# ETSI TS 102 744-3-4 V1.1.1 (2015-10)

TECHNICAL SPECIFICATION

**Satellite Earth Stations and Systems (SES);**
**Family SL Satellite Radio Interface (Release 1);**
**Part 3: Control Plane and User Plane Specifications;**
**Sub-part 4: Bearer Connection Layer Operation**

Reference

DTS/SES-00299-3-4

Keywords

3GPP, GPRS, GSM, GSO, interface, MSS, radio, satellite, TDM, TDMA, 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

# Contents

# 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://ipr.etsi.org).

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 ETSI Technical Committee Satellite Earth Stations and Systems (SES).

The present document is part 3, sub-part 4 of a multi-part deliverable. Full details of the entire series can be found in ETSI TS 102 744-1-1 [i.1].

# Modal verbs terminology

In the present document "**shall**", "**shall not**", "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the ETSI Drafting Rules (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

# Introduction

This multi-part deliverable (Release 1) defines a satellite radio interface that provides UMTS services to users of mobile terminals via geostationary (GEO) satellites in the frequency range 1 518,000 MHz to 1 559,000 MHz (downlink) and 1 626,500 MHz to 1 660,500 MHz and 1 668,000 MHz to 1 675,000 MHz (uplink).

# 1 Scope

The present document defines the Bearer Connection Layer operation of the Family SL satellite radio interface between the Radio Network Controller (RNC) and the User Equipment (UE) used in the satellite network. The Bearer Connection Layer (BCn) peer-to-peer interface is described in ETSI TS 102 744-3-3 [9].

# 2 References

## 2.1 Normative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the reference document (including any amendments) applies.

Referenced documents which are not found to be publicly available in the expected location might be found at http://docbox.etsi.org/Reference.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are necessary for the application of the present document.

[1] ETSI TS 125 322: "Universal Mobile Telecommunications System (UMTS); Radio Link Control (RLC) protocol specification (3GPP TS 25.322 Release 4)".

[2] ETSI TS 133 102: "Universal Mobile Telecommunications System (UMTS); 3G security; Security architecture (3GPP TS 33.102 Release 4)".

[3] ETSI TS 133 105: "Universal Mobile Telecommunications System (UMTS); Cryptographic algorithm requirements (3GPP TS 33.105 Release 4)".

[4] ETSI TS 135 201: "Universal Mobile Telecommunications System (UMTS); Specification of the 3GPP confidentiality and integrity algorithms; Document 1: f8 and f9 specifications (3GPP TS 35.201 Release 4)".

[5] ETSI TS 102 744-1-3: "Satellite Earth Stations and Systems (SES); Family SL Satellite Radio Interface (Release 1); Part 1: General Specifications; Sub-part 3: Satellite Radio Interface Overview".

[6] ETSI TS 102 744-1-4: "Satellite Earth Stations and Systems (SES); Family SL Satellite Radio Interface (Release 1); Part 1: General Specifications; Sub-part 4: Applicable External Specifications, Symbols and Abbreviations".

[7] ETSI TS 102 744-3-1: "Satellite Earth Stations and Systems (SES); Family SL Satellite Radio Interface (Release 1); Part 3: Control Plane and User Plane Specifications; Sub-part 1: Bearer Control Layer Interface".

[8] ETSI TS 102 744-3-2: "Satellite Earth Stations and Systems (SES); Family SL Satellite Radio Interface (Release 1); Part 3: Control Plane and User Plane Specifications; Sub-part 2: Bearer Control Layer Operation".

[9] ETSI TS 102 744-3-3: "Satellite Earth Stations and Systems (SES); Family SL Satellite Radio Interface (Release 1); Part 3: Control Plane and User Plane Specifications; Sub-part 3: Bearer Connection Layer Interface".

[10] ETSI TS 102 744-3-5: "Satellite Earth Stations and Systems (SES); Family SL Satellite Radio Interface (Release 1); Part 3: Control Plane and User Plane Specifications; Sub-part 5: Adaptation Layer Interface".

[11] ETSI TS 102 744-3-6: "Satellite Earth Stations and Systems (SES); Family SL Satellite Radio Interface (Release 1); Part 3: Control Plane and User Plane Specifications; Sub-part 6: Adaptation Layer Operation".

## 2.2 Informative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the reference document (including any amendments) applies.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

[i.1] ETSI TS 102 744-1-1: "Satellite Earth Stations and Systems (SES); Family SL Satellite Radio Interface (Release 1); Part 1: General Specifications; Sub-part 1: Services and Architectures".

# 3 Symbols and abbreviations

## 3.1 Symbols

For the purposes of the present document, the symbols given in ETSI TS 102 744-1-4 [6], clause 3 apply.

## 3.2 Abbreviations

For the purposes of the present document, the abbreviations given in ETSI TS 102 744-1-4 [6], clause 3 apply.

# 4 General Architecture

## 4.0 Overview

The function of the satellite radio interface Bearer Connection Layer (BCn) is to provide a number of different data transport services to upper layers. Figure 4.1 illustrates the position of the Bearer Connection Layer within the Family SL air interface protocol stack. An overview of the radio interface layering and relationship to the Bearer Control Layer is provided in ETSI TS 102 744-1-3 [5], clause 4 and ETSI TS 102 744-3-3 [9], clause 4. An overview of the Bearer Connection Layer operation is provided in ETSI TS 102 744-1-3 [5], clause 6.

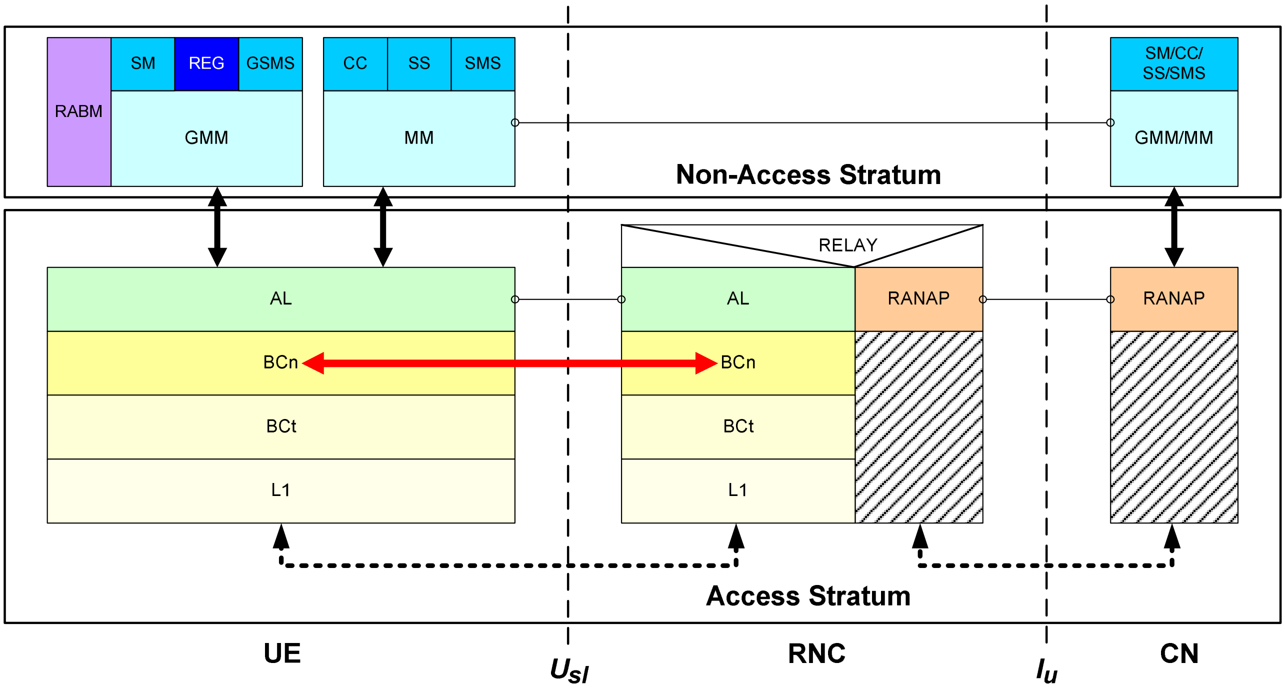


**Figure 4.1: Bearer Connection Layer Position in Protocol Stack (Control Plane Illustrated)**

The Bearer Connection Layer is generic to the radio interface and is responsible for the following:

- Queuing;
- QoS Policing;
- QoS Monitoring;
- Segmentation and Re-assembly;
- Ciphering; and
- Selectable ARQ.

The data transport services provided by the Bearer Connection Layer are used to carry signalling and data PDUs across the satellite radio interface. Three main data transport modes are supported:

- Acknowledged Mode (AM)
- Transparent Mode (TM)
- Unacknowledged (Numbered Frame) Mode (UM)

These are provided to the upper layers via a number of different service access points (SAP). The control plane and user plane architecture of the Bearer Connection Layer is illustrated in Figure 4.2 and Figure 4.3 respectively.

The following clauses describe the Bearer Connection Layer SAP primitives for the upper interface to the Adaptation Layer. The SAP primitives for the lower interface to the Bearer Control Layer are described in ETSI TS 102 744-3-2 [8].

## 4.1 Bearer Connection Layer Entities

### 4.1.0 General

The Bearer Connection Layer consists of a Bearer Connection Manager and instances of one of four possible data handlers (COM_DH, AM_DH, NUM_DH and TM_DH) with their associated Service Access Points (SAPs).

The Control Plane of each UE consists of an AM_DH for the UE Specific Signalling connection, a COM_DH for managing the Common Signalling connection and a Bearer Connection Manager. On the RNC side, the Control Plane consists of multiple sets of AM_DH and Bearer Connection Manager entities, one for each UE registered. It also has a COM_DH for each of the Primary Shared Access Bearer.

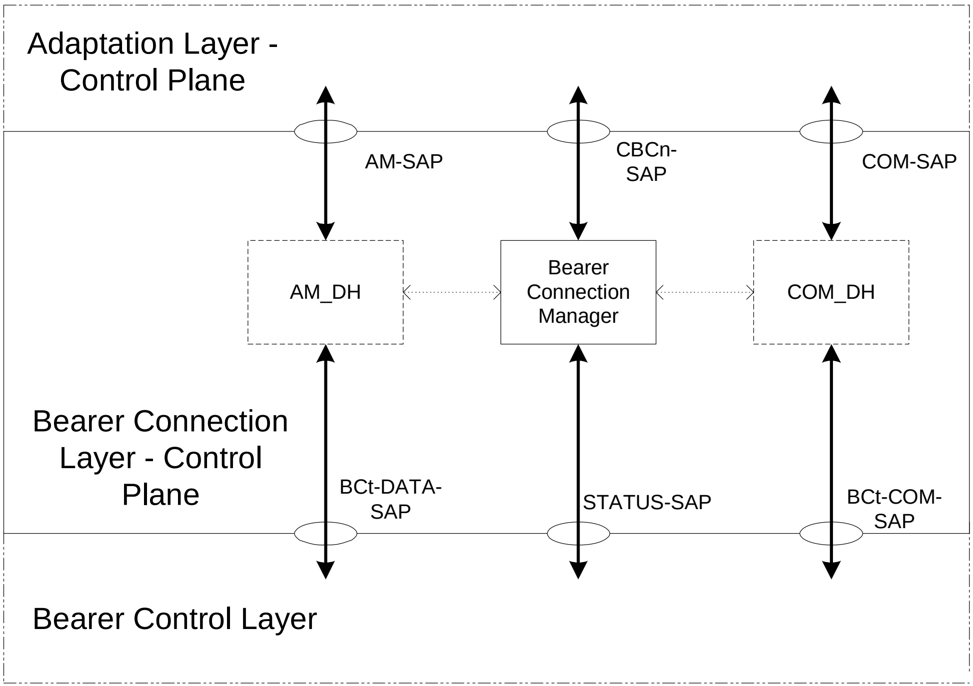


**Figure 4.2: Bearer Connection Layer - Control Plane**

The User Plane of each UE consists of a Bearer Connection Manager and one data handler (either TM_DH, NUM_DH or AM_DH) for each data connection that is set up. On the RNC side, the User Plane consists of multiple sets of these entities, one for each UE registered.

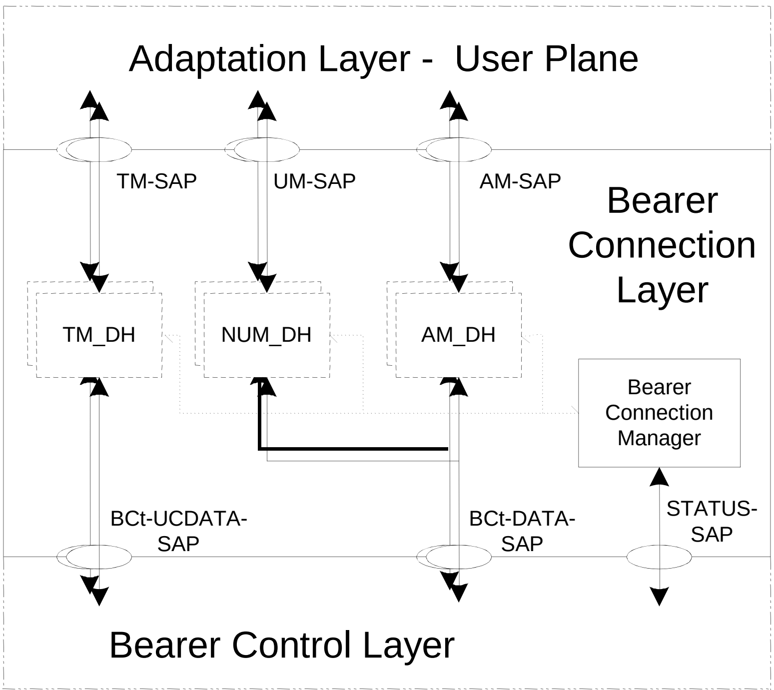


**Figure 4.3: Bearer Connection Layer - User Plane**

### 4.1.1 Bearer Connection Manager (BCnM)

The Bearer Connection Manager (BCnM) provides bearer connection management services to the Adaptation Layer via the CBCn-SAP. This includes the creation and removal of a particular connection, and the setting of its QoS parameters.

Four data handlers are supported by the Bearer Connection Layer:

- Acknowledged Mode Data Handler (AM_DH)
- Transparent Mode Data Handler (TM_DH)
- Numbered Frame Data Handler (NUM_DH)
- Common Signalling Data Handler (COM_DH)

These shall be created by the BCnM based on the requests received from the Adaptation Layer.

### 4.1.2 Acknowledged Mode Data Handler (AM_DH)

The Acknowledged Mode Data Handler shall be created when a reliable delivery message transport is required e.g. for packet data and signalling connections. In the control plane, it is used to transport UE Specific Signalling (UESS) messages from the Adaptation Layer (AL). Note that the SIG-SAP as defined in the Adaptation Layer (see ETSI TS 102 744-3-6 [11]) is an instance of the AM-SAP.

Reliable delivery of data is provided by the AM_DH using an ARQ mechanism. It can also guarantee the ordered delivery of packets and provide ciphering of data packets. The ciphered data Protocol Data Unit (PDU) is sent to the Bearer Control Layer via the BCt-DATA-SAP. Queue status in the AM_DH is reported to the Bearer Control Layer by the BCnM via the STATUS-SAP.

### 4.1.3 Transparent Mode Data Handler (TM_DH)

The transparent mode data handler (TM_DH) is created when a transparent data link is requested. This is primarily used to support circuit switched (CS) traffic. In transparent mode, no connection layer overhead is added to the data arriving from the layer above.

Data coming from the higher layer via the TM-SAP is buffered by the TM_DH before forwarding to the Bearer Control Layer. No ciphering is performed at the connection layer and the data is sent to the Bearer Control Layer via the BCt-UCDATA-SAP. The size of the buffer in TM_DH is always reported to the Bearer Control Layer by the BCnM via the STATUS-SAP.

