TS RAN S2.22 V~~0~~1.~~01~~0.~~2 ~~0 (1999-04)

Technical Specification

TSG-RAN Working Group 2 (Radio layer 2 and Radio layer 3) *TSGR2#4(99) 358*
Berlin, Germany, 25 - 28 May 1999

Agenda Item: **4.2**

Source: **Editor**

Title: **3GPP 25.322 V1.0.0: Description of the RLC Protocol**

Document for:

3GPP

3rd Generation Partnership Project (3GPP);
Technical Specification Group (TSG) RAN;
Working Group 2 (WG2);

Description of the RLC Protocol

3GPP

Reference

<Workitem> (<Shortfilename>.PDF)

Keywords

Digital cellular telecommunications system,
Universal Mobile Telecommunication System
(UMTS), UTRA, IMT-2000

3GPP

Postal address

Office address

Internet

secretariat@3gpp.org
Individual copies of this deliverable
can be downloaded from
http://www.3gpp.org

Copyright Notification

Contents

Intellectual Property Rights

IPRs essential or potentially essential to the present deliverable may have been declared to ETSI/3GPP. The information pertaining to these essential IPRs, if any, is publicly available for ETSI members and non-members, free of charge. This can be found in the latest version of the ETSI Technical Report: ETR 314: "Intellectual Property Rights (IPRs); Essential or potentially Essential, IPRs notified to ETSI in respect of ETSI standards". The most recent update of ETR 314, is available on the ETSI web server or on request from the Secretariat.
Pursuant to the ETSI Interim 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 the ETR 314, which are, or may be, or may become, essential to the present document.

Foreword

This Technical Specification (TS) has been produced by the 3rd Generation Partnership Project (3GPP).
The contents of this TS are subject to continuing work within 3GPP TSG-RAN and may change following formal TSG RAN approval.

1. Scope

The scope of this description is to describe the RLC protocol. A description document is intermediate between a stage 2 document and a protocol specification. Once completed, it should be sufficient for manufacturers to start some " high level design " activities. It should allow as well to assess the complexity of the associated protocol. After the completion of a description document, the drafting of the protocol specification should not have to face difficulties which would impact the other protocols i.e. the radio interface protocol architecture should be stable. This means that some procedures which are felt critical in terms of complexity will need to be studied in more details in the description document so that no problem is faced in the writing of the final protocol.

The following lists typical contents for a description document :

1. list of procedures
2. logical flow diagrams for normal procedures
3. logical description of message (where it should be possible to guess roughly the size of the various information elements)
4. principles for error handling
5. some exceptional procedures which are felt critica
6. It should, as far as possible, have the same format and outline as the final specification

The following is not covered

1. exact message format
2. all scenarios

2. References

[1] UMTS XX.XX, UTRAN Architecture description;
[2] Vocabulary used in the UMTS L2&L3 Expert Group;
[3] S2.01, Radio Interface Protocol Architecture Ver. 0.0.1
[4] S2.02, Layer 1; General requirements, Ver. 0.0.1
[5] S2.03, Description of UE States and Procedures in Connected Mode, Ver. 0.0.1
[6] S2.04 , UE Procedures in Idle Mode
[7] S2.21, Description of the MAC Protocol, Ver. 0.0.1
[8] S2.31b, Description of the RRC Protocol, Ver. 0.0.1

3. Definitions and Abbreviations

ARQ	Automatic Repeat Request
BCCH	Broadcast Control Channel
BCH	Broadcast Channel
C-	Control-
CC	Call Control
CCCH	Common Control Channel
CCH	Control Channel
CCTrCH	Coded Composite Transport Channel
CN	Core Network
CRC	Cyclic Redundancy Check
DC	Dedicated Control (SAP)
DCCH	Dedicated Control Channel
DCH	Dedicated Channel
DL	Downlink
DSCH	Downlink Shared Channel
DTCH	Dedicated Traffic Channel
FACH	Forward Link Access Channel
FCS	Frame Check Sequence
FDD	Frequency Division Duplex
GC	General Control (SAP)
HO	Handover
ITU	International Telecommunication Union
kbps	kilo-bits per second
L1	Layer 1 (physical layer)
L2	Layer 2 (data link layer)
L3	Layer 3 (network layer)
MAC	Medium Access Control
MS	Mobile Station
MM	Mobility Management
Nt	Notification (SAP)
PCCH	Paging Control Channel
PCH	Paging Channel
PDU	Protocol Data Unit
PU	Payload Unit.
PHY	Physical layer
PhyCH	Physical Channels
RACH	Random Access Channel
RLC	Radio Link Control
RNTI	Radio Network Temporary Identity
RRC	Radio Resource Control
SAP	Service Access Point
SCCH	Synchronization Control Channel
SCH	Synchronization Channel
SDU	Service Data Unit
TCH	Traffic Channel
TDD	Time Division Duplex
TFI	Transport Format Indicator
TFCI	Transport Format Combination Indicator
TPC	Transmit Power Control
U-	User-
UE	User Equipment
UL	Uplink
UMTS	Universal Mobile Telecommunications System
URA	UTRAN Registration Area
UTRA	UMTS Terrestrial Radio Access
UTRAN	UMTS Terrestrial Radio Access Network

4. General

4.1. Objective

4.2. Overview on sublayer architecture

[The RLC Sublayer supports, for the RLC PDU Mechanism, the following features:

- *Fixed Size RLC PDU with the possibility to adjust the number of PU per transmission time interval.*
- *Multiple Fixed Size RLC PDU with a RLC PDU Header Compression (This second feature should be mandatory for the U.E).*

One of the two options can be chosen during the RLC Configuration Phase.]

4.2.1. Model of RLC

Figure 4-1~~Figure 4-1Figure 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. The dashed lines between the AM-Entities illustrate the possibility to send the RLC control data (e.g. resynchronisation PDUs and acknowledgements) and data PDUs on separate logical channels. More detailed descriptions of the different entities are given in subsections 4.2.1.1, 4.2.1.2,4.2.1.3.

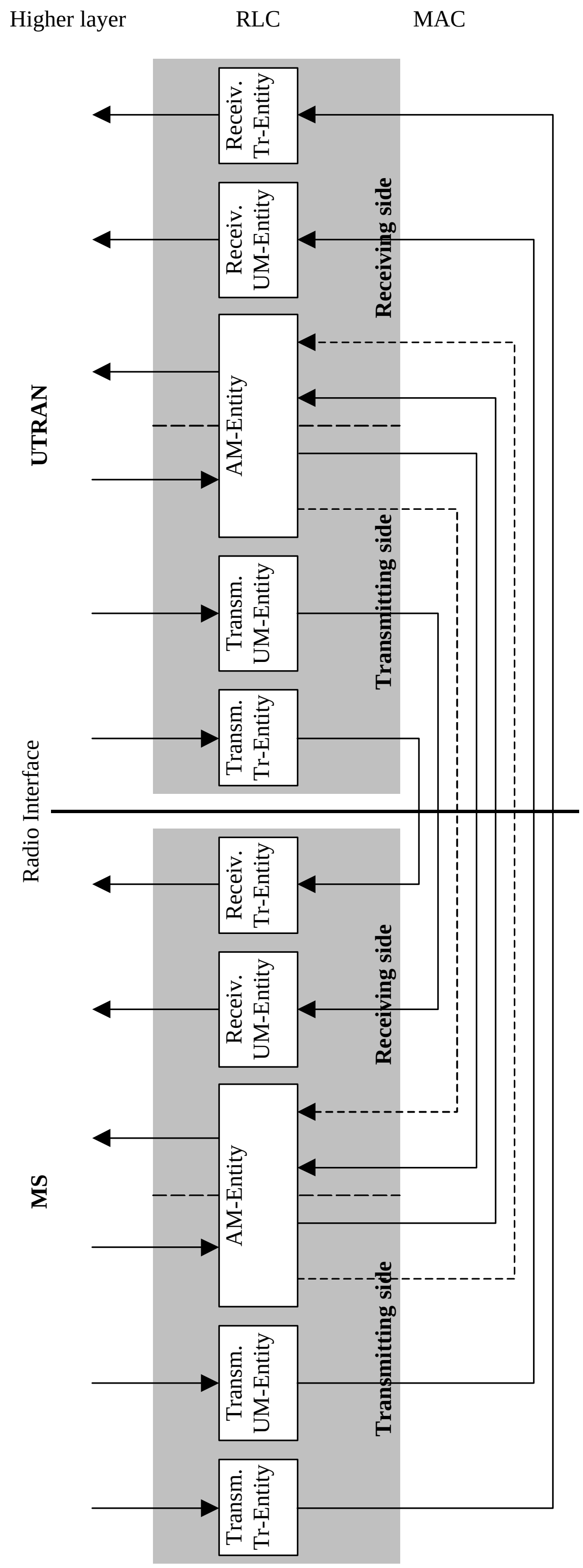


Figure 4-1 Overview model of RLC.

4.2.1.1. Transparent mode entities

Figure 4-2~~Figure 4-2Figure 4-2~~ below shows the model of two transparent mode peer entities.

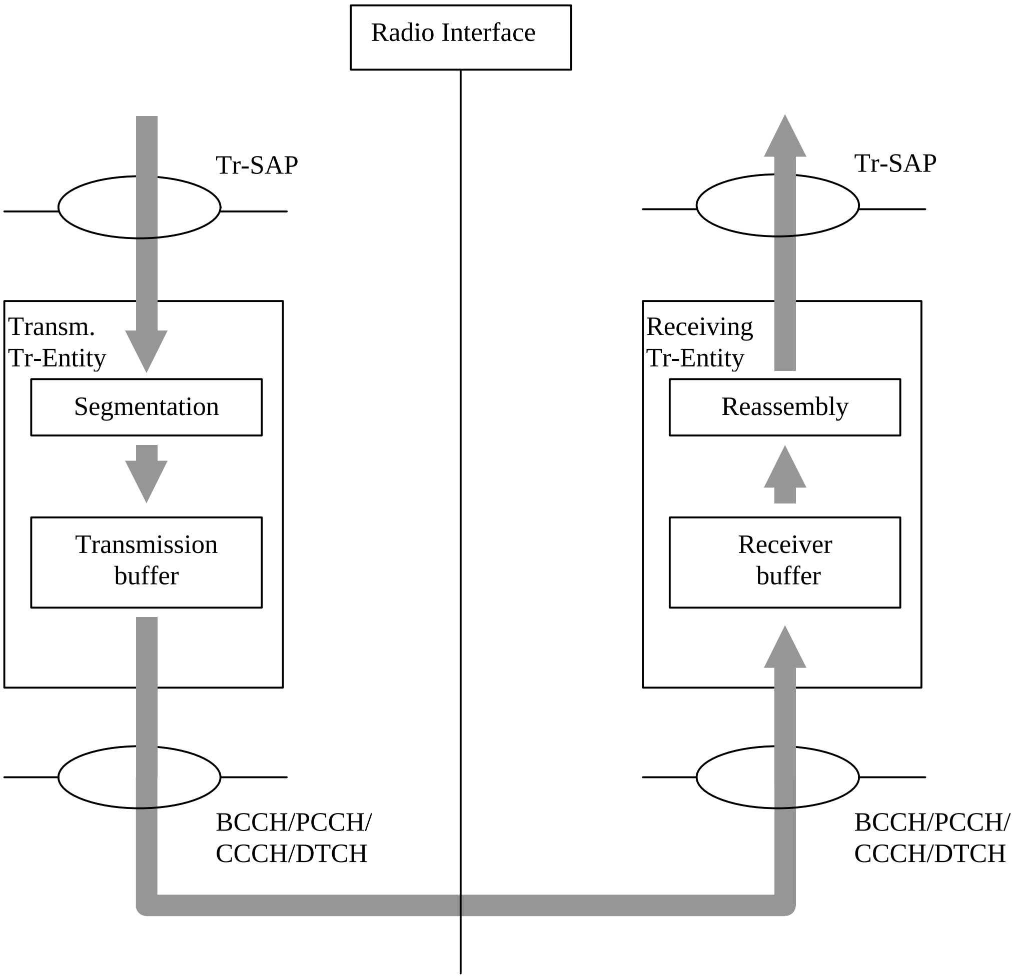


Figure 4-2~~222~~ Model of two transparent mode peer entities.

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 MAC through either a BCCH, PCCH or a DTCH. The delivery of RLC PDUs to MAC through CCCH is FFS. Which type of logical channel depends on if the higher layer is located in the control plane (BCCH, PCCH, CCCH) or user plane (DTCH).

The Tr-entity receives PDUs through from one of the logical channels from the MAC sublayer. 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~~Figure 4-3Figure 4-3~~ below shows the model of two unacknowledged mode peer entities.

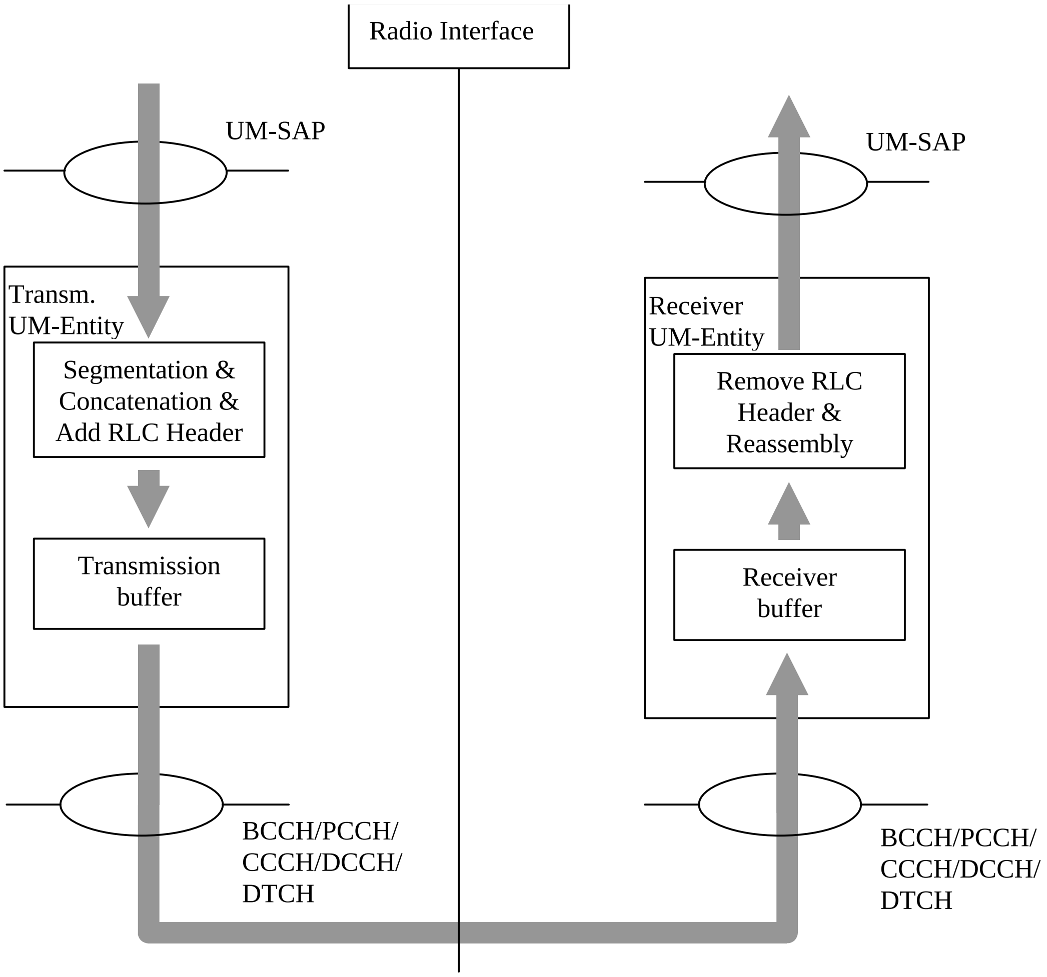


Figure 4-3~~333~~ Model of two unacknowledged mode peer entities.

The transmitting UM-entity receives SDUs from the higher layers. If the SDU is very large it is segmented into RLC PDUs of appropriate size. The SDU might also be concatenated with other SDUs. RLC adds a header and the PDU is placed in the transmission buffer. RLC delivers the RLC PDUs to MAC through either a DCCH or a DTCH. The delivery of RLC PDU's to MAC through BCCH, PCCH, CCCH, is for FFS. Which type of logical channel depends on if the higher layer is located in the control plane (BCCH, PCCH, CCCH, DCCH) or user plane (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. After that the SDUs are delivered to the higher layer.

4.2.1.3. Acknowledged mode entity

Figure 4-4~~Figure 4-4Figure 4-4~~ below shows the model of an acknowledged mode entity.

