ETSI TS 102 744-3-1 V1.1.1 (2015-10)

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

Reference

DTS/SES-00299-3-1

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 1 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**", "**may not**", "**need**", "**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 Control Layer (BCt) peer-to-peer interface of the Family SL satellite radio interface between the Radio Network Controller (RNC) and the User Equipment (UE) used in the satellite network.

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 125 331: "Universal Mobile Telecommunications System (UMTS); Radio Resource Control (RRC); Protocol specification (3GPP TS 25.331 Release 4)".

[2] ETSI TS 123 003: "Digital cellular telecommunications system (Phase 2+); Universal Mobile Telecommunications System (UMTS); Numbering, addressing and identification (3GPP TS 23.003 Release 4)".

[3] International Telegraph and Telephone Consultative Committee CCITT (now ITU-T) Red Book, Recommendation X.25.

[4] ETSI TS 124 008: "Digital cellular telecommunications system (Phase 2+); Universal Mobile Telecommunications System (UMTS); Mobile radio interface Layer 3 specification; Core network protocols; Stage 3 (3GPP TS 24.008 Release 4)".

[5] "Global Positioning System Standard Positioning Service Signal Specification", 2nd Edition, 2nd June 1995, GPS Navstar Joint Program Office.

[6] 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".

[7] 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".

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

[9] ETSI TS 102 744-2-2: "Satellite Earth Stations and Systems (SES); Family SL Satellite Radio Interface (Release 1); Part 2: Physical Layer Specifications; Sub-part 2: Radio Transmission and Reception".

[10] 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".

[11] 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".

[12] ETSI TS 102 744-3-4: "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".

[13] 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".

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 [7], clause 3.1 apply.

3.2 Abbreviations

For the purposes of the present document, the abbreviations given in ETSI TS 102 744-1-4 [7], clause 3.2 apply.

4 Bearer Control Interface

4.1 Radio Interface Layering

As described in ETSI TS 102 744-1-3 [6], the satellite communication protocol is considered as a number of communication layers, as follows:

- Adaptation Layer (AL);
- Bearer Connection Layer (BCn);
- Bearer Control Layer (BCt); and
- Physical Layer (L1).

The satellite radio interface protocol stack is designed to seamlessly integrate with UMTS Non-Access Stratum entities, such as GPRS Mobility Management (GMM) and Mobility Management (MM), residing in the Core Network (CN) and in the upper layers of the User Equipment (UE).

The Bearer Control Layer is responsible for controlling the access to the physical layer (channel resource) for each of the connections which are established. The present document defines the Bearer Control Layer (BCt) peer-to-peer interface between the Radio Network Controller (RNC) and the User Equipment (UE), as shown in Figure 4.1.

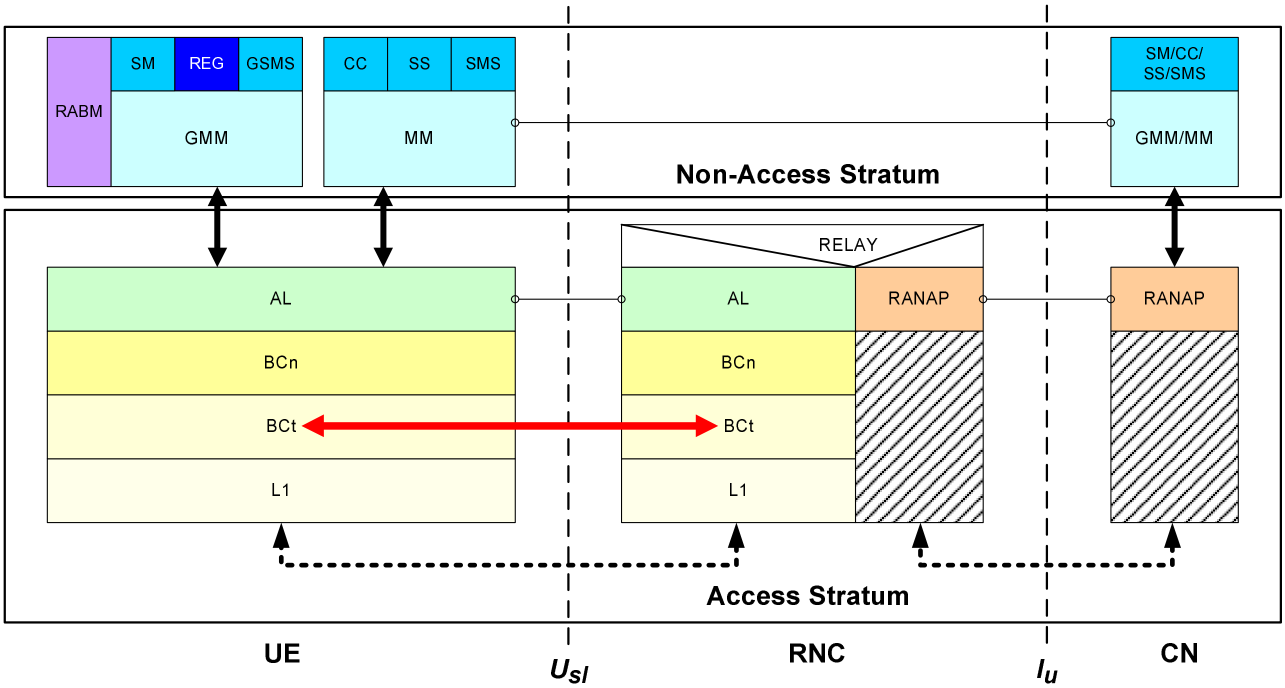


Figure 4.1: Control Plane Protocol Stack Layering with Bearer Control Layer peer-to-peer interface indicated

4.2 Bearer Control Layer

The Bearer Control Layer includes at least one Bearer Control process, which operates over and manages specific Bearer Types (physical layer specifications for the Shared Access Bearer). The detailed behaviour of the Bearer Control process, and the interface definitions for peer-peer communication of the Bearer Control processes are unique to the particular Bearer Control Type, although the Bearer Control Layer as a whole provides the following functionality:

- transfer of Bearer Connection PDUs and Common Signalling PDUs between RNC and UE using the available physical layer capabilities;
- link adaptation to compensate for mobile transmission characteristics as required;
- scheduling of transmissions to match mobile transceiver capabilities and mode of operation (for example sleep mode and multi-channel operation);
- connection admission control for the purposes of determining the available quality of service for a connection which has requested a certain capacity;
- management of satellite resources (by means of allocation and deallocation of physical bearers) in such a way as to keep the risk of failing to meet the quality of service agreed on a connection basis acceptably low whilst keeping overall bearer efficiencies high.

The Bearer Control Layer Protocol Data Unit (PDU) structure and peer-peer Bearer Control Layer Signalling Data Unit (SDU) definitions are described in the present document.

4.3 Conventions used in the present document

4.3.1 Presentation

The following conventions are applied throughout the present document:

- In the ASN.1 notation, variable names are always in lower case letters with hyphenation used to improve readability (e.g. ret-bct-pdu-header). Data Types in the ASN.1 always start with an upper case letter and may contain additional upper case letters to improve readability (e.g. ReturnBCtPDUHeader).
- In the explanatory text, these variables are referred to in italics (.e.g. *ret-bct-pdu-header)*, while Data Types are shown in Helvetica typeface (e.g. BCnPDU).

The layout of the data structures defined in the ASN.1 is also shown in a graphical representation. In general, the variable names are presented in the same way they are presented in the ASN.1, with the following exceptions:

- insufficient space does not allow the complete variable name to be presented and it is therefore abbreviated;
- only one particular value can be assigned to a variable in the particular structure that is presented - in this case the variable is replaced by the appropriate numerical value;
- additional information may be added in brackets for explanatory reasons.

4.3.2 "Reserved" Fields and Values

Fields shown as Reserved BITSTRING (..) in the ASN.1 structures shall be set to zero by the sender and shall be ignored by the receiver.

Values not allocated in Distinguished Value Lists shall not be used by the sender and shall be ignored by the receiver.

NOTE 1: Distinguished Value Lists of type Integer are being used instead of the ENUMERATED data type, where the allocated number range is larger than the number of items to be enumerated.

NOTE 2: It should be noted that UEs may only support a lower RI-Version than the one supported by the RNC (the RI-Version is defined in ETSI TS 102 744-3-5 [13], clause 6.1.2.2). In this case, it is likely that Broadcast SDUs/AVPs transmitted by the RNC contain values that are considered as "reserved" by those UEs.

4.3.3 Boolean Variables

BOOLEAN variables shall be encoded as follows:

```
TRUE    ::= 1
FALSE   ::= 0
```

4.3.4 ASN.1 Encoding Rules

The ASN.1 presentation provided in the present document for this interface specification is normative. The encoding rules used for this interface specification are non-standard, using a structured form of packed encoding that ensures efficient packing of each encoded BCtPDU while maintaining preservation of octet boundaries for key fields. The presence or absence of optional parameters is signalled using flags which are explicitly encoded into the ASN.1 specification, and the number of elements in a list is either explicitly encoded in an ASN.1 specified field, or implicit due to a defined constraint. The encoding is represented in diagrammatic form with examples shown for each BCtPDU. The diagrams represent the encoded data structures and are normative for all data structures specified in the present document.

The encoding of all integers as specified in the ASN.1 into the data structures presented in diagrammatic form shall follow the rules below:

- Unsigned INTEGER values are encoded such that the range of values determines the field width (in terms of the number of bits required to encode the range), with the encoded value of 0 representing the lowest value in the range e.g.:
  - INTEGER(0..7) shall be encoded in three bits with '000' representing a value of 0, '001' representing 1, etc. to '111' representing a value of 7.
  - INTEGER(1..8) shall be encoded in three bits with '000' representing a value of 1, '001' representing 2, etc. to '111' representing a value of 8.
- Signed INTEGER values (i.e. those containing a negative range) shall ALWAYS be represented as two's complement, with the number of bits required for the encoding being one greater than (due to the need for a sign bit) the number of bits required to encode the maximum of either the positive or negative range; e.g.:
  - INTEGER (-8…7) shall be encoded in four bits with '0000' representing 0, '1000' representing -8, and '0111' representing +7.
  - INTEGER (-3…6) shall be encoded in four bits with '0000' representing 0, '1101' representing -3, and '0110' representing +6.

5 Bearer Control Process

5.1 Bearer Control Protocol Data Unit (BCtPDU)

5.1.1 Return Bearer Control PDU Structure (RetBCtPDU)

The Return Bearer Control PDU structure depends on the Return Bearer type used on the Physical Layer as follows.

For all Bearer Types except R80T0.5Q and R80T1Q, each RetBCtPDU consists of a Return BCtPDU Header, an optional sequence of one or more Bearer Control Signalling Data Units (BCtSDUs), an optional BCtPayload and a 16 bit Cyclic Redundancy Check as illustrated in Figure 5.1.

For Bearer Types R80T0.5Q and R80T1Q only, each RetBCtPDU consists of a Return BCtPDU Header, an optional sequence of one or more Bearer Control Signalling Data Units (BCtSDUs), an optional BCtPayload as illustrated in Figure 5.2. The Cyclic Redundancy Check is not present but later added by the Physical Layer per burst.

```
RetBCtPDU ::=
    SEQUENCE {
        ret-bct-pdu-header
            RetBCtPDUHeader,
        bct-sdu-list
            SEQUENCE OF BCtSDU OPTIONAL,
        bct-payload
            BCtPayload OPTIONAL,
        crc
            -- present only for all Bearer Types
            -- except R80T0.5Q and R80T1Q
            INTEGER(0 .. 65535)
    }
```

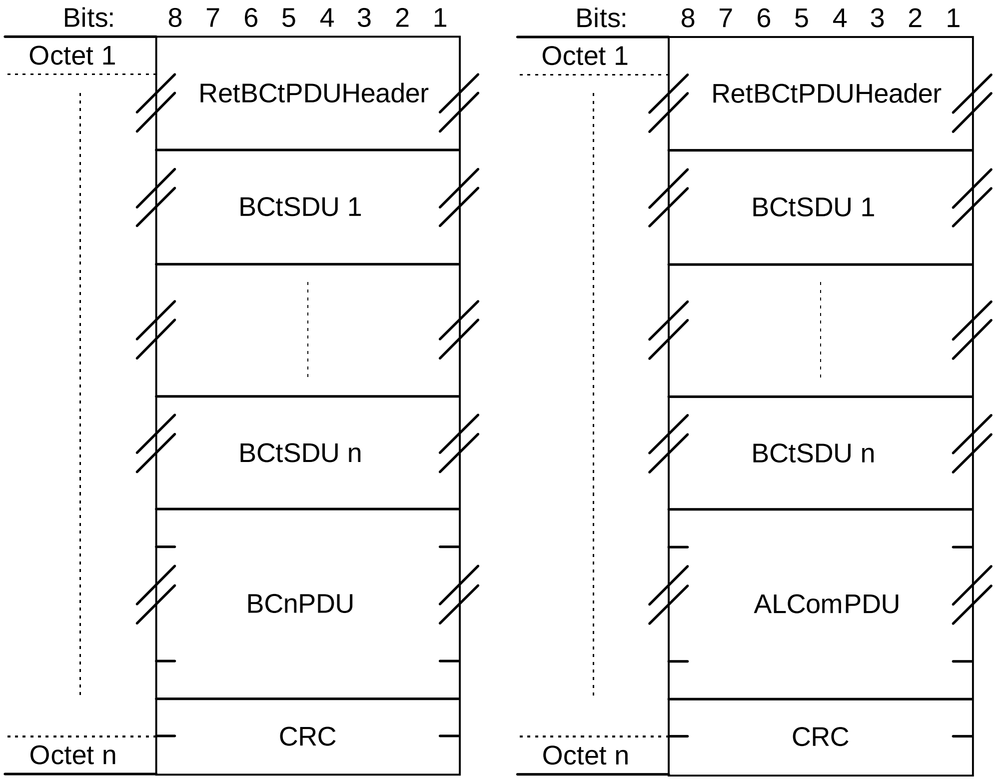


Figure 5.1: Return Bearer Control PDU Structures (all Bearer Types except R80T0.5Q and R80T1Q)

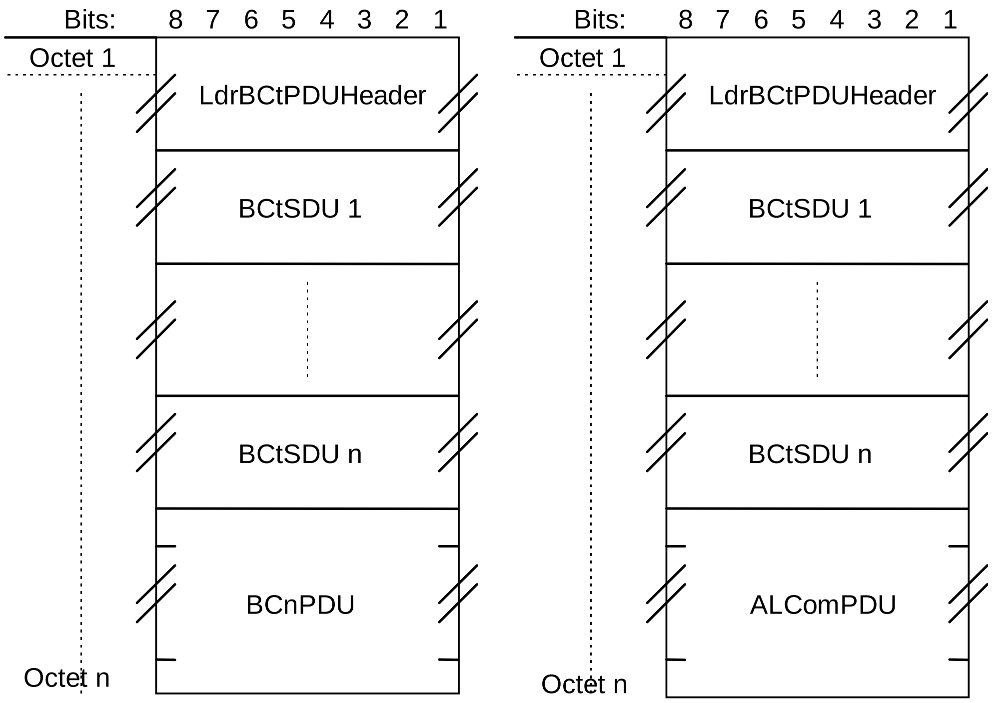


Figure 5.2: Return Bearer Control PDU Structures (Bearer Types R80T0.5Q and R80T1Q only)

Although not more than one BCnPDU or ALComPDU is present at the end of a BCtPDU, it is possible that additional BCnPDUs may be encapsulated within BCtSDUs of type ' BCnPDU'. This is specifically to maintain efficiency of the transport mechanisms when Bearer Connection PDUs contain High Level Data Link Control (HDLC) signalling information.

5.1.2 Forward Bearer Control PDU Structure (FwdBCtPDU)

The Bearer Control PDU has the following structure as shown below and in Figure 5.2a:

```
FwdBCtPDU ::=
    SEQUENCE {
        fwd-bct-pdu-header
            FwdBCtPDUHeader,
        bct-sdu-list
            SEQUENCE OF BCtSDU OPTIONAL,
        bct-payload
            BCtPayload OPTIONAL,
        crc
            INTEGER(0..65535)
    }
```

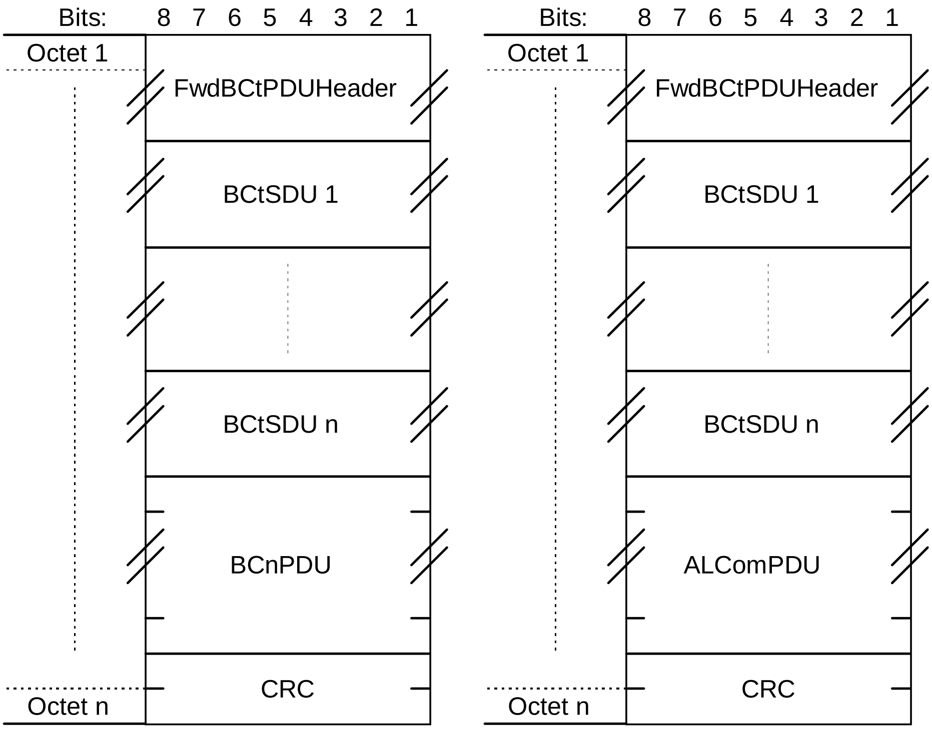


Figure 5.2a: Forward Bearer Control PDU Structures

Although not more than one BCnPDU or ALComPDU is present at the end of a BCtPDU, it is possible that additional BCnPDUs may be encapsulated within BCtSDUs of type ' BCnPDU'. This is specifically to maintain efficiency of the transport mechanisms when Bearer Connection PDUs contain HDLC signalling information.

5.1.3 Return Bearer Control PDU Header Structure (RetBCtPDUHeader)

5.1.3.1 RetBCtPDU Header Structure Type 1

The Bearer Control PDU Header Type 1 shall be used for all BCtPDUs except for the first BCtPDU sent in a Return Block structure (see ETSI TS 102 744-2-2 [9]) using Return Bearer Types R80T0.5Q or R80T1Q. This Bearer Control PDU Header has the following structure:

```
RetBCtPDUHeader ::=
    SEQUENCE {
        bct-sdu-follows
            BOOLEAN,
        length-present
            BOOLEAN,
        comsig-or-ext-addr-present
            BOOLEAN,
        bearer-number-present
            BOOLEAN,
        header-structure
            CHOICE {
                tbcn-id         -- if comsig-or-ext-addr-present FALSE
                    TranslatedBearerConnectionID,
                id-and-type -- if comsig-or-ext-addr-present TRUE
                    SEQUENCE {
                        comsig-or-ext-addr
```

```
                            BOOLEAN,
                        address-structure
                            CHOICE {
                                ext-addr-type-and-address
                                -- if comsig-or-ext-addr FALSE
                                    SEQUENCE {
                                        ext-addr-type
                                            ExtAddrType,
                                        extended-address
                                            BCnID
                                    },
                                com-sig-type-and-address
                                -- if comsig-or-ext-addr TRUE
                                    SEQUENCE {
                                        com-sig-type
                                            FromMobileComSigType,
                                        comsig-addr
                                            ComSigAddress
                                    }
                            }
                    }
                },
        timing
            -- included if bearer-number-present TRUE
            SEQUENCE {
                timing-required
                    BOOLEAN,
                fwd-bearer
                    FwdBearer,
                slot-num
                    SlotNumber
            } OPTIONAL,
        length
            INTEGER (0..255) OPTIONAL
    }
```

Figures 5.3 to 5.14 illustrate the various combinations possible for the RetBCtPDUHeader format.

(MSB) 8	7	6	5	Bit No 4	3	2	(LSB) 1	
X	0	0	0					Octet 1
tbcn-id								Octet 2

Figure 5.3: RetBCtPDUHeader (Connection specific, no length field)

(MSB) 8	7	6	5	Bit No 4	3	2	(LSB) 1	
X	1	0	0					Octet 1
tbcn-id								Octet 2
length								Octet 3

Figure 5.4: RetBCtPDUHeader (Connection specific, with length field)

8 (MSB)	7	6	5	4	3	2	1 (LSB)	
X	0	0	1					Octet 1
tbcn-id								Octet 2
t	fwd-bearer			slot-num				Octet 3

Figure 5.5: RetBCtPDUHeader (Connection specific, no length field, timing octet present)

8 (MSB)	7	6	5	4	3	2	1 (LSB)	
X	1	0	1					Octet 1
tbcn-id								Octet 2
t	fwd-bearer			slot-num				Octet 3
length								Octet 4

Figure 5.6: RetBCtPDUHeader (Connection specific, with length field, timing octet present)

8 (MSB)	7	6	5	4	3	2	1 (LSB)	
X	0	1	0	0	ext-addr-type=7			Octet 1
bcn-id								Octet 2
								Octet 3
								Octet 4

Figure 5.7: RetBCtPDUHeader (BCnID, no length field)

8 (MSB)	7	6	5	4	3	2	1 (LSB)	
X	1	1	0	0	ext-addr-type=7			Octet 1
bcn-id								Octet 2
								Octet 3
								Octet 4
length								Octet 5

Figure 5.8: RetBCtPDUHeader (BCnID, with length field)

(MSB) Bit No (LSB)

8	7	6	5	4	3	2	1	
X	0	1	1	0	ext-addr-type=7			Octet 1
bcn-id								Octet 2
								Octet 3
								Octet 4
t	fwd-bearer			slot-num				Octet 5

Figure 5.9: RetBCtPDUHeader (BCnID, no length field)

(MSB) Bit No (LSB)

8	7	6	5	4	3	2	1	
X	1	1	1	0	ext-addr-type=7			Octet 1
bcn-id								Octet 2
								Octet 3
								Octet 4
t	fwd-bearer			slot-num				Octet 5
length								Octet 6

Figure 5.10: RetBCtPDUHeader (BCnID, with length field)

(MSB) Bit No (LSB)

8	7	6	5	4	3	2	1	
X	0	1	0	1	com-sig-type			Octet 1
comsig-addr								Octet 2
								Octet 3
								Octet m

Figure 5.11: RetBCtPDUHeader (Comsig-addr, no length field)

(MSB) Bit No (LSB)

8	7	6	5	4	3	2	1	
X	1	1	0	1	com-sig-type			Octet 1
comsig-addr								Octet 2
								Octet 3
length								Octet m

Figure 5.12: RetBCtPDUHeader (Comsig-addr, with length field)

(MSB) 8	7	6	5	4	3	2	(LSB) 1	
X	0	1	1	1	com-sig-type			Octet 1
comsig-addr								Octet 2
								Octet 3
t	fwd-bearer			slot-num				Octet m

Figure 5.13: RetBCtPDUHeader (Comsig-addr, no length field, timing octet present)

(MSB) 8	7	6	5	4	3	2	(LSB) 1	
X	1	1	1	1	com-sig-type			Octet 1
comsig-addr								Octet 2
								Octet 3
t	fwd-bearer			slot-num				
length								Octet m

Figure 5.14: RetBCtPDUHeader (Comsig-addr, with length field, timing octet present)

5.1.3.2 RetBCtPDU Header Structure Type 2

The Bearer Control PDU Header Type 2 shall be used for the first BCtPDUs sent in a Return Block structure (see ETSI TS 102 744-2-1 [8]) using Return Bearer Types R80T0.5Q or R80T1Q. This Bearer Control PDU Header has the following structure:

```
LdrBCtPDUHeader ::=
    SEQUENCE {
        bct-sdu-follows
            BOOLEAN,
        length-present
            BOOLEAN,
        comsig-or-ext-addr-present
            BOOLEAN,
        continuation-burst
            BOOLEAN,                -- con in figures
        header-structure
            CHOICE {
                backoff-and-bcnid
                -- if comsig-or-ext-addr-present FALSE
                    SEQUENCE {
                        backoff
                            Backoff,
                        ext-address
                            BCnID
                    },
                com-or-ext-addr-type
                -- if comsig-or-ext-addr-present TRUE
                    SEQUENCE {
                        comsig-or-ext-addr
                            BOOLEAN,
                        address-structure
                            CHOICE {
```

```
                                ext-addr-type-and-address
                                -- if comsig-or-ext-addr FALSE
                                    SEQUENCE {
                                        ext-addr-type
                                            ExtAddrType,
                                        fwd-bearer-info
                                            FwdBCtInfo,
                                        ext-addr
                                            BCnID
                                    },
                                com-sig-type-and-address
                                -- if comsig-or-ext-addr TRUE
                                    SEQUENCE {
                                        com-sig-type
                                            FromMobileComSigType,
                                        fwd-bearer-info
                                            FwdBCtInfo,
                                        comsig-addr
                                            ComSigAddress
                                    }
                            }
                    }
            },
    length
        INTEGER (0..255) OPTIONAL
}
```

where FwdBCtInfo is defined as follows:

```
FwdBCtInfo ::=
    SEQUENCE {
        rnc-id
            INTEGER (0..255),
        timing-required
            BOOLEAN,
        f-bearer
            FwdBearer,
        bct-id
            BCtID,
        spot-beam-id
            SpotBeamID
    }
```

The information relating to the forward bearer, including RNC-Id, FbearerNo, BCtId and Spot-beam-ID is transmitted after the mobile terminal has retuned to a new forward bearer and before it has received communications from the RNC on the new forward physical bearer. These information elements may also be included if timing is required by the mobile terminal. The mobile terminal cannot describe the backoff when operating in this mode, so it shall use the Initial Reference Level for use with this spot beam type (unless overridden by the RNC using a broadcast or UE-specific signalling mechanism).

Figures 5.15 to 5.20 illustrate the combinations possible for the RetBCtPDUHeader format.

(MSB)			Bit No			(LSB)		
8	7	6	5	4	3	2	1	
sdu	0	0	con	backoff				Octet 1
bcn-id								Octet 2
								Octet 3
								Octet 4

Figure 5.15: LdrBCtPDUHeader (BCnID, no length field)

(MSB)			Bit No				(LSB)	
8	7	6	5	4	3	2	1	
sdu	1	0	con	backoff				Octet 1
bcn-id								Octet 2
								Octet 3
								Octet 4
Length								Octet 5

Figure 5.16: LdrBCtPDUHeader (BCnID, with length field)

(MSB)			Bit No				(LSB)	
8	7	6	5	4	3	2	1	
sdu	0	1	con	0	ext-addr-type=7			Octet 1
rnc-id								Octet 2
t	f-bearer			bct-id				Octet 3
spot-beam-id								Octet 4
bcn-id								Octet 5
								Octet 6
								Octet 7

Figure 5.17: LdrBCtPDUHeader (Ext-addr, no length field)

(MSB)			Bit No				(LSB)	
8	7	6	5	4	3	2	1	
sdu	1	1	con	0	ext-addr-type=7			Octet 1
rnc-id								Octet 2
t	f-bearer			bct-id				Octet 3
spot-beam-id								Octet 4
bcn-id								Octet 5
								Octet 6
								Octet 7
length								Octet 8

Figure 5.18: LdrBCtPDUHeader (Ext-addr, length field)

(MSB)			Bit No				(LSB)	
8	7	6	5	4	3	2	1	
sdu	0	1	con	1	com-sig-type			Octet 1
rnc-id								Octet 2
t	f-bearer			bct-id				Octet 3
spot-beam-id								Octet 4
comsig-addr								Octet 5
								Octet 6
⋮								
								Octet m

Figure 5.19: LdrBCtPDUHeader (Comsig-addr, no length field)

(MSB)			Bit No				(LSB)	
8	7	6	5	4	3	2	1	
sdu	1	1	con	1	com-sig-type			Octet 1
rnc-id								Octet 2
t	f-bearer			bct-id				Octet 3
spot-beam-id								Octet 4
comsig-addr								Octet 5
								Octet 6
⋮								
length								Octet m

Figure 5.20: LdrBCtPDUHeader (Comsig-addr, with length field)

5.1.4 Forward Bearer Control PDU Header Structure (FwdBCtPDUHeader)

In the forward direction, the Bearer Control PDU Header has the following structure:

```
FwdBCtPDUHeader ::=
    SEQUENCE {
        bct-sdu-follows
            BOOLEAN,
        length-present
            BOOLEAN,
        bct-pdu-addr-type
            BCtPDUAddrType,
        header-structure
            CHOICE {
                tbcn-id     -- if comsig-or-ext-addr-present FALSE
                    TranslatedBearerConnectionID,
                id-and-type
                    SEQUENCE {
                        comsig-or-ext-address
                            BOOLEAN,
                        address-structure
                            CHOICE {
                                ext-addr-type-and-address
                                -- if comsig-or-ext-addr FALSE
                                    SEQUENCE {
                                        ext-addr-type
```

```
                                        ExtAddrType,
                                    extended-address
                                        BCnID
                                },
                            com-sig-type-and-address
                            -- if comsig-or-ext-addr TRUE
                                SEQUENCE {
                                    com-sig-type
                                        ToMobileComSigType,
                                    address
                                        CHOICE {
                                            broadcast
                                                NULL,
                                            comsig-addr
                                                ComSigAddress
                                        }
                                }
                        }
                }
        },
    length
        INTEGER (0..255) OPTIONAL
}
```

The boolean variable comsig-or-ext-address is set to TRUE if common signalling is used in the PDU, or set to FALSE if extended addressing is used. For extended addressing, the format of extended-address is specified by the value in the field ext-addr-type as defined in clause 5.1.5.12.4. Figures 5.21 to 5.28 illustrate the various combinations possible for the BCtPDU Header format.

8 (MSB)	7	6	5	4	3	2	1 (LSB)	
X	0	0	0	1	com-sig-type			Octet 1

Figure 5.21: FwdBCtPDUHeader (Broadcast, no length field)

8 (MSB)	7	6	5	4	3	2	1 (LSB)	
X	1	0	0	1	com-sig-type			Octet 1
length								Octet 2

Figure 5.22: FwdBCtPDUHeader (Broadcast, with length field)

8 (MSB)	7	6	5	4	3	2	1 (LSB)	
X	0	0	1					Octet 1
tbcn-id								Octet 2

Figure 5.23: FwdBCtPDUHeader (Connection specific, no length field)

8 (MSB)	7	6	5	4	3	2	1 (LSB)	
X	1	0	1					Octet 1
tbcn-id								Octet 2
length								Octet 3

Figure 5.24: FwdBCtPDUHeader (Connection specific, with length field)

(MSB)			Bit No				(LSB)	
8	7	6	5	4	3	2	1	
X	0	1	0	0	ext-addr-type=7			Octet 1
bcn-id								Octet 2
								Octet 3
								Octet 4

Figure 5.25: FwdBCtPDUHeader (ext-address-type BCnID, no length field)

(MSB)			Bit No				(LSB)	
8	7	6	5	4	3	2	1	
X	1	1	0	0	ext-addr-type=7			Octet 1
bcn-id								Octet 2
								Octet 3
								Octet 4
Length								Octet 5

Figure 5.26: FwdBCtPDUHeader (ext-address-type BCnID, with length field)

(MSB)			Bit No				(LSB)	
8	7	6	5	4	3	2	1	
X	0	1	0	1	com-sig-type			Octet 1
comsig-addr								Octet 2
								Octet 3
								Octet m

Figure 5.27: FwdBCtPDUHeader (Comsig-addr, no length field)

(MSB)			Bit No				(LSB)	
8	7	6	5	4	3	2	1	
X	1	1	0	1	com-sig-type			Octet 1
comsig-addr								Octet 2
								Octet 3
length								Octet m

Figure 5.28: FwdBCtPDUHeader (Comsig-addr, with length field)

5.1.5 Header Parameters

5.1.5.1 Bct-sdu-follows

The BOOLEAN variable *bct-sdu-follows* indicates, when set to value '1', that a BCtSDU follows the header; else a BCnPDU follows the header.

5.1.5.2 Length-present

The BOOLEAN variable *length-present* indicates the presence of a length field within the header when set to the value '1'.

5.1.5.3 Length

The *length* field, if present, is an eight-bit value which specifies the content length of the PDU from the end of the header to the start of the CRC field. If the *length* field is absent, the PDU either fills the remaining space in a physical layer block or slot (up to a maximum content length of 256 octets), or is a PDU with the maximum content length of 256 octets.

