# ETSI EN 301 790 V1.5.1 (2009-05)

*European Standard (Telecommunications series)*

## Digital Video Broadcasting (DVB);
## Interaction channel for satellite distribution systems

European Broadcasting Union    Union Européenne de Radio-Télévision

EBU·UER

Reference

REN/JTC-DVB-230

Keywords

broadcasting, DVB, interaction, satellite

*ETSI*

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

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

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

*Important notice*

Individual copies of the present document can be downloaded from:
http://www.etsi.org

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

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

If you find errors in the present document, please send your comment to one of the following services:
http://portal.etsi.org/chaircor/ETSI_support.asp

*Copyright Notification*

*ETSI*

# Contents

Intellectual Property Rights ............................................................................................................................8

Foreword............................................................................................................................................................8

1        Scope ......................................................................................................................................................9

2        References ..............................................................................................................................................9
2.1          Normative references .....................................................................................................................................9
2.2          Informative references ..................................................................................................................................11

3        Definitions, symbols and abbreviations ...............................................................................................11
3.1          Definitions....................................................................................................................................................11
3.2          Symbols ........................................................................................................................................................12
3.3          Abbreviations ...............................................................................................................................................12

4        Reference models for satellite interactive networks in DVB ...............................................................14
4.1          Protocol stack model ....................................................................................................................................14
4.2          System model ...............................................................................................................................................15
4.3          Reference model of the Satellite Interactive Network ..................................................................................16
4.4          Dynamic connectivity....................................................................................................................................18

5        Forward link .........................................................................................................................................19
5.1          Spectrum spreading in the forward link (Mobile Opt) .................................................................................19

6        Return link base-band physical layer specification and multiple access definition ............................20
6.1          RCST synchronization ..................................................................................................................................21
6.1.1              Timing control .........................................................................................................................................21
6.1.2              Carrier synchronization............................................................................................................................22
6.1.3              Burst synchronization ..............................................................................................................................22
6.1.4              Symbol clock synchronization.................................................................................................................22
6.2          Burst format..................................................................................................................................................22
6.2.1              Traffic (TRF) burst formats .....................................................................................................................22
6.2.1.1                 ATM TRF burst .................................................................................................................................23
6.2.1.2                 Optional MPEG2-TS TRF burst .......................................................................................................23
6.2.2              Synchronization and acquisition burst formats .......................................................................................23
6.2.2.1                 Synchronization (SYNC) burst format..............................................................................................24
6.2.2.2                 Acquisition (ACQ) burst ...................................................................................................................24
6.2.3              Common Signalling Channel (CSC) burst format ...................................................................................25
6.2.4              Bit numbering and interpretation .............................................................................................................27
6.2.5              Transmission order ..................................................................................................................................27
6.3          Randomization for energy dispersal .............................................................................................................28
6.4          Coding ..........................................................................................................................................................28
6.4.1              CRC error detection code ........................................................................................................................28
6.4.2              Reed-Solomon outer coding ....................................................................................................................29
6.4.3              Convolutional inner coding .....................................................................................................................29
6.4.4              Turbo code................................................................................................................................................30
6.4.4.1                 Description of the turbo code permutation........................................................................................31
6.4.4.2                 Determination of the circulation states.............................................................................................32
6.4.4.3                 Rates and puncturing map .................................................................................................................32
6.4.4.4                 Order of transmission and mapping to QPSK constellation..............................................................33
6.4.5              Countermeasures for non-line-of-sight channels (Mobile Opt) ..............................................................34
6.4.5.1                 LL-FEC Frame ..................................................................................................................................35
6.4.5.1.1                     Filling of Application Data Table................................................................................................35
6.4.5.1.2                     Generation of FEC Data Table ...................................................................................................36
6.4.5.1.2.1                         Reed-Solomon code............................................................................................................36
6.4.5.1.2.2                         Raptor code.........................................................................................................................37
6.4.5.2                 Carriage of LL-FEC frames ..............................................................................................................37
6.4.5.3                 Carriage of application data in transport streams .............................................................................37
6.4.5.4                 Carriage of parity data in transport streams .....................................................................................38
6.4.5.4.1                     Carriage of Reed-Solomon code parity data...............................................................................38

*ETSI*

# 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://webapp.etsi.org/IPR/home.asp).

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 European Standard (Telecommunications series) has been produced by Joint Technical Committee (JTC) Broadcast of the European Broadcasting Union (EBU), Comité Européen de Normalisation ELECtrotechnique (CENELEC) and the European Telecommunications Standards Institute (ETSI).

NOTE: The EBU/ETSI JTC Broadcast was established in 1990 to co-ordinate the drafting of standards in the specific field of broadcasting and related fields. Since 1995 the JTC Broadcast became a tripartite body by including in the Memorandum of Understanding also CENELEC, which is responsible for the standardization of radio and television receivers. The EBU is a professional association of broadcasting organizations whose work includes the co-ordination of its members" activities in the technical, legal, programme-making and programme-exchange domains. The EBU has active members in about 60 countries in the European broadcasting area; its headquarters is in Geneva.

European Broadcasting Union
CH-1218 GRAND SACONNEX (Geneva)
Switzerland
Tel:　+41 22 717 21 11
Fax:　+41 22 717 24 81

Founded in September 1993, the DVB Project is a market-led consortium of public and private sector organizations in the television industry. Its aim is to establish the framework for the introduction of MPEG-2 based digital television services. Now comprising over 200 organizations from more than 25 countries around the world, DVB fosters market-led systems, which meet the real needs, and economic circumstances, of the consumer electronics and the broadcast industry.

| National transposition dates | |
|---|---|
| Date of adoption of this EN: | 11 May 2009 |
| Date of latest announcement of this EN (doa): | 31 August 2009 |
| Date of latest publication of new National Standard or endorsement of this EN (dop/e): | 28 February 2010 |
| Date of withdrawal of any conflicting National Standard (dow): | 28 February 2010 |

# 1      Scope

The present document forms the specification for the provision of the interaction channel for GEO satellite interactive networks with fixed Return Channel Satellite Terminals (RCST). The present document facilitates the use of RCSTs for individual or collective installation (e.g. SMATV) in a domestic environment. It also supports the connection of such terminals with in-house data networks. The present document may be applied to all frequency bands allocated to GEO satellite services.

The solutions provided for interaction channel for satellite interactive networks are a part of a wide set of alternatives to implement interactive services through Digital Video Broadcasting (DVB) systems.

The revision accomplished in 2002 provides the means to extend the applicability of the standard to regenerative satellite systems. This revision also allows for reduction in terminal costs without significantly impacting the performance.

The revision accomplished in 2004 integrates the DVB-S2 standard for forward link transmission. DVB-S2 is the second generation standard for satellite transmission, which provides higher power and bandwidth efficiency as well as adaptive coding and modulation.

The revision accomplished in 2008 provides the means for comprehensive support of mobile and nomadic terminals, including techniques to overcome the transmission channel impairments experienced by mobile terminals and the necessary mechanisms to control and mange mobile terminals.

# 2      References

References are either specific (identified by date of publication and/or edition number or version number) or non-specific.

- For a specific reference, subsequent revisions do not apply.

- Non-specific reference may be made only to a complete document or a part thereof and only in the following cases:

  - if it is accepted that it will be possible to use all future changes of the referenced document for the purposes of the referring document;

  - for informative references.

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

For online referenced documents, information sufficient to identify and locate the source shall be provided. Preferably, the primary source of the referenced document should be cited, in order to ensure traceability. Furthermore, the reference should, as far as possible, remain valid for the expected life of the document. The reference shall include the method of access to the referenced document and the full network address, with the same punctuation and use of upper case and lower case letters.

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

## 2.1      Normative references

The following referenced documents are indispensable for the application of the present document. For dated references, only the edition cited applies. For non-specific references, the latest edition of the referenced document (including any amendments) applies.

[1]            ETSI EN 300 421: "Digital Video Broadcasting (DVB); Framing structure, channel coding and modulation for 11/12 GHz satellite services".

[2]        ETSI ETS 300 802: "Digital Video Broadcasting (DVB); Network-independent protocols for DVB interactive services".

[3]        ETSI EN 300 468: "Digital Video Broadcasting (DVB); Specification for Service Information (SI) in DVB systems".

[4]        ETSI EN 301 192: "Digital Video Broadcasting (DVB); DVB specification for data broadcasting".

[5]        ETSI EN 301 459: "Satellite Earth Stations and Systems (SES); Harmonized EN for Satellite Interactive Terminals (SIT) and Satellite User Terminals (SUT) transmitting towards satellites in geostationary orbit in the 29,5 GHz to 30,0 GHz frequency bands covering essential requirements under article 3.2 of the R&TTE Directive".

[6]        IETF RFC 2684 (1999): "Multiprotocol Encapsulation over ATM Adaptation Layer 5".

[7]        ISO/IEC 13818-1 (1996): "Information technology - Generic coding of moving pictures and associated audio information: Systems".

[8]        ITU-T Recommendation Q.2931 (1995): "Digital Subscriber Signalling System No. 2 - User-Network Interface (UNI) layer 3 specification for basic call/connection control".

[9]        IEEE 802.3: "Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements - Part 3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications".

[10]       ITU-T Recommendation I.432 (all parts): "B-ISDN user-network interface - Physical layer specification".

[11]       ETSI ES 200 800: "Digital Video Broadcasting (DVB); DVB interaction channel for Cable TV distribution systems (CATV)".

[12]       IETF RFC 2104 (1997): "HMAC: Keyed-Hashing for Message Authentication".

[13]       ANSI/IEEE 754 (1985): "IEEE Standard for Binary Floating-Point Arithmetic".

[14]       ISO/IEC 13818-6 (1998): "Information technology - Generic coding of moving pictures and associated audio information - Part 6: Extensions for DSM-CC".

[15]       ITU-T Recommendation I.363-5 (1996): "B-ISDN ATM Adaptation Layer specification: Type 5 AAL".

[16]       ETSI EN 302 307: "Digital Video Broadcasting (DVB); Second generation framing structure, channel coding and modulation systems for Broadcasting, Interactive Services, News Gathering and other broadband satellite applications".

[17]       ETSI EN 302 186: "Satellite Earth Stations and Systems (SES); Harmonized EN for satellite mobile Aircraft Earth Stations (AESs) operating in the 11/12/14 GHz frequency bands covering essential requirements under article 3.2 of the R&TTE Directive".

[18]       ETSI EN 301 427: "Satellite Earth Stations and Systems (SES); Harmonized EN for Low data rate Mobile satellite Earth Stations (MESs) except aeronautical mobile satellite earth stations, operating in the 11/12/14 GHz frequency bands covering essential requirements under article 3.2 of the R&TTE Directive".

[19]       ETSI EN 302 340: "Satellite Earth Stations and Systems (SES); Harmonized EN for satellite Earth Stations on board Vessels (ESVs) operating in the 11/12/14 GHz frequency bands allocated to the Fixed Satellite Service (FSS) covering essential requirements under article 3.2 of the R&TTE Directive".

[20]       ETSI EN 301 358: "Satellite Earth Stations and Systems (SES); Satellite User Terminals (SUT) using satellites in geostationary orbit operating in the 19,7 GHz to 20,2 GHz (space-to-earth) and 29,5 GHz to 30 GHz (earth-to-space) frequency bands".

[21]     Directive 1999/5/EC of the European Parliament and of the Council of 9 March 1999 on radio equipment and telecommunications terminal equipment and the mutual recognition of their conformity (R&TTE Directive).

[22]     National Imagery and Mapping Agency (NIMA) Technical Report TR8350.2, "Department of Defense World Geodetic System 1984".

[23]     ETSI TS 102 602: "Satellite Earth stations and Systems (SES); Connection Control Protocol (C2P) for DVB-RCS".

[24]     ETSI EN 302 448: "Satellite Earth Stations and Systems (SES); Harmonized EN for tracking Earth Stations on Trains (ESTs) operating in the 14/12 GHz frequency bands covering essential requirements under article 3.2 of the R&TTE directive".

[25]     ETSI TS 102 606: "Digital Video Broadcasting (DVB); Generic Stream Encapsulation (GSE) Protocol".

[26]     ETSI TS 102 472: "Digital Video Broadcasting (DVB); IP Datacast over DVB-H: Content Delivery Protocols".

[27]     IANA: "Unidirectional Lightweight Encapsulation (ULE) Next-Header Registry".

NOTE:     Available at http://www.iana.org/assignments/ule-next-headers/.

## 2.2     Informative references

The following referenced documents are not essential to the use of the present document but they assist the user with regard to a particular subject area. For non-specific references, the latest version of the referenced document (including any amendments) applies.

[i.1]     ETSI TR 101 790: "Digital Video Broadcasting (DVB); Interaction channel for Satellite Distribution Systems; Guidelines for the use of EN 301 790".

[i.2]     ETSI TR 101 202: "Digital Video Broadcasting (DVB); Implementation guidelines for Data Broadcasting".

[i.3]     ETSI TR 100 815: "Digital Video Broadcasting (DVB); Guidelines for the handling of Asynchronous Transfer Mode (ATM) signals in DVB systems".

[i.4]     ETSI TR 101 154: "Digital Video Broadcasting (DVB); Implementation guidelines for the use of MPEG-2 Systems, Video and Audio in satellite, cable and terrestrial broadcasting applications".

[i.5]     ITU Radio Regulations.

[i.6]     IETF RFC 1112: "Host extensions for IP multicasting".

# 3     Definitions, symbols and abbreviations

## 3.1     Definitions

For the purposes of the present document, the following terms and definitions apply:

**reserved:** when used in the clauses defining the coded bit stream, indicates that the value may be used for future extensions

NOTE:     The value of reserved bits follows EN 300 468 [3] except in encrypted DVB-RCS specific messages as explicitly stated in clause 8.

