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 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 receiver

Upon reception of a RESET PDU the receiver shall respond with a RESET ACK PDU. The receiver resets the state variables in 9.4 to their initial value and resets configurable parameters to their configured value. Both the transmitter and receiver side 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.

When a RESET PDU is received, the receiver shall set the 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 received RESET PDU.

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

## 11.4.4 Reception of the RESET ACK PDU by the sender

When the sender is in reset pending 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 is received in UE or UL HFN when the RESET ACK 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 transmitter and receiver side of the AM RLC entity is 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.

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 the RESET ACK PDU is discarded.

## 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.

### 11.4.5.2 Unrecoverable error ( $VT(RST) \geq MaxRST$ )

If VT(RST) becomes larger or equal to MaxRST, unrecoverable error shall be indicated to higher layer.

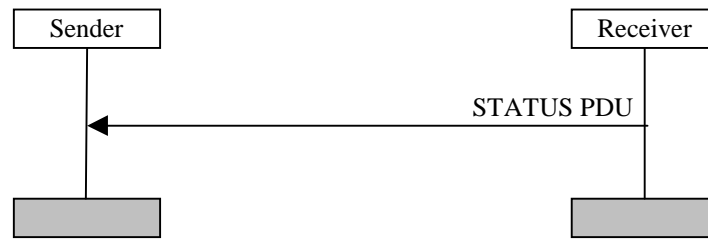
### 11.4.5.3 Reception of the RESET PDU by the sender

Upon reception of a RESET PDU in reset pending state, the sender shall 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 transmitter and receiver side 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 the reset pending state. The sender shall enter data transfer ready state only upon reception of a RESET ACK PDU with the same RSN value as in the corresponding RESET PDU.

## 11.5 STATUS report transfer procedure

### 11.5.1 Purpose

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 the sender of AMD PDUs and it is either the network or the UE.



**Figure 11.5: Status report transfer procedure**

## 11.5.2 Initiation

The receiver in any of the following cases initiates this procedure:

- 1) The poll bit in a received AMD PDU is set to 1.
- 2) Detection of missing PDUs is used and a missing PDU is detected.
- 3) The timer based STATUS transfer is used and the timer `Timer_Status_Periodic` has expired.

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

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. The sending of a piggybacked STATUS PDU follows the same rules as the sending of an ordinary STATUS PDU.

### 11.5.2.2 STATUS PDU contents to set

The size of the STATUS PDU shall be equal to one of the allowed PDU sizes. The information that needs to be transmitted in a status report can be split into several STATUS PDUs if one STATUS PDU does not accommodate all the information. A SUFI cannot be split into several STATUS PDUs. Indication of the same PDU shall not be given in more than one STATUS PDU of a STATUS report, but the ACK SUFI can be present in more than one STATUS PDU of a status report.

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 PDUs. List SUFI and/or Relative List SUFI are used to indicate missing PDUs only. Acknowledgement

SUFI is used to indicate the received PDUs. (For SUFI details see 9.2.2.11.) No information shall be given for PDUs with  $SN \geq VR(H)$ , i.e. PDUs that have not yet reached the receiver.

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

The sender shall upon reception of the STATUS PDU/piggybacked STATUS PDU update the state variables VT(A) and VT(MS) according to the received STATUS PDU/piggybacked STATUS PDU.

If the STATUS PDU includes negatively acknowledged PDUs, the acknowledged data transfer procedure shall be initiated and the PDUs shall be retransmitted. If a PDU is indicated as missing more than once in a STATUS PDU, the PDU shall be retransmitted only once. Retransmitted PDUs have higher priority than new PDUs.

### 11.5.4 Abnormal cases

#### 11.5.4.1 VR(EP) reaches zero and the requested PDUs have not been received

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

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

## 11.6 SDU discard with explicit signalling procedure

### 11.6.1 Purpose

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.

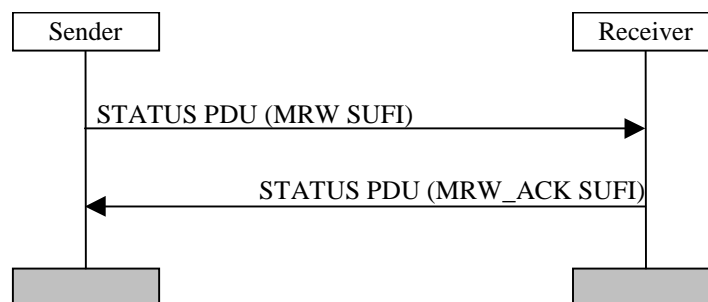


Figure 11.6: SDU discard with explicit signalling

## 11.6.2 Initiation

This procedure is initiated by the sender when the following conditions are 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 retransmissions is used, and MaxDAT number of retransmissions 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<sub>LENGTH</sub> > 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' or 'Timer\_EPC' are active.