5.1.5.4 BCtPDUAddrType

The variable *bct-pdu-addr-type* is a two-bit field which is used in the to-UE direction only, and determines the addressing mechanism used for the BCtPDU. The variable is encoded as follows and as shown in Table 5.1:

```
BCtPDUAddrType ::=
    INTEGER {
        broadcast (0),
        tbcn-id (1),
        comsig-or-ext-addr (2)
        -- reserved (3)
        } (0..3)
```

Table 5.1: BCtPDUAddrType

Type	Address Mechanism
0	broadcast (NULL address)
1	tbcn-id (specific connection association)
2	comsig-addr (common signalling connection) or extended-address (specific connection association)
3	not used (reserved)

FEC Blocks (see ETSI TS 102 744-2-1 [8]) shall be filled by the RNC in the following order:

1) Broadcast BCtPDUs (bct-pdu-addr-type = 0);

2) BCtPDUs carrying an ALComPDU (bct-pdu-addr-type = 2 and comsig-or-ext-addr = TRUE);

3) BCtPDUs addressed by tBCnId (bct-pdu-addr-type = 1);

4) BCtPDUs addressed by BCnId (bct-pdu-addr-type = 2 and comsig-or-ext-addr = FALSE).

5.1.5.5 Comsig-or-ext-addr-present

The BOOLEAN variable *comsig-or-ext-addr-present* determines the choice of the addressing mechanism of the BCtPDU. If this variable is set to '1', then the sequence *comsig-type-and-address or ext-addr-type-and-address* is present in the header, else *tbcn-id* is present.

5.1.5.6 Bearer-number-present

The BOOLEAN variable *bearer-number-present* is used in the return (from-UE) direction only and specifies whether the sequence *timing* (containing the information fields *timing-required*, *fwd-bearer* and s*lot-num*) is present. If this flag is set to '1', then this sequence is present. The information is used by the RNC to determine when the UE is uncertain about timing and which Forward Bearer the UE is tuned to.

5.1.5.7 Timing-required (T)

The BOOLEAN variable *timing-required* indicates whether the UE requires timing correction information. This flag is set to '1' whenever the UE transmits an initial or un-timed random access burst in a contention slot.

5.1.5.8 FwdBearer

The variable *fwd-bearer* field is a three-bit field which contains the *f-bearer* value transmitted by the RNC in the bulletin board (see clause 5.4.4) identifying the forward bearer to which the UE is currently tuned.

```
FwdBearer ::=
    INTEGER (0..7)
```

5.1.5.9 SlotNumber

The variable *slot-num* field is a four-bit field which indicates in which slot position within the return frame the UE considers that it is currently transmitting. Return slot timing is synchronized, but offset from, forward frames. *Slot-num* indicates the slot position within the return frame, with the return frame having a defined offset in time against the forward frame (as defined in ETSI TS 102 744-2-1 [8]).

```
SlotNumber ::=
    INTEGER (0..15)
```

The slot numbering period is 80 ms (i.e. equal to a return frame) and each slot has a duration of 5 ms, hence there are 16 slots numbered from 0 through to 15. A *slot-num* of zero indicates the first 5 ms period that corresponds with the start of the return frame and a *slot-num* of 15 indicates the last 5 ms slot in the (retimed) 80 ms frame.

Slot-num values are statically determined from the position of slots within a return frame and not related to the number of slots described in the Return Schedule slot plans. This information is used by the RNC to determine the timing correction information to provide to the UE.

For return bearer types R80T25X and R80T5X, *slot-num* is irrelevant and shall always be set to zero if *slot-num* is included in the BCtPDU header. For these bearer types the following paragraphs in the subsection are not applicable.

Figure 5.29 illustrates the slot numbering covering 80 ms (for return bearers other than R20T0.5Q, R80T0.5Q and R80T1Q) which would apply for the case where two slot plans (see clause 5.4.5.7) support the transmission of three consecutive 20 ms bursts, followed by 4 consecutive 5 ms bursts in the same frame period.

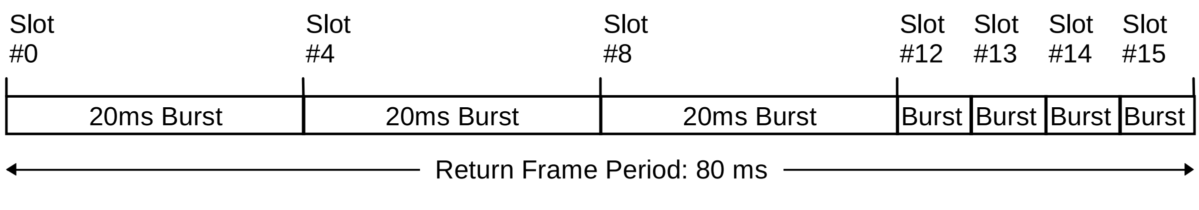


Figure 5.29: Example for two Slot Plans over 80 ms Return Frame Period showing Slot Numbering

Figure 5.30 illustrates the slot numbering for the bearer type R20T0.5Q. In this example, two slot plans define 320 ms while each slot has a duration of 20 ms, i.e. the time axis is stretched by a factor of four. In this case, the two slot plans cover four return frames of 80 ms duration each. However, the slot numbering remains the same as for all other bearer types and the start of Slot 0 shall coincide with the start of a return frame. This implies that for bearer type R20T0.5Q only slot numbers 0, 4, 8 and 12 are valid.

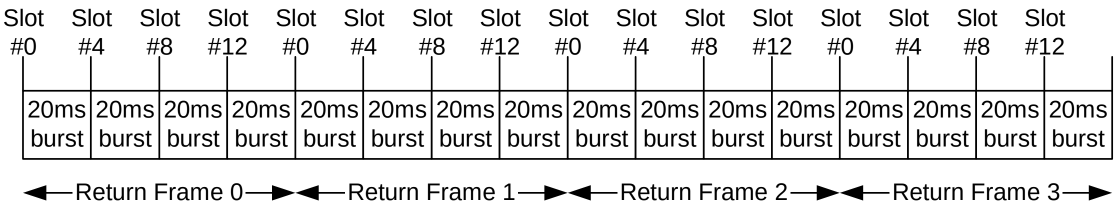


Figure 5.30: Example for two Slot Plans defining a 320 ms Period showing Return Frame and Slot Numbering

5.1.5.10 TranslatedBearerConnectionID

The variable *tbcn-id* is a 12-bit address field used to identify a specific connection within the scope of the Bearer Control object managing this bearer. It is mapped to a Bearer Connection ID during the Establishment, Modify or Handover processes.

```
TranslatedBearerConnectionID ::=
    INTEGER (0..4095)
```

5.1.5.11 ComSigAddress

5.1.5.11.0 General

The parameter *comsig-addr* carries the Non Access Stratum (NAS) UE Identity which is used to address common signalling messages, and the data type ComSigAddress is defined as follows:

```
ComSigAddress ::=
    SEQUENCE {
        ue-id-type
            UEIdType,
        intial-ue-identity
            CHOICE {
                p-tmsi-seq
                    SEQUENCE {
                        reserved1
                            BIT STRING (SIZE (4)),
                        plmn-id
                            PLMN-Identity,
                        lac
                            BIT STRING (SIZE (16)),
                        p-tmsi
                            P-TMSI-GSM-MAP
                    },
                tmsi-seq
                    SEQUENCE {
                        reserved2
                            BIT STRING (SIZE (4)),
                        plmn-id
                            PLMN-Identity,
                        lac
                            BIT STRING (SIZE (16)),
                        tmsi
                            TMSI-GSM-MAP
                    },
                imsi
                    IMSI-GSM-MAP,
                imei
                    IMEI,
                imsi-DS-41-seq
                    SEQUENCE {
                        length
                            INTEGER (0..15),
                        imsi-DS-41
                            IMSI-DS-41  -- 5 to 7 octets
                    }
            }
    }
```

5.1.5.11.1 Ue-id-type

The data type UEIdType is defined as follows:

```
UEIdType ::=
    INTEGER {
        p-tmsi (0),
        tmsi (1),
        imsi (2),
        imei (3),
        imsi-DS-41 (4)
    } (0..15)
```

The data types used in the *initial-ue-identity* field are as specified in ETSI TS 125 331 [1], clause 10.3.3.15.

5.1.5.11.2 P-tmsi-seq

The *p-tmsi-seq* is used if the registration signalling uses a P-TMSI in the ComSigAddress. Since P-TMSI is only unique within one routing area, the *plmn-id* and *lac* (location area code) to which the P-TMSI relates are included.

NOTE: Normally the Routing Area Identification (RAI) would also need to be included to uniquely identify the Routing Area. However, due to the limited number of octets which may be available to carry the entire registration message, the RAI has not been included.

The parameter *plmn-id* is of type PLMN-ID (Public Land Mobile Network Identity) which is defined in ETSI TS 125 331 [1], clauses 10.3.1.11 and 11.3 as follows:

```
PLMN-Identity ::=
    SEQUENCE {
        mcc
            MCC,
        mnc
            MNC
    }
```

with

```
MCC ::= SEQUENCE SIZE (3) OF Digit
```

and

```
MNC ::= SEQUENCE SIZE (3) OF Digit
    -- encode third digit as 0x0F if size of mnc = 2
```

In this case *ue-id-type* and *initial-ue-identity* are encoded in a total of ten octets as shown in Figure 5.31:

(MSB) 8	7	6	5	Bit No 4	3	2	(LSB) 1	
0	0	0	0	reserved				Octet 1
mcc (1st digit)				mcc (2nd digit)				Octet 2
mcc (3rd digit)				mnc (1st digit)				Octet 3
mnc (2nd digit)				mnc (3rd digit)				Octet 4
lac								Octet 5
								Octet 6
p-tmsi								Octet 7
								Octet 8
								Octet 9
								Octet 10

Figure 5.31: Ue-id-type and Initial-ue-identity: P-TMSI

5.1.5.11.3 Tmsi-seq

The *tmsi-seq* is used if the registration signalling uses a Temporary Mobile Subscriber Identity (TMSI) in the ComSigAddress. Since TMSI is only unique within one location area, the *plmn-id* and *lac* (location area code) to which the TMSI relates are included.

In this case *ue-id-type* and *initial-ue-identity* are encoded in a total of ten octets as shown in Figure 5.32:

8 (MSB)	7	6	5	4	3	2	1 (LSB)	
0	0	0	1	reserved				Octet 1
mcc (1st digit)				mcc (2nd digit)				Octet 2
mcc (3rd digit)				mnc (1st digit)				Octet 3
mnc (2nd digit)				mnc (3rd digit)				Octet 4
lac								Octet 5
								Octet 6
tmsi								Octet 7
								Octet 8
								Octet 9
								Octet 10

Figure 5.32: Ue-id-type and Initial-ue-identity: TMSI

5.1.5.11.4 Imsi

In the case that neither TMSI nor P-TMSI are available then the International Mobile Subscriber Identity (IMSI) shall be used as the *initial-ue-identity* (*ue-id-type* = 2). The IMSI is defined in ETSI TS 125 331 [1], clauses 10.3.1.5 and 11.3 as follows:

```
IMSI-GSM-MAP ::=
    SEQUENCE (SIZE (6..15)) OF Digit
```

with

```
Digit ::=
    INTEGER (0..9)
```

In this case *ue-id-type* and *initial-ue-identity* are encoded in a total of eight octets as shown in Figure 5.33, with 7,5 octets providing for 15 digits to be encoded. In the event that the IMSI occupies less than 15 digits, then any remaining digits at the end of the sequence shall be filled with 0xF.

8 (MSB)	7	6	5	4	3	2	1 (LSB)	
0	0	1	0	imsi				Octet 1
imsi								Octet 2
								Octet 7
								Octet 8

Figure 5.33: Ue-id-type and Initial-ue-identity: IMSI

5.1.5.11.5 Imei

In the case that no SIM card is inserted in the UE and that registration without SIM card is permitted (see clause 5.7.19) then the International Mobile Equipment Identity (IMEI) shall be used as the *initial-ue-identity* (*ue-id-type* = 3). The IMEI is defined in ETSI TS 125 331 [1], clauses 10.3.1.4 and 11.3 as follows:

```
IMEI ::=
    SEQUENCE (SIZE (15)) OF IMEI-Digit
```

with

```
IMEI-Digit ::=
    INTEGER (0..15)
```

In this case, *ue-id-type* and *initial-ue-identity* are encoded as shown in Figure 5.34.

(MSB)			Bit No				(LSB)	
8	7	6	5	4	3	2	1	
0	0	1	1					Octet 1
imei								Octet 2
								Octet 7
								Octet 8

Figure 5.34: Ue-id-type and Initial-ue-identity: IMEI

5.1.5.11.6 Imsi-DS-41

The *imsi-DS-41* parameter is used if a SIM card issued by a network to ANSI-41 standard is inserted into the UE. In this case, *ue-id-type* = 4 and the data type IMSI-DS-41 is defined in ETSI TS 125 331 [1], clause 11.3 as follows:

```
IMSI-DS-41 ::=
    OCTET STRING (SIZE (5..7))
```

The variable length of the *imsi-DS-41* parameter is signalled in the *length* field. The *ue-id-type* and *initial-ue-identity* are encoded as shown in Figure 5.35.

(MSB)			Bit No				(LSB)	
8	7	6	5	4	3	2	1	
0	1	0	0	length				Octet 1
imsi-DS-41								Octet 2
								Octet 7
								Octet 8

Figure 5.35: Ue-id-type and Initial-ue-identity: IMSI-DS-41

5.1.5.12 Common Signalling (ComSig) Types

5.1.5.12.1 ToMobileComSigType

The variable *com-sig-type* is defined in the to-UE direction as follows:

```
ToMobileComSigType ::=
    INTEGER {
        information (1),
        paging-type-1 (2),
        register-ack (3),
        register-rej (4),
        deregister-common (5)
    } (0..7)
```

5.1.5.12.2 FromMobileComSigType

The variable *com-sig-type* is defined in the from-UE direction as follows:

```
FromMobileComSigType ::=
    INTEGER {
        register (2)
    } (0..7)
```

5.1.5.12.3 Comsig-or-ext-addr

The boolean variable *comsig-or-ext-addr* is used to specify whether Common Signalling or extended addressing is used in the BCtPDU Header. If *comsig-or-ext-addr* is set to '1' then Common Signalling is used in the BCtPDU Header (see clause 5.1.5.11). If *comsig-or-ext-addr* is set to '0', then extended addressing is used in the BCtPDU Header (see clause 5.1.5.12.4).

5.1.5.12.4 Ext-addr-type-and-address

The sequence *ext-addr-type-and-address* consists of the variable *ext-addr-type* and the variable *extended-address*. At present, *extended-address* is always of type BCnID that allows a connection to be addressed by its Bearer Connection ID (BCnID).

The variable *ext-addr-type* allows adding further addressing mechanisms in both the forward and return directions. At present only one value is defined that allows a connection to be addressed by its Bearer Connection ID (BCnID):

```
ExtAddrType ::=
    INTEGER {
        reserved-addr-0(0),
        reserved-addr-1(1),
        reserved-addr-2(2),
        reserved-addr-3(3),
        reserved-addr-4(4),
        reserved-addr-5(5),
        reserved-addr-6(6),
        bcnid (7)
    } (0..7)
```

5.1.5.13 Continuation-burst

The Boolean variable *continuation-burst (con)* is set to TRUE when a burst will be transmitted in the next slot only if the burst or continuous sequence of bursts is being preceded with a Preamble Acquisition sequence. This information is used to signal to the RAN demodulator and decoder that a subsequent transmission will occur immediately.

5.1.5.14 Backoff-and-bcnid

The sequence *backoff-and-bcnid* is used in the LdrBCtPDUHeader for two purposes:

1) to provide for the UE to use addressing on the basis of Bearer Connection IDs (see clause 5.1.3.2); and

2) to allow the UE to inform the RNC of the maximum level of backoff (in the variable *backoff* in units of one dB) from the maximum power at which the mobile terminal is capable of operating (see clause 5.7.21).

5.1.6 BCtPayload

5.1.6.0 General

The *bct-payload* field contains either a Bearer Connection PDU (BCnPDU) or an Adaptation Layer Common Signalling PDU (ALComPDU) as determined by the header parameters.

```
BCtPayload ::=
    CHOICE {
        common-pdu
            ALComPDU,
        bcn-pdu
            BCnPDU
    }
```

5.1.6.1 BCnPDU

The *bcn-pdu* contains a Bearer Connection Layer Protocol Data Unit, the contents of which are transparent to the Bearer Control Layer. For a definition of Bearer Connection Protocol Data Unit, refer to ETSI TS 102 744-3-3 [11].

```
BCnPDU ::=
    OCTET STRING (SIZE (0..255))
```

5.1.6.2 ALComPDU

The *common-pdu* contains an Adaptation Layer Common Signalling Protocol Data Unit (ALComPDU), the contents of which are transparent to the Bearer Control Layer. For a definition of Adaptation Layer Common Signalling Protocol Data Unit, refer to ETSI TS 102 744-3-5 [13].

```
ALComPDU ::=
    OCTET STRING (SIZE (0..255))
```

5.1.7 CRC

The *crc* field is used to identify erroneous Bearer Control PDUs. Bearer Control PDUs which do not have a valid CRC shall be discarded by the receiving Bearer Control process unless the Bearer Connection supports the delivery of erroneous PDUs (see ETSI TS 102 744-3-2 [10]).

The Generator Polynomial $x^{16} + x^{12} + x^5 + 1$ as specified in [3], clause 2.2.7, shall be used to calculate the CRC.

The CRC calculation is illustrated in Figure 5.36.

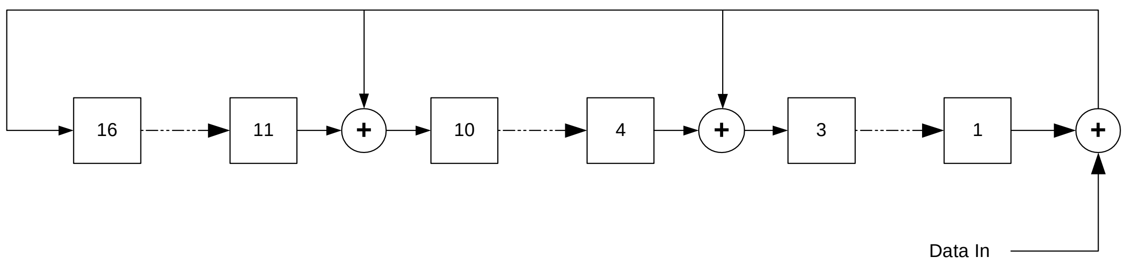


Figure 5.36: CRC Calculation

On initialization, all delay elements are initialized to "1". The BCtPDU data (excluding the two CRC octets) is then clocked into the shift register, starting with Bit 1 of the first octet. After the last bit (Bit 8 of the last BCtPDU octet) has been clocked in, the ones complement of the shift register contents forms the CRC, which shall be appended to the BCtPDU as shown in Figure 5.37.

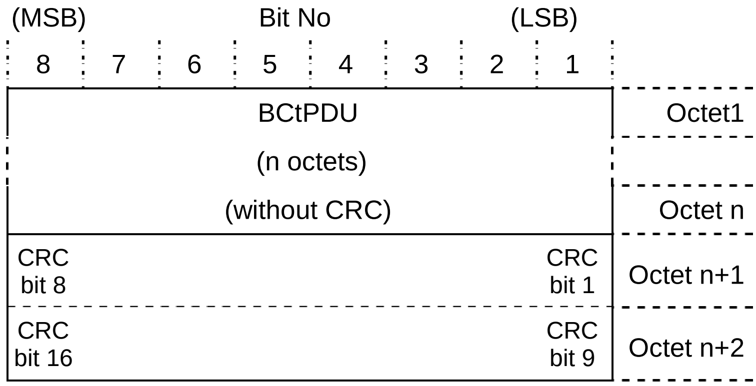


Figure 5.37: Mapping of CRC Bits

On receiving a BCtPDU, the entire BCtPDU (including the CRC octets) shall be clocked into the shift register, which has been initialized with all ones. Providing the BCtPDU has not been corrupted, the shift register will contain 0xF0B8 after the last bit of the BCtPDU (i.e. bit 8 of the second CRC octet) has been shifted in.

5.2 Bearer Control Embedded Protocol Data Unit (BCtEPDU)

5.2.1 Bearer Control Embedded PDU Structure

The Bearer Control Embedded PDU (BCtEPDU) is contained within an Establish, Modify or Handover Adaptation Layer Signalling PDU (ALSigPDU) as well as the RegisterAck Common Signalling PDU (ALComPDU) and is used to carry Bearer Control specific information associated with the connection synchronously with the AL-SigPDU. The Embedded PDU contains a sequence of Embedded Bearer Control SDUs (BCtESDUs) as follows:

```
BCtEPDU::=
    SEQUENCE {
        bct-esdu-list
            CHOICE {
                attach-bctesdu-list
                    AttachBCtESDUList,
                reattach-bctesdu-list
                    ReattachBCtESDUList,
                system-information-bctesdu-list
                    SystemInformationBCtESDUList,
                modify-bctesdu-list
                    ModifiedConnInfo
            }
    }
```

5.2.2 BCtESDU Lists

5.2.2.1 AttachBCtESDUList

The AttachBCtESDUList is used to carry Bearer Control parameters required to attach a connection to the Bearer Control (as occurs during the Establishment of new connections, including the RegisterAck process) and is structured as follows:

```
AttachBCtESDUList ::=
    SEQUENCE {
        attached-conn-info
            AttachedConnInfo,
        hardware-info-list
            SEQUENCE OF BCtESDU OPTIONAL
                -- of type HardwareAVPList
    }
```

The *hardware-info-list* is only normally present if the UE receiver is being retuned or if sleep mode parameters are being modified. For a definition of HardwareAVPList see clause 5.6.5.

5.2.2.2 ReattachBCtESDUList

The ReattachBCtESDUList is used to carry Bearer Control parameters required to reattach a connection to the Bearer Control (as occurs during the Handover of connections) and is structured as follows:

```
ReattachBCtESDUlist ::=
    SEQUENCE {
        reattached-conn-info-list
            SEQUENCE OF ReattachedConnInfo,
        hardware-info-list
            SEQUENCE OF BCtESDU
                -- of type HardwareAVPList
    }
```

A ReattachedConnInfo is present when a set of connections is being handed over between Bearer Control processes during a Handover operation. The *hardware-info-list* contains channel information if the UE receiver is being retuned and may also contain sleep mode parameters if these are being modified. For a definition of HardwareAVPList see clause 5.6.5.

5.2.2.3 SystemInformationBCtESDUList

The ESDU-List is sent in the SystemInformation AL-Sig-PDU to transfer system information to a UE in connected mode. The ESDU-List has the following structure:

```
SystemInformationBCtESDUList ::=
    SEQUENCE {
        spot-beam-maps
            SEQUENCE OF BCtESDU OPTIONAL,   -- of type SpotBeamMap
        bearer-table-update
            SEQUENCE OF BCtESDU OPTIONAL,   -- of type BearerTableUpdate
        sys-info-avp-list
            BCtESDU OPTIONAL,               -- of type AVPList
        hardware-info-list
            BCtESDU OPTIONAL            -- of type HardwareAVPList
    }
```

The sys-info-avp-list may contain any of the following AVP types: NASSystemInfo, PrimaryBearer, PLMNInfo, AccessControl, ReturnLinkReferenceLevel, ReturnLinkReferenceLevelSet, InitialReferenceLevelSet, MaxDelayAndDelayRange and CommonSigRetry while the hardware-info-list may contain a SleepMode AVP. This mechanism is typically used if the System Information needs to be updated rapidly for a particular UE (e.g. during a Handover). It is not a replacement for broadcast system information.

5.2.3 BCtESDU Sequences

5.2.3.1 AttachedConnInfo

This sequence of SDUs is used to provide the parameters associated with the connection which is being established at the bearer connection level.

```
AttachedConnInfo ::=
    SEQUENCE {
        conn-association
            BCtESDU OPTIONAL    -- of type ConnectionAssociation,
        conn-avp-list
            BCtESDU OPTIONAL    -- of type AVPList
    }
```

Conn-association (see clause 5.6.1) defines the association between the bearer connection and the translated bearer connection IDs which map to each hardware element within the UE. For a definition of AVPList see clause 5.4.10.

5.2.3.2 ReattachedConnInfo

This sequence of SDUs is used to provide the parameters associated with the bearer connections which are being handed over to this bearer control during a Handover operation.

```
ReattachedConnInfo ::=
    SEQUENCE {
        conn-reassociation
            BCtESDU             -- of type ConnectionReassociation,
        conn-avp-list
```

```
        BCtESDU OPTIONAL    -- of type AVPList
    }
```

Conn-reassociation (see clause 5.6.2) defines the association between the bearer connection and the translated bearer connection IDs which map to each hardware element within the UE. For a definition of AVPList see clause 5.4.10.

5.2.3.3 ModifiedConnInfo

This sequence of SDUs is used to provide bearer control layer parameters associated with the bearer connection which is being modified.

```
ModifiedConnInfo ::=
    SEQUENCE {
        conn-avp-list
            BCtESDU     -- of type AVPList
    }
```

NOTE: Although the data type contains one element only, it has been defined as SEQUENCE in order to comply with ASN.1 rules.

For a definition of AVPList see clause 5.4.10.

5.3 Bearer Control SDUs (BCtSDU)

5.3.0 General

Bearer Control SDUs (BCtSDUs) are used for the peer-to-peer communication between Bearer Control Processes at the UE and the RNC. Tables 5.2 and 5.3 define the different BCtSDU Types which can be sent either by the UE or the RNC.

Table 5.2: From-UE Bearer Control SDU Types

SDUType	Interpretation	Contained In
0x00	ConnPDU	Connection Specific BCtPDU
0x01	Status	Connection Specific BCtPDU
0x02..0x03	Reserved	-
0x04	QRate	Connection Specific BCtPDU
0x05	QLen	Connection Specific BCtPDU
0x06	Reserved	-
0x07	AVPList	Connection Specific BCtPDU
0x08..0x3F	Reserved	-

In the from-UE direction, the only AVPs supported in an AVPList SDU are the ReferenceLevelAcknowledge and ReceivedSignalQuality AVPs.

Table 5.3: To-UE Bearer Control SDU Types

SDUType	Interpretation	Contained In
0x00	ConnPDU	Connection Specific BCtPDU
0x01	BulletinBoard	Broadcast BCtPDU
0x02	ReturnSchedule	Broadcast BCtPDU Connection Specific BCtPDU
0x03	StatusAckList	Broadcast BCtPDU
0x04	Reserved	
0x05	Reserved	
0x06	SpecificAVPList	Broadcast BCtPDU
0x07	AVPList	Any BCtPDU
0x08	SpotBeamMap	Any BCtPDU
0x09	BearerTables	Broadcast BCtPDU
0x0A	GPSEphemeris	Broadcast BCtPDU
0x0B	Reserved	
0x0C	SystemInfoIndex	Broadcast BCtPDU
0x0D	BearerTableUpdate	Connection Specific BCtPDU
0x0E..0x3F	Reserved	

The ReturnSchedule may only be included in a Connection Specific BCtPDU when used for dedicated return channel operation with R80T2.5X and R80T5X bearers. In this case, the Return Schedule shall be the only BCtSDU within this BCtPDU.

The column 'Contained In' refers to the type of BCtPDU in which the particular SDU Type may be carried:

'Broadcast BCtPDU' a BCtPDU using the Broadcast addressing mechanism (BCtPDUAddrType = 0);

'Connection Specific BCtPDU' a BCtPDU using a translated Bearer Connection ID as the addressing mechanism (BCtPDUAddrType = 1) or using a Bearer Connection ID as the extended addressing mechanism (BCtPDUAddrType = 2 and *comsig-or-ext-address* = FALSE);

'Any BCtPDU' a BCtPDU using any valid addressing mechanism (BCtPDUAddrType = 0, 1 or 2).

5.3.1 Bearer Control SDU Structure (BCtSDU)

The Bearer Control SDU (BCtSDU) Structure is as follows and the format is shown in Figures 5.38 and 5.39:

```
BCtSDU ::=
    SEQUENCE {
        bct-sdu-follows
            BOOLEAN,
        type-and-length
            CHOICE {
                short
                    SEQUENCE {
                        extended-length
                            BOOLEAN, -- FALSE
                        s-type
                            ShortBCtSigType,
                        s-length
                            INTEGER (1..8)
                    },
                long
                    SEQUENCE {
                        extended-length
                            BOOLEAN, -- TRUE,
                        sdu-type
                            BCtSigType,
                        length
                            INTEGER (1..256)
                    }
            },
        sdu-payload
            SDUPayload
    }
```

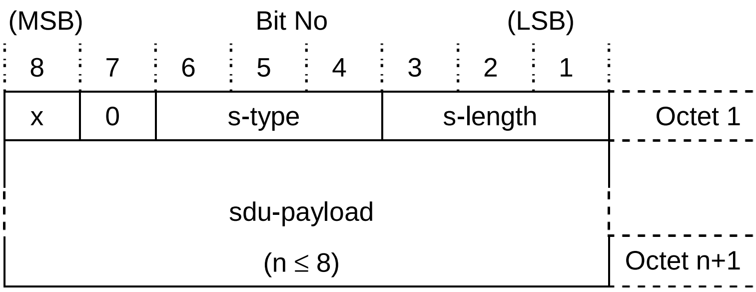


Figure 5.38: BCtSDU Structure (Short Length)

(MSB)			Bit No				(LSB)	
8	7	6	5	4	3	2	1	
x	1	sdu-type						Octet 1
length								Octet 2
sdu-payload (n > 8)								Octet n+2

Figure 5.39: BCtSDU Structure (Extended Length)

5.3.2 BCtSDU Parameters

5.3.2.1 Bct-sdu-follows

This is a single-bit flag which, when set to '1' indicates that another BCtSDU follows this BCtSDU, else either the CRC, ALComPDU or a BCnPDU follows this BCtSDU.

5.3.2.2 Extended-length

This is a single-bit flag which, when set to '1' indicates that an extended type and length field are utilized within the BCtSDU header.

5.3.2.3 S-length / length

The *s-length* and *length* fields are a 3-bit and 8-bit value respectively, both of which are used to indicate the length of the SDU. The value of *s-length* / *length* represents the number of octets in the *sdu-payload*. The encoding of these parameters, which are defined as INTEGER(1..8) and INTEGER (1..256) respectively, follows the ASN.1 Packed Encoding Rules (PER - see clause 4.4.4) hence zero length payloads are not supported.

5.3.2.4 BCtSigType and ShortBCtSigType

The *s-type* / *sdu-type* parameter defines the Bearer Control Signalling Data Unit type. Values in the range 0x00..0x07 shall be encoded as a ShortBCtSigType, in a 3-bit field (i.e. the BCtSDU *type-and-length* field is of CHOICE *short)*, unless the length of the SDU exceeds 8 bytes, in which case they shall be encoded as a BCtSigType, a 6-bit field. In this case the BCtSDU *type-and-length* field is of CHOICE *long.* Values in the range 0x08..0x3F are always encoded as a BCtSigType, a 6-bit field (i.e. *type-and-length* field is of CHOICE *long)*. Different coding for this field is used in the From-UE and To-UE directions.

```
BCtSigType ::=
    CHOICE {
        to-mobile-sdu-type
            INTEGER {
                conn-pdu (0),
                bulletin-board (1),
                return-schedule (2),
                status-ack-list (3),
                specific-avp-list (6),
                avp-list (7),
                spot-beam-map (8),
                bearer-tables (9),
                gps-ephemeris (10),
                -- reserved (11)
                system-info-index(12),
                bearer-table-update(13)
                -- placeholder for priority-spot-beam-map (14)
            } (0..63),
        from-mobile-sdu-type
            INTEGER {
                conn-pdu (0),
                status (1),
                q-rate (4),
                avp-list (7)
            } (0..63)
    }
```

```
ShortBCtSigType ::=
    CHOICE {
        to-mobile-sdu-type
            INTEGER {
                conn-pdu (0),
                bulletin-board (1),
                return-schedule (2),
                status-ack-list (3),
                specific-avp-list (6),
                avp-list (7)
            } (0..7),
        from-mobile-sdu-type
            INTEGER {
                conn-pdu (0),
                status (1),
                q-rate (4),
                q-len (5),
                avp-list (7)
            } (0..7)
    }
```

5.3.2.5 SDUPayload

This field contains the body of the bearer control signalling PDU and is type dependent.

```
SDUPayload ::=
    CHOICE {
        conn-pdu
            BCnPDU,
        status
            Status,
        q-rate
            QRate,
        q-len
            QLen,
        bulletin-board
            BulletinBoard,
        return-schedule
            ReturnSchedule,
        status-ack-list
            StatusAckList,
        specific-avp-list
            SpecificAVPList,
        avp-list
            AVPList,
        spotbeam-map
            SpotBeamMap,
        bearer-tables
            BearerTables,
        gps-ephemeris
            GPSEphemeris,
        system-info-index
            SystemInfoIndex,
        bearer-table-update
            BearerTableUpdate
    }
```

5.4 Bearer Control SDU Payloads

5.4.0 General

NOTE: Throughout clauses 5.4.1 to 5.4.15, the ASN.1 defines components of SDUPayload, while the figures illustrate the entire BCtSDU structure (i.e. including the fields *bct-sdu-follows* and *type-and-length*).

5.4.1 ConnPDU

The ConnPDU SDU is transferred in a connection specific BCtPDU and is **only** used when multiple Bearer Connection PDUs are to be transferred within a single Bearer Control PDU. The SDU Payload contains a BCnPDU and the format is shown in Figures 5.40 and 5.41.