### 4.1.4 Numbered Frame Data Handler (NUM_DH)

For Un-Acknowledged Mode data coming from the layer above via the UM-SAP, a Numbered Frame Data Handler (NUM_DH) shall be used. The NUM_DH provides in-sequence delivery, segmentation and re-assembly, and ciphering of data. However, packet delivery is not guaranteed. The ciphered data PDUs are sent to the Bearer Control Layer via the BCt-DATA-SAP.

### 4.1.5 Common Signalling Data Handler (COM_DH)

The Common Signalling Data Handler exists only in the control plane of the Bearer Connection Layer. This handler provides buffering for Common Signalling messages from the Adaptation Layer. It also supports the reporting of STATUS information via the STATUS-SAP to the control layer. No Ciphering is required for Common Signalling messages in the Connection Layer.

## 4.2 Interfaces to Upper Layers

### 4.2.1 User Plane Interfaces

#### 4.2.1.0 General

Three SAPs (AM-SAP, UM-SAP and TM-SAP) are defined between the Bearer Connection Layer and the upper layers in the user plane. Their definitions are based on [1]. The primitives and parameters used are summarized below.

#### 4.2.1.1 Acknowledged Mode (AM)-SAP Primitives and Parameters

Three primitives are defined for the AM-SAP, as shown in Table 4.1.

**Table 4.1: Primitives at the AM-SAP**

| Primitive Name | Direction | Parameters |
|---|---|---|
| BCn_AM_DATA_REQ | To BCn Layer | Data-PDU, PDCP-Info, CNF, DiscardReq, MUI, BcnID, [Unsegmentable], [SUSP] |
| BCn_AM_DATA_IND | From BCn Layer | Data-PDU, PDCP-Info, DiscardInfo |
| BCn_AM_DATA_CNF | From BCn Layer | Status, MUI |

The BCn_AM_DATA_REQ primitive is used by the higher layer to request transmission of a data PDU.

The BCn_AM_DATA_IND primitive is used by the BCn AM_DH to deliver a PDU to the higher layer entity.

The BCn_AM_DATA_CNF primitive is used by the BCn AM_DH to confirm to the higher layer entity that a data PDU has been successfully transmitted to the peer AM_DH, or to inform it that a data PDU has been discarded. The definitions of the parameters are as follows:

Data-PDU This is the higher layer data packet that is transferred.

PDCPInfo Optional parameter used by PDCP to transfer up to 5 bits of information to the peer PDCP entity.

CNF Confirmation Request, specifies whether higher layer requires confirmation of delivery of the PDU.

DiscardReq Indicates whether the higher layer requires notification when the PDU is discarded.

MUI Message Unit ID, identifies the higher layer PDU that is being confirmed or discarded.

BcnID Bearer Connection ID, used to identify the Bearer Connection used.

DiscardInfo Indicates to the higher layer that a PDU has been discarded. It is applicable only when in-sequence delivery is configured and it is to be used when the higher layer requires reliable data transfer.

Status Indicates whether a higher layer PDU is successfully transmitted or discarded.

Unsegmentable Optional parameter, indicating the fixed segment size to be used when transmitting this PDU.

SUSP Optional parameter used by the UESS AM_DH when in the suspended state, see clause 5.3.1.4.

#### 4.2.1.2 Transparent Mode (TM) SAP Primitives and Parameters

Three primitives are defined for the transparent mode TM-SAP, as shown in Table 4.2.

**Table 4.2: Primitives at the TM-SAP**

| Primitive Name | Direction | Parameters |
|---|---|---|
| BCn_TM_DATA_REQ | To BCn Layer | Data-PDU, DiscardReq, MUI, BcnID, [Unsegmentable] |
| BCn_TM_DATA_IND | From BCn Layer | Data-PDU, Err_ind |
| BCn_TM_DATA_CNF | From BCn Layer | MUI |

The BCn_TM_DATA_REQ primitive is used by the higher layer to request transmission of a data PDU in transparent mode.

The BCn_TM_DATA_IND primitive is used by the BCn TM_DH to deliver to the higher layer a PDU that has been transmitted in transparent mode.

The BCn_TM_DATA_CNF primitive is used by the BCn TM_DH to inform the higher layer of a discarded data PDU.

The parameter definitions are the same as in AM-SAP, with the following addition:

Err_ind Error indicator, indicates that the PDU is erroneous. This is used to support the "delivery of Erroneous PDU" mode of operation.

### 4.2.1.3 Unacknowledged Mode (UM) SAP Primitives and Parameters

Three primitives are defined for the UM-SAP, as shown in Table 4.3.

**Table 4.3: Primitives at the UM-SAP**

| Primitive Name | Direction | Parameters |
|---|---|---|
| BCn_UM_DATA_REQ | To BCn Layer | Data-PDU, PDCP-info, DiscardReq, MUI, BcnID, [Unsegmentable] |
| BCn_UM_DATA_IND | From BCn Layer | Data-PDU, PDCP-info |
| BCn_UM_DATA_CNF | From BCn Layer | MUI |

The BCn_UM_DATA_REQ primitive is used by the higher layer to request transmission of a data PDU in un-acknowledged mode.

The BCn_UM_DATA_IND primitive is used by the BCn NUM_DH to deliver to the higher layer a PDU that has been transmitted in un-acknowledged mode.

The BCn_UM_DATA_CNF primitive is used by the BCn NUM_DH to confirm to the higher layer of a discarded data PDU.

The parameter definitions are the same as in AM-SAP.

## 4.2.2 Control Plane Interfaces

### 4.2.2.0 General

In the Control Plane, the Bearer Connection Layer interfaces to the Adaptation Layer above. Three SAPs are used in the control plane, namely AM-SAP, CBCn-SAP and COM-SAP.

The AM-SAP in the control plane is used for transporting UE specific signalling messages between Adaptation Layer peers. The Adaptation Layer SIG-SAP is an instance of AM-SAP and carries the same definitions (see clause 4.2.1.1) except that it is connected to the AL instead of the PDCP Layer in the User Plane.

### 4.2.2.1 CBCn-SAP Primitives and Parameters

The CBCn-SAP is used for the exchange of control messages between the Bearer Connection Layer and the Adaptation Layer. The following primitives are defined for the CBCn-SAP, as shown in Tables 4.4 to 4.6.

**Table 4.4: Primitives at the CBCn-SAP common to UE and RNC**

| Primitive Name | Direction | Parameters |
|---|---|---|
| CBCn_CREATE_REJ | To AL | ALPD, BcnID, Rejection Cause |
| CBCn_MODIFY_REJ | To AL | ALPD, BcnID, Rejection Cause |
| CBCn_DESTROY_REQ | To BCn layer | ALPD, SEQUENCE OF {BcnID(SIG-SAP / DATA-SAP)} |
| CBCn_DESTROY_CNF | To AL | ALPD, SEQUENCE OF {BcnID} |
| CBCn_SECURITY_REQ | To BCn layer | ALPD, Mode (Start / Modify), CK, START, SEQUENCE OF {BcnID, UL Activation Time, DL Activation Time} |
| CBCn_SECURITY_CNF | To AL | ALPD, SEQUENCE OF {BcnID} |
| CBCn_FAILURE_IND | To AL | ALPD, BcnID, Failure Cause |
| CBCn_SUSPEND_REQ | To BCn layer | ALPD, SEQUENCE OF {BcnID} |
| CBCn_SUSPEND_CNF | To AL | ALPD, SEQUENCE OF {BcnID, Next BCn Send Sequence Number / Frame Number} |
| CBCn_SUSPEND_REJ | To AL | ALPD, SEQUENCE OF {BcnID}, Rejection Cause |
| CBCn_RESUME_REQ | To BCn | ALPD, SEQUENCE OF {BcnID} |
| CBCn_RESUME_CNF | To AL | ALPD, SEQUENCE OF {BcnID} |

**Table 4.5: Additional Primitives at the RNC CBCn-SAP**

| Primitive Name | Direction | Parameters |
|---|---|---|
| CBCn_CREATE_REQ | To RNC BCn | ALPD, BcnID, (SIG-SAP / DATA-SAP), BCn/AL parameter list, CK OPTIONAL |
| CBCn_CREATE_CNF | To RNC AL | ALPD, BcnID, BctID, {BCt EPDU} OPTIONAL |
| CBCn_MODIFY_REQ | To RNC BCn | ALPD, BcnID, (SIG-SAP / DATA-SAP), BCn/AL parameter list, {CK} OPTIONAL |
| CBCn_MODIFY_CNF | To RNC AL | ALPD, BcnID, BctID, {BCt EPDU} OPTIONAL |
| CBCn_HANDOVER_IND | To RNC AL | ALPD, BCt EPDU |
| CBCn_HANDOVER_RSP | To RNC BCn | ALPD |
| CBCn_HANDOVER_REJ | To RNC BCn | ALPD, Rejection Cause |
| CBCn_HANDOVER_REQ | To RNC BCn | ALPD, Target Spot Beam ID, {Lease Group ID} OPTIONAL |

**Table 4.6: Additional Primitives at the UE CBCn-SAP**

| Primitive Name | Direction | Parameters |
|---|---|---|
| CBCn_CREATE_REQ | To UE BCn | ALPD, BcnID, (SIG-SAP / DATA-SAP), BCn/AL parameter list, BctID, {BCt EPDU} OPTIONAL,<br>{CK} OPTIONAL |
| CBCn_CREATE_CNF | To UE AL | ALPD, BcnID |
| CBCn_MODIFY_REQ | To UE BCn | ALPD, BcnID, (SIG-SAP / DATA-SAP), BCn/AL parameter list, BctID, {BCt EPDU} OPTIONAL,<br>{CK,} OPTIONAL |
| CBCn_MODIFY_CNF | To UE AL | ALPD, BcnID |
| CBCn_HANDOVER_REQ | To UE BCn | ALPD, BctID, BCt EPDU |
| CBCn_HANDOVER_CNF | To UE AL | ALPD |

The CBCn_CREATE_REQ primitive is used by the Adaptation Layer to request the BCnM to create a new instance of a data handler and its associated SIG-SAP or DATA-SAP (i.e. AM-SAP, UM-SAP or TM-SAP). The new bearer connection is identified by the Bearer Connection ID (BcnID) specified in the primitive's parameter. The QoS and connection setup parameters to be used for the new connection are specified in the BCn/AL parameter list, which consists of:

BCn_Type, DefUnseg, QoS parameter list

The BCn_Type parameter specifies the type of Connection to be created and is specified in ETSI TS 102 744-3-5 [10]. The mapping between the type and connection configuration parameters is summarized in Table 4.7.

**Table 4.7: Bearer Connection Types (BCn-Types)**

| BCn Type | Data Handler | Configuration Parameters |
|---|---|---|
| 0 | TM_DH | DelvErrSDU = False, InSeqDelv = True |
| 1 | TM_DH | DelvErrSDU = True, InSeqDelv = True |
| 2 | AM_DH | DelvErrSDU = False, InSeqDelv = True |
| 3 | AM_DH | DelvErrSDU = False, InSeqDelv = False |
| 4 | UM_DH | DelvErrSDU = False, InSeqDelv = True |
| 5 | - | (reserved) |
| 6 | UM_DH | DelvErrSDU = False, InSeqDelv = True, To UE only |
| 7 | - | (reserved) |

For a Transparent Mode connection, the Boolean parameter DelvErrSDU is defined. If set to "true", the Data Handler shall pass on any erroneous PDUs to the higher layer together with the flag Err_ind set. The default value is "false", i.e. only error free PDUs are sent to the higher layer.

For an Acknowledged Mode connection, the Boolean parameter InSeqDelv can be specified. When set to "false", the AM_DH shall deliver completed PDUs as they are received and shall not wait until all previous PDUs have been completed. This could mean PDUs are delivered to the higher layer out of order. The default value is "true", i.e. all data shall be delivered in sequence.

The parameter *DefUnseg* specifies the default unsegmentable size and controls whether any restriction on the segment size applies to this connection. The default is zero, i.e. un-restricted segmentation.

The QoS parameter list contains all UMTS specific QoS parameters from the AL, and the satellite radio interface QoS parameters for the connection and control layers as specified in ETSI TS 102 744-3-5 [10] and ETSI TS 102 744-3-1 [7]. Connection layer parameters in this list include *AllocationSize*, *MeanRate*, *TargetLatency*, *DiscardLatency*, *ResponseTime* and *MaxIdleTime*.

On the UE side, the CBCn_CREATE_REQ also carries the BctID and any embedded BCt information that the Adaptation Layer may have received from the RNC. Such *BCt EPDU* information shall be passed to the Bearer Control Layer, In addition, at the UE side, any Bearer Connection specific parameters, such as the group CK values to be applied to a Multimedia Broadcast Multicast Services (MBMS) Connection are received in the CBCn_CREATE_REQ.

After the data handler and SAP have been successfully created, the Bearer Connection Layer shall respond with a CBCn_CREATE_CNF primitive. On the RNC side, any Bearer Control information (e.g. ConnectionAssociation, ConnectionReasociation, etc.) that requires to be embedded, shall be passed to the Adaptation Layer via the *BCt EPDU* field and in addition, any Bearer Connection specific parameters, such as the group CK values to be applied to a MBMS Connection.

If for some reason the operation failed, then a CBCn_CREATE_REJ primitive shall be returned with the Rejection Cause code.

The CBCn_MODIFY_REQ, CBCn_MODIFY_CNF and CBCn_MODIFY_REJ primitives are similar to the CBCn_CREATE primitives, except that they are only used when modification of an existing connection is requested.

The CBCn_DESTROY_REQ primitive is used by the Adaptation Layer to request the release of a bearer connection (identified by its BcnID) and the removal of the associated data handler and DATA-SAP. Any un-transmitted data will be lost. After the connection is removed, the Bearer Connection Layer shall respond with the CBCn_DESTROY_CNF primitive.

The CBCn_SECURITY_REQ primitive is used by the AL to set up the ciphering operation for a set of bearer connections (identified by their BcnIDs). The ciphering parameters for all connections shall then be configured according to the specified CK and START parameters at the specified DL and UL Activation Time. For any Transparent Mode Connection, the Bearer Connection shall pass the ciphering configuration to the Bearer Control Layer which shall confirm the setup. It should be noted that this security information only applies to UE-specific connections, and not the forward direction associated with a MBMS Connection, which uses a group security association.

After all the ciphering setup is completed (and confirmed from the Bearer Control Layer), the Bearer Connection Layer shall respond to the Adaptation Layer with the CBCn_SECURITY_CNF primitive.

The CBCn_FAILURE_IND primitive is used by the BCn to signal to the Adaptation Layer a failure of the bearer connection. This will occur, for example, if the MaxIdle Timer times out. Two Connection Layer Failure Cause codes, namely *ConnFailure* and *MaxIdleReached*, are currently defined. This primitive is also used to forward Bearer Control Layer error conditions to the Adaptation Layer (i.e. received via the BCt_FAILURE_IND primitive). Valid Bearer Control failure codes include *MaxUnAckReached*, *NoForwardBearer* and *IncorrectForwardBearer* on the UE side, and *UEChkLimitReachedRelease* on the RNC side.

The CBCn_SUSPEND_REQ primitive is used by the AL to stop the transmission of any new (unsegmented) data for the specified bearer connections. Retransmission of old data, selective reject, poll, poll response signalling are still allowed. Once all connections are halted, the Bearer Connection Layer shall respond with CBCn_SUSPEND_CNF and specify the next send sequence number for each connection to the Adaptation Layer. If the operation failed (for example, if there is an error in the list of connections to be suspended), a CBCn_SUSPEND_REJ primitive shall be returned with the corresponding Rejection Cause code.

The Bearer Connection Layer shall resume the sending of new data for these connections when the CBCn_RESUME_REQ primitive is received. It shall reply with CBCn_RESUME_CNF when the connections are restarted.