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.

### 11.6.2.1 Piggybacked STATUS PDU

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.

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.

In order to discard a single SDU that ends in a PDU with  $SN \geq VT(A) + \text{Configured\_Tx\_Window\_Size}$ , 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 SDUs must end (i.e. the LI must be located) in a PDU with SN in the interval  $VT(A) \leq SN < VT(A) + \text{Configured\_Tx\_Window\_Size}$ .

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

The receiver shall upon reception of the STATUS PDU/piggybacked STATUS PDU discard PDUs and update the state variables VR(R), VR(H) and VR(MR) according to the received STATUS PDU/piggybacked STATUS PDU. Additionally the receiver should inform the higher layers about all of the discarded SDUs.

The receiver shall initiate the transmission of a status report containing an MRW\_ACK SUFI.

In the MRW\_ACK SUFI, SN\_ACK shall be set to the new value of VR(R), updated after reception of the MRW SUFI. The N field in the MRW\_ACK SUFI shall be set to N<sub>LENGTH</sub> field in the received MRW SUFI if the SN\_ACK field is equal to SN\_MRW<sub>LENGTH</sub>. Otherwise N shall be set to 0.

The last discarded data octet is the octet indicated by the  $N_{\text{LENGTH}}$ :th LI field of the PDU with sequence number  $\text{SN\_MRW}_{\text{LENGTH}}$  and the succeeding data octet is the first data octet to be reassembled after the discard. When  $N_{\text{LENGTH}} = 0$ , the first data octet of the PDU with sequence number  $\text{SN\_MRW}_{\text{LENGTH}}$  is the first data octet to be reassembled after the discard.

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

## 11.6.4 Termination

The procedure is terminated in the sender in the following cases:

1. On the reception of a STATUS PDU which contains an MRW\_ACK SUFI with  $\text{SN\_ACK} > \text{SN\_MRW}_{\text{LENGTH}}$  and with the N field set equal to zero.
2. On the reception of a STATUS PDU which contains an ACK SUFI indicating  $\text{VR}(\text{R}) > \text{SN\_MRW}_{\text{LENGTH}}$
3. On reception of a STATUS PDU which contains an MRW\_ACK with  $\text{SN\_ACK} = \text{SN\_MRW}_{\text{LENGTH}}$  and with the N field set equal to the  $N_{\text{LENGTH}}$  indicated in the transmitted MRW SUFI.

If one of the termination criteria above is fulfilled, Timer\_MRW shall be stopped and the discard procedure is terminated. The SDUs that are requested to be discarded shall not be confirmed to higher layer.

When  $\text{VT}(\text{MRW})$  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 discard procedure is terminated, the MRW SUFI shall be retransmitted,  $\text{VT}(\text{MRW})$  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.

## 11.6.6 Abnormal cases

### 11.6.6.1 Obsolete/corrupted MRW command

If the MRW command contains outdated information about the receiver window (receiver window already moved further than MRW command is indicating), the MRW command shall be discarded and a status report containing SUFI MRW\_ACK shall be transmitted indicating the value of  $\text{VR}(\text{R})$  and the N field shall be set to zero.

### 11.6.6.2 $\text{VT}(\text{MRW})$ equals MaxMRW

If the number of retransmission of an MRW command (i.e.  $\text{VT}(\text{MRW})$ ) reaches MaxMRW, an error indication shall be passed to RRC and RESET procedure shall be performed.