(MSB)			Bit No				(LSB)	
8	7	6	5	4	3	2	1	
x	0	0	0	0	s-length			Octet 1
BCnPDU								
								Octet n ≤ 9

Figure 5.40: ConnPDU Bearer Control SDU (Short Length)

(MSB)			Bit No				(LSB)	
8	7	6	5	4	3	2	1	
x	1	0	0	0	0	0	0	Octet 1
length								Octet 2
BCnPDU								
								Octet n ≤ 258

Figure 5.41: ConnPDU Bearer Control SDU (Extended Length)

If the Bearer Connection needs to send HDLC signalling information, such as a selective reject (SREJ), or if only a small segment is to be transferred, it may not utilize the entire allocation offered by the transmit scheduler within the Bearer Control process. When this occurs, the transmit scheduler within the Bearer Control process **may** offer the residue of the allocation to the same Bearer Connection process. This BCtSDU should be used when the transmit scheduler receives multiple, consecutive segments from the same Bearer Connection process for transmission in the same block or slot (subject to the maximum size limit of a BCtPDU).

If the transmit scheduler offers resource to another Bearer Connection before offering the residue of a block or slot to the current Bearer Connection, the Bearer Connection PDUs will be encapsulated within different Bearer Control PDUs.

Where one or more ConnPDU SDUs are used to carry multiple segments in a BCtPDU, the first segment shall be carried in the first ConnPDU and the last segment in the BCt-Payload field of the BCtPDU. Similarly, on reception of a BCtPDU containing one or more ConnPDU SDUs, the ordering of the data is such that the first ConnPDU SDU is processed first, followed by the next ConnPDU SDU, until all ConnPDU SDUs are processed. The data contained in the BCt-Payload is processed last.

5.4.2 Status

5.4.2.0 General

The Status SDU is transmitted in a BCtPDU to transfer information about the queue status for a specific connection at a UE to the RNC and is used for the purpose of requesting resources. A detailed description on the use of this SDU is provided in ETSI TS 102 744-3-2 [10]. The SDU-Payload has the following structure:

```
Status ::=
    SEQUENCE {
        sequence-number
            SequenceNumber,
        retry-count
            INTEGER (0..15),
        queue-length
            SEQUENCE {
                exponent
                    INTEGER (0..15),
                mantissa
                    INTEGER (0..1023)
            },
        time-head
```

```
        INTEGER (0..511),
    delivery-rate
        INTEGER (0..511),
    status-avp-list
        AVPList OPTIONAL
}
```

The format of the BCtSDU when carrying a Status PDU is as shown in Figure 5.42.

(MSB) 8	7	6	5 (Bit No)	4	3	2	(LSB) 1	
x	0	0	0	1	1	0	0	Octet 1
sequence-number				retry-count				Octet 2
exponent				<				Octet 3
mantissa					>	<		Octet 4
time-head						>	<	Octet 5
delivery-rate							>	Octet 6

Figure 5.42: Status Bearer Control SDU

The *retry-count* parameter shall be set to zero if the Status SDU is transmitted in a reserved slot. If transmitted in a contention slot, *retry-count* shall initially be set to one and incremented for each retransmission required (in a contention slot). The format of the BCtSDU when carrying a Status PDU with *status-avp-list* included is as shown in Figure 5.43.

(MSB) 8	7	6	5 (Bit No)	4	3	2	(LSB) 1	
x	1	0	0	0	0	0	1	Octet 1
length								Octet 2
sequence-number				retry-count				Octet 3
exponent				<				Octet 4
mantissa					>>	<		Octet 5
time-head						>	<	Octet 6
delivery-rate							>	Octet 7
status-avp-list								
								Octet n

Figure 5.43: Status Bearer Control SDU with optional Status AVP List

5.4.2.1 SequenceNumber

This field contains a *sequence-number* which is incremented for each Status Message sent.

```
SequenceNumber ::=
    INTEGER (0..15)
```

5.4.2.2 Queue-length

This parameter provides the *queue-length* in units of bytes. It is divided into two parts, a four bit *exponent* and a ten bit *mantissa*. The queue length can then be calculated as follows:

$$queue\text{-}length = mantissa \times 2^{exponent}$$

When sending the parameter, *queue-length* shall be expressed with the maximum possible precision.

5.4.2.3 Time-head

Time-head refers to the delivery time of the first PDU in the calling process queue. The units of *time-head* are in 80 ms units, offset by the time the Status SDU was created minus 8 s. The expected delivery times of the head of the queue is therefore calculated from the value *time-head* according to the following formula:

$$Expected\text{-}delivery\text{-}time = time\text{-}head \times 0{,}08\ s + time\text{-}of\text{-}creation\text{-}of\text{-}SDU - 8\ s$$

If the delivery time is too large to fit in the field, the maximum field value 0x1FF shall be used. If the delivery time was more than 8 s earlier than the SDU creation time, a field value of 0x000 shall be used.

The parameter is intended for the purpose of policing QoS. If *expected-delivery-time* is more than (after) *time-of-creation*, the queue is currently ahead of schedule, exceeding the QoS. A negative value of (*expected-delivery-time* minus *time-of-creation*) means that the rate of servicing the queue up to now has fallen below minimum QoS, and that the rate needs to be increased to catch up. To report a negative value an offset of 8 seconds is added to the definition of *time-head*. Values implying a lag by more than 8 seconds are clipped to 8 seconds, as the service QoS is already critical; 8 seconds is a value chosen to represent deliveries significantly failing QoS.

5.4.2.4 Delivery-rate

The *delivery-rate* parameter gives the required delivery rate as calculated by the UE such that the entire queue at the UE can be delivered meeting the specified QoS. The parameter is defined in units of 128 bytes/s.

5.4.2.5 Status-avp-list

The *status-avp-list* may carry information from the UE to the RNC, such as Received Signal Quality. This parameter is of type AVPList which is defined as follows:

```
AVPList ::=
    SEQUENCE OF BCtAVP
```

The data type BCtAVP is defined in clause 5.7.

5.4.3 BulletinBoard

5.4.3.0 General

The BulletinBoard SDU is transferred in a Broadcast BCtPDU used to transfer information to all UEs about the current bearer. The BulletinBoard SDU is transmitted at regular intervals, but not necessarily in every frame. If a BulletinBoard SDU is scheduled for transmission in a particular frame, then:

- the BulletinBoard SDU shall be the first BCtSDU within the Broadcast BCtPDU (i.e. immediately after the BCtPDU header); and
- this Broadcast BCtPDU shall be transmitted as the first BCtPDU in the first FEC Block (see ETSI TS 102 744-2-1 [8]) of the frame.

The SDU Payload has the following structure, with format shown in Figures 5.44 to 5.46:

```
BulletinBoard ::=
    SEQUENCE {
        rnc-id
            INTEGER (0..255),
        net-ver
            INTEGER (0..15),
        frame-no
            FrameNumber,
        sb-presence
```

```
        CHOICE {
            sb-not-present
                SEQUENCE {
                    spot-beam-present
                        BOOLEAN,    -- FALSE
                    f-bearer
                        FwdBearer,
                    bct-id
                        BCtID
                },
            sb-present
                SEQUENCE {
                    spot-beam-present
                        BOOLEAN,    -- TRUE
                    f-bearer
                        FwdBearer,
                    bct-id
                        BCtID,
                    spot-beam-id
                        SpotBeamID
                    }
        },
    bb-avp-list
        AVPList OPTIONAL
}
```

(MSB) 8	7	6	5 Bit No	4	3	2	(LSB) 1	
x	0	0	0	1	0	1	1	Octet 1
rnc-id								Octet 2
net-ver				<				Octet 3
frame-no							>	Octet 4
0	f-bearer			bct-id				Octet 5

Figure 5.44: BulletinBoard Bearer Control SDU (*sb-not-present*)

(MSB) 8	7	6	5 Bit No	4	3	2	(LSB) 1	
x	0	0	0	1	1	0	0	Octet 1
rnc-id								Octet 2
net-ver				<				Octet 3
frame-no							>	Octet 4
1	f-bearer			bct-id				Octet 5
spot-beam-id								Octet 6

Figure 5.45: BulletinBoard Bearer Control SDU (*sb-present*)

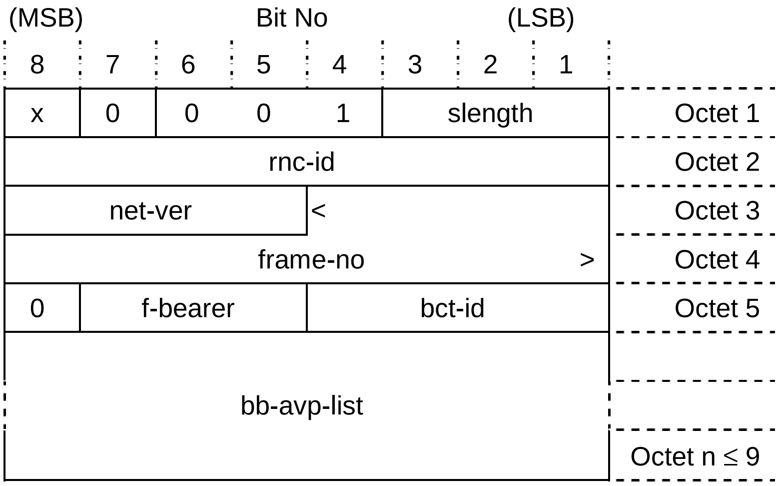


Figure 5.46: BulletinBoard Bearer Control SDU (*sb-not-present* with *bb-avp-list*)

5.4.3.1 Rnc-id

This 8-bit field is used to identify a RNC within a region. The purpose of the field is to assist with the process of recovery in the event that a channel is allocated to another RNC.

5.4.3.2 Net-ver

This 4-bit field is incremented whenever the RNC performs a software reset, or whenever the information held about UEs by the particular bearer may be obsolete. The purpose of the field is to assist with the process of recovery in the event of a problem within the RNC.

5.4.3.3 FrameNumber

This field is a 12-bit field of type INTEGER (0x000 .. 0xFFF). The value is used to identify the Frame Number of the Bearer Control at the RNC. The purpose of the field is to synchronize sleep mode and ciphering operation between the RNC and the UE.

```
FrameNumber ::=
    INTEGER (0..4095)
```

5.4.3.4 Spot-beam-present

This BOOLEAN field indicates whether the optional *spot-beam-id* field is present.

5.4.3.5 FwdBearer

This field is a 3-bit field which is used to identify this Forward Bearer Number within the Bearer Control. The purpose of the field is to assist with the process of locating the UE within the Bearer Control.

5.4.3.6 BCtID

This field is a 4-bit field which is used to identify this Bearer Control within the Spot Beam (identified by *spot-beam-id*, if present) at a particular RNC (identified by the *rnc-id*). The purpose of the field is to identify error conditions within the UE.

```
BCtID ::=
    INTEGER (0..15)
```

5.4.3.7 SpotBeamID

The *spot-beam-id* field specifies the spot-beam in which the bearer is being transmitted.

```
SpotBeamID ::=
    INTEGER (0..255)
```

5.4.3.8 BB-avp-list

The *bb-avp-list* field is used to carry various configuration parameters which are intended for all UEs on the forward bearer. The field is of data type AVPList which is defined as follows:

```
AVPList ::=
    SEQUENCE OF BCtAVP
```

The data type BCtAVP is defined in clause 5.7.

5.4.4 ReturnSchedule

5.4.4.0 General

The ReturnSchedule SDU is sent in a Broadcast BCtPDU to transfer information to UEs about the availability of slots in the return direction, and may also be transmitted to a specific UE to allocate dedicated return channel resources. The SDU Payload has the following structure:

```
ReturnSchedule::=
    SEQUENCE {
        return-channel-no
            RetChannelNumber,
        bearer-type-and-spotid-included
            BOOLEAN,
        res-plan-included
            BOOLEAN,
        number-of-slot-plans
            INTEGER (1..4),
        bearer-type-and-spotid
            SEQUENCE {
                r-bearer-type
                    ReturnBearerTypeShort,
                spot-beam-id
                    SpotBeamID
            } OPTIONAL,
        slot-plans
            SEQUENCE SIZE (1..4) OF SlotPlan,
        res-plan
            ResourcePlan OPTIONAL,
        tbcn-id-list
            SEQUENCE SIZE (0..32) OF TranslatedBearerConnectionID
    }
```

5.4.4.1 RetChannelNumber

5.4.4.1.0 General

This field is used to specify the Channel Number to which the Return Schedule applies.

```
RetChannelNumber ::=
    SEQUENCE {
        offset
            Offset,
        channel-index
            ChannelIndex
    }
```

If RetChannelNumber is 0 or 0xFFFF, this indicates that the channel is unavailable.

5.4.4.1.1 Offset

The *offset* field is a one bit field defined as follows:

```
Offset ::=
    INTEGER (0..1)
```

It is interpreted as shown in Table 5.4.

Table 5.4: Offset Field Values

Offset Value:	Interpretation
0:	0 Hz offset
1:	+1,25 kHz offset

5.4.4.1.2 ChannelIndex

The remaining 15 bits represent the *channel-index* field, which is expressed as a hexadecimal number, such that the L-Band transmit frequency can be calculated as described in ETSI TS 102 744-2-1 [8]. The return channel centre frequency is calculated from *channel-index* and *offset* as follows:

$$\textit{return frequency [MHz]} = \textit{channel-index} \times 0{,}0025 + \textit{offset} \times 0{,}00125 + 1\,611{,}500$$

The data type ChannelIndex is defined as follows:

```
ChannelIndex ::=
    INTEGER (0..32767)
```

5.4.4.2 Bearer-type-and-spotid-included

The *bearer-type-and-spot-id-included* field is of type BOOLEAN, and if set to '1' indicates that the *bearer-type-and-spot-id* sequence is included.

5.4.4.3 Res-plan-included

The *res-plan-included* field is of type BOOLEAN, and if set to '1' indicates that the *res-plan* field is present.

5.4.4.4 Number-of-slot-plans

This field is used to specify the number of slot-plans included in the ReturnSchedule SDU.

5.4.4.5 ReturnBearerTypeShort

The value in *r-bearer-type* field specifies the type of return bearer that shall be used by the UE in conjunction with the particular Return Schedule and the data type ReturnBearerTypeShort is defined as follows:

```
ReturnBearerTypeShort::=
    INTEGER {
        r20t05q-or-r80t05q (0),
        r20t1q (1),
        r5t2q-or-r20t2q (2),
        r5t45q-or-r20t45q (3),
        r20t1q-or-r80t1q (4),
        r5t1x-or-r20t1x (5),
        r5t2x-or-r20t2x (6),
        r5t45x-or-r20t45x (7),
        r80t25x4 (8),
        r80t5x4 (9),
        r80t25x16 (10),
        r80t5x16 (11),
        r80t25x32 (12),
        r80t5x32 (13),
        r80t25x64 (14),
        r80t5x64 (15)
    } (0..15)
```

Bearer types are defined in ETSI TS 102 744-2-1 [8].

ReturnBearerTypeShort does not make a distinction based upon the use of the Distributed Unique Word burst format as it is implied by the UE Class.

If the field *r-bearer-type* is not included then the ReturnBearerTypeShort is broadcast to UEs in the ReturnBearerType AVP (see clause 5.7.3).

5.4.4.6 SpotBeamID

The *spot-beam-id* field is used if the bearer control at the RNC supports return channels associated with multiple spot beams; e.g. when the forward bearer (on which this schedule is sent) is transmitted in the regional beam but narrow beams are available to receive return channel transmissions. In this case, the *spot-beam-id* field specifies the spot beam to which the return schedule applies. The value of zero (0x00) is specifically used as a global beam identifier while the value of 255 (0xFF) shall be treated by the UE as if the *spot-beam-id* field was not present. The data type SpotBeamID is defined in clause 5.4.4.7.

5.4.4.7 SlotPlan

Values of s*lot-plans* are defined as shown in Figure 5.47.

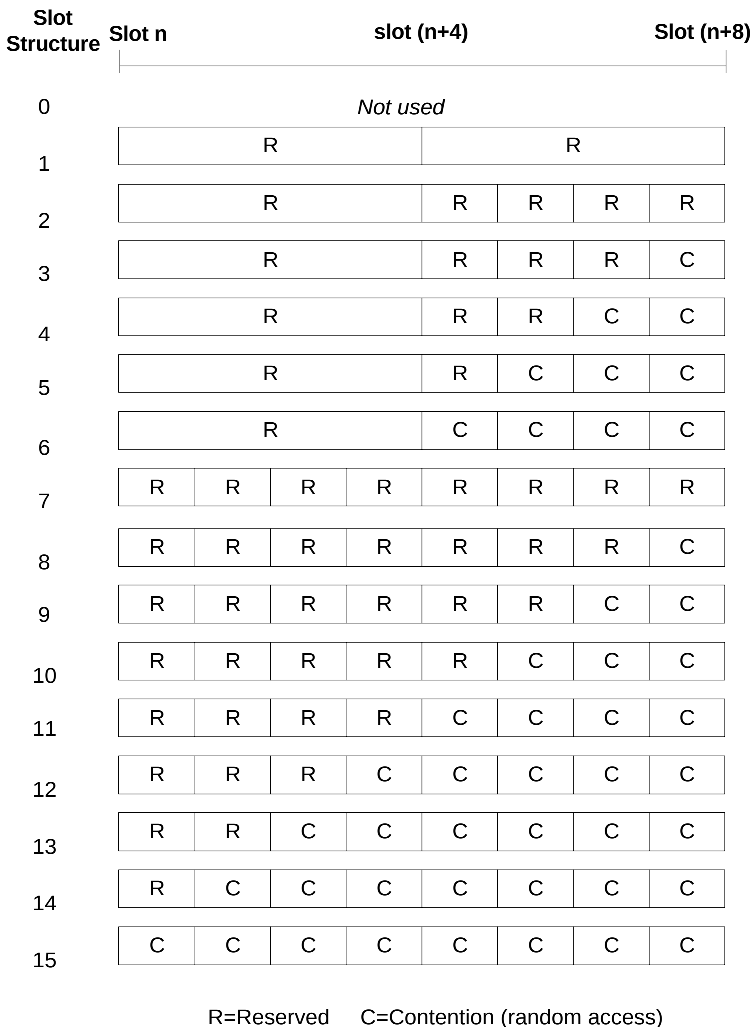


Figure 5.47: SlotPlan Structure

The values of s*lot-plan* are defined in relation to a slot number, where four (short) slots may be combined to provide a long (reserved) slot, as is the case in slot structures 1 to 6. The actual slot duration (in ms) depends on the return bearer type (see clause 5.4.5.5) in use as shown in Table 5.5.

Table 5.5: Slot Durations vs. Return Bearer Types

R-Bearer-Type	Short Slot Duration	Long Slot Duration
0, 4	20 ms	80 ms
1	n/a	20 ms
2, 3, 5, 6, 7	5 ms	20 ms
8 to 15	80 ms	(not used)

5.4.4.8 ResourcePlan

The *res-plan* field, if present, indicates whether or not reserved slots for the slot plan(s) have Translated Bearer Connection IDs listed below. The mapping of Translated Bearer Connection IDs to reserved slots is explained in ETSI TS 102 744-3-2 [10].

If *res-plan* is not present, then Translated Bearer Connection IDs are included for **all** reserved slots in the slot plan, or there are no reserved slots in the plan.

If *res-plan* is present, each bit in *res-plan* is associated with a reserved slot in the slot plan. Starting with the first reserved slot in the first slot plan, working through the first slot plan and then starting with the first reserved slot in the second slot plan, etc., each reserved slot is taken in turn and has a corresponding bit in *res-plan*. The number of reserved slots depends on the number of slot plans specified as well as the actual slot plan(s) specified. The number of reserved slots for each slot plan can be determined from Figure 5.47. The total number of corresponding bits required in *res-plan* is then equal to the sum of the reserved slots specified in each of the slot plans included in the Return Schedule.

The resulting *res-plan* field is constructed such that it is aligned to a 4-bit boundary. It is packed such that the first (most-significant bit) relates to the first reserved slot etc., up to the total number of corresponding bits. Any unused bits (up to the next four-bit boundary) at the end of *res-plan* are set to zero.

As a result of the above, the ResourcePlan type is defined as follows:

```
ResourcePlan ::=
    CHOICE {
        res-plan4
            BIT STRING (SIZE  (4)),
        res-plan8
            BIT STRING (SIZE  (8)),
        res-plan12
            BIT STRING (SIZE (12)),
        res-plan16
            BIT STRING (SIZE (16)),
        res-plan20
            BIT STRING (SIZE (20)),
        res-plan24
            BIT STRING (SIZE (24)),
        res-plan28
            BIT STRING (SIZE (28)),
        res-plan32
            BIT STRING (SIZE (32))
}
```

5.4.4.9 Return Schedule Examples

This clause provides examples of different Return Schedules and shall not be considered as exhaustive.

The example shown in Figure 5.48 illustrates the case where only contention slots are available on this channel for mobiles tuned to this bearer.

(MSB)			Bit No				(LSB)	
8	7	6	5	4	3	2	1	
x	0	0	1	0	0	1	0	Octet 1
return-								Octet 2
channel-no								Octet 3
0	0	0	0	slot-plan = 15				Octet 4

Figure 5.48: ReturnSchedule Example: Contention Slots only

Figure 5.49 illustrates an example of a Return Schedule using a short BCtSDU structure with the maximum length of eight octets in the Return Schedule itself. Up to three tbcn-ids can be included in this case, leaving sufficient space for a *res-plan4* field. Since this Resource Plan is limited to four bits, the maximum number of reserved slots is limited to four. The example shows *slot-plan* as either 3 or 11 (both slot plans have four reserved slots). Slot Plans with more than four reserved slots would require an eight bit *res-plan*, while for slot plans with less than four reserved slots, there would be no requirement to include a *res-plan* and/or there would be too many tbcn-ids included. Suitable values for *res-plan* in this example would be 0111, 1011, 1101 or 1110. The bitmap positions set to '1' select three out of the four reserved slots specified in *slot-plan* and associate the translated Bearer Connection IDs with the selected slots.

(MSB) 8	7	6	5 Bit No	4	3	2	1 (LSB)	
x	0	0	1	0	1	1	1	Octet 1
return-								Octet 2
channel-no								Octet 3
0	1	0	0	slot-plan = 3 or 11				Octet 4
res-plan4				<				Octet 5
tbcn-id 1 >								Octet 6
< tbcn-id 2								Octet 7
>				<				Octet 8
tbcn-id 3 >								Octet 9

Figure 5.49: ReturnSchedule Example: Res-Plan included

Figure 5.50 illustrates an example similar to the one above, however, for a value of *slot-plan* equal to 4 or 12 (i.e. three reserved slots). In this case, there is no requirement to include a *res-plan* field and the *tbcn-id-list* starts immediately after the *slot-plan* field. This leaves the last four bits in octet 9 unused, which therefore have to be zero padded. These two examples also illustrate that the first tbcn-id (represented by *tbcn-id 1* in the figures) may start on any 4-bit boundary.

(MSB) 8	7	6	5 Bit No	4	3	2	1 (LSB)	
x	0	0	1	0	1	1	1	Octet 1
return-								Octet 2
channel-no								Octet 3
0	0	0	0	slot-plan = 4 or 12				Octet 4
< tbcn-id 1								Octet 5
>				<				Octet 6
tbcn-id 2 >								Octet 7
< tbcn-id 3								Octet 8
>				zero padding				Octet 9

Figure 5.50: ReturnSchedule Example: All reserved slots allocated

Figure 5.51 illustrates a case where two slot plans are included in one Return Schedule, which for return bearers with a slot duration of 5 ms (for short slots) represents a schedule period of 80 ms. The slot plans chosen for this example provide a total of three reserved slots over the entire schedule period.

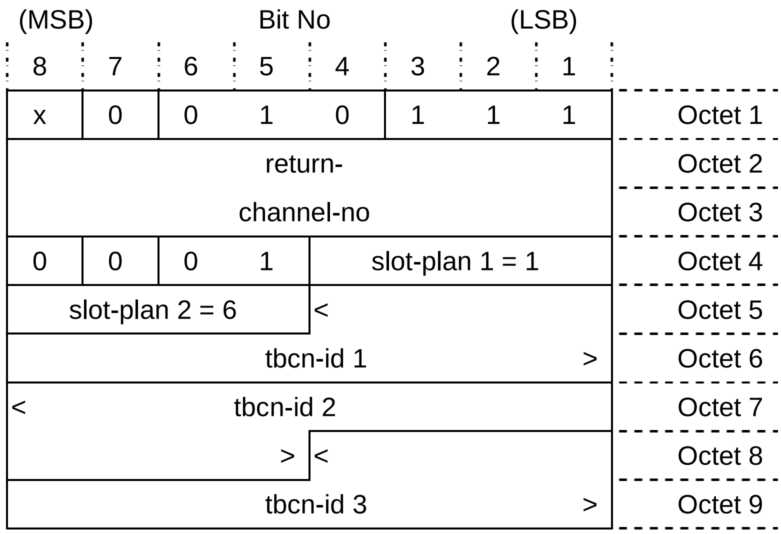


Figure 5.51: Return Schedule Example: Two Slot Plans included

A representation of a return burst slot plan associated with the example schedule (assuming slot durations of 20 ms and 5 ms), together with the allocation of tbcn-ids, is shown in Figure 5.52.

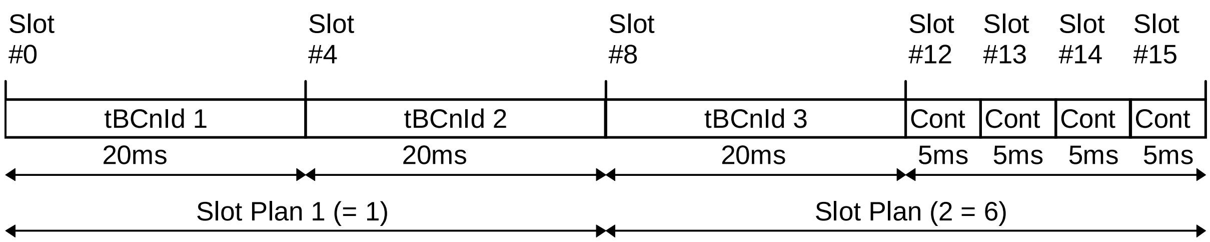


Figure 5.52: Example Return Slot Plan for 80 ms Schedule Period

Figure 5.53 illustrates an example of a Return Schedule using a long BCtSDU structure and where the *bearer-type* and *spot-beam-id* fields are included. In this particular example, no *res-plan* is included, so for all reserved slots a corresponding tbcn-id is included in the *tbcn-id-list*.

Although all the examples provided in this clause are of return schedules with single, dual and triple slot-plans, quadruple slot-plans are also possible (corresponding to a return schedule period of 160 ms for return bearers with a slot duration of 5 ms for short slots).

(MSB) 8	7	6	5	Bit No 4	3	2	(LSB) 1	
x	1	0	0	0	0	1	0	Octet 1
length								Octet 2
return-								Octet 3
channel-no								Octet 4
1	0	1	0	r-bearer-type				Octet 5
spot-beam-id								Octet 6
slot-plan 1				slot-plan 2				Octet 7
slot-plan 3				<				Octet 8
tbcn-id 1 >								Octet 9
< tbcn-id 2								Octet 10
>				<				Octet 11
tbcn-id 3 >								Octet 12
tbcn-id-list								
(continued)								Octet n

Figure 5.53: ReturnSchedule Example: *bearer-type-and-spotid* included

All of the above examples are appropriate for when the Return Schedule is transmitted in a broadcast BCtPDU for allocation of resources to multiple mobile terminals. The Return Schedule is also used to allocate dedicated resources to individual mobile terminals, in which case it will be transmitted in a BCtPDU addressed to a specific mobile terminal. In this case there is no requirement for inclusion of tBCnID list as the tBCnID is implied by the BCtPDU addressing mechanism. This mode of operation is typically used for allocation of resources to a specific mobile terminal for High Data Rate (HDR) operation. Figure 5.54 illustrates the ReturnSchedule for allocation of resources to a specific mobile terminal, this example allocating resources using a single slot plan (scheduling period is 640 ms).

(MSB) 8	7	6	5	Bit No 4	3	2	(LSB) 1	
x	1	0	0	0	0	1	0	Octet 1
Length								Octet 2
return-								Octet 3
channel-no								Octet 4
1	1	0	0	r-bearer-type				Octet 5
spot-beam-id								Octet 6
slot-plan				Res-plan				Octet 7

Figure 5.54: ReturnSchedule Example for use within addressed BCtPDU

5.4.5 StatusAckList

The StatusAckList SDU is transferred in a Broadcast BCtPDU and is used to acknowledge the receipt of status messages from UEs. The SDU Payload has the following structure:

```
StatusAckList::=
    SEQUENCE (1..64) OF StatusAck
```

with StatusAck defined as follows:

```
StatusAck ::=
    SEQUENCE {
        sequence-number
            SequenceNumber,
        tbcn-id
            TranslatedBearerConnectionID
    }
```

where *sequence-number* is equal to the sequence number (defined in clause 5.4.2.1) sent in the Status SDU to be acknowledged. The format is as shown in Figure 5.55.

(MSB) 8	7	6	Bit No 5	4	3	2	(LSB) 1	
x	0	0	1	1	s-length			Octet 1
sequence-number 1				<				Octet 2
tbcn-id 1							>	Octet 3
sequence-number 2				<				Octet 4
tbcn-id 2							>	Octet 5
status-ack-list (continued)								Octet n ≤ 9

Figure 5.55: StatusAckList Bearer Control SDU (short length)

An extended length field is required to describe the length of the schedule if more than four tbcn-ids are acknowledged in the list, as shown in Figure 5.56.

(MSB) 8	7	6	Bit No 5	4	3	2	(LSB) 1	
x	1	0	0	0	0	1	1	Octet 1
length								Octet 2
sequence-number 1				<				Octet 3
tbcn-id 1							>	Octet 4
sequence-number 2				<				Octet 5
tbcn-id 2							>	Octet 6
status-ack-list (continued)								Octet n ≤ 130

Figure 5.56: StatusAckList Bearer Control SDU (extended length)

5.4.6 QRate

The QRate SDU is transmitted in a BCtPDU to transfer information about the characteristics of the traffic being presented at the input to a specific connection to the RNC. The SDU-Payload currently only carries a QueueRate parameter and is formatted as below with structure as shown in Figure 5.57.

```
QRate ::=
    SEQUENCE {
        qr-range
```

```
            QRateRange,
        qr-value
            QRateValue
    }
```

where

```
QRateRange ::=
    INTEGER (0..3)
QRateValue ::=
    INTEGER (0..63)
```

(MSB)			Bit No				(LSB)	
8	7	6	5	4	3	2	1	
x	0	1	0	0	0	0	0	Octet 1
qr-range		qr-value						Octet 2

Figure 5.57: QRate SDU

The default definition of the QRate SDU is to carry a *queue-rate* parameter that gives the required delivery rate as calculated by the UE such that the entire queue at the UE can be delivered while meeting the specified QoS. The parameter is defined as shown in Table 5.6.

Table 5.6: QRate Range and Value Interpretation

QRate Range	Minimum Rate (kbit/s)	QRate Value units	Rate Range (kbit/s)
00	0	5 Octets per 80 ms (500 bps)	0 to 31,5
01	32	10 Octets per 80 ms (1 kbps)	32 to 95
10	96	20 Octets per 80 ms (2 kbps)	96 to 222
11	224	40 Octets per 80 ms (4 kbps)	224 to 476

The UE shall periodically update the QRate information towards the RNC as shown in the Default Update Rate column of the above table. The approach for calculation of the value to be provided in this SDU by the UE is described in ETSI TS 102 744-3-2 [10].

5.4.7 QLen

The QLen SDU is transmitted in a BCtPDU to transfer information about the state of the queue for a specific connection to the RNC and is used for the purpose of requesting resources for Background class connections and Interactive class connections with low information rate. A detailed description on the use of this SDU is provided in ETSI TS 102 744-3-2 [10]. The SDU-Payload currently only carries a Queue Length parameter and is formatted as follows:

```
QLen ::=
    SEQUENCE {
        ql-range
            QLenRange,
        ql-value
            QLenValue
    }
```

where:

```
QLenRange ::=
    INTEGER (0..3)
QLenValue ::=
    INTEGER (0..63)
```

The format of the BCtSDU when carrying a QLen SDU containing only queue length information is as shown in Figure 5.58.

8 (MSB)	7	6	5	4	3	2	1 (LSB)	
x	0	1	0	1	0	0	0	Octet 1
ql-range		ql-value						Octet 2

Figure 5.58: QLen SDU

The purpose of the QLen SDU is to carry a *queue-length* parameter that reports the required volume of data for the entire queue at the UE. The parameter is defined as shown in Table 5.7.

Table 5.7: QLen Range and Value Interpretation

QLen Range	Minimum Value (octets)	QLen Value Unit (octets)	Qlength Range supported (octets)
00	0	32	0 to 2 016
01	2 048	128	2 048 to 10 112
10	10 240	512	10 240 to 42 496
11	43 008	2 048	43 008 to 172 032

If the queue length exceeds 172 032 octets, then the maximum value 172 032 octets shall be reported, and continue to be reported when the Significant Delta QLen volume of data has been transmitted. The approach for calculation of the value to be provided in this SDU by the UE is described in ETSI TS 102 744-3-2 [10].

5.4.8 SpecificAVPList

The SpecificAVPList SDU may be used within a Broadcast BCtPDU to direct information at a specific UE, a connection within a UE, or hardware within a UE (depending upon the AVP context), for instance when a timing, frequency or power correction is required. The SDU Payload has the following structure, with format as shown in Figures 5.59 and 5.60:

```
SpecificAVPList ::=
    SEQUENCE {
        reserved
            BIT STRING (SIZE(4)),
        tbcn-id
            TranslatedBearerConnectionID,
        bct-avp-list
            SEQUENCE OF BCtAVP
    }
```

8 (MSB)	7	6	5	4	3	2	1 (LSB)	
x	0	1	1	0	s-length			Octet 1
reserved				<				Octet 2
tbcn-id							>	Octet 3
bct-avp-list								
								Octet n ≤ 9

Figure 5.59: SpecificAVPList Bearer Control SDU (short length)

(MSB)			Bit No				(LSB)	
8	7	6	5	4	3	2	1	
x	1	0	0	0	1	1	0	Octet 1
length								Octet 2
reserved				<				Octet 3
tbcn-id							>	Octet 4
bct-avp-list								
								Octet n

Figure 5.60: SpecificAVPList Bearer Control SDU (extended length)

The data type TranslatedBearerConnectionID is defined in clause 5.1.5.10, while BCtAVP is specified in clause 5.7.