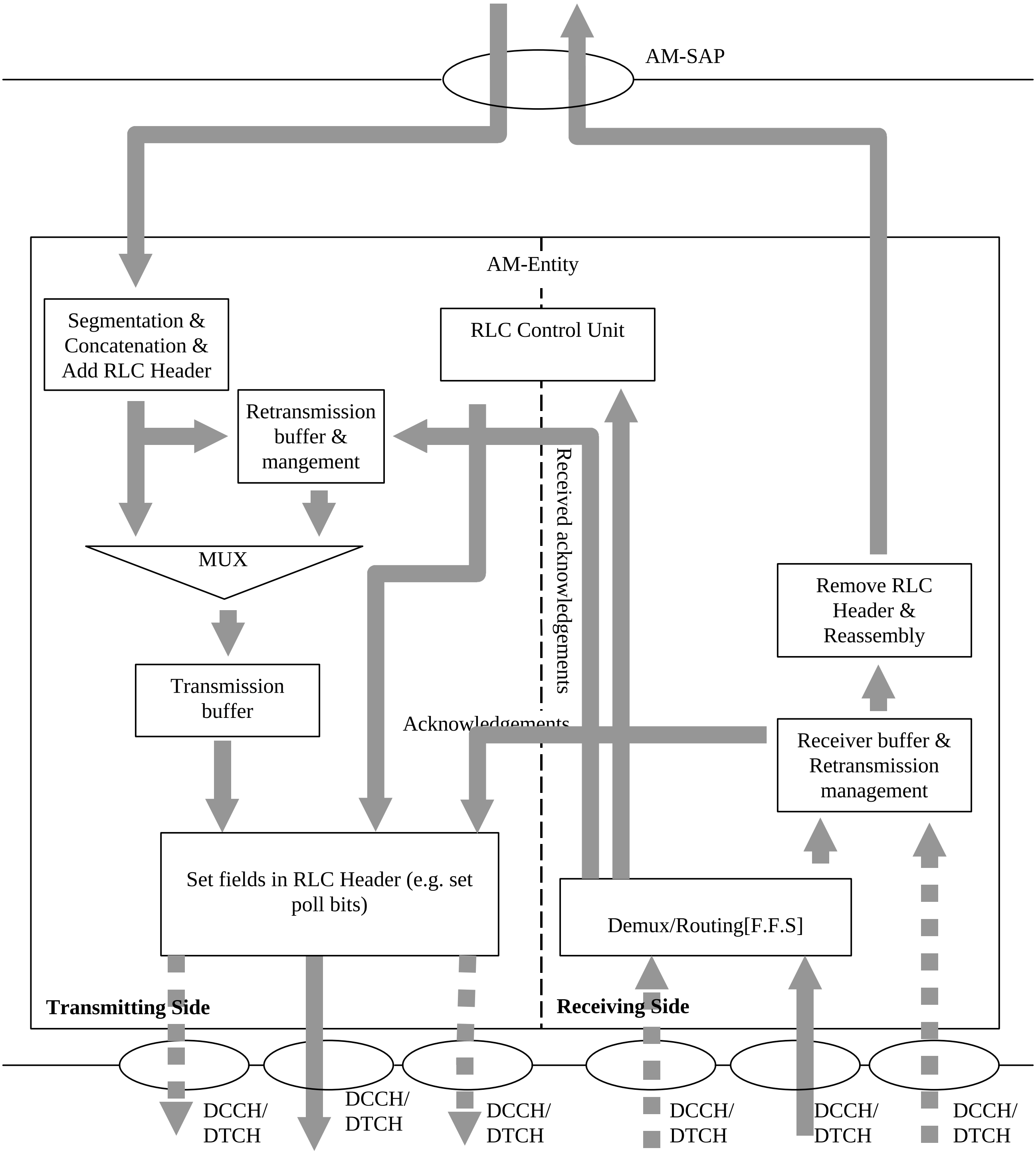

AM-SAP
AM-Entity
Segmentation & Concatenation & Add RLC Header
RLC Control Unit
Retransmission buffer & mangement
Received acknowledgements
MUX
Remove RLC Header & Reassembly
Transmission buffer
Acknowledgements
Receiver buffer & Retransmission management
Set fields in RLC Header (e.g. set poll bits)
Demux/Routing[F.F.S]
Transmitting Side
Receiving Side
DCCH/ DTCH
DCCH/ DTCH
DCCH/ DTCH
DCCH/ DTCH
DCCH/ DTCH
DCCH/ DTCH

Figure 4-4444 Model of a acknowledged mode entity.

[The possibility to send higher layer information during set up of acknowledged mode is [F.F.S]]

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

For purposes of RLC buffering and retransmission handling, the operation is the same as if there would be one PU per PDU. For concatenation or padding purposes, bits of information on the length and extension, are inserted into the beginning of the last PU where data from an SDU is included. If several SDU:s fit into one PU, they are concatenated and the appropriate length indicators are inserted into the beginning of the PU. After that the PU:s are placed in the retransmission buffer and the transmission buffer. RLC PDU is constructed from PU buffers.

The MUX then decides which PDUs and when the PDUs are delivered to MAC, e.g. it could be useful to send RLC control PDUs on one logical channel and data PDUs on another logical channel. The PDUs are delivered via a function that completes the RLC-PDU header. This includes setting the poll bit,*[The setting of other bits in the Header is [F.F.S]. and will be specified when the AMD PDU Structure will be defined]*, compressing subsequent PU:s into one RLC-PDU or setting up the extended RLC-PDU header (PU:s not in sequence) where applicable. The dashed lines illustrate the case where AMD PDUs and control PDUs are transmitted on separate logical channels. The retransmission buffer also receives acknowledgements from the receiving side, which are used to indicate retransmissions of PUs and when to delete a PU from the retransmission buffer.

The Receiving Side of the AM-entity receives PDUs through one of the logical channels from the MAC sublayer. The RLC-PDU:s are expanded into separate PUs and placed in the receiver buffer until a complete SDU has been received. The receiver buffer requests retransmissions of PUs by sending negative acknowledgements to the peer entity. After that the headers are removed from the PDUs and the PDUs are reassembled into a 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.

[~~*Editor`s note: A description of tasks performed by RLC control unit is expected, as well as the relationship between the RLC control unit and other functional blocks such as multiplexing unit and retransmission management. It should be considered also the possibilities of multiplexing UM PDUs and AM PDUs in one data flow*~~]

5. Functions

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

- Connection Control;
- 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.

The following potential function(s) are regarded as further study items:

- Suspend/resume function;
- Ciphering.
- Quick repeat (FFS).

6. Services provided to upper layers

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

- **RLC connection establishment/release;**
- **Transparent data transfer Service**

  Following functions are needed to support transparent data transfer:

  - Segmentation and reassembly
  - Transfer of user data;
- **Unacknowledged data transfer Service**

  Following functions are needed to support unacknowledged data transfer:

  - Segmentation and reassembly
  - Concatenation
  - Transfer of user data;
- **Acknowledged data transfer Service**

  Following functions are needed to support acknowledged data transfer:

  - Segmentation and reassembly
  - Concatenation
  - Transfer of user data
  - Error correction
  - In-sequence delivery of higher layer PDUs
  - Duplicate detection
  - Flow Control
  - Protocol error detection and recovery;
- **QoS setting**;
- **Notification of unrecoverable errors**.
- **Multicast delivery of higher layer messages. (FFS)**

Table 6-1~~6-1~~: RLC modes and functions in UE uplink side

Service	Functions	CCCH	DCCH	DTCH
Transparent Service	Applicability	+	-	+
	Segmentation	-	-	+
Unacknowledged Service	Applicability	FFS	+	+
	Segmentation	-	+	+
	Concatenation	-	+	+
Acknowledged Service	Applicability	-	+	+
	Segmentation	-	+	+
	Concatenation	-	+	+
	Flow Control	-	+	+
	Error Correction	-	+	+
	Protocol error correction & recovery	-	+	+

.

Table 6-2~~6-2~~: RLC modes and functions in UE downlink side

Service	Functions	SCCH	BCCH	PCCH	CCCH	DCCH	DTCH
Transparent Service	Applicability	+	+	+	+	-	+
	Reassembly	+	+	+	-	-	+
Unacknowledged Service	Applicability	+	FFS~~+~~	FFS~~+~~	FFS	+	+
	Reassembly	+	+	+	-	+	+
Acknowledged Service	Applicability	-	-	-	-	+	+
	Reassembly	-	-	-	-	+	+
	Error correction	-	-	-	-	+	+
	Flow Control	-	-	-	-	+	+
	In sequence delivery	-	-	-	-	+	+
	Duplicate detection	-	-	-	-	+	+
	Protocol error correction & recovery	-	-	-	-	+	+

Table 3.

Table 6-3~~6-3~~: RLC modes and functions in UTRAN downlink side

Service	Functions	SCCH	BCCH	PCCH	CCCH	DCCH	DTCH
Transparent Service	Applicability	+	+	+	+	-	+
	Segmentation	+	+	+	-	-	+
Unacknowledged Service	Applicability	+	FFS~~+~~	FFS~~+~~	FFS	+	+
	Segmentation	+	+	+	-	+	+
	Concatenation	+	+	+	-	+	+
Acknowledged Service	Applicability	-	-	-	-	+	+
	Segmentation	-	-	-	-	+	+
	Concatenation	-	-	-	-	+	+
	Flow Control	-	-	-	-	+	+
	Error Correction	-	-	-	-	+	+
	Protocol error correction & recovery	-	-	-	-	+	+

Table 6-4~~6-4~~: RLC modes and functions in UTRAN uplink sidef

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

7. Services expected from MAC

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

- Data transfer;
- Acknowledged data transfer service by MAC for transmission on RACH/FACH is FFS.

8. Elements for layer-to-layer communication

8.1. Primitives between RLC and higher layers

The primitives between RLC and upper layers are shown in Table 8-1~~Table 8-1Table 8-1~~.

Table 8-1~~8-1~~ : Primitives between RLC and upper layers

Generic Name	Parameter			
	Req.	ind.	Resp.	conf.
RLC-AM-DATA	MU	MU	Not Defined	Not Defined
RLC-UM-DATA	MU, QR (ffs)	MU	Not Defined	Not Defined
RLC-TR-DATA	MU	MU	Not Defined	Not Defined
CRLC-CONFIGURE				
CRLC RELEASE			Not Defined	Not Defined
RLC-ESTABLISH				
RLC-RELEASE				

Each Primitive is defined as follows:

a) RLC-AM-DATA req./ind.
It is used for acknowledged data transmission mode of point-to-point connection between the same level user entities.

[Editor's note: Confirmation for the RLC-AM-DATA procedure is FFS.]

b) RLC-UM-DATA req./ind.
It is used for unacknowledged data transmission mode of point-to-point connection between the same level user entities.

c) RLC-TR-DATA req./ind
It is used for trasparent data transmission mode of point-to-point connection between the same level user entities.

d) CRLC-CONFIGURE
[FFS]

e) CRLC RELEASE
[FFS]

f) RLC-ESTABLISH
[FFS]

g) RLC-RELEASE
[FFS]

The parameter Message Unit (MU) is mapped on MU field on RLC PDU transparently in the case of RLC-AM-DATA req. or RLC-UM-DATA req. And the MU field of RLC PDU received is mapped on MU in the case of RLC-AM-DATA ind. or RLC-UM-DATA ind. transparently. Length of MU must be n octets (n is integer).
The Quick Repeat indicator (QR) indicates whether UMD PDU will be transmitted with Quick Repeat or not. It holds one of two values: "Yes" or "No". *(The need of this indicator is FFS)*

9. Elements for peer-to-peer communication

In unacknowledged transmission, only one type of unacknowledged data PDU is exchanged between peer RLC entities In acknowledged transmission, both (acknowledged) data PDUs and control PDUs are exchanged between peer RLC entities.

9.1. Protocol data units

[All the section shall be reviewed when the protocol is defined]

Data PDU

a) AMD PDU (Acknowledged Mode Data PDU)

The AMD PDU is used to convey sequentially numbered PUs containing RLC SDU data. The AMD PDU is used by the RLC when it is in the acknowledged 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 the RLC when using the unacknowledged data transfer.

Control PDU

a) BGN PDU (Begin)

The BGN PDU is used by a RLC entity in order to establish a RLC link between the entity and its peer entity.

b) BGAK PDU (Begin Acknowledge)

The BGAK PDU is an acknowledgement to the BGN PDU.

c) BGREJ PDU (Begin Reject)

The BGREJ PDU is used to reject the RLC link setup request of the peer RLC entity.

d) END PDU (End)

The END PDU is used by a RLC entity in order to release the RLC link between the entity and its peer entity.

e) ENDAK PDU (End Acknowledge)

The ENDAK PDU is an acknowledgement to the END PDU.

f) STATUS PDU ~~(Solicited Status Response)~~

The STATUS PDU is used to inform the transmitting entity about missing PUs at the receving entity either upon detection of a missing PU (unsolicited) or as a response to a polling request from the transmitting entity.

~~The STAT PDU is used to respond to a status request from the peer RLC entity. It informs the transmitter side about missing PUs at the Receiver RLC.~~

~~g) USTAT PDU (Unsolicited Status Response)~~

~~The USTAT PDU is transmitted upon detection of an erroneous transmission of one or more data PUs. It is used to inform the transmitter side about missing AMD PUs at the receiver RLC.~~

Table 9-1: RLC PDU names and descriptions

Functionality	PDU name	Description
Management of the connection	BGN	Request Initialization
	BGAK	Request Acknowledgement
	BGREJ	Connection Reject
	END	Disconnect Command
	ENDAK	Disconnect Acknowledgement
Acknowledged Data Transfer	AMD	Sequenced acknowledged mode data
	STATUS [FFS]	Solicited or Unsolicited Status Report
	~~USTAT [FFS]~~	~~Unsolicited Status Report~~
Unacknowledged Data Transfer	UMD	Sequenced unacknowledged mode data

9.2. Formats and parameters

[All the section shall be reviewed when the protocol is defined]

AMD PDU

Note: R bit may be H bit. It is FFS.

Transfers user data and requests status report by setting Poll bit.

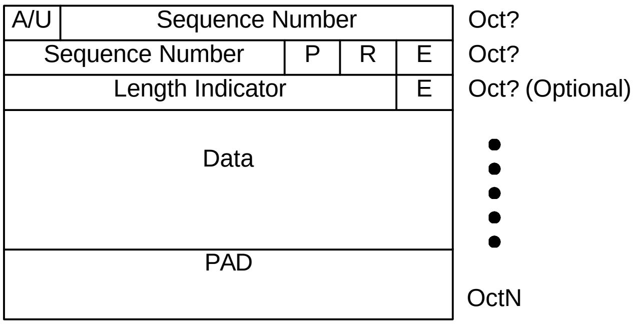


Figure 9-1. AMD PDU

UMD PDU

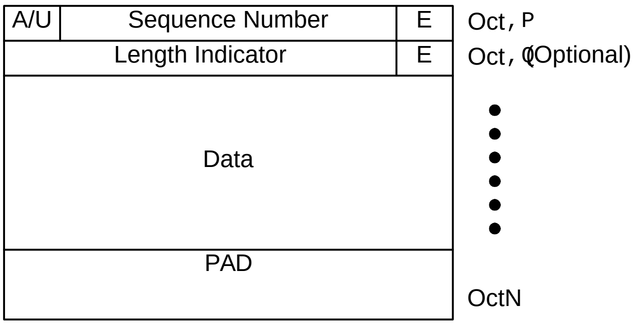


Figure 9-2. UMD PDU

BGN PDU

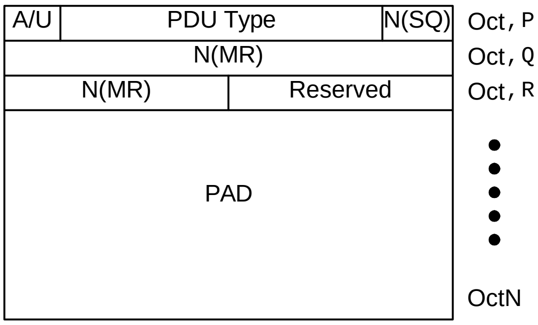


Figure 9-3. BGN PDU

BGAK PDU

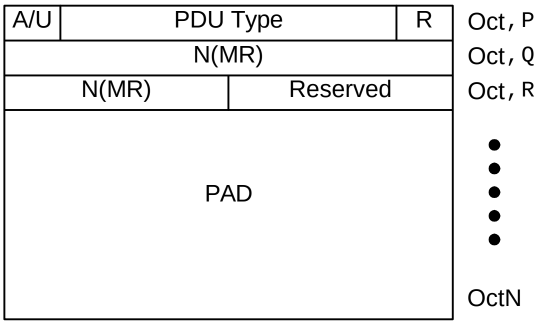


Figure 9-4. BGAK PDU

BGREJ, END, ENDAK PDU

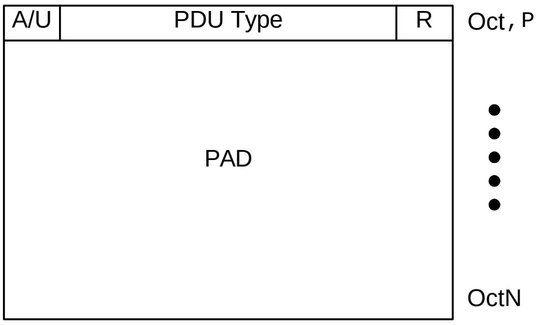


Figure 9-5. BGREJ, END, ENDAK PDU

STATUS PDU

Reports the status of receiver to transmitter when AMD PDU with status report request is received, or to inform the transmitting entity about missing PUs.

[The message format will be reconsidered when the protocol will be defined]

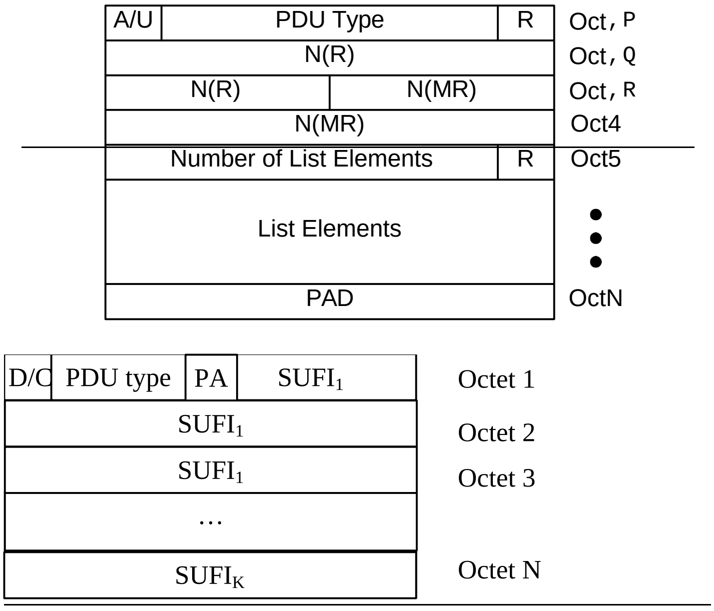


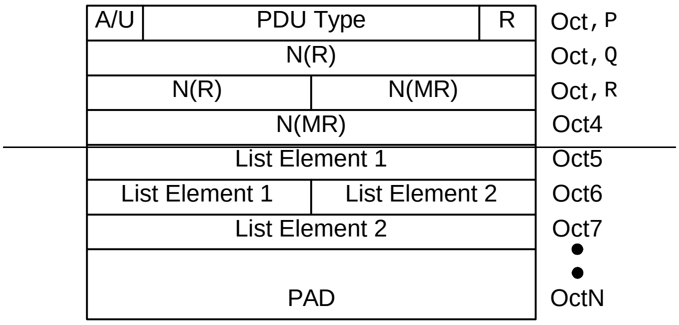
Figure 9-6. Status Information Control PDU (STATUS PDU)

The inclusion of a credit value/retransmission window size is FFS.