The UE Specific Signalling connection shall not be suspended (special transmit path processing rules apply while in the suspended state, see clause 5.3.1.4), but the next Send Sequence number shall be returned via CBCn_SUSPEND_CNF primitive. Any Transparent Mode connections shall not be suspended, but the current frame number shall be returned via the CBCn_SUSPEND_CNF primitive.

The CBCn_HANDOVER_IND primitive is used by the RNC BCn to signal the Adaptation Layer to send a *Handover* signal for all of the UE's connections. This is triggered when the RNC Local Resource Manager (LRM) function initiates a Handover operation in the Bearer Control Layer via the STATUS_SAP. The primitive also specifies all the associated *ConnectionAssociation* and *ConnectionReassociation* information via the *BCt EPDU* parameter. When the Adaptation Layer receives the *HandoverAck* message, it shall return the CBCn_HANDOVER_RSP primitive to the Bearer Connection Layer. If the handover operation fails, the Adaptation Layer shall send a CBCn_HANDOVER_REJ primitive to the Bearer Connection Layer with the corresponding Rejection Cause code. The Bearer Connection Layer shall route this on to the Bearer Control Layer.

The Adaptation Layer on the RNC side can send the CBCn_HANDOVER_REQ primitive to the Bearer Connection Layer to start the UE-Initiated Handover Procedure or the Lease Mode Handover Procedure in the Bearer Control Layer. The Bearer Connection Layer shall pass this onto the Bearer Control Layer in order to trigger the LRM functions to start the Handover procedure.

On the UE side, when the Adaptation Layer receives the *Handover* signalling PDU, it signals to the Bearer Connection with the CBCn_HANDOVER_REQ to reconfigure the BCn and BCt layers. When the UE Bearer Connection and Control Layer finished retuning to the new bearer, the CBCn_HANDOVER_CNF message shall be returned to the Adaptation Layer. The Adaptation Layer shall then send the *HandoverAck* message to the RNC.

#### 4.2.2.2 COM-SAP Primitives and Parameters

Two Primitives are defined for the COM-SAP, as shown in Tables 4.8 and 4.9.

**Table 4.8: Primitives at the UE COM-SAP**

| Primitive | Direction | Parameters |
|---|---|---|
| AL_COMDATA_REQ | To BCn | UE NAS ID and Type, REGM PDU Type, REGM PDU, RetryCount |
| AL_COMDATA_IND | To AL | REGM PDU Type, REGM PDU |

**Table 4.9: Primitives at the RNC COM-SAP**

| Primitive | Direction | Parameters |
|---|---|---|
| AL_COMDATA_REQ | To BCn | UE NAS ID and Type, REGM PDU Type, REGM PDU |
| AL_COMDATA_IND | To AL | UE NAS ID and Type, REGM PDU type, REGM PDU |

The AL_COMDATA_REQ primitive is used by the Adaptation Layer to request the Bearer Connection Layer to send a Common Signalling PDU (i.e. a Registration Manager (REGM) PDU) to its peer. On the UE side, when the Adaptation Layer retransmits a REGM PDU, it shall specify the correct *RetryCount* value. This is to ensure the Bearer Control Layer performs the correct back-off when processing repeated REGM PDUs. The *RetryCount* value is defined as the number of times the message has been sent before. Thus, when the message is sent the first time, the *RetryCount* value shall be zero. The *RetryCount* parameter is not required on the RNC side.

When a Common Signalling PDU is received, the Bearer Connection Layer shall deliver it to the Adaptation Layer using the AL_COMDATA_IND primitive. On the UE side, only the REGM PDU and its type are passed up. On the RNC side, the decoded UE NAS (Non Access Stratum) ID shall also be delivered at the same time.

## 4.3 Interfaces to the Lower Layer

The Bearer Connection Layer interfaces to the Bearer Control Layer below. Four interface SAPs are defined for this purpose, namely BCt-DATA-SAP for ciphered data, BCt-UCDATA-SAP for un-ciphered data, BCt-COM-SAP for Common Signalling Messages and STATUS-SAP for status information. Their definitions are specified in ETSI TS 102 744-3-2 [8].

# 5 Bearer Connection Operations

## 5.0 General

The following clauses describe the detailed operation of the various entities in the Bearer Connection Layer.

## 5.1 Bearer Connection Manager

The primary function of the Bearer Connection Manager (BCnM) is to respond to AL commands requesting the creation and removal of particular bearer connections and their associated data handlers and SAPs. It is also responsible for the configuration of specific QoS Settings for created data handlers.

On receiving a CBCn_CREATE_REQ message from the AL, the BCnM will create the appropriate data SAP and the associated data handler. Only one data handler shall be created for each bearer connection. The BCnM also sets up the data handler to supply the required QoS for that connection.

Once a data handler is created, the BCnM is also responsible for monitoring its queue status and sending it to the Bearer Control Layer via the STATUS-SAP. The BCnM is also responsible for routing data request messages from the Bearer Control Layer (via the STATUS-SAP) to the corresponding data handler. The BCnM shall also set up the data handler so that its output data is associated with the appropriate Bearer Control Data SAP.

Unless the Unsegmentable parameter is set by the higher layer for an individual PDU (via BCn_TM_DATA_REQ, BCn_UM_DATA_REQ or BCn_AM_DATA_REQ interface), the default Unsegmentable size shall be used in all status messages sent to the Bearer Control Layer. When a different Unsegmentable size is set for a particular PDU, then the BCnM shall ensure that when the PDU reaches the head of the queue, a new queue status is generated to reflect the requested Unsegmentable size.

## 5.2 Quality of Service (QoS) Control Function

Each bearer connection has a negotiated Quality of Service (QoS). It is possible that some sources may be unable to schedule to a given QoS. Thus, when sources are exceeding the negotiated QoS, the Bearer Connection Layer will respond by buffering the data, and extend the targeted delivery time (target latency) accordingly. This behaviour is common to all data handlers.

All the bearer connection data handlers are expected to keep records, for each PDU received from the upper layer, of the time by which the PDU is expected to be delivered. Whenever a PDU arrives, the data handler shall calculate the expected delivery time of this PDU and the expected delivery time of the PDUs at the head and tail of its data queue, according to the following formula.

*If PDU is first in queue then*
*TdeliveryTime(n) = Tnow + max(QueueSize / MeanRate + rtt/2, TargetLatency)*
*else*
*TdeliveryTime(n) = max(TdeliveryTime(n-1), QueueSize/MeanRate + rtt/2+Tnow, TargetLatency+Tnow)*
*end if*
*TimeFront = TdeliveryTime (PDU at head of queue)*
*TimeEnd = TdeliveryTime (last PDU in the queue)*

| | | |
|---|---|---|
| *where* | *TdeliveryTime(n)* | *is the expected delivery time of the $n^{th}$ PDU in the queue* |
| | *Tnow* | *is the current time (at the time the $n^{th}$ PDU is put in the queue)* |
| | *rtt* | *is the round trip time (include processing delay) of the connection (default value = 668 ms)* |
| | *MeanRate* | *is the target mean rate in the direction for this connection (at the time the $n^{th}$ PDU is put in the queue)* |
| | *QueueSize* | *is the size of the queue of unsent data up to and including the nth PDU excluding connection and control layer overheads. (The size is calculated just after the $n^{th}$ PDU is added to the queue.)* |
| | *TargetLatency* | *is the targeted latency for packets in the direction of this connection (at the time the $n^{th}$ PDU is put in the queue).* |

In Acknowledged Mode, if retransmission is required, those data segments requested are placed at the effective head of the queue, and *TimeFront* is set to:

*TimeFront = TdeliveryTime(first segment of PDU in the queue)*

When the requirement to send a RR (Receive Ready) Poll or a poll response (either RR or Selective Reject (SREJ)) is generated, a BCt_DATASTATUS_REQ (see ETSI TS 102 744-3-2 [8]) message should be sent to the Bearer Control layer with *TimeFront* set to:

*TimeFront = Tnow*

indicating that the signalling data should be sent immediately (see clause 5.3.6.3).

If the Bearer Connection signalling data (i.e. RR or SREJ) is the only data to be sent, then

*TimeEnd = Tnow*

but if there is other data in the queue then *TimeEnd* shall be calculated and set to *TdeliveryTime(last PDU in the queue)* as above.

In a similar way, the Bearer Connection Layer also keeps track of the Discard Time of each incoming PDU. The Discard Time of a PDU is given by:

*If PDU is first in queue then*
*TdiscardTime(n) = Tnow + max(QueueSize / MeanRate + rtt/2, DiscardLatency)*
**else**
*TdiscardTime(n) = max(TdiscardTime(n-1), QueueSize/MeanRate + rtt/2+Tnow, DiscardLatency + Tnow)*
*end if*

*where* *TdiscardTime(n)* *is the Discard time of the $n^{th}$ PDU in the queue*
*DiscardLatency* *is the Discard latency for packets in the direction of this connection (at the time the $n^{th}$ PDU is put in the queue).*

When the discard time of a PDU is reached, then it shall be removed from any transmit buffer for the bearer connection. Whenever data is discarded by the Bearer Connection Layer, it shall be reported to the Bearer Control Layer using the STATUS_SAP. The primary function of the discard behaviour is to remove data that the application layer no longer requires. For example, real time data like voice frames should not be queued but dropped instead.

# 5.3 Acknowledged Mode Operation

## 5.3.0 General

This clause describes the operation of the Acknowledged Mode Data Handler AM_DH, especially the ARQ mechanism and data handling behaviour.

## 5.3.1 AM_DH Architecture

### 5.3.1.0 Overview

The acknowledged mode data handler consists of three major components: a transmit path processing unit, a receive path processing unit and a control unit. Figure 5.1 shows a model of an acknowledged mode data handler.

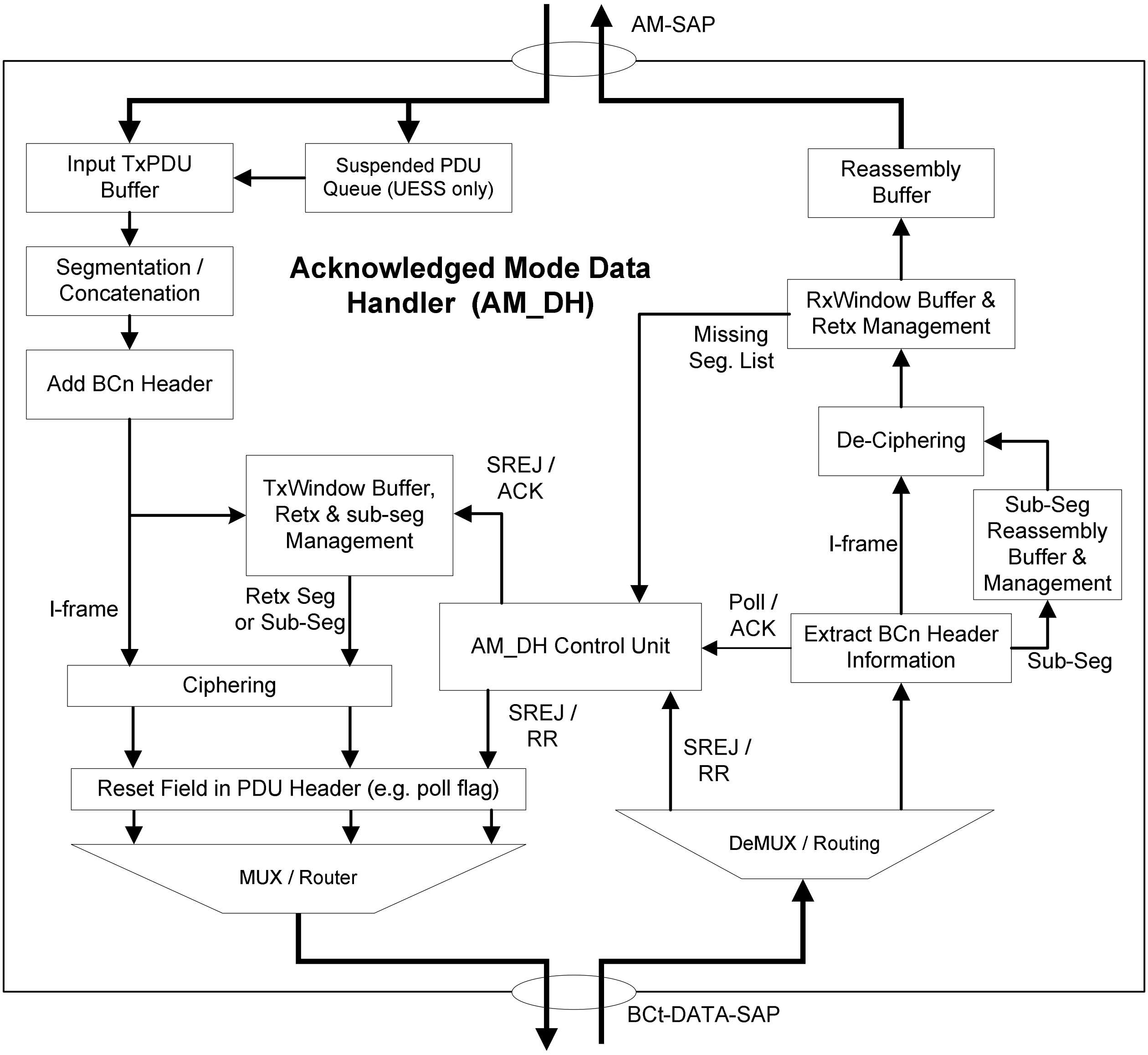


**Figure 5.1: Model of an Acknowledged Mode Data Handler**

### 5.3.1.1 Transmit Path Processing

At the input of the transmit path unit is the main data storage entity, the TxPDU buffer, where the AM_DH stores all incoming data awaiting transmission. The variable-length PDUs are stored in a FIFO (First In First Out) queue along with information on the whole PDU's target and discard time. Along with the actual data storage area the TxPDU buffer entity has mechanisms for accessing and maintaining the buffer, i.e. adding new PDUs, discarding PDUs (or partial PDUs) that have passed the discard time, and obtaining an appropriately sized segment ready for transmission.

When capacity is available for sending a new segment of data, the correct amount of data is retrieved from the TxPDU, segmented (or concatenated), and header information added to form a BCn PDU. The data part of the segmented PDU can now be ciphered and a copy of the un-ciphered data can be stored in the TxWindow buffer awaiting acknowledgement from the receiving end.

The control unit might update the header field, (e.g. with latest poll state information), before sending it to the Bearer Control Layer.

Once a data PDU has entered the TxWindow buffer, it shall only be removed when either an acknowledgement has been received or its discard time has expired. In the meantime, if an SREJ is received, then selected segments will be marked for retransmission.

If there are data segments awaiting retransmission, then these shall be transmitted ahead of any new data. If insufficient capacity is available to carry the original segment, then the segment shall be split up into sub-segments (see clause 5.3.3.4) for retransmission.

These segments or sub-segments are retrieved from the TxWindow buffer and their headers adjusted before passing on to the Bearer Control Layer. The original ciphered data shall therefore be sent unchanged.

If the discard time of a data segment in the TxWindow buffer expired before an acknowledgement is received, then the data shall be discarded. However, the segment and hence its sequence number shall still be maintained. If an SREJ is received, then an empty header only (i.e. either a zero length data-seg part or a zero length pdu-data part) segment with the original segment number and type shall be transmitted instead of the discarded data segment. If a Beginning of Message (bom) or Single Segment Message (ssm) BCnPDU is received with zero length data-seg part, then the receiver shall assume the PDU-length is zero. An upper layer PDU shall be discarded by the receiving end if there is a mismatch between the length field and the actual data volume. If all BCn segments forming an upper layer PDU are discarded by the transmitter, then no data is received by the connection layer and no upper layer PDU shall be forwarded. If the discard time expires while the segment is partially retransmitted using the sub-segment format, then the remaining sub-segments shall be discarded. An empty segment with the original segment number and type shall be transmitted instead of the rest of the sub-segments.