## 3.2 Symbols

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

| | |
|---|---|
| $\times$ | multiplication |
| ^ | power |
| ~ | concatenation |
| mod | modulo division |
| (unsigned char)x | ANSI C cast operator: converts value x to unsigned char |
| "" | empty string (zero length) |
| nonce1 | random string (NCC) |
| nonce2 | random string (RCST) |
| $N_{atm}$ | Number of ATM cells in an ATM TRF burst (1, 2 or 4) |
| $N_{mpeg}$ | Number of MPEG packets in an optional MPEG2-TS TRF burst (1, $2 \times n$ for n = 1 to 12) |
| $N_{p,atm}$ | Number of bytes of the optional prefix used on ATM TRF bursts (0, 2 or 4) |
| $N_{p,sync}$ | Number of bytes of the optional SAC field used on SYNC bursts, after randomization and including optional CRC: 0, 2…31 for concatenated code, 0, 12 or 16 for the Turbo code (see clause 6.4) |

## 3.3 Abbreviations

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

| | |
|---|---|
| AAL | ATM Adaptation Layer |
| ACM | Adaptive Coding and Modulation |
| ACQ | ACQuisition burst |
| ADT | Application Data Table |
| ATM | Asynchronous Transfer Mode |
| AUU | ATM User-to-ATM-User |
| AVBDC | Absolute Volume-Based Dynamic Capacity |
| BAT | Bouquet Association Table |
| BBHEADER | Baseband Header (DVB-S2) |
| BCD | Binary Coded Decimal |
| BTP | Burst Time Plan |
| C2P | Connection Control Protocol |
| CBC | Cipher Block Chaining |
| CCC-IE | Continuous Carrier Control Information Element |
| CCM | Constant Coding and Modulation |
| CMF | Control and Monitoring Functions |
| CMT | Correction Message Table |
| CNI | Carrier to Noise plus Interference ratio |
| CPCS-PDU | Common Part Convergence Sublayer Protocol Data Unit |
| CR | Capacity Requests |
| CRA | Constant-Rate Assignment |
| CRC | Cyclic Redundancy Check |
| CRSC | Circular Recursive Systematic Convolutional |
| CSC | Common Signalling Channel |
| CTRL/MNGM | Control/Management virtual channel used in DULM |
| DES | Data Encryption Standard |
| DSM-CC | Digital Storage Medium - Command and Control |
| DULM | Data Unit Labelling Method |
| DVB | Digital Video Broadcast |
| DVB-S | Digital Video Broadcast by Satellite |
| DVB-S2 | Digital Video Broadcasting - Satellite transmission 2nd generation |
| EIRP | Emitted Isotropic Radiated Power |
| EIT | Event Information Table |
| EKE | Explicit Key Exchange |
| ESI | Encoding Symbol Identifier |
| FCA | Free Capacity Assignment |
| FCT | Frame Composition Table |

FDT          Forward Error Correction Data Table
FEC          Forward Error Correction
FECFRAME     Forward Error Correction Frame (DVB-S2)
FLS          Forward Link Signalling
GEO          Geostationary Earth Orbit
GFC          Generic Flow Control
GSE          Generic Stream Encapsulation
HMAC         Hash-based Message Authentication Code
I            In-phase
IANA         Internet Assigned Numbers Authority
ID           IDentifier
IDU          InDoor Unit
IE           Information Element
IEC          International Electrotechnical Commission
IEEE         Institute of Electrical and Electronics Engineers
IP           Internet Protocol
ISDN         Integrated Services Digital Network
ISO          International Standards Organization
ITU          International Telecommunication Union
IV           Initialization Vector
LFSR         Linear Feedback Shift Register
LLC          Logical Link Control
LL-FEC       Link Layer Forward Error Correction
LSB          Least Significant Bit
M&C          Monitoring and Control
MAC          Medium Access Control
MF-TDMA      Multiple-Frequency Time-Division Multiple Access
MIB          Management Information Base
MKE          Main Key Exchange
Mobile Opt   Optional requirement for RCST's supporting Mobility
Mobile Req   Mandatory provisions for RCST's supporting Mobility
MODCOD       MODulation-CODing combination (DVB-S2)
MPAF         MPEG Adaptation Field method
MPE          Multiple Protocol Encapsulation
MPEG         Moving Picture Experts Group
MSB          Most Significant Bit
NCC          Network Control Centre
NCR          Network Clock Reference
NIT          Network Information Table
NIU          Network Interface Unit
NLOS         Non-Line-of-Sight
NMC          Network Management Center
ODU          Outdoor unit
OPCR         Original Program Clock Reference
OSI          Open Systems Interconnection
PAT          Program Association Table
PC           Personal Computer
PCR          Program Clock Reference
PDU          Payload Data Unit
PID          Packet IDentifier
PLFRAME      Physical Layer Frame (DVB-S2)
PLHEADER     Physical Layer Header (DVB-S2)
PLR          Packet Loss Ratio
PMT          Program Map Table
ppm          parts per million
PRBS         Pseudo Random Binary Sequence
PRNG         Pseudo-Random Number Generator
PSI          Program Specific Information
PSTN         Public Switched Telephone Network
PVC          Permanent Virtual Circuit
Q            Quadrature
QKE          Quick Key Exchange

| | |
|---|---|
| QPSK | Quadrature Phase-Shift Keying |
| RBDC | Rate-Based Dynamic Capacity |
| RCS | Return Channel via Satellite??? |
| RCST | Return Channel Satellite Terminal |
| RMT | RCS Map Table |
| RS | Reed-Solomon |
| SAC | Satellite Access Control |
| SAR | Segmentation And Re-assembly |
| SBN | Source Block Number |
| SCT | Superframe Composition Table |
| SDT | Service Description Table |
| SF | Spreading Factor |
| SI | Service Information |
| SIT | Satellite Interactive Terminal |
| SMATV | Satellite Master Antenna Television |
| SNAP | Sub Network Access Protocol |
| SNMP | Simple Network Management Protocol |
| SOF | Start Of Frame |
| SPT | Satellite Position Table |
| SUT | Satellite User Terminal |
| SVC | Switched Virtual Circuit |
| SYNC | SYNChronization burst type |
| TBTP | Terminal Burst Time Plan |
| TCT | Time-slot Composition Table |
| TDMA | Time-Division Multiple Access |
| TG | Traffic Gateway |
| TIM | Terminal Information Message |
| TRF | Traffic (burst type) |
| TS | Transport Stream |
| Tx | Transmitter |
| UNI | User Network Interface |
| VBDC | Volume-Based Dynamic Capacity |
| VCI | Virtual Circuit Identifier |
| VPI | Virtual Path Identifier |
| WGS84 | World Geodetic System 1984 |
| XFECFRAME | CompleX FECFRAME (DVB-S2) |

# 4 Reference models for satellite interactive networks in DVB

## 4.1 Protocol stack model

For interactive services supporting broadcast to the end user with return channel, a simple communications model consists of the following layers:

**physical layer:** where all the physical (electrical) transmission parameters are defined.

**transport layer:** defines all the relevant data structures and communication protocols like data containers, etc.

**application layer:** is the interactive application software and runtime environment (e.g. home shopping application, script interpreter, etc.).

A simplified model of the OSI layers was adopted to facilitate the production of specifications for these layers. Figure 1 points out the lower layers of the simplified model and identifies some of the key parameters for the lower two layers.
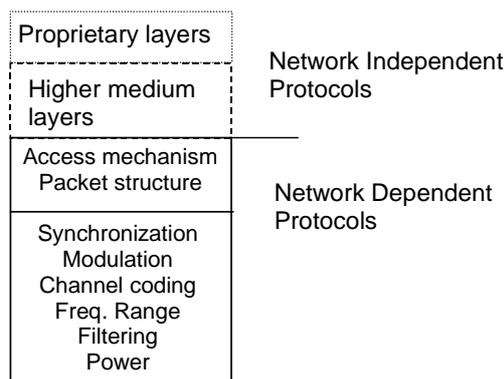
**Figure 1: Layer structure for generic system reference model**

The present document addresses the satellite interactive network dependent aspects only.

# 4.2    System model

Figure 2 shows the system model which is to be used within DVB for interactive services.

In the system model, two channels are established between the service provider and the user:

- **Broadcast Channel:** a unidirectional broadband Broadcast Channel including video, audio and data is established from the service provider to the users. It may include the Forward Interaction Path.

- **Interaction Channel:** a bi-directional Interaction Channel is established between the service provider/user and the user for interaction purposes. It is formed by:

    - **Return Interaction Path (Return Channel):** from the user to the service provider. It is used to make requests to the service provider/user, to answer questions or to transfer data.

    - **Forward Interaction Path:** from the service provider to the user. It is used to provide information from the service provider/user to the user(s) and any other required communication for the interactive service provision. It may be embedded into the Broadcast Channel. It is possible that this channel is not required in some simple implementations which make use of the Broadcast Channel for the carriage of data to the user.

The RCST is formed by the Network Interface Unit (consisting of the Broadcast Interface Module and the Interactive Interface Module) and the Set Top Unit. The RCST provides interface for both Broadcast and Interaction Channels. The interface between the RCST and the interaction network is via the Interactive Interface Module.
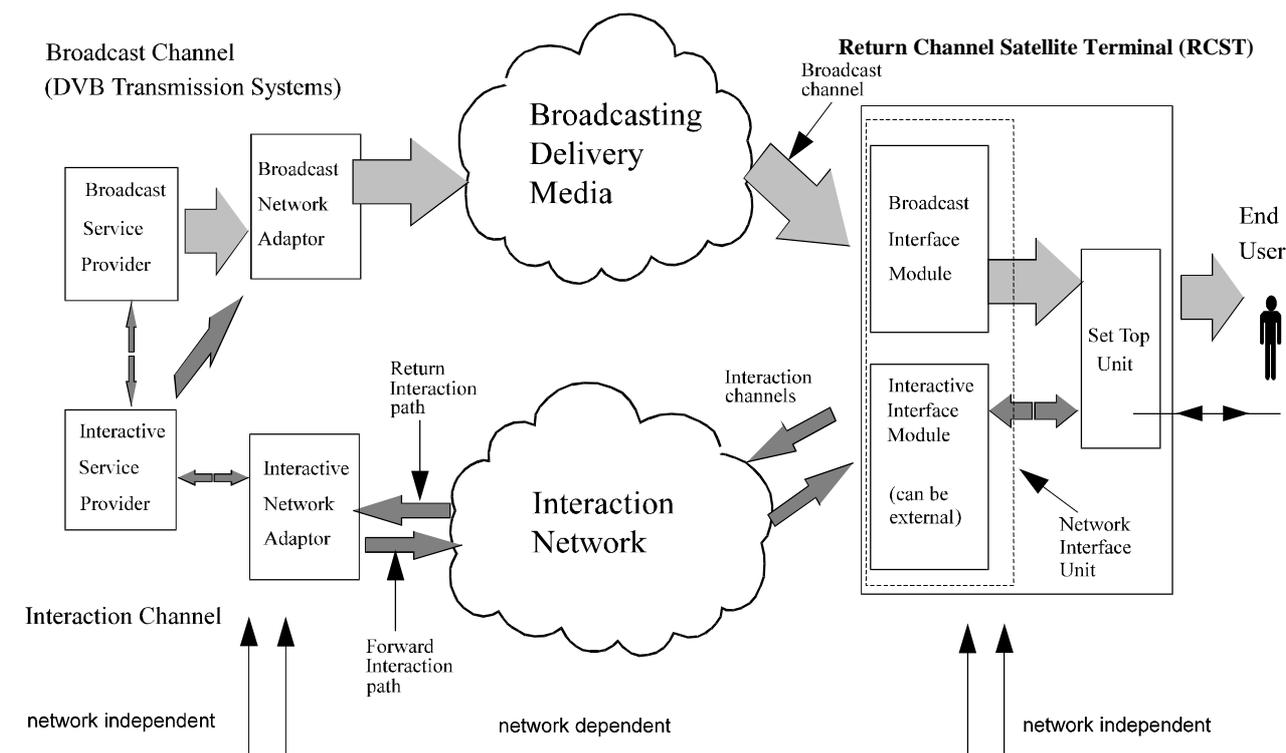
**Figure 2: A generic system reference model for interactive systems**

# 4.3      Reference model of the Satellite Interactive Network

An overall Satellite Interactive Network, within which a large number of Return Channel Satellite Terminal (RCST) will operate, will comprise the following functional blocks, as shown in figure 3:

- **Network Control Centre:** a NCC provides Control and Monitoring Functions (CMF). It generates control and timing signals for the operation of the Satellite Interactive Network to be transmitted by one or several Feeder Stations.

- **Traffic Gateway:** a TG receives the RCST return signals, provides accounting functions, interactive services and/or connections to external public, proprietary and private service providers (data bases, pay-per-view TV or video sources, software download, tele-shopping, tele-banking, financial services, stock market access, interactive games, etc.) and networks (Internet, ISDN, PSTN, etc.).

- **Feeder:** a Feeder transmits the forward link signal, which is a standard satellite digital video broadcast (DVB-S or DVB-S2) uplink, onto which are multiplexed the user data and/or the control and timing signals needed for the operation of the Satellite Interactive Network.

An RCST is e.g. a SIT or a SUT as described in [5].
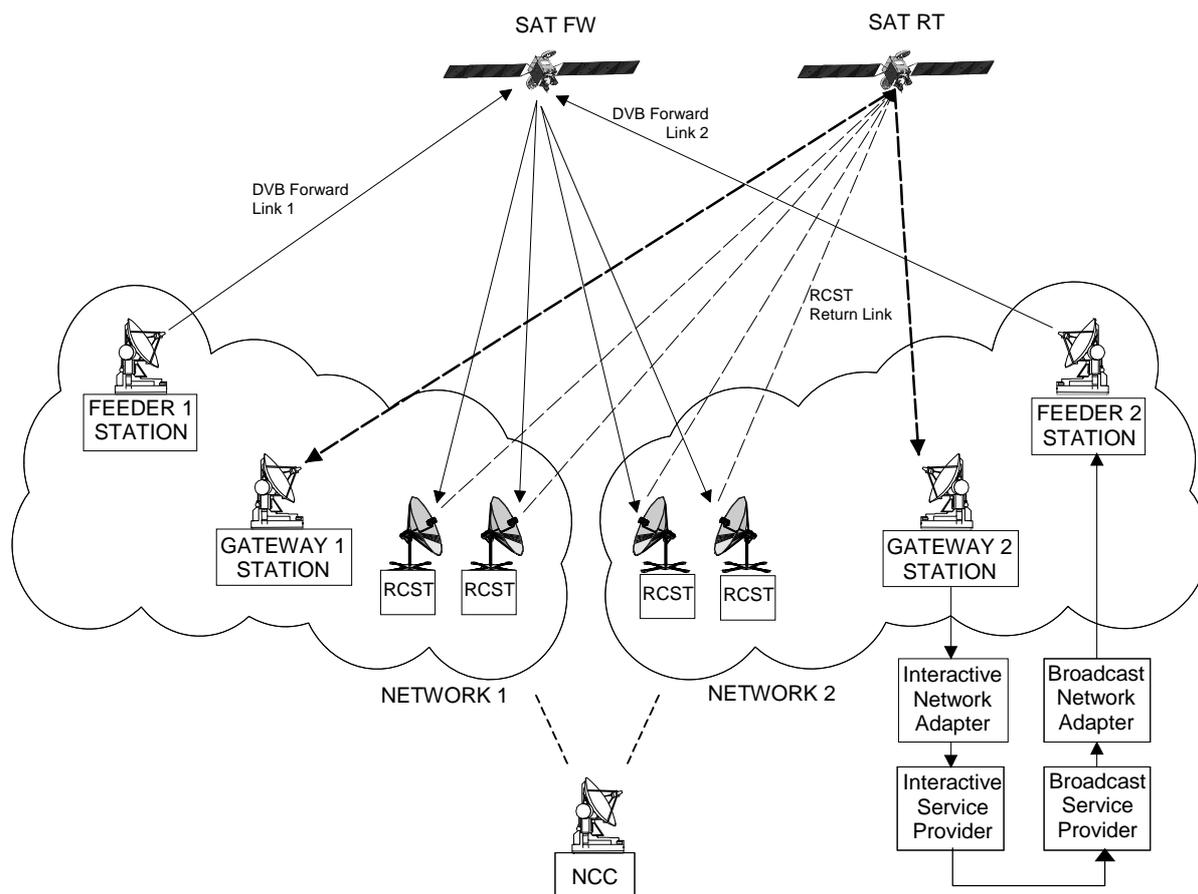
**Figure 3: Reference model for the Satellite Interactive Network**

The forward link carries signalling from the NCC and user traffic to RCSTs. The signalling from the NCC to RCSTs that is necessary to operate the return link system is called "Forward Link Signalling" in the following. Both the user traffic and forward link signalling can be carried over different forward link signals. Several RCST configurations are possible depending on the number of forward link receivers present on the RCST.

The Satellite Interactive Network for the mobile scenario comprises the same functional blocks (e.g. NCC, NMC, Gateway, Interactive Network Adapter) and signalling mechanisms (forward and return link) as described for the fixed scenario. The reference model for this scenario is shown in figure 4.