The maximum size of a STATUS PDU is bounded by the maximum RLC PDU size.

~~USTAT PDU~~

~~The USTAT PDU is transmitted upon detection of an erroneous transmission of one or more PUs. It is used to inform the transmitter side about missing PUs at the receiver RLC.~~



~~Figure 9-7. USTAT PDU~~

Note: Regarding STATUS-PDU~~and USTAT,~~ it is FFS. whether a bitmap type of PDU status indication would be more efficient than List elements.

The RLC PDU parameters are defined as follows:

- A/U bit: 1bit
  This field indicates Acknowledged mode data PDU or Unacknowledged mode data PDU/ Control PDU. If it indicates Acknowledged mode, the PDU is AMD PDU.

Bit	Description
0	Unacknowledged mode data PDU/ Control PDU
1	Acknowledged mode data PDU

- D/C bit: 1bit
  This 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

[Editor's note: at this stage the D/C field is related only to the STATUS PDU format. The replacement of A/D field with D/C field should be extended also to the other PDUs, since it was agreed that it is only necessary to distinghish between AMD- data and control PDU.]

- PDU Type: 6bit, for STATUS PDU this field is 3 bit length [FFS]
  This field indicates the type of Control PDU. They are indicated by the special values of sequence number field.

Bit	PDU Type	Bit	PDU Type
111111	BGN	111010 *[FFS]*	STATUS
111110	BGAK	~~111001~~	~~USTAT~~
111101	BGREJ	111000 – 110000	Reserved
111100	END		
111011	ENDAK		

[Editor's note:in accordance with the new STATUS PDU format reported, the PDU type is 3 bit length, but this differs from the other control PDU formats, which require a PDU type field of 6 bit length]

- Sequence Number (SN)
  This field indicates the sequence number of the payload unit. In normal acknowledged-mode RLC-PDU header it is the sequence number of the first PU in the PDU. If the PU:s are not in sequence, a sequence number is indicated separately for each PU in the extended header.

PDU type	Length	Notes
AMD PDU	12 bit	Used for retransmission and reassembly
UMD PDU	6 bit	Used for reassembly ***Especially "110000" – "111111" are reserved for PDU Type (Control PDU)***

- Polling bit (P): 1bit
  This field is used to request a status report (STAT PDU) from the receiver RLC.

Bit	Description
0	-
1	Request a status report

- Extension bit (E): 1bit
  This bit indicates whether the next octet will be header information (LI) or data.

Bit	Description
0	The next octet is data
1	The next octet is header information (LI)

- Reserved (R):
  One function of this field is to achieve octet alignment. Other functions are FFS. Where no functions are defined, this field shall be coded as zero. This field ignored by the receiver.

- Length Indicator (LI): 7bit
  This field is optional and is used if concatenation or padding takes place in RLC. It indicates the end of the last segment of a SDU. Especially "0000000" indicates that the previous RLC PDU is exactly filled with the last segment of a RLC SDU, and "1111111" indicates that the rest part of the RLC PDU is padding.

- Poll Answer (PA): 1bit
  This field indicates whether the status report is the answer to a poll or not

Bit	Description
0	The status report is not the answer to a polling request
1	The status report is the answer to a polling request

- **SUFI** (SUper-FIeld): 1bit
  The SUFI includes three fields: type information (type of super-field, e.g. list, bitmap or acknowledgement), length information (providing the length of a variable length field within the following value field) and a value. Figure 9-7~~Figure 9-7~~ shows the structure of the super-field. The size of the type field is non-zero but the size of the other fields may be zero.

Type
Length
Value

Figure 9-7. The Structure of a Super-Field

- Type**:** 2 bits (FFS)

Bit	Description
00	List (**LIST**)
01	Bitmap (**BITMAP**)
10	Acknowledgement (**ACK**)
11	No More Data (**NO_MORE**)

- Length**:** depending on the super-field type
  Gives the length of the variable size part of the following value field

- Value**:** variable number of bits given by the Type and the Length fields
  SUFI for a List
  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 9-8~~Figure 9-8~~ below:

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

Figure 9-8~~88~~. The List fields in a STATUS PDU for a list

LENGTH**:** 4 bits (FFS)

The number of (SN_i , L_i)-pairs in the super-field of type LIST.

SN_i **:** 12 bits
Sequence number of PU which was not correctly received.

L_i **:** 4 bits (FFS)
Number of consecutive PUs not correctly received following PU with sequence number SN_i.

SUFI for a Bitmap

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 9-9~~Figure 9-9~~ below:

Type = **BITMAP**
LENGTH
FSN
Bitmap

Figure 9-9~~99~~. The Bitmap fields in a STATUS PDU.

LENGTH**:** 4 bits (FFS)
The size of the bitmap in octets (maximum bitmap size: 2^4*8=128 bits).

FSN**:** 12 bits

The sequence number for the first bit in the bitmap.

Bitmap**:** variable number of octets given by LENGTH
Status of the SNs in the interval [FSN, FSN + LENGTH*8 - 1] indicated in the bitmap where each position can have two different values (0 and 1) with the following meaning (bit_position∈[0,LENGTH*8 - 1]):

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

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

SUFI for an Acknowledgment
The Ack Super-Field consists of a type identifier field (ACK) and a sequence number (FSN) as shown in Figure 9-10~~Figure 9-10~~ below:

Type = **ACK**
LSN

Figure 9-10~~1010~~. The ACK fields in a STATUS PDU

LSN**:** 12 bits
Acknowledges the reception of all PUs with sequence numbers < LSN (Last Sequence Number) that are *not* indicated to be erroneous in earlier parts of the STATUS PDU.

SUFI for No More Data (FFS)
The 'No More Data' Super-Field indicates the end of the data part of a STATUS PDU and is shown in Figure 9-11~~Figure 9-11~~ below:

Type=**NO_MORE**

Figure 9-11~~1111~~. NO_MORE field in a STATUS PDU

- N(SQ): 1bit
  This field carries the connection sequence value. VT(SQ) is mapped to N(SQ) whenever a new BGN PDU is transmitted. This field is used by the receiver together with VR(SQ) to identify retransmitted BGN PDU.

- N(R): 12bit
  VR(R) is mapped to N(R) whenever a STAT or USTAT PDU is generated.

- N(MR): 12bit
  VR(MR) is mapped to N(R) whenever a STAT, USTAT, BGN, or BGAK PDU is generated. This is the basis for credit granting by the receiver.

- Number of List Elements: 7bit
  It indicates the number of list elements that included in the STAT PDU.

- Header extension flag (H): 1bit
  The header extension flag indicates that the following two octets contain an extended header (SN+H+E) in the AMD PDU. The use of this flag is *[F.F.S.]*

- Data:
  In this field data from higher layer PDUs is mapped.

9.3. Protocol states

9.3.1. State model for transparent mode entities

Figure 9-12~~Figure 9-12Figure 9-8~~ 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 an RLC-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 an RLC-CONFIG.req from higher layer the RLC entity is terminated and the null state is entered.

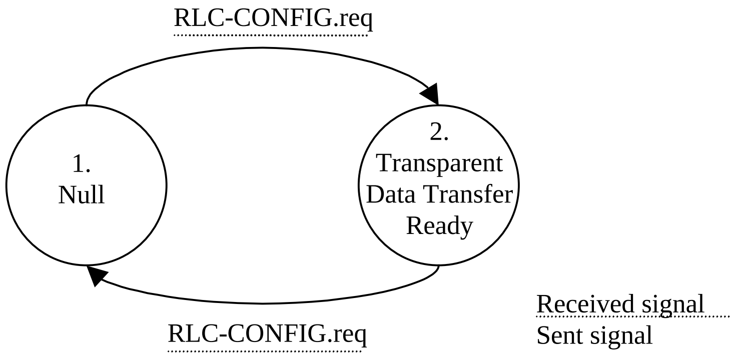


Figure 9-12~~12128~~. The state model for transparent mode entities.

9.3.2. State model for unacknowledged mode entities

Figure 9-13~~Figure 9-13Figure 9-9~~ illustrates the state model for unacknowledged mode RLC entities. 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 an RLC-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, unacknowldged mode data can be exchanged between the entities. Upon reception of an RLC-CONFIG.req from higher layer the RLC entity is terminated and the null state is entered.

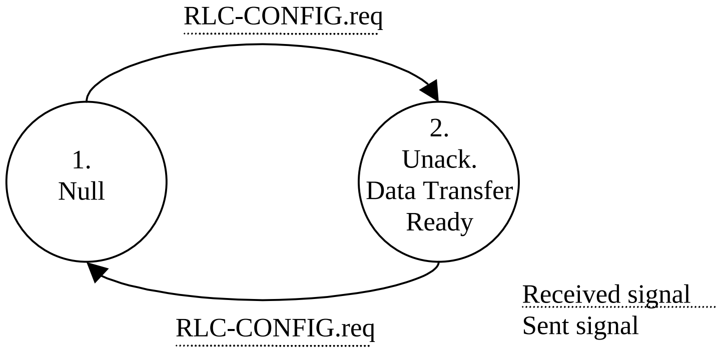


Figure 9-13~~13139~~. The state model for unacknoledged mode entities.

9.3.3. State model for acknowledged mode entities

Figure 9-14~~Figure 9-14Figure 9-10~~ illustrates the state model for the acknowledged mode RLC entity. 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 an C-RLC-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, acknowledged mode data can be exchanged between the entities. Upon reception of an RLC-CONFIG.req from higher layer the RLC entity is terminated and the null state is entered.

9.3.3.3. Recovery Pending State

In the recovery pending state the entity waits for a response from its peer entity and no data can be exchanged between the entities. Upon reception of an RLC-CONFIG.req from higher layer the RLC entity is terminated and the null state is entered.

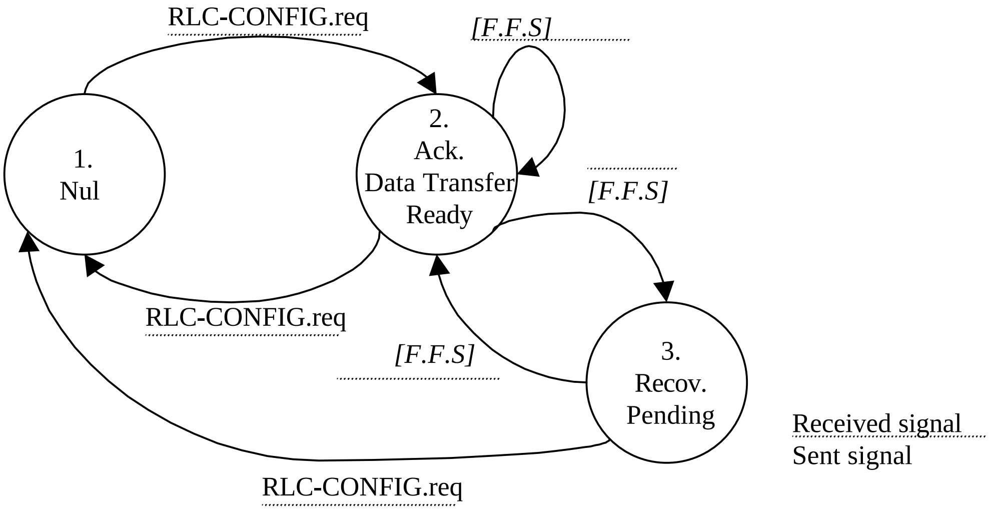


Figure 9-14~~141410~~. The state model for the acknoledged mode entities.

The messages that cause the transition between the "Ack. Data Transfer Ready" state and the "Recovery Pending" State, are [F.F.S.].

9.4. State variables

[All the section shall be reviewed when the protocol is defined]

This sub-clause describes the state variables used in the specification of the peer-to-peer protocol. PUs 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} and the sequence numbers cycle through the entire range, 0 through $2^{12} – 1$. All arithmetic operations on the following state variables and sequence numbers contained in this Recommendation are affected by the modulus: VT(S), VT(A), VT(MS), VR(R), VR(H), and VR(MR). When performing arithmetic comparisons of transmitter variables, VT(A) is assumed to be the base. When performing arithmetic comparisons of receiver variables, VR(R) is assumed to be the base. In addition, the state variables VT(SQ) and VR(SQ) use modulo 2 arithmetic and VT(US) and VT(UR) use modulo 48. The RLC maintains the following state variables at the transmitter.

a) VT(S) - Send state variable
The sequence number of the next PU to be transmitted for the first time (i.e. excluding retransmission). Incremented after transmission of a PU for the first time (i.e. excluding retransmission).

b) VT(A) - Acknowledge state variable
The sequence number of the next in-sequence PU expected to be acknowledged, which forms the lower edge of the window of acceptable acknowledgments. VT(A) is updated upon acknowledgment of in-sequence PUs.

c) VT(DAT)
This state variable is used to count the retransmission number of each PU. VT(DAT) is incremented by sending PU.

d) VT(MS) - Maximum Send state variable
The sequence number of the first PU not allowed by the peer receiver [i.e. the receiver will allow up to VT(MS) – 1]. This value represents the upper edge of the transmit window. The transmitter shall not transmit a new PU if VT(S) = VT(MS). VT(MS) is updated based on receipt of a USTAT PDU, STAT PDU, BGN PDU, BGAK PDU.

e) VT(CC) - Connection Control state variable
The number of unacknowledged BGN, END PDUs. VT(CC) is incremented upon transmission of a BGN, END PDU. If an END PDU is transmitted in response to a protocol error, RLC does not wait for an ENDAK PDU [i.e. RLC moves directly to state 1 (Idle)] and VT(CC) is not incremented.

f) VT(SQ) - Transmitter Connection Sequence state variable
This state variable is used to allow the receiver to identify retransmitted BGN PDUs. This state variable is initialized to 0 upon creation of the RLC process and incremented and then mapped into the N(SQ) field before the initial transmission of either a BGN PDU.

g) VT(US) - Unit data state variable
This state variable means new sequence number of UMD-PDU which will send next. After new UMD-PDU is sent, VT(US) will be incremented.

h) VT(QR) - Quick repeat state variable (FFS)
This state variable is used to count the retransmission number when UMD-PDU is sent by quick repeat scheme. It is incremented after UMD-PDU is sent and quick repeat will be continued until VT(QR) becomes to equal MaxQR.

The RLC maintains the following state variables at the receiver:
a) VR(R) - Receive state variable
The sequence number of the next in-sequence PU expected to be received. Incremented upon receipt of the next in-sequence PU.

b) VR(H) - Highest expected state variable
The sequence number of the next highest expected PU. This state variable is updated whenever a new PU is received.

c) VR(MR) - Maximum acceptable Receive state variable
The sequence number of the first PU not allowed by the receiver [i.e. the receiver will allow up to VR(MR) – 1]. The receiver shall discard PUs with N(S) = VR(MR), (in one case, such a PU may cause the transmission of a USTAT). Updating VR(MR) is implementation dependent, but VR(MR) should not be set to a value < VR(H).

d) VR(SQ) - Receiver Connection Sequence state variable
This state variable is used to identify retransmitted BGN PDUs. Upon reception of a BGN PDU, this state variable is compared to the value of N(SQ) and then assigned the value of N(SQ). If the values are different, the PDU is processed and VR(SQ) is set to N(SQ). If they are equal, the PDU is identified as a retransmission.

e) VR(US) - Receiver Send Sequence state variable
The sequence number of the latest UMD PDU to be received. It is used to check the duplication receive. When new UMD PDU is received, VR(US) is compared with N(US). If VR(US) is equal to N(US), this PDU is quashed because duplication receive happens. And if not, N(US) is substituted for VR(US).
f) VR(EP) – Estimated PDU Counter state variable (FFS)
The number of PUs that should have been received after the latest USTAT was sent. In acknowledged mode, this state variable is updated at the end of each transmission time interval. It is incremented by the number of PUs that should have been received during the transmission time interval. If VR(EP) is equal to the number of requested PUs in the latest USTAT, then check if all PUs requested for retransmission have been received.

9.5. Timers

[All the section shall be reviewed when the protocol is defined]

a) Timer_STAT
It is used to detect the loss of the response from receiver side. This timer is set when transmitted AMD PDU requests status report (i.e. P bit is set to "1"). And it will be stopped when the transmitter receive Acknowledgement of the PUs in that AMD PDU by STAT PDU or Non Acknowledgment (Nack) by USTAT PDU. When this timer is over, the PUs of the oldest unconfirmed AMD PDU should be retransmitted with requesting status report, and this timer is set again. If polling is taken place during this timer is active the timer will be stopped and set again.
b) Timer_Prohibit
It is used to prohibit transmission of polling message within a certain period. If polling is taken place during this timer is active, it will be once stopped and set again. This timer will not be stopped by Ack or Nack. When this timer expires no action is performed.*[the values recommended for this timer are [FFS]]*
c) Timer_CC
Timer_CC protects the transmission of PU between connection establishment and connection release, during re-synchronization or during error recovery. Timer_CC indicates retransmission interval when confirmation isn't received against BGN PDU and ENDPDU. The value of Timer_CC should be a little larger than the round-trip delay.

d) Timer_QR (FFS)
Transmission interval of quick repeat for UMD PDU.
e) Timer_EPC (FFS)
This timer accounts for the roundtrip delay, i.e. the time when the first retransmitted PU should be received after a STAT/USTAT has been sent. The value of Timer_EPC is heavily based on the transmission time interval (corresponding to the Layer 1 interleaving depth). When changing the transmission time interval, then the value of the EPC timer also needs to be changed.

9.6.Protocol Parameters

[All the section shall be reviewed when the protocol is defined]

The value of each RLC protocol parameter is application specific and may be defined in another Recommendation which references this Recommendation.

a) MaxCC
Maximum value for the state variable VT(CC), corresponding to the maximum number of transmissions of a BGN, END.

b) MaxDAT
It is the maximum value for the number of retransmissions of a PU. This parameter is an upper limit of counter VT(DAT). When the value of VT(DAT) comes to MaxDAT, error recovery procedure will be performed.

c) MaxQR
Maximum successive transmission number of UMD PDU. This parameter is an upper limit for counter VT(QR).

d) MaxSTAT
Maximum number of list elements placed in a STAT PDU. When the number of list items exceeds MaxSTAT, the STAT message shall be segmented. All of the PDUs carrying the segmented STAT message, except possibly the last one, contain MaxSTAT list items. This parameter is not used by the receiver of a STAT PDU for length checking, but is only used by the sender of the STAT message for segmentation purposes. This parameter should be an odd integer greater than or equal to 3.

e) Credit
This parameter is used to coordinate credit notifications to layer management. When RLC is blocked from transmitting a new AMD PDU due to insufficient credit, “Credit” is assigned the value “No”. When RLC is permitted to transmit a new AMD PDU, “Credit” is assigned the value of “Yes”. Credit is initially assigned “Yes”.