5.4.9 AVPList

The AVPList SDU may be used in any of the following ways:

- in a Broadcast BCtPDU to change global parameters within all UEs tuned to the forward bearer;
- in a Common Signalling BCtPDU (RegisterAck) to carry timing and return link reference level information to a UE; and
- in a Connection Specific BCtPDU (i.e. which is addressed by a tbcn-id) to change parameters in a specific UE, in a specific Bearer Control within a UE, or specific hardware within a UE.

If the AVPList SDU is contained in a Broadcast BCtPDU (referred to as a broadcast AVPList throughout the present document), then it shall be transmitted as the first BCtSDU in the BCtPDU (i.e. immediately after the BCtPDU header). In addition, if the ForwardBearerCodeRate AVP is present, the AVPlist shall be transmitted in the first BCtPDU of an FEC block. If a BulletinBoard SDU is scheduled for transmission in the same FEC Block (see ETSI TS 102 744-2-1 [8]) then the bb-avp-list (see clause 5.4.4.8) shall be used to carry these AVPs instead (i.e. within a Broadcast BCtPDU, AVPList SDU and BulletinBoard SDU are mutually exclusive).

The SDU Payload has the following structure, with format as shown in Figures 5.61 and 5.62:

```
AVPList ::=
    SEQUENCE OF BCtAVP
```

(MSB)			Bit No				(LSB)	
8	7	6	5	4	3	2	1	
x	0	1	1	1	s-length			Octet 1
bct-avp-list								Octet 2
								Octet n ≤ 9

Figure 5.61: AVPList Bearer Control SDU (short length)

8 (MSB)	7	6	5	4	3	2	1 (LSB)	
x	1	0	0	0	1	1	1	Octet 1
length								Octet 2
bct-avp-list								Octet 3
								Octet n

Figure 5.62: AVPList Bearer Control SDU (extended length)

The data type BCtAVP is specified in clause 5.7.

5.4.10 SpotBeamMap

5.4.10.0 General

This SDU is used to describe spot beam contour information to the UEs and has the following structure, with format as shown in Figure 5.63:

```
SpotBeamMap::=
    SEQUENCE {
        spotbeam-map-version
            INTEGER (0..255),
        total-number-of-spots
            INTEGER (0..255),
        spotbeam-map-number
            INTEGER (0..255),
        spotbeam-info-list
            SEQUENCE OF SpotBeamInfo
    }
```

8 (MSB)	7	6	5	4	3	2	1 (LSB)	
x	1	0	0	1	0	0	0	Octet 1
length								Octet 2
spotbeam-map-version								Octet 3
total-number-of-spots								Octet 4
spotbeam-map-number								Octet 5
spotbeam-info-list								
								Octet n

Figure 5.63: SpotBeamMap Bearer Control SDU

5.4.10.1 Spotbeam-map-version

The *spotbeam-map-version* field is used to indicate to the UEs that the spotbeam information has been updated and that it should obtain a complete list of spotbeam map information prior to taking any further action.

5.4.10.2 Total-number-of-spots and spotbeam-map-number

The parameters *total-number-of-spots* and *spotbeam-map-number* are provided in order for the UE to track the reception of spotbeam maps if a complete set of spotbeam maps is transmitted in a number of SpotBeamMap SDUs. This would be the case if the capacity of the physical layer is insufficient to transmit all SpotBeamInfo elements in a single SDU, or if the complete *spotbeam-info-list* would exceed the maximum length of the BCtPDU.

The *total-number-of-spots* parameter specifies the total number of spotbeam maps provided in the set. The parameter carries the same value in each of the SpotBeamMap SDUs transmitted.

The *spotbeam-map-number* parameter gives the sequence number of the first SpotBeamInfo element in the SpotBeamMap SDU, i.e. for the first SpotBeamMap SDU in a set, *spotbeam-map-number* equals 1.

5.4.10.3 SpotBeamInfo

5.4.10.3.0 General

The *spot-beam-info* structure is defined as follows, with format as shown in Figures 5.64 and 5.65:

```
SpotBeamInfo::=
    SEQUENCE {
        spotbeam-id
            SpotBeamID,
        reserved
            BIT STRING (SIZE (4)),
        sb-prf
            INTEGER (0..3),
        sb-data
            CHOICE {
                sb-vertices
                    SEQUENCE {
                        sb-type
                            SpotBeamMapType, -- sb-type = 0
                        number-of-vertices
                            INTEGER (0..63),
                        sb-vertex-list
                            SEQUENCE OF SpotbeamVertex
                    }
                regular-sb
                    SEQUENCE {
                        sb-type
                            SpotBeamMapType, -- sb-type = 1
                        spotbeam-azimuth
                            SpotBeamAzimuth,
                        spotbeam-elevation
                            SpotBeamElevation,
                        spotbeam-radius
                            SpotBeamRadius
                    }
            }
    }
```

(MSB) 8	7	6	5	Bit No 4	3	2	(LSB) 1	
spotbeam-id								Octet 1
reserved				sb-prf		0	0	Octet 2
number-of-vertices								Octet 3
spotbeam-vertex-list								Octet 4
								Octet n

Figure 5.64: SpotBeamInfo Structure using Spotbeam-Vertex-List Format

8 (MSB)	7	6	5	4	3	2	1 (LSB)	
spotbeam-id								Octet 1
reserved				sb-prf		0	1	Octet 2
spotbeam-azimuth								Octet 3
spotbeam-elevation								Octet 4
spotbeam-radius								Octet 5

Figure 5.65: SpotBeamInfo Structure using Regular Spotbeam Format

The data type SpotBeamID is defined in clause 5.4.4.7.

5.4.10.3.1 Sb-prf

The value in the *sb-prf* field indicates to the UE the spot beam preference level, i.e. which spot-beam the UE should move into (or the spot beam to which the mobile should make a request to move into). If there is more than one suitable spot beam to chose from then the UE shall choose the one with the highest *sb-prf* value.

5.4.10.3.2 SpotBeamMapType

The value in the two-bit field *sb-type* specifies the format of the remaining octets in the spotbeam-info structure:

```
SpotBeamMapType ::=
    INTEGER {
        spotbeam-vertex-list (0),
        regular-spotbeam (1)
    } (0..3)
```

Current implementations shall ignore the spot-beam information if *sb-type* is set to a reserved value.

5.4.10.3.3 Number-of-vertices

The *number-of-vertices* field specifies the length of the *spot-beam-vertex-list* expressed as the number of vertices (each vertex occupies two octets) as defined in clause 5.4.11.3.4.

5.4.10.3.4 SpotbeamVertex

The data type SpotbeamVertex is defined as:

```
SpotbeamVertex ::=
    INTEGER (0..65535)
```

and the value of a SpotbeamVertex is calculated as follows:

```
SpotBeamVertexValue = [360(ϕ+90)]+θ+180
```

where:

```
ϕ := Latitude in degrees, rounded to the nearest integer with
    (90°S ≡ -90) ≤ ϕ ≤ (90 ≡ 90°N)
```

and:

```
θ := Longitude in degrees, rounded to the nearest integer with
    (180°W ≡ -180) ≤ θ ≤ (179 ≡ 179°E)
```

Values between 0xFE88 and 0xFFFF are reserved.

5.4.10.3.5 SpotBeamAzimuth

The data type SpotbeamAzimuth is defined as:

```
SpotbeamAzimuth ::=
    INTEGER (0..255)
```

This element specifies the centre of a spot beam in terms of the azimuth angle from the satellite to the spot beam centre. The azimuth angle, A_{zc}, is expressed in degrees and encoded as follows:

A_z = (SpotBeamAzimuth-128)/10

5.4.10.3.6 SpotBeamElevation

The data type SpotbeamElevation is defined as:

```
SpotbeamElevation ::=
    INTEGER (0..255)
```

This element specifies the centre of a spot beam in terms of the elevation angle from the satellite to the spot beam centre. The elevation angle, E_{lc}, is expressed in degrees and encoded as follows:

E_l = (SpotbeamElevation -128)/10

5.4.10.3.7 SpotBeamRadius

The data type SpotbeamRadius is defined as:

```
SpotbeamRadius ::=
    INTEGER (0..255)
```

This field describes the boundary of a spot beam as a constant "distance" in terms of satellite-centred azimuth and elevation angles from the satellite, i.e. the radius:

$$R = \sqrt{(A_z - A_{zc})^2 + (E_l - E_{lc})^2}$$

where A_{zc} and E_{lc} are the azimuth and elevation angles which point to the beam centre. The radius R is expressed in degrees and encoded as follows:

R = SpotBeamRadius/20

5.4.11 BearerTables

5.4.11.0 General

The BearerTable SDU may be used by the RNC to signal any changes in the properties of the physical layer return bearer types and subtypes in a connection specific BCtPDU. The implementation of this SDU is mandatory only for UE Classes 1 to 3 of RI-Versions below 0x83 (see ETSI TS 102 744-3-5 [13], clause 6.1.2.2). For all UE Classes from RI-Version 0x83 onwards, this function is provided by the BearerTableUpdate SDU (see clause 5.4.14). The SDU has the following structure:

```
BearerTables ::=
    SEQUENCE (1..12) OF BearerDefinition
```

The data type BearerDefinition is specified as follows, with format as shown in Figures 5.66 to 5.68:

```
BearerDefinition ::=
    SEQUENCE {
        bearer-table-follows
            BOOLEAN,
        lowest-coding-rate-index-included
            BOOLEAN,
        coding-rates-offsets-included
            BOOLEAN,
        coding-rates-usage-included
            BOOLEAN,
        r-bearer-type
```

```
        ReturnBearerTypeFull,
    lowest-coding-rate-index-octet
        SEQUENCE {
            reserved2
                BIT STRING (SIZE (2)),
            lowest-coding-rate-index
                ControlIndex
        } OPTIONAL,
    coding-rates-offsets-list
        SEQUENCE SIZE (10) OF
            coding-rates-offset
                INTEGER (0..3) OPTIONAL,
    coding-rates-usage-bitmap
        BIT STRING (SIZE (12)) OPTIONAL,
    reserved4
        BIT STRING (SIZE (4)) OPTIONAL
        -- padding to start next bearer definition
        -- on octet boundary
}
```

(MSB)			Bit No				(LSB)	
8	7	6	5	4	3	2	1	
x	1	0	0	1	0	0	1	Octet 1
length								Octet 2
btf	lcrii = 1	croi = 1	crui = 1	r-bearer-type				Octet 3
reserved2		lowest-coding-rate-index						Octet 4
offset 1		offset 2		offset 3		offset 4		Octet 5
offset 5		offset 6		offset 7		offset 8		Octet 6
offset 9		offset 10		<				Octet 7
coding-rates-usage-bitmap >								Octet 8
bearer-definition-list (continued)								Octet n

Figure 5.66: BearerTables Bearer Control SDU (all elements present)

8 (MSB)	7	6	5	4	3	2	1 (LSB)	
x	1	0	0	1	0	0	1	Octet 1
length								Octet 2
btf	lcrii = 1	croi = 1	crui = 0	r-bearer-type				Octet 3
reserved2		lowest-coding-rate-index						Octet 4
offset 1		offset 2		offset 3		offset 4		Octet 5
offset 5		offset 6		offset 7		offset 8		Octet 6
offset 9		offset 10		reserved4				Octet 7
bearer-definition-list								
(continued)								Octet n

Figure 5.67: BearerTables Bearer Control SDU (*coding-rates-usage-bitmap* absent)

8 (MSB)	7	6	5	4	3	2	1 (LSB)	
x	1	0	0	1	0	0	1	Octet 1
length								Octet 2
btf	lcrii = 1	croi = 0	crui = 1	r-bearer-type				Octet 3
reserved2		lowest-coding-rate-index						Octet 4
< coding-rates-usage-bitmap								Octet 7
>				reserved4				Octet 8
bearer-definition-list								
(continued)								Octet n

Figure 5.68: BearerTables Bearer Control SDU (*coding-rates-offsets-list* absent)

The BearerTables SDU can be used to describe to the UEs the full set of Return Bearer Types and Subtypes in relation to each other. A default set of Bearer Tables is stored in the UE and hence the Bearer Tables SDU needs to include entries only for those Bearers which are relevant to the particular operational configuration which have changed one or more of its properties. A detailed description of the application of this SDU can be found in ETSI TS 102 744-3-2 [10].

5.4.11.1 Bearer-table-follows

This BOOLEAN flag indicates that the current Bearer Table will be continued in another Bearer Tables SDU. Bearer Tables may need to be split over several SDUs if the number of octets available in a forward bearer FEC block is insufficient to accommodate a full Bearer Table.

5.4.11.2 Lowest-coding-rate-index-included

This BOOLEAN flag indicates that the field *lowest-coding-rate-index* is included in the structure.

5.4.11.3 Coding-rates-offsets-included

This BOOLEAN flag indicates that the *coding-rates-offsets-list* is included in the structure.

5.4.11.4 Coding-rates-usage-included

This BOOLEAN flag indicates that the *coding-rates-usage-bitmap* is included in the structure.

5.4.11.5 ReturnBearerTypeFull

The *r-bearer-type* field specifies the type of return bearer to which the bearer definition refers. The data type ReturnBearerTypeFull is defined as follows:

```
ReturnBearerTypeFull ::=
    INTEGER {
            r20t05q (0),
            r80t05q (1)
            r80t1q (2)
            r20t1q (3),
            r5t2q (4),
            r20t2q (5),
            r5t45q (6),
            r20t45q (7),
            r80t25x (8)
            r80t5x (9)
            r5t1x (10),
            r20t1x (11),
            r5t2x (12),
            r20t2x (13),
            r5t45x (14),
            r20t45x (15)
        } (0..15)
```

5.4.11.6 ControlIndex

The value in the *lowest-coding-rate-index* field is of the following type:

```
ControlIndex ::=
    INTEGER (0..63)
```

Values of ControlIndex provide an index to the columns in the Bearer Tables (see ETSI TS 102 744-2-1 [8]) where a step from one column to the next represents a step of 0,5 dB in Carrier-to-Noise (C/N_0) Ratio. The ControlIndex scale is based on a value of zero being equal to C/N_0 = 36,5 dBHz, with each increment representing an increment in C/N_0 by 0,5 dB. The *lowest-coding-rate-index* defines the leftmost column in the bearer table, i.e. the position of the lowest usable coding rate for the specified bearer type.

5.4.11.7 Coding-rates-offset

The value in the *coding-rates-offset* fields defines the number of columns to the next coding rate for the specified bearer. A value of zero indicates that no further coding rates are specified. The effective ControlIndex for a given coding rate for the specified bearer (i.e. the lowest-coding-rate-index plus the sum of the coding-rates-offset up to the given coding rate) may be in the range 0 to 79.

5.4.11.8 Coding-rates-usage-bitmap

The *coding-rates-usage-bitmap* defines which coding rates may be used by the UEs. The leftmost bit in this bitmap refers to the coding rate in the leftmost column in the bearer table (i.e. the one indexed by *lowest-coding-rate-index*), etc.

This bitmap is 12 bits in length (although only 11 bits are required) in order to align to a half-octet boundary. Therefore the 12th bit in the bitmap is always zero.

5.4.11.9 Example BearerTables SDU

An example for the creation of a BearerTables SDU is explained in this clause. Figure 5.69 shows a selection from the Return Link Bearer Tables.

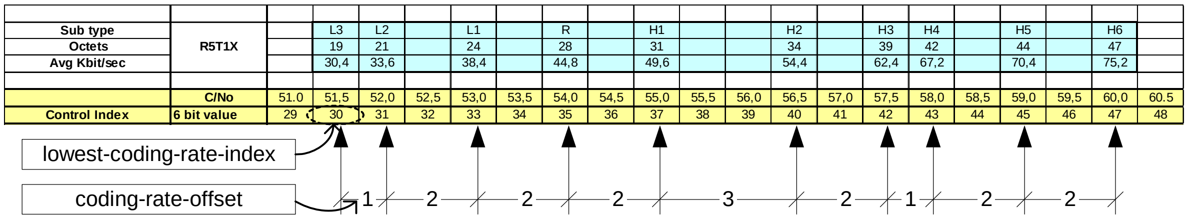


Figure 5.69: Selection from Bearer Table illustrating BearerTables SDU

The leftmost column (*lowest-coding-rate-index* = 30) contains the L3 bearer coding rate subtype. Moving from left to right, the next bearer subtype (L2) is located in the column marked by the ControlIndex value 31, hence the first *coding-rates-offset* value is 1. Subtype L1 is indexed by a ControlIndex value of 33, so the second *coding-rates-offset* value is 2, etc. This will result in a *coding-rate-offsets-list* of 1 2 2 2 3 2 1 2 2. Since this list is of a fixed length of ten *coding-rate-offset* items, a '0' has to be appended to indicate the end of the list.

Finally, the *coding-rates-usage-bitmap* needs to be included. Assuming, for example, that coding rates L3, L2 and H6 are to be excluded, then the bitmap would be 0 0 1 1 1 1 1 1 1 0. Another two bits set to zero need to be appended to fill the entire *coding-rates-usage-bitmap,* which has a size of 12 bits.

The resulting SDU structure for the above example is then as shown in Figure 5.70:

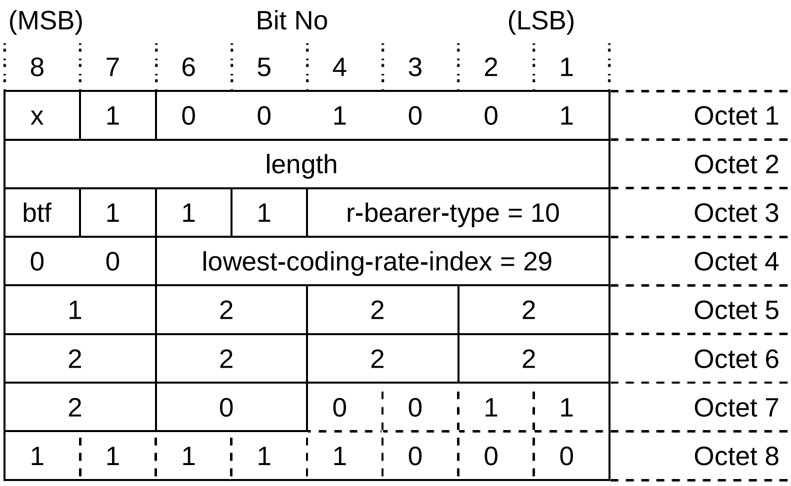


Figure 5.70: Example BearerTables BCtSDU

5.4.12 GPSEphemeris

5.4.12.0 General

The GPSEphemeris SDU is transferred in a Broadcast BCtPDU and is used to transmit GPS Ephemeris data to UEs, to speed up the acquisition time of their GPS chips, from Cold to Hot acquisition.

As the amount of data to be transferred may exceed the capacity of the bearer type suitable for transmission on the Global Beam, the data is split into two separate SDUs, which are distinguished by means of the *gps-section* field (also see [5], clause 2.4.3, Table 2-2 and clause 2.4.4, Tables 2-4 and 2-5).

The SDU payload has the following structure:

```
GPSEphemeris ::=
    SEQUENCE {
        iodc
            INTEGER (0..1023),
        gps-section
            INTEGER (0..3),
```

```
        reserved
            BIT STRING (SIZE(4)),
        prn-code
            INTEGER(0..255),
        gps-data
            CHOICE{
                -- as appropriate to the value of gps-section
                gps-data-first-section
                    GPSDataFirstSection,    -- if gps-section = 0
                gps-data-second-section
                    GPSDataSecondSection    -- if gps-section = 1
            }
    }
```

5.4.12.1 Iodc

The *iodc* (issue of data and clock) field is provided as specified in [5], clause 2.4.3 Table 2-2, to specify when the whole set of ephemeris data is being updated.

5.4.12.2 PRN-code

The 8-bit *prn-code* structure is as defined in [5], Table 2-1 (although the Signal Specification only specifies prn-codes in the range 1 to 32; higher values are for expansion by WAAS or EGNOS).

5.4.12.3 GPSDataFirstSection

5.4.12.3.0 General

The structure GPSDataFirstSection is defined as follows:

```
GPSDataFirstSection ::=
    SEQUENCE {
        week-no
            INTEGER (0..1023),
        satellite-accuracy
            INTEGER (0..15),
        satellite-health
            BIT STRING (SIZE(6)),
        reserved1
            BIT STRING (SIZE(4)),
        t-gd
            INTEGER (-128..127),
        t-oc
            INTEGER (0..65535),
        reserved2
            BIT STRING (SIZE(2)),
        af-0
            INTEGER (-2097152..2097151),
        af-1
            INTEGER (-32768..32767),
        af-2
            INTEGER (-128..127),
        c-rs
            INTEGER (-32768..32767),
        delta-n
            INTEGER (-32768..32767),
        m-0
            INTEGER (-2147483648..2147483647),
        c-uc
            INTEGER (-32768..32767),
        eccentricity
            INTEGER (0..4294967295),
        c-us
            INTEGER (-32768..32767)
    }
```

All negative integers are coded in two's complement arithmetic.

The resulting SDU structure is shown in Figure 5.71.

8 (MSB)	7	6	5	4	3	2	1 (LSB)	
x	1	0	0	1	0	1	0	Octet 1
length								Octet 2
< iodc								Octet 3
>		0	0	reserved				Octet 4
Prn-code								Octet 5
< week-no								Octet 6
>		satellite-accuracy				<		Octet 7
health >				reserved1				Octet 8
t-gd								Octet 9
t-oc								Octet 10
								Octet 11
reserved2								Octet 12
af-0								Octet 13
								Octet 14
af-1								Octet 15
								Octet 16
af-2								Octet 17
c-rs								Octet 18
								Octet 19
delta-n								Octet 20
								Octet 21
								Octet 22
m-0								
								Octet 25
c-uc								Octet 26
								Octet 27
								Octet 28
eccentricity								
								Octet 31
c-us								Octet 32
								Octet 33

Figure 5.71: GPSEphemeris Bearer Control SDU (containing GPSDataFirstSection)

5.4.12.3.1 Week-no

The *week-no* field is as defined in [5], clause 2.4.3 Table 2-2, representing the current GPS week at the start of the data transmission.

5.4.12.3.2 Satellite-accuracy

The *satellite-accuracy* structure is as defined in [5], clause 2.4.3 Table 2-2, being a 4-bit number which provides a look-up to the User Range Accuracy, whose meaning is defined in [5] clause 2.4.3.2.

5.4.12.3.3 Satellite-health

The *satellite-health* structure is as defined in [5], clause 2.4.3 Table 2-2 and clause 2.4.5.3, being a set of six BOOLEAN flags.

5.4.12.3.4 T-gd

The *t-gd* field is a satellite group delay correction term, in units of 2^{-31} seconds as defined for T_{gd} in [5], clause 2.4.3 Table 2-2 and clause 2.4.3.5.

5.4.12.3.5 T-oc

The *t-oc* field is the satellite clock correction parameter T_{oc} as defined in [5], clause 2.4.3 Table 2-2, in units of 2^4 seconds.

5.4.12.3.6 Af-2

The *Af-2* field is the satellite clock correction parameter A_{f2} as defined in [5], clause 2.4.3 Table 2-2, in units of 2^{-55} sec/sec^2.

5.4.12.3.7 Af-1

The *Af-1* field is the satellite clock correction parameter A_{f1} as defined in [5], clause 2.4.3 Table 2-2, in units of 2^{-43} sec/sec.

5.4.12.3.8 Af-0

The *Af-0* field is the satellite clock correction parameter A_{f0} as defined in [5], clause 2.4.3 Table 2-2, in units of 2^{-31} seconds.

5.4.12.3.9 C-rs

The *C-rs* field is the amplitude of the sine harmonic correction term to the orbit radius satellite clock correction parameter C_{rs} as defined in [5], clause 2.4.3 Tables 2-4 and 2-5, in units of 2^{-5} meters.

5.4.12.3.10 Delta-n

The *delta-n* field is the mean motion difference from computed value sparameter Δn as defined in [5], clause 2.4.3 Table 2-5, in units of 2^{-43} semicircles/sec.

5.4.12.3.11 M-0

The *m-0* field is the mean anomaly at reference time parameter M_0 as defined in [5], clause 2.4.3 Table 2-5, in units of 2^{-31} semicircles.

5.4.12.3.12 C-uc

The *C-uc* field is the amplitude of the cosine correction term to the argument of latitude parameter C_{uc} as defined in [5], clause 2.4.3 Table 2-5, in units of 2^{-29} radians.

5.4.12.3.13 Eccentricity

The *Eccentricity* field is the eccentricity parameter e as defined in [5], clause 2.4.3 Table 2-5, unitless, scale factor 2^{-33}.

5.4.12.3.14 C-us

The *C-us* field is the amplitude of the sine harmonic correction term to the argument of latitude parameter C_{us} as defined in [5], clause 2.4.3 Table 2-5, in units of 2^{-29} radians.

5.4.12.4 GPSDataSecondSection

5.4.12.4.0 General

The structure GPSDatasecondSection is defined as follows:

```
GPSDataSecondSection ::=
    SEQUENCE {
        a-half
            INTEGER (0..4294967295),
        t-oe
            INTEGER (0..65535),
        c-ic
            INTEGER (-32768..32767),
        omega-0
            INTEGER (-2147483648..2147483647),
        c-is
            INTEGER (-32768..32767),
        i-0
            INTEGER (-2147483648..2147483647),
        c-rc
            INTEGER (-32768..32767),
        omega
            INTEGER (-2147483648..2147483647),
        omega-dot
            INTEGER (-8388608..8388607),
        i-dot
            INTEGER (-8192.. 8191),
        reserved1
            BIT STRING (SIZE(2))
    }
```

All negative integers are coded in two's complement arithmetic.

The resulting SDU structure is shown in Figure 5.72.

(MSB)			Bit No				(LSB)	
8	7	6	5	4	3	2	1	
x	1	0	0	1	0	1	0	Octet 1
length								Octet 2
< iodc								Octet 3
>		0	1	reserved1				Octet 4
Prn-code								Octet 5
a-half								Octet 6
								Octet 9
t-oe								Octet 10
								Octet 11
c-ic								Octet 12
								Octet 13
omega-0								Octet 14
								Octet 17
c-is								Octet 18
								Octet 19
i-0								Octet 20
								Octet 23
c-rc								Octet 24
								Octet 25
omega								Octet 26
								Octet 29
omega-dot								Octet 30
								Octet 31
								Octet 32
< i-dot								Octet 33
>						reserved2		Octet 34

Figure 5.72: GPSEphemeris Bearer Control SDU (containing GPSDataSecondSection)

5.4.12.4.1 A-half

The *A-half* field is the square root of the semi-major axis parameter $A^{1/2}$ as defined in [5], clause 2.4.3 Table 2-5, in units of 2^{-19} meters$^{1/2}$.

5.4.12.4.2 T-oe

The *T-oe* field is the reference time ephemeris parameter T_{oe} as defined in [5], clause 2.4.3 Table 2-5, in units of 2^4 seconds.

5.4.12.4.3 C-ic

The *C-ic* field is the amplitude of the cosine harmonic term to the angle of inclination parameter C_{ic} as defined in [5], clause 2.4.3 Table 2-5, in units of 2^{-29} radians.

5.4.12.4.4 Omega-0

The *omega-0* field is the longitude of ascending node of orbit plane at weekly epoch parameter $OMEGA_0$ as defined in [5], clause 2.4.3 Table 2-5, in units of 2^{-31} semicircles.

5.4.12.4.5 C-is

The *C-is* field is the amplitude of the sine harmonic term to the angle of inclination parameter C_{is} as defined in [5], clause 2.4.3 Table 2-5, in units of 2^{-29} radians.

5.4.12.4.6 I-0

The *i-0* field is the inclination angle at reference parameter i_0 as defined in [5], clause 2.4.3 Table 2-5, in units of 2^{-31} semicircles.

5.4.12.4.7 C-rc

The *C-rc* field is the amplitude of the cosine harmonic correction term to the orbit radiussatellite clock correction parameter C_{rc} as defined in [5], clause 2.4.3 Table 2-4 and Table 2-5, in units of 2^{-5} meters.

5.4.12.4.8 omega

The *omega* field is the argument of perigee parameter ω as defined in [5], clause 2.4.3 Table 2-5, in units of 2^{-31} semicircles.

5.4.12.4.9 Omega-dot

The *omega-dot* field is the rate of right ascension parameter OMEGADOT as defined in [5], clause 2.4.3 Table 2-5, in units of 2^{-43} semicircles/sec.

5.4.12.4.10 I-dot

The *i-dot* field is the rate of inclination angle parameter IDOT as defined in [5], clause 2.4.3 Table 2-5, in units of 2^{-43} semicircles/sec.

5.4.13 SystemInfoIndex

The SystemInfoIndex SDU is a dynamic catalogue of all System Information elements that the RNC is broadcasting on the current forward bearer. This SDU is transferred in a Broadcast BCtPDU.

The SDU payload has the following structure, with format as shown in Figure 5.73:

```
SystemInfoIndex ::=
    SEQUENCE {
        index-parts-present
            BIT STRING {
                index-part-one-present (0),
                index-part-two-present (1),
                index-part-three-present (2),
                index-part-four-present (3),
                index-part-five-present (4),
                index-part-six-present (5),
                index-part-seven-present (6),
                reserved (7)
            },
        index-parts
            SEQUENCE SIZE (1..7) OF SEQUENCE {
```

```
                index-part-elements-present
                    CHOICE {
                        index-part-one-elements-present
                            BitmapPartOne,
                        index-part-two-elements-present
                            BitmapPartTwo,
                        index-part-three-elements-present
                            BitmapPartThree,
                        index-part-four-elements-present
                            BitmapPartFour,
                        index-part-five-elements-present
                            EmptyBitmap,
                        index-part-six-elements-present
                            EmptyBitmap,
                        index-part-seven-elements-present
                            EmptyBitmap
                    },
                index-elements
                    SEQUENCE SIZE (1..8) OF SystemInfoElement
            }
    }

BitmapPartOne ::=
    BIT STRING {
        primary-bearer-present (0),
        spot-beam-map-present (1),
        gps-ephemeris-present (2),
        plmn-info-present (3),
        satellite-location-present (4),
        satellite-state-vectors-present (5),
        utc-date-and-time-present (6),
        timing-correction-update-interval-present (7)
    } (SIZE (8))

BitmapPartTwo ::=
    BIT STRING {
        common-signalling-retry-present (0),
        max-delay-and-delay-range-present (1),
        randomising-control-present (2),
        access-control-present (3),
        initial-random-access-burst-present (4),
        gps-policy-info-param-present (5),
        nas-system-information-present (6),
        return-bearer-type-present (7)
    } (SIZE (8))

BitmapPartThree ::=
    BIT STRING {
        beam-info-present (0),
        subband-centre-frequency-offsets-present (1),
        forward-bearer-table-present (2),
        bearer-tables-present (3),
        randomised-initial-access-delay-present (4),
        intermod-test-info-present (5),
        reserved1 (6),
        -- placeholder for priority-spot-beam-map-present (6)
        reserved0 (7)
        -- placeholder for priority-primary-bearer-present (7)
    } (SIZE (8))

BitmapPartFour ::=
    BIT STRING {
        subband-cf-offset-change-present (0),
        initial-reference-level-present (1),
        leap-second-present (2),
        reserved4 (3),
        reserved3 (4),
        reserved2 (5),
        reserved1 (6),
        reserved0 (7)
    } (SIZE (8))

EmptyBitmap ::=
    BIT STRING {
        reserved7 (0),
        reserved6 (1),
        reserved5 (2),
        reserved4 (3),
```

```
        reserved3 (4),
        reserved2 (5),
        reserved1 (6),
        reserved0 (7)
    } (SIZE (8))
```

(MSB)			Bit No				(LSB)	
8	7	6	5	4	3	2	1	
x	1	0	0	1	1	0	0	Octet 1
length								Octet 2
index-parts-present								Octet 3
index-part-elements-present								Octet 4
< index-part-elements >								Octet n
index-part-elements-present								Octet n+1
< index-part-elements >								Octet n+2 … Octet k

Figure 5.73: System Information Index SDU

The SystemInfoIndex is a sequence of one to seven *index-parts*. The *index-parts-present* bitmap indicates which *index-parts* are present in the sequence. The order of *index-parts* in the sequence is implied by the position of the corresponding bit in the *index-parts-present* bitmap, starting from the most significant bit position (index part seven) to the least significant bit position (index part one).

Each *index-part* contains an *index-part-elements-present* bitmap followed by a sequence of one to eight *index-elements*. The *index-part-elements-present* bitmap indicates which *index-elements* are present in the sequence. The order of *index-elements* in the sequence is implied by the position of the corresponding bit in the *index part-elements-present* bitmap, starting from the most significant bit position to the least significant bit position.

Since a version number does not apply to either of the UTCDateAndTime and SatelliteStateVectors AVPs and since there is always one instance of each of these AVPs, their presence will be indicated in the *index-part-one-elements-present* bitmap when they are being transmitted, but there will not be a corresponding *index-element* in the sequence for index part one.