The mobile RCSTs, NCC and Gateway implement additional features (e.g. spreading, blocking channel countermeasures) which are specific to the mobile environment. Mobility management mechanisms and techniques are included to address handover requirements in different application scenarios and environment conditions (Line of Sight, non-Line of Sight) Specific signalling between NCC, gateway and RCSTs to manage beam handovers is introduced.
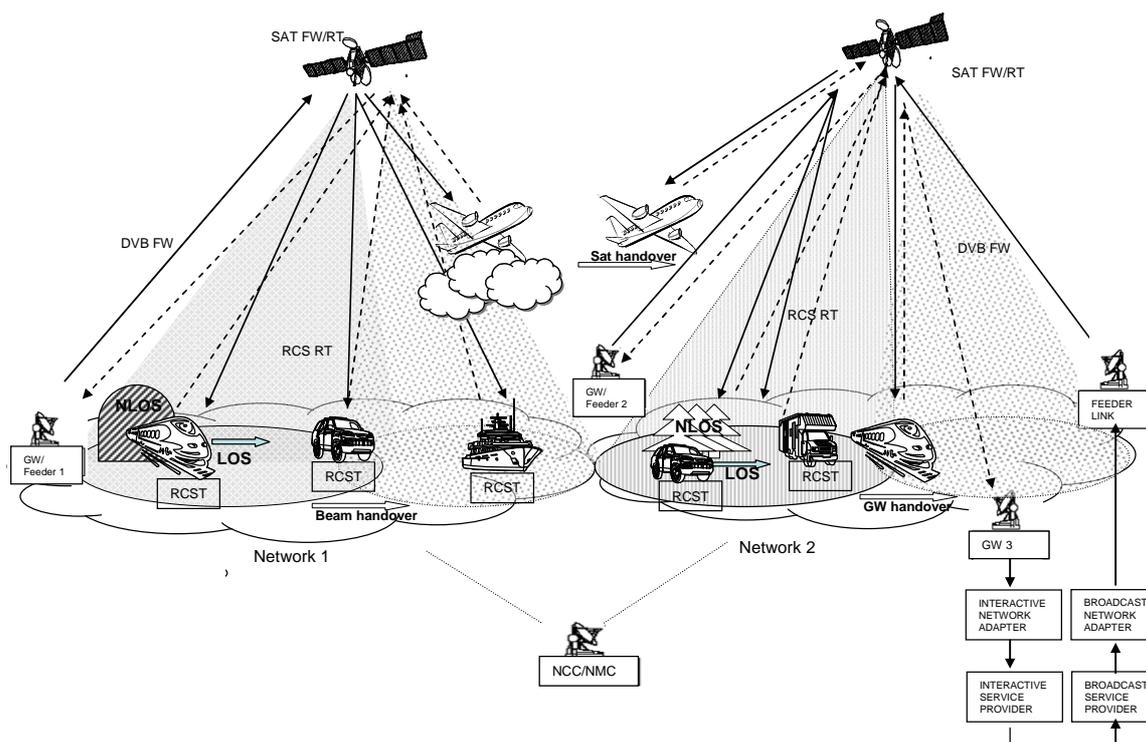
**Figure 4: Reference model for the Mobile Satellite Interactive Network**

# 4.4     Dynamic connectivity

Dynamic connectivity is defined as the capability to establish, modify or release link layer connections between Hub/RCST based upon events occurring on traffic/control or management level.

Dynamically controlled connectivity can be implemented as dynamic connectivity within the existing links in the reference model.

Additional types of satellite interactive networks that exceed the reference model can be built by implementing dynamic connectivity in the form of:

- Dynamically controlled connectivity via direct mesh links between RCSTs, through satellite on-board conversion from MF-TDMA to one or more TDM carriers.

- Dynamically controlled connectivity via direct mesh links between RCSTs equipped with an MF-TDMA receiver, through the MF-TDMA interaction channel over a transparent satellite.

# 5       Forward link

The RCST shall be able to receive digital signals conforming to EN 300 421 [1], TR 101 202 [i.2], ETS 300 802 [2], EN 300 468 [3], EN 301 192 [4], TR 101 154 [i.4] and EN 302 307 [16], as applicable.

With DVB-S2, two profiles may be used, the broadcast profile using only Constant Coding and Modulation (CCM); the interactive profile using adaptive coding and modulation (ACM). When applied, ACM on the forward link under control of the NCC is enabled by transmitting over the return link the CNI parameter and the MODCOD_RQ parameter that are defined in clause D.5 of [16]. The two parameters are transmitted by the ACM sub-field of the SAC field (see clause 6.6.1.1), or by the ACM information element of the DULM (see clauses 6.6.2.1 and 6.6.2.2).

The ACM sub-field and the ACM information element consist of a 8-bit unsigned integer number that takes the values 0 to 255 of the CNI parameter, a 7-bit field for the MODCOD_RQ parameter and a 1-bit reserved field for future use. As defined in [16], the MODCOD_RQ parameter allows either requesting a particular transmission mode characterized by MODCOD and the presence of pilot symbols, or indicating that information is not available and no particular transmission mode is requested. The RCST must transmit the currently measured CNI parameter and the derived MODCOD_RQ parameter each time it gets assigned a time slot containing the ACM sub-field. (With DULM applied in a network, the condition for transmitting the ACM parameters is not yet defined.)

The Transmission Mode Support Table, which is defined in clause 8.5.5.13, describes the transmission modes actually supported by the network for forward link transmission. The table contains a loop over transmission mode definitions, each characterized by the MODCOD value, the use of pilot symbols and the possible FECFRAME length. When the RCST must transmit a MODCOD_RQ parameter value, it either selects one of the transmission modes from the table and composes the MODCOD_RQ value accordingly, or it uses the special MODCOD_RQ value indicating information is not available as defined in [16].

In response to ACM sub-fields or information elements from an RCST, the NCC adapts accordingly the transmission mode of each PLFRAME that the RCST shall receive. It uses either the transmission mode that the RCST has requested by MODCOD_RQ, or a transmission mode that appears in the Transmission Mode Support Table before the requested one.

## 5.1      Spectrum spreading in the forward link (Mobile Opt)

A DVB-S2 forward link transmission can be spread in bandwidth using the provisions in this clause. Such spreading is applied in two stages: spreading and scrambling. The first operation, spreading, multiplies every ($I+jQ$) symbol by a sequence of chips to enlarge the bandwidth of the signal. The number of chips per symbol is called the Spreading Factor ($SF$). When $SF = 1$, the transmission is a conventional DVB-S2 signal.

The second operation, scrambling, applies a scrambling code to the spread signal. The processing is illustrated in figure 5.
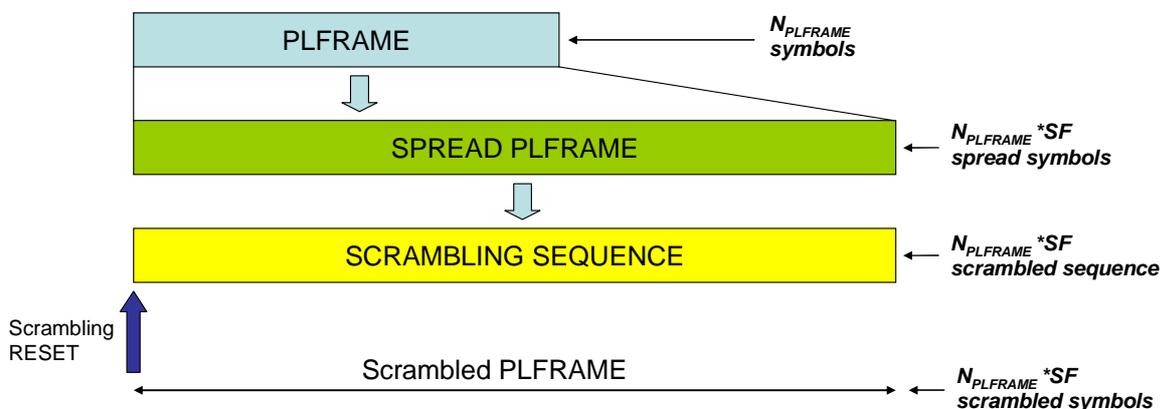


**Figure 5: Forward link spectrum spreading**

Spreading shall be applied on a PLFRAME basis, following the conventional DVB-S2 physical layer scrambling process. Each symbol in a PLFRAME, including the PLHEADER and pilot symbols if used, shall be spread by a repetition of a real-valued spreading code $C(i)$. The output of the spreading for each symbol on the $I$ and $Q$ branches shall thus be a sequence of $SF$ chips corresponding to the spreading code chip sequence, multiplied by the corresponding, real-valued symbol component value. The spreading code sequence shall be time-aligned time with the symbol boundary.

If $\{d[k]\}$, $k = 0, 1,\ldots, N_{PLFRAME}-1$, represents the $(I+jQ)$ symbols of the PLFRAME, where $N_{PLFRAME}$ is the number of symbols in one PLFRAME, then the spreading operation yields the spread sequence $s(i)$:

$$s(i) = d\left(\lfloor i / SF \rfloor\right) C\left(\mathrm{mod}(i, SF)\right) \text{ for } i = 0, 1,\ldots, (N_{PLFRAME} \times SF)\text{-}1$$

Spreading codes $C(i)$ are defined for spreading factors of 1, 2, 3 and 4 and are signalled in the Satellite Forward Link Descriptor in clause 8.5.5.10.11. In terms of the reference modulator signal flow defined in DVB-S2 [16], the spreading shall be performed immediately prior to the physical layer scrambling.

The second operation, scrambling, shall be achieved through the use of the same method as that defined for physical layer scrambling in clause 5.5.4 of [16], except that:

- the length of the scrambling sequence is here equal to $N_{PLFRAME} \times SF$, rather than $N_{PLFRAME}$; and

- the scrambling sequence is applied to the entire spread PLFRAME, including the PLHEADER and pilots if used.

The scrambling sequence shall be aligned with the PLFRAME epoch, and it shall be re-initialized at the beginning of each PLFRAME.

The sequence of complex valued chips shall be scrambled (complex chip-wise multiplication) by the complex-valued scrambling code, $w(i)$, defined in [16], clause 5.5.4, when $SF$ is greater than 1. The Spread PLFRAME duration depends on the selected modulation and the adopted spreading factor. The scrambled symbols, $z(i)$, shall be obtained by directly multiplying the spread symbols, $s(i)$, by the scrambling sequence, $w(i)$, as follows:

$$z(i) = s(i) \times w(i \text{ modulo } 66420), i=0,1,2,\ldots,(N_{PLFRAME} \times SF)\text{-}1$$

After scrambling, the signal $\{z(i)\}$ shall be square root raised cosine filtered as described in [16], clause 5.6.

It is mandatory to define an explicit scrambling sequence in the corresponding satellite forward link descriptor (clause 8.5.5.10.11) when $SF$ is greater than 1.

# 6       Return link base-band physical layer specification and multiple access definition

Specifications for the base-band physical layer are given in this clause. Figure 6 represents the generic digital signal processing to be performed at the RCST transmitter side, from the burst formatting of the serial information bit-stream, to the modulation representing the digital to analogue conversion. The signal processing to be performed by each subset is described in the following clauses.
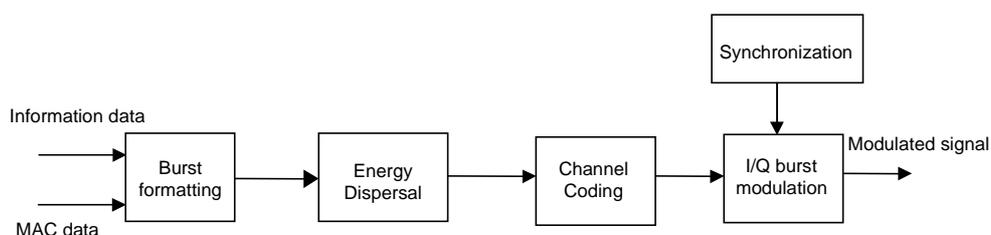


**Figure 6: Block diagram of the RCST return link baseband signal processing**

# 6.1      RCST synchronization

## 6.1.1    Timing control

The synchronization of the RCST is an important feature of the satellite interactive network. Constraints are imposed on the RCSTs to obtain an efficient TDMA system with minimum interference between users and maximum throughput, although they can be minimized if the NCC performs tasks such as satellite frequency translation error and common-mode Doppler compensation for RCST carrier frequency. For this reason, the synchronization scheme is based on information contained within the Forward Link Signalling as follows:

- Network Clock Reference (NCR);

- signalling in DVB/MPEG2-TS private sections.

The NCR is distributed with a specific PID within the MPEG2 Transport Stream that carries the Forward Link Signalling. If DVB-S or DVB-S2 with CCM is used on the forward link, then the NCR distribution follows the PCR distribution mechanism as defined in ISO/IEC 13818-1 [7], which is usually derived from an MPEG video encoder, whereas here the NCR is derived from the NCC reference clock. The NCC reference clock will have an accuracy of 5 ppm or better.

The following mechanism shall be applied when the forward link uses DVB-S2 with ACM.

To be able to construct a reference time axis for TDMA transmissions in case of a DVB-S2 forward link with ACM, the RCST will associate a successfully received NCR field value with the arrival time at a system dependent reference point of a forward link reference_symbol.

The reference_symbol shall be the first symbol of the Start-Of-Frame field of the N-th DVB-S2 physical layer frame for an NCR field the most significant bit of which is carried in the (N+2)th DVB-S2 physical layer frame.

The offset of 2 frames accommodates the encoding time in the forward link equipment. No ambiguity arises if an NCR field is split over two physical layer frames since the most significant NCR bit is always transmitted in the first physical layer frame, as shown in figure 5.
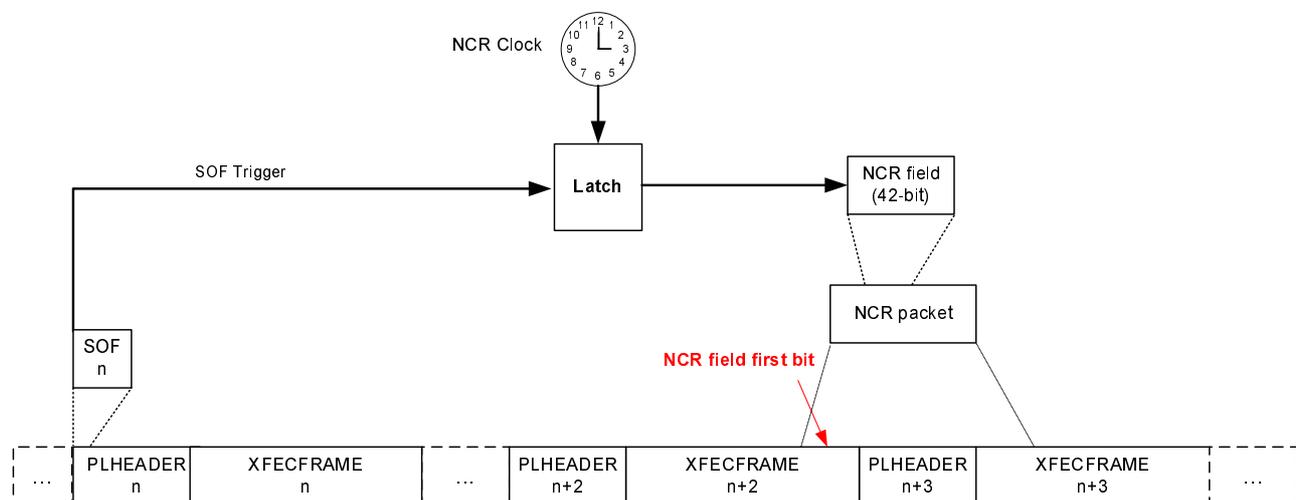


**Figure 7: Association of NCR to SOF event in the transmitter**

Figure 7 illustrates potential splitting of an NCR field over two DVB-S2 frames. In case the DVB-S2 signal carries a single Transport Stream, both fragments of the NCR will be transmitted in consecutive DVCB-S2 frames n + 2 and n + 3, as shown. More generally a DVB-S2 signal could have multiple transport streams. In that case, frames n + 3, …, n + k - 1 belonging to other transport stream ID may occur between the frames n + 2 and n + k (k ≥ 3) that carry the fragments of the NCR field pointing to the event SOF n.