9.7. Specific functions

[All the section shall be reviewed when the protocol is defined]
[The Hybrid ARQ (Type II/III) mechanism scheme is considered for the downlink only; in particular an incremental effort of protocol implementation will be followed. In this way it is possible to estimate the real performance of this scheme but also the impact that the introduction of such a scheme has on the protocol implementation. It is [FFS] if Hybrid ARQ mechanism for the downlink is mandatory for the UE (for both TDD and FDD)]

9.7.1. Retransmission Scheme

9.7.1.1. Basic Concept

1) Type of retransmission
   - Selective retransmission

2) Acknowledgement Confirmation
   - Receiver Status Report in response to the Transmitter Polling Request;
   - Unsolicited status report from the receiver caused by detecting the latest loss of PUs.

3) The Retransmission takes place when:
   - A Status Report, transmitted by the Receiver, is received;
   - An Unsolicited Status Report (USTAT) is received;
   - Retransmission timer expires.

4) Timing of polling
   Basically to confirm acknowledgement every RLC SDU can reduce extra overhead and improve the throughput. But if small RLC SDUs are given continuously, many status reports will be transmitted and it will cause overhead. To solve the problem, this retransmission scheme uses the timer which prohibits excessive polling. This timer only prohibits polling for every RLC SDU.

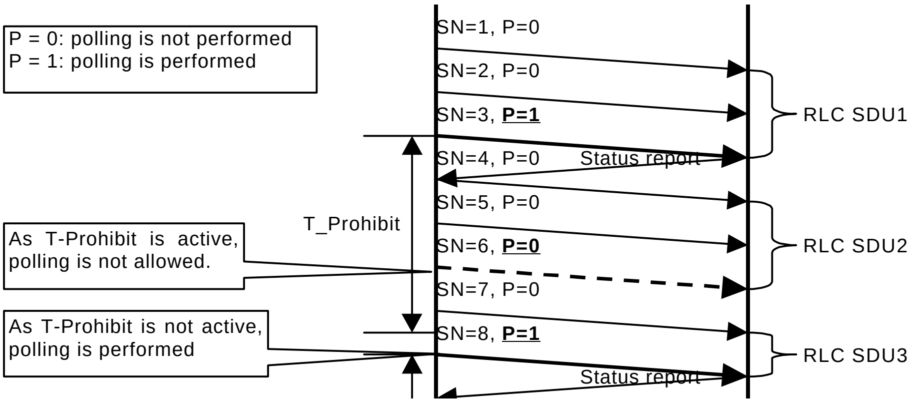

P = 0: polling is not performed
P = 1: polling is performed
SN=1, P=0
SN=2, P=0
SN=3, P=1
SN=4, P=0
Status report
SN=5, P=0
SN=6, P=0
SN=7, P=0
SN=8, P=1
Status report
RLC SDU1
RLC SDU2
RLC SDU3
T_Prohibit
As T-Prohibit is active, polling is not allowed.
As T-Prohibit is not active, polling is performed

Figure 9-15~~151511~~ : Reduction of excessive polling

9.7.1.2. Outline of proposed retransmission scheme

[In this section it is described the outline of the retransmission scheme presented. The list of the PDU and timers used in the retransmission scheme described are reported in this section, also if a complete description of them is presented in the proper sections of this specification. In chapter 12 are also reported the SDL diagrams of this scheme that actually are [FFS]]

9.7.1.3. PDUs used for retransmission

Following PDUs are necessary for this retransmission scheme.

- AMD PDU:

Transfers user data and requests status report by setting Poll bit.

- STAT PDU:

Reports the status of receiver to transmitter when AMD PDU with status report request is received.

- USTAT PDU:

The USTAT PDU is transmitted upon detection of an erroneous transmission of one or more data PUs. It is used to inform the transmitter side about missing PUs at the receiver RLC.

9.7.1.4. Timers used for retransmission

Following two timers are necessary for this retransmission scheme.

- Timer_STAT:

This timer is set when AMD PDU with polling (i.e. P bit is set to “1”) is transmitted. And it will be stopped when the transmitter receives Ack or Nack for the AMD PDU with polling. If polling is taken place during this timer is active, it will be once stopped and set again.

- Timer_Prohibit:

This timer is set when AMD PDU with polling is transmitted. If polling is taken place during this timer is active, it will be once stopped and set again. This timer will not be stopped by Ack or Nack. When this timer expires no action is performed.

9.7.1.5. Trigger of Polling

Polling message is transmitted when;the last segment of AMD SDU is transmitted (every RLC SDU) and Timer_Prohibit is not active
the retransmission timer (Timer_STAT) expires
the last PU in the transmission queue is transmitted
the transmitter window has to move
the last PU among those which are requested to be retransmitted by a STAT is transmitted.

9.7.1.6. Flow of retransmission

Case1) In case STAT is received:

- Polling is performed from the transmitter comply with the polling trigger.
- At this moment, the transmitter sets Timer_STAT and Timer_Prohibit.
- The receiver transmits STAT which requests retransmission for SN = 2, 3, 4 of PU in response to the polling.
- The transmitter receives the STAT and stops Timer_STAT. Then the PUs (SN = 2, 3, 4) requested by the STAT are retransmitted.
- When the PU whose SN =4 is transmitted, poll bit is set to 1 (polling is performed).
- At this moment, Timer_STAT is set and Timer_Prohibit is reset.
- When the transmitter receives STAT from the receiver, Timer_STAT is stopped.

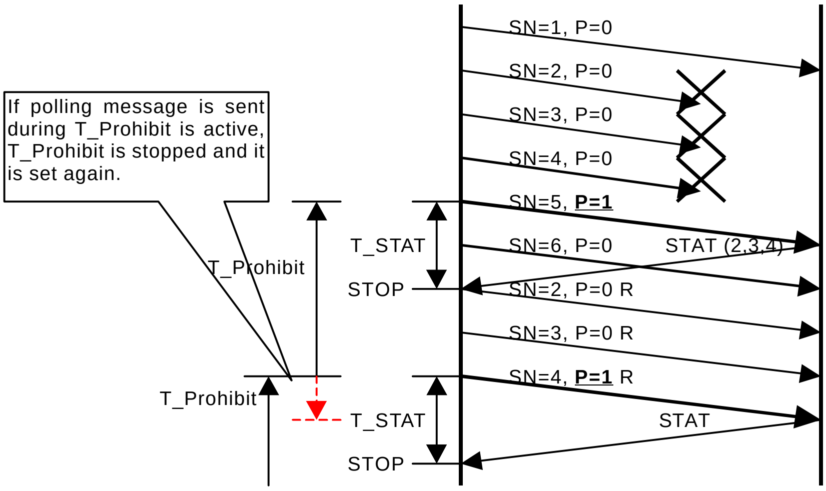


Figure 9-16~~161612~~ : Retransmission Scheme Behaviour when STAT is received

- Case2) In case USTAT is received:
- If the receiver detects new loss of PUs, it transmits USTAT which requests retransmission of SN = 2, 3 of PU to the transmitter.
- The transmitter receives the USTAT and retransmits the requested PUs (SN = 2,3).
- In this case polling is not performed when the PU whose SN =3 is retransmitted.

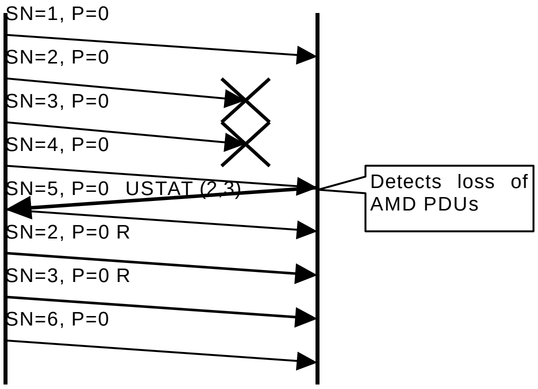


Figure 9-17~~171713~~ : Retransmission Scheme Behaviour USTAT is received

Case3) In case Timer_STAT expires:

- If Timer_STAT expires due to loss of the AMD PDU with polling or STAT, the transmitter retransmits the AMD PDU with polling.
- At this moment, Timer_STAT is set and Timer_Prohibit is reset.
- When the transmitter receives STAT from the receiver, Timer_STAT is stopped.

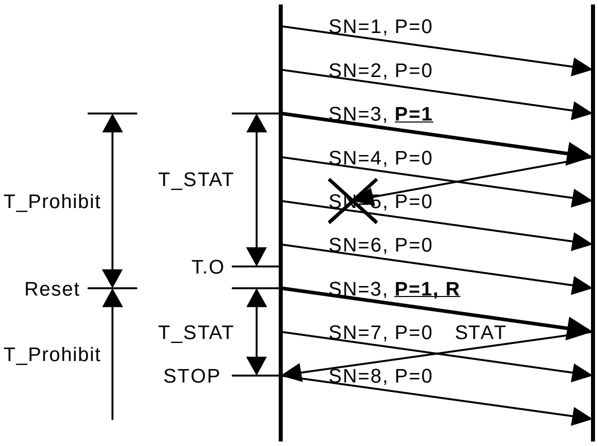


Figure 9-18~~181814~~ : Retransmission Scheme Behaviour when Timer_STAT expires

[The Timer T-Prohibit is related only to the UE; there's no need for "T_Prohibit Timer" in the network as it is an implementation aspect. The values for T_Prohibit_Timer are [F.F.S]]

9.7.2. Credit and peer-to-peer flow control

Credit is granted by the RLC receiver to allow the peer RLC transmitter to transmit new AMD PDUs. The process by which a receiver entity determines credit is not subject to standardization, but is related to the buffer availability and the bandwidth/delay of the connection**.**

Details of the usage of Crediting is FFS.

9 9.7.3 Local flow control

RLC events, such as reception of PDUs and external and internal signals, are normally processed in the order in which they occurred. However, events pertaining to the exchange of RLC link status information have priority over data transfer.

An implementation may detect congestion (for example, a long queuing delay) in its lower protocol layers. If so, data transfer should be temporarily suspended in order to give priority to connection control messages. The means by which an RLC entity decides whether or not it is congested depends on the protocol environment, including protocol timer values, and is not subject to standardization.

If a RLC entity detects local congestion (“lower layer busy” in the SDL specification), it can elect to suspend the servicing of RLC-AM-DATA.request, RLC-UM-DATA.request It can also suspend the retransmission of requested AMD PDUs. The data transfer procedures allow this to occur without causing protocol errors.

Therefore, in terms of transmitting PDUs to the peer receiver, all types of PDUs except AMD PDU and UMD PDU are given highest priority. The AMD PDUs and UMD PDUs have equal priority. Among the AMD PDUs, retransmission have priority over new transmission if both types are pending. These priorities are only internal to RLC.

9.8. RLC Toolbox concept

The RLC toolbox concept specifies a number of basic functions. These functions can then be combined in different ways in order to get a complete and functional protocol. How to combine the different functions is signalled by RRC, before setting up a new RLC entity. The toobox concept may be applied to both transparent and non-transparent mode entities.

9.8.1. Toolbox concept for acknowledged mode RLC entities.

This section describes the different functions included in the RLC toolbox for acknowledged mode. This section concentrates on functions for ARQ mechanism, but other function can be added to the toolbox. The functions have been divided into two groups, transmitting side functions and receiving side functions. It is for FFS if all functions have to be supported by the UE. The presence field rather indicates if the function is always supported by the acknowledged mode entity or wheter its implementation is network controlled. No explicit signalling is needed if the function is always supported.

9.8.1.1. Transmitting side functions

9.8.1.1.1. When to poll

It is optional to apply a polling mechanism on transmitter side. If a polling mechanism is applied, Table 9-2~~Table 9-2~~ below summerizes the functions that control when the transmitter should poll the peer entity for a status report.

Table 9-2 List of functions that control when to poll the receiver for a status report.

Trigger	*Presence*
Last PU in buffer.	Always
Poll timer.	Always
Every X PU.	Network controlled
Every X SDU.	Network controlled
Last PU in retransmission buffer.	Network controlled
X% of transmission window.	Network controlled
Timer based.	Network controlled
$T_{prohibit}$	Network controlled

[The list of parameters shall be reviewed during the evolution of the prodocol definition]

- **Last PU in buffer**

The transmitting side polls the peer entity for a status report, when the last PDU in the transmission buffer is transmitted. This function is mandatory for the transmitting side, if polling should be applied.

- **Poll timer**

The poll timer is started when a poll is transmitted to the peer entity and if no status report has been received before the poll timer expires a new poll is transmitted to the receiver. The value of the timer is signalled by RRC. This function is mandatory for the transmitting side, if polling should be applied.

- **Every X PU**

The transmitting side polls the peer entity for a status report every X PU. The value of X is signalled by RRC. This function is optional for the transmitting side.

- **Every X SDU**

The transmitting side polls the peer entity for a status report every X SDU. The value of X is signalled by RRC. This function is optional for the transmitting side.

- **Last PU in retransmission buffer**

The transmitting side polls the peer entity for a status report at transmission of the last PDU in the retransmission buffer.

- **X% of transmission window**

The transmitting side polls the peer entity for a status report when it has reached X % of the transmission window. The value of X is signalled by RRC. This function is optional for the transmitting side.

- **Timer based**

The transmitting side polls the peer entity for a status report periodically. The value of the time period is signalled by RRC. The function is optional for the transmitting side.

- **$T_{prohibit}$**

This function controls how often the transmitting side is allowed to poll the peer entity. The $T_{prohibit}$ is started when a poll is transmitted to the peer entity. As long as the timer is running the transmitting side is not allowed to poll the peer entity. The value of the timer is signalled by RRC. This function is optional for the transmitting side.

9.8.1.1.2. How to react upon a status report

Table 9-3~~Table 9-3~~ below summerizes the functions that control how to react upon a status report.

Table 9-3~~33~~ List of functions that control how to react upon a status report.

Trigger	*Presence*
Adjust transmission window.	Always

Retransmit PUs.	Always
Plausibility check.	Network controlled

- **Adjust transmission window**

The transmission window should be updated according to the received status report. It is mandatory for the transmitting side to support this function.

- **Retransmit AM PUs**

This function retransmits the AM PUs that are requested by the status report. If no plausibility check is applied, the PUs shall be retransmitted immediately and have higher priority than new AM PUs. This function is mandatory for the transmitting side.

- **Plausibility check**

This function checks whether the contents of a status report is reasonable or not. It can prohibit or delay retransmissions requested by a status report. For example, the status report could contain negative acknowledgements of PUs which may not had arrived at the receiver before the status report was transmitted. The transmitter should not retransmit these PUs. This function is optional for the transmitting side.

9.8.1.2. Receiving side functions

9.8.1.2.1. How to react upon a poll

The receiving side should send a status report if it receives a poll. It should send the status report immediately. This function is mandatory for the receiving side.

9.8.1.2.2. When to send a status report

Table 9-4~~Table 9-4~~ below summerizes the functions that control when to send a status report.

Table 9-444 List of functions that control when to send a status report.

Trigger	*Presence*
Reception of poll.	Always
EPC	Network controlled
Detection of missing PU(s).	Network controlled
Every X SDU.	Network controlled
Every X PU.	Network controlled
X% of receiving window.	Network controlled
Timer based.	Network controlled
$T_{prohibit}$	Network controlled

- **Reception of poll**

The receiving side sends a status report to the peer entity upon reception of a poll, see section 9.8.1.2.1. The status report should be transmitted immediately. This function is mandatory for the receiving side.

- **Detection of missing PU(s)**

The receiving side sends a status report to the peer entity upon detection of missing AM PU(s). The status report should be transmitted immediately. This function is optional for the receiving side.

- **Every X SDU**

The receiving side sends a status report to its peer entity every X SDU. The value of X is signalled by RRC. This function is optional for the receiving side.

- **Every X PU**

The receiving side sends a status report to its peer entity every X PU. The value of X is signalled by RRC. This function is optional for the receiving side.

- **X% of receiving window**

The receiving side sends a status report when X % of the transmission window has been reached. The value of X is signalled by RRC. This function is optional for the receiving side.

- **Timer based**

The receiving side sends a status report periodically to the peer entity. The value of the time period is signalled by RRC. The function is optional for the receiving side.

- **$T_{prohibit}$**

This function controls how often the receiving side is allowed to send status reports the peer entity. The $T_{prohibit}$ is started when a status report is transmitted to the peer entity. As long as the timer is running the receiving side is not allowed to send a status reports to the peer entity. The value of the timer is signalled by RRC. This function is optional for the receiving side.

10. Handling of unknown, unforeseen and erroneous protocol data

11. Elementary procedures

(Examples: idle, data transfer, RLC connection setup, RLC connection release, re-synchronisation)

12. SDL diagrams

[All the section shall be reviewed when the protocol is defined;
all the SDL diagrams presented are [FFS]]

The resultant SDL diagrams (Timer_Prohibit scheme) are followed is shown in ANNEX 1.
Estimated PDU Counter (EPC) scheme (receiving side) (FFS)

1. Send a status report (USTAT), requesting for the retransmission of K number of missing PDUs.
2. Start Timer_EPC. This timer accounts for the roundtrip delay, i.e. the time when the first retransmitted PDU should be received.
3. When the timer expires, start counting the received PDUs, or rather the PDUs that should have been received using the state variable VT(EP)
4. If VT(EP) = K, then check if all PDUs (requested in the status report in step 1) have been received.
   a) If some of the previously missing PDUs are still missing, then repeat the procedure from step 1 for the PDUs that are still missing.
   b)If none of the previously missing PDUs are still missing, then no status report needs to be sent, unless a poll had been transmitted or a new missing PDU has been detected. In case of a poll or a new missing PDU, then repeat the procedure from step 1.

Every poll received during the time when the Timer_EPC is active and VT(EP) < K will be discarded by the receiving side, i.e. neither STATs nor USTATs will be sent from the receiving side during this time.