Each *index-element* is of type SystemInfoElement and indicates the version and number of instances of each element of broadcast System Information (AVP or SDU). In all cases, a version number of 0 means "always receive", therefore this special instance of version number shall not be included in the range for wraparound of version numbers used to check the "uniqueness period". See ETSI TS 102 744-3-2 [10] for further information.

```
SystemInfoElement ::=
    CHOICE {
        primary-bearer                          Type1IndexElement,
        spot-beam-map                           Type3IndexElement,
        gps-ephemeris                           Type1IndexElement,
        plmn-info                               Type2IndexElement,
        satellite-location                      Type2IndexElement,
        timing-correction-update-interval       Type3IndexElement,
        common-signalling-retry                 Type3IndexElement,
        max-delay-and-delay-range               Type3IndexElement,
        randomising-control                     Type3IndexElement,
        access-control                          Type3IndexElement,
        initial-random-access-burst             Type3IndexElement,
        gps-policy-info-param                   Type3IndexElement,
        nas-system-information                  Type3IndexElement,
        return-bearer-type                      Type3IndexElement,
        subband-centre-frequency-offsets        Type3IndexElement,
```

```
        forward-bearer-table                    Type2IndexElement,
        bearer-tables                           Type1IndexElement,
        beam-info                               Type3IndexElement,
        randomised-initial-access-delay         Type3IndexElement,
        intermod-test-info                      Type3IndexElement,
        subband-centre-frequency-offset-change  Type3IndexElement,
        initial-reference-level                 Type2IndexElement,
        leap-second                             Type3IndexElement
    }

Type1IndexElement ::=
    SEQUENCE {
        version
            INTEGER (0..255),
            -- value '0' denotes 'always receive'
        instances
            INTEGER (0..255)
    }

Type2IndexElement ::=
    SEQUENCE {
        version
            INTEGER (0..15),
            -- value '0' denotes 'always receive'
        instances
            INTEGER (0..15)
    }

Type3IndexElement ::=
    SEQUENCE {
        version
            INTEGER (0..255)
            -- value '0' denotes 'always receive'
            -- one instance implied except in case of SpotBeamMap
            -- where number of instances is advertised within the
            -- corresponding SpotBeamMap SDU
    }
```

5.4.14 BearerTableUpdate

5.4.14.0 General

The BearerTableUpdate SDU may be used by the RNC to signal any changes in the properties of the physical layer return bearer types and subtypes in a connection specific BCtPDU. The implementation of this SDU is mandatory for UE Classes 6 to 15 of RI-Version 0x82 and for all UE Classes from RI-Version 0x83 onwards (see ETSI TS 102 744-3-5 [13], clause 6.1.2.2). The SDU has the following structure, with format as shown in Figure 5.74:

```
BearerTableUpdate ::=
    SEQUENCE {
        reserved
            BIT STRING (SIZE (4)),
        bearer-table-version
            BearerTableVersion,
        bearer-table-definitions
            SEQUENCE SIZE (1..16) OF ExtBearerDefinition
    }
```

The data type BearerTableVersion is specified as follows:

```
BearerTableVersion ::=
    INTEGER (0..15) -- value '0' is reserved
```

The data type ExtBearerDefinition is specified as follows:

```
ExtBearerDefinition ::=
    SEQUENCE {
        bearer-table-follows
            BOOLEAN,
        lowest-coding-rate-index-included
            BOOLEAN,
        coding-rates-offsets-included
            BOOLEAN,
        coding-rates-usage-included
            BOOLEAN,
```

```
    r-bearer-type
        ReturnBearerTypeFull,
    lowest-coding-rate-index-octet
        SEQUENCE {
            r-bearer-x
                CHOICE {
                -- depending on the value of r-bearer-type
                    -- if r-bearer-type = 8 or 9
                    modulation-rate
                        ModulationRate,
                    -- else (any other r-bearer-type)
                    reserved2
                        BIT STRING (SIZE (2))
                },
            lowest-coding-rate-index
                ControlIndex
        } OPTIONAL,
    coding-rates-offsets-list
        CHOICE {
        -- depending on the value of r-bearer-type
            normal-bearer-coding-rate-offsets
            -- if r-bearer-type <>  1 or 2
                SEQUENCE {
                    coding-rates-offset
                        SEQUENCE SIZE (10) OF INTEGER (0..3)
                },
            ldr--bearer-coding-rate-offsets
            -- if r-bearer-type =  1 or 2
                SEQUENCE {
                    coding-rates-offset
                        SEQUENCE SIZE (16) OF INTEGER (0..3)
                }
        } OPTIONAL,
    coding-rates-usage-bitmap
        CHOICE {
            normal-bearer-usage-map
                    BIT STRING (SIZE (12)),
            ldr-bearer-usage-map
                    BIT STRING (SIZE (16))
        } OPTIONAL,
    reserved4
        BIT STRING (SIZE (4)) OPTIONAL
        -- padding to start next bearer definition
        -- on octet boundary
}
```

(MSB) 8	7	6	5 Bit No	4	3	2	(LSB) 1	
x	1	0	0	1	1	0	1	Octet 1
Length								Octet 2
reserved				bearer-table-version				Octet 3
btf	lcrii = 1	croi = 1	crui = 1	r-bearer-type				Octet 4
r-bearer-x		lowest-coding-rate-index						Octet 5
offset 1		offset 2		offset 3		offset 4		Octet 6
offset 5		offset 6		offset 7		offset 8		Octet 7
offset 9		offset 10		<				Octet 8
coding-rates-usage-bitmap >								Octet 9
bearer-definition-list								
(continued)								Octet n

Figure 5.74: BearerTableUpdate Bearer Control SDU (all elements present)

An example showing the usage for the description of an LDR bearer (r-bearer-type = 8 or 9) is shown below.

(MSB) 8	7	6	5 Bit No	4	3	2	(LSB) 1	
x	1	0	0	1	1	0	1	Octet 1
length								Octet 2
reserved				bearer-table-version				Octet 3
btf	lcrii = 1	croi = 1	crui = 1	r-bearer-type				Octet 4
r-bearer-x		lowest-coding-rate-index						Octet 5
offset 1		offset 2		offset 3		offset 4		Octet 6
offset 5		offset 6		offset 7		offset 8		Octet 7
offset 9		offset 10		offset 11		offset 12		Octet 8
offset 13		offset 14		offset 15		offset 16		Octet 9
bearer-definition-list								
(continued)								Octet n

Figure 5.75: BearerTablesUpdate SDU (*coding-rates-usage-bitmap absent*), RBearerType = R80T0.5Q or R80T1Q

The BearerTableUpdate SDU is almost identical in format to the BearerTables SDU. It has two additional fields, *bearer-table-version* and *bearer-x*. The BearerTableUpdate SDU can be used to describe a full or partial set of Return Bearer Types and Subtypes in relation to each other.

The UE shall store the current Bearer Table and the associated version number in non-volatile programmable memory. At registration time, the UE reports the Bearer Table version to the RNC. If the version stored in the UE is not the most recent (i.e. not the version stored in the RNC), then the RNC will send a BearerTableUpdate SDU to the UE. The BearerTableUpdate SDU includes only the bearer information that is required to update the UE BearerTable to the most recent version (i.e. differences between the versions in the UE and RNC).

The BearerTableUpdate SDU cannot be broadcast because bearer tables are dependent upon UE class. A detailed description of the application of this SDU can be found in ETSI TS 102 744-3-2 [10].

5.4.14.1 Bearer-table-version

This field indicates the version of the current Bearer Table in the RNC. The information in the BearerTableUpdate SDU will update the Bearer Table stored in the UE to this version. The value '0' is reserved.

5.4.14.2 Modrate

This field is used to carry modulation information for the R80T25X and R80T5X bearer types as follows:

```
ModulationRate ::=
    INTEGER {
        x4-modulation (0),
        x16-modulation (1),
        x32-modulation (2),
        x64-modulation (3)
    } (0..3)
```

This variable is only defined when the r-bearer-type field contains either value 8 or 9 corresponding to the R80T2.5X and R80T5X bearers.

5.5 Embedded Bearer Control SDU (BCtESDU)

5.5.0 General

Embedded Bearer Control SDUs are also used for the peer-to-peer communication between Bearer Control Processes at the UE and the RNC, however, they are embedded in an Embedded BCtPDU (see clause 5.2) within UE Specific Signalling Protocol Data Units (ALSigPDUs). The Bearer Control Embedded Signalling Data Units that are defined in the to-UE direction are shown in Table 5.8.

Table 5.8: Bearer Control Embedded SDU Types (To-UE only)

ESType/ ESDUType	Interpretation
0x00	Reserved
0x01	ConnectionAssociation
0x02	ConnectionReassociation
0x03	GroupConnectionAssociation
0x04	GroupConnectionReassociation
0x05	HardwareAVPList
0x06	Reserved
0x07	AVPList
0x08	SpotBeamMap
0x09..0x0C	Reserved
0x0D	BearerTableUpdate
0x0E..0x3F	Reserved

5.5.1 Embedded Bearer Control SDU Structure

The Embedded Bearer Control SDU Structure is identical to the BCtSDU structure (see clause 5.3), except that *sdu-type* and *s-type* are replaced by *esdu-type* and *es-type* respectively.

```
BCtESDU ::=
    SEQUENCE {
        bct-esdu-follows
            BOOLEAN,
        type-and-length
            CHOICE {
```

```
            short
                SEQUENCE {
                    extended-length
                        BOOLEAN, -- FALSE
                    es-type
                        ShortBCtESDUType,
                    es-length
                        INTEGER (1..8)
                },
            long
                SEQUENCE {
                    extended-length
                        BOOLEAN, -- TRUE
                    esdu-type
                        BCtESDUType,
                    length
                        INTEGER (1..256)
                }
        }
    esdu-payload
        CHOICE {
        -- as appropriate to the value of
        -- es-type or esdu-type
            conn-association
                ConnectionAssociation,
            conn-reassociation
                ConnectionReassociation,
            avp-list
                AVPList,
            hardware-avp-list
                HardwareAVPList,
            spot-beam-map
                SpotBeamMap,
            bearer-table-update
                BearerTableUpdate
        }
}
```

The body of the Bearer Control Embedded SDU is dependent on the value of *es-type* / *esdu-type*, with formats as shown in Figures 5.76 and 5.77.

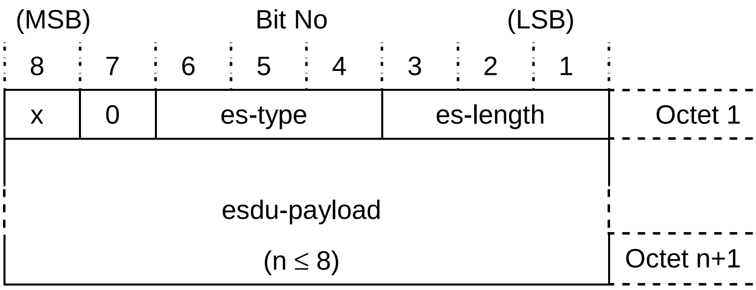


Figure 5.76: Bearer Control Embedded SDU Structure (Short Length)

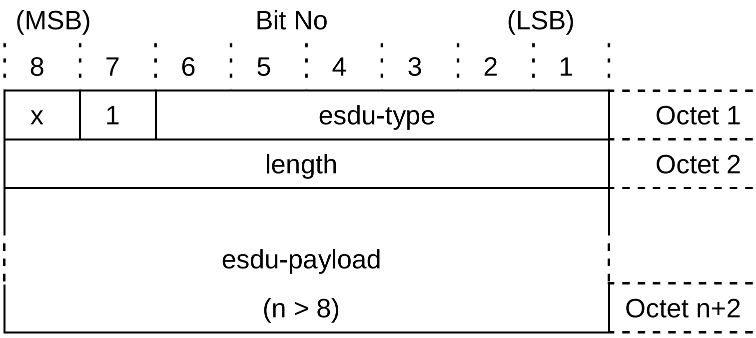


Figure 5.77: Bearer Control Embedded SDU Structure (Extended Length)

5.5.2 BCtESDU Parameters

5.5.2.1 Bct-esdu-follows

This is a single-bit flag which, when set to '1' indicates that another BCtESDU follows this BCtESDU, else it indicates that this BCtESDU is the last one in the *bct-esdu-list*.

5.5.2.2 Extended-length

This is a single-bit flag which, when set to '1' indicates that an extended type and length field are utilized within the BCt-ESDU header.

5.5.2.3 Es-length/length

The *es-length* and *length* fields are a 3-bit and 8-bit value respectively, both of which are used to indicate the length of the ESDU. The value of this field represents the number of octets in the *esdu-payload*. The encoding of these parameters, which are defined as INTEGER(1..8) and INTEGER (1..256) respectively, follows the ASN.1 Packed Encoding Rules (PER - see clause 4.4.4), hence zero length payloads are not supported.

5.5.2.4 BCtESDUType and ShortBCtESDUType

The *es-type*/*esdu-type* parameter defines the Bearer Control Embedded Signalling Data Unit type. Values in the range 0x00..0x07 shall be encoded as a ShortBCtESDUType, in a 3-bit field (i.e. the BCtESDU *type-and-length* field is of CHOICE *short)*, unless the length of the ESDU exceeds 8 bytes, in which case they shall be encoded as a BCtESDUType, a 6-bit field. In this case, the BCtESDU *type-and-length* field is of CHOICE *long.* Different coding for this field may be used in the From-UE and To-UE directions, however, at present ESDUs are sent in the To-UE direction only.

```
BCtESDUType ::=
    INTEGER {
        conn-association (1),
        conn-reassociation (2),
        hardware-avp-list (5),
        avp-list (7),
        spot-beam-map (8),
        bearer-table-update (13)
    } (0..63)

ShortBCtESDUType ::=
    INTEGER {
        conn-association (1),
        conn-reassociation (2),
        hardware-avp-list (5),
        avp-list (7)    } (0..7)
```

5.6 Bearer Control ESDU Payloads

5.6.0 General

NOTE: Throughout clauses 5.6.1 to 5.6.8, the ASN.1 defines components of ESDUPayload, while the figures illustrate the entire BCtESDU structure (i.e. including the fields *bct-esdu-follows* and *type-and-length*).

5.6.1 ConnectionAssociation

5.6.1.0 General

The ConnectionAssociation ESDU is transferred in an Embedded BCtPDU attached to Adaptation Layer signalling (Register-Ack or Establish PDUs) if there is a requirement to establish the relationship between a Bearer Connection ID and a translated Bearer Connection ID for each hardware element within the UE the connection is to operate over. The ConnectionAssociation ESDU is not present for connections that are addressed directly by their Bearer Connection ID rather than a translated Bearer Connection ID. The ESDU Payload has the following structure:

```
ConnectionAssociation::=
    SEQUENCE OF HardwareAndTBCnId
```

The ConnectionAssociation BCtESDU may also be included in a Modify ALSigPDU. The format is as shown in Figures 5.78 and 5.79.

(MSB) 8	7	6	Bit No 5	4	3	2	(LSB) 1	
x	0	0	0	1	0	0	1	Octet 1
direction		hw-id		<				Octet 2
tbcn-id							>	Octet 3

Figure 5.78: ConnectionAssociation Bearer Control Embedded SDU (single UE hardware element)

(MSB) 8	7	6	Bit No 5	4	3	2	(LSB) 1	
x	0	0	0	1	length			Octet 1
direction 1		hw-id 1		<				Octet 2
tbcn-id 1							>	Octet 3
direction 2		hw-id 2		<				Octet 4
tbcn-id 2							>	Octet 5
HardwareAndTBCnId-List (continued)								Octet n ≤ 9

Figure 5.79: ConnectionAssociation Bearer Control Embedded SDU (multiple UE hardware elements)

If a connection is distributed over more than four UE hardware elements (extremely unlikely), then a long BCtESDU header needs to be used.

5.6.1.1 HardwareAndTBCnId

This is used to associate a specific *tbnc-id* with a specific *hardware-id*.

```
HardwareAndTBCnId ::=
    SEQUENCE {
        hardware-id
            UEHardwareID,
        tbcn-id
            TranslatedBearerConnectionID
    }
```

5.6.1.2 UEHardwareID

The UEHardwareId is used to identify specific transceiver hardware within the UE. This information element also includes a direction identifier that is used to control specific aspects of the UE hardware, or to reference unidirectional connections that are to be supported on this hardware.

```
UEHardwareID ::=
    SEQUENCE {
        direction
            Direction,
        hw-id
            HardwareId
    }
```

For a UE which only supports a single transceiver pair, the value of '1' shall be associated with this hardware and used in the HardwareId information element. To reference the hardware without specifying a direction, the value of 0 shall be used for the Direction information element.

5.6.1.3 Direction

The Direction is used to identify a specific direction for which the tBCnId is applicable.

```
Direction ::= INTEGER {
    both-directions (0),
    downlink-only (1),
    uplink-only (2),
    reserved (3)
} (0..3)
```

5.6.1.4 HardwareID

The HardwareId is used to identify specific transceiver hardware within the UE.

```
HardwareID ::=
    INTEGER (0..3)
```

The value of zero is reserved. For a UE which only supports a single transceiver pair, the value of '1' shall be associated with this hardware.

5.6.2 ConnectionReassociation

The ConnectionReassociation ESDU is transferred in an Embedded BCtPDU attached to Adaptation Layer signalling, (Modify or Handover ALSigPDUs) and is used to modify the relationship between a Bearer Connection ID and a translated Bearer Connection ID for each hardware element within the UE that the connection is to operate over. The Embedded SDU Payload has the following structure:

```
ConnectionReassociation ::=
    SEQUENCE {
        bearer-connection-id
            BCnID,
        connection-list
            SEQUENCE OF HardwareAndTBCnId OPTIONAL
    }
```

The ConnectionReassociation ESDU is included in a Handover ALSigPDU and may also be included in a Modify ALSigPDU if a handover process is occurring simultaneously. The format is shown in Figure 5.80.

(MSB) 8	7	6	Bit No 5	4	3	2	(LSB) 1	
x	0	0	1	0	1	0	0	Octet 1
bearer-connection-id								Octet 2
								Octet 3
								Octet 4
direction		hw-id		<				Octet 5
tbcn-id							>	Octet 6

Figure 5.80: ConnectionReassociation Bearer Control Embedded SDU (single UE hardware element)

If more than two hardware elements are associated with the connection (unlikely), then a long BCtESDU header is required as illustrated in Figure 5.81.

(MSB) 8	7	6	5	Bit No 4	3	2	(LSB) 1	
x	1	0	0	0	0	1	0	Octet 1
length								Octet 2
bearer-connection-id								Octet 3
								Octet 4
								Octet 5
direction 1		hw-id 1		< tbcn-id 1				Octet 6
tbcn-id 1 >								Octet 7
direction 2		hw-id 2		< tbcn-id 2				Octet 8
tbcn-id 2 >								Octet 9
HardwareAndTBCnId-List								
								Octet n

Figure 5.81: ConnectionReassociation Bearer Control Embedded SDU (multiple UE hardware elements)

HardwareAndTBCnId is defined in clause 5.6.1.1.

The ConnectionReassociation ESDU may be transmitted without a tBCnId, as shown in Figure 5.82. In this case the tBCnID shall be no longer associated with this connection, and the UE may only transfer data for this connection if using BCnID extended addressing mode. After the tBCnID has been removed, a new tBCnID may be configured by the RNC using a Handover or Modify ALSigPDU containing a ConnectionReassociation ESDU containing a SEQUENCE OF HardwareAndTBCnId. The Handover ALSigPDU may be used to modify the UESS addressing mode, and this mechanism will be used if the addressing mode is being changed while the UE is maintained on the same Bearer Control.

(MSB) 8	7	6	5	Bit No 4	3	2	(LSB) 1	
x	0	0	1	0	0	1	0	Octet 1
bearer-connection-id								Octet 2
								Octet 3
								Octet 4

Figure 5.82: ConnectionReassociation Bearer Control Embedded SDU (tBCnID being removed by RNC)

5.6.3 GroupConnectionAssociation

5.6.3.0 General

The GroupConnectionAssociation ESDU is used whenever a MBMS connection is either being associated or modified, and is transferred in an Embedded BCtPDU attached to Adaptation Layer signalling (Establish PDUs) to establish the relationship between a Bearer Connection ID and one or more translated Bearer Connection IDs for each hardware element within the UE the connection is to operate over. It is also used to transfer the optional COUNT_C value, as defined in clause 5.8.1.2 in ETSI TS 102 744-3-4 [12], to be used for ciphering of the MBMS Connection when such behaviour is required.

The ESDU Payload has the following structure:

```
GroupConnectionAssociation::=
    SEQUENCE {
        forward-tbcnid HardwareAndTBCnId,
        return-tbcnid HardwareAndTBCnId OPTIONAL,
        count-c CountC OPTIONAL
    }
```

The GroupConnectionAssociation BCtESDU will always be included in an Establish ALSigPDU when a MBMS connection is being established, and may be included in a Modify ALSigPDU. The formats are as shown in Figures 5.83 to 5.86.

(MSB)			Bit No				(LSB)	
8	7	6	5	4	3	2	1	
x	0	0	1	1	0	0	1	Octet 1
direction		hw-id		<				Octet 2
		forward-tbcn-id					>	Octet 3

Figure 5.83: GroupConnectionAssociation Bearer Control Embedded SDU (unidirectional downlink MBMS Connection)

(MSB)			Bit No				(LSB)	
8	7	6	5	4	3	2	1	
X	0	0	1	1	0	1	1	Octet 1
direction 1		hw-id		<				Octet 2
		forward-tbcn-id					>	Octet 3
direction 2		hw-id		<				Octet 4
		return-tbcn-id					>	Octet 5

Figure 5.84: GroupConnectionAssociation Bearer Control Embedded SDU (bidirectional MBMS Connection)

(MSB)			Bit No				(LSB)	
8	7	6	5	4	3	2	1	
x	0	0	1	1	1	0	1	Octet 1
direction		hw-id		<				Octet 2
		forward-tbcn-id					>	Octet 3
			count-c					
								Octet 7

Figure 5.85: ConnectionAssociation Bearer Control Embedded SDU (uni-directional downlink ciphered MBMS Connection)

(MSB)			Bit No				(LSB)	
8	7	6	5	4	3	2	1	
x	0	0	1	1	1	1	1	Octet 1
direction 1		hw-id		<				Octet 2
forward-tbcn-id							>	Octet 3
direction 2		hw-id		<				Octet 4
return-tbcn-id							>	Octet 5
count-c								
								Octet 9

Figure 5.86: ConnectionAssociation Bearer Control Embedded SDU (bi-directional ciphered MBMS Connection)

Only one hardware ID can be signalled with this SDU. If connections need to be setup on multiple hardware elements, then this SDU will need to be transmitted multiple times, once for each Connection: UEHardwareID association. In addition, a maximum of four (4) UEHardwareID values are supported.

5.6.3.1 Count-C

The Count-C value is an INTEGER that is present if MBMS Connections are ciphered, and is used in deciphering of the downlink portion of the MBMS Connection only.

5.6.4 GroupConnectionReassociation

The GroupConnectionReassociation ESDU is used when a mobile terminal already has an association with a MBMS connection, and this connection is being modified. The ESDU is transferred in an Embedded BCtPDU attached to Adaptation Layer signalling, (Modify or Handover ALSigPDUs) and is used to modify the relationship between a Bearer Connection ID and one or more translated Bearer Connection IDs for each hardware element within the UE that the connection is to operate over. The Embedded SDU Payload has the following structure:

```
GroupConnectionReassociation::=
    SEQUENCE {
        bearer-connection-id
        BCnID,
        forward-tbcnid HardwareAndTBCnId,
        return-tbcnid HardwareAndTBCnId OPTIONAL,
        count-c CountC OPTIONAL
    }
```

The GroupConnectionReassociation ESDU will be included in a Handover ALSigPDU for any bi-directional bearer connections with which the UE is associated, and may also be included in a Modify ALSigPDU if a modification and handover process are occurring simultaneously. The formats are as shown in Figures 5.87 to 5.90.

(MSB)			Bit No				(LSB)	
8	7	6	5	4	3	2	1	
x	0	1	0	0	1	0	0	Octet 1
bearer-connection-id								Octet 2
								Octet 3
								Octet 4
direction		hw-id		<				Octet 5
forward-tbcn-id							>	Octet 6

Figure 5.87: GroupConnectionReassociation Bearer Control Embedded SDU (unidirectional downlink MBMS Connection)

(MSB)			Bit No				(LSB)	
8	7	6	5	4	3	2	1	
X	0	1	0	0	1	1	0	Octet 1
Bearer-connection-id								Octet 2
								Octet 3
								Octet 4
direction 1		hw-id		<				Octet 5
forward-tbcn-id							>	Octet 6
direction 2		hw-id		<				Octet 7
return-tbcn-id							>	Octet 8

Figure 5.88: GroupConnectionReassociation Bearer Control Embedded SDU (bidirectional MBMS Connection)

(MSB)			Bit No				(LSB)	
8	7	6	5	4	3	2	1	
x	1	0	0	0	1	0	0	Octet 1
Length (=8)								Octet 2
bearer-connection-id								Octet 3
								Octet 4
								Octet 5
direction		hw-id		<				Octet 6
forward-tbcn-id							>	Octet 7
count-c								8
								9
								10
								Octet 11

Figure 5.89: GroupConnectionReassociation Bearer Control Embedded SDU (uni-directional downlink ciphered MBMS Connection)

8 (MSB)	7	6	5	4	3	2	1 (LSB)	
x	1	0	0	0	1	0	0	Octet 1
Length (=10)								Octet 2
bearer-connection-id								Octet 3
								Octet 4
								Octet 5
direction 1		hw-id		<				Octet 6
forward-tbcn-id							>	Octet 7
direction 2		hw-id		<				Octet 8
return-tbcn-id							>	Octet 9
count-c								10
								11
								12
								Octet 13

Figure 5.90: GroupConnectionReassociation Bearer Control Embedded SDU (bi-directional ciphered MBMS Connection)

Only one hardware ID can be signalled with this SDU. If connections need to be setup on multiple hardware elements, then this SDU will need to be transmitted multiple times, once for each Connection: UEHardwareID association. In addition, a maximum of Four UEHardwareID values are supported.

5.6.5 HardwareAVPList

The HardwareAVPList ESDU may be used within a BCtEPDU attached to an Adaptation Layer Signalling PDU if the AVPs within the *bct-avp-list* are directed at specific hardware within the UE. The ESDU Payload has the following structure, with format as shown in Figures 5.91 to 5.92:

```
HardwareAVPList ::=
    SEQUENCE {
        hardware-id
            UEHardwareID,
        reserved
            BIT STRING (SIZE (4)),
        bct-avp-list
            SEQUENCE OF BCtAVP
    }
```

8 (MSB)	7	6	5	4	3	2	1 (LSB)	
x	0	1	0	1	s-length			Octet 1
direction		hw-id		reserved				Octet 2
bct-avp-list								
								Octet n ≤ 9

Figure 5.91: HardwareAVPList Bearer Control Embedded SDU (short length)

(MSB)			Bit No				(LSB)	
8	7	6	5	4	3	2	1	
x	1	0	0	0	1	0	1	Octet 1
length								Octet 2
direction		hw-id		reserved				Octet 3
bct-avp-list								
								Octet n

Figure 5.92: HardwareAVPList Bearer Control Embedded SDU (extended length)

UEHardwareId is defined in clause 5.6.1.2.

5.6.6 AVPList

The structure of the AVPList ESDU is identical to that of the AVPList SDU, which is specified in clause 5.4.10. Used within a BCt-EPDU attached to an ALSigPDU, the AVPs in this ESDU are used to control the UE as a whole rather than specific hardware, or to provide information about the bearer control to which the UE is being tuned.

5.6.7 SpotBeamMap

The structure of the SpotBeamMap ESDU is identical to that of the SpotBeamMap SDU, which is specified in clause 5.4.11.

5.6.8 BearerTableUpdate

The structure of the BearerTableUpdate ESDU is identical to that of the BearerTableUpdate SDU, which is specified in clause 5.4.15.

5.7 Bearer Control AVPs

Bearer Control AVPs (BCtAVP) are used to transfer parameters between Bearer Control processes within the Bearer Control Layer. Table 5.9 lists the BCtAVPs which may be carried from the RNC in BCtSDUs, while Table 5.10 lists those BCtAVPs which are supported in BCtESDUs. Table 5.11 summarizes those BCtAVPs that are applicable in the from-UE direction.

Table 5.9: BCtAVPs supported in BCtSDUs (from RNC to UE)
(OP: optional presence, not applicable)

	Broadcast		**Specific**
AVP Type	***bb-avp-list* or *avp-list***	***specific-avp-list***	***avp-list***
TimingCorrection	-	OP	OP
ReturnBearerType	OP	-	-
FracPeakDataVolAndLfrac	OP	-	-
StatusAckControl	OP	-	-
ReportControl	OP	OP	-
MinContStatusDelay	OP	-	-
RandomisingControl	OP	OP	-
ResWaitMultiplier	OP	OP	-
CommonSigRetry	OP	OP	-
ControlledRandomAccess	OP	-	-
SharedReservationAccess	OP	-	-
ForwardBearerCodeRate	OP	-	-
SignalQualityMeasurementInterval	OP	OP	OP
AccessControl	OP	-	-
InitialRandomAccessBurst	OP	-	-
ReturnLinkReferenceLevel	-	OP	OP
InitialReferenceLevelAndMaxCodeRate	-	OP	OP

	Broadcast		Specific
AVP Type	***bb-avp-list* or *avp-list***	***specific-avp-list***	***avp-list***
InitialReferenceLevelSet	-	OP	OP
InitialReferenceLevelParam	OP		-
ReturnLinkReferenceLevelSet	-	OP	OP
MaxReturnCodeRate	-	OP	OP
TimingCorrectionUpdateInterval	OP		
MaxDelayAndDelayRange	OP	-	-
ForwardQoSControl	OP	OP	OP
SubbandCentreFrequencyOffsets	OP	-	-
SubbandCentreFrequencyOffsetChange	OP	-	-
SatelliteLocation	OP	-	-
ChannelNo	OP	OP	OP
BeamInfo	OP	OP	OP
PLMNInfo	OP	-	-
PrimaryBearer	OP	-	-
SatelliteStateVectors	OP	-	-
UTCDateAndTime	OP		
TMPayloadPositionParam	-	-	OP
NASSystemInfo	OP	-	-
GPSPolicyInfoParam	OP	-	-
RandomisedInitialAccessDelayParam	OP	-	-
IntermodTestInfoParam	OP	-	-
LeapSecondParam	OP	-	-

Table 5.10: BCtAVPs supported in BCtESDUs (from RNC to UE)
(OP: optional presence, not applicable)

	Embedded	
AVP Type	***avp-list***	***hardware-avp-list***
ReturnBearerType	-	OP
MinResWait	OP	-
AllocationSize	OP	-
SlotSharing	OP	-
FracPeakDataVolAndLfrac	OP	-
StatusAckControl	OP	-
ReportControl	OP	-
MinContStatusDelay	OP	-
RandomisingControl	OP	-
ResWaitMultiplier	OP	-
RNCId	OP	OP
Qdiff	OP	-
ChannelNo	OP	OP
BeamInfo	OP	OP
SlotSizeControl	OP	-
SleepMode	-	OP
Type0QoS	OP	-
ReturnLinkReferenceLevel	OP	OP
InitialReferenceLevelAndMaxCodeRate	OP	OP
InitialReferenceLevelSet	OP	OP
ReturnLinkReferenceLevelSet	OP	OP
MaxReturnCodeRate	OP	OP
MaxDelayAndDelayRange	OP	OP
ForwardCarrierLoss	OP	OP
SignalQualityMeasurementInterval	OP	OP
NASSystemInfo	OP	-
PrimaryBearer	OP	-
Access Control	OP	-
CommonSigRetry	OP	-

The relevance of each of the above AVPs on the UE behaviour is specified in ETSI TS 102 744-3-2 [10].