If the discard time of a data segment in the TxPDU buffer expired before the whole PDU is sent, the remaining data shall be discarded. However, when the next opportunity to send data arrives, an empty End-of-Message (eom) segment shall be transmitted. This is done to enable the correct termination of the data PDU at the receiving end. This also causes the PDU to be discarded by the receiving end because of a mismatch between the length field and the actual data volume.

If the discard time of a data PDU in the TxPDU buffer expired before any segment can be sent, then the whole PDU shall be discarded from the TxPDU buffer. It is assumed that the higher layer protocol handles this failure scenario.

In addition, the AM_DH entity can transmit control messages Selective Reject (SREJ) and Receive Ready (RR). These messages are generated by the AM_DH control unit and are sent without ciphering. These messages include SREJ and RR. A SREJ or RR poll response message has a higher priority over all data segments while a RR poll message has the lowest priority.

### 5.3.1.2 Receive Path Processing

When a BCnPDU I-frame or sub-segment is received from the Bearer Control Layer, its control information (e.g. poll flag, sequence no.) is first recovered from the header and sent to the AM_DH control. If it is an I-frame, then it is routed to the de-ciphering unit in order to recover the data segment. If it is a sub-segment, then it is routed to the Sub-Segment Reassembly Buffer for processing. Once all the sub-segments are received, then the original segment is reconstructed, and routed to the de-ciphering unit for data recovery.

If a data segment is received out of order, then it is stored in the RxWindow buffer. The RxWindow buffer & Retx Management unit is also responsible for detecting missing segments and updating the AM_DH control unit with the current list of missing segments.

In-sequence data segments are then transferred to the Reassembly buffer. Once all segments of a data PDU have been received, the PDU is forwarded to the layer above. If there is a mismatch between the declared length of the PDU and the actual data received, then the received PDU shall be discarded. This can occur as a result of the PDU's discard timer expiring at the sender's end.

When an acknowledged mode control message (i.e. SREJ or RR) is received, it is routed to the AM_DH control unit for processing.

### 5.3.1.3 AM_DH Control Unit

The primary function of the AM_DH Control unit is to provide an interface between the transmit and receive path processing units, in order to handle acknowledged mode control messages (i.e. SREJ and RR).

The AM_DH control unit will route any received SREJ and RR poll response messages and implicit acknowledgements embedded in the BCn header information to the TxWindow Buffer and Management unit. Reception of these allows acknowledged segments to be discarded from the TxWindow buffer and any lost segments to be retransmitted. The AM_DH will also generate any RR poll request when necessary.

The AM_DH also collects information on missing segments from the RxWindow buffer and Retx Management unit in order to construct any necessary SREJ messages. It is also responsible for detecting and responding to any poll requests from received RR poll messages or from a poll flag set in the BCnPDU header information.

All the bearer connection acknowledged mode timers are also managed by the AM_DH control unit.

#### 5.3.1.4 Transmit Path Processing in the Suspended State (UESS AM_DH only)

When the BCnM receives the CBCn_SUSPEND_REQ primitive from the AL, it instructs the UESS AM_DH to enter the suspended state and initialize a separate FIFO queue for suspended PDUs ("the suspended PDU queue").

When in the suspended state and an AL PDU is received via the BCn_AM_DATA_REQ primitive, the UESS AM_DH evaluates the "SUSP" flag:

- if the flag is FALSE, then the AL PDU is placed at the tail of the suspended PDU queue;
- if the flag is TRUE, then the contents of the suspended PDU queue are transferred to the TxPDU buffer, preserving the FIFO order (i.e. the PDU at the head of the suspended PDU queue is placed at the tail of the TxPDU queue and so forth). Next the PDU received in the BCn_AM_DATA_REQ primitive is placed at the tail of the TxPDU buffer. The suspended PDU queue shall be empty after this operation is completed.

When the BCnM receives the CBCn_RESUME_REQ primitive from the AL, it instructs the UESS AM_DH to transfer the contents of the suspended PDU queue to the TxPDU buffer, preserving the FIFO order. The suspended PDU queue shall be empty after this operation is completed. The UESS AM_DH then exits the suspended state and resumes normal operation.

### 5.3.2 The ARQ Mechanism

An ARQ system fundamentally consists of two independent event driven processes, which communicate asynchronously with each other to exchange data as efficiently as possible whilst avoiding delays and stalls in the data flow. The satellite radio interface ARQ mechanism is based on the standard HDLC (High Level Data Link Control) Asynchronous Balanced Mode and utilizes multiple request SREJs and an extended sequence numbering scheme.

The ARQ mechanism ensures complete ordered transmission of data segments by tracking and acknowledging them using sequence numbers. User data PDUs are segmented according to the available capacity at the time of transmission. Each data segment (I-frame) is given a unique sequence number (in the field ns) by the sender. In order to avoid large overheads, the sequence number shall be of limited size and reusable. For this reason, modulo N numbering is used (where N is the number range of the sequence number, which is 1 024). Thus all calculation for sequence number operation described below shall be done on a modulo 1 024 basis. As numbers are reused and ordering is of vital importance, it is essential to ensure that a transmission identified by a sequence number cannot be confused with a transmission using the same number from a previous or later time around the numbering. This is achieved by having a window of operation.

The maximum window size of operation is defined as (N/2 - 1 = 511), which allows the receiver's window of operation to get a full window size ahead of the transmitter's without incorrectly identifying a transmission. The window of operation also allows the data transmissions to run ahead of the acknowledgements and it is therefore sized to allow new transmissions to continue to be sent whilst multiple attempts are made to retransmit an old segment (i.e. sufficient values to cover the number of segments which can be transmitted within multiple round trip delays).

Once a missing segment has been detected (via out-of-order sequence number) by the receiver, the detected segment is added to the Selective Reject (SREJ) request list. A given sequence number can appear in this list only once and the list is ordered to ensure that the oldest missing segment is the first to be requested. When an opportunity to send arrives, a Selective Reject (SREJ) command is returned back to the sender. As many of the sequence numbers in the SREJ request list as will fit into the offered packet are placed in this SREJ Frame.

In order to retain the integrity of the ARQ mechanism the window of operation shall not be exceeded. This means the sequence number of the next segment to transmit cannot exceed the sequence number of the next expected acknowledgement, i.e. the oldest segment of which retransmission may be required, plus the size of the window. In the event that this limit is reached, transmission of new segments is halted until the next expected acknowledgement is received and the window moves on. A Poll will be generated at this point to trigger a response.

An acknowledgement is normally provided via ordinary I-frames or sub-segments. It takes the form of the field nr, which is the sequence number of the next in-order segment that the sender is expecting. This effectively acknowledges the correct reception of segments up to and including (nr-1).

As a steady stream of return frames cannot be guaranteed, a Poll / Poll Response system is also provided to ensure acknowledgements are received regularly. A 1-bit Poll Flag (pf) is allocated to all types of segments to allow the transmission of a Poll or Poll Response. If no data needs to be transferred, a special RR frame is created for the purpose of transporting the Poll / Poll Response. This Polling mechanism is also used to ensure that the last segment of a transmission is received. An ARQ Timer is also set at the sender's end, to ensure regular responses are received.

In order for the ARQ mechanism to function, a number of state variables and lists are maintained at both sender and receiver end processes. These include:

| | |
|---|---|
| VR: | The sequence number of the next in-order segment expected to be received |
| VS: | The sequence number of the next in-order segment to be transmitted |
| VT: | The sequence number of the latest segment to be transmitted (i.e. VS-1) when the ARQ Timer is set or restarted |
| VA: | The sequence number of oldest un-acknowledged segment |
| PR: | Poll Response Required |
| WPR: | Waiting for Poll Response |
| PNS: | Send Poll on next available segment |
| SREJ request list: | List of missing segment detected but not requested |
| SREJ waiting list: | List of missing segment requested but not acknowledged |
| Retx list: | List of segments still required to be retransmitted |

## 5.3.3 Data Packet Structures

### 5.3.3.0 General

The detailed structures of all possible data segments are specified in ETSI TS 102 744-3-3 [9]. The following clauses only summarize the definition of the fields used in the ARQ mechanism.

### 5.3.3.1 Definition of the I-Frame

An I-frame is used to transmit the actual user data, either new or retransmission. Its header consists of three fields used for the ARQ operations: a Poll Flag (pf), a transmit sequence number (ns) and the expected receive sequence number (nr).

The transmit sequence number (ns) is the sequence number of the actual data segment sent within the I-frame, whereas the receive sequence number (nr) is the next in-order segment expected to be received by the sender. This is used to acknowledge the correct reception of segments up to and including (nr-1).

The Poll Flag field (pf) is set to 1 if the sender wishes to obtain an acknowledgement (i.e. a Poll Response) back from the remote end. Note that an I-frame shall only be used to carry a Poll and NOT a Poll Response.

If the I-frame is carrying new data, then the Poll Flag shall only be set if the segment is also the end of a PDU (i.e. either an End-of-Message eom segment or a Single Segment Message ssm segment). This is to align the possible request for resource at the end of a higher layer PDU (e.g. TCP ACK).

### 5.3.3.2 Definition of the RR-Frame

A RR-frame is a supervisory frame used to carry either a Poll or Poll Response. It contains a Poll Flag (pf) field and a sequence number field (nseg).

When the Poll Flag (pf) is set to 1, the RR-frame is being used as a Poll, and the sequence number field (nseg) shall be set to the sequence number of the latest transmitted segment (i.e. nseg = VS - 1). Thus, no RR poll shall be sent before the first ever I-frame (ns=0) is transmitted.

When the Poll Flag (pf) is set to 0, the RR-frame is being used as a Poll Response, and the sequence number field (nseg) shall be set to VR, the next in-order segment expected to be received by the sender.

#### 5.3.3.3 Definition of the SREJ-Frame

An SREJ-frame is a supervisory frame used to carry a list of missing segment sequence numbers. It also contains a Poll Flag (pf) field. This shall be set to 1 when the SREJ frame is used as a Poll Response. A SREJ frame shall not be used as a Poll.

The list of missing segment sequence numbers is carried in the nr field and the nrs fields. The number of entries in the nrs field shall be specified in the num-sel field. Thus, in case there is only one missing segment, the sequence number is carried in nr and num-sel is 0. The list shall always be sorted with the earliest sequence number in the nr field.

When a Poll is received, the list of missing segments shall be reset, i.e. all un-acknowledged SREJ segments shall be put back into the SREJ request list. Thus, the SREJ frame, which is also a Poll Response, shall contain a list of all missing segments. Furthermore, the nr field shall also be the same as VR. Such an SREJ frame also serves as an acknowledgement to segments up to and including (nr - 1).

If the SREJ-frame is too large to fit into the available space, then it shall be split into two or more SREJ-frames. Furthermore, if a Poll Response is also required, the pf field shall be set for the first SREJ-frame only. Subsequent SREJ frames shall be sent with pf set to 0. The nr field of any of these SREJ frames shall be set to the earliest sequence number in the SREJ request list, as per normal SREJ.

#### 5.3.3.4 Definition of the Sub-Segment Frame

A Sub-Segment Frame is used to carry part of a retransmitted I-frame. Its header contains the sequence number (ns), and segment type (s-type) of the original segment. All sub-segments shall be numbered (ssegn), starting from "0" and can go up to "15". Hence, the mechanism allows the original PDU segment to be split into a maximum of 16 sub-segments. The last sub-segment shall always be marked by the end-of-subsegment flag (eos).

The header also contains the pf and nr field, as defined for an I-frame. This allows the Sub-segment frame to carry a Poll and acknowledge the correct reception of segments up to and including (nr-1). In common with an I-frame, a sub-segment frame shall not be used as a Poll Response.

The data-sub-seg part of the sub-segment frame contains part of the data-seg from the original I-frame. Data in the sub-segment shall be ciphered using the original ciphering bitstream in order that the data-seg reconstructed from the sub-segments is identical to the original data segment.

### 5.3.4 Sending I-Frames

An I-frame is used to send new data or retransmit lost segments. When new data is transmitted, the new segment is given a sequence number (ns = VS), and the next sequence number (VS) is then incremented. If it is a retransmission, then the old sequence number is used, and there is no change in VS.

Whenever an I-frame is used to send new data, the size of the un-acknowledged buffer is checked. If the number of un acknowledged segments (VS - VA) is larger than a quarter of the TxWindow Size, and if the oldest un acknowledged segment was transmitted more than a round trip time ago, then a Poll request shall be generated. This is then used to update the Polling state machine to determine whether the poll flag will be set in the next segment.

If the I-frame is carrying the last segment in a transmit sequence (i.e. the input TxPDU Buffer is empty after transmitting this segment), then a Poll request shall also be generated, and the Polling State updated. The Poll Flag associated with the I-frame shall then be set according to the updated Polling State (see clause 5.3.6). When this I-frame is sent, the ARQ Timer shall also be started, providing it is not already running. The sequence number of the I-frame shall then be stored in VT.

Whenever a segment is being sent, the discard time of the oldest un-acknowledged segment is checked. If it is due to expire less then (2 × ResponseTimeParam) from the current transmit time, then a Poll Request is generated. This is designed to ensure that the receiving end is polled at least once before segments are discarded. The discard latency of any segment shall not be set to less than (3 × ResponseTimeParam).

For an I-frame with new data, the Poll flag is only set if the data segment is also the last or only segment of a PDU (i.e. it is a eom or ssm).

If the I-frame is used for a retransmission, and if it is the last segment in the Retx list, then a Poll request is generated. This will update the Polling State and the Poll Flag of the I-frame shall be set accordingly. When this I-frame is sent, the ARQ Timer shall be re-started with VT set to the latest transmitted sequence number (VS-1).

To complete the ARQ part of the I-frame, the sequence number of the next expected segment (i.e. VR) is stored in the nr field. This is to act as an acknowledgement.

## 5.3.5 Receiving I-Frames

When an I-frame is received, the value of nr and ns is first checked to ensure that the received I-frame is valid (i.e. VS ≥ nr ≥ VA and (VR - 1 + N/2 - 1) ≥ ns ≥ VR, mod N arithmetic applies (see note)). If the I-frame is invalid, then it shall be discarded. Otherwise, the Poll Flag of the received segment is checked. If the flag is set, then a Poll has been received, and a Poll response shall be scheduled. All items in the SREJ waiting list are also transferred to the SREJ request list, i.e. all missing segments are being re-requested. The nr field is also checked to identify any newly acknowledged (nr-1) segments.

NOTE: X > Y mod N is defined as ((N/2 > (X - Y) > 0) OR ((Y - X) ≥ N/2))
X ≥ Y mod N is defined as ((N/2 > (X - Y) > 0) OR ((Y - X) ≥ N/2) OR (X == Y)).

When an I-frame is received, the sub-segment reassembly buffer shall also be cleared of any sub-segments. This shall be done regardless of whether the sequence number of the I-frame matches that of those sub-segments in the buffer. The data-seg part of the PDU can now be de-ciphered.

The sequence number of the received I-frame (ns) is then checked against VR, to see if it is the next one expected to be received. If this is the next expected segment it is added to the Reassembly Buffer via the reassembly routine and the value of VR is updated. The missing segments list is then checked to see if a retransmission request for this segment has been scheduled; if so it is cancelled (i.e. removed from the SREJ request list or SREJ waiting list). An updated status message is then sent to the bearer control layer with the latest queue size information, including the revised outstanding retransmission requests.

A check is performed to see if there are any contiguous segments previously received, i.e. to see if the new next expected segment has already arrived, in which case it is added to the Reassembly Buffer via the re-assembly routine, the value of the sequence number of the next segment expected by this end (VR) is updated, and the check is repeated.

If this is not the next expected segment, then it is added to the RxWindow buffer. It should be noted that the initial check on the value of ns already ensures that it is within the RxWindow size limit. A check is then carried out to see if a retransmission request for this segment has been scheduled; if so it is cancelled (i.e. removed from the SREJ request list or SREJ waiting).

If the AM_DH is configured with InSeqDelv set to "false", then the RxWindow shall be scanned after the segment is stored. If a complete PDU can be formed then it is assembled and delivered to the higher layer. In effect, the PDU shall be delivered out-of sequence.