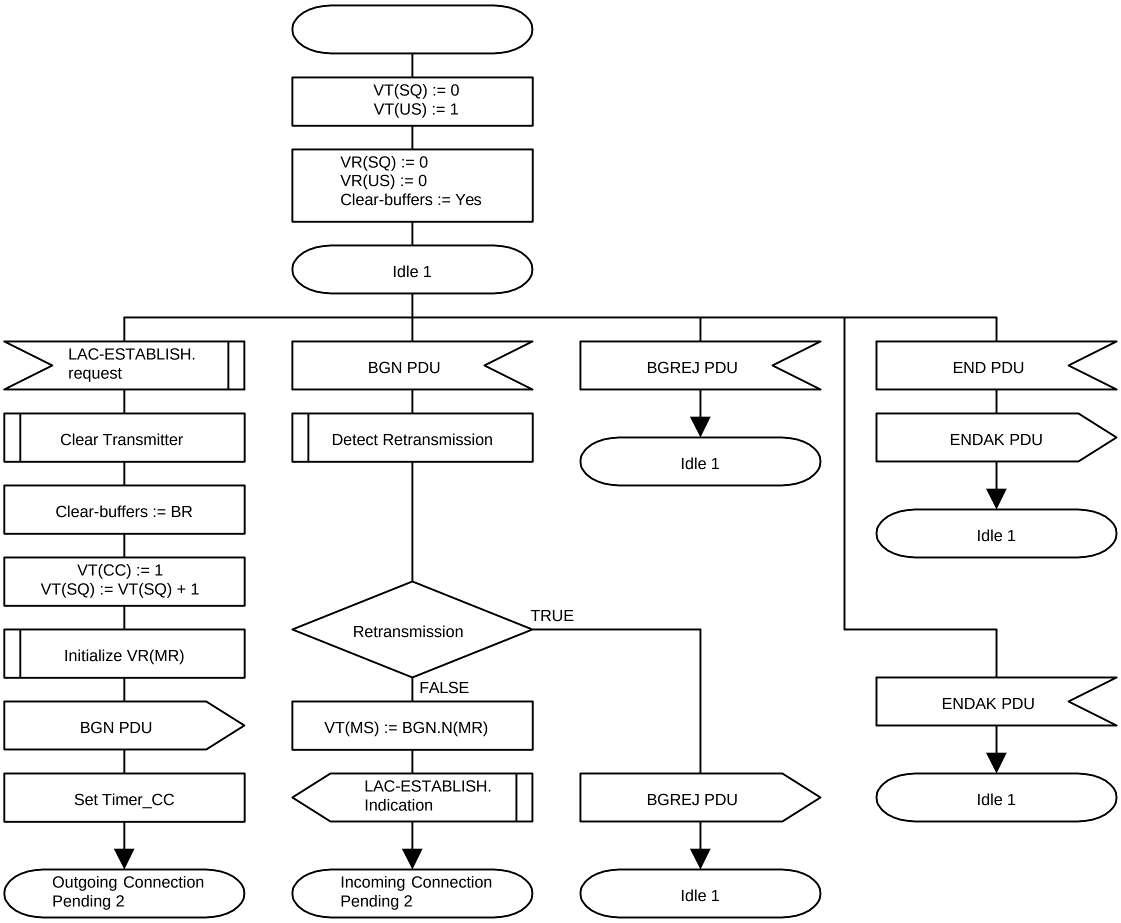


Figure 12-1

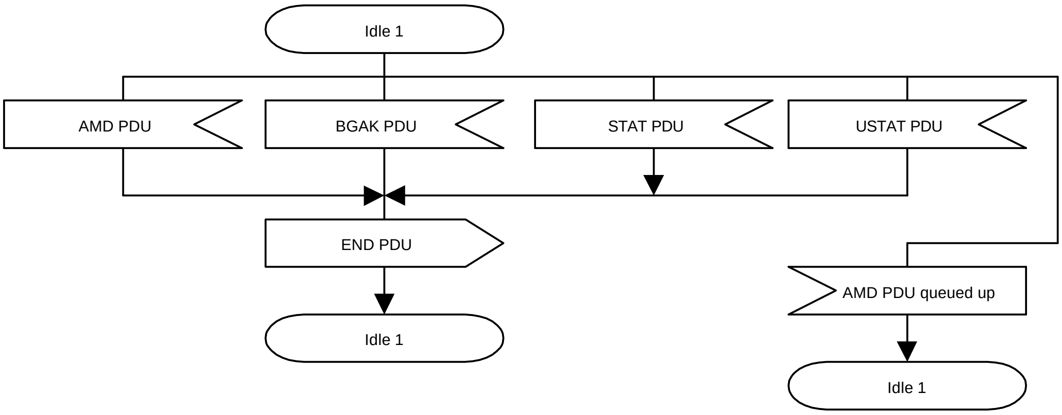


Figure 12-2

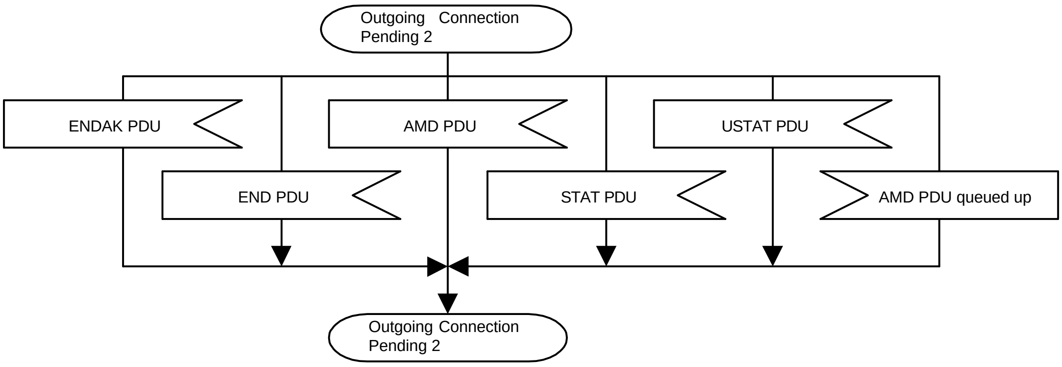


Figure 12-3

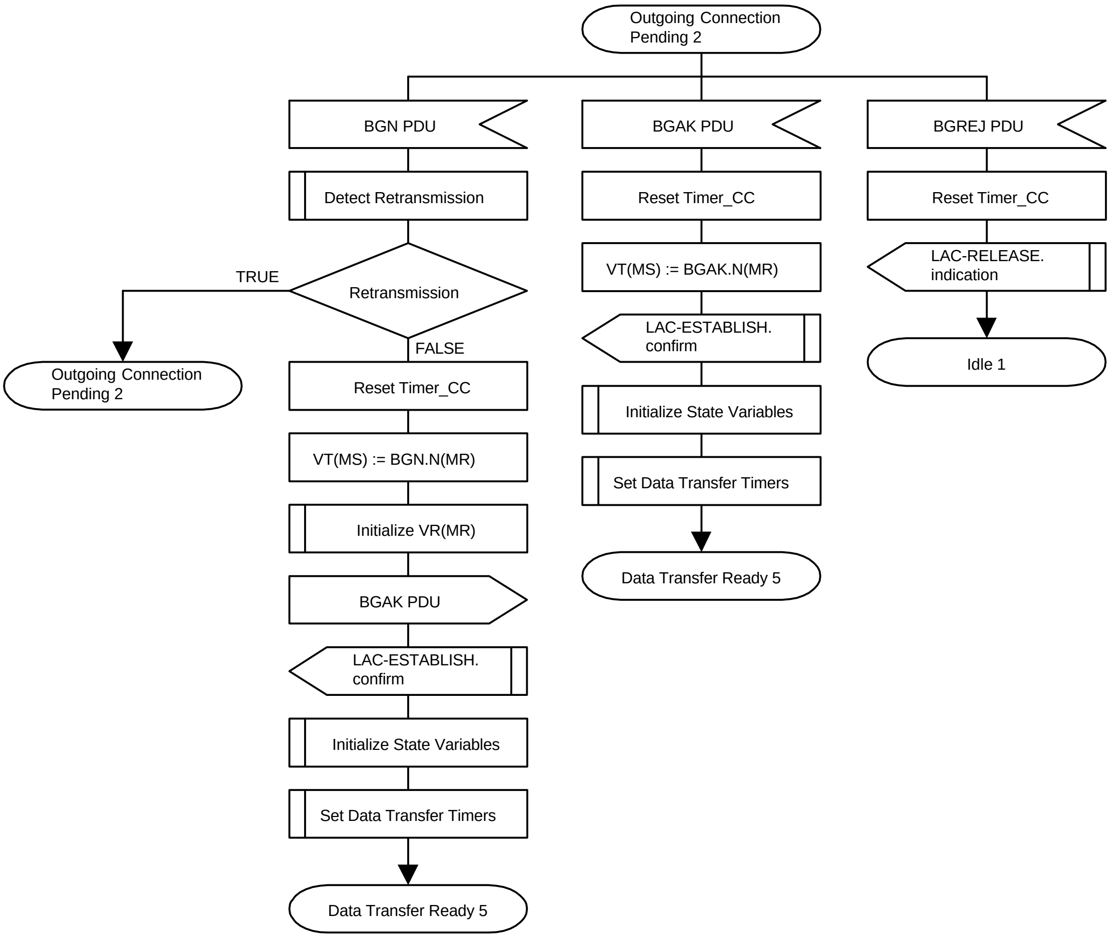


Figure 12-4

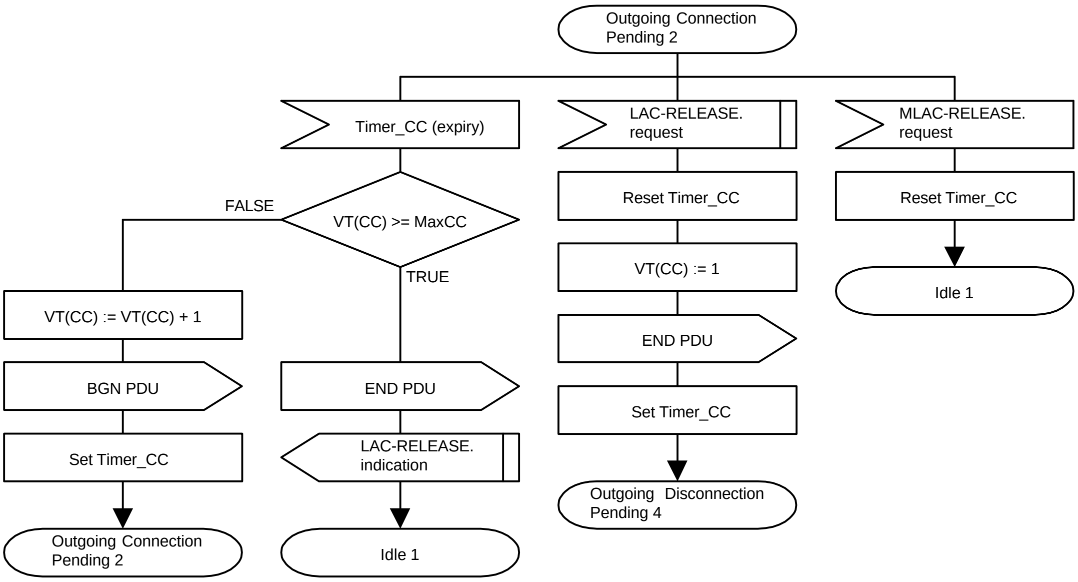


Figure 12-5

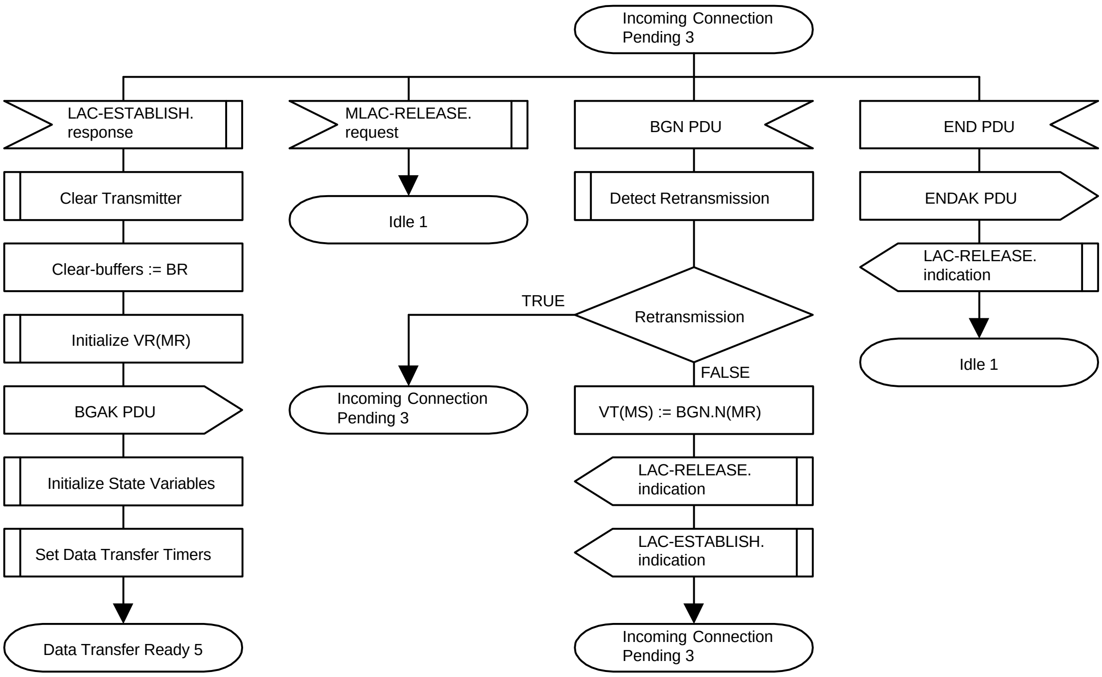


Figure 12-6

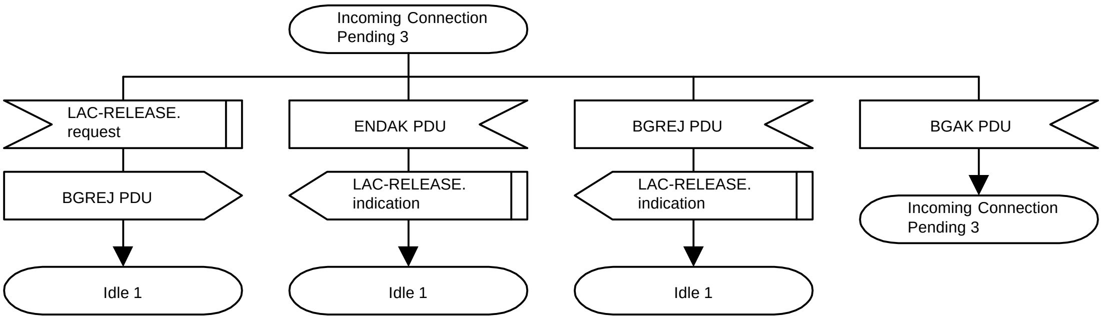


Figure 12-7

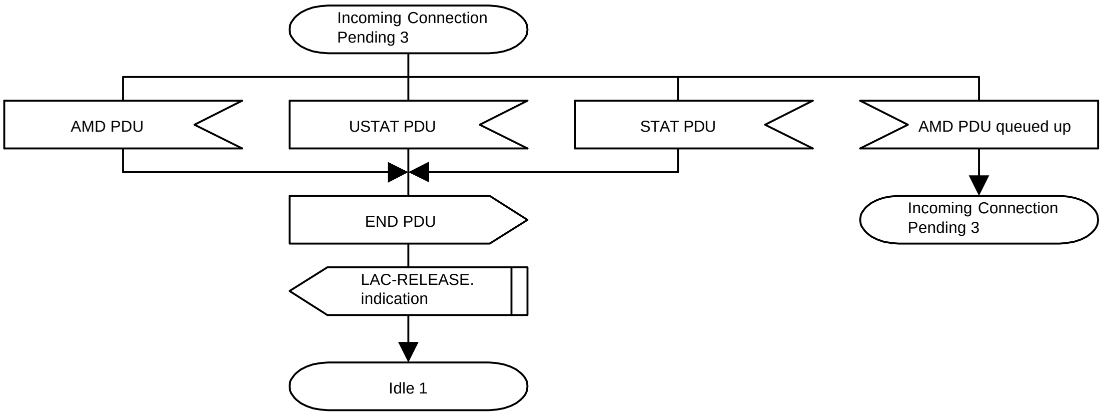


Figure 12-8

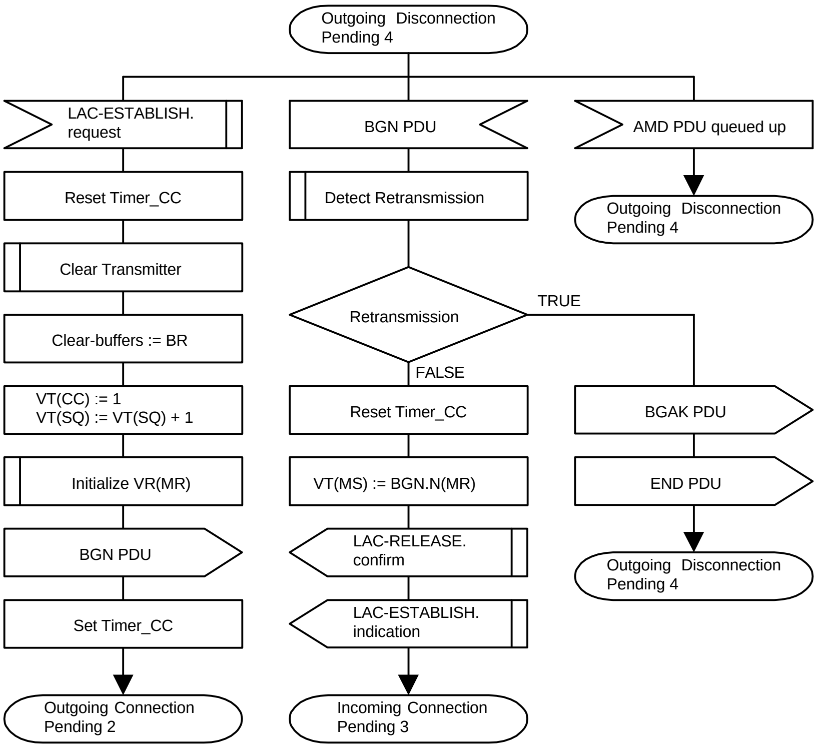


Figure 12-9

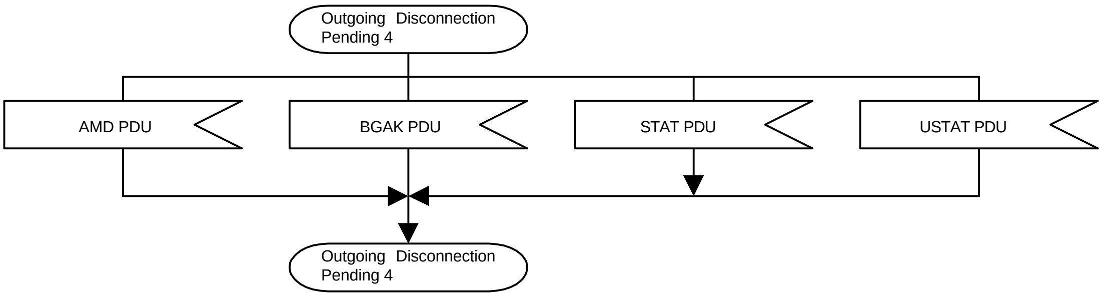

Outgoing Disconnection Pending 4
AMD PDU
BGAK PDU
STAT PDU
USTAT PDU
Outgoing Disconnection Pending 4