## 6.1.2    Carrier synchronization

The MPEG2-TS that carries the Forward Link Signalling contains a NCR information which provides a 27 MHz reference of the NCC reference clock to the RCSTs.

Normalized carrier frequency accuracy shall be better than $10^{-8}$ (root mean square).

## 6.1.3    Burst synchronization

The RCSTs retrieve the centre frequency, the start time and the duration of their transmit bursts by examining the forward link signalling (more precisely the SCT, FCT and TCT tables described in clause 8.3.1).

The contention between RCSTs on the return link is resolved as described in clause 6.7.

The bursts are sent according to the Burst Time Plan (BTP) received in the Forward Link Signalling (see clause 6.7.2). The BTP is expressed in terms of centre frequency and absolute start time (given in NCR-counter value) of superframes and associated frequency and time offsets of burst allocations along with a description of the time slot properties. A superframe always starts at a given value of the RCST local NCR counter, which serves as a reference for all burst allocations within the superframe. For the purpose of synchronizing to the network, the RCST reconstructs the absolute value of the NCC reference clock. The RCST compares the reconstructed value with the NCR value given by the BTP. The time reference for counting timeslots occurs when the values are equal.

Burst synchronization accuracy shall be within 50 % of a symbol period. The resolution shall be 1 NCR count interval. The burst synchronization accuracy is the worst case deviation of the scheduled start of burst time and the actual start of burst time at the transmitter output. The scheduled start of burst time is the point in time when the ideal reconstructed NCR equals the value written in the TBTP for that burst. The ideal reconstructed NCR is defined as observed at the output of an ideal delay-less DVB-S receiver. Compensation for the receiver delay, if required to achieve the specified accuracy, shall be done by the RCST.

When spectrum spreading by burst repetition is applied, the transmission instants of individual replica's of the same burst shall not vary relative to the start instants of their respective sub-timeslots by more than ±5 % of the symbol period.

## 6.1.4    Symbol clock synchronization

Symbol clock accuracy shall be within 20 ppm from the nominal symbol_rate value in the TCT (see clause 8.5.5.4). The symbol clock rate shall have a short-term stability that limits the time error of any symbol within a burst to 1/20 symbol duration.

## 6.2    Burst format

There are four types of bursts:

- TRaFfic (TRF);

- ACQuisition (ACQ);

- SYNChronization (SYNC); and

- Common Signalling Channel (CSC).

The burst formats are described in the following.

## 6.2.1    Traffic (TRF) burst formats

Traffic (TRF) bursts are used for carrying useful data from the RCST to the Gateway(s)/RCST. Two types of traffic bursts carrying either ATM cells or MPEG2-TS packets are defined here below. Channel coding of these bursts is defined in clause 6.4. A TRF is usually followed by a guard time to turn-off transmitted power and compensate for time offset as described in clause 6.5.4.

### 6.2.1.1        ATM TRF burst

The payload of an ATM traffic burst is composed of $N_{atm}$ concatenated ATM cells, each of length 53 bytes, plus an optional $N_{p,atm}$ byte prefix, as described in clause 6.6.1.1. ATM cells follow the structure of an ATM cell but do not necessarily support ATM classes of service. See figure 8 for a description of the ATM TRF burst.
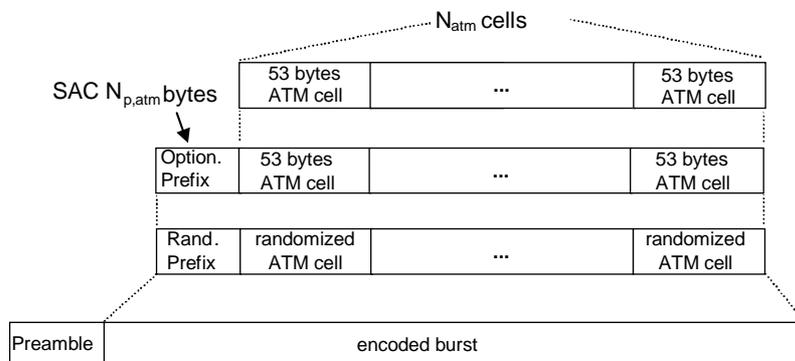


**Figure 8: Composition of an ATM TRF burst**

### 6.2.1.2        Optional MPEG2-TS TRF burst

In the case that MPEG2-TS Packets are the basic containers a burst contains $N_{mpeg}$ concatenated MPEG2-TS packets, each of length 188 bytes. The burst is composed of several channel coding blocks as described in clause 6.4. See figure 9 for a description of the MPEG2-TS TRF burst.

RCSTs can deduce the number of MPEG2 packets in a TRF time slot from the time_slot_duration field of the TCT (see clause 8.5.5.4), after subtracting the time duration of other fields. Transmission of MPEG2-TS TRF bursts is optional. The RCST will inform the NCC that it supports this mechanism in the CSC burst (see clause 6.2.3).
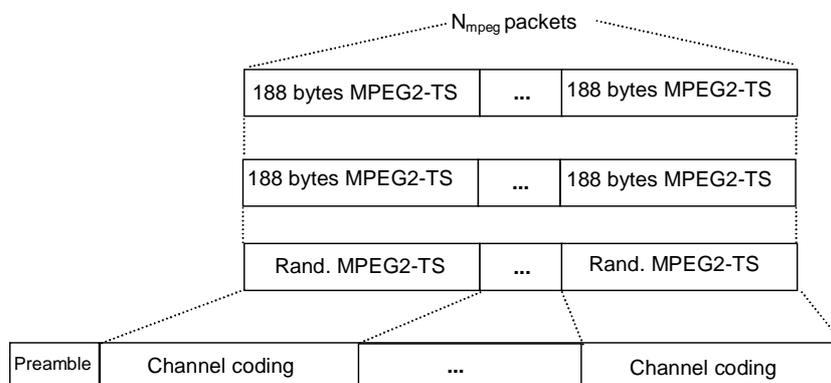


**Figure 9: Composition of the optional TRF burst carrying MPEG2-TS packets**

### 6.2.2        Synchronization and acquisition burst formats

Synchronization and Acquisition bursts are required to accurately position RCST burst transmissions during and after logon. Two separate burst types are defined for this purpose (SYNC and ACQ) as defined in the following clauses.

## 6.2.2.1        Synchronization (SYNC) burst format

A SYNC burst is used by an RCST for the purpose of maintaining synchronization and sending control information to the system. SYNC bursts are composed of a preamble for burst detection (configurable and indicated to the RCST through the TCT, as described in clause 8.5.5.4), and an optional SAC_length byte Satellite Access Control (SAC) field as described in clause 6.6.1.1. After randomization (as described in clause 6.3) an optional CRC (as described in clause 6.4) can be added to this field, giving a total container size of Np,sync bytes. This container is further protected with the appropriate error control coding as described in clause 6.4. Like a TRF a SYNC is usually followed by a guard time to decrease transmitted power and compensate for time offset (see clause 6.5.4). Figure 10 depicts the SYNC burst. The extent to which the SYNC burst is used depends on the capabilities of the NCC.

NOTE:       SYNC bursts can be used in contention mode.
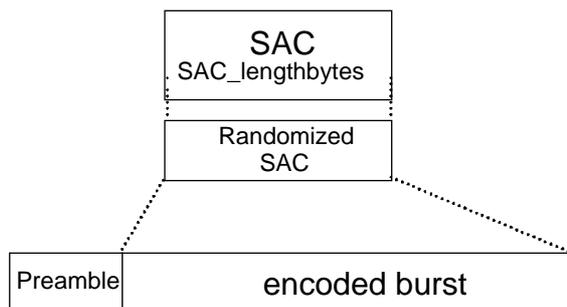


**Figure 10: Composition of a SYNC burst**

## 6.2.2.2        Acquisition (ACQ) burst

An ACQ burst can be used to achieve synchronization, prior to operational use of the network by the RCST.

Transmissions in an ACQ burst shall have the format shown on figure 11.
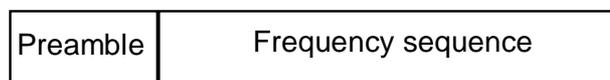


**Figure 11: Composition of the ACQ burst**

The preamble length and content (including the frequency sequence) are sent to the RCSTs via the TCT (see clause 8.5.5.4). The ACQ is surrounded by a guard interval as described in clause 6.5.4.

## 6.2.3    Common Signalling Channel (CSC) burst format

Common signalling channel (CSC) bursts are only used by an RCST to start the login procedure and return link synchronisation and to also indicate the terminal beam location of mobile and portable terminals while not logged in. They are composed of a preamble for burst detection and start of burst detection, three fields describing the RCST capabilities, the RCST MAC address, CSC_Route_ID, a reserved field and a burst type identifier (see figure 12). Coding of these bursts is defined in clause 6.4. Table 1 gives the CSC burst content. The CSC is surrounded by a guard interval as described in clause 6.5.4.

A mobile RCST that is not logged on may use the CSC burst with the mobility support field set to "Location Update" in order to indicate to a hub that is known to support of location updates, that the mobile RCST is located in the beam in which the CSC burst is transmitted. An RCST must not issue such location updates if this is not specifically known to be allowed by the hub.
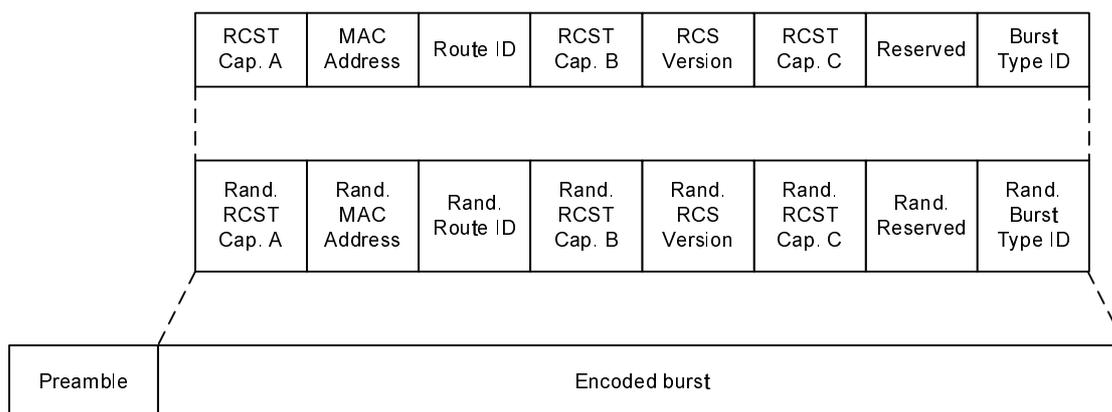
**Figure 12: Composition of a CSC burst**

**Table 1: CSC burst data field parameters**

| Field Name | Size (bits) | Description/Content |
|---|---|---|
| Preamble | variable | Preamble for burst detection and start of burst detection. Definition by TCT. See clause 8.5.5.4 |
| RCST Capability "A" | 24 | See table 2 |
| RCST MAC Address | 48 | RCST MAC address as per IEEE 802.3 [9] |
| CSC_Route_ID | 16 | Enables to define a destination forward (downlink) link for the CSC burst in a regenerative system. If the RCST indicates that this field is overloaded, the value is system dependent. |
| RCST Capability "B" | 5 | See table 3 |
| RCST Protocol Version | 2 | See table 4 |
| RCST Capability "C" | 8 | See table 5 |
| Reserved | 8 | Reserved |
| Burst type Identifier | 1 | "1" (for identification of CSC burst) |

Table 2 defines the different bit patterns within RCST capability field "A". The 24-bit field is numbered from LSB to MSB using the notation b0 through b23.

**Table 2: RCST capability "A"**

| Parameter | Bit Size | Description |
|---|---|---|
| Security mechanism | 1 (b23) | "1" for RCST implementing security mechanism as described in clause 9.4. "0" otherwise. |
| SNMP | 1 (b22) | "1" for RCST supporting SNMP (see clauses 8.4.2, 8.4.3 and 8.5.5.10.2). "0" otherwise. |
| ATM connectivity | 1 (b21) | "1" for RCST capable of ATM connectivity (type B), "0" for not capable (Type A). |
| MPEG2-TS TRF | 1 (b20) | "1" for RCST capable of MPEG2-TS TRF, "0" for not capable. |
| RCST boards | 2 (b19-b18) | Number of RCST forward link receivers: "00" for 1 receiver, "01" for 2, "10" for more than 2, "11" reserved. |
| RCST ACQ | 1 (b17) | "0" for RCST not requiring ACQ burst, "1" for ACQ required. |
| Multi_IDU | 1 (b16) | "0" for single indoor unit/single outdoor unit configuration, "1" when two or more IDUs are connected to a single ODU. |
| S/W Version | 8 (b15-b8) | System Dependent. Can be used to define the RCST software version. |
| Freq Hopping Range | 2 (b7-b6) | Defines the RCST burst to burst frequency hopping range capability: "00" for 20 MHz, "01" for 120 MHz. Other patterns System Dependent. |
| MF-TDMA | 1 (b5) | "1" for RCST supporting dynamic MF-TDMA. "0" for RCST supporting fixed MF-TDMA (see clause 6.7.1). |
| RCST Class | 2 (b4-b3) | System Dependent. |
| Route_ID capable | 1 (b2) | "1" indicates that the RCST is capable of inserting a Route_ ID in the SAC field. "0" otherwise. |
| RCST Mode | 2 (b1-b0) | "00" for Installation Mode (see clause 8.5.5.10.5), "01" for Operational Mode "10" for Reference RCST mode (can be used for measuring satellite frequency translation offset, D/L rain fade, etc.) "11" reserved. |

Table 3 defines the different bit patterns within RCST capability field "B". The 5-bit field is numbered from LSB to MSB using the notation b0 through b4.

**Table 3: RCST capability "B"**

| Parameter | Bit Size | Description |
|---|---|---|
| Dynamic connectivity | 1 (b4) | "0" for RCST supporting Dynamic Connectivity according to [23] and supporting 8-bit Channel_ID, "1" otherwise. Declaring support for dynamic connectivity may trigger additional, optional logon stages to determine C2P version and capabilities, as defined in [23]. |
| Frequency Hopping | 1 (b3) | "1" for RCST supporting frequency hopping between adjacent time slots, "0" for RCST requiring one TRF slot between transmissions on different carrier frequencies. |
| DVB-S capability | 1 (b2) | RCST capable of using DVB-S on forward link. The field is "1" if the DVB-S is capable, "0" otherwise. |
| DVB-S2 capability | 2 (b1-b0) | RCST capable of using DVB-S2 for forward link reception: The field is "11" for not DVB-S2 capable. "01" for DVB-S2 capable of using CCM only, "00" for DVB-S2 capable of both ACM and CCM. Value "10" is reserved. |

Table 4 defines the coding used for indication of which version of the DVB-RCS standard that is implemented.