### 11.6.6.3 Reception of obsolete MRW\_ACK

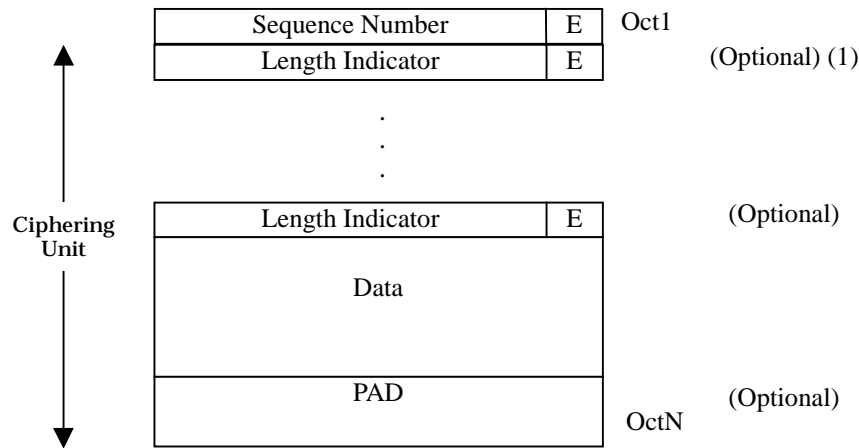
The received MRW\_ACK shall be discarded in the following cases.

1. If timer Timer\_MRW is not active.
2. If the SN\_ACK field in the received MRW\_ACK  $< \text{SN\_MRW}_{\text{LENGTH}}$  in the transmitted MRW SUFI.
3. If the SN\_ACK field in the received MRW\_ACK is equal to the  $\text{SN\_MRW}_{\text{LENGTH}}$  in the transmitted MRW SUFI and the N field in the received MRW\_ACK is not equal to the  $N_{\text{LENGTH}}$  field in the transmitted MRW SUFI
4. If the SN\_ACK field in the received MRW\_ACK  $> \text{SN\_MRW}_{\text{LENGTH}}$  in the transmitted MRW SUFI and the N field in the received MRW\_ACK is not equal to zero.

## 11.7 Cipherring

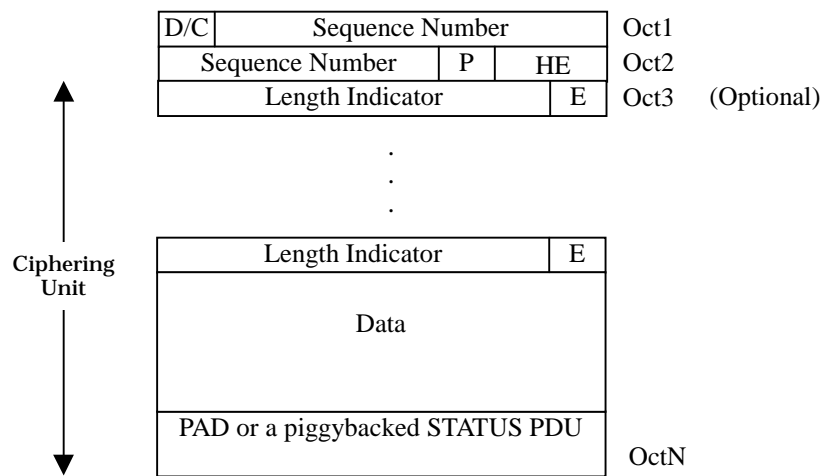
The cipherring function is performed in RLC, according to the following rules if a radio bearer is using a non-transparent RLC mode (AM or UM). The data unit that is cipherring, depends on the transmission mode as described below.

- For RLC UM mode, the cipherring unit is the UMD PDU excluding the first octet, i.e. excluding the RLC UMD PDU header. This is shown below in Figure 11.7.1.



**Figure 11.7.1: Cipherring unit for a UMD PDU**

- For RLC AM mode, the cipherring unit is the AMD PDU excluding the first two octets, i.e. excluding the RLC AMD PDU header. This is shown below in Figure 11.7.2.



**Figure 11.7.1: Cipherring unit for an AMD PDU**

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

The parameters that are required by RLC for cipherring are defined in [10] and are input to the cipherring 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 (Cipherring Key)

## 11.8 RLC re-establishment procedure

The RLC re-establishment procedure is used when higher layer request 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. All RLC PDUs in the RLC receiver and transmitter shall be discarded. The hyper frame number (HFN) in UL and DL shall be set to the value configured by higher layer. After the re-establishment, RLC shall enter the data transfer ready state.