Figure 12-10

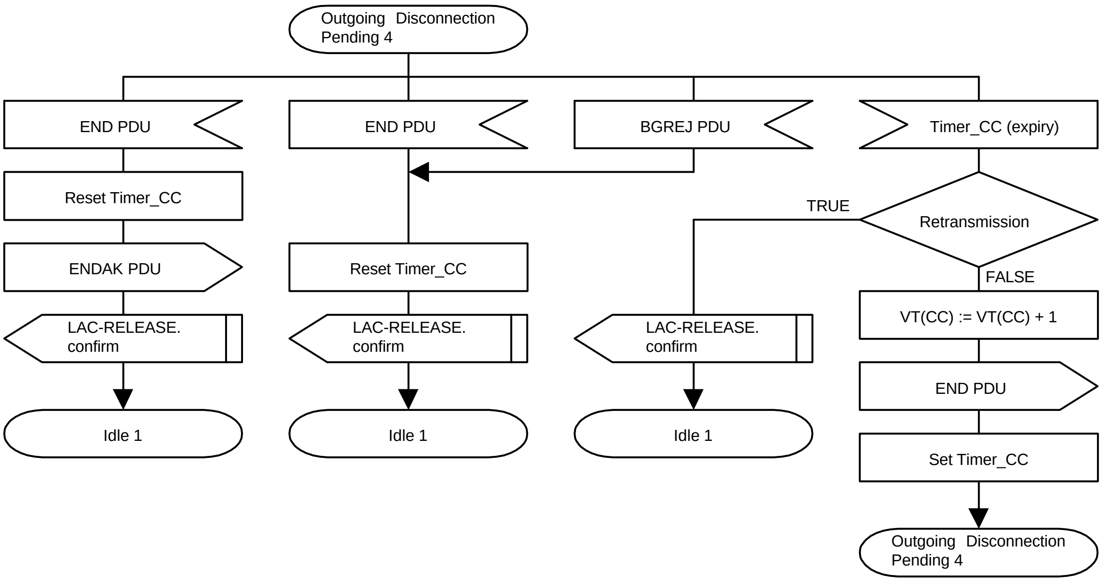


Figure 12-11

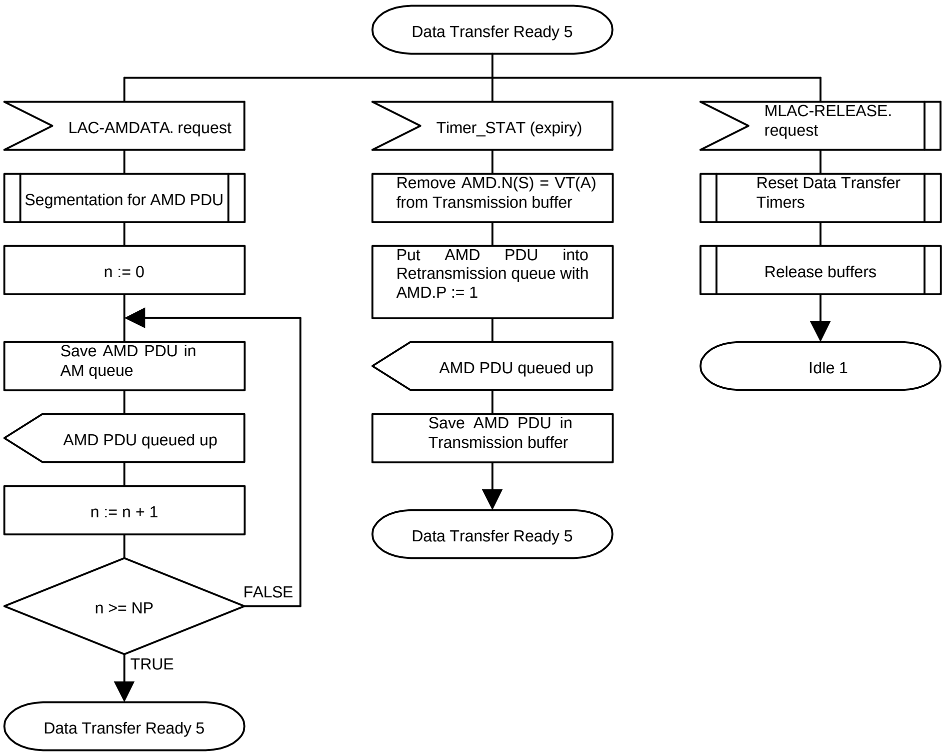


Figure 12-12

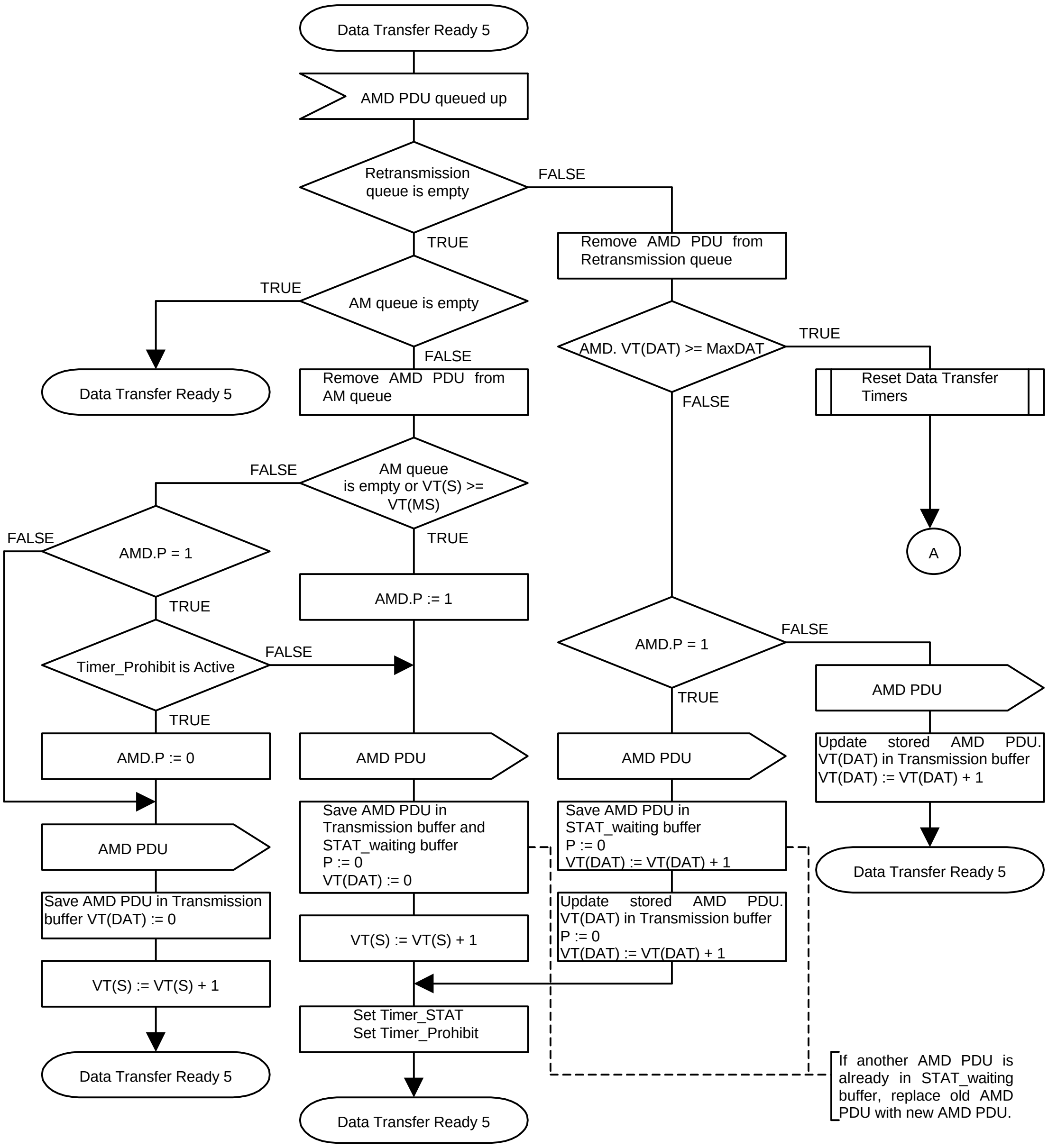

Data Transfer Ready 5
AMD PDU queued up
Retransmission queue is empty
FALSE
TRUE
Remove AMD PDU from Retransmission queue
TRUE
AM queue is empty
FALSE
AMD. VT(DAT) >= MaxDAT
TRUE
Data Transfer Ready 5
Remove AMD PDU from AM queue
FALSE
Reset Data Transfer Timers
FALSE
AM queue is empty or VT(S) >= VT(MS)
TRUE
FALSE
AMD.P = 1
A
TRUE
AMD.P := 1
FALSE
AMD.P = 1
Timer_Prohibit is Active
FALSE
AMD PDU
TRUE
TRUE
AMD.P := 0
AMD PDU
AMD PDU
Update stored AMD PDU. VT(DAT) in Transmission buffer VT(DAT) := VT(DAT) + 1
AMD PDU
Save AMD PDU in Transmission buffer and STAT_waiting buffer P := 0 VT(DAT) := 0
Save AMD PDU in STAT_waiting buffer P := 0 VT(DAT) := VT(DAT) + 1
Data Transfer Ready 5
Save AMD PDU in Transmission buffer VT(DAT) := 0
Update stored AMD PDU. VT(DAT) in Transmission buffer P := 0 VT(DAT) := VT(DAT) + 1
VT(S) := VT(S) + 1
VT(S) := VT(S) + 1
Set Timer_STAT Set Timer_Prohibit
Data Transfer Ready 5
If another AMD PDU is already in STAT_waiting buffer, replace old AMD PDU with new AMD PDU.
Data Transfer Ready 5

Figure 12-13

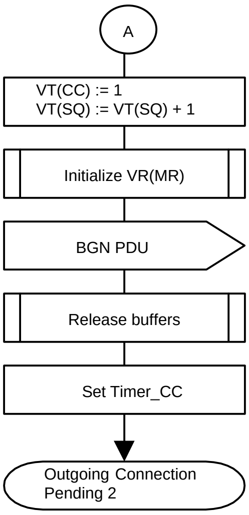

A
VT(CC) := 1
VT(SQ) := VT(SQ) + 1
Initialize VR(MR)
BGN PDU
Release buffers
Set Timer_CC
Outgoing Connection Pending 2

Figure 12-14

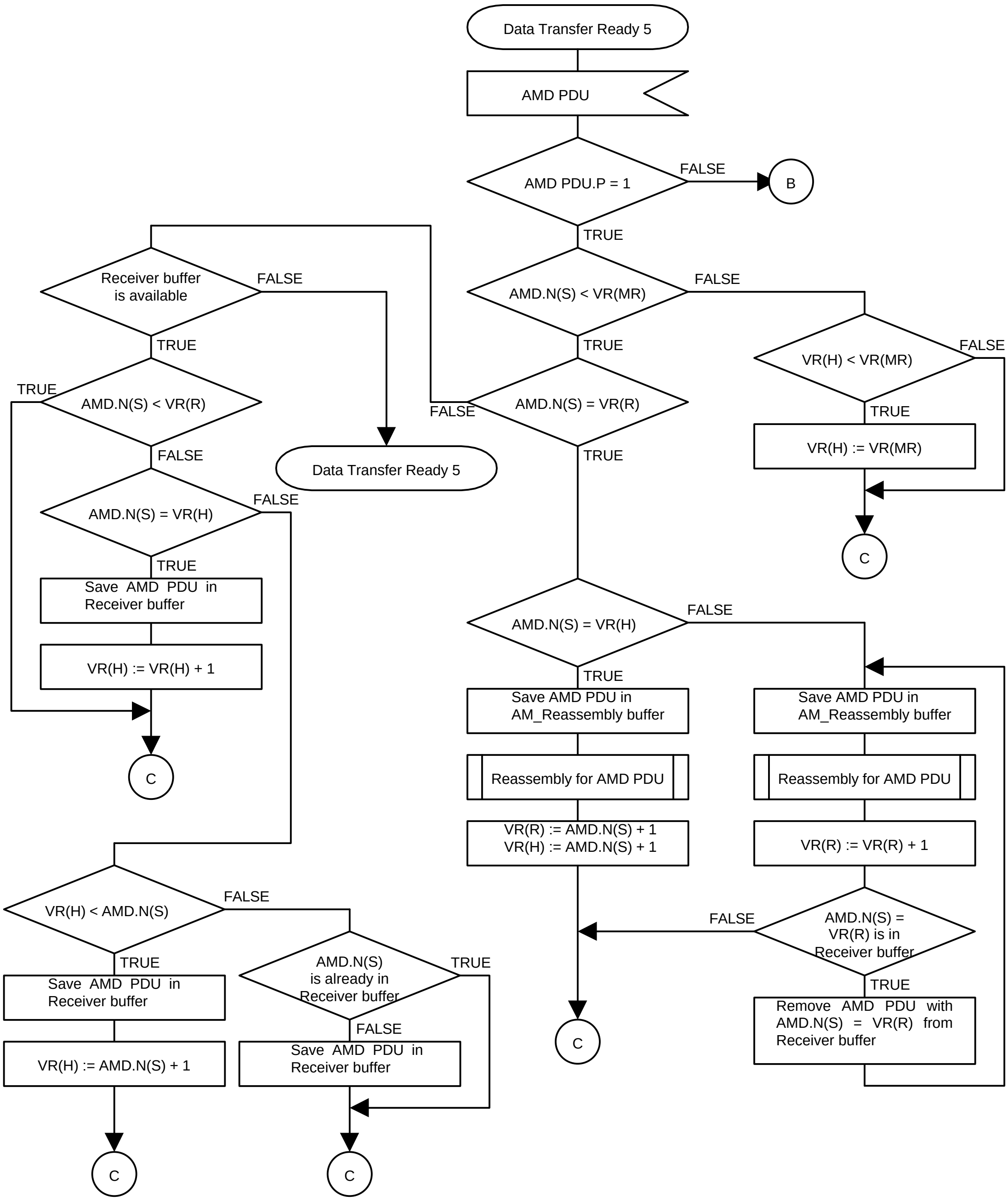

Data Transfer Ready 5
AMD PDU
AMD PDU.P = 1
FALSE
B
TRUE
AMD.N(S) < VR(MR)
FALSE
VR(H) < VR(MR)
FALSE
TRUE
VR(H) := VR(MR)
C
TRUE
AMD.N(S) = VR(R)
FALSE
Receiver buffer is available
FALSE
Data Transfer Ready 5
TRUE
AMD.N(S) < VR(R)
TRUE
FALSE
AMD.N(S) = VR(H)
FALSE
TRUE
Save AMD PDU in Receiver buffer
VR(H) := VR(H) + 1
C
TRUE
AMD.N(S) = VR(H)
FALSE
TRUE
Save AMD PDU in AM_Reassembly buffer
Reassembly for AMD PDU
VR(R) := AMD.N(S) + 1
VR(H) := AMD.N(S) + 1
C
Save AMD PDU in AM_Reassembly buffer
Reassembly for AMD PDU
VR(R) := VR(R) + 1
AMD.N(S) = VR(R) is in Receiver buffer
FALSE
TRUE
Remove AMD PDU with AMD.N(S) = VR(R) from Receiver buffer
VR(H) < AMD.N(S)
FALSE
TRUE
Save AMD PDU in Receiver buffer
VR(H) := AMD.N(S) + 1
C
AMD.N(S) is already in Receiver buffer
TRUE
FALSE
Save AMD PDU in Receiver buffer
C