Table 5.11: BCtAVPs supported in BCtSDUs (from UE to RNC)

AVP Type	May be carried in
ReceivedSignalQuality	status-avp-list, avp-list
ReferenceLevelAcknowledge	status-avp-list, avp-list

5.7.1 BCtAVP Structure

5.7.1.0 General

The structure of a Bearer Control AVP is as follows:

```
BCtAVP ::=
    SEQUENCE {
        bct-avp-type    BCtAVPType,
        param-value
            CHOICE { -- as appropriate to the value of bct-avp-type
                access-control-param                AccessControlParam,
                allocation-size-param               AllocationSizeParam,
                beam-info-param                     BeamInfoParam,
                channel-no-param                    ChannelNoParam,
                controlled-random-access                ControlledRandomAccessParam,
                common-sig-retry-param              CommonSigRetryParam,
                forward-carrier-loss-param          ForwardCarrierLossParam,
                fwd-bearer-code-rate-param          ForwardBearerCodeRateParam,
                fwd-qos-control-param               ForwardQoSControlParam,
                frac-peak-data-vol-and-lfrac-param      FracPeakDataVolAndLFracParam,
                gps-policy-info-param               GPSPolicyInfoParam,
                initial-random-access-burst-param       InitialRandomAccessBurstParam,
                initial-reference-level-param           InitialReferenceLevelParam,
                initial-reference-level-and-max-code-rate-param
                        InitialReferenceLevelAndMaxCodeRateParam,
                initial-reference-level-set-param       InitialReferenceLevelSetParam,
                intermod-test-info-param                IntermodTestInfoParam,
                leap-second                         LeapSecondParam,
                max-return-code-rate-param          MaxReturnCodeRateParam,
                maxdelay-and-delayrange-param           MaxDelayAndDelayRangeParam,
                min-cont-status-delay-param         MinContStatusDelayParam,
                min-res-wait-param                  MinResWaitParam,
                nas-sys-info-param                  NASSystemInfoParam,
                plmn-info-param                     PLMNInfoParam,
                primary-bearer-param                    PrimaryBearerParam,
                q-diff-param                        QDiffParam,
                randomised-initial-access-delay-param
                        RandomisedInitialAccessDelayParam,
                randomising-control-param           RandomisingControlParam,
                received-signal-quality-param           ReceivedSignalQualityParam,
                reference-level-acknowledge         ReferenceLevelAcknowledgeParam,
                report-control-param                    ReportControlParam,
                res-wait-multiplier-param               ResWaitMultiplierParam,
                ret-bearer-type-param               ReturnBearerTypeParam,
                ret-link-reference-level-param          ReturnLinkReferenceLevelParam,
                ret-link-reference-level-set-param
                        ReturnLinkReferenceLevelSetParam,
                rnc-id-param                        RNCIdParam,
                satellite-location-param                SatelliteLocationParam,
                satellite-state-vectors-param           SatelliteStateVectorsParam,
                shared-reservation-access               SharedReservationAccessParam,
                signal-qual-meas-interval-param
                        SignalQualityMeasurementIntervalParam,
                sleep-mode-param                    SleepModeParam,
                slot-sharing-param                  SlotSharingParam,
                slot-size-control-param             SlotSizeControlParam,
                status-ack-control-param                StatusAckControlParam,
                subband-cf-offset-param             SubbandCentreFrequencyOffsetsParam,
                subband-cf-offset-change-param
                        SubbandCentreFrequencyOffsetChangeParam,
                timing-correction-param             TimingCorrectionParam,
                timing-corr-update-interval-param
                        TimingCorrectionUpdateIntervalParam,
```

```
            tm-pay-load-position-param          TMPayloadPositionParam,
            type-0-qos-param                    Type0QoSParam,
            utc-date-and-time                   UTCDateAndTimeParam
        }
    }
```

Each Parameter-Value has a maximum length of 8 bytes. The parameter *bct-avp-type* determines the type of parameter. The data type BCtAVPType is defined as follows:

```
BCtAVPType ::=
    INTEGER {
        fwd-qos-control (1),
        type-0-qos (5),
        timing-correction-len-1 (8),
        timing-correction-len-2 (9),
        channel-no-len-3 (10),
        channel-no-len-4 (11),
        ret-bearer-type (16),
        q-diff (17),
        sleep-mode (19),
        slot-size-control (24),
        subband-cf-offset (25),
        -- void: previously defined adjacent-channel (27),
        primary-bearer-len-5 (28),
        primary-bearer-len-6 (29),
        primary-bearer-len-7 (30),
        primary-bearer-len-8 (31),
        min-res-wait (32),
        beam-info (33),
        -- placeholder for priority-primary-bearer-len-5 (36)
        -- placeholder for priority-primary-bearer-len-7 (38)
        -- placeholder for priority-primary-bearer-len-8 (39)
        allocation-size (40),
        subband-cf-offset-change-len-3 (42),
        subband-cf-offset-change-len-6 (45),
        slot-sharing (48),
        frac-peak-data-vol-and-lfrac (56),
        status-ack-control (64),
        report-control (72),
        min-cont-status-delay (80),
        initial-reference-level-len-2 (89),
        initial-reference-level-len-3 (90),
        initial-reference-level-len-4 (91),
        initial-reference-level-len-5 (92),
        initial-reference-level-len-6 (93),
        initial-reference-level-len-7 (94),
        initial-reference-level-len-8 (95),
        forward-carrier-loss (96),
        randomising-control (104),
        res-wait-multiplier (112),
        common-sig-retry (120),
        rncid (128),
        controlled-random-access-len-3 (130),
        controlled-random-access-len-4 (131),
        controlled-random-access-len-5 (132),
        controlled-random-access-len-6 (133),
        shared-reservation-access-len-4 (139),
        shared-reservation-access-len-5 (140),
        shared-reservation-access-len-6 (141),
        shared-reservation-access-len-7 (142),
        shared-reservation-access-len-8 (143),
        fwd-bearer-code-rate-len-1 (144),
        fwd-bearer-code-rate-len-2 (145),
        fwd-bearer-code-rate-len-3 (146),
        fwd-bearer-code-rate-len-4 (147),
        fwd-bearer-code-rate-len-5 (148),
        fwd-bearer-code-rate-len-6 (149),
        fwd-bearer-code-rate-len-7 (150),
        fwd-bearer-code-rate-len-8 (151),
        received-signal-quality (152),
        signal-qual-meas-interval (160),
        leap-second (168),
        satellite-location-len-2 (169),
        access-control (170),
        gps-policy-info (171),
        satellite-state-vectors (172),
        satellite-location-len-6 (173),
        initial-random-access-burst-len-1 (176),
```

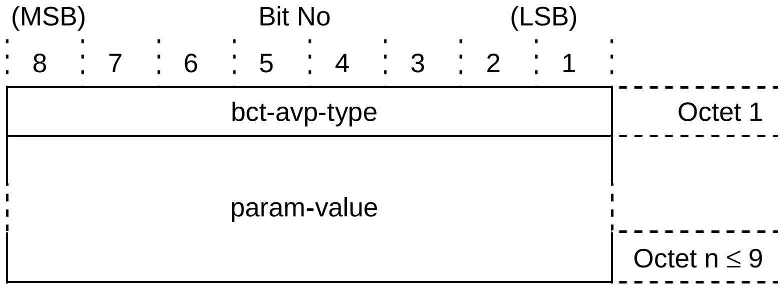
```
        initial-random-access-burst-len-2 (177),
        initial-random-access-burst-len-3 (178),
        initial-random-access-burst-len-4 (179),
        initial-reference-level-set (183),
        ret-link-reference-level-len-1 (184),
        ret-link-reference-level-len-2 (185),
        initial-reference-level-and-max-code-rate (188),
        max-return-code-rate (189),
        ret-link-reference-level-set (191),
        timing-corr-update-interval (192),
        maxdelay-and-delayrange (200),
        reference-level-acknowledge (208),
        utc-date-and-time (212),
        tm-pay-load-position (217),
        -- void: previously defined forward-bearer-tables (224) to (227),
        randomised-initial-access-delay (233),
        intermod-test-info (237),
        plmn-info-len-3 (242),
        plmn-info-len-4 (243),
        nas-sys-info-len-1 (248),
        nas-sys-info-len-2 (249),
        nas-sys-info-len-3 (250),
        nas-sys-info-len-4 (251),
        nas-sys-info-len-5 (252),
        nas-sys-info-len-6 (253),
        nas-sys-info-len-7 (254),
        nas-sys-info-len-8 (255)
    } (0..255)
```

The values are allocated such that the parameter length can be obtained from the lower three bits of *bct-avp-type*. Hence the definition of BCtAVPType above is equivalent to the following:

NOTE: This structure is shown in the text for explanatory purposes and is not included in Annex A.

```
BCtAVPType ::=
    SEQUENCE {
        param-len-type
            INTEGER (0..31),    -- PrmLenType
        param-len
            INTEGER (1..8)      -- PrmLen
    }
```

The parameter type is defined independently for each *param-len* value. This results in a possible 32 parameter types for each parameter-length value. The resulting parameter structure is as follows:



which is equivalent to that shown in Figure 5.93:

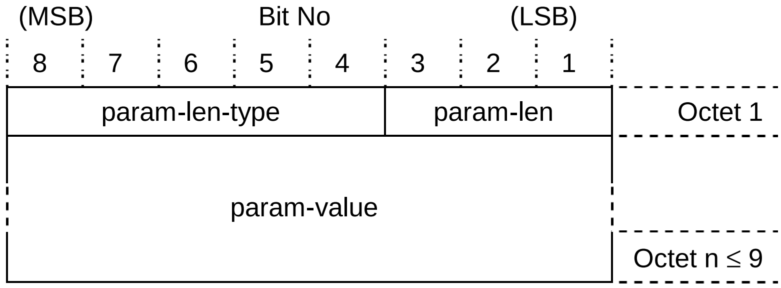


Figure 5.93: BCtAVP Structure

5.7.1.1 Param-len

This three bit field indicates the length of the current parameter. This field forms the least significant three bits of the *param-type* field. The length of the *param-value* field is encoded in the *param-len* field as INTEGER(1..8) which implies that zero length *param-value* fields are not supported.

5.7.1.2 Param-len-type

This five bit field indicates the type of the parameter within the context of the *param-len* field value. This field forms the most significant five-bits of the *param-type* field.

5.7.1.3 Parameters with length ::= 1

Single octet parameters are defined as shown in Table 5.12.

Table 5.12: Bearer Control AVPs containing Param-Values with length ::= 1

ParamLen-Type	ParamType	Param-Value
0x01	0x08	TimingCorrectionParam
0x02	0x10	ReturnBearerTypeParam
0x03	0x18	SlotSizeControlParam
0x04	0x20	MinResWaitParam
0x05	0x28	AllocationSizeParam
0x06	0x30	SlotSharingParam
0x07	0x38	FracPeakDataVolAndLFracParam
0x08	0x40	StatusAckControlParam
0x09	0x48	ReportControlParam
0x0A	0x50	MinContStatusDelayParam
0x0C	0x60	ForwardCarrierLossParam
0x0D	0x68	RandomisingControlParam
0x0E	0x70	ResWaitMultiplierParam
0x0F	0x78	CommonSigRetryParam
0x10	0x80	RNCIdParam
0x12	0x90	ForwardBearerCodeRateParam
0x13	0x98	ReceivedSignalQualityParam
0x14	0xA0	SignalQualityMeasurementIntervalParam
0x15	0xA8	LeapSecondParam
0x16	0xB0	InitialRandomAccessBurstParam
0x17	0xB8	ReturnLinkReferenceLevelParam
0x18	0xC0	TimingCorrectionUpdateIntervalParam
0x19	0xC8	MaxDelayAndDelayRangeParam
0x1A	0xD0	ReferenceLevelAcknowledgeParam
0x1F	0xF8	NASSystemInfoParam

5.7.1.4 Parameters with length ::= 2

Parameter types with length ::= 2 are as shown in Table 5.13.

Table 5.13: Bearer ControlAVPs containing Param-Values with Length ::= 2

ParamLen-Type	Param-Type	Param-Value
0x00	0x01	ForwardQoSControlParam
0x01	0x09	TimingCorrectionParam
0x02	0x11	QDiffParam
0x03	0x19	SubbandCentreFrequencyOffsetsParam
0x04	0x21	BeamInfoParam
0x0B	0x59	InitialReferenceLevelParam
0x12	0x91	ForwardBearerCodeRateParam
0x15	0xA9	SatelliteLocationParam
0x16	0xB1	InitialRandomAccessBurstParam
0x17	0xB9	ReturnLinkReferenceLevelParam
0x1B	0xD9	TMPayloadPositionParam
0x1D	0xE9	RandomisedInitialAccessDelayParam
0x1F	0xF9	NASSystemInfoParam

5.7.1.5 Parameters with length ::= 3

Parameter types with length ::= 3 are as shown in Table 5.14.

Table 5.14: Bearer ControlAVPs containing Param-Values with Length ::= 3

ParamLen-Type	Param-Type	Param-Value
0x01	0x0A	ChannelNoParam
0x05	0x2A	SubbandCentreFrequencyOffsetChangeParam
0x0B	0x5A	InitialReferenceLevelParam
0x10	0x82	ControlledRandomAccessParam
0x12	0x92	ForwardBearerCodeRateParam
0x15	0xAA	AccessControlParam
0x16	0xB2	InitialRandomAccessBurstParam
0x1E	0xF2	PLMNInfoParam
0x1F	0xFA	NASSystemInfoParam

5.7.1.6 Parameters with length ::= 4

Parameter types with length ::=4 are as shown in Table 5.15.

Table 5.15: Bearer ControlAVPs containing Param-Values with Length ::= 4

ParamLen-Type	Param-Type	Param-Value
0x01	0x0B	ChannelNoParam
0x02	0x13	SleepModeParam
0x0B	0x5B	InitialReferenceLevelParam
0x10	0x83	ControlledRandomAccessParam
0x11	0x8B	SharedReservationAccessParam
0x12	0x93	ForwardBearerCodeRateParam
0x15	0xAB	GPSPolicyInfoParam
0x16	0xB3	InitialRandomAccessBurstParam
0x1E	0xF3	PLMNInfoParam
0x1F	0xFB	NASSystemInfoParam

5.7.1.7 Parameters with length ::= 5

Parameter types with length ::= 5 are as shown in Table 5.16.

Table 5.16: Bearer ControlAVPs containing Param-Values with Length ::= 5

ParamLen-Type	Param-Type	Param-Value
0x03	0x1C	PrimaryBearerParam
0x0B	0x5C	InitialReferenceLevelParam
0x10	0x84	ControlledRandomAccessParam
0x11	0x8C	SharedReservationAccessParam
0x12	0x94	ForwardBearerCodeRateParam
0x15	0xAC	SatelliteStateVectorsParam
0x17	0xBC	InitialReferenceLevelAndMaxCodeRateParam
0x1A	0xD4	UTCDateAndTimeParam
0x1F	0xFC	NASSystemInfoParam

5.7.1.8 Parameters with length ::= 6

Parameter types with length ::= 6 are as shown in Table 5.17.

Table 5.17: Bearer ControlAVPs with Containing Param-Values with Length ::= 6

ParamLen-Type	Param-Type	Param-Value
0x00	0x05	Type0QoSParam
0x03	0x1D	PrimaryBearerParam
0x05	0x2D	SubbandCentreFrequencyOffsetChangeParam
0x0B	0x5D	InitialReferenceLevelParam
0x10	0x85	ControlledRandomAccessParam
0x11	0x8D	SharedReservationAccessParam
0x12	0x95	ForwardBearerCodeRateParam
0x15	0xAD	SatelliteLocationParam
0x17	0xBD	MaxReturnCodeRateParam
0x1F	0xFD	NASSystemInfoParam

5.7.1.9 Parameters with length::= 7

Parameter types with length ::= 7 are as shown in Table 5.18.

Table 5.18: Bearer ControlAVPs containing Param-Values with Length ::= 7

ParamLen-Type	Param-Type	Param-Value
0x03	0x1E	PrimaryBearerParam
0x0B	0x5E	InitialReferenceLevelParam
0x11	0x8E	SharedReservationAccessParam
0x12	0x96	ForwardBearerCodeRateParam
0x1F	0xFE	NASSystemInfoParam

5.7.1.10 Parameters with length ::= 8

Parameter types with length ::= 8 are as shown in Table 5.19.

Table 5.19: Bearer ControlAVPs containing Param-Values with Length ::= 8

ParamLen-Type	Param-Type	Param-Value
0x03	0x1F	PrimaryBearerParam
0x0B	0x5F	InitialReferenceLevelParam
0x11	0x8F	SharedReservationAccessParam
0x16	0xB7	InitialReferenceLevelSet
0x17	0xBF	ReturnLinkReferenceLevelSet
0x1F	0xFF	NASSystemInfoParam

5.7.2 TimingCorrectionParam

This parameter is used within a SpecificAVPList or AVPList SDU to provide timing correction information to a particular UE for all UE Hardware Units controlled by the particular instance of this Bearer Control process type. The structure is defined below, with format as shown in Figures 5.94 to 5.95.

```
TimingCorrectionParam ::=
    CHOICE {
        timing-correction-8
            INTEGER (-128 .. 127),
        timing-correction-16
            INTEGER (-32768 .. 32767)
    }
```

(MSB) 8	7	6	5	Bit No 4	3	2	(LSB) 1	
0	0	0	0	1	0	0	0	Octet 1
TimingCorrectionParam								Octet 2

Figure 5.94: TimingCorrection BCtAVP (one octet)

(MSB) 8	7	6	5	Bit No 4	3	2	(LSB) 1	
0	0	0	0	1	0	0	1	Octet 1
Timing								Octet 2
CorrectionParam								Octet 3

Figure 5.95: TimingCorrection BCtAVP (two octets)

The value represents the timing correction in units of 0,25 symbols (rounded to the nearest unit) with a symbol in this context being referenced to the T1 bearer type (33 600 Bd). It is encoded either as a single byte signed (two's-complement) integer providing an adjustment range from -32 symbols to +31,75 symbols or as a two-byte signed (two's-complement) integer providing an adjustment range from -8 192 symbols to 8 191,75 symbols. Positive values indicate that the UE transmits too early and negative values indicate that the UE transmits too late.

5.7.3 ReturnBearerTypeParam

This parameter is used within a *bb-avp-list* or broadcast AVPList to specify the default return bearer type to be used by all UEs tuned to the forward bearer. If a return bearer type is specified in the ReturnSchedule SDU transmitted on the same forward bearer (see clauses 5.4.5.2 and 5.4.5.5) then the value in the ReturnSchedule SDU shall override the default. This AVP will also be sent in a BCtESDU in a Handover message to signal the default bearer type on the target bearer. The structure is defined below, with format as shown in Figure 5.96.

```
ReturnBearerTypeParam ::=
    SEQUENCE {      cw80-acquisition-present
            BOOLEAN,
        only-timed-access-allowed
            BOOLEAN,
        reserved
            BIT STRING (SIZE (2)),
        r-bearer-type-short
            ReturnBearerTypeShort
    }
```

(MSB)			Bit No				(LSB)	
8	7	6	5	4	3	2	1	
0	0	0	1	0	0	0	0	Octet 1
cw80	otaa	reserved		r-bearer-type-short				Octet 2

Figure 5.96: ReturnBearerType BCtAVP

The flags *cw80-acquisition-present* and *only-timed-access-allowed* shall apply to all R80T0.5Q and R80T1Q bearers regardless of the value in the *r-bearer-type-short* field.

5.7.4 SlotSizeControlParam

This parameter is used to specify the parameters which are to be used within the UE Bearer Control algorithms for calculating and reporting queue sizes. The structure is defined below, with format as shown in Figure 5.97.

```
SlotSizeControlParam ::=
    INTEGER (0..255)    -- min-offered-size
```

(MSB)			Bit No				(LSB)	
8	7	6	5	4	3	2	1	
0	0	0	1	1	0	0	0	Octet 1
min-offered-size								Octet 2

Figure 5.97: SlotSizeControl BCtAVP

The units of *min-offered-size,* is in bytes. The default value for *min-offered-size* is 5 bytes. The use of these parameters is specified in ETSI TS 102 744-3-2 [10].

5.7.5 MinResWaitParam

This parameter is used in an AVPList SDU within an EPDU carried with an Establish or Modify AL-SigPDU to define the minimum timeout period that the specific connection shall use when waiting for a slot reservation. The structure is defined below, with format as shown in Figure 5.98.

```
MinResWaitParam ::=
    INTEGER (0 .. 255)
```

(MSB)			Bit No				(LSB)	
8	7	6	5	4	3	2	1	
0	0	1	0	0	0	0	0	Octet 1
MinResWaitParam								Octet 2

Figure 5.98: MinResWait BCtAVP

MinResWait is in units of 40 ms so the range of values supported is 0 to 10,2 seconds. If this parameter is absent then the default value of 6 seconds shall be utilized in the UE.

5.7.6 AllocationSizeParam

This parameter is used within an AVPList SDU within an EPDU carried with an Establish or Modify AL-SigPDU and defines the default segment size minus one to be used for the connection. The structure is defined below, with format as shown in Figure 5.99.

```
AllocationSizeParam ::=
    INTEGER (0 .. 255)
```

(MSB)			Bit No				(LSB)	
8	7	6	5	4	3	2	1	
0	0	1	0	1	0	0	0	Octet 1
AllocationSizeParam								Octet 2

Figure 5.99: AllocationSize BCtAVP

AllocationSize in units of bytes. If this parameter is absent the default value of 32 bytes is used.

5.7.7 SlotSharingParam

This parameter is used within in an AVPList SDU within an EPDU carried with an Establish or Modify AL-SigPDU and defines whether slots allocated to this connection may also carry Queue Status or Data from other connections from this UE. The structure is defined below, with format as shown in Figure 5.100.

```
SlotSharingParam ::=
    SEQUENCE {
        uess-takes-priority
            BOOLEAN,
        other-vbr-allowed
            BOOLEAN,
        other-cbr-allowed
            BOOLEAN,
        qlen-based-signalling
            BOOLEAN,
        variable-bit-rate
            BOOLEAN,
        constant-bit-rate
            BOOLEAN,
        other-data-allowed
            BOOLEAN,
        other-status-allowed
            BOOLEAN
    }
```

(MSB)			Bit No				(LSB)	
8	7	6	5	4	3	2	1	
0	0	1	1	0	0	0	0	Octet 1
utp	ova	oca	qbs	vbr	cbr	oda	osa	Octet 2

Figure 5.100: SlotSharing BCtAVP

The *uess-takes-priority* flag is set if UE specific signalling connection takes priority over the data from this connection (i.e. the UESS is allowed to 'steal' resources from this connection). The default value for this parameter is FALSE.

The *other-vbr-allowed* flag is set if data or signalling from other variable bit rate connections (those which utilize QRate-based-signalling) is allowed to share this slot. The default value for this parameter is FALSE.

The *other-cbr-allowed* flag is set if data from other constant bit rate data connections (those which do not utilize volume or rate-based-signalling mechanisms) is allowed to share this slot. The default value for this parameter is FALSE.

The *qlen-based-signalling* flag is set when the QLen SDUs are to be used as the signalling mechanism instead of the acknowledged Status SDUs for UE specific signalling connections and RABs that require queue-length based signalling. The default value for this parameter is FALSE.

The *variable-bit-rate* flag is set for RABs that are to utilize the QRate SDUs. The default value for this parameter is FALSE.

The *constant-bit-rate* flag is set if a constant bit rate connection is established (a connection that does not utilize either volume or rate-based-signalling), the default value of this parameter is TRUE for Transparent mode connections or FALSE otherwise.

The *other-data-allowed* flag is set if RABs utilize the queue-length based signalling (either QLen or Status SDUs) are allowed to include data in resources allocated to this connection. The default value for this parameter is TRUE.

The *other-status-allowed* flag is set if RABs utilize the queue-length based signalling (either QLen or Status SDUs) are allowed to include queue status reports within resources allocated to this connection. The default value for this parameter is TRUE.

The use of *qlen-based-signalling*, *variable-bit-rate*, and *constant-bit-rate* flags is exclusive - only one of these flags shall be set for any connection.

5.7.8 FracPeakDataVolAndLFracParam

This parameter is used within a SpecificAVPList or AVPList SDU and specifies parameters to be used within the UE Bearer Control process for all connections. The structure is defined below, with format as shown in Figure 5.101.

```
FracPeakDataVolAndLFracParam ::=
    SEQUENCE {
        frac-peak-data-vol      INTEGER (0..15),
        l-frac                  INTEGER (0..15)
    }
```

(MSB) 8	7	6	5	Bit No 4	3	2	(LSB) 1	
0	0	1	1	1	0	0	0	Octet 1
frac-peak-data-vol				l-frac				Octet 2

Figure 5.101: FracPeakDataVolAndLFrac BCtAVP

Units of *frac-peak-data-vol* and *l-frac* are 0,125 so the value range is between 0 and 1,875. The default values for *frac-peak-data-vol* and *l-frac* are both 0,5.

5.7.9 StatusAckControlParam

This parameter is used within a SpecificAVPList or AVPList SDU and specifies parameters to be used within the UE Bearer Control process for all connections. The structure is defined below, with format as shown in Figure 5.102.

```
StatusAckControlParam ::=
    SEQUENCE {
        t-ack-wait          INTEGER (0..15),
        max-unackstatus     INTEGER (0..15)
    }
```

(MSB) 8	7	6	5	Bit No 4	3	2	(LSB) 1	
0	1	0	0	0	0	0	0	Octet 1
t-ack-wait				max-unackstatus				Octet 2

Figure 5.102: StatusAckControl BCtAVP

Units of *t-ack-wait* are 40 ms. The default value for *t-ack-wait* is 6, resulting in 240 ms. This time is offset by the round-trip-time (rtt) specified for the UE algorithms in ETSI TS 102 744-3-2 [10]. The default value for *max-unackstatus* is 8.

5.7.10 ReportControlParam

This parameter is used within a SpecificAVPList or AVPList SDU and specifies a parameter to be used within the UE Bearer Control process for all connections. The structure is defined below, with format as shown in Figure 5.103.

```
ReportControlParam ::=  INTEGER (0 .. 255)
```

8 (MSB)	7	6	5	4	3	2	1 (LSB)	
0	1	0	0	1	0	0	0	Octet 1
ReportControlParam								Octet 2

Figure 5.103: ReportControl BCtAVP

Units of *Report Control* are 0,0625, so the value specified is between 0 and 15,9375. The default value is 0,3125.

5.7.11 MinContStatusDelayParam

This parameter is used within a SpecificAVPList or AVPList SDU and specifies a parameter to be used within the UE Bearer Control process for all connections. The structure is defined below, with format as shown in Figure 5.104.

```
MinContStatusDelayParam ::= INTEGER (0 .. 255)
```

8 (MSB)	7	6	5	4	3	2	1 (LSB)	
0	1	0	1	0	0	0	0	Octet 1
MinContStatusDelayParam								Octet 2

Figure 5.104: MinContStatusDelay BCtAVP

Units of *MinContStatusDelay* are 40 ms so the range of values supported is 0 to 10,2 seconds. The default value for *MinContStatusDelay* is 80 ms.

5.7.12 RandomisingControlParam

This parameter is used within a SpecificAVPList or AVPList SDU and specifies parameters to be used within the UE Bearer Control process for all connections. The structure is defined below, with format as shown in Figure 5.105.

```
RandomisingControlParam ::=
    SEQUENCE {
        max-randomising-level       INTEGER (0..7),
        frame-randomiser            INTEGER (0..31)
    }
```

8 (MSB)	7	6	5	4	3	2	1 (LSB)	
0	1	1	0	1	0	0	0	Octet 1
max-rand-level			frame-randomiser					Octet 2

Figure 5.105: RandomisingControl BCtAVP

The default value for *max-randomising-level* is 4 and for *frame-randomiser* is 2.

5.7.13 ResWaitMultiplierParam

This parameter is broadcast by the RNC in congestion conditions to extend the time that UEs shall wait for reserved slots (before resending Status SDUs). This parameter overrides previously specified values. The structure is defined below, with format as shown in Figure 5.106.

```
ResWaitMultiplierParam ::=
    INTEGER (0..255)
```

(MSB)			Bit No				(LSB)	
8	7	6	5	4	3	2	1	
0	1	1	1	0	0	0	0	Octet 1
ResWaitMultiplier								Octet 2

Figure 5.106: ResWaitMultiplier BCtAVP

ResWaitMultiplier is in units of 0,125 so the range of values supported is 0 to 31,875. If this parameter is absent the default value of 1 shall be used in the UE.

5.7.14 CommonSigRetryParam

This parameter is used within an AVPList SDU or within a *bb-avp-list* to define retry parameters for the Common Signalling connection (i.e. Registration and Deregistration related packets). CommonSigRetry is defined as follows, with format as shown in Figure 5.107:

```
CommonSigRetryParam ::=
    SEQUENCE {
        maxretrycount
            INTEGER (0..7),
        timeout
            INTEGER (0..31)
    }
```

(MSB)			Bit No				(LSB)	
8	7	6	5	4	3	2	1	
0	1	1	1	1	0	0	0	Octet 1
maxretrycount			timeout					Octet 2

Figure 5.107: CommonSigRetry BCtAVP

The default value of *maxretrycount* is 3 (i.e. the UE shall make four attempts before giving up). *Timeout* is in units of 0,5 s. The default value of *timeout* is 10 (i.e. 5 s).

5.7.15 RNCIdParam

This parameter is used during inter-RNC handover to advise the UE of the RNC ID of the target RNC and is sent in a BCtESDU of the Handover message. The structure is defined below, with format as shown in Figure 5.108.

```
RNCIdParam ::=
    SEQUENCE {
        rnc-id
            INTEGER (0..255)
    }
```

(MSB)			Bit No				(LSB)	
8	7	6	5	4	3	2	1	
1	0	0	0	0	0	0	0	Octet 1
RNCId								Octet 2

Figure 5.108: RNCId BCtAVP

5.7.16 ForwardBearerCodeRateParam

5.7.16.0 General

This parameter describes coding rate and modulation index changes in the FEC blocks of the current or the next frame as specified in clause 5.7.16.1. On forward bearers that use no outer interleaving, this AVP is only provided if the coding rate changes from the coding rate of the first FEC block in the frame which is implicitly signalled in the unique word.

On outer interleaved forward bearers (i.e. F80T2.5X and F80T5X) this AVP is included if the coding rate or modulation index of any FEC Block in the frame are different from the first FEC block in the same frame. Optionally, this AVP may be included in every frame even if there is no such change.

This AVP shall be carried in a *bb-avp-list* (see clause 5.4.4.8) if a BulletinBoard SDU is scheduled for transmission, otherwise it shall be carried in a Broadcast AVPList BCtSDU (see clause 5.4.10). Furthermore, this AVP shall be the first one in the AVPList used. The structure is defined below, with format as shown in Figure 5.109.

```
ForwardBearerCodeRateParam ::=
    SEQUENCE (1..8) OF BlockRate
BlockRate ::=
    SEQUENCE {
        modulation-index-increase
            ModulationIndexIncrease,
        block-num
            INTEGER (0..7),
        coding-rate
            CodeRate
    }
```

The parameter *modulation-index-increase* specifies an increase in the modulation index being used in the FEC block specified by *block-num*. The increase in the modulation index specified shall remain selected for the rest of the frame or until another change is signalled as another BlockRate parameter in the same AVP. Due to the limitations of the physical layer, this information element is only used to switch between 4-QAM and 16-QAM (when the value is set to TRUE). When the value is set to FALSE, the modulation index is the same as that used for the first FEC block of the frame.

(MSB) 8	7	6	Bit No 5	4	3	2	(LSB) 1	
1	0	0	1	0	prm-len (= n - 1)			Octet 1
mii	block-num 1			coding-rate 1				Octet 2
mii	block-num n			coding-rate n				Octet n+1

Figure 5.109: ForwardBearerCodeRate BCtAVP

5.7.16.1 Block-num

The parameter *block-num* defines the FEC Block within the frame at which the coding rate changes, where *block-num* = 0 refers to the first block in the frame, regardless of the FEC block in which the AVP is transmitted.

When operating with F80T1Q-4B, F80T1X-4B and F80T45X-8B bearers, the AVP refers to FEC blocks in the current frame.

EXAMPLE 1: When used with an F80T45X-8B for the FEC blocks:

block-num-1 = 2; coding-rate-1 = -1
(FEC Blocks 0-1, Signalled by UW)
(FEC Block 2, Bearer Sub-type L1)
block-num-2 = 3; coding-rate-2 = -3
(FEC Block 3, Bearer Sub-type L3)
block-num-3 = 4; coding-rate-3 = 0
(FEC Blocks 4-7, Bearer Sub-type R)

When operating with F80T2.5X and F80T5X bearers, the AVP refers to FEC blocks in the next frame. Also when operating with these forward bearers, in order to signal FEC block numbers for forward bearers with more than 8 FEC blocks per frame, the following rules apply:

If the value of '*block-num-n'* is less than or equal to the value of '*block-num-m' where m<n*, then the FEC block number to which this and all subsequent information element refers shall be increased by 8 to reference the FEC block number *(8 +* value of *block-num-n)*. This allows this AVP to reference any FEC block in forward bearers supporting more than 8 FEC blocks per frame, however the constraint remains that a maximum number of 8 BlockRate values may be signalled with this AVP.

EXAMPLE 2: When used with an F80T5X16-9B for the FEC blocks in the next frame:

block-num-1 = 0; coding-rate-1 = -3
(FEC Blocks 0-2, Bearer Sub-type L3 (same as next frame UW))
block-num-2 = 3; coding-rate-2 = -2
(FEC Blocks 3-5, Bearer Sub-type L2)
block-num-3 = 6; coding-rate-3 = 0
(FEC Blocks 6-7, Bearer Sub-type R)
block-num-4 = 0; coding-rate-4 = 1
(FEC Block 8, Bearer Sub-type H1)

When operating with F80T2.5X4 and F80T5X4 bearers which support variable rate modulation, the AVP signals both the coding rate and the modulation for FEC blocks in the next frame.

EXAMPLE 3: When used with an F80T5X4-9B for the FEC blocks in the next frame:

modulation-index-increase =0, block-num-1 = 0; coding-rate-1 = -4
(4-QAM mod, FEC Blocks 0-2, Bearer Sub-type L4 (same as next frame UW))
modulation-index-increase =0, block-num-2 = 3; coding-rate-2 = -2
(4-QAM mod, FEC Blocks 3-5, Bearer Sub-type L2)
modulation-index-increase =1, block-num-3 = 6; coding-rate-3 = 0
(16-QAM mod, FEC Blocks 6-7, Bearer Sub-type R)
modulation-index-increase =1, block-num-4 = 0; coding-rate-4 = 1
(16-QAM mod, FEC Block 8, Bearer Sub-type H1)

5.7.16.2 CodeRate

The parameter *coding-rate* specifies the bearer subtype being used in the FEC block specified by *block-num*. The new bearer subtype specified shall remain selected for the rest of the frame or until another change is signalled, either as another block-num and coding-rate pair in the same AVP or in another AVP included in a further FEC block.

```
CodeRate ::=
    INTEGER (-8 .. 7)
```

where Bearer Subtypes (i.e. coding rates) map onto CodeRate values as shown in Table 5.20.

Table 5.20: CodeRate Values

CodeRate Value	-8	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7
R80T0.5Q and R80T1Q only	L8	L7	L6	L5	L4	L3	L2	L1	R	H1	L14	L13	L12	L11	L10	L9
All other Bearers	L8	L7	L6	L5	L4	L3	L2	L1	R	H1	H2	H3	H4	H5	H6	n/a

5.7.17 ReceivedSignalQualityParam

This parameter which is carried in a *status-avp-list* (or AVPList for UESS connections that use the QLen signalling) is used by the UE to report the Carrier-to-Noise Ratio (C/N_0) of the received forward bearer to the RNC. The structure is defined below, with format as shown in Figure 5.110.

```
ReceivedSignalQualityParam ::=
    INTEGER (0..255)
```

(MSB)			Bit No				(LSB)	
8	7	6	5	4	3	2	1	
1	0	0	1	1	0	0	0	Octet 1
ReceivedSignalQuality								Octet 2

Figure 5.110: ReceivedSignalQuality BCtAVP

The value of ReceivedSignalQualityParam is calculated by subtracting 20 dB from the measured C/N_0 value and multiplying the result by 4. The value therefore represents a C/N_0 range from 20 dBHz to 83,75 dBHz.

5.7.18 SignalQualityMeasurementIntervalParam

This parameter is used by RNC to control the number of frames over which the UE shall average the Carrier-to-Noise Ratio (C/N_0) measurement reported in the Received Signal Quality Bearer Control AVP. The structure is defined below, with format as shown in Figure 5.111.

```
SignalQualityMeasurementIntervalParam ::=
    SEQUENCE {
        reporting-on
            BOOLEAN,
        reserved
            BIT STRING (SIZE (2)),
        interval
            INTEGER (1..32)
    }
```

(MSB)			Bit No				(LSB)	
8	7	6	5	4	3	2	1	
1	0	1	0	0	0	0	0	Octet 1
RO	reserved		interval					Octet 2

Figure 5.111: SignalQualityMeasurementInterval BCtAVP

The value in the *interval* field shall be multiplied by eight to specify the number of frames over which the sliding average for the signal quality measurement shall operate, providing a range from 640 ms (*interval* = 1) to 20,48 s (*interval* = 32). The Boolean flag *reporting-on* is used by the RNC to control the reporting at a UE.