## Annex A (informative): Change history

Change history							
Date	TSG #	TSG Doc.	CR	Rev	Subject/Comment	Old	New
10/1999	RP-05	RP-99465	-		Approved at TSG-RAN #5 and placed under Change Control	-	3.0.0
12/1999	RP-06	RP-99641	001		RLC: Editorial corrections	3.0.0	3.1.0
	RP-06	RP-99641	002	1	Editorial changes on RLC protocol specification	3.0.0	3.1.0
	RP-06	RP-99643	003	1	MRW procedure	3.0.0	3.1.0
	RP-06	RP-99643	004		SDU Discard Functionality	3.0.0	3.1.0
	RP-06	RP-99643	005	2	Change in RLC control PDU format	3.0.0	3.1.0
	RP-06	RP-99642	006	1	Editorial corrections regarding CTCH	3.0.0	3.1.0
	RP-06	RP-99641	007		Updated RLC SDL	3.0.0	3.1.0
	RP-06	RP-99642	011		RLC Editorial Changes	3.0.0	3.1.0
	RP-06	RP-99642	013		Editorial Modification on RLC specification	3.0.0	3.1.0
	RP-06	RP-99641	014		Editorial changes	3.0.0	3.1.0
	RP-06	RP-99642	015		Change to one PU in a AMD PDU	3.0.0	3.1.0
	RP-06	RP-99643	016	1	Introduction of RLC suspend state	3.0.0	3.1.0
	RP-06	RP-99641	017	1	RLC editorial corrections	3.0.0	3.1.0
01/2000	-	-	-		Editorial corrections in title and Annex A (SDL)	3.1.0	3.1.1
	-	-	-		Correction of persistent error regarding SDL in Table of Contents	3.1.1	3.1.2
03/2000	RP-07	RP-000040	018	1	RLC editorial changes	3.1.2	3.2.0
	RP-07	RP-000040	021	1	Corrections to RLC	3.1.2	3.2.0
	RP-07	RP-000040	025	2	Corrections to RLC	3.1.2	3.2.0
	RP-07	RP-000040	026	1	STATUS PDUs	3.1.2	3.2.0
	RP-07	RP-000040	027	1	Clarification of RLC AMD Model	3.1.2	3.2.0
	RP-07	RP-000040	028		Corrections to Timer_discard procedures	3.1.2	3.2.0
	RP-07	RP-000040	029	1	Segmentation of RLC SDUs	3.1.2	3.2.0
	RP-07	RP-000040	030	2	Modification of SDU discard to support virtual PDCP sequence numbers	3.1.2	3.2.0
	RP-07	RP-000040	031		Removal of SCCH	3.1.2	3.2.0
	RP-07	RP-000040	032		Updated RLC SDL	3.1.2	3.2.0
	RP-07	RP-000040	033	1	RLC Editorial Changes	3.1.2	3.2.0
	RP-07	RP-000040	034		Order of bit transmission for RLC PDUs	3.1.2	3.2.0
06/2000	RP-08	RP-000220	038		Corrections to RLC	3.2.0	3.3.0
	RP-08	RP-000220	039		Correction to the description of the MRW SUFI fields	3.2.0	3.3.0
	RP-08	RP-000220	040	1	Editorial corrections to length indicators and local suspend rate	3.2.0	3.3.0
	RP-08	RP-000220	041	4	Clarification of the RESET PDU	3.2.0	3.3.0
	RP-08	RP-000220	043	1	Clarification of RLC/MAC interaction	3.2.0	3.3.0
	RP-08	RP-000220	044	2	General RLC corrections	3.2.0	3.3.0
	RP-08	RP-000220	045		Clarification of RLC Transparent Mode operation	3.2.0	3.3.0
	RP-08	RP-000220	048		Editorial corrections to abbreviations, SCCH, BCCH	3.2.0	3.3.0
	RP-08	RP-000220	052		Updated RLC SDL	3.2.0	3.3.0
	RP-08	RP-000220	053		Correction to RLC	3.2.0	3.3.0
	RP-08	RP-000220	055		RLC Logical Channel mapping	3.2.0	3.3.0
	RP-08	RP-000220	057		Correction of EPC timer mechanism	3.2.0	3.3.0
09/2000	RP-09	RP-000358	059	1	State variables after window change	3.3.0	3.4.0
	RP-09	RP-000358	060	4	SDU discard	3.3.0	3.4.0
	RP-09	RP-000358	061	5	General RLC corrections	3.3.0	3.4.0
	RP-09	RP-000358	066		Editorial changes to RLC	3.3.0	3.4.0
	RP-09	RP-000358	067	4	Correction to RLC window size range	3.3.0	3.4.0
	RP-09	RP-000358	068	2	Window based polling	3.3.0	3.4.0
	RP-09	RP-000358	070	2	General corrections to RLC	3.3.0	3.4.0
	RP-09	RP-000358	071		State Transition in RLC Acknowledged Mode	3.3.0	3.4.0
	RP-09	RP-000358	073		Clarification of the Length Indicators	3.3.0	3.4.0
	RP-09	RP-000358	076	1	RLC corrections	3.3.0	3.4.0
	RP-09	RP-000358	077	1	Corrections to reset procedure and length indicator definitions	3.3.0	3.4.0
	RP-09	RP-000358	078		RLC Modes for SHCCH	3.3.0	3.4.0
	RP-09	RP-000358	079		CCCH in UM RLC	3.3.0	3.4.0
12/2000	RP-10	RP-000568	080	1	Length Indicator and PDU formats	3.4.0	3.5.0
	RP-10	RP-000568	083	3	Clarification to the Estimated PDU Counter	3.4.0	3.5.0
	RP-10	RP-000568	084	2	Model of UM and AM entities	3.4.0	3.5.0
	RP-10	RP-000568	085	1	General RLC corrections	3.4.0	3.5.0
	RP-10	RP-000568	086	1	General RLC corrections	3.4.0	3.5.0
	RP-10	RP-000568	087	5	RLC timers	3.4.0	3.5.0
	RP-10	RP-000568	088	1	Reset procedure	3.4.0	3.5.0
	RP-10	RP-000568	089	1	Editorial corrections to RLC	3.4.0	3.5.0