Figure 12-15

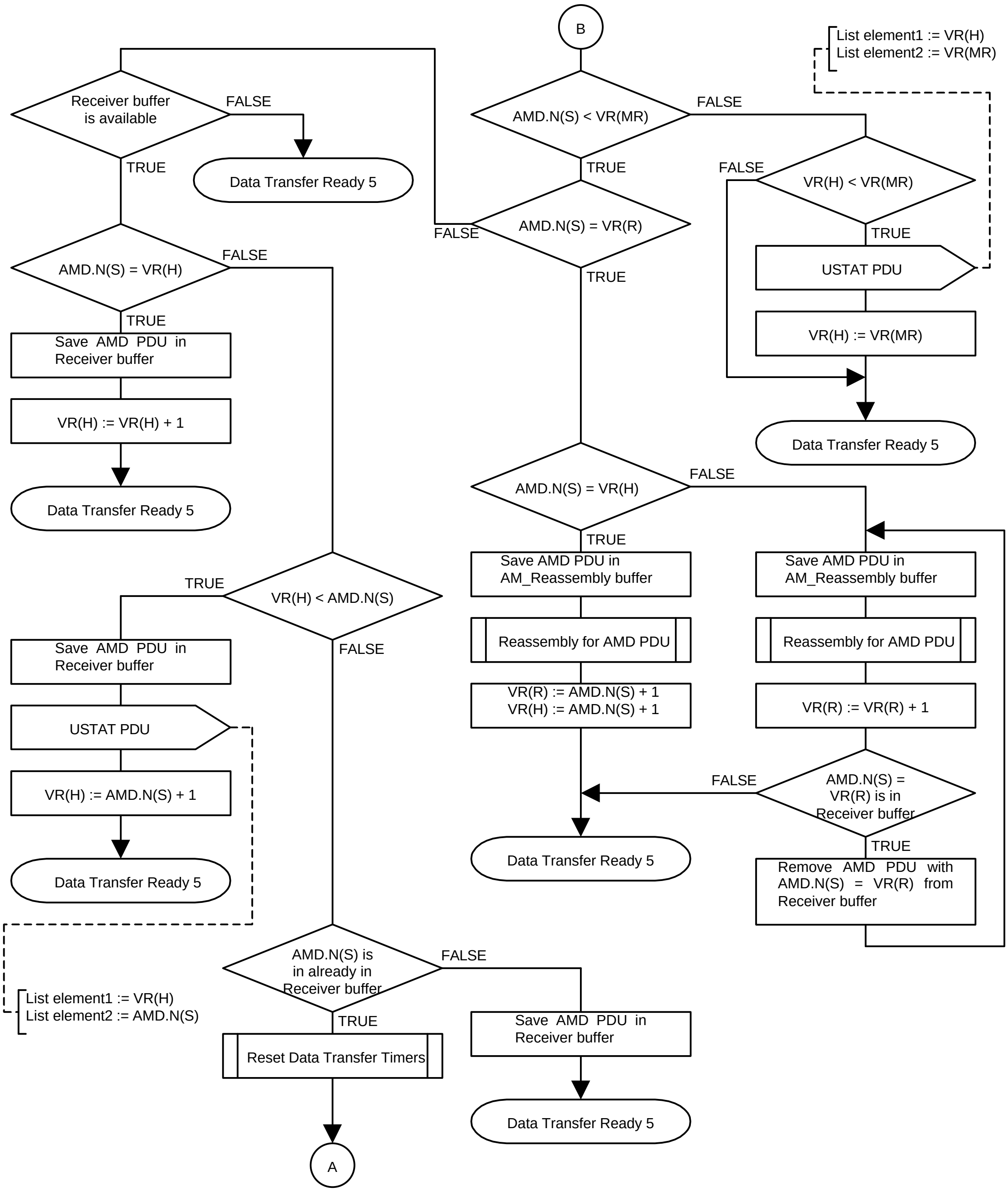

B
List element1 := VR(H)
List element2 := VR(MR)
Receiver buffer is available
FALSE
Data Transfer Ready 5
TRUE
AMD.N(S) < VR(MR)
FALSE
TRUE
FALSE
VR(H) < VR(MR)
AMD.N(S) = VR(R)
FALSE
TRUE
USTAT PDU
TRUE
AMD.N(S) = VR(H)
FALSE
TRUE
Save AMD PDU in Receiver buffer
VR(H) := VR(MR)
VR(H) := VR(H) + 1
Data Transfer Ready 5
Data Transfer Ready 5
AMD.N(S) = VR(H)
FALSE
TRUE
TRUE
VR(H) < AMD.N(S)
Save AMD PDU in AM_Reassembly buffer
Save AMD PDU in AM_Reassembly buffer
FALSE
Save AMD PDU in Receiver buffer
Reassembly for AMD PDU
Reassembly for AMD PDU
VR(R) := AMD.N(S) + 1
VR(H) := AMD.N(S) + 1
VR(R) := VR(R) + 1
USTAT PDU
VR(H) := AMD.N(S) + 1
FALSE
AMD.N(S) = VR(R) is in Receiver buffer
TRUE
Data Transfer Ready 5
Data Transfer Ready 5
Remove AMD PDU with AMD.N(S) = VR(R) from Receiver buffer
AMD.N(S) is in already in Receiver buffer
FALSE
List element1 := VR(H)
List element2 := AMD.N(S)
TRUE
Save AMD PDU in Receiver buffer
Reset Data Transfer Timers
Data Transfer Ready 5
A

Figure 12-16

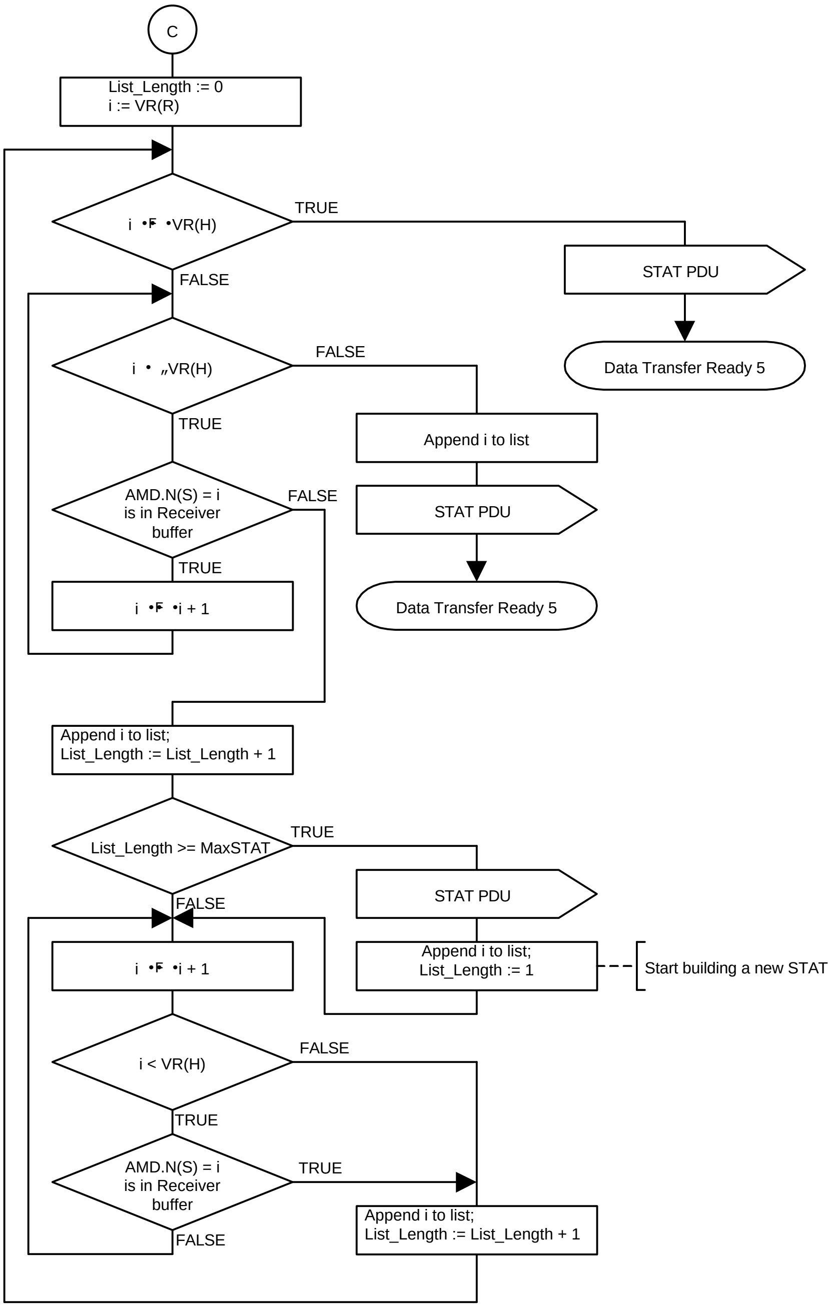

C
List_Length := 0
i := VR(R)
i •F •VR(H)
TRUE
FALSE
STAT PDU
Data Transfer Ready 5
i • „VR(H)
FALSE
TRUE
Append i to list
AMD.N(S) = i
is in Receiver
buffer
FALSE
TRUE
STAT PDU
i •F •i + 1
Data Transfer Ready 5
Append i to list;
List_Length := List_Length + 1
List_Length >= MaxSTAT
TRUE
FALSE
STAT PDU
i •F •i + 1
Append i to list;
List_Length := 1
Start building a new STAT
i < VR(H)
FALSE
TRUE
AMD.N(S) = i
is in Receiver
buffer
TRUE
FALSE
Append i to list;
List_Length := List_Length + 1

Figure 12-17

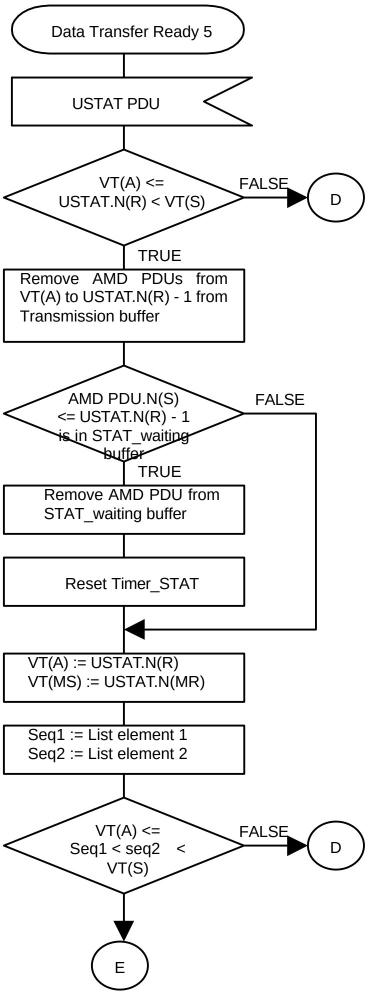

Data Transfer Ready 5
USTAT PDU
VT(A) <=
USTAT.N(R) < VT(S)
FALSE
D
TRUE
Remove AMD PDUs from VT(A) to USTAT.N(R) - 1 from Transmission buffer
AMD PDU.N(S) <= USTAT.N(R) - 1 is in STAT_waiting buffer
FALSE
TRUE
Remove AMD PDU from STAT_waiting buffer
Reset Timer_STAT
VT(A) := USTAT.N(R)
VT(MS) := USTAT.N(MR)
Seq1 := List element 1
Seq2 := List element 2
VT(A) <=
Seq1 < seq2 <
VT(S)
FALSE
D
E

Figure 12-18

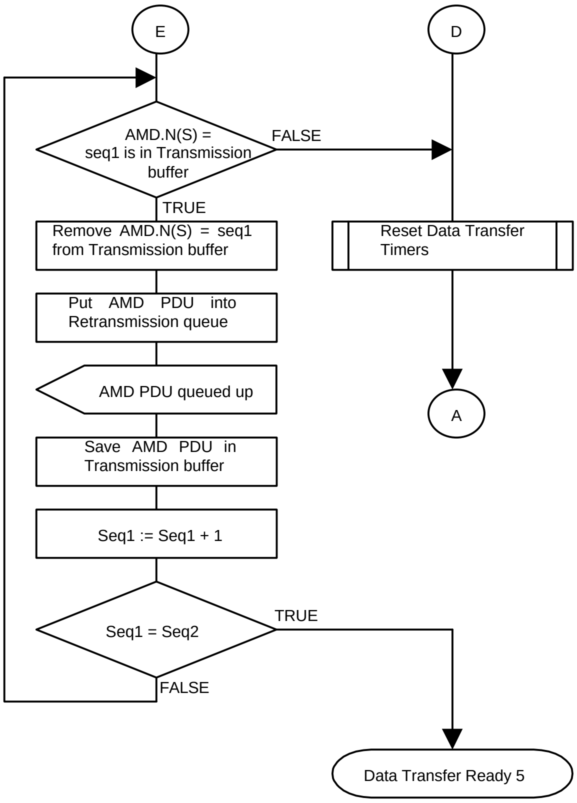

E
D
AMD.N(S) = seq1 is in Transmission buffer
FALSE
TRUE
Remove AMD.N(S) = seq1 from Transmission buffer
Reset Data Transfer Timers
Put AMD PDU into Retransmission queue
AMD PDU queued up
A
Save AMD PDU in Transmission buffer
Seq1 := Seq1 + 1
Seq1 = Seq2
TRUE
FALSE
Data Transfer Ready 5

Figure 12-19

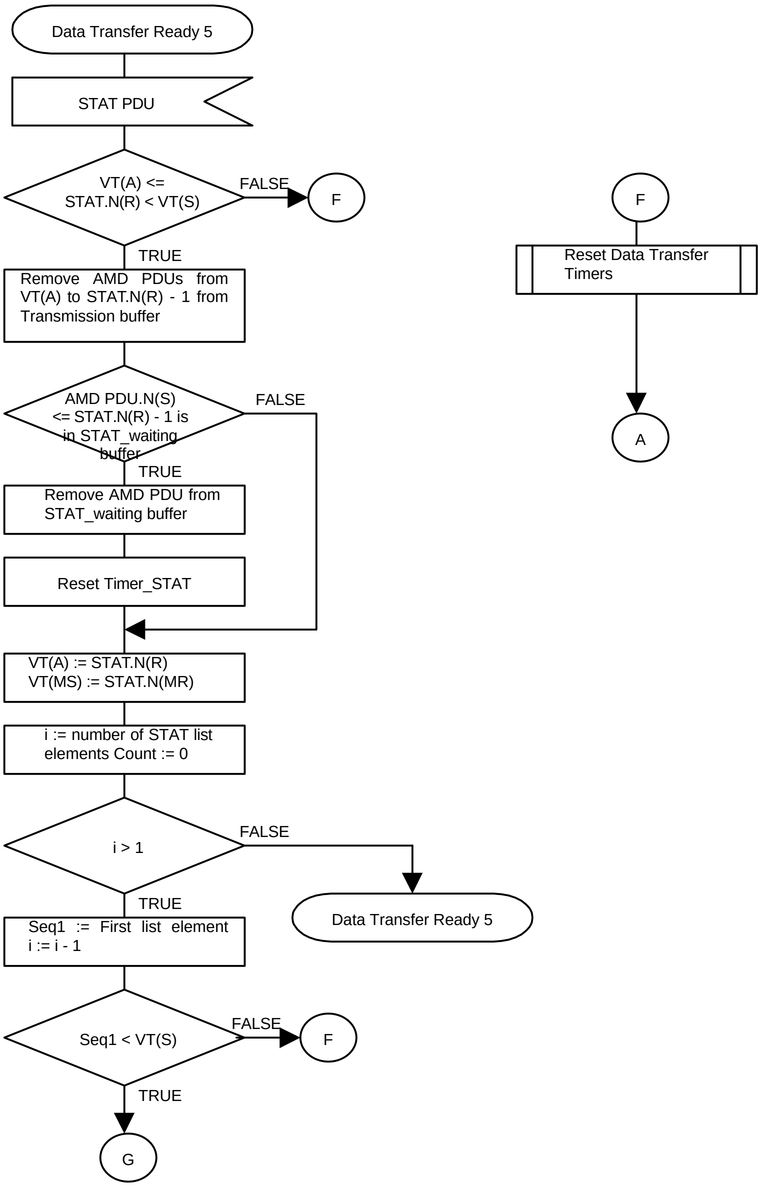

Data Transfer Ready 5
STAT PDU
VT(A) <=
STAT.N(R) < VT(S)
FALSE
F
TRUE
Remove AMD PDUs from VT(A) to STAT.N(R) - 1 from Transmission buffer
AMD PDU.N(S)
<= STAT.N(R) - 1 is
in STAT_waiting
buffer
FALSE
TRUE
Remove AMD PDU from STAT_waiting buffer
Reset Timer_STAT
VT(A) := STAT.N(R)
VT(MS) := STAT.N(MR)
i := number of STAT list elements Count := 0
i > 1
FALSE
TRUE
Data Transfer Ready 5
Seq1 := First list element
i := i - 1
Seq1 < VT(S)
FALSE
F
TRUE
G
F
Reset Data Transfer Timers
A