**Table 4: Implemented RCS standard version**

| Value | Version number |
|---|---|
| 11 | Version 1.4 or earlier |
| 10 | Version 1.5 |
| 01 | Reserved |
| 00 | Reserved |

Table 5 defines the different bit patterns within RCST capability field "C". The 8-bit field is numbered from LSB to MSB using the notation b0 through b7.

**Table 5: RCST Capability "C"**

| Parameter | Bit Size | Description |
|---|---|---|
| Route_ID_overload | 1 (b7) | Indicates alternative, system-dependent use of the CSC_route_ID field. "1" indicates that the field is used as defined in table 1. "0" indicates that the value is system dependent. |
| Mobility Support | 3 (b6-b4) | Mobility support, see table 6. |
| Continuous ACM | 1 (b3) | RCST capable of return link ACM in continuous mode. "0" indicates capable, "1" indicates not capable. |
| NLOS countermeasure support | 1 (b2) | RCST capable of non-line-of-sight mobile channel countermeasures. "0" indicates capable, "1" indicates not capable. |
| Transparent mesh reception support | 2 (b1-b0) | Indicates support for reception of transparent mesh signals. The field is coded as:<br>11: No burst mode reception supported<br>10: Single-carrier burst mode receiver<br>01: Multi-carrier burst mode receiver<br>00: Reserved |

**Table 6: Mobility support field values**

| Value | Capability set |
|---|---|
| 000 | Location update |
| 001 | No spread spectrum, no continuous carrier |
| 010 | No spread spectrum, basic continuous carrier |
| 011 | No spread spectrum, enhanced continuous carrier |
| 100 | Spread spectrum, no continuous carrier |
| 101 | Spread spectrum, basic continuous carrier |
| 110 | Spread spectrum, enhanced continuous carrier |
| 111 | No mobility support |
| NOTE: | Spread-spectrum transmission in either direction and continuous-carrier operation of the return link transmissions are defined in clause 6.5.5 and in clause 10, respectively. |

## 6.2.4    Bit numbering and interpretation

The term "bit 0" shall refer to the least significant bit of a multi-bit field. The most significant bit of a k-bit unsigned field shall be designated "bit k - 1". For a signed field, "bit k - 1" shall be the sign bit and "bit k - 2" the most significant magnitude-related bit.

## 6.2.5    Transmission order

Fields in data structures shall be transmitted in the order in which they are defined.

Unsigned values shall be transmitted starting with the most significant bit and ending with the least significant bit.

Signed values shall be transmitted starting with the sign bit, followed by the most significant bit and ending with the least significant bit.

Bytes shall be processed MSB first.

# 6.3    Randomization for energy dispersal

The return link data stream shall be organized in bursts as described in clause 6.2. In order to comply with ITU Radio Regulations [i.5] and to ensure adequate binary transitions, serial data bit stream in a burst shall be randomized. The polynomial of the Pseudo-Random Binary Sequence (PRBS) shall be as the one of EN 300 421 [1] (see figure 13), i.e. $1 + x^{14} + x^{15}$.

The data is randomized using the output of a 15 register Linear Feedback Shift Register (LFSR) randomized sequence (see figure 13) to ensure a random distribution of ones and zeroes. The randomizer performs modulo-2 addition of the data with the pseudo-random sequence. The initial content of the SR-1 to SR-15 registers is given in table 7. The first bit of the pseudo-random sequence is to be added in the first bit of the serial data bit stream, i.e. the first bit after the burst preamble. The randomizer is reset to the initial content before processing the next burst.

**Table 7: Initial contents of the randomizer register**

| Shift register | SR1 | SR2 | SR3 | SR4 | SR5 | SR6 | SR7 | SR8 | SR9 | SR10 | SR11 | SR12 | SR13 | SR14 | SR15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bit value | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
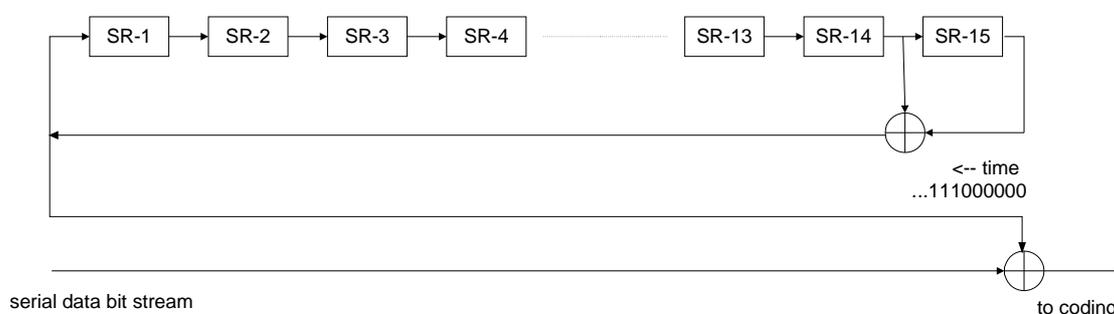


**Figure 13: Randomizer**

# 6.4    Coding

Coding for channel error protection is applied to traffic and control data, which are transmitted in the types of bursts described in clause 6.2. Two coding schemes are described: Turbo (see clause 6.4.4) and concatenated coding. RCST shall implement both schemes. Within a session (see clause 7), RCSTs are not requested to change the coding scheme (i.e. during a given session, an RCST will either use the Turbo or the concatenated code). In the case of the concatenated coding, the outer code is a by-passable Reed-Solomon (RS) code and the inner code is a by-passable non-systematic convolutional code (EN 300 421 [1]). For both coding schemes, a by-passable CRC can also be applied on CSC and SYNC bursts in order to allow error detection.

## 6.4.1    CRC error detection code

A CRC-16 can be applied on CSC and SYNC bursts in order to allow error detection. The CRC polynomial is $x^{16} + x^{15} + x^2 + 1$. The NCC indicates via the TCT (see clause 8.3.1.3) to the RCST if the CRC is to be applied. If used, the CRC is appended at the end of the burst before any other coding. CRC is applied on the randomized bit stream. The CRC is the remainder of the division of the burst payload by the polynominal. The CRC code is mandatory on turbo coded CSC bursts.

The CRC shall be equivalent to that computed by a circuit as shown in figure 14. The shift register cells shall be initialized to 0 before the start of the computation. First, the switches are in position "A", and the data word is shifted in (and simultaneously transmitted). After the last data bit, the switches are moved to position "B", and the contents of the shift register are transmitted, starting with the bit at the end of the register. This is the CRC word.
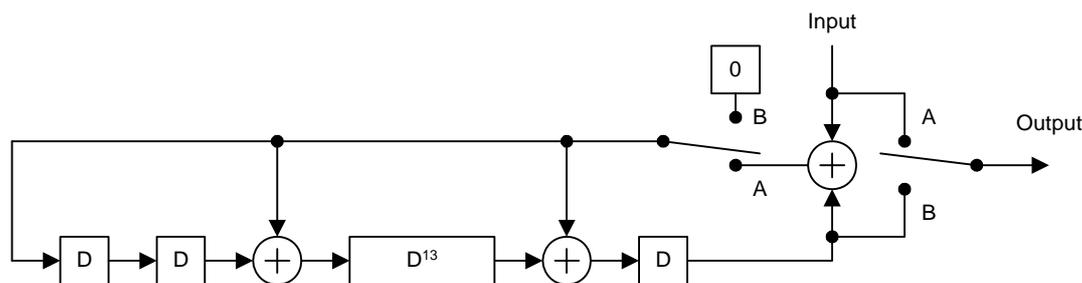
**Figure 14: CRC calculation**

## 6.4.2    Reed-Solomon outer coding

A Reed-Solomon RS (N-B, K-B, T) shortened code EN 300 421 [1] derived from the original RS (255, 239, 8) code, shortened by B bytes, can be applied for some burst formats. The code generator polynomial is $g(x) = (x + \lambda^0)\ (x + \lambda^1)\ (x + \lambda^2) \ldots (x + \lambda^{15})$, where $\lambda = 02_{HEX}$. The field generator polynomial is $p(x) = x^8 + x^4 + x^3 + x^2 + 1$. This code is similar to the one used in EN 300 421 [1].

For ATM traffic bursts the length of the encoded information word, K - B, is $N_{atm} \times 53 + N_{p,atm}$. In the case that the basic container is an MPEG2-TS packet K - B = 188 applies.

The outer code can be bypassed. The outer code is always by-passed when using Turbo codes.

If both the CRC and RS codes are used, the burst payload CRC is first computed and the RS parity bytes are then added.

## 6.4.3    Convolutional inner coding

Processing of the convolutional encoder shall be in accordance with EN 300 421 [1], as summarized in the following.

The return link shall allow for a range of punctured convolutional codes, based on a rate 1/2 mother convolutional code with constraint length K = 7 corresponding to 64 trellis states (see figure 15). The generator polynomials are $G_0 = 171$ and $G_1 = 133$ in octal representation. This choice will allow selection of the most appropriate level of error correction for a given service or data rate. Code rates of 1/2, 2/3, 3/4, 5/6 and 7/8 shall be supported. The inner code can be bypassed. In that case, the MSB bit is affected to the I channel, the next bit to the Q channel and so on. The convolutional inner code is always by-passed when using Turbo codes.

The encoder register shall be initialized to all zeroes before encoding the first data bit.

At the end of each data block, the encoder shall be flushed by 6 zero bits. This block is called the "Postamble". The output shall be continued until the encoder is in its all-zero state. If the inner code is bypassed, then the postamble is also omitted.

The puncturing pattern period counter shall be initialized before encoding the first data bit so that the first encoded (C2,C1) symbol always corresponds to an (Y1,X1) pair in table 8.

After encoding the last 0 bit of the postamble an incomplete symbol (##,C1) can remain if the message length is not divisible by the puncturing period. In that case the missing C2 is set to 0 and the burst is terminated.

X output (171 octal)
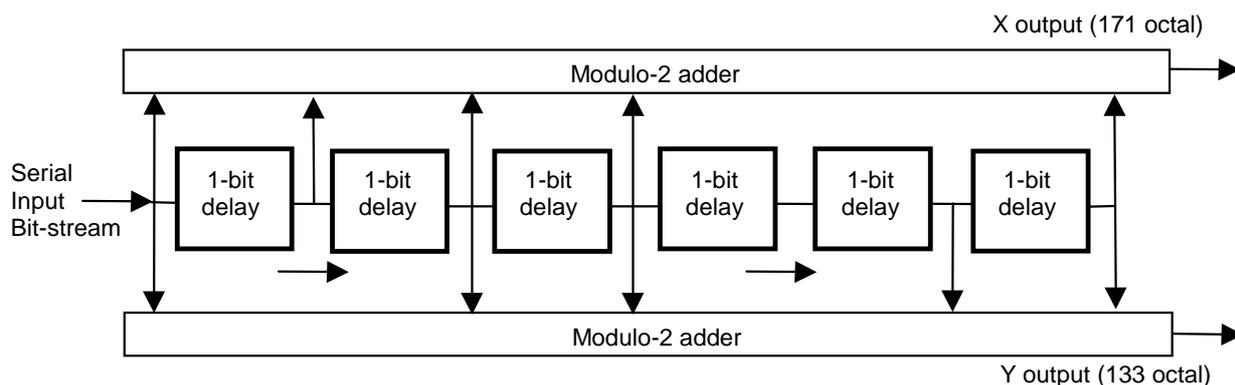


**Figure 15: Convolutional code of rate 1/2**

The punctured convolutional code shall be used as given in table 8, according to EN 300 421 [1].

**Table 8: Punctured code definition**

| Original code | | | Code rates | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | 1/2 | | 2/3 | | 3/4 | | 5/6 | | 7/8 | | | |
| K | $G_1$ (X) | $G_2$ (Y) | P | $d_{free}$ | P | $d_{free}$ | P | $d_{free}$ | P | $d_{free}$ | P | $d_{free}$ | | |
| 7 | $171_{OCT}$ | $133_{OCT}$ | X: 1 Y: 1 | 10 | X: 1 0 Y: 1 1 | 6 | X: 1 0 1 Y: 1 1 0 | 5 | X: 1 0 1 0 1 Y: 1 1 0 1 0 | 4 | X: 1 0 0 0 1 0 1 Y: 1 1 1 1 0 1 0 | 3 | | |
| | | | $C1 = X_1$ $C2 = Y_1$ | | $C1 = X_1 Y_2 Y_3$ $C2 = Y_1 X_3 Y_4$ | | $C1 = X_1 Y_2$ $C2 = Y_1 X_3$ | | $C1 = X_1 Y_2 Y_4$ $C2 = Y_1 X_3 X_5$ | | $C1 = X_1 Y_2 Y_4 Y_6$ $C2 = Y_1 Y_3 X_5 X_7$ | | | |
| NOTE: 1 = transmitted bit. 0 = non transmitted bit. | | | | | | | | | | | | | | |

## 6.4.4    Turbo code

The Turbo encoder is depicted in figure 16. It uses a double binary Circular Recursive Systematic Convolutional (CRSC) code. The MSB bit of the first byte after the burst preamble is assigned to A, the next bit to B and so on for the remaining of the burst content.

The encoder is fed by blocks of $k$ bits or $N$ couples ($k = 2 \times N$ bits). $N$ is a multiple of 4 ($k$ is a multiple of 8).

The polynomials defining the connections are described in octal and symbolic notations as follows:

- for the feedback branch: 15 (in octal), equivalently $1 + D + D^3$ (in symbolic notation);

- for the Y parity bits: 13, equivalently $1 + D^2 + D^3$;

- for the W parity bits: 11, equivalently $1 + D^3$.

The input A bit is connected to tap "1" of the shift register and the input B bit is connected to the taps "1", D and $D^2$.

First, the encoder (after initialization by the circulation state $\mathbf{S}_{C_1}$, see clause 6.4.4.2) is fed by the sequence in the natural order (switch on position 1) with incremental address $i = 0, ...N - 1$. This first encoding is called $C_1$ encoding.

Then the encoder (after initialization by the circulation state $\mathbf{S}_{C_2}$, see clause 6.4.4.2) is fed by the interleaved sequence (switch in position 2) with incremental address $j = 0, ...N-1$. This second encoding is called $C_2$ encoding. The function $\Pi(j)$ that gives the natural address $i$ of the considered couple, when reading it at place $j$ for the second encoding, is given in clause 6.4.4.1.
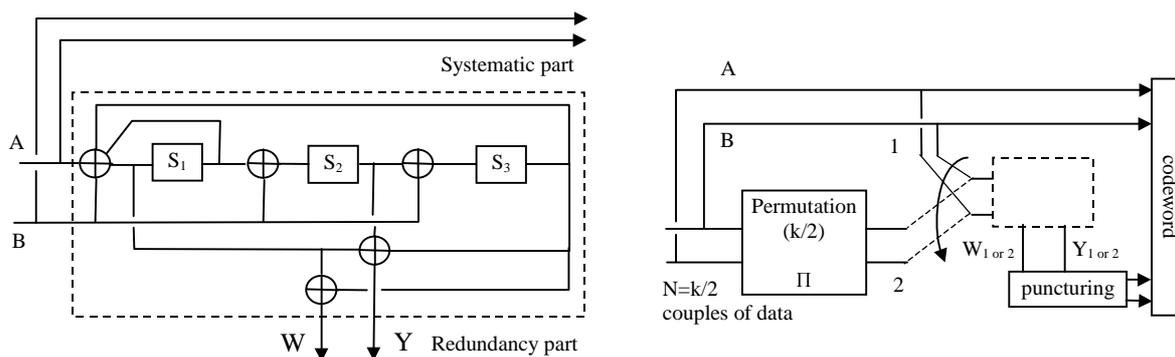
**Figure 16: Encoder block diagram (turbo code)**

## 6.4.4.1        Description of the turbo code permutation

The permutation is done on two levels, the first one inside the couples (**level 1**), the second one between couples (**level 2**):

Set the permutation parameters $P_0, P_1, P_2$ and $P_3$

$$j = 0, ...N - 1$$

**level 1**

if $j$ mod. 2 = 0, let (A,B) = (B,A) (invert the couple).