5.7.19 AccessControlParam

This parameter is used within a *bb-avp-list* to control the initial access of UEs to the RNC. This AVP is only included if access restrictions are to be enforced. The structure is defined below, with format as shown in Figure 5.112.

```
AccessControlParam ::=
    SEQUENCE {
        emergency-call-override-access-class
            BOOLEAN,
        reserved
            BIT STRING (SIZE (5)),
        sim-card-control
            SimCardControl,
        access-class
            BIT STRING (SIZE (16))
    }
```

(MSB) 8	7	6	5	4	3	2	(LSB) 1	
1	0	1	0	1	0	1	0	Octet 1
e	reserved					sim-card-control		Octet 2
access-class (class 15..8)								Octet 3
access-class (class 7..0)								Octet 4

Figure 5.112: AccessControl BCtAVP

If the field *emergency-call-override-access-class* is TRUE, then the access restrictions shall not apply to UEs attempting to make an emergency call. Otherwise only those UEs belonging to an access class for which the corresponding bit is set to one in the field *access-class* shall attempt to access the RNC. Bit 8 of Octet 3 corresponds to Access Class 15, while Bit 1 of Octet 4 corresponds to Access Class 0.

The field *sim-card-control* specifies whether a UE without a SIM card may register at the RNC. The data type SimCardControl is defined as follows:

```
SimCardControl ::=
    INTEGER {
        imei-not-permitted (0),
        imei-permitted-emergency-call-only (1),
        imei-permitted (2)
        } (0..3)
```

5.7.20 InitialRandomAccessBurstParam

5.7.20.0 General

This parameter is used within a *bb-avp-list* to specify the burst types (together with coding rates) which may be used by UEs for initial random access to the RNC. The AVP is only sent if there is a change from the default Initial Random Access burst types and coding rates specified in ETSI TS 102 744-2-1 [8]. The structure is defined below, with format as shown in Figure 5.113.

```
InitialRandomAccessBurstParam ::=
    SEQUENCE (1..4) OF AccessBurst
AccessBurst ::=
    SEQUENCE{
        burst-type
            BurstType,
        coding-rate
            CodeRate
    }
```

(MSB) 8	7	6	5	4	3	2	(LSB) 1	
1	0	1	1	0	prm-len (= n - 1)			Octet 1
burst-type 1				coding-rate 1				Octet 2
burst-type n				coding-rate n				Octet n+1

Figure 5.113: InitialRandomAccessBurst BCtAVP

The parameter *coding-rate* is defined in clause 5.7.16.2.

5.7.20.1 BurstType

The *burst-type* field specifies the type of burst that shall be used by the UE for initial random access purposes and is defined as follows:

```
BurstType   ::=
    INTEGER {
        r20t05q (0),
        -- reserved (1),
        -- reserved (2),
        r20t1q (3),
        r5t2q(4),
        -- reserved (5),
        r5t45q (6),
        -- reserved (7),
        r80t0.5q (8),
        r80t1q (9),
        r5t1x (10),
        -- reserved (11),
        r5t2x (12),
        -- reserved (13)
        r5t45x (14),
        -- reserved (15)
        } (0..15)
```

5.7.21 ReturnLinkReferenceLevelParam

This parameter is used within a SpecificAVPList or AVPList SDU to specify the reference level for the return link adaptation to a particular UE Hardware Unit. It may also include a transmit power backoff value or a backoff range over which the UE can trade-off transmit EIRP versus coding rate. The particular UE Hardware Unit that the AVP is intended for is determined by the translated Bearer Connection ID contained in the SpecificAVPList SDU or the translated Bearer Connection ID of the BCtPDU used to carry the AVPList SDU. The structure is defined below, with format as shown in Figures 5.114 and 5.115.

```
ReturnLinkReferenceLevelParam ::=
    SEQUENCE {
        reserved
            BIT STRING (SIZE (2)),
        reference-level
            ControlIndex,
        backoff-field
            SEQUENCE {
                mode
                    INTEGER (0..3),
                reserved
                    BIT STRING (SIZE (2)),
                backoff
                    INTEGER (0..15)
        } OPTIONAL
    }
```

(MSB)			Bit No				(LSB)	
8	7	6	5	4	3	2	1	
1	0	1	1	1	0	0	0	Octet 1
reserved		reference-level						Octet 2

Figure 5.114: ReturnLinkReferenceLevel BCtAVP (*backoff* value not included)

(MSB)			Bit No				(LSB)	
8	7	6	5	4	3	2	1	
1	0	1	1	1	0	0	1	Octet 1
reserved		reference-level						Octet 2
mode		reserved		backoff				Octet 3

Figure 5.115: ReturnLinkReferenceLevel BCtAVP (*backoff* value included)

The parameter *reference-level* is of data type ControlIndex which is specified in clause 5.4.12.6. UE classes that support wrap around of control index shall interpret the received reference-level value according to the following rule:

If (reference-level < 16) then
reference-level = reference-level + 64
EndIf

The parameter *backoff* shall be used in different ways depending on the value of *mode*, as shown in Table 5.21. Further information on the reference level and the use of the parameter *backoff* is provided in ETSI TS 102 744-3-2 [10], clause 8.5.1.1.

Table 5.21: ReturnLinkReferenceLevel Mode Values: Expected UE Behaviour

Value of *mode*	EIRP and Coding Rate
0	The UE hardware unit shall reduce its EIRP level by the amount (in dB) specified in *backoff*. The UE shall use a fixed coding rate which matches the operating point specified in *reference-level* (i.e. power control only).
1	The UE hardware unit shall use the value in backoff to determine the minimum EIRP level at which the UE can transmit such that the coding rate may be selected over a possible range of ControlIndex values from reference-level to reference-level − backoff + 1. The value of backoff is thus applied in units of 0,5 dB.
2	The UE hardware unit shall use the value in backoff to determine the minimum EIRP level at which the UE can transmit, such that the coding rate may be selected over a possible range of ControlIndex values from reference-level to reference-level − (2 × backoff) + 1 The value of backoff is thus applied in units of 1 dB.
3	[reserved]

If the AVP does not include *mode* and *backoff* then the UE shall use the last value of *mode* and *backoff* received. If no *mode* value has been received, a default *backoff* of 0 dB (with *mode* ::= 1) shall be used.

5.7.22 TimingCorrectionUpdateIntervalParam

This AVP is used within a *bb-avp-list* or a broadcast AVPList to specify the maximum time interval until a UE has to update its return channel transmission timing as specified in ETSI TS 102 744-3-2 [10]. The value of TimingCorrectionUpdateInterval is expressed in minutes. A value of zero is invalid and shall be ignored. The AVP has the following structure, with format as shown in Figure 5.116:

```
TimingCorrectionUpdateIntervalParam ::=
    INTEGER (0..255)
```

(MSB)			Bit No				(LSB)	
8	7	6	5	4	3	2	1	
1	1	0	0	0	0	0	0	Octet 1
TimingCorrectionUpdateInterval								Octet 2

Figure 5.116: TimingCorrectionUpdateInterval Bearer Control AVP

5.7.23 MaxDelayAndDelayRangeParam

5.7.23.0 General

This parameter is used to specify the maximum delay and the delay range required for initial access of UEs to the RNC (see ETSI TS 102 744-3-2 [10]). The data type MaxDelayAndDelayRangeParam is defined below and the corresponding AVP has the structure shown in Figure 5.117:

```
MaxDelayAndDelayRangeParam ::=
    SEQUENCE {
        use-primary-rctc
            BOOLEAN,
        max-delay
            INTEGER (0..7),
        delay-range
            INTEGER (0..15)
    }
```

(MSB)			Bit No				(LSB)	
8	7	6	5	4	3	2	1	
1	1	0	0	1	0	0	0	Octet 1
up	max-delay			delay-range				Octet 2

Figure 5.117: MaxDelayAndDelayRange BCtAVP

5.7.23.1 Use-primary-rctc

The flag *use-primary-rctc* is used to control the method that shall be used by UEs for return channel timing control. If the flag is TRUE, then the UEs shall use the combination of primary and secondary method (as described in ETSI TS 102 744-3-2 [10]). If the flag is FALSE, then the UEs shall use the secondary method only.

5.7.23.2 Max-delay

The field *max-delay* specifies the maximum delay from the satellite to a UE located in the beam. The maximum delay is calculated from this value as follows:

$$\text{maximum delay} = 120 \text{ ms} + \textit{max-delay} \times 5 \text{ ms}$$

5.7.23.3 Delay-range

The field *delay-range* specifies the maximum delay variation within the beam in units of 5 ms.

5.7.24 ReferenceLevelAcknowledgeParam

This parameter, which is carried in a *status-avp-list* (or AVPList for connections that do not require the transmission of Status SDUs, e.g. constant bit rate), is used by the UE to report that a ReturnLinkReferenceLevel, ReturnLinkReferenceLevelSet, InitialReferenceLevelSet or InitialReferenceLevelAndMaxCodeRate AVP has been received from the RNC. The data type ReferenceLevelAcknowledgeParam is defined below and the corresponding AVP has the structure as shown in Figure 5.118:

```
ReferenceLevelAcknowledgeParam ::=
    SEQUENCE {
        reserved
            BIT STRING (SIZE (7)),
        ack
```

```
        BOOLEAN
}
```

(MSB)			Bit No				(LSB)	
8	7	6	5	4	3	2	1	
1	1	0	1	0	0	0	0	Octet 1
reserved							ack	Octet 2

Figure 5.118: ReferenceLevelAcknowledge Bearer Control AVP

The *ack* parameter is always TRUE when this AVP is sent, since a negative acknowledgement is impossible (the UE would have no knowledge that it had missed receiving the ReturnLinkReferenceLevel AVP). One octet is necessary to convey this information since the BCtAVP structure does not support empty AVPs.

5.7.25 ForwardQoSControlParam

This parameter is used to adjust the *response-time* parameter used in the Connection Layer of the UE. The AVP can be broadcast in a *bb-avp-list* or AVPList, or sent to a specific UE in a SpecificAVPList or AVPList SDU. The data type ForwardQoSControlParam is defined below and the corresponding AVP has the structure as shown in Figure 5.119:

```
ForwardQoSControlParam :: =
    SEQUENCE {
        thp
            TrafficHandlingPriority,
        reserved
            BIT STRING (SIZE (3)),
        f-bearer
            FwdBearer,
        response-time
            INTEGER (0..255)
    }
```

(MSB)			Bit No				(LSB)	
8	7	6	5	4	3	2	1	
0	0	0	0	0	0	1	0	Octet 1
thp		reserved			f-bearer			Octet 2
response-time								Octet 3

Figure 5.119: ForwardQoSControl Bearer Control AVP

If this AVP is broadcast, then all the UEs which operate Acknowledged Mode connection(s) which belong to the specified UMTS Traffic Class and Priority specified in *thp* shall reconfigure these connections with the new *response-time* setting.

The parameter *thp* is defined as follows:

```
TrafficHandlingPriority ::=
    INTEGER {
        traffic-handling-priority-15-or-background (0),
        traffic-handling-priority-1 (1),
        traffic-handling-priority-2 (2),
        traffic-handling-priority-3 (3)
    } (0..3)
```

The parameter value reflects the Traffic Handling Priority received with the RANAP Radio Access Bearer (RAB) AssignmentRequest message if the Core Network requested a RAB setup for an Interactive Class connection. If a Background Class connection was requested by the Core Network then this shall be signalled to the UE as *traffic-handling-priority-15-or-background.*

If the AVP is addressed to a specific connection, the parameter *thp* shall be ignored and the value of *response-time* shall be applied to the specified connection only.

The parameter *f-bearer* specifies which forward bearer the *response-time* value applies. The parameter *response-time* is defined in units of 40 ms.

5.7.26 InitialReferenceLevelParam

This parameter is used to override the default Initial Reference Level behaviour that is used by a UE on this bearer control prior to link adaptation being completed, to determine the power level for the Initial Random Access Burst transmitted by a UE. When utilized, this AVP will be broadcast in a *bb-avp-list* of a Bulletin Board BCtSDU or AVPList. The structure is defined below, with format as shown in Figure 5.120.

```
InitialReferenceLevelParam ::=
    SEQUENCE {
        msb
            BIT STRING (SIZE (1)),
            -- most significant bit of UE-class
            --for all UEs in this list
        cim
            BIT STRING (SIZE (1)),
            -- control-index-offset-multiplier
        control-index-base
            ControlIndex,
        ue-init-ref-list
            SEQUENCE SIZE (1..7) OF UeInitialRef
}
UeInitialRef ::=
    SEQUENCE {
        ue-class-lsb
            INTEGER(0..15),
        -- the least significant 4 bits of the UE class are specified
        -- in this information element
        ctrl-index-offset
            INTEGER (0..15)
        -- Control Index Offset in steps determined by cim:
        -- if cim = 0 then steps are 0.5dB, else 1dB
    }
```

8 (MSB)	7	6	5	4	3	2	1 (LSB)	
0	1	0	1	1	prm-len (= n)			Octet 1
msb	cim	control-index-base						Octet 2
ue-class-lsb-1				ctrl-ind-offset-1				Octet 3
⋮								
ue-class-lsb-n				ctrl-ind-offset-n				Octet n+1

Figure 5.120: InitialReferenceLevelParam BCtAVP

This AVP defines Initial Reference Levels for each UE class that is expected to operate on this Bearer Control, and up to 7 UE classes may be described within each AVP.

The InitialReferenceLevel (in units of 0,5 dB) for each UE class specified in the AVP shall be derived as follows:

$$InitialReferenceLevel = control\text{-}index\text{-}base + (1+cim) \times ctrl\text{-}ind\text{-}offset$$

The behaviour for utilizing the InitialReferenceLevel information is described in ETSI TS 102 744-3-2 [10].

5.7.27 QDiffParam

This parameter is used within an AVPList SDU within an establish or modify SDU to control the reporting of Q-status information for a particular Bearer Connection within the UE Bearer Control. The structure is defined below, with format as shown in Figure 5.121.

```
QDiffParam ::=
    INTEGER (0 .. 65535)
```

(MSB)			Bit No				(LSB)	
8	7	6	5	4	3	2	1	
0	0	0	1	0	0	0	1	Octet 1
QDiff								Octet 2
								Octet 3

Figure 5.121: QDiff BCtAVP

Units of *QDiff* are in bytes. The default value is 1 024 bytes.

5.7.28 SubbandCentreFrequencyOffsetsParam

This parameter is used by RNC to signal to the UE the centre frequency of a 200 kHz subband. A separate value is provided for the forward and return link offset. The structure is defined below, with format as shown in Figure 5.122.

```
SubbandCentreFrequencyOffsetsParam ::=
    SEQUENCE {
        forward-centre-frequency-offset
            INTEGER (0..255),
        return-centre-frequency-offset
            INTEGER (0..255)
    }
```

(MSB)			Bit No				(LSB)	
8	7	6	5	4	3	2	1	
0	0	0	1	1	0	0	1	Octet 1
forward-link-centre-frequency-offset								Octet 2
return-link-centre-frequency-offset								Octet 3

Figure 5.122: SubbandCentreFrequencyOffsets BCtAVP

The value in the *forward-(return)-link-centre-frequency-offset* field shall be multiplied by 1,25 kHz to determine the offset of a 200 kHz subband from a frequency grid aligned to integer multiples of 200 kHz starting at 1 518,000 MHz for the forward and 1 626,500 MHz for the return link. The default value of *forward-(return)-link-centre-frequency-offset* is 80 (equal to 100 kHz) and the AVP is only included if the actual values are different from the default.

5.7.29 SubbandCentreFrequencyOffsetChangeParam

This parameter is used by RNC to signal to the UE changes to the centre frequencies of the 200 kHz subband which may be applicable to parts of the forward or return frequency range. The structure is defined below, with format as shown in Figure 5.123.

```
SubbandCentreFrequencyOffsetChangeParam ::=
    SEQUENCE SIZE (1..2) OF SubbandCFOffsetChangeInfo

SubbandCFOffsetChangeInfo ::=
    SEQUENCE{
        dir
            INTEGER {
                forward (0),
                return(1)
            } (0..1),
        f-start
            INTEGER (0..32767),
        centre-frequency-offset
            INTEGER (0..255)
    }
```

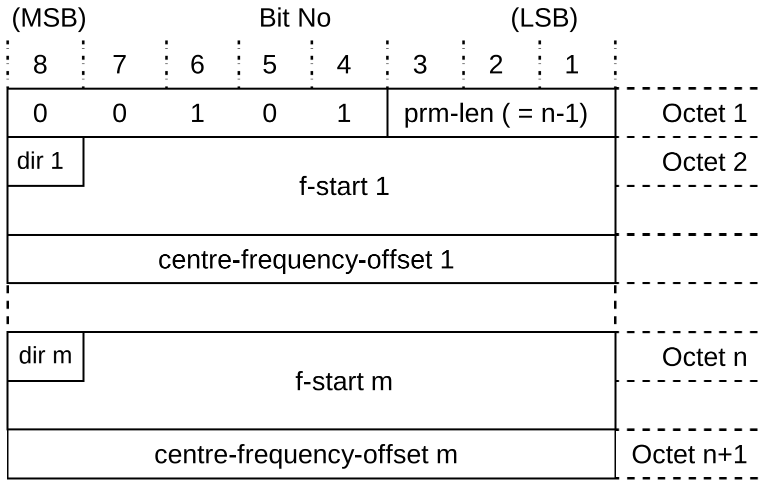


Figure 5.123: SubbandCentreFrequencyOffsetChange BCtAVP

The *dir* field determines the direction (forward or return) while the value in the *f-start* field determines the lowest frequency to which the modified subband frequency offset is applicable. The value in the *f-start* field is provided in units of 100 kHz.

If the subband centre frequency offset is changed more than once across the forward and/or return frequency band, then the SubbandCFOffsetChangeInfo elements shall be in ascending order of *f-start*.

The value in the centre-frequency-offset field shall be multiplied by 1,25 kHz to determine the offset of a 200 kHz subband from a frequency grid aligned to integer multiples of 200 kHz starting at 1 518,000 MHz for the forward and 1 626,500 MHz for the return link.

5.7.30 SatelliteLocationParam

5.7.30.0 General

This parameter is used to describe the satellite location to the UE. The information is carried in the bulletin board. A RNC may transmit information about other satellites in the same *bb-avp-list* or broadcast AVPList, but the information about the current satellite (i.e. the satellite via which the information is being transmitted) shall always be the first SatelliteLocation AVP in this AVP List. If the information for a satellite has been modified (i.e. satellite longitude and/or primary and alternate frequencies have changed), the SatelliteLocation AVP corresponding to that satellite shall always contain the *identifying-frequencies* element (i.e. the AVP shall be of length 6). If the information for the current satellite is modified, there shall be two SatelliteLocation AVPs in the AVP list for this satellite: one AVP of length 2 (first in the AVP list) followed by one of length 6. The structure is defined below, with format as shown in Figure 5.124.

```
SatelliteLocation::=
    SEQUENCE {
        satellite-id
            INTEGER (0..15),
        satellite-location
            PointOneDegreesLongitude,
        identifying-frequencies
            SEQUENCE {
                primary
                    FwdChannelNumber,
                alternate
                    FwdChannelNumber
            } OPTIONAL
    }
```

(MSB)			Bit No				(LSB)	
8	7	6	5	4	3	2	1	
1	0	1	0	1	0	0	1	Octet 1
satellite-id				<				Octet 2
satellite-location							>	Octet 3

Figure 5.124: SatelliteLocation BCtAVP

5.7.30.1 PointOneDegreesLongitude

This field is 12 bits long and describes the location in terms of degrees of longitude East or West to a resolution of 0,1 degrees.

```
PointOneDegreesLongitude ::=
    INTEGER (-2048..2047)
```

The value shall be divided by ten to obtain the satellite location in degrees to an accuracy of 0,1 degree. Positive values of degrees longitude correspond to degrees East; negative values correspond to degrees West. Values in the range -2 048 to -1 801 and 1 800 to 2 047 are invalid.

5.7.31 ChannelNoParam

5.7.31.0 General

This parameter is used to instruct a UE Hardware Unit to tune to the specified channel number. The particular UE Hardware Unit to be tuned is determined by the Translated Bearer Connection Id contained in the SpecificAVPList or the translated Bearer Connection ID of the BCtPDU that carries the AVPList.

The ChannelNoParam AVP may also be included in an AVPList on an LDR or HDR bearer to broadcast a retune instruction to all mobile terminals on the physical bearer. This mechanism is used to allow the RNC to modify a bearer type within a current satellite sub-band.

The structure is defined below, with format as shown in Figures 5.125 and 5.126.

```
ChannelNoParam ::=
    SEQUENCE {
        channel-number
            FwdChannelNumber,
        ack-required
            BOOLEAN,
        f-bearer
            FwdBearer,
        f-bearer-type
            FBearerType
        count-down-field
            SEQUENCE {
                reserved
                    BIT STRING (SIZE (4)),
                count-down
                    CountDown
        } OPTIONAL
    }
```

(MSB)			Bit No				(LSB)	
8	7	6	5	4	3	2	1	
0	0	0	0	1	0	1	0	Octet 1
channel								Octet 2
number								Octet 3
ar	f-bearer			f-bearer-type				Octet 4

Figure 5.125: ChannelNo BCtAVP (count-down not included)

<table>
<tr><td colspan="3">(MSB)</td><td colspan="3">Bit No</td><td colspan="2">(LSB)</td><td></td></tr>
<tr><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td></td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>1</td><td>Octet 1</td></tr>
<tr><td colspan="8">channel</td><td>Octet 2</td></tr>
<tr><td colspan="8">number</td><td>Octet 3</td></tr>
<tr><td>ar</td><td colspan="3">f-bearer</td><td colspan="4">f-bearer-type</td><td>Octet 4</td></tr>
<tr><td colspan="4">reserved</td><td colspan="4">count-down</td><td>Octet 5</td></tr>
</table>

Figure 5.126: ChannelNo BCtAVP (count-down included)

5.7.31.1 Ack-required

The *ack-required* flag shall be set if an acknowledgement (using mechanism described in ETSI TS 102 744-3-2 [10]) is requested from the particular UE to which the AVP is addressed. This flag shall not be set if the AVP is used to move multiple UE.

5.7.31.2 F-bearer and FBearerType

The *f-bearer* field contains a copy of the *f-bearer* number which is present in the Bulletin Board and represents the number of the bearer within the Bearer Control to which the UE is being asked to tune. The *f-bearer-type* field contains a definition of the Bearer Type of the forward bearer to which the UE hardware is being tuned. The data type FBearerType is defined as follows:

```
FBearerType ::=
    INTEGER {
        -- reserved (0)
        f80t1q4b (1),
        f80t1x4b (2),
        f80t45x8b (3),
        f80t1q1b (4),
        f80t25x4-5b (5),
        f80t25x16-5b (6),
        f80t25x32-6b (7),
        f80t25x64-7b (8),
        f80t5x4-9b (9),
        f80t5x16-9b (10),
        f80t5x32-11b (11),
        f80t5x64-13b (12)
        -- reserved (13..15)
        } (0..15)
```

5.7.31.3 CountDown

The count-down parameter in this AVP is used to signal the number of forward frames until the retune instruction is to be actioned by the mobile terminals.

```
CountDown ::=
    INTEGER (0..15)
```

The information is broadcast on several consecutive frames prior to the RNC changing the bearer type or centre frequency of the bearer. The value of zero indicates that the bearer centre frequency or type will change for the next frame.

5.7.31.4 FwdChannelNumber

The parameter *channel-number* specifies the new receive frequency which the UE shall tune to. The data type FwdChannelNumber is defined as follows:

```
FwdChannelNumber ::=
    SEQUENCE {
        offset
            INTEGER (0..1),
        channel-index
```

```
        INTEGER (0..32767)
    }
```

The parameters offset and channel-index are as specified in clauses 5.4.5.1.1 and 5.4.5.1.2, with the exception that the forward frequency is calculated as follows:

forward frequency [MHz] = channel-index × 0,0025 + offset × 0,00125 + 1 510,000

5.7.32 ControlledRandomAccessParam

5.7.32.0 General

This parameter is used to define radio resources in the return direction that may be used for Controlled Random Access purposes. The approach that is taken is to define groups of resources each of which is referenced by a tBCnID value that is specified by the Radio Network Controller and signalled in a Return Schedule. This allows the purpose of the random access resource to be constrained, and also allows control parameters to be transmitted to define the usage.

The ControlledRandomAccessParam AVP applies only to the current return scheduling period (which may be from 160 ms to 640 ms in duration, depending upon the number of slot-plans included in a Return Schedule). A new ControlledRandomAccessParam AVP will be transmitted for each return schedule period.

```
ControlledRandomAccessParam ::=
    SEQUENCE {
    ret-sched-tbcn-id
            TranslatedBearerConnectionID,
    code-rate
            CodeRate,
    rand-param-list
        CHOICE {
            rand-param-list1
                SEQUENCE {
                    prob-access         AccessProbability,
                    priority-access         AccessPriority
                    },
            rand-param-list2
                SEQUENCE {
                    prob-access         AccessProbability,
                    priority-access         AccessPriority,
                    max-power               PowerLevel,
                    min-power               PowerLevel
                    },
            rand-param-list3
                SEQUENCE {
                    prob-access         AccessProbability,
                    priority-access         AccessPriority,
                    max-power               PowerLevel,
                    min-power               PowerLevel,
                    freq-offset-range       FreqOffsetRange
                    },
            rand-param-list4
                SEQUENCE {
                    prob-access         AccessProbability,
                    priority-access         AccessPriority,
                    max-power               PowerLevel,
                    min-power               PowerLevel,
                    freq-offset-range       FreqOffsetRange,
                    timing-offset-range     TimingOffsetRange
                    }
            }
    }
```

As a consequence of the optional information elements, the AVP may be variable length. The most relevant optional information elements are specified first to allow the optional inclusion of these elements without the requirement for presence flags. Illustrations of the shortest and longest variants for the RandomAccessBearer AVP are provided in Figures 5.127 and 5.128. For a definition of CodeRate, refer to clause 5.7.16.2.

(MSB) Bit No (LSB)

8	7	6	5	4	3	2	1	
1	0	0	0	0	0	1	0	Octet 1
ret-sched-tbcn-id								Octet 2
				code-rate				Octet 3
prob-access				priority-access				Octet 4

Figure 5.127: ControlledRandomAccess BCtAVP (Minimum parameter list)

(MSB) Bit No (LSB)

8	7	6	5	4	3	2	1	
1	0	0	0	0	1	0	1	Octet 1
ret-sched-tbcn-id								Octet 2
				code-rate				Octet 3
prob-access				priority-access				Octet 4
max-power				min-power				Octet 5
freq-offset-range								Octet 6
timing-offset-range								Octet 7

Figure 5.128: ControlledRandomAccess BCtAVP (all control parameters shown)

5.7.32.1 AccessProbability

The *AccessProbability* field is used to control the rate at which mobile terminals may access the random access slots. The probability of access may be modified in the range from 1/16 to 1.

```
AccessProbability ::=
    INTEGER (1..16)         -in units of 0.0625
```

A value of 16 corresponds to a probability of 1, which allows the mobile terminal to use the random access slots at any time as required. A value of 1 corresponds to a probability of 1/16, which allows a mobile terminal that needs to transmit to use each time slot with a probability of 1/16.

5.7.32.2 AccessPriority

The *AccessPriority* field is used to control the priority of connections from mobile terminals which may utilize the random access slots identified by this tBCnId. The *Access Priority* field allows 16 levels of priority which are directly mapped from RABAccessPriority defined in ETSI TS 102 744-3-5 [13] and allocated by the RNC during the RAB establishment procedure. The value of 0 has the lowest Access Priority. Common signalling shall use an implicit *AccessPriority* value of 15, while UE specific signalling connections shall use an implicit *AccessPriority* value of 14. The rules for use of Access Priority are defined in ETSI TS 102 744-3-2 [10].

```
AccessPriority ::=
    INTEGER  (0..15)
```

5.7.32.3 PowerLevel

This parameter is used to define the EIRP range to be used by a mobile terminal when transmitting in a particular time-slot. This may be modified dynamically by the RAN. These values are specified as power level values, which represent relative increases in units of dB from the Nominal Reference Level required to close the link, as defined in the Bearer Tables annex to ETSI TS 102 744-2-1 [8]. If the PowerLevel information element is absent then the mobile terminal shall transmit at the nominal reference level for this bearer type, subject to any constraints regarding maximum backoff level for this UE class (i.e. the mobile terminal EIRP shall always be equal to or greater than the minimum EIRP for this bearer type, code rate and UE class).

```
PowerLevel ::=
    INTEGER (0..15) -- in units of 1 dB
```

The parameter controls the minimum or maximum power level increase relative to the nominal Reference Level power level for this bearer in this type of spot beam (Reference Level values for each bearer type are specified in the physical layer chapter of the SDM). The parameter is specified in units of 1 dB.

5.7.32.4 FrequencyOffsetRange

This parameter is used to override the default Frequency Offset Range to be used by a mobile terminal when transmitting in a particular time-slot on a particular channel. This frequency offset range is relative to the centre frequency as defined by the channel number.

```
FreqOffsetRange ::=
    INTEGER (0..255)
-- Controls the maximum frequency offset from the nominal centre frequency value as specified by the
Channel Number.  Specified in units of 10Hz  The default Frequency Offset Range value is 50,
corresponding to a default frequency randomisation of +/- 500Hz.
```

5.7.32.5 Timing Offset Range

This parameter is used to define the relative Timing Offset Range to be used by a mobile terminal when transmitting in a particular time-slot on a particular channel. This timing offset range is relative to the nominal start time for each timeslot. The UE shall randomise over the specified Timing Offset Range when utilizing Controlled Random Access mode.

```
TimingOffsetRange ::=
    INTEGER (0..255)
-- Controls the maximum timing offset from the nominal correct timing value for this slot.
Specified in units of 1 symbol with a symbol in this context being referenced to the T1 bearer type
(33,600 Bd).  The default value is 50, corresponding to 50 symbols of timing offset randomisation.
```

5.7.33 SharedReservationAccessParam

5.7.33.0 General

Shared Reservation Access mode is used for access to LDR bearers when used for radio access bearers supporting streaming services, or for whenever a background or interactive radio access bearer requires capacity for a duration. The Bearer Connection requiring temporary radio resources is mapped to a "return-schedule-tBCnID" by the RNC, where this tBCnId is used solely for the purposes of identifying return radio resources for a specified period and is the value that will be used in the Return Schedule. The mobile terminal when using the allocated resources will always use the connection-specific tBCnID or BCnID in the return burst construct (as required for operation with the signalled bearer type).

```
SharedReservationAccessParam ::=
    SEQUENCE {
        tbcn-id
            TranslatedBearerConnectionID,
        return-schedule-tbcn-id
            TranslatedBearerConnectionID,
        resv-param-list
            CHOICE {
                resv-param-list1
                    SEQUENCE {
                        code-rate1      CodeRate,
                        power-level1    PowerLevel
                    },
                resv-param-list2
```

```
                SEQUENCE {
                    code-rate2  CodeRate,
                    power-level2    PowerLevel,
                    res-duration2   ReservationDuration
                },
            resv-param-list3
                SEQUENCE {
                    code-rate3  CodeRate,
                    power-level3    PowerLevel,
                    res-duration3   ReservationDuration,
                    freq-offset3    FreqOffset
                },
            resv-param-list4
                SEQUENCE {
                    code-rate4  CodeRate,
                    power-level4    PowerLevel,
                    res-duration4   ReservationDuration,
                    freq-offset4    FreqOffset,
                    timing-offset4  TimingOffset
                },
            resv-param-list5
                SEQUENCE {
                    code-rate5  CodeRate,
                    power-level5    PowerLevel,
                    res-duration5   ReservationDuration,
                    freq-offset5    FreqOffset,
                    timing-offset5  TimingOffset,
                    dtx-interval5   DTXInterval
                }
        }
}
```

The AVP may be of variable length (4, 5, 6, 7 or 8 octet payload), as illustrated in Figure 5.129:

(MSB) 8	7	6	5	4	3	2	(LSB) 1	
1	0	0	0	1	0	1	1	Octet 1
tbcn-id								Octet 2
(tbcn-id)				return-schedule-tbcn-id				Octet 3
return-schedule-tbcn-id								Octet 4
code-rate				power-level				Octet 5

Figure 5.129: SharedReservationAccess AVP Descriptor with resv-param-list1

The duration of the association between the connection specific '*tBCnID*' and the '*return-schedule-tBCn-ID*' is specified by the RNC using the '*reservation-duration*' information element in this AVP. For the case above, whereby this BCtAVP only contains the CodeRate and Backoff, the persistence of the association is a single return schedule duration (i.e. the duration of the association applies to the return schedules received in the same forward frame as the SharedReservationAccess AVP). When the association needs to be for a longer duration, the RNC will include a ReservationDuration information element within the BCtAVP, as shown in Figure 5.130.

(MSB) 8	7	6	5	4	3	2	(LSB) 1	
1	0	0	0	1	1	0	0	Octet 1
tbcn-id								Octet 2
								Octet 3
return-schedule-tbcn-id								Octet 4
code-rate				power-level				Octet 5
reservation-duration								Octet 6

Figure 5.130: SharedReservationAccess AVP Descriptor with resv-param-list2

For improved dispersion of the transmitted signals, the RNC may optionally specify a particular frequency offset range for the transmissions from the mobile terminal, as shown in the version of the BCtAVP in Figure 5.131.

(MSB) 8	7	6	5	4	3	2	(LSB) 1	
1	0	0	0	1	1	0	1	Octet 1
tbcn-id								Octet 2
								Octet 3
return-schedule-tbcn-id								Octet 4
code-rate				power-level				Octet 5
reservation-duration								Octet 6
freq-offset								Octet 7

Figure 5.131: SharedReservationAccess AVP Descriptor with resv-param-list3

For optimal dispersion of the transmitted signals, the RNC may optionally specify a particular frequency and timing offset for the transmissions from the mobile terminal, as shown in the version of the BCtAVP in Figure 5.132.

(MSB) 8	7	6	5	4	3	2	(LSB) 1	
1	0	0	0	1	1	1	0	Octet 1
tbcn-id								Octet 2
								Octet 3
return-schedule-tbcn-id								Octet 4
code-rate				power-level				Octet 5
reservation-duration								Octet 6
freq-offset								Octet 7
timing-offset								Octet 8

Figure 5.132: SharedReservationAccess AVP Descriptor with resv-param-list3

To override the default DTX interval, the RNC may specify the DTX interval using the following construct for the BCtAVP, as shown in Figure 5.133.