If ns is higher than those previously received then all the segments between the highest previously received and ns are deemed to be missing. These are added to the SREJ request list, except for those already in the SREJ waiting list. The reasons for using the next highest value previously received rather than VR are twofold. Firstly, errors may occur in bursts, and therefore even though values prior to the currently received ones are missing it is not necessarily true that all values between this and VR are missing. This mechanism therefore avoids requesting segments that have actually arrived. Secondly, since the calculation is done each time an out of order segment is received, it avoids segments being requested multiple times unnecessarily.

If the list of outstanding retransmission requests has changed, then the queue size information at the bearer control layer shall be updated (via STATUS-SAP). The delivery time used for these requests shall always be immediate to ensure the request is sent as quickly as possible.

## 5.3.6 Polling Mechanism

### 5.3.6.0 General

The satellite radio interface ARQ mechanism uses a system of Poll / Poll Response to ensure that acknowledgements are sent at regular intervals. Apart from the Poll and Poll Response messages, the system also utilizes an ARQ Timer and internal state variables to operate the Polling mechanism. The following clauses describe the operations of these components.

### 5.3.6.1 ARQ Timer

The ARQ Timer is used to control the sending of the Poll and the receiving of the Poll response. A Poll can be sent either via the pf field in an I-frame or a Sub-segment frame, or as a RR Poll message. Whenever a Poll is sent, the ARQ Timer is started and the sequence number of the latest transmitted segment is recorded (in VT = VS - 1). In the case where the I-frame carries a new data segment, the value of VT is the same as ns. The timer is only stopped whenever the corresponding Poll Response is received.

In the RNC to UE direction, the response should be sent in the first burst received from the UE after a round-trip time (propagation delay plus processing delay). The RNC can control the Timer value according to when capacity is allocated to the UE.

In the UE to RNC direction, the value of the ARQ Timer shall be set according to the *ResponseTime* BCn-AVP (see ETSI TS 102 744-3-5 [10]) or the *ForwardQoSControl* BCt-AVP. In this case, more margin is required to take into account the queuing delay, as the UE cannot control when capacity will be allocated to it by the RNC. The ResponseTimeParam value is dependent on the current loading on the channel and also possibly on the Traffic Handling Priority. The RNC shall adjust the value accordingly.

The SREJ mechanism described in clause 5.3.5 will detect all missing data segments during normal data flow conditions. However, as it relies on the reception of a later segment at the receiving end, it might not always detect if consecutive segments at the end of a transmission sequence have been lost. When the final segment in the TxPDU Buffer is being transmitted, a request is generated to send a Poll along with the segment. This allows the receiving end to detect any missing segments, and request their retransmission via a SREJ Poll Response. By sending a Poll, the sending end ARQ Timer is also reset, thus ensuring that any lost Poll will be retransmitted.

### 5.3.6.2 Sending a Poll

In order to ensure a Poll / Poll Response pair does not go out-of-sync, the system is restricted to sending one Poll at a time. No more Polls shall be sent until a Poll response is received or the ARQ Timer expires. Any Poll request received in the mean time is delayed. Multiple Poll requests during this time shall be combined into one delayed Poll. Figure 5.2 shows the Polling State machine.

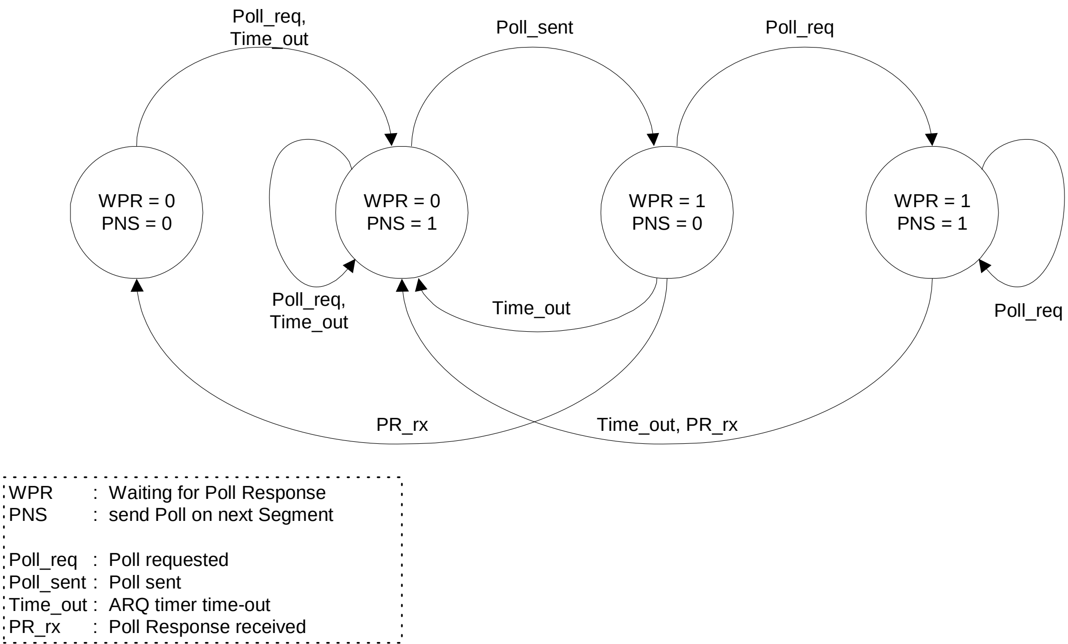


**Figure 5.2: Polling State Machine**

A Poll can only be sent via an I-frame or a Sub-segment frame or an RR-frame by setting the Poll Flag (pf) to 1. An SREJ-frame cannot be used as a Poll. If an I-frame containing new data is to be used for sending a Poll, it shall also be the last segment of a PDU (i.e. an eom or ssm segment). I-frames containing retransmitted segments or Sub-segment frames are not affected by this restriction.

The criteria for generating a Poll request (Poll_req) is:

$((VS - VA) > \frac{TxWindowSize}{4})$ and (Tnow > (rtt + Tva)) OR

(Tnow ≥ (Tda - 2 × Response-Time)) and (Tnow > (Response-Time + Tva)) OR

ARQ Timer expired OR

the next segment is the last in a sequence OR

the next segment is the last retransmission in the Retx list OR

the next sub-segment is the last sub-segment of the last retransmission in the Retx list OR

the TxWindowSize has been reached OR

an invalid SREJ has been received (see clause 5.3.8)

Where *Tva* is the transmit time and *Tda* is the discard time of the oldest un-acknowledged segment, and *Tnow* is the transmit time of the next segment.

Whenever a Poll is sent, the ARQ Timer is restarted and VT is set to VS-1. When the Poll is sent from the RNC to the UE, the RNC should ensure that enough return capacity is allocated to the UE to allow the Poll Response to be sent back within the time limit. This can be achieved by updating the queue status information sent to the Bearer Control Layer.

If the Poll is sent from the UE side, then the Poll Response should be given a short latency, so that it is sent to the UE within the time limit.

As a Poll can be sent together with an I-frame or a Sub-segment frame, there is no need to always request additional capacity from the Bearer Control Layer. Only when the (WPR=0, PNS=1) state is entered does the system check if the data queue is empty. If the data queue is empty, then it shall send an updated queue status to the Bearer Control Layer to include an RR frame. The delivery time used shall be set to immediate.

### 5.3.6.3 Response to a Poll

When a Poll is received, the value of ns is checked against VR. If a new missing segment is detected then it is added to the SREJ request list. Any items that still remain in the SREJ waiting list are also transferred to the SREJ request list. Thus, all missing segments are to be re-requested. A Poll Response is then scheduled (i.e. PR is set to 1, and capacity for the Poll Response is requested).

When a Poll Response is scheduled, either as an SREJ frame or an RR frame, a delivery time of immediate shall be used in the queue status update request message. This is to ensure a quick response for reporting the remote end's status.

A Poll Response is sent using either an RR-frame (pf = 0) or an SREJ-frame (pf = 1). An I-frame or a Sub-segment frame cannot be used to send a Poll Response. If the SREJ request list is empty, an RR-frame shall be used as the Poll Response, otherwise, the SREJ-frame shall be used. The sequence numbers of the missing segments are packed into the SREJ Poll Response in the same way as a normal SREJ-frame, i.e. the oldest sequence number is put into nr first. If all the sequence numbers cannot be packed into the SREJ Poll Response, then they will be carried in the next available slot as a normal SREJ-frame.

### 5.3.6.4 Response to a Poll Response

A Poll Response can occur in two forms, from an RR-frame (pf = 0) or from an SREJ-frame (pf = 1). When an RR Poll Response is received, the value of nr is first checked to ensure that it is valid (i.e. VS ≥ nr ≥ VA, mod N arithmetic applies). If it is not valid, then the message shall be discarded. If the RR Poll Response is valid then nr is compared to VT, and the following applies.

```
If (nr > VT)
    All segments up to (nr-1) are acknowledged, and can be cleared from TxWindow Buffer
    Stop ARQ Timer and update Polling State Machine
else
    All segments up to (nr-1) are acknowledged, and can be cleared from TxWindow Buffer
```

It should be noted that a RR Poll response is only received if there are no missing segments at the receiving end. Therefore if nr ≤ VT, then the Poll response shall be the result of an earlier Poll. Thus, apart from acknowledging up to (nr-1) segments, no other action shall be taken. The current ARQ Timer shall still continue to run.

When a SREJ is received the following algorithm applies.

```
If (SREJ is NOT a Poll Response, i.e. pf = 0)
    All requested segments are put into Retx list, and scheduled for retransmission
else
    All requested segments, except any that were wholly or partially retransmitted since the Poll
                    command, will be retransmitted (i.e. put into Retx list, and the required capacity requested)
    All segments up to (nr-1) are acknowledged, and can be cleared from TxWindowBuffer
    Stop ARQ Timer and update Polling State Machine
```

The Retx list is always sorted such that the oldest segment will always be tested for retransmission first. A Poll request is always generated whenever the last segment (or the last sub-segment of the last segment) in the Retx list is sent (i.e. the list is being emptied).

When retransmission capacity is requested from Bearer Control Layer, the original delivery times of the segment shall be used in the request.

It should be noted that a normal SREJ (pf=0) cannot be used as an acknowledgement, because the list of missing segments in this frame is incremental and not a complete list. Thus, the field nr does not always contain the same value as VR. However, when an SREJ Poll Response (pf=1) is sent, the list is rebuilt and hence the value of nr is always the same as VR. Thus, the SREJ Poll Response can be used as an acknowledgement to segments up to (nr-1).

## 5.3.7 Transmit Behaviour

### 5.3.7.0 General

The main decision tree that applies when a Bearer Connection gets an opportunity to send a frame is shown in Figure 5.3. In general, responses to the remote end are given a higher priority than sending activities.

The first priority when given an opportunity to send a packet is to notify the remote end of any retransmissions it needs to make. The reasons for the highest priority are two-fold. A retransmission request starts the longest individual sequence of communications required to get a data segment across the connection and thus needs to be sent at the earliest opportunity to improve the chances of the required target time for its particular PDU being achieved. In order to avoid stalls in the data pipeline due to window fill, missing data segments shall be retransmitted and acknowledged as quickly as possible and thus shall be requested as quickly as possible.

If the SREJ request list is not empty, an SREJ-frame is constructed and sent. The Poll Flag bit is set if a Poll Response is needed, and the Polling state machine is then updated. As many items as will fit into the offered segment size are then copied into the body of the SREJ-frame from the SREJ request list and the count of items copied is recorded in the num-sel field. All items copied are moved to the SREJ waiting list. The SREJ-frame is then passed to the Bearer control for transmission. The queue size information at the bearer control shall also be updated.

If there was no outstanding retransmission request, then the next priority is to send any Poll Response required. In that case, a RR-frame is constructed with Poll Flag set to 0 and the nr field set to VR.

If no Poll Response is required, the discard time of the oldest un-acknowledged segment is checked. A poll request is generated if the discard time is less than (2 × Response-Time away).

The next priority is to send any outstanding retransmission. The reasons for this are similar to the retransmission requests. A data segment that needs retransmission is by definition already later than originally intended and thus needs to be sent at the earliest opportunity to improve the chances of the required target time for its particular PDU being achieved. In order to avoid stalls in the data pipeline due to window fill, missing data segments shall be retransmitted and acknowledged as quickly as possible.

Each item in the list of required retransmissions is checked to see if their discard time has been exceeded, starting with the one with the oldest sequence number. If so, then an empty segment (i.e. a segment with the same sequence number but containing no data) is sent in its place. If not, then it is checked to see if it will fit the size offered for transmission. If it fits, then it is copied out of the TxWindow buffer into an I-frame and removed from the list of required retransmissions. The current value of VR is then placed in the nr field and the sequence number of the item being sent is placed in the ns field.

If insufficient capacity is available to carry the original segment, then the segment shall be split up into sub-segments (see clause 5.3.3.4) for retransmission. A sub-segment up to the size of the offered capacity is then generated. The remaining data shall be sent in further sub-segments when the next transmit opportunity is available. Each segment can be sub-divided into a maximum of 16 sub-segments. Each sub-segment shall be numbered (from 0 to 15) and the last sub-segment shall be marked with the End-of-Subsegment (eos) flag.

If the capacity available to transmit the $16^{th}$ sub-segment is less than the size of the remaining data, then the sub-segment shall not be sent. The transmitting end shall wait for the next available opportunity, such that all remaining data can be sent on the last possible sub-segment.

If the tBCnID of the connection changes (e.g. due to handover operation) before the last sub-segment is sent, then the remaining sub-segments shall be abandoned. The whole PDU shall be retransmitted again using the new tBCnID. Sub-segments shall be used only if necessary.

A Poll Request shall be generated when the last segment (or the last sub-segment of the last segment) in the Retx list is being retransmitted. The Poll Flag is then set if required, the ARQ Timer started, and the Polling State updated as necessary. The I-frame or sub-segment is then passed to the Bearer Control Layer for transmission.

If there are no data segments to be retransmitted then the presence of data in the Tx PDU buffer is checked to ensure that the window limit has not been reached. The Tx PDU buffer is accessed for the presence of a valid PDU, i.e. one which has not passed its discard time. Any PDUs that have passed their discard time are removed from the Tx PDU buffer and discarded. If a partially sent PDU has passed its discard time, then the remaining segments shall be discarded and an empty eom segment shall be sent to terminate the data PDU.

If no PDU is found and no Poll is required to be sent, then a size of zero will be reported to the Bearer Control Layer. If a Poll is required, then an RR-frame is constructed and sent with the Poll Flag set to 1 and the field ns set to VS-1. If any data has been discarded from the TxPDU or TxWindow buffer, then the Bearer Control Layer shall be updated with the latest queue size information.

If there is new data to be sent and the TxWindow size limit has not been reached, then a segment up to the size of the offered segment size is generated. If this is also the last segment of a data PDU, the returned segment size may be smaller than the offered size. The segment for transmission is placed into an I-frame and a copy of it is placed in the appropriate place in the TxWindow buffer along with the Target time of the PDU it is a part of. The current value of VR is placed in the nr field, the sequence number field ns is set to VS, and VS is then incremented.

If this is the last segment (i.e. the TxWindow buffer is empty) or if the number of the un-acknowledged segment is larger than a quarter of the TxWindowSize, then a Poll request is generated and the Polling State is updated. If the new segment is an eom or ssm, then both the Poll Flag and the ARQ Timer are set if required. The I-frame is then ciphered and passed to the Bearer Control Layer for transmission, and the Polling state is updated.


**Figure 5.3: Transmit Opportunity Behaviour**

### 5.3.7.1 Support for final-seg-retx-allowed (fsr) Mode

If the UE supports the final segment retransmit "fsr" mode then the UE may use this facility if the Bearer Control process signals that the Bearer being used for the transmission supports the this mode of operation.

A UE supporting RI-Version 0x83 or later (see ETSI TS 102 744-3-5 [10], clause 6.1.2.2) shall support the final segment retransmit "fsr" mode of operation.