**level 2**

- if $j$ mod. 4 = 0, then $P = 0$;

- if $j$ mod. 4 = 1, then $P = N/2 + P_1$;

- if $j$ mod. 4 = 2, then $P = P_2$;

- if $j$ mod. 4 = 3, then $P = N/2 + P_3$.

$$i = P_0 \times j + P + 1 \text{ mod. } N$$

Table 9 provides the combinations of the default parameters to be used. Those parameters can be updated by the TCT (see clause 8.5.5.4). The interleaving relations satisfy the odd/even rule (i.e. when $j$ is even, $i$ is odd and vice-versa) that enables the puncturing patterns to be identical for both encodings.

**Table 9: Turbo code permutation parameters**

| Frame size in couples | $P_0$ | $\{P_1, P_2, P_3\}$ |
|---|---|---|
| N = 48 (12 bytes) | 11 | {24,0,24} |
| N = 64 (16 bytes) | 7 | {34,32,2} |
| N = 212 (53 bytes) | 13 | {106,108,2} |
| N = 220 (55 bytes) | 23 | {112,4,116} |
| N = 228 (57 bytes) | 17 | {116,72,188} |
| N = 424 (106 bytes) | 11 | {6,8,2} |
| N = 432 (108 bytes) | 13 | {0,4,8} |
| N = 440 (110 bytes) | 13 | {10,4,2} |
| N = 848 (212 bytes) | 19 | {2,16,6} |
| N = 856 (214 bytes) | 19 | {428,224,652} |
| N = 864 (216 bytes) | 19 | {2,16,6} |
| N = 752 (188 bytes) | 19 | {376,224,600} |

## 6.4.4.2        Determination of the circulation states

The state of the encoder is denoted $\mathbf{S}$ ($0 \le \mathbf{S} \le 7$) with $\mathbf{S} = 4 \times s_1 + 2 \times s_2 + s_3$ (see figure 16). The circulation states $\mathbf{S}_{C1}$ and $\mathbf{S}_{C2}$ are determined by the following operations:

1)   initialize the encoder with state 0. Encode the sequence in the natural order for the determination of $\mathbf{S}_{C1}$ or in the interleaved order for the determination of $\mathbf{S}_{C2}$ (without producing redundancy). In both cases, the successive states of the encoder are denoted $\mathbf{S}^0_k$, $0 \le k \le N$. $\mathbf{S}^0_0$ is the initialization state and $\mathbf{S}^0_N$ is the final state (i.e. the state of the encoder after all the $N$ couples have been encoded);

2)   according to the length $N$ of the sequence, use the following correspondence to find $\mathbf{S}_{C1}$ or $\mathbf{S}_{C2}$ (table 10).

**Table 10: Circulation state correspondence table**

| $\mathbf{S}^0_N \rightarrow$ $\downarrow$ N mod. 7 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| 1 | $S_C = 0$ | $S_C = 6$ | $S_C = 4$ | $S_C = 2$ | $S_C = 7$ | $S_C = 1$ | $S_C = 3$ | $S_C = 5$ |
| 2 | $S_C = 0$ | $S_C = 3$ | $S_C = 7$ | $S_C = 4$ | $S_C = 5$ | $S_C = 6$ | $S_C = 2$ | $S_C = 1$ |
| 3 | $S_C = 0$ | $S_C = 5$ | $S_C = 3$ | $S_C = 6$ | $S_C = 2$ | $S_C = 7$ | $S_C = 1$ | $S_C = 4$ |
| 4 | $S_C = 0$ | $S_C = 4$ | $S_C = 1$ | $S_C = 5$ | $S_C = 6$ | $S_C = 2$ | $S_C = 7$ | $S_C = 3$ |
| 5 | $S_C = 0$ | $S_C = 2$ | $S_C = 5$ | $S_C = 7$ | $S_C = 1$ | $S_C = 3$ | $S_C = 4$ | $S_C = 6$ |
| 6 | $S_C = 0$ | $S_C = 7$ | $S_C = 6$ | $S_C = 1$ | $S_C = 3$ | $S_C = 4$ | $S_C = 5$ | $S_C = 2$ |

## 6.4.4.3        Rates and puncturing map

Seven code rates are defined for the Turbo mode: $R = 1/3, 2/5, 1/2, 2/3, 3/4, 4/5, 6/7$. This is achieved through selectively deleting the parity bits (puncturing). The puncturing patterns of table 11 are applied. These patterns are identical for both codes $C_1$ and $C_2$ (deleting is always done in couples) and are repeated an integer or fractional number of times, as appropriate. The puncturing rate is indicated to the RCSTs via the TCT (see clause 8.5.5.4).

Rates 1/3, 2/5, 1/2, 2/3 and 4/5 are exact ones, independently of the block size. Rates 3/4 and 6/7 are exact ones only if $N$ is a multiple of 3. In other cases, the actual rates are very slightly lower than the nominal ones.

Depending on the code rate, the length of the encoded block is:

- $2N + M$ for $R < 1/2$, with:
  - $M = N$ for $R = 1/3$;
  - $M = N/2$ for $R = 2/5$.

- $N + M$ for $R \ge 1/2$, with:
  - $M = N$ for $R = 1/2$;
  - $M = N/2$ for $R = 2/3$;
  - for R = 3/4.

- $M = N/3$ (if $N$ mod. 3 = 0); or

- $M = (N - 4) / 3 + 2$ (if $N$ mod. 3 = 1); or

- $M = (N - 8) / 3 + 3$ (if $N$ mod. 3 = 2).
  - $M = N/4$ for $R = 4/5$;
  - for R = 6/7.

- $M = N/6$ (if $N$ mod. 3 = 0); or

- $M = (N - 4) / 6 + 1$ (if $N$ mod. 3 = 1); or

- $M = (N - 8) / 6 + 2$ (if $N$ mod. 3 = 2).

**Table 11: Puncturing patterns for Turbo codes "1" = keep**

$$1/3 \frac{Y}{W}\begin{bmatrix}1\\1\end{bmatrix} \qquad 2/5 \frac{Y}{W}\begin{bmatrix}1 & 1\\1 & 0\end{bmatrix}$$

$$1/2 \frac{Y}{W}\begin{bmatrix}1\\0\end{bmatrix} \qquad 2/3 \frac{Y}{W}\begin{bmatrix}1 & 0\\0 & 0\end{bmatrix}$$

$$3/4 \frac{Y}{W}\begin{bmatrix}1 & 0 & 0\\0 & 0 & 0\end{bmatrix} \qquad 4/5 \frac{Y}{W}\begin{bmatrix}1 & 0 & 0 & 0\\0 & 0 & 0 & 0\end{bmatrix}$$

$$6/7 \frac{Y}{W}\begin{bmatrix}1 & 0 & 0 & 0 & 0 & 0\\0 & 0 & 0 & 0 & 0 & 0\end{bmatrix}$$

## 6.4.4.4       Order of transmission and mapping to QPSK constellation

Two orders of transmission are allowed:

- in the natural order, all couples (A,B) are transmitted first, followed by all couples $(Y_1,Y_2)$ that remain after puncturing and then all couples $(W_1,W_2)$ that remain after puncturing (see figure 17);

- in the reverse order, the couples $(Y_1,Y_2)$ are transmitted first, in their natural order, followed by the couples $(W_1,W_2)$, if any, and then finally by the couples (A,B).

Each couple is mapped to one QPSK constellation point as shown in figure 22. In figure 17, the row with the A symbols is mapped on the I channel ($C_1$ in figure 22).

The order of transmission is signalled by the NCC as an inner code parameter in the Time Slot Composition Table (see clause 8.5.5.4).

n

| | 0 | 2 | | N-2 | N | N+2 | | 2N-2 | 2N | 2N+2 | | 2N+M-2 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| R<1/2 | A A A | | // | A A | $Y_1$ $Y_1$ $Y_1$ | | // | $Y_1$ $Y_1$ | $W_1$ $W_1$ $W_1$ | | // | $W_1$ $W_1$ |
| | B B B | | | B B | $Y_2$ $Y_2$ $Y_2$ | | | $Y_2$ $Y_2$ | $W_2$ $W_2$ $W_2$ | | | $W_2$ $W_2$ |
| | 1 | | // | N-1 | N+1 | | // | 2N-1 | 2N+1 | | // | 2N+M-1 |

n

| | 0 | 2 | | N-2 | N | N+2 | | N+M-2 |
|---|---|---|---|---|---|---|---|---|
| R≥1/2 | A A A | | // | A A | $Y_1$ $Y_1$ $Y_1$ | | // | $Y_1$ $Y_1$ |
| | B B B | | | B B | $Y_2$ $Y_2$ $Y_2$ | | | $Y_2$ $Y_2$ |
| | 1 | | // | N-1 | N+1 | | // | N+M-1 |
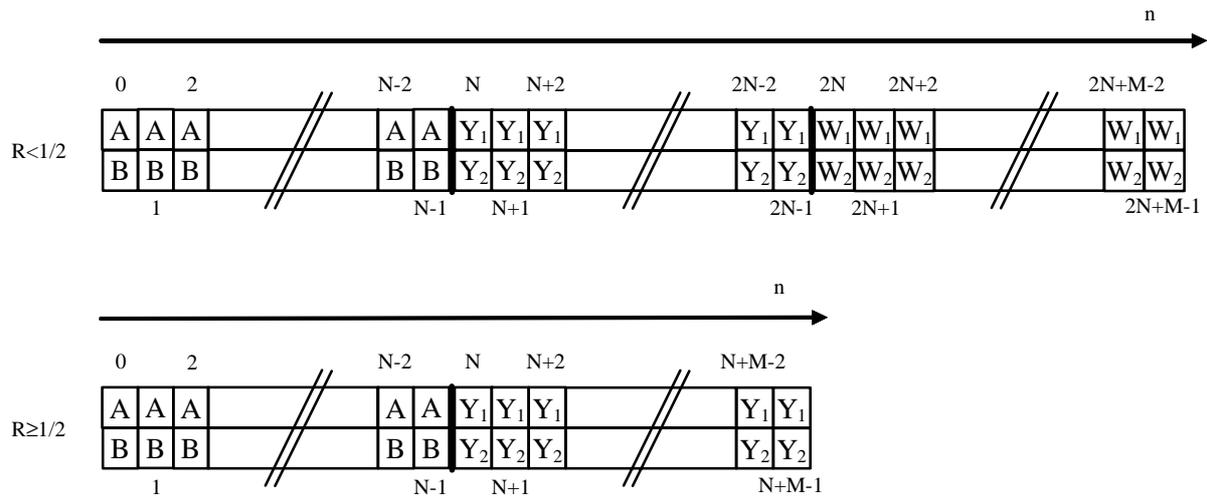
**Figure 17: Encoded blocks (natural order)**

## 6.4.5 Countermeasures for non-line-of-sight channels (Mobile Opt)

Transmissions of multicast and unicast traffic data can be protected against channel impairments such as short interruptions and shadowing by the inclusion and processing of additional coding in accordance with the provisions of this clause. The technique employed is called Link Layer Forward Error Correction (LL-FEC). RCSTs that declare support for Non-Line-Of-Sight (NLOS) countermeasures in the CSC burst shall be able to receive and process a forward link signal transmitted in accordance with these provisions. This technique can also be applied to the optional continuous return link carrier transmissions defined in clause 10.

LL-FEC is introduced to support reception in situations of high Packet Loss Ratio (PLR) at the MPE section level. Such high PLR may occur for example on mobile channels when the speed is too high and/or the signal-to-noise ratio is too low. It may also occur due to obstruction, blockage, or other situations in which the line of sight is interrupted. With the LL-FEC, a variable amount of capacity is allocated to parity overhead.

Transmissions employing LL-FEC use the same basic data structures as other MPE transmissions. However, since certain fields in the headers are interpreted in different ways, the possibility exists that an LL-FEC header may be interpreted as a regular DSM-CC header, addressed to a terminal that does not support LL-FEC. To avoid this, transmissions using LL-FEC should not use the same elementary streams as regular MPE transmissions.

The use of LL-FEC is optional and is defined separately for each elementary stream in the transport stream. Each elementary stream may configure different code parameters, resulting in different delays, level of protection and FEC overhead. Systems employing Generic Stream Encapsulation (GSE) can use the provisions in clause 6.4.5.7 for encapsulation of applications and parity data. LL-FEC carried over GSE is defined separately from LL-FEC carried over MPE. GSE is defined to be carried over Generic Streams while MPE is defined to be carried over transport streams. As an interim solution, an LL-FEC identifier descriptor referring to LL-FEC carried over GSE can be transmitted over a transport stream.

LL-FEC can use the Raptor code as specified in annex C of [26] for LL-FEC frame ADT sizes up to 12 Mbytes or the MPE-FEC Reed-Solomon code as specified in clause 9.5.1 of [4] any LL-FEC frame ADT sizes up to 191 Kbytes. The chosen code is identified in the forward link signalling.

For the purpose of the present clause, the following definitions shall apply.

**Datagram:** A network layer (OSI-layer 3) data frame. In the case of Internet Protocol, a datagram is an IP datagram.

**GSE-FEC Stream:** A sequence of GSE packets with the same gse_fec_id identifier.

**LL-FEC:** Method to deliver parity data codes for datagrams delivered either on Multi-Protocol (MPE) sections or GSE packets.

**LL-FEC Frame:** The collection of data and parity sections/packets of one elementary stream/GSE-FEC stream with identical fec_frame_number.

**LL-FEC Frame Application Data Table:** The collection of data sections/packets of one elementary stream/GSE-FEC stream with identical fec_frame_number. It also defines the mapping of the respective datagrams to the LL-FEC Frame.

**LL-FEC Frame FEC Data Table:** The collection of parity sections/packets of one elementary stream/GSE-FEC stream with identical fec_frame_number. It also defines the generation of parity symbols for the LL-FEC Frame.

**Receiver:** The receiver is an entity within an RCST, consisting of Radio Frequency front-end, channel decoding and demultiplexing. Input to a Receiver is an RF signal, and the output is Network layer datagrams.

## 6.4.5.1    LL-FEC Frame

The LL-FEC frame is a conceptual construction used to generate LL-FEC parity sections from a sequence of layer 3 datagrams. It is composed of the ADT and the FDT. The LL-FEC frame shall conceptually be arranged as a matrix with a flexible number of columns for both the ADT and the FDT. The maximum number for no_adt_columns and no_fdt_columns depend on the type of code used. The no_adt_columns is signaled in each parity section/packet transmitted along with this LL-FEC frame. The no_fdt_columns is not explicitly signalled for Raptor, but is signalled for the Reed-Solomon code. The matrix has a flexible number of rows with a maximum that depends on the type of code used. Figure 18 shows the conceptual organisation of the frame.