Change history							
Date	TSG #	TSG Doc.	CR	Rev	Subject/Comment	Old	New
	RP-10	RP-000568	090	2	RLC UM protocol	3.4.0	3.5.0
	RP-10	RP-000568	092	2	Clarification to window size parameters, MRW SUFI and window based polling	3.4.0	3.5.0
	RP-10	RP-000568	093	3	General RLC Corrections	3.4.0	3.5.0
	RP-10	RP-000568	094	1	RLC Reset handling	3.4.0	3.5.0
	RP-10	RP-000568	095		Inclusion of stage 3 for ciphering	3.4.0	3.5.0
03/2001	RP-11	RP-010026	097	1	Clarification on LIST SUFI and RLIST SUFI	3.5.0	3.6.0
	RP-11	RP-010026	098	1	Corrections and clarifications for SDU discard without explicit signalling	3.5.0	3.6.0
	RP-11	RP-010026	099	1	Tr mode operation	3.5.0	3.6.0
	RP-11	RP-010026	100	1	Timer based discard with explicit signalling	3.5.0	3.6.0
	RP-11	RP-010026	101		Annex updates	3.5.0	3.6.0
	RP-11	RP-010026	103		Clarification on MRW SUFI and SDU discard procedure	3.5.0	3.6.0
	RP-11	RP-010026	104	1	General clarification on SN arithmetic comparison	3.5.0	3.6.0
	RP-11	RP-010026	105	2	General clarification on RLC header and PDU header	3.5.0	3.6.0
	RP-11	RP-010026	106	1	Clarification on the primitives between RLC and higher layers	3.5.0	3.6.0
	RP-11	RP-010026	107	1	Clarification on the model of AM entity	3.5.0	3.6.0
	RP-11	RP-010026	109	2	Clarification on UMD transfer procedure	3.5.0	3.6.0
	RP-11	RP-010026	110	1	RLC status transmission in CELL_PCH and URA_PCH	3.5.0	3.6.0
	RP-11	RP-010026	111		Re-establishment description	3.5.0	3.6.0
	RP-11	RP-010026	112	1	Clarifications on the RESET and RESET ACK PDU sizes	3.5.0	3.6.0
	RP-11	RP-010026	113	1	Editorial corrections and clarifications	3.5.0	3.6.0
	RP-11	RP-010026	114	1	Clarifications on the RLC-AM-DATA-Conf primitive	3.5.0	3.6.0
	RP-11	RP-010026	116		Removal of the payload unit concept	3.5.0	3.6.0
	RP-11	RP-010026	118	2	Padding Blocks and TFC selection pre-empting	3.5.0	3.6.0

---

## History

<b>Document history</b>		
V3.1.2	January 2000	Publication
V3.2.0	March 2000	Publication
V3.3.0	June 2000	Publication
V3.4.0	September 2000	Publication
V3.5.0	December 2000	Publication
V3.6.0	March 2001	Publication