When operating in "fsr mode" ("fsr" flag is set), and an ARQ timer expires in the UE, the UE may retransmit the final un-acked data segment instead of a Receive Ready frame to Poll the RNC. This shall only be done if:

1) There is neither outstanding retransmission nor new data in the TxPDU buffer (for any connection).

2) There is at least one un-acknowledged data segment.

3) There is sufficient space in the return burst to accommodate the whole of the last un-acked segment.

The UE may additionally back-fill any other un-acked segments within the available return burst capacity. All retransmitted segments shall be transmitted in the correct sequence within the burst, with the poll flag set in the final data segment. The ARQ Timer shall be restarted, and the procedure terminated.

It should be noted that this mode only allows the retransmission of segments in a single return burst, where otherwise the residual capacity in this burst would have been wasted.

## 5.3.8 Receive Behaviour

### 5.3.8.0 General

The main decision tree that occurs when an ARQ bearer connection receives a segment from the Bearer Control Layer is shown in Figures 5.4 and 5.5.

The receive behaviour depends on whether the BCnPDU contains an I-frame or a sub-segment (see Figure 5.4), or an RR or SREJ supervisory frame (see Figure 5.5).

If the received segment is a retransmission request (SREJ), then all the sequence numbers in the SREJ list shall be checked first. If any of these is invalid (i.e. $VS > ns \geq VA$, mod N arithmetic applies), then the whole SREJ is discarded and a poll request is generated in order to protect against the reception of an erroneous SREJ message.

The sequence number in the nr field and the sequence numbers from the body of the message shall then be added to the Retx list. The RetryCount value of each of these segments shall be incremented, except for those segments whose sequence number is already in the Retx list. The Bearer Control Layer shall be updated with the new queue size including all of the retransmissions requested. The new RetryCount of the head segment of the queue shall also be reported.

If the SREJ segment also acts as a Poll Response (i.e. the Poll Flag is set), then any segments that were retransmitted since the Poll are excluded from the Retx list. The RetryCount value of all other segments shall be incremented except for those segments whose sequence number is already in the Retx list. The ARQ Timer shall be stopped and the Polling state updated. This segment also acknowledges the correct reception of segments up to and including (nr-1), thus those segments can be removed from the TxWindow Buffer and the window is moved forward. This shall include any segments that have been marked for retransmission. The Bearer Control Layer shall be updated with the new queue size, and the RetryCount of the head segment of the queue shall also be reported.

When an RR-frame is received, the Poll Flag is checked to identify whether it is a Poll or Poll Response. If an RR Poll is received, then the value of ns is checked to ensure that it is valid (i.e. $(VR-1+N/2-1) \geq ns \geq (VR-1)$, mod N arithmetic applies). If the value of ns is invalid, then the messages shall be discarded. If the message is valid, then any sub-segments remaining in the sub-segment reassembly buffer shall be discarded. The sequence number ns of the RR Poll is then compared to (VR-1). If they are not equal, then the segments from VR to ns are declared missing, and added to the SREJ request list. All other items in the SREJ waiting list are also merged into the SREJ list and sorted. A Poll Response is then scheduled.

If the RR frame is used as a Poll Response (pf = 0), then the behaviour is as described in clause 5.3.6.4. If the segment is an I-frame, then the behaviour is as described in clause 5.3.5.

If a sub-segment frame is received, then the values of nr and ns are first checked to ensure that the message is valid (i.e. $VS \geq nr \geq VA$ and $(VR - 1 + N/2 - 1) \geq ns \geq VR$, mod N arithmetic applies). If the message is invalid then it shall be discarded. If the message is valid, then its header shall be examined and the pf and nr field extracted. These shall be processed in the same way as for an I-frame (see clause 5.3.5). The sub-segment shall then be added to the Sub-segment reassembly buffer.

If this is not in-sequence or if it is a duplicate, then all the sub-segments already in the buffer shall be discarded. Furthermore, if the received sub-segment is not the first sub-segment (i.e. ssegn > 0), then it too shall be discarded. If it is the first sub-segment, then it shall be stored in the reassembly buffer. These procedures ensure that if any sub-segment is missing, then all sub-segments are discarded.

If the received sub-segment is an in-sequence sub-segment, then it is appended to the reassembly buffer. Furthermore, if it is the last sub-segment (eos field is set to "1"), then the completed segment has been recovered. The completed segment shall then be passed to the de-ciphering process and be treated as a normal retransmitted I-frame.

### 5.3.8.1 RNC Support for final-seg-retx-allowed ("fsr") Mode

When handling a UE which supports RI-Version 0x83 or later (see ETSI TS 102 744-3-5 [10], clause 6.1.2.2) , the RAN shall consider "fsr" mode active for all AM connections to the UE.

When operating in "fsr" mode, the RNC may receive duplicate segments within a single burst transmitted by the UE, for instance if a previous RR poll response was missed by the UE. Duplicate segments shall be discarded, however when this mode is enabled in a connection, the RNC shall also generate a RR poll response when a duplicate of the last received segment (i.e. sequence number equals vr-1) is received with its poll flag set. The behaviour is shown in Figure 5.4.

This mode is particularly useful to reduce latency when the connection is being served using the controlled random access mode.

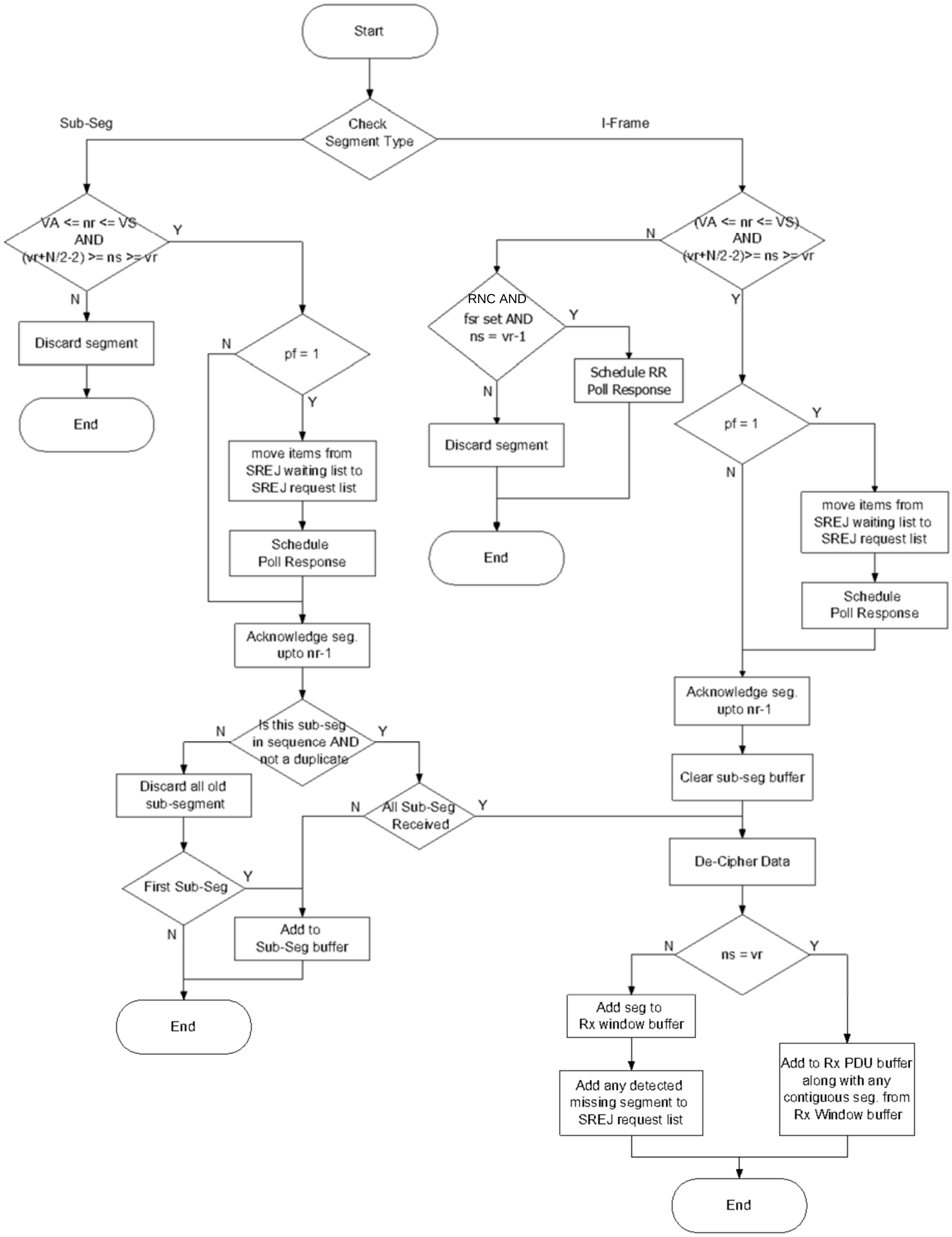


**Figure 5.4: Segment Reception Behaviour (I-frame and Sub-segment)**

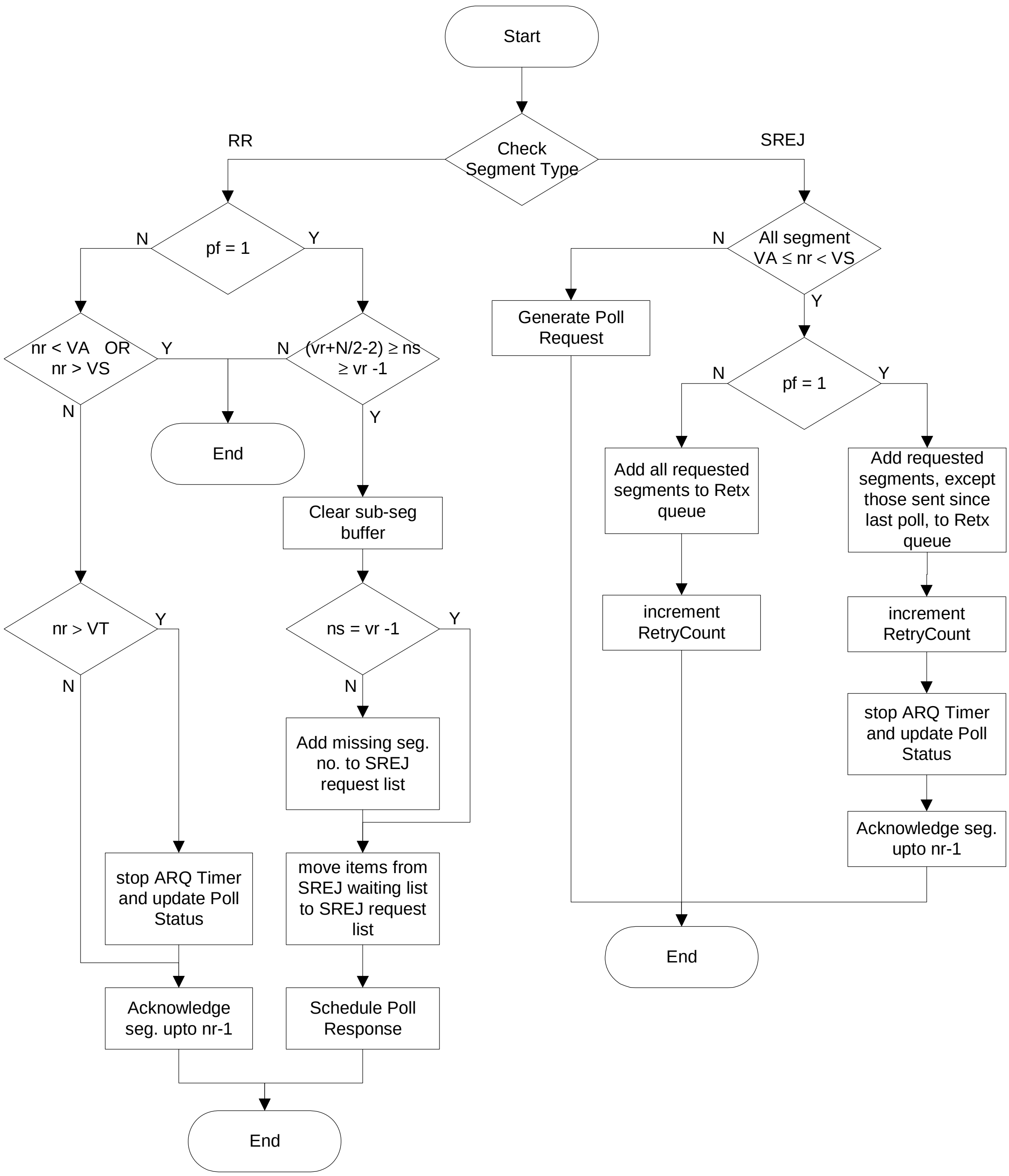


**Figure 5.5: Segment Reception Behaviour (RR and SREJ)**

## 5.3.9 Tx and Rx Window Size Control

For an Acknowledged mode connection, the transmit window size can be a limiting factor in achieving the maximum throughput. The minimum window size is given by:

$$TxWindowSize_{\min} = ResponseTimeParam \bullet \frac{AvDataRate}{AvSegmentSize}$$

where *ResponseTimeParam* is the value used to set the ARQ timer as specified in either the *ResponseTime* BCn-AVP or *ForwardQoSControl* BCt-AVP and is used to set the ARQ timer. AvDataRate and AvSegmentSize are the average data rate and average segment size respectively of the connection.

For example, for a single UE operating in a spot beam occupying the full channel and transmitting R20T4.5X bursts with highest possible coding rate (H4), a window size of approximately 255 segments (with an average segment size of 186 octets and response time of 0,85 sec) is required to sustain the connection at full rate (448 kbit/s). If the loss of one burst is considered, then the size of window buffer required is 510 segments.

As the ARQ mechanism is driven by the transmitting end, the RxWindowSize of the receiver should be the same as the TxWindowSize of the transmitter.

The UE shall provide the necessary window buffer to sustain the connection at the requested QoS. It should be noted that a UE specific signalling connection generally does not require the maximum level of buffering, as it is required to handle only a limited data volume. Furthermore, not all applications require full rate in both directions. Therefore, there is the possibility for the UE to optimize its memory usage whilst maintaining the QoS. However, this shall be done without affecting the achievable throughput specified by the QoS parameters.

## 5.3.10 Acknowledged Mode Timers

The timers used in the acknowledged mode Bearer Connections are summarized in Table 5.1.

**Table 5.1: Acknowledged Mode Timers**

| Timer Name | UE | RNC | Operation |
|---|---|---|---|
| ARQ Timer | Yes | Yes | This timer is used to ensure that the transmitting end of a connection receives up-to-date acknowledgements. The timer is started whenever a Poll is transmitted in order to force an acknowledgement. On timer expiry, a Poll request shall be generated in order to trigger a retransmission of the Poll. When there are no more outstanding acknowledgements, the timer is stopped. |
| MaxIdle Timer | Yes | Yes | This timer is used to ensure that un-utilized connection resources are removed. The MaxIdle Timer shall be started at the beginning of the connection and shall be reset whenever user or signalling (i.e. RR (see note) or SREJ) data is transmitted or received. If the timer expires, the connection shall be closed down by signalling the Adaptation Layer with a CBCn_FAILURE_IND. If either the transmit or receive buffer is non-empty or any signalling data (RR or SREJ or poll or poll response) is outstanding when the timer expires, then ConnFailed shall be returned as the failure cause, otherwise MaxIdleReached shall be returned. |
| ConnFailure Timer | Yes | Yes | This timer is used for the UE Specific Signalling connection only. It detects the failure of the opposite end of the connection and reports the failure to the Adaptation Layer. The ConnFailure Timer shall be started when data is added to either the transmit or receive buffer or if a poll is received. The timer shall be reset whenever user or signalling (i.e. RR (see note) or SREJ) data is transmitted or received. The timer shall be stopped when all of the connection's receive and transmit buffers are empty and the ARQ timer is not running and no poll or poll response is pending. If the timer expires, then the Adaptation Layer shall be signalled using the CBCn_FAILURE_IND with the ConnFailed cause code. |

For the UE specific signalling connection, the MaxIdle Timer is disabled and the ConnFailure Timer shall be used instead. In this case, the ConnFailure Timer shall be programmed with the value specified in the *MaxIdleTime* BCn-AVP.