The number of rows is signalled in the LL-FEC identifier descriptor (clause 8.5.5.10.24). Each position in the matrix can hold an information byte. The left part of the LL-FEC Frame is used for OSI layer 3 (Network layer) datagrams (e.g. IP datagrams) and possible padding, and is called the Application Data Table. The right part of the LL-FEC Frame is dedicated for the parity information of the FEC code and is called the FEC Data Table (FDT). The number of columns in the ADT and FDT can vary frame-by-frame.
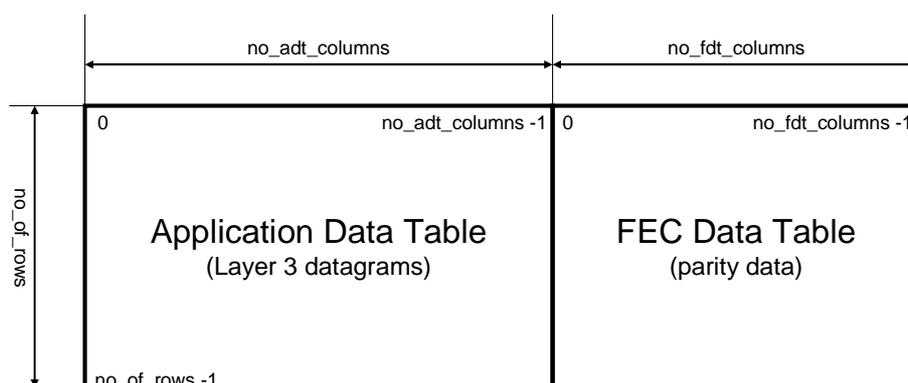


**Figure 18: LL-FEC Frame**

### 6.4.5.1.1        Filling of Application Data Table

Layer 3 datagrams shall be inserted consecutively, starting with the first byte of the first datagram in the upper left corner of the ADT matrix; going downwards in the first column and wrapping to the next column when the last row in a column has been filled. The length of the datagrams may vary.

Insertion of the datagrams depends on the addressing granularity, which is signalled implicitly through the frame_size parameter in the LL-FEC identifier descriptor (see clause 8.5.5.10.24). The process is illustrated in figure 19.
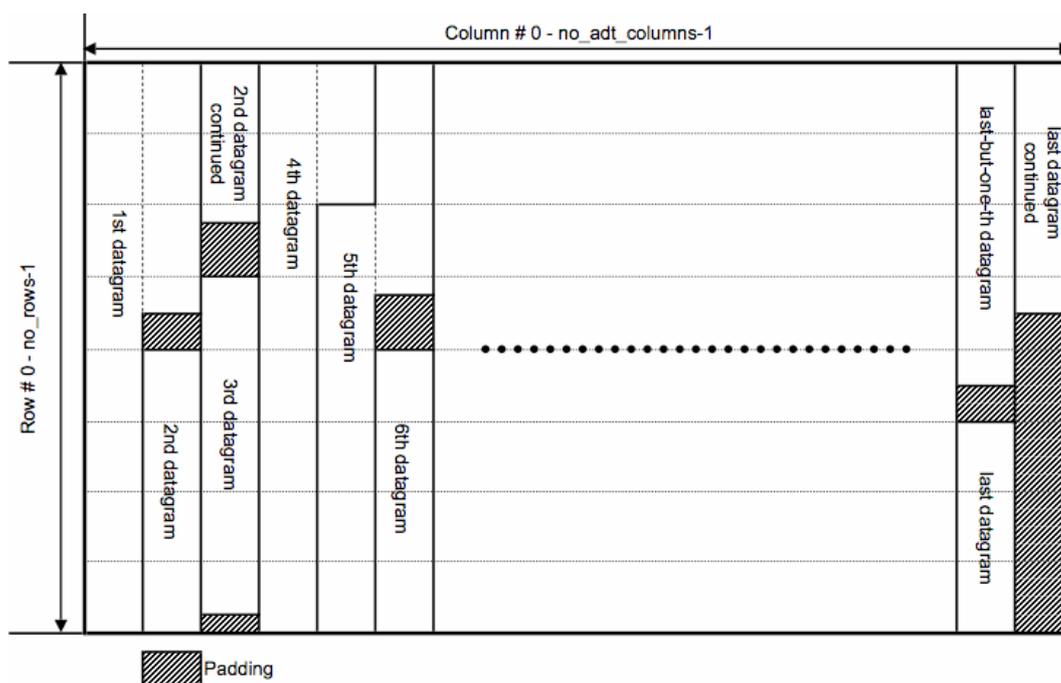
**Figure 19: Application data table**

Each layer 3 datagram shall be assigned a unique address within the LL-FEC ADT table. Zero-padding bytes are inserted, if necessary, in the last column of the ADT to fill the column completely. The last column must contain at least one byte of a layer 3 datagram.

For addressing granularity equal to 1, datagrams are inserted in the ADT consecutively and without any padding.

When the addressing granularity is greater than 1, each layer 3 datagram is inserted in the ADT as follows: its first byte must be inserted at the next ADT address which is an integer multiple of the address granularity. Any bytes between the last byte of the previous layer 3 datagram and the first byte of the new layer 3 datagram in the ADT must be filled with zeros. Each layer 3 datagram gets assigned a unique address within the LL-FEC ADT table such that the address is an integer multiple of the address granularity.

Signalling of parameters associated with each individual datagram is defined in clause 6.4.5.6.

## 6.4.5.1.2    Generation of FEC Data Table

Once the ADT is filled, parity data columns for the FDT can be computed by applying the selected coding technique. The decision on the completeness of an ADT table is implementation and/or system specific and not within the scope of the present document. It may depend on latency consideration, the LL-FEC code rate and other parameters. However, the transmitter shall ensure that the difference in time between the transmission of the first and last sections/packets within a given LL-FEC frame does not exceed the buffer_timeout signalled in the LL-FEC identifier descriptor.

### 6.4.5.1.2.1    Reed-Solomon code

The Reed-Solomon code shall be that specified in clause 9.5.1 of [4]. The maximum no_adt_columns in this case is 191 and the maximum no_fdt_columns is 64. In case no_adt_columns is less than 191, the ADT shall be extended with 191-adt_columns zero columns and code shortening as specified in clauses 9.3.3.1 of [4] shall be applied. In case no_fdt_columns is less than 64, the last 64-no_fdt_columns shall be punctured as specified in clause 9.3.3.2 of [4].

The LL-FEC frame shall be constructed in the same manner as the MPE-FEC frame defined in clause 9.3.1 of [4]. The correspondence between the MPE-FEC frame elements of [4] and the LL-FEC Frame elements is the following:

- The FDT is equivalent to Reed-Solomon Data Table (RSDT) defined in [4];

- Time-slicing as defined in [4] shall not be used;

### 6.4.5.1.2.2          Raptor code

The systematic Raptor encoding procedure in [26], clause C.4 shall be applied. The maximum no_adt_columns in this case is 8 192 and the maximum no_fdt_columns is 65 536-no_adt_columns. The encoding procedure shall be applied in such a way that the ADT with no_adt_columns corresponds to the source block with no_adt_columns source symbols and each column of the ADT corresponds to a source symbol. In case no_adt_columns is less than 4, the ADT column shall be extended with 4-adt_columns zero columns and code shortening as specified in clauses 9.3.3.1 of [4] shall be applied.

The FDT is defined as the consecutive encoding symbols of the Raptor codes, whereby the first FDT column corresponds to the encoding symbol ID (ESI) no_adt_columns. Each row of the FDT thus contains exactly one Raptor symbol. The sub-blocking option specified in [26] shall not be applied.

The number of FDT columns shall be at most 65 536 minus the number of ADT columns.

NOTE 1:   Raptor symbols that are not transmitted need not be generated; therefore, puncturing is generally not necessary. The no_fdt_columns is not signalled to the receiver.

NOTE 2:   For each LL-FEC frame at the receiver, the decoder needs:

- The number of ADT columns, no_adt_columns for this FEC frame which corresponds to the Raptor source block size as long as no_adt_columns $\geq$ 4. Note that no_adt_columns may change for every LL-FEC frame

- The Source Block Number (SBN), equivalent to the fec_frame_number.

- In addition, the decoder needs for each received encoding symbol the encoding symbol id (ESI). The mapping of the ESI signalling to the Raptor parity data is specified in clause 6.4.5.4.

## 6.4.5.2          Carriage of LL-FEC frames

The datagrams shall be encapsulated in datagram sections which are compliant to the DSM-CC section format for private data [14]. These sections are referred as MPE sections, since they follow the data transport specification defined in clause 7 of [4] for Multiple Protocol Encapsulation (MPE). The ADT columns are carried in accordance with clause 6.4.5.3. The FDT columns are carried in LL-FEC sections as specified in clause 6.4.5.4.

As an alternative, systems employing Generic Stream Encapsulation (GSE) can use the provisions in clause 6.4.5.7 for encapsulation of Layer 3 datagrams and the LL-FEC frame FDT columns. The GSE packets carrying the Layer 3 datagrams and the FDT columns from the same LL-FEC Frame shall be carried in one GSE-FEC stream. Each GSE packet carried within this GSE-FEC stream shall be marked with the same gse_fec_id identifier as specified in the corresponding LL-FEC identifier descriptor. The gse_fec_id shall be carried in the LL-FEC extension header specified in clause 6.4.5.7.

The first section carrying data of a given LL-FEC Frame shall be the MPE section carrying the Application data datagram at address "0". All sections carrying Application datagrams of a given LL-FEC Frame shall be transmitted prior to the first section carrying parity data of the LL-FEC Frame. Within an elementary stream, all sections carried between the first and last section of an LL-FEC Frame shall carry data belonging to that LL-FEC Frame. Within an elementary stream, sections delivering data of different LL-FEC Frames shall not be interleaved.

The section following the last section carrying Application data datagram on an LL-FEC Frame, shall contain either the first section carrying the parity data of the same LL-FEC Frame, or the first Application data section of the next LL-FEC Frame. In the latter case, parity data of the first LL-FEC Frame are not transmitted.

Padding shall not exist between delivered Application data in the Application data table; i.e. any padding inserted for the purpose of computing parity data shall be removed prior to transmission. Datagrams shall not overlap in an Application data table. Padding shall not exist between delivered parity data in the parity data table.

## 6.4.5.3          Carriage of application data in transport streams

The provisions in this clause apply to each elementary stream for which the ll_fec_identifier_descriptor indicates that LL-FEC is used:

- Each LL-FEC Frame shall only contain complete datagrams (i.e. datagrams shall not be fragmented between LL-FEC Frames).

- At least one MPE section shall be delivered in each LL-FEC Frame.

- The section_syntax_indicator in each MPE section on the elementary stream shall be set to "1", indicating that CRC_32 is used at the end of the section.

- The datagrams within the elementary stream shall be transmitted in the payload of MPE sections, encapsulated in accordance with clause 7.1 of [4].

- The MAC_address_1 … MAC_address_6 bytes in the header of each MPE section delivered in the elementary stream using LL-FEC shall carry valid real time parameters as defined in clause 6.4.5.6 as well as the NLOS_RCST_ID defined in clause 6.4.5.5. The mapping of these parameters into the MAC_address bytes is defined in figure 20.

The method for carrying application data in GSE packets is defined in clause 6.4.5.7.
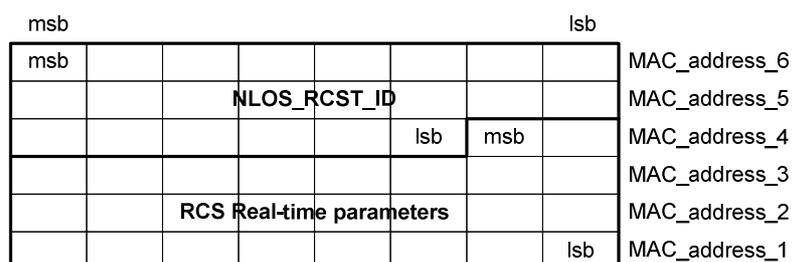


**Figure 20: Mapping of real-time parameters to MAC_address bytes**

## 6.4.5.4        Carriage of parity data in transport streams

The parity data of each LL-FEC Frame shall be delivered in LL-FEC sections as defined in clause 8.5.5.14. LL-FEC sections shall be carried in the same elementary stream as the corresponding Application data.

LL-FEC sections are compliant to the DSMCC_section Type "User private" (see ISO/IEC 13818-6 [14]). The mapping of the section into MPEG-2 Transport Stream packets is defined in MPEG-2 Systems ISO/IEC 13818-1 [7].

### 6.4.5.4.1        Carriage of Reed-Solomon code parity data

When Reed-Solomon codes are used, MPE-FEC real-time parameters as defined in clause 9.10 of [4] shall be carried in the RCS real time parameters in accordance with table 12. The RCS real time parameters are defined in clause 6.4.5.6.

**Table 12: Mapping of real time parameters between MPE-FEC and LL-FEC**

| MPE-FEC (Clause 9.10 of [4]) | LL-FEC (Clause 6.4.5.6) | Comments |
|---|---|---|
| delta_t (5 lsb) | fec_frame_number | Only 5 lsb carried in LL-FEC |
| delta_t (7 msb) | - | 7 msb not carried in LL-FEC |
| table_boundary | table_boundary | |
| frame_boundary | - | Not carried in LL-FEC |
| address | dt_position (18 lsb) | 18 bits mapped into the dt_position lsb's |
| - | dt_position (2 msb) | Bits set to "00" |

### 6.4.5.4.2        Carriage of Raptor code parity data

Each LL-FEC section shall carry at least one encoding symbol, but may carry several consecutive columns of the FEC Data Table. This corresponds to a repair symbol group associated with an encoding_symbol_id (ESI) and an integer number of encoding symbols. The number of encoding symbols in each FEC section can be determined by dividing the section payload length by the number of rows in the LL-FEC frame. The length of a column, i.e. the number of rows, is signalled in the frame_size field of the corresponding ll_fec_identifier_descriptor.

The encoding_symbol_id (16 bit) of the first encoding symbol in the section shall be signalled in the section header.

LL-FEC sections shall only carry encoding symbols with a value of encoding_symbol_id greater than or equal to no_adt_columns and smaller than 65 536.

## 6.4.5.5        Non-line-of-sight RCST identifier

This 22-bit field is a unique identifier which shall be locally or statically configured in each NLOS-capable RCST in the network. It can be used for filtering of the received data in place of the conventional MAC address, which is not available due to the carriage of the real-time parameters. A value of 0x3FFFFF shall be used to indicate broadcast or multicast. This value shall be used for sections carrying parity data not exclusively directed to a single RCST. The nlos_rcst_id shall be configurable in the RCST. It is recommended that the default value of this identifier be equal to the lower 22 bits of the RCST's MAC address;

## 6.4.5.6        Real-time parameters

In streams where LL-FEC is used, each MPE section and LL-FEC section shall carry real time parameters described in table 13. This shall also apply to the corresponding packets in GSE-FEC streams.

**Table 13: Real time parameters**

| Syntax | No. of bits | Identifier |
|---|---|---|
| rcs_real_time_parameters () { | | |
| table_boundary | 1 | bslbf |
| fec_frame_number | 5 | bslbf |
| dt_position | 20 | bslbf |
| } | | |

Semantics for rcs_real_time_parameters:

- **table_boundary:** This 1-bit flag, when set to "1", indicates that the current section is the last section of a table within the current LL-FEC Frame. If the section is an MPE section, this flag indicates the last section of Application data table.

    NOTE:    A decoder not supporting MPE-FEC may ignore all subsequent sections until the end of the LL-FEC Frame. The table_boundary may also be used to ignore any upcoming LL-FEC parity data in case no loss has been detected in the LL-FEC Frame ADT. Finally, the table-boundary should be used by receivers to insert padding in the last column of the LL-FEC Frame ADT.