8 (MSB)	7	6	5	4	3	2	1 (LSB)	
1	0	0	0	1	1	1	1	Octet 1
tbcn-id								Octet 2
tbcn-id (cont.)				return-schedule-tbcn-id				Octet 3
return-schedule-tbcn-id								Octet 4
code-rate				power-level				Octet 5
reservation-duration								Octet 6
freq-offset								Octet 7
timing-offset								Octet 8
dtx-interval								Octet 9

Figure 5.133: SharedReservationAccess AVP Descriptor with resv-param-list4 (multiple return schedule duration with freq & timing offset control & DTX override)

CodeRate is defined in clause 5.7.16.2 and PowerLevel is defined in clause 5.7.32.3.

5.7.33.1 Reservation Duration

This parameter is used to define the duration of the reservation in slot-plans:

```
ReservationDuration ::=
    INTEGER (0..255)        -- in units of slot-plans
```

For example, when the value is 1, the association applies to a single 160 ms duration. When the value is 4 it corresponds to 640 ms duration, and when it is 16 it corresponds to 2 560 ms.

If this parameter is absent, or has a value of 0, the association applies to all slot-plans in any Return Schedules transmitted in the same FEC block as the SharedReservationAccess AVP. In this case, if there are two slot-plans in the Return Schedule, then the Reservation Duration is implicitly 320 ms.

5.7.33.2 FrequencyOffset

This parameter is used to override the default Frequency Offset to be used by a mobile terminal when transmitting in a particular time-slot on a particular channel. This frequency offset is relative to the centre frequency as defined by the channel number.

```
FreqOffset ::=
    INTEGER (-128..127)
-- Controls the frequency offset from the nominal centre frequency value as specified by the Channel
Number.  Specified in units of 10Hz  The default Frequency Offset value is 0.
```

5.7.33.3 Timing Offset

This parameter is used to define the relative Timing Offset to be used by a mobile terminal when transmitting in a particular time-slot on a particular channel. This timing offset is relative to the normal transmit slot position of the UE with its timing corrected in primary or secondary timing mode (i.e. including Self Imposed Delay).

```
TimingOffset ::=
    INTEGER (0..255)
-- Controls the maximum timing offset from the nominal correct timing value for this slot.
Specified in units of 1 symbol with a symbol in this context being referenced to the T1 bearer type
(33,600 Bd).  The default value is 0.
```

5.7.33.4 DTXInterval

This parameter is used to override the default DTX Interval to be used by a mobile terminal.

```
DTXInterval ::=
    INTEGER (0..255)
-- Specified in units of 1 second. The default DTX Interval value is 10.
```

5.7.34 PLMNInfoParam

5.7.34.0 General

This parameter is used to broadcast a PLMN-ID (Public Land Mobile Network Identity) to all UEs. The structure is defined below, with format as shown in Figures 5.134 and 5.135.

```
PLMNInfoParam ::=
    SEQUENCE {
        plmn-identity
            SEQUENCE {
                mcc
                    SEQUENCE SIZE (3) OF Digit,
                mnc
                    SEQUENCE SIZE (3) OF Digit,
            }
        extension
            SEQUENCE {
                plmn-index
                    INTEGER (0..15),
                reserved
                    BIT STRING (SIZE (4))
            } OPTIONAL
    }
```

(MSB) 8	7	6	5 Bit No	4	3	2	(LSB) 1	
1	1	1	1	0	0	1	0	Octet 1
mcc (1st digit)				mcc (2nd digit)				Octet 2
mcc (3rd digit)				mnc (1st digit)				Octet 3
mnc (2nd digit)				mnc (3rd digit)				Octet 4

Figure 5.134: PLMNInfo BCtAVP (single PLMN supported)

(MSB) 8	7	6	5 Bit No	4	3	2	(LSB) 1	
1	1	1	1	0	0	1	1	Octet 1
mcc (1st digit)				mcc (2nd digit)				Octet 2
mcc (3rd digit)				mnc (1st digit)				Octet 3
mnc (2nd digit)				mnc (3rd digit)				Octet 4
plmn-index				reserved				Octet 5

Figure 5.135: PLMNInfo BCtAVP (multiple PLMNs supported)

5.7.34.1 Plmn-identity

The parameter *plmn-identity* shall be encoded as specified in [2], clause 12.1. If the Mobile Network Code (*mnc*) consists of two digits only, then the third digit shall be set to 0xF (i.e. not in the valid range of 0 to 9) and shall be ignored by the UE.

5.7.34.2 Plmn-index

If only one PLMN is supported by the Global Beam Common Channel on which this AVP is broadcast, then the *extension* octet is not present. If more than one PLMN needs to be supported, then one PLMN-ID Bearer Control AVP is transmitted for each PLMN and the *extension* octet is included. In this case, the parameter *plmn-index* shall be set to zero for the first PLMN-ID and then incremented by one for each additional PLMN-ID being broadcast.

5.7.35 SleepModeParam

5.7.35.0 General

This parameter is used to define the sleep-mode operation of the UE. Sleep mode AVPs are only sent in an AVPList SDU which forms part of a BCtSDU-List attached to Adaptation layer signalling. The structure is defined below, with format as shown in Figure 5.136.

```
SleepModeParam ::=
    SEQUENCE {
        idle-count
            INTEGER (0 .. 15),
        logperiod
            INTEGER (0 .. 15),
        start-offset
            INTEGER (0..4095),
        on-period
            INTEGER (0..4095)
    }
```

(MSB) Bit No (LSB)

8	7	6	5	4	3	2	1	
0	0	0	1	0	0	1	1	Octet 1
idle-count				logperiod				Octet 2
< start-offset								Octet 3
>				<				Octet 4
on-period >								Octet 5

Figure 5.136: SleepMode Bearer Control AVP

5.7.35.1 Logperiod

The *logperiod* value is used to specify the duration of the sleep mode period as follows:

$$sleep\_mode\_period = 2^{logperiod}$$

where sleep mode period is counted in frames (each frame is 80 ms duration). Since the frame number increments from 0 through to 2^{12}-1 and then resets to zero, a sleep mode period that is a power of 2 ensures that there are an exact number of sleep mode periods within the frame numbering space.

5.7.35.2 Start-offset

The frame numbers (*m*) of those frames where the UE is ready to receive after an inactive period (start of on-period) is calculated from the value of *start-offset* as follows:

$$m = (start\_offset + n \times sleep\_mode\_period) \text{ MOD } 2^{12}$$

where *n* is an integer between 0 and $2^{(12\text{-logperiod})}$-1.

5.7.35.3 On-period

The *on-period* defines the minimum duration for which the UE shall receive the forward bearer. *On-period* is specified in units of 80 ms frames.

The *off-period* duration can be calculated by subtracting *on-period* from *sleep_mode_period*.

5.7.35.4 Idle-count

Once the UE has left sleep mode in order to transmit BCtPDUs to the RNC , it shall continuously receive the forward bearer for at least the *idle-period*, which is determined as follows:

$$idle\text{-}period = sleep\_mode\_period \times idle\text{-}count$$

Any further signalling activity by the UE during the *idle-period* shall cause the *idle-period* to be restarted. If no activity occurs within the *idle-period*, the UE may re-enter sleep mode operation.

5.7.36 PrimaryBearerParam

5.7.36.0 General

This parameter is used to specify the Primary Shared Access Bearer to be used by UEs located in the referenced spot-beam. The structure is defined below, with format as shown in Figures 5.137 and 5.138.

```
PrimaryBearerParam ::=
    SEQUENCE {
        spotbeam-id
            SpotBeamID,
        channel-number
            FwdChannelNumber,
        plmn-index-included
            BOOLEAN,
        f-bearer
            FBearer,
        f-bearer-type
            FBearerType,
        extension
            SEQUENCE {
                plmn-index
                    INTEGER (0..15),
                reserved
                    BIT STRING (SIZE (4))
            } OPTIONAL,
        class-control
            CHOICE{
                cc-8
                    BIT STRING (SIZE  (8)),
                cc-16
                    BIT STRING (SIZE (16)),
                cc-24
                    BIT STRING (SIZE (24))
        } OPTIONAL
    }
```

(MSB) 8	7	6	Bit No 5	4	3	2	(LSB) 1	
0	0	0	1	1	1	0	0	Octet 1
spotbeam-id								Octet 2
channel								Octet 3
number								Octet 4
0	f-bearer				f-bearer-type			Octet 5
class-control (class 8..1)								Octet 6

Figure 5.137: PrimaryBearer BCtAVP (with 8 bit *class-control* field, *plmn-index* not included)

(MSB)			Bit No				(LSB)	
8	7	6	5	4	3	2	1	
0	0	0	1	1	1	1	0	Octet 1
spotbeam-id								Octet 2
channel								Octet 3
number								Octet 4
1	f-bearer			f-bearer-type				Octet 5
plmn-index				reserved				Octet 6
class-control (class 8..1)								Octet 7
class-control (class 16..9)								Octet 8

Figure 5.138: PrimaryBearer Bearer Control AVP (with 16 bit *class-control* field and *plmn-index* included)

The data type SpotBeamID is defined in clause 5.4.4.7 while the data type FwdChannelNumber is defined in clause 5.7.32.5.

5.7.36.1 F-bearer and F-Bearer Type

The *f-bearer-type* field specifies the type of forward bearer of the specified Primary Bearer. The data type FBearerType is defined in clause 5.7.31.2.

The field *f-bearer* and its purpose is defined in clause 5.4.4.5.

5.7.36.2 Class-control

The *class-control* field is a bitmap that indicates which classes of UEs may use the Primary Bearer that is specified in this AVP. A UE may use the specified bearer if the corresponding bit is set in the bitmap. The size of the bitmap may be either 8, 16 or 24 bits and can be determined from the *prmlen* field in the AVP header and whether the *extension* octet is present. If the size of the *class-control* field is 16 bits, then the first octet shall relate to UE classes 8 through to 1 and the second octet shall relate to UE classes 16 through to 9. If the size of *class-control* is 24 bits then the third octet shall relate to UE classes 24 through to 17. If the *class-control* field is present in any form, the default interpretation for any octet of the bitmap which is not signalled shall be 0x00.

5.7.36.3 Plmn-index

Optionally, the parameter *plmn-index* may be present in the Primary Bearer AVP. In this case, a UE shall only select the advertized Primary Bearer if the *plmn-index* in this AVP is identical to the *plmn-index* included in the PLMNInfo AVP (see clause 5.7.34) that relates to the PLMN selected by the UE.

5.7.37 SatelliteStateVectorsParam

5.7.37.0 General

This parameter describes the exact position of the satellite in relation to its nominal position and allows the UE to calculate the round trip time to and from the satellite. The structure is defined below, with format as shown in Figure 5.139.

```
SatelliteStateVectorsParam ::=
    SEQUENCE {
        x-vector
            INTEGER (-1024..+1023),
        y-vector
            INTEGER (-1024..+1023),
        z-vector
            INTEGER (-8192..+8191),
        interval
            INTEGER (0..15)
    }
```

(MSB)			Bit No				(LSB)	
8	7	6	5	4	3	2	1	
1	0	1	0	1	1	0	0	Octet 1
<			x-vector					Octet 2
		>	<					Octet 3
			y-vector			>	<	Octet 4
			z-vector					Octet 5
			>	interval				Octet 6

Figure 5.139: SatelliteStateVectors Bearer Control AVP

5.7.37.1 Interval

The *interval* value is used to specify the duration of the interval until the transmission of the next set of state vectors in units of 512 frames of 80 ms duration. This field is used to allow UEs entering sleep mode to be switched on in time to receive a new state vector AVP.

5.7.37.2 X-Vector, Y-Vector, Z-Vector

The satellite state vectors are given by the parameters *x-vector*, *y-vector* and *z-vector* in units of 488 m. These parameters are represented as a signed (two's-complement) integer.

5.7.38 Type0QoSParam

5.7.38.0 General

This parameter carries data used to inform the UE of the return QoS information used to control both connection and Bearer Control Layer behaviour for a specific connection. Type0QoS AVPs are only sent in an AVPList SDU which forms part of a BCtSDU-List attached to Connection layer signalling. Other Connection QoS AVPs may be defined in future. The structure is defined below, with format as shown in Figure 5.140.

```
Type0QoSParam ::=
    SEQUENCE {
        return-target-latency
            INTEGER (0 .. 4095),
        return-discard-latency
            INTEGER (0 .. 4095),
        return-mean-rate
            INTEGER (0 .. 4095),
        return-peak-rate
            INTEGER (0 .. 4095)
    }
```

(MSB)			Bit No				(LSB)	
8	7	6	5	4	3	2	1	
0	0	0	0	0	1	0	1	Octet 1
<			return-target-latency					Octet 2
			>	<				Octet 3
			return-discard-latency				>	Octet 4
<			return-mean-rate					Octet 5
			>	<				Octet 6
			return-peak-rate				>	Octet 7

Figure 5.140: Typ0QoS BCtAVP

5.7.38.1 Return-target-latency

The parameter *return-target-latency* is used to specify the return target latency (RetL) for the connection being established. RetL is derived as follows:

RetL ::= *return-target-latency* × 80 ms.

The default value for *return-target-latency* shall be 30, resulting in a default value of RetL of 2,4 seconds.

5.7.38.2 Return-discard-latency

The parameter *return-discard-latency* is used to specify the return discard latency (RetDiscardL) for the connection being established. RetDiscardL is derived as follows:

If (*return-discard-latency* > 0)

RetDiscardL ::= *return-discard-latency* × 20 ms

Else

RetDiscardL ::= Infinity

The default value for *return-discard-latency* shall be 3 000, resulting in a default value of RetDiscardL of 60 seconds. A value of 0 in *return-discard-latency* indicates that the data should never be discarded.

5.7.38.3 Return-mean-rate

The parameter *return-mean-rate* is used to define the mean rate (the rate at which this connection will nominally be serviced) in the return direction for the connection being established. The Return Mean Rate (RetMeanRate) is derived as follows:

RetMeanRate ::= *return-mean-rate* × 32 bytes/second

The default value for *return-mean-rate* is 20, resulting in a RetMeanRate value of 640 bytes/second or 5 120 bits/second.

5.7.38.4 Return-peak-rate

The parameter *return-peak-rate* is used to define the peak rate (the maximum rate at which this connection will be serviced) in the return direction for the connection being established. The Return Peak Rate (RetPeakRate) is derived as follows:

RetPeakRate ::= *return-peak-rate* × 32 bytes/second

The default value for *return-peak-rate* is 80, resulting in a RetPeakRate value of 2 560 bytes/second or 20 480 bits/second.

5.7.39 LeapSecondParam

This parameter is provided to allow UEs to calculate true UTC time from the GPS UTC time provided by the UTCDateAndTime AVP. The parameter is defined below and its structure is shown in Figure 5.141.

```
LeapSecondParam ::=
    INTEGER (-128..127)
```

(MSB) 8	7	6	5	Bit No 4	3	2	(LSB) 1	
1	0	1	0	1	0	0	0	Octet 1
LeapSecond								Octet 2

Figure 5.141: LeapSecond BCtAVP

5.7.40 NASSystemInfoParam

This parameter is used to carry the Information Elements (IEs) required by the Non-Access-Stratum (NAS) Mobility Management entities of the UE. The contents are not interpreted by the Bearer Control Layer but passed directly to the Adaptation Layer, hence the type definition within the Bearer Control Layer is as follows.

```
NASSystemInfoParam ::=
    OCTET STRING (SIZE (1..8))
```

The information elements used within the Non Access Stratum at the UE are mapped onto the structure as shown in ETSI TS 124 008 [4], clause 10.5.1.12, and are illustrated in Figure 5.142. For a definition of the data types and its purpose refer to ETSI TS 124 008 [4], clause 10.5.1.12.

(MSB)			Bit No				(LSB)	
8	7	6	5	4	3	2	1	
1	1	1	1	1	1	0	1	Octet 1
LAC								Octet 2
								Octet 3
T3212								Octet 4
spare							ATT	Octet 5
RAC								Octet 6
spare							NMO	Octet 7

Figure 5.142: NAS-SystemInfo BCtAVP (with mapping of NAS Information elements)

5.7.41 UTCDateAndTimeParam

This parameter specifies the GPS UTC date and time the forward bearer transmitted a frame with frame number zero (as indicated in the Bulletin Board SDU, see clause 5.4.4.3). It is used to broadcast the current time to UEs in order to assist GPS acquisition.

The GPS time transmitted in this AVP differs from UTC by the integer LEAP seconds (the value of LEAP was 14 on 1 January 2006 and is likely to change in future). The value broadcast by the RNC shall be the GPS time, and hence can be used directly by the UE without correction for the unknown value LEAP.

The structure is defined below, with format as shown in Figure 5.143.

```
UTCDateAndTimeParam ::=
    SEQUENCE {
        year
            INTEGER (2000..2127),
        month
            INTEGER (0..15),
        day
            INTEGER (0..31),
        hours
            INTEGER (0..31),
        minutes
            INTEGER (0..63),
        seconds
            INTEGER (0..63),
        ten-milli-seconds
            INTEGER (0..127)
    }
```

(MSB) Bit No (LSB)

8	7	6	5	4	3	2	1	
1	1	0	1	0	1	0	0	Octet 1
year							<	Octet 2
month >			day					Octet 3
hours					< minutes			Octet 4
minutes (cont) >			< seconds					Octet 5
>	ten-milli-seconds							Octet 6

Figure 5.143: UTCDateAndTime BCtAVP

5.7.42 TMPayloadPositionParam

5.7.42.0 General

This parameter is used within an AVPList SDU within a connection-specific BCtPDU to specify the position of the next Transparent Mode BCnPDU (BCtPDU payload), enabling the UE to recover the next BCnPDU for that connection, even if the CRC fails. The AVP is only present if the connection supports the delivery of erroneous PDUs. The structure is defined below, with format as shown in Figure 5.144.

```
TMPayloadPositionParam ::=
    SEQUENCE {
        reserved
            BIT STRING (SIZE (3)),
        fec-block-num
            INTEGER (0..7),
        start-pos
            INTEGER (0..1023)
    }
```

(MSB) Bit No (LSB)

8	7	6	5	4	3	2	1	
1	1	0	1	1	0	0	1	Octet 1
reserved		fec-block-num			<			Octet 2
start-pos							>	Octet 3

Figure 5.144: TMPayloadPosition BCtAVP

5.7.42.1 Fec-block-num

This parameter specifies the number of the FEC block containing the next BCnPDU for the connection, where *fec-block-num* = 0 refers to the first block in the frame, regardless of the FEC block in which the AVP is transmitted.

5.7.42.2 Start-pos

This parameter specifies the number of the first octet of the Transparent Mode BCnPDU within the FEC block for the connection where *start-pos* = 0 refers to the first octet within the FEC block.

5.7.43 GPSPolicyInfoParam

The GPSPolicyInfoParam provides information regarding the GPS Policy as it is enforced by the RNC in the beam (Location Area) in which it is broadcasted and allows the UE to optimize its GPS acquisition strategy. The parameter has the following structure, with format as shown in Figure 5.145:

```
GPSPolicyInfoParam ::=
    SEQUENCE {
        position-quality
            PositionQuality,
```

```
        gps-display-enable
            GPSDisplayEnable,
        position-accuracy
            PositionAccuracy,
        loa
            LOA,
        position-age
            PositionAge
    }
```

(MSB) 8	7	6	Bit No 5	4	3	2	(LSB) 1	
1	0	1	0	1	0	1	1	Octet 1
position-quality			gde	pos-accuracy				Octet 2
loa								Octet 3
position-age								Octet 4
								Octet 5

Figure 5.145: GPSPolicyInfo BCtAVP

The parameter *position-quality* specifies the type of position information which the RNC expects during the registration process. The data type PositionQuality is defined as follows:

```
PositionQuality ::=
    INTEGER {
        reserved(0),
        gps-fix-or-irs-fix(1),
        gps-fix-or-irs-fix or-user-specified(2),
        gps-fix-or-irs-fix-or-spot-beam-id(3),
        spot-beam-id-only(4),
        any(5)
    } (0..7)        -- type of position info required
                    -- at Registration time
```

The flag *gps-display-enable* (abbreviated as gde in Figure 5.145) indicates if TRUE that it is permitted to make the GPS coordinates accessible to the user in the current spot beam (Location Area). The type GPSDisplayEnable is defined as follows:

```
GPSDisplayEnable ::=
    BOOLEAN
```

The parameter *position-accuracy* specifies the minimum number of GPS satellites that shall be used to obtain a position fix:

```
PositionAccuracy ::=
    INTEGER (0..15)
    -- minimum number of satellites for GPS fix
```

The parameter *loa* (loss of acquisition) specifies the maximum time in minutes for which the signal from the Forward Bearer received by the UE may be lost.

```
LOA ::=
    INTEGER (0..255)
    -- maximum LOA time in minutes
```

The parameter *position-age* specifies the maximum time in minutes for which the last GPS fix will be accepted by the RNC:

```
PositionAge ::=
    INTEGER (0..65535)
    -- maximum fix age in minutes
```

5.7.44 ReturnLinkReferenceLevelSetParam (Extension Class UE Only)

This parameter is used within a SpecificAVPList or AVPList SDU to set the reference levels for the return link adaptation for a set of bearer types for a particular UE Hardware Unit. It also includes a transmit power backoff value or a backoff range over which the UE can trade-off transmit EIRP versus coding rate. The particular UE Hardware Unit that the AVP is intended for is determined by the translated Bearer Connection ID contained in the SpecificAVPList SDU or the translated Bearer Connection ID of the BCtPDU used to carry the AVPList SDU. On occasion it may be necessary to send multiple instances of this parameter with different values of *bearer-set* to ensure all bearer types being used by the translated Bearer Connection are represented. The structure is defined below, with format as shown in Figure 5.146.

```
ReturnLinkReferenceLevelSetParam ::=
    SEQUENCE {
        bearer-set
            INTEGER (0..3),
        reference-level
            ControlIndex,
        backoff
            INTEGER (0..15),
        mode
            INTEGER (0..3),
        reference-offsets
            SEQUENCE SIZE (10) OF ReferenceOffset
    }

ReferenceOffset ::= INTEGER (0..31)
```

<table>
<tr><td>(MSB)</td><td></td><td></td><td colspan="2">Bit No</td><td></td><td></td><td>(LSB)</td><td></td></tr>
<tr><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td></td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>Octet 1</td></tr>
<tr><td colspan="2">bearer-set</td><td colspan="6">reference-level</td><td>Octet 2</td></tr>
<tr><td colspan="4">backoff</td><td colspan="2">mode</td><td colspan="2">< offset-0</td><td>Octet 3</td></tr>
<tr><td colspan="3">offset-0 ></td><td colspan="5">< offset-1 ></td><td>Octet 4</td></tr>
<tr><td colspan="5">< offset-2 ></td><td colspan="3">< offset-3</td><td>Octet 5</td></tr>
<tr><td colspan="2">offset-3 ></td><td colspan="5">< offset-4 ></td><td><</td><td>Octet 6</td></tr>
<tr><td colspan="4">offset-5 ></td><td colspan="4">< offset -6</td><td>Octet 7</td></tr>
<tr><td>></td><td colspan="5">< offset-7 ></td><td colspan="2">< offset-8</td><td>Octet 8</td></tr>
<tr><td colspan="3">offset-8 ></td><td colspan="5">< offset-9 ></td><td>Octet 9</td></tr>
</table>

Figure 5.146: ReturnLinkReferenceLevelSet BCtAVP

The parameter *reference-level* is of data type ControlIndex which is specified in clause 5.4.12.6. UE classes that support wrap around of control index shall interpret the received reference-level value according to the following rule:

If (reference-level < 16) then
reference-level = reference-level + 64
EndIf

The parameter *backoff* shall be used in different ways depending on the value of *mode*, as shown in Table 5.21.

When any Bearer Type within the bearer-set is not applicable for a UE class the value of the parameter ReferenceOffset will be set to 31.

5.7.45 InitialReferenceLevelSetParam (Extension Class UE Only)

This parameter is used within a SpecificAVPList or AVPList SDU to change the initial reference levels for the return link adaptation for a set of bearer types for a particular UE Hardware Unit without causing the current reference levels to be changed. It has the same structure as the *ReturnLinkReferenceLevelSetParam*. The structure is defined below, with format as shown in Figure 5.147.

```
InitialReferenceLevelSetParam ::=
    SEQUENCE {
        bearer-set
            INTEGER (0..3),
        reference-level
            ControlIndex,
        backoff
            INTEGER (0..15),
        mode
            INTEGER (0..3),
        reference-offsets
            SEQUENCE SIZE (10) OF ReferenceOffset
    }

ReferenceOffset ::= INTEGER (0..31)
```

<table>
<tr><th>(MSB)</th><th colspan="6">Bit No</th><th>(LSB)</th><th></th></tr>
<tr><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td></td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td><td>Octet 1</td></tr>
<tr><td colspan="2">bearer-set</td><td colspan="6">reference-level</td><td>Octet 2</td></tr>
<tr><td colspan="4">backoff</td><td colspan="2">mode</td><td colspan="2">< offset-0</td><td>Octet 3</td></tr>
<tr><td colspan="3">offset-0 ></td><td colspan="5">< offset-1 ></td><td>Octet 4</td></tr>
<tr><td colspan="5">< offset-2 ></td><td colspan="3">< offset-3</td><td>Octet 5</td></tr>
<tr><td colspan="2">offset-3 ></td><td colspan="5">< offset-4 ></td><td><</td><td>Octet 6</td></tr>
<tr><td colspan="4">offset-5 ></td><td colspan="4">< offset -6</td><td>Octet 7</td></tr>
<tr><td>></td><td colspan="5">< offset-7 ></td><td colspan="2">< offset-8</td><td>Octet 8</td></tr>
<tr><td colspan="3">offset-8 ></td><td colspan="5">< offset-9 ></td><td>Octet 9</td></tr>
</table>

Figure 5.147: InitialLinkReferenceLevelSet BCtAVP

5.7.46 BeamInfoParam

This AVP provides the UE with the type and identity of the current satellite beam (i.e. the beam in which the current forward bearer is being transmitted). The structure is defined below, with format as shown in Figure 5.148.

```
BeamInfoParam ::=
    SEQUENCE {
        reserved
            BIT STRING (SIZE(5)),
        satellite-beam-type
            INTEGER {
                global(0),
                regional(1),
                narrow(2)
            } (0..7),
        current-spot-beam-id
            SpotBeamID
    }
```

<table>
<tr><th>(MSB)</th><th colspan="6">Bit No</th><th>(LSB)</th><th></th></tr>
<tr><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td></td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>Octet 1</td></tr>
<tr><td colspan="5">reserved</td><td colspan="3">sat-beam-type</td><td>Octet 2</td></tr>
<tr><td colspan="8">current-spot-beam-id</td><td>Octet 3</td></tr>
</table>

Figure 5.148: BeamInfo BCtAVP

5.7.47 ForwardCarrierLossParam (Extension Class UE Only)

This parameter is used to define the forward carrier synchronization loss timer duration of the UE. Forward Carrier Loss AVPs are only sent in an AVPList SDU which forms part of a BCtSDU-List attached to Adaptation layer signalling. The structure is defined below, with format as shown in Figure 5.149.

```
ForwardCarrierLossParam ::=
    SEQUENCE {
        sync-loss-timer
            INTEGER (1 .. 256)  }
```

8 (MSB)	7	6	5	4	3	2	1 (LSB)	
0	1	1	0	0	0	0	0	Octet 1
sync-loss-timer								Octet 2

Figure 5.149: ForwardCarrierLoss Bearer Control AVP

sync-loss-timer is in units of seconds. If this AVP is absent the default value defined in ETSI TS 102 744-3-2 [10] is used.

5.7.48 MaxReturnCodeRateParam (Extension Class UE Only)

This parameter is used within a SpecificAVPList or AVPList SDU to set the maximum code rate the UE can use on each bearer type. The particular UE Hardware Unit that the AVP is intended for is determined by the translated Bearer Connection ID contained in the SpecificAVPList SDU or the translated Bearer Connection ID of the BCtPDU used to carry the AVPList SDU. On occasion it may be necessary to send multiple instances of this parameter with different values of *bearer-set* to ensure all bearer types being used by the translated Bearer Connection are represented. The structure is defined below, with format as shown in Figure 5.150.

```
MaxReturnCodeRateParam ::=
    SEQUENCE {
        bearer-set
            INTEGER (0..3),
        reserved
            BIT STRING (SIZE(6)),
        max-code-rates
            SEQUENCE SIZE (10) OF CodeRate
    }

CodeRate ::= INTEGER (-8 .. 7)
```

8 (MSB)	7	6	5	4	3	2	1 (LSB)	
1	0	1	1	1	1	0	1	Octet 1
bearer-set		reserved						Octet 2
code-rate-0				code-rate-1				Octet 3
code-rate-2				code-rate-3				Octet 4
code-rate-4				code-rate-5				Octet 5
code-rate-6				code-rate-7				Octet 6
code-rate-8				code-rate-9				Octet 7

Figure 5.150: MaxReturnCodeRate Bearer Control AVP

5.7.49 InitialReferenceLevelAndMaxCodeRateParam

This parameter is used within a SpecificAVPList or AVPList SDU to set the initial reference levels and maximum return code rate used for the return link adaptation for two bearer types for a particular UE Hardware Unit. The return bearer types to which the offset and maximum code rate values are applicable are included in the message as ReturnBearerTypeFull immediately preceding each offset/code-rate pair.

It also includes a transmit power backoff value or a backoff range over which the UE can trade-off transmit EIRP versus coding rate. The particular UE Hardware Unit that the AVP is intended for is determined by the translated Bearer Connection ID contained in the SpecificAVPList SDU or the translated Bearer Connection ID of the BCtPDU used to carry the AVPList SDU.

The parameter *backoff* shall be used in different ways depending on the value of *mode*, as defined in Table 5.21. The structure is defined below, with format as shown in Figure 5.151.

```
InitialReferenceLevelAndMaxCodeRateParam ::=
    SEQUENCE {
        reference-level
            ControlIndex,
        reserved
            BIT STRING (SIZE(2)),
        backoff
            INTEGER (0..15),
        mode
            INTEGER (0..3),
        r-bearer-type-0
            ReturnBearerTypeFull,
        offset-0
            ReferenceOffset,
        code-rate-0
            CodeRate,
        r-bearer-type-1
            ReturnBearerTypeFull,
        offset-1
            ReferenceOffset,
        code-rate-1
            CodeRate
    }

ReferenceOffset ::= INTEGER (0..31)

CodeRate ::= INTEGER (-8 .. 7)
```

<table>
<tr><td colspan="8">(MSB) Bit No (LSB)</td><td></td></tr>
<tr><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td></td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td><td>Octet 1</td></tr>
<tr><td colspan="6">reference-level</td><td colspan="2">reserved</td><td>Octet 2</td></tr>
<tr><td colspan="4">backoff</td><td colspan="2">mode</td><td colspan="2">< r-bearer</td><td>Octet 3</td></tr>
<tr><td colspan="2">-type-0 ></td><td colspan="5">offset-0</td><td><</td><td>Octet 4</td></tr>
<tr><td colspan="3">code-rate-0 ></td><td colspan="4">r-bearer-type-1</td><td><</td><td>Octet 5</td></tr>
<tr><td colspan="4">offset-1 ></td><td colspan="4">code-rate-1</td><td>Octet 6</td></tr>
</table>

Figure 5.151: InitialReferenceLevelAndMaxCodeRate Bearer Control AVP

5.7.50 RandomisedInitialAccessDelayParam

This parameter is used within a broadcast avp-list to signal a time period over which the UE shall randomise its initial registration attempts. This AVP would typically be broadcast immediately after a site switch, to smooth out the rate at which UEs are allowed to reconnect to the network. The structure is defined below, with format as shown in Figure 5.152.

```
RandomisedInitialAccessDelayParam ::=
    SEQUENCE {
        initial-fixed-delay
```

```
        INTEGER (0..255),
    scaling
        INTEGER (0..3),
    randomised-delay-interval
        INTEGER (0..63)
}
```

(MSB) 8	7	6	5	Bit No 4	3	2	(LSB) 1	
1	1	1	0	1	0	0	1	Octet 1
initial-fixed-delay								Octet 2
scaling		randomised-delay-interval						Octet 3

Figure 5.152: RandomisedInitialAccessDelay Bearer Control AVP

initial-fixed-delay is in units of 10 seconds, giving a possible range of 0 to 2 550 seconds in steps of 10 seconds. Units for *randomised-delay-interval* depend on the value of scaling, as shown in Table 5.22.

Table 5.22: RandomisedInitialAccessDelay Scaling values: Expected UE Behaviour

Scaling	Units for randomised-delay-interval	Possible range
0	seconds	0 to 63 s
1	10 seconds	0 to 630 s
2	30 seconds	0 to 1 890 s
3	60 seconds	0 to 3 780 s

Upon the expiry of *initial-fixed-delay*, the UE is required to select a random time within the *randomised-delay-interval* specified, after which it is permitted to attempt registration as per normal.

5.7.51 IntermodTestInfoParam

This parameter is used to broadcast two transmit and one receive frequency to be used by Aeronautical class UEs for intermodulation test purposes. The structure is defined below, with format as shown in Figure 5.153.

```
IntermodTestInfoParam ::=
    SEQUENCE {
        tx-frequency-one
            RetChannelNumber,
        tx-frequency-two
            RetChannelNumber,
        rx-frequency
            FwdChannelNumber
            }
```

(MSB) 8	7	6	5	Bit No 4	3	2	(LSB) 1	
1	1	1	0	1	0	0	1	Octet 1
tx-frequency-one								Octet 2
								Octet 3
tx-frequency-two								Octet 4
								Octet 5
rx-frequency								Octet 6
								Octet 7

Figure 5.153: IntermodTestInfo Bearer Control AVP

Annex A (normative): ASN.1

This annex collates the data structures in ASN.1 notation from the present document in alphabetical order, in a format that may be used in a program code compiler.

The code is reproduced in a text file that is contained in archive ts_1027440301v010101p0.zip which accompanies the present document.

History

Document history		
V1.1.1	October 2015	Publication