NOTE: Under exceptional conditions when the opposite stack is no long available, it is possible for the Connection Layer to enter a state where it will transmit an RR Poll indefinitely. To improve the robustness of the Connection Layer stack, the following can optionally be implemented:

*When a RR Poll message is transmitted, the connection layer can optionally NOT reset the MaxIdle or ConnFailure timer.*

If this is implemented, then the connection layer would only send the RR Poll for a maximum period equal to the MaxIdle time duration. When the timer expires, the ConnFailed condition would be declared and the connection removed.

## 5.3.11 Example Retransmission Sequence

This clause gives an example of a possible retransmission sequence. Figure 5.6 illustrates a basic sequence with a single direction data transfer from the Bearer Connection Layer at one end A (BC_A) to the corresponding Bearer Connection Layer at the other end B (BC_B). This retransmission sequence demonstrates the use of normal error detection and retransmission requests.

Two new PDUs arrive at BC_A and transmission space for the appropriate segments is requested. BC_A is given opportunities to transmit such that each PDU is split into 4 segments. In this example, the TxWindowSize is assumed to be 24. When the first segment of the first PDU is transmitted, the sequence number of the frame is set to 0, and BC_A's VS advances to 1.

When frame 0 arrives at the BC_B end, it is added to the Reassembly buffer and VR advances to 1. Similarly, after frame 1 is received at BC_B, it is added to the Reassembly buffer and VR advances to 2.

Frame 2 from BC_A is lost during the transmission. This is detected when frame 3 arrives at BC_B. Frame 2 is added to the SREJ request list. Frame 3 is stored in the RxWindow buffer awaiting re-assembly, while VR remains at 2. BC_B requests transmission space to send an SREJ.

When BC_A transmits frame 3, no poll request is generated, even though it is the last segment of a PDU. This is because the TxPDU buffer is not empty but contains another PDU.

Frame 4 is also lost during the transmission and this is detected by BC_B when frame 5 is received. Frame 4 is added to the SREJ request list while frame 5 is stored in the RxWindow buffer. VR will remains at 2.

At some point BC_B is given an opportunity to send the SREJ to request retransmission of frame 2 and 4. BC_B then moves frames 2 and 4 from SREJ request to the SREJ waiting list.

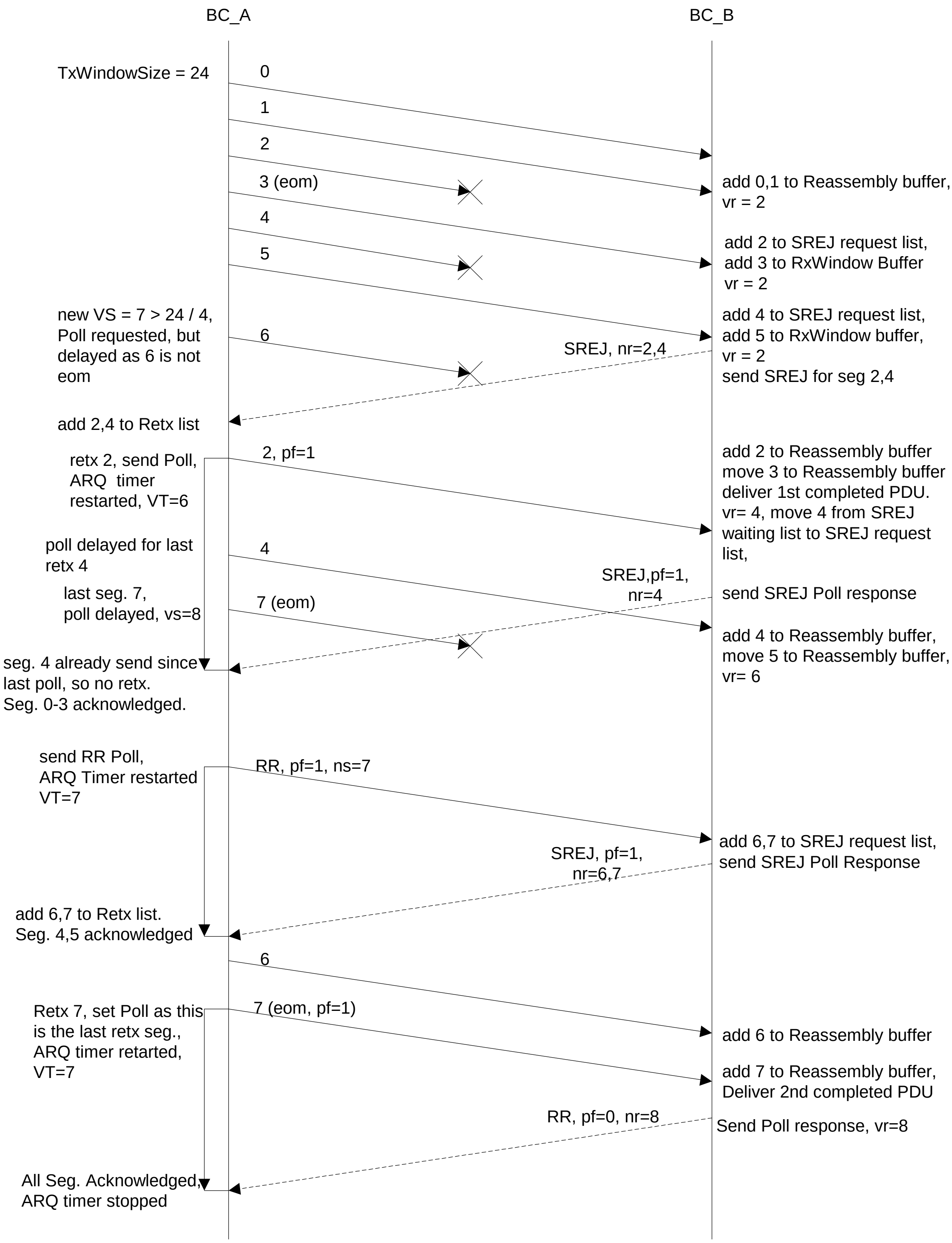


**Figure 5.6: ARQ Behaviour Example**

The transmission of frame 6 from BC_A is also lost, and VS has now advanced to 7. As the VS value increases to 7, which is larger than (TxWindowsSize / 4), a Poll request is generated by BC_A. However, a Poll is not transmitted with frame 6 because it is not the last segment of a PDU.

After the transmission of frame 6, the SREJ frame arrives from BC_B requesting the retransmission of frames 2 and 4. Thus, on the next transmit opportunity, BC_A retransmits frame 2 with the Poll Flag set to 1 and restarts the ARQ Timer with VT=6. As this is not a new segment, it can carry the poll flag even if it is not an ssm or eom segment.

When the retransmission of frame 2 arrives at BC_B, it is added to the Reassembly buffer. As frame 3 is already received, it is now moved from the RxWindow buffer to the Reassembly buffer. The Reassembly buffer now contains the first completed PDU and can be delivered to the higher layer. VR can now be moved to 4.

As frame 2 is also a Poll, the remaining item (frame number 4) in the SREJ waiting list is moved to the SREJ request list and a Poll response is scheduled. At the next transmit opportunity, an SREJ Poll response is sent with nr = 4.

When BC_A retransmits frame 4, a poll request is generated as a result of frame 4 being the last one in the Retx list. However, a Poll is not transmitted with frame 4 because a poll response is still pending.

After the retransmission of frame 4 is received by BC_B, it is added to the Reassembly buffer. Frame 5 is also retrieved from the RxWindow buffer and added to the Reassembly buffer. VR now advances to 6.

After retransmitting frame 4, BC_A sends frame 7, which is also lost during the transmission. As this is the last segment of the PDU, a Poll request is generated. However, as a Poll is already in progress, (i.e. BC_A is still waiting for a Poll response) this Poll request is delayed and is effectively combined with the last delayed poll request.

When BC_A receives the SREJ Poll response with nr=4, no retransmission is generated since frame 4 has already been retransmitted since the last Poll command. The ARQ Timer is also stopped when the Poll response is received. As this is a Poll response, the value of nr also acts as an acknowledgement, thus segments 0 to 3 can be removed from the TxWindow buffer.

As there are no more data segments to send, an RR Poll is generated at the next transmit opportunity. The sequence number of the RR Poll is set to VS-1=7, and the ARQ Timer is also restarted, with VT=7.

When BC_B receives the RR Poll, it detects from the sequence number that frames 6 and 7 are missing, and generates a SREJ Poll response for the next transmit opportunity.

When the SREJ Poll response returns to BC_A, it triggers the retransmission of both frames 6 and 7. A Poll is also scheduled with the retransmission of frame 7. The SREJ Poll response also acknowledged frames 4 and 5, and they can be removed from the TxWindow buffer.

When both retransmitted frames 6 and 7 are received by BC_B, they are move to the Reassembly buffer. With the reception of frame 7, the second PDU is completed, and this is therefore delivered to the upper layer. An RR Poll response is generated with nr=8.

When BC_A receives the RR Poll Response, both data PDUs are acknowledged, and the ARQ Timer can be turned off.

# 5.4 Transparent Mode Operations

## 5.4.0 General

In transparent mode, data shall be transported across the connection layer without adding any extra PDU overhead. The architecture of the transparent mode data handler (TM_DH) is shown in Figure 5.7.

The TM_DH only performs buffering and possibly segmentation operations. On the transmit side, the data PDU received from the layer above is buffered at the Input TxPDU buffer. When transmit capacity is available, the correct amount of data is extracted from the TxPDU buffer and sent to the BCt-DATA-SAP.

If the data discard timer expires before any part of a higher layer data PDU is sent, then it is removed from the TxPDU buffer. Furthermore, if the higher layer had requested notification of data being discarded, then a BCn_TM_DATA_CNF shall be generated and sent to the higher layer.

If the discard time of a data segment in the TxPDU buffer expires before the whole PDU is sent (i.e. it is only partially transmitted), the remaining data shall NOT be discarded in order to minimize the possibility of the system losing synchronization.

In the receive direction, transparent data segments received from BCt-DATA-SAP are passed directly to the layer above. If the Bearer Control Layer has signalled that the received BCn-PDU is corrupted, then TM_DH shall set the Err_ind flag when delivering the PDU to the higher layer using BCn_TM_DATA_IND.

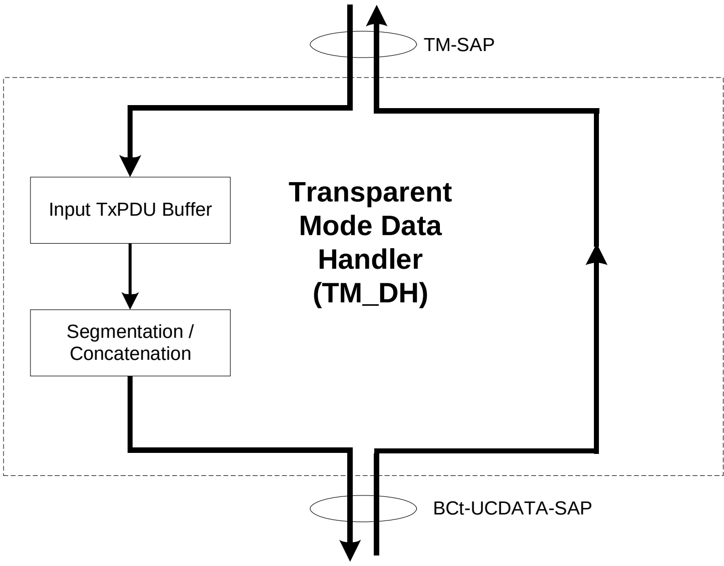


**Figure 5.7: Model of a Transparent Mode Data Handler**

In transparent mode, there is no default data PDU size. Thus, segmentation (or concatenation) is implicit as a result of available transmit capacity, which may vary. The higher layer is responsible for recovering the correct PDU size, even after a loss of data. The higher layer can, if required, enforce a fixed segmentation size by using the Unsegmentable parameter in the BCn_TM_DATA_REQ or by setting a default Unsegmentable size at the creation of the TM_DH. This will ensure that TM_DH always sends only completed PDUs and always delivers one or more complete PDUs to the higher layer.

Ciphering of transparent mode data is not carried out in the Connection Layer but in the Bearer Control Layer instead. After creating a transparent connection, the BCn layer on the RNC side shall discard any received segments until the Radio Bearer Control (RBC) instructs it to begin forwarding segments to the upper layer. When the RBC receives the EstablishAck for the connection, it shall instruct the control layer to set up the ciphering, before instructing the BCn layer to start forwarding all received segments.

It should be noted that when the TM_DH is used for transporting Circuit Switched (CS) voice and data (ISDN), the connection buffer should be integrated into the CS data handler to minimize any delay in the transmission path.

## 5.4.1 Transparent Mode Timers

The timers used in the transparent mode Bearer Connections are summarized in Table 5.2.

**Table 5.2: Transparent Mode Timer**

| Timer Name | UE | RNC | Operation |
|---|---|---|---|
| MaxIdle Timer | Yes | Yes | This timer is used to ensure that un-utilized connection resources are removed. The MaxIdle Timer shall be started at the beginning of the connection and shall be reset whenever data is transmitted or received. If the timer expires, the connection shall be closed down by signalling the Adaptation Layer with a CBCn_FAILURE_IND. If either the transmit or receive buffer is non-empty when the timer expires, then ConnFailed shall be returned as the failure cause, otherwise MaxIdleReached shall be returned. |

# 5.5 Un-Acknowledged Mode Operations

## 5.5.0 General

In un-acknowledged mode operation, no ARQ is needed, but segmentation, ordered delivery and ciphering are required. This is supported in the Bearer Connection Layer by the Numbered Frame Mode Data Handler (NUM_DH). The architecture of the NUM_DH is shown in Figure 5.8.

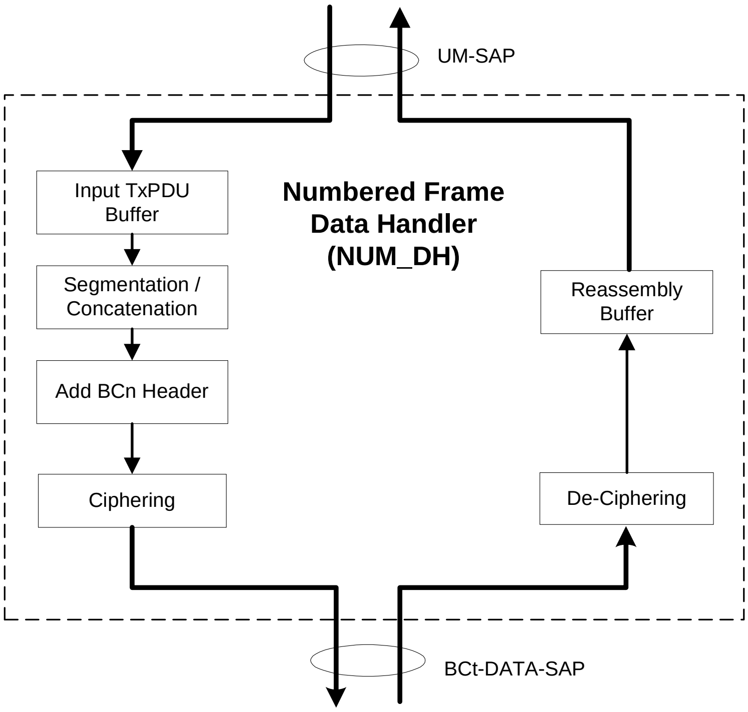


**Figure 5.8: Model of a Numbered Frame Mode Data Handler**

In the transmit path, incoming data PDUs from the higher layer are first buffered in the Input TxPDU Buffer. These PDUs are all time-stamped and the updated queue size is sent to the Bearer Control Layer via the BCnM. When transmit capacity is available, data is then extracted from the TxPDU Buffer and segmented, before the BCnPDU header with the correct sequence number is added. The data part of the resulting BCnPDU is then ciphered, before the BCnPDU is passed to the Bearer Control Layer via the BCt-DATA-SAP.