Figure 12-20

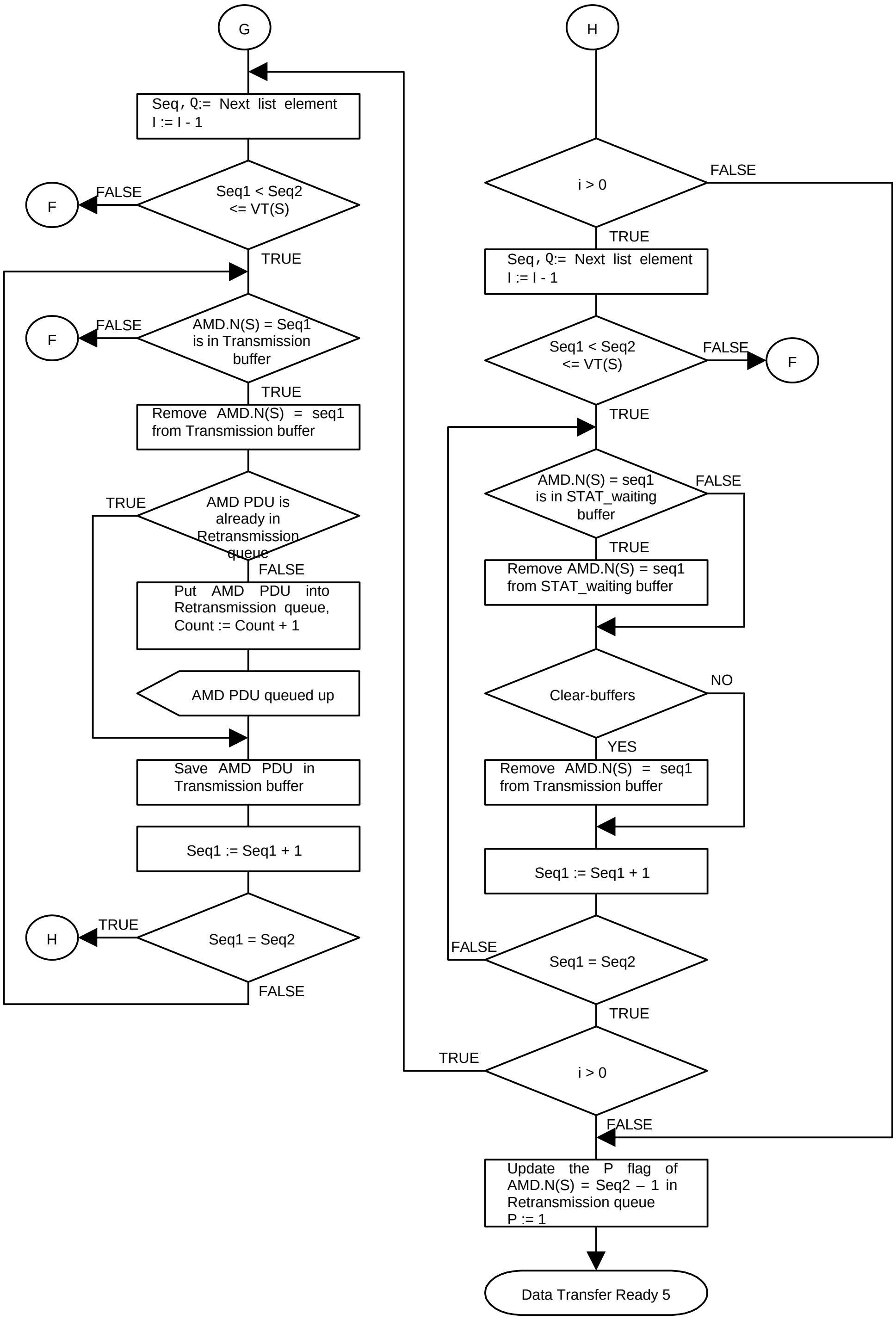

G
Seq，Q:= Next list element
I := I - 1
Seq1 < Seq2
<= VT(S)
FALSE
F
TRUE
AMD.N(S) = Seq1
is in Transmission
buffer
FALSE
F
TRUE
Remove AMD.N(S) = seq1
from Transmission buffer
AMD PDU is
already in
Retransmission
queue
TRUE
FALSE
Put AMD PDU into
Retransmission queue,
Count := Count + 1
AMD PDU queued up
Save AMD PDU in
Transmission buffer
Seq1 := Seq1 + 1
Seq1 = Seq2
TRUE
H
FALSE
H
i > 0
FALSE
TRUE
Seq，Q:= Next list element
I := I - 1
Seq1 < Seq2
<= VT(S)
FALSE
F
TRUE
AMD.N(S) = seq1
is in STAT_waiting
buffer
FALSE
TRUE
Remove AMD.N(S) = seq1
from STAT_waiting buffer
Clear-buffers
NO
YES
Remove AMD.N(S) = seq1
from Transmission buffer
Seq1 := Seq1 + 1
FALSE
Seq1 = Seq2
TRUE
TRUE
i > 0
FALSE
Update the P flag of
AMD.N(S) = Seq2 – 1 in
Retransmission queue
P := 1
Data Transfer Ready 5

Figure 12-21

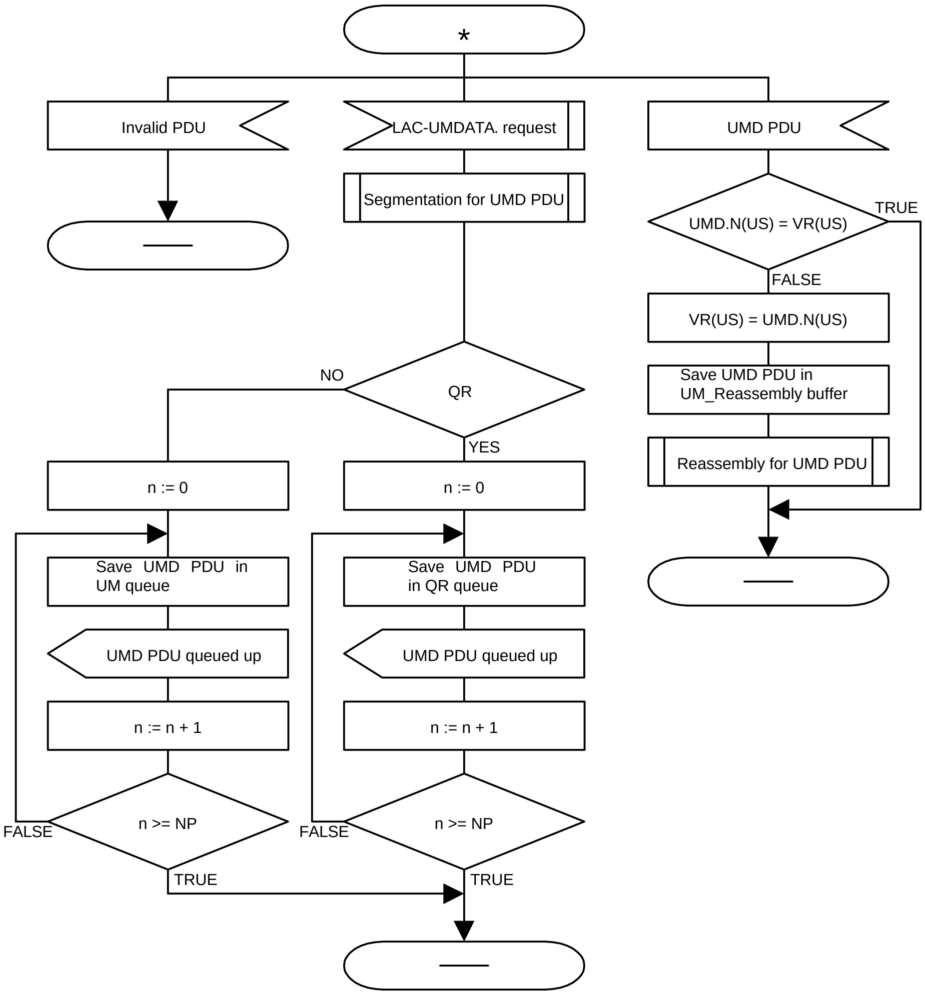

*
Invalid PDU
LAC-UMDATA. request
UMD PDU
Segmentation for UMD PDU
UMD.N(US) = VR(US)
TRUE
FALSE
VR(US) = UMD.N(US)
Save UMD PDU in
UM_Reassembly buffer
Reassembly for UMD PDU
NO
QR
YES
n := 0
n := 0
Save UMD PDU in
UM queue
Save UMD PDU
in QR queue
UMD PDU queued up
UMD PDU queued up
n := n + 1
n := n + 1
FALSE
n >= NP
FALSE
n >= NP
TRUE
TRUE

Figure 12-22

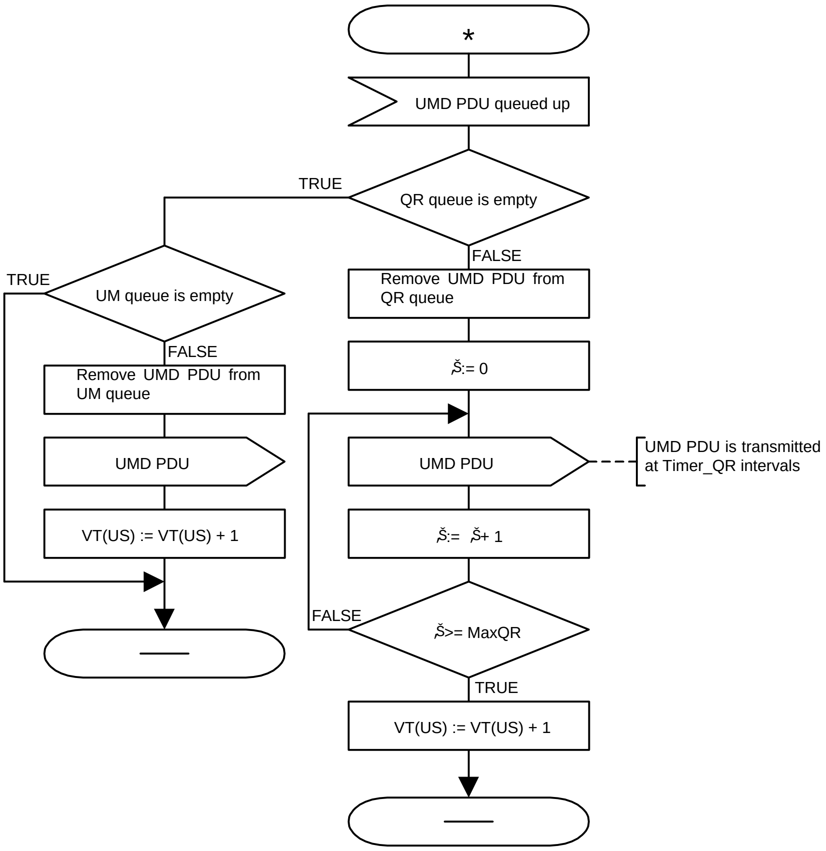


Figure 12-23

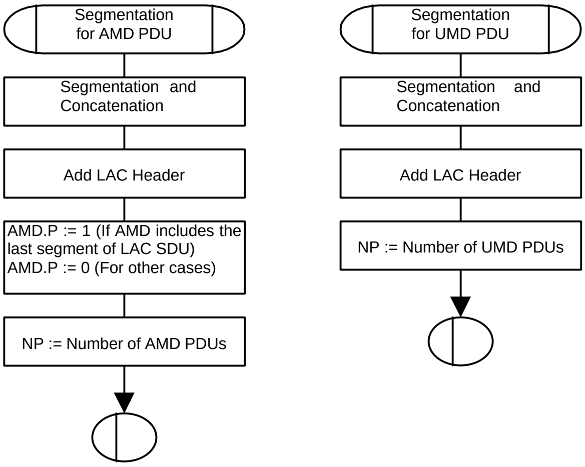

Segmentation for AMD PDU
Segmentation for UMD PDU
Segmentation and Concatenation
Segmentation and Concatenation
Add LAC Header
Add LAC Header
AMD.P := 1 (If AMD includes the last segment of LAC SDU)
AMD.P := 0 (For other cases)
NP := Number of UMD PDUs
NP := Number of AMD PDUs

Figure 12-24

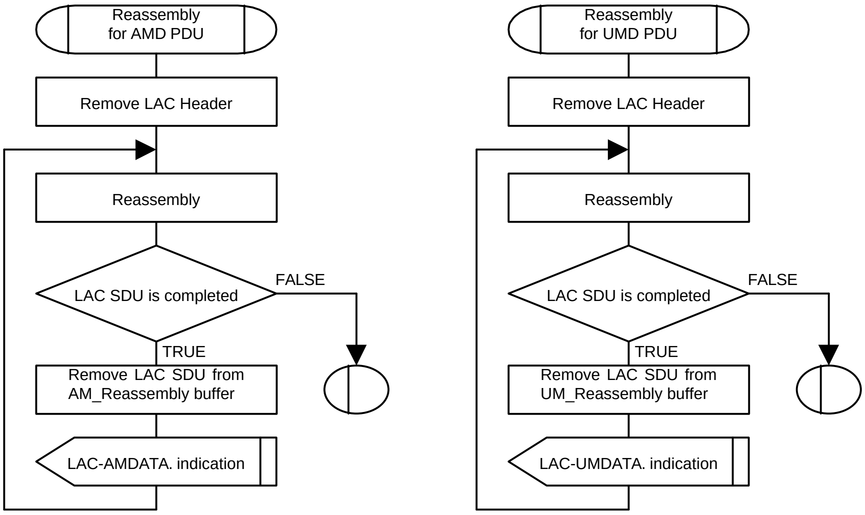


Figure 12-25

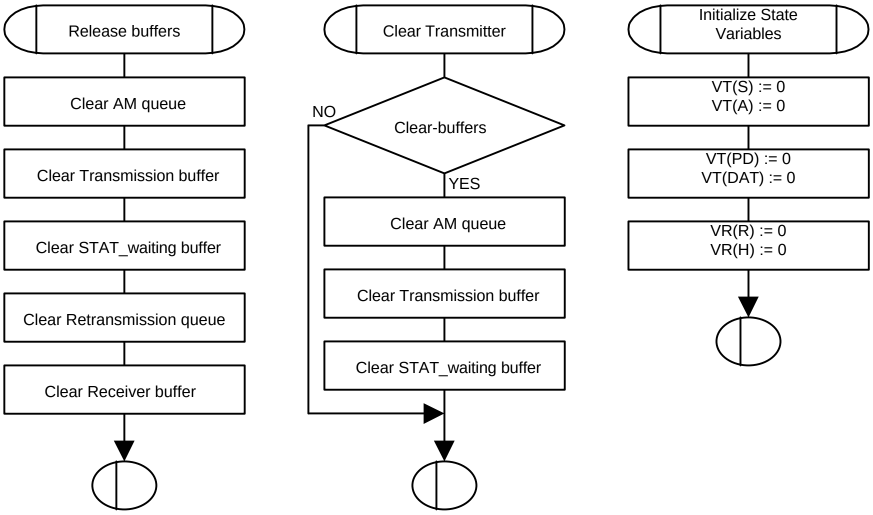
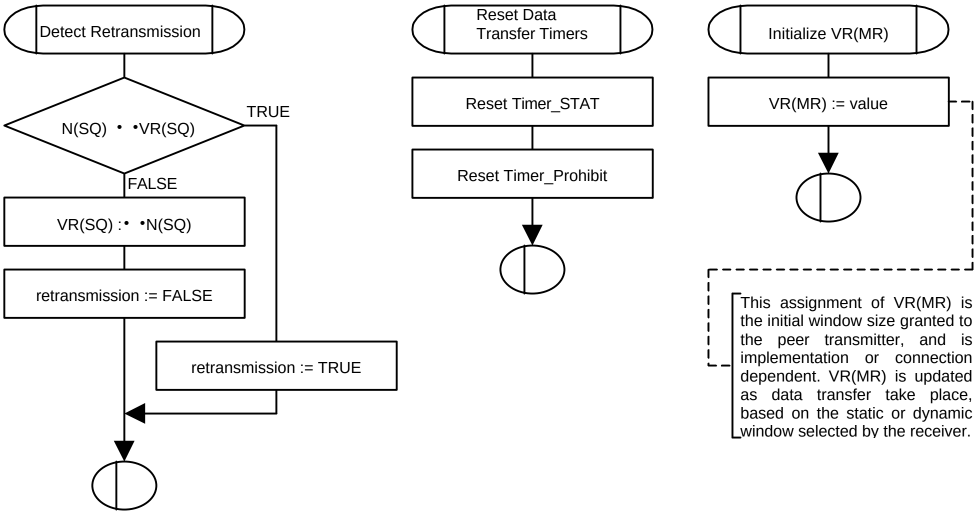


Figure 12-26



This assignment of VR(MR) is the initial window size granted to the peer transmitter, and is implementation or connection dependent. VR(MR) is updated as data transfer take place, based on the static or dynamic window selected by the receiver.

Figure 12-27

Appendix

1. Recommended values

1.1 PDU length

The length of the data field in AMD / UMD PDUs is k (>=0) octets.

1.2 MaxCC

4

1.3 MaxDAT

[FFS]

1.4 MaxQR

[FFS]

1.5 MaxSTAT

This parameter should be an odd integer greater than or equal to 3.

1.6 Timer_STAT

[FFS]

1.7 Timer_Prohibit

[FFS]

1.8 Timer_CC

1 sec

1.9 Timer_QR

[FFS]

13. Annex A Pseudo code describing AMD PDU header Compression

The following Pseudo-Code is an example of algorithm to describe the exact Header Compression Operation that takes place wher several PUs are packed into one RLC PDU.

```
/* Prior to calling this procedure it must be checked that <pus_in_pdu> consecutive PU:s
   are to be transmitted (or there is padding in the end)*/

Compress_PDU (pus_in_pdu, pu_size) {

  li_addition = 0;                  // reset the variable that counts data in full pu:s

  Loop through pus_in_pdu {

    d_e_flag = E-flag for this PU;

    If (d_e_flag == FALSE) {

      Append PU data to PDU data;   // complete PU is SDU-data
      li_addition += pu_size;       // to be added to the next LI

    } else {                        // E-flag is TRUE, so LI-field(s) exist
```

```
      Previous E-flag in PDU = TRUE;   // Either in PDU header or pdu_li_vector;

      j = 0;                        // reset LI-counter for this PU
      pu_data_size = 0;             // reset data size counter for this PU

      Loop until (d_e_flag == FALSE) {

        d_li = next LI;             // in octet j of PU;
        d_e_flag = next E_FLAG;     // in octet j of PU;

        if (d_li is not PADDING) {

          pu_data_size += d_li;     // to keep track of data segment size in this PU);
          d_li += li_addition;      // to add data from previous PU:s to LI-value);
          li_addition = 0;          // reset li_addition;

        }

        Append (d_li + d_e_flag) to pdu_li_vector;

        j++;                        // go to next li_octet, if d_e_flag is TRUE);

      } /* end-of-loop (exit when d_e_flag is TRUE) */

      Append pu_data_size segments starting from j to RLC-PDU data;

    } /* end-of e-flag == TRUE */

  } /* end-of loop through PU:s in PDU */

} /* end-of Compress_PDU */
```

14. History

Document history

Date	Version	Comment
May 1999	1.0.0	The old numbering S2.22 has been removed and replaced with new one 25.322. The document was noted by the TSG/RAN plenary (Yokohama 21-24 April) and the old version 0.1.0 has been upgraded to 1.0.0.
April 1999	0.1.0	The content of Tdoc 99/253 concerning the new STAUS PDU format has been included in section 9.2. The content of Tsoc 99/255 on the RLC toolbox has been included in section 9.8. Approved by WG2.
March 1999	0.0.2	The content of Td155 was included on section 9.7; the principle for the Multiple fixed size RLC PDU with RLC PDU Header compression expressed in td115, td116 were included and part of the proposed changes in td117 applied. The RLC Repetition Scheme proposed in Td 155 was included in Section 9.7. The changes to the RLC Model presented in Td 147 were included in Section 4.2.1. The RLC Protocol States presented in Td 148 were included in Section 9.3.
January 1999	0.0.1	Document created. Based on TSG RAN WG2 Tdoc 016/99, 006/99 and 021/99..

Rapporteur for 3GPP TSG RAN WG2 S2.22 is:

CSELT

Marco Mastroforti
Tel. : +39 011 228 7596
Fax : +39 011 228 7055

Email : marco.mastroforti@cselt.it

Daniele Franceschini
Tel: +39 011 228 5203
Fax: +39 011 228 7613

E-mail: daniele.franceschini@cselt.it

This document is written in Microsoft Word version 6.0c/95.