If there is insufficient transmit capacity available, then this might result in the discard timer of some of the buffered PDUs expiring, or the buffer could be full. In these cases, the NUM_DH shall remove these PDUs from the TxPDU buffer. Furthermore, if the higher layer has requested discard notification (DiscardReq set), then a BCn_UM_DATA_CNF signal shall also be sent with the corresponding MUI.

If the discard time of a data segment in the TxPDU buffer expires before the whole PDU is sent, the remaining data shall be discarded. However, when the next opportunity to send data arrives, an empty End-of-Message (eom) segment shall be transmitted to allow the termination of the data PDU at the receiving end.

If the discard time of a data PDU in the TxPDU buffer expires before any segment can be sent, then the whole PDU shall be discarded from the TxPDU buffer. It is assumed that the higher layer protocol handles this failure scenario.

In the receive path, received BCnPDUs are first de-ciphered before being stored in the Reassembly Buffer. Once a complete un-acknowledged mode data PDU is received at the Reassembly Buffer, it shall be forwarded to the higher layer.

If one or more segments of a PDU were lost, then this is apparent from sequence numbers being out-of-sequence. In this case, all affected PDU(s) shall be discarded from the Reassembly buffer. Furthermore, if there is a mismatch between the declared length of the PDU and the actual data received, then the received PDU shall be discarded. This can occur as a result of the PDU's discard timer expiring at the sender's end.

### 5.5.1 Un-Acknowledged Mode Timers

The timers used in the un-acknowledged mode Bearer Connections are summarized in Table 5.3.

**Table 5.3: Un-Acknowledged Mode Timer**

| Timer Name | UE | RNC | Operation |
|---|---|---|---|
| MaxIdle Timer | Yes | Yes | This timer is used to ensure that un-utilized connection resources are removed. The MaxIdle Timer shall be started at the beginning of the connection, and shall be reset whenever data is transmitted or received. If the timer expires, the connection shall be closed down by signalling the Adaptation Layer with a CBCn_FAILURE_IND. If either the transmit or receive buffer is non-empty when the timer expires, then ConnFailed shall be returned as the failure cause, otherwise MaxIdleReached shall be returned. |

## 5.6 Common Signalling Data Handler Operation

The Bearer Connection Layer provides a buffering and status reporting service for the handling of Common Signalling Data. No ciphering is required. The architecture of the Common Signalling Data Handler is shown in Figure 5.9.

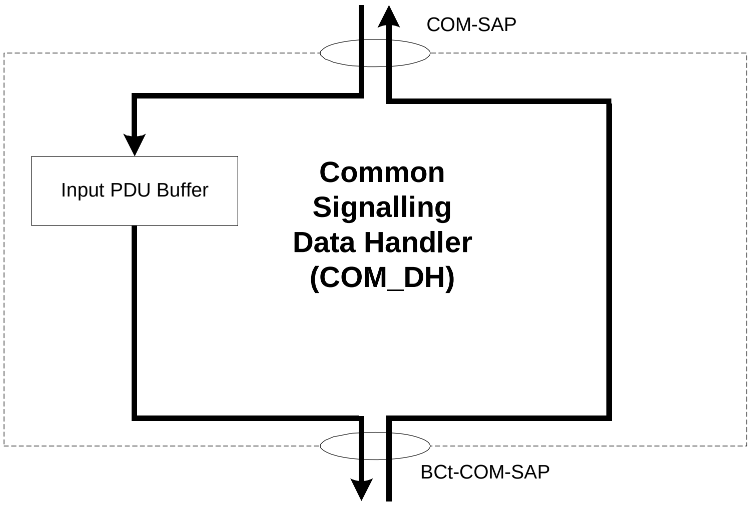


**Figure 5.9: Common Signalling Data Handler**

Common Signalling PDUs are first buffered at the Input PDU Buffer. The change of buffer status shall be reported to the Bearer Control by the Bearer Control Manager (BCtM) via the STATUS-SAP. On the UE side, this shall include the RetryCount parameter (obtained from the Adaptation Layer) which is used to indicate that the PDU is being retransmitted. When capacity is available, the Common Signalling messages shall be sent via the BCt-COM-SAP. Common Signalling Messages shall not be segmented.

On the receive path, any Common Signalling Messages received are sent to the Adaptation Layer.

## 5.7 Ciphering Operation

### 5.7.0 General

Ciphering is only performed in the Bearer Connection Layer in Acknowledged Mode and Un-Acknowledged Mode operation. For Transparent Mode operation, ciphering is performed in the Bearer Control Layer instead.

Ciphering shall be implemented in accordance with ETSI TS 133 102 [2] and ETSI TS 133 105 [3], however, some minor modifications are required to accommodate the differences between UTRAN Radio Link Control (RLC) and the satellite radio interface Bearer Connection Layer. Figure 5.10 illustrates the ciphering and deciphering process.

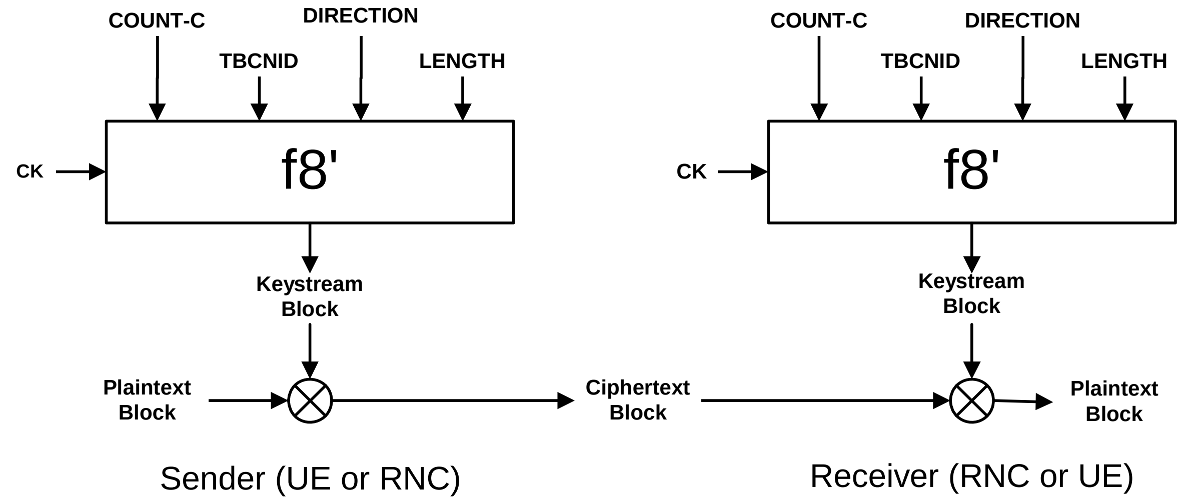


**Figure 5.10: Ciphering of user and signalling data transmitted over the satellite radio interface**

The standard UMTS KASUMI ciphering algorithm (f8) shall be used, however, the input parameters have to be modified to match parameters readily available in the satellite radio interface protocol. Hence, the algorithm is referred to as f8'. The parameters COUNT-C (see clause 5.7.1.2) and TBCNID (see clause 5.7.1.3) are modified to accommodate the satellite radio interface protocol, while all other input parameters are applied as specified in [2].

After creating a connection, the BCn layer on the RNC side shall buffer any received segments until the RBC instructs it to begin processing. When the RBC has received the EstablishAck for the connection, it shall set up the cipher instructions before instructing the BCn layer to start processing all received segments.

## 5.7.1 Input Parameters to the Ciphering Algorithm

### 5.7.1.0 General

It can be seen from Figure 5.10 that there are five input parameters to the ciphering algorithm, namely CK, COUNT-C, TBCNID, DIRECTION and LENGTH. These are described in detail below.

### 5.7.1.1 CK

The cipher key CK is 128 bits long and shall be applied as specified in [2]. No modification is required for the satellite network. This value is supplied by the AL via the CBCn-SAP using the CBn_SECURITY_REQ message.

### 5.7.1.2 COUNT-C

The ciphering sequence number COUNT-C is 32 bits long and is modified for the satellite network to reflect the counters available in the satellite radio interface protocol stack.

In Acknowledged Mode (AM) and Unacknowledged Mode (UM) COUNT-C is composed as follows:

- The 10 least significant bits of COUNT-C are equal to the send sequence number (ns), which is present in the BCnPDU Header. This part of COUNT-C is referred to in [2] as the "short" sequence number.
- The remaining 22 most significant bits of COUNT-C are incremented by one each time the "short" sequence number (NS) wraps around (see clause 5.7.3). This part of COUNT-C is referred to in [2] as the "long" sequence number.

| | |
|---|---|
| **BCN HFN (22 bits)** | **NS (10 bits)** |

**Figure 5.11: Structure of COUNT-C for Acknowledged and Un-Acknowledged Mode**

There is one COUNT-C value per uplink (return link) connection and one COUNT-C value per downlink (forward link) connection using numbered information frames (i.e. AM or UM connection).

For Unicast Bearer Connections, when ciphering of a new Bearer Connection is started, the 20 most significant bits of the Bearer Connection Hyperframe Number (BCN HFN) shown in Figure 5.11 are initialized with the current value of the parameter START (see [2], clause 6.4.8) while the two least significant bits of BCN HFN are initialized to zero. This approach also applies for the ciphering of the uplink traffic associated with bi-directional MBMS Connections.

For MBMS Connections, when deciphering of a new Bearer Connection is started in a UE, the COUNT-C value shall be initialized to the value transmitted to the UE within the AL Establish message for the downlink (to mobile traffic).

#### 5.7.1.3 TBCNID and BCNID

The translated bearer connection identifier TBCNID is 12 bits long and is available in the BCt-PDU header. This is used to replace the radio bearer identifier BEARER used in the UTRAN ciphering configuration.

For UEs that comply with RI-Version 0x83 or later (see ETSI TS 102 744-3-5 [10], clause 6.1.2.2) the transmitter and receiver shall always use the least significant 12 bits of the BCnID as the TBCNID value in support of ciphering operation, regardless of whether the AM or UM connection is addressed by BCnID or tBCnId.

#### 5.7.1.4 DIRECTION

The direction identifier DIRECTION is 1 bit long. Its value is applied as specified in [2]:

DIRECTION = 0 for messages from UE to RNC

DIRECTION = 1 for messages from RNC to UE.

#### 5.7.1.5 LENGTH

The length indicator LENGTH is 16 bits long. It specifies the length of the required keystream block in bits and does not affect the actual bits in the keystream block.

### 5.7.2 Initialization of Keystream Generator

The initialization of the keystream generator is specified in [4], clause 3.4 for UTRAN. In order to accommodate the input parameters used in the satellite network (as specified in clause 5.7.1), the 64-bit register A shall be set to:

COUNT-C || TBCNID || DIRECTION || 0…0

(left justified with the right most 19 bits set to 0), such that:

A = COUNT-C[0]…COUNT-C[31] TBCNID[0]…TBCNID[11] DIR[0] 0…0

All other requirements of [4] remain unchanged.

### 5.7.3 Maintenance of COUNT-C

This clause specifies the rules that shall be used in order to keep COUNT-C values synchronous at both ends of the connection. In this clause, *ns* (in lower case letters) refers to the sequence number carried in the received BCn-PDU, while *NS* (in upper case letters) refers to the value of NS stored as the ten least significant bits of COUNT-C.

For Unacknowledged Mode connections:

- At the sender, if *ns* (and hence *NS*) wrap around from 0x3FF to 0x000 then BCN-HFN shall be incremented by one.
- At the sender, the 2$^{nd}$ LSB of BCN-HFN is used as a parity bit (*hfn-flag*) and is sent in the BCn-PDU header.
- At the receiver, if the received *hfn-flag* is different to that of the BCN-HFN then the BCN-HFN shall be incremented until the two match. The value of NS shall be set equal to *ns*.
- At the receiver, if the *hfn-flag* matches but the *ns* (in the received frame) is less than or equal to *NS*, then BCN-HFN shall be incremented by one. This comparison shall be made in normal integer arithmetic and NOT in module N (i.e. 1 023 > 1). If the new BCN-HFN does not match the *hfn-flag*, then the BCN-HFN shall be incremented again until it matches. The value of *NS* shall be set equal to *ns*.

For Acknowledged Mode connections:

- At the sender, if *NS* wraps around from 0x3FF to 0x000, then BCN-HFN shall be incremented by one.
- At the receiver, if *ns* (in the received frame) is less than *NS* **and** (*NS* - *ns*) is larger than *tx-window-size* then BCN-HFN shall be incremented by one.
- In the case of a retransmitted segment, the sender shall additionally use the BCN-HFN and *ns* that were used for the original transmission of the segment, for the ciphering of the retransmitted segment, i.e. the ciphering keystream is the same for both the original and the retransmitted segment. The receiver is aware of the fact that a retransmission is taking place and shall use the appropriate input parameters for the keystream generator.

### 5.7.4 MBMS Connections

MBMS Connections are shared between multiple receivers, each of which may join at a different time, and the multiple MBMS Connections associated with the same MBMS Service may be initiated at different times. Each MBMS Connection associated with the same MRAB (Multicast Radio Access Bearer) has the same CK value, and this value is transferred in the AL Establish message. It should be noted that the tBCnID and CountC value are independent for each MBMS Connection, and these parameters are passed in an Embedded BCtPDU during Establish, Modify or Handover. These differences only apply in the forward direction. In the return direction the normal security mechanisms apply.

## 5.8 Handover Operations

Bearer Connections may need to be handed over to a different Bearer Control Process, e.g. when a UE is transferred from a Regional Beam to a Spot Beam (or vice versa), or when traffic within a beam is being redistributed, or when a UE is crossing spot beam boundaries. In these cases, the LRM will initiate the Handover procedure.

The first step in the Handover procedure is to create a UE Control unit (UE_BCt, one per UE) at the target Bearer Control Process. As part of this process, the ability of the target Bearer Control to support the QoS of all the connections shall be checked. In cases where a connection cannot be supported by the target Bearer Control process (e.g. insufficient capacity, unsupported bearer type), it shall first be closed down by the original Bearer Control before the Handover procedure can start.

Once the new Bearer Control information has been obtained, it is sent to the Adaptation Layer to trigger the sending of the *Handover* message over the UE Specific Signalling Connection.

Once the *Handover* message is sent, the RNC shall transfer all connections to the target Bearer Control. The data and states of all connections shall be preserved during the transfer between Bearer Controls. For Acknowledged Mode connections, any retransmission sent on the target Bearer Control, shall be re-ciphered with the new tBCnID. Any incomplete retransmissions being sent with sub-segments shall be abandoned. The whole PDU shall be retransmitted again using the new tBCnID for ciphering.

The details of the Handover operations are described in ETSI TS 102 744-3-2 [8].

Once the UE has received the *Handover* message, it shall immediately retune to the new frequency and acquire the new bearer information. Once the new bearer is acquired, the Adaptation Layer at the UE shall respond with a *HandoverAck* message over the UE Specific Signalling connection. All other connections shall operate as normal over the new bearer. The Handover procedure is only completed when the *HandoverAck* message is received by the Adaptation Layer at the RNC.

For Acknowledged Mode connections, any retransmission sent after the UE retunes to the new frequency shall be re-ciphered with the new tBCnID. Any incomplete retransmissions being sent with sub-segments shall be abandoned. The whole PDU shall be retransmitted again using the new tBCnID for ciphering.

New connections shall only be set up after all existing connections have been successfully handed over to the new Bearer Control.

# History

| Document history | | |
|---|---|---|
| V1.1.1 | October 2015 | Publication |
| | | |
| | | |
| | | |
| | | |