- **fec_frame_number:** The field supports a cyclic LL-FEC Frame index within the elementary stream. The value of the field increases by one for each subsequent LL-FEC Frame. After value "11111", the field restarts from "00000". This field can be used to resolve ambiguities resulting from long sequences of lost data.

- **dt_position:** This 20-bit field specifies the position in the corresponding LL-FEC Frame table of the first byte of the payload carried within the section. In case the layer 3 datagram is fragmented over multiple MPE sections, each MPE section indicates the **dt_position** in the Application data table of the first byte of the datagram fragment carried within the section. All sections delivering data for any LL-FEC Frame table shall be delivered in ascending order according to the value of this field. The dt_positon is derived by dividing the address by the address granularity.

The byte position is a zero-based linear address within an LL-FEC Frame ADT, starting from the first row of the first column, and increasing towards the end of the column. At the end of the column, the next byte position is at the first row of the next column.

For each LL-FEC Frame, exactly one MPE section shall be transmitted with dt_position field set to value "0".

For each LL-FEC Frame for which RS parity data is transmitted as specified in clause 6.4.5.4.1, exactly one LL-FEC section shall be transmitted with dt_position field set to value "0".

For each LL-FEC Frame for which Raptor parity data is transmitted as specified in clause 6.4.5.4.2, the dt_position field shall be a reserved field and shall be set to "0xFFFFF".

## 6.4.5.7        Provisions for carrying LL-FEC frames in generic streams

This clause provides a mechanism for carriage of LL-FEC information in systems employing Generic Stream Encapsulation and an interim method for signalling of LL_FEC in such systems. The interim signalling method shall be superseded by provisions that use a signalling mechanism designed specifically for GSE, once such a mechanism is available.

When using this form of encapsulation, the stream_type_indication flag in the ll_fec_identifier_descriptor shall be set to "1". Encapsulation of applications data and parity data into GSE PDUs shall be in accordance with [25].

### 6.4.5.7.1        LL-FEC identifier location for GSE-FEC streams

In the absence of a dedicated signalling framework for GSE streams, the ll_fec_identifier_descriptor shall be used to signal information about LL-FEC for GSE streams. This descriptor shall be located in a transport stream which can be either a native TS or a TS carried over GSE. The latter method is defined in [25].

### 6.4.5.7.2        Carriage of application data over GSE-FEC streams

The following provisions apply to each GSE-FEC stream for which the ll_fec_identifier_descriptor indicates that LL-FEC is used:

- The application data packets shall be encapsulated in accordance with [25]. There shall be no padding between applications data; i.e. any padding inserted for the purpose of computation of parity data shall be removed prior to transmission.

- Real-time parameters and identification of the LL-FEC process shall be carried in an optional extension header as defined in clause 6.4.5.7.2.1.

- Each LL-FEC Frame shall only contain complete datagrams (i.e. datagrams shall not be fragmented between LL-FEC Frames).

- For each LL-FEC Frame, at least one GSE packet carrying application data shall be delivered.

- The first packet carrying data of a given LL-FEC Frame shall be the GSE packet carrying the Application data datagram at address "0".

- All packets carrying Application data datagrams of a given LL-FEC Frame shall be transmitted prior to the first packet carrying parity data of the LL-FEC Frame (i.e. packets carrying Application data datagrams shall not be interleaved with packets carrying parity data within a single LL-FEC frame).

- Within a GSE-FEC stream, all packets carried between the first and the last packet of an LL-FEC Frame shall carry the data belonging to the LL-FEC Frame (i.e. only GSE packets carrying datagrams and LL-FEC packets carrying parity data are allowed).

- Within a GSE-FEC stream, packets delivering data of different LL-FEC Frames shall not be interleaved.

- When the layer 3 datagram needs to be divided over multiple GSE packets, the optional extension header as defined in clause 6.4.5.7.2.1 shall be carried only in the GSE packet carrying the first datagram fragment and shall indicate the dt_position in the Application data table of the first byte of the datagram.

- Additional reliability information for the reception process may be obtained by applying the NLOS adaptation optional extension header defined in clause 6.4.5.7.2.2.

6.4.5.7.2.1                    GSE-FEC application data optional header extension

The GSE optional extension header for carrying application data shall be referred to as LL_RCS_FEC_ADT and is defined in table 14.

**Table 14: GSE optional header extension for carrying application data**

| Syntax | No. of bits | Identifier |
|---|---|---|
| LL_RCS_FEC_ADT () { | | |
| Reserved | 2 | bslbf |
| gse_fec_id | 14 | uimsbf |
| reserved_for future_use | 6 | bslbf |
| rcs_real_time_parameters () | 26 | See semantics |
| } | | |

Semantics for LL_RCS_FEC_ADT:

- **reserved:** Shall be set to "11".

- **gse_fec_id:** This 14-bit field shall refer to a LL-FEC Frame that has been defined with a LL-FEC identifier descriptor using the same gse_fec_id value, assuming that stream_type field in the descriptor has been set to "1". This field shall be used to differentiate the GSE-FEC streams by their corresponding LL-FEC Frame. It can also be used for filtering.

- **reserved_for_future_use:** This 8-bit field shall be set to "11111111".

- **rcs_real_time_parameters:** This 26-bit field carries real-time parameters for the application data. The details are specified in clause 6.4.5.6.

The presence of an optional extension header is defined by using an invalid protocol_type with a value lower than 0x600. The protocol_type field can either be in the main GSE header or after an optional header as specified in [25]. The 16-bit optional header type field carried in the protocol_type field is formed as defined in table 15.

**Table 15: GSE optional header extension type definition**

| Syntax | No. of bits | Identifier |
|---|---|---|
| optional_extension_header_type () { | | |
| start_indicator | 5 | bslbf |
| header_length | 3 | bslbf |
| optional_header_type | 8 | uimsbf |
| } | | |

Semantics for optional_extension_header_type:

- **start_indicator:** This 5-bit field shall be set to a value of "00000".

- **header_length:** This 3-bit field specifies the length of the optional header, which allows receivers ignorant of certain optional header type to skip the header and still be able to decode the GSE payload. This shall be set to "100", indicating a 6-byte header length as defined in [25].

- **optional_header_type:** This 8-bit field uniquely identifies this optional extension header; its value shall be as defined in [27].

6.4.5.7.2.2                    NLOS adaptation optional header extension

The optional extension header defined in this clause may be used for LL-FEC frames carried over GSE-FEC streams. Its purpose it to improve performance. This extension header shall be referred to as LL_CRC32 and is described in table 16. This extension header may be used only in GSE packets carrying a non-fragmented layer 3 datagram.

**Table 16: GSE CRC-32 optional header extension**

| Syntax | No. of bits | Identifier |
|---|---|---|
| CRC_32_extension () { | | |
|    LL_CRC32 | 32 | rpchof |
| } | | |

The LL_CRC32 field shall be computed over all bytes be starting from the GSE Length field (included) to the end of the GSE packet, but not including the CRC extension header fields. The computation method shall otherwise be equivalent to that defined in clause 4.2.2 of [25].

The header type definition for the CRC_32_extension shall use the syntax defined in table 15, with the following semantics:

- **start_indicatior:** This 5-bit field shall be set to a value of "00000".

- **header_length**: This 3-bit field specifies the length of the optional header, which allows receivers ignorant of certain optional header type to skip the header and still be able to decode the GSE payload. This shall be set to "011", indicating a 4-byte header length as defined in [25].

- **optional_header_type:** "This 8-bit field uniquely identifies this optional extension header; its value shall be as defined in [27].

### 6.4.5.7.3       Carriage of parity data over GSE-FEC streams

Parity data and its associated real-time parameters shall be carried in GSE packets as defined in this clause. This packet format defines a mandatory extension header. This header shall be referred to as LL_RCS_FEC_FDT.

NOTE 1:   The use of a mandatory extension header ensures that receivers that do not support LL-FEC will discard the entire packet, in accordance with [25].

When carrying raptor Code parity data, each PDU shall carry exactly one repair symbol or group of repair symbols, i.e. one FDT column or a group of several consecutive FDT columns.

When carrying Reed-Solomon parity data, each PDU shall carry one FDT column.

The packet format shall be in accordance with table 17.

NOTE 2:   The encapsulation_type meta-variable is not carried explicitly in this packet. The pertinent value is defined in the LL_FEC_identifier_descriptor entry (clause 8.5.5.10.24) that applies to the LL-FEC data being transported.

**Table 17: GSE packet format for parity data**

| Syntax | No. of bits | | Information Mnemonic |
|---|---|---|---|
| | Reserved | Information | |
| GSE_packet () { | | | |
|   start_indicator | | 1 | bslbf |
|   end_indicator | | 1 | bslbf |
|   label_type_indicator | | 2 | bslbf |
|   gse_length | | 12 | uimsbf |
|   if ((start_indicator =="0") OR (end_indicator ==0)) { | | | |
|     frag_id | | 8 | uimsbf |
|   } | | | |
|   if ((start_indicator == "1") AND (end_indicator == "0")) { | | | |
|     total_length | | 16 | uimsbf |
|   } | | | |
|   if (start_indicator == "1") { | | | |
|     protocol_type | | 16 | uimsbf |
|     gse_fec_id | 2 | 14 | uimsbf |
|     reserved_for_future_use | 6 | | bslbf |
|     if (encapsulation_type == 0) { | | | |
|       padding_columns | | 8 | uimsbf |
|       column_number | | 8 | uimsbf |
|       last_column_number | | 8 | uimsbf |
|       rcs_real_time_parameters() | | 26 | See semantics |
|     } | | | |
|     else { | | | |
|       no_adt_columns | 3 | 13 | uimsbf |
|       encoding_symbol_id | | 16 | uimsbf |
|       fec_frame_number | 13 | 5 | uimsbf |
|     } | | | |
|   } | | | |
|   for (i=0; i<N; i++) { | | | |
|     parity_data_byte | | 8 | bslbf |
|   } | | | |
|   if ((start_indicator=="1") AND (end_indicator == "1") { | | | |
|     parity_data_crc_32 | | 32 | rpchof |
|   } | | | |
|   if ((start_indicator=="0") AND (end_indicator == "1") { | | | |
|   crc_32 | | | |
|   } | | | |
| } | | | |

Semantics for GSE_packet:

- **start_indicator:** This field shall be interpreted as defined in [25].

- **end_indicator:** This field shall be interpreted as defined in [25].

- **label_type_indicator:** This is a 2-bit field. It shall be set to "10" to indicate that no label field is present.

- **gse_length:** This field shall be interpreted as defined in [25].

- **frag_id:** This field shall be interpreted as defined in [25].

- **total_length:** This field shall be interpreted as defined in [25].

- **protocol_type:** This 16-bit field shall indicate the presence of a LL_RCS_FEC_FDT mandatory extension. As defined in [i.1], the 8 MSB shall be set to "0000 0000" and the 8 LSB, corresponding to the header type field, shall uniquely identify this mandatory extension header. The value of this field shall be as defined in [27].

- **gse_fec_id:** This 14-bit field shall refer to a LL-FEC Frame that has been defined with a LL-FEC identifier descriptor using the same gse_fec_id value, assuming that stream_type field in the descriptor has been set to "1". This field shall be used to differentiate the GSE-FEC streams by their corresponding LL-FEC Frame. It can also be used for filtering.

- **reserved_for future_use:** These six bits shall be set to "111111".

- **padding_columns:** This 8-bit field indicates the number of full columns of the Application data table of the actual LL-FEC Frame filled with padding bytes only; i.e. it is equal to 191 minus the number of columns that contain Application data. The value indicated shall be from 0 to 190. The value may vary frame by frame.

- **column_number:** This 8-bit field gives the number of the FDT column containing the RS data. The column_number of the first column carrying RS data of an LL-FEC Frame shall be 0x00. The column_number shall be incremented by 1 with each additional column containing RS data of the concerned LL-FEC Frame.

- **last_column_number:** This 8-bit field shall indicate the number of the last FDT column that contains RS data of the current LL-FEC Frame.

- **rcs_real_time_parameters:** This 26-bit field carries RCS Real-time parameters for the FEC parity data. The details are specified in clause 6.4.5.6.

- **no_adt_columns:** This 13-bit field indicates the number of columns of the Application data table of the actual FEC Frame. The value indicated shall be from 0 to 8 191. The value may vary frame by frame.

- **encoding_symbol_id:** This 16-bit field specifies the ESI of the first encoding symbol of this section. The ESI shall be greater than or equal to *no_adt_columns* for the LL-FEC Frame;

- **fec_frame_number:** This field shall be interpreted as defined in clause 6.4.5.6.

- **parity_data_byte:** These bytes contain the parity data.

- **parity_data_crc_32:** This 32-bit field shall be computed over all bytes be starting from the GSE length field (included) to the end of the GSE packet, but not including the CRC field. The computation method shall be equivalent to that defined in clause 4.2.2 of [25].

- **crc_32:** This field shall be interpreted as defined in [25].

# 6.5    Modulation

The signal shall be modulated using QPSK with baseband shaping as described in clauses 6.5.1 to 6.5.4, except when the provisions for spectrum spreading in clause 6.5.5 are applied.

## 6.5.1    Bit mapping to QPSK constellation

Mapping into the QPSK constellation shall be as follows.

The preamble shall be configurable and indicated to the RCST through the TCT, as described in clause 8.5.5.4.

Immediately after the preamble insertion, the outputs C1 and C2 of the encoder shall be sent without modification to the QPSK bit mapper (see figure 21).
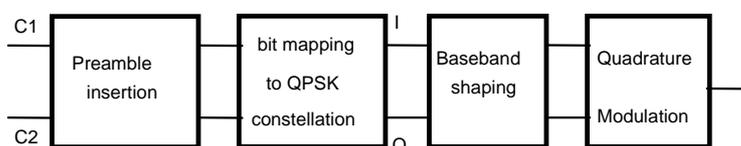


**Figure 21: Processing after the encoder**

Conventional Gray-coded QPSK modulation with absolute mapping (no differential coding) shall be used. Bit mapping in the QPSK constellation shall follow figure 22. If the normalization factor $1/\sqrt{2}$ is applied to the I and Q components, the corresponding average energy per symbol becomes equal to 1.
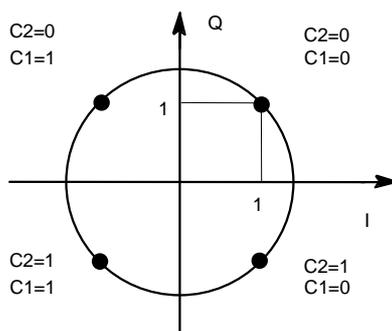
**Figure 22: Bit mapping into QPSK constellation**

The output C1 of the channel coding shall be mapped to the I channel of the modulation. The output C2 shall be mapped to the Q channel of the modulation.

Following the modulation process, the spectrum of the QPSK signal shall not be inverted.

## 6.5.2   Baseband shaping and quadrature modulation

Prior to modulation, the I and Q signals (mathematically represented by a succession of Dirac delta functions, multiplied by the amplitudes I and Q, spaced by the symbol duration TS = 1/RS) shall be square root raised cosine filtered. The roll-off factor shall be 35 % as indicated by the TCT (see clause 8.5.5.4). The baseband square root raised cosine filter shall have a theoretical function defined by the following expression:

$$
\begin{aligned}
H(f) &= 1 && \text{for} \quad |f| < f_N(1-\alpha) \\
H(f) &= \sqrt{\frac{1}{2} + \frac{1}{2}\sin\frac{\pi(f_N - |f|)}{2\alpha f_N}} && \text{for} \quad f_N(1-\alpha) \le |f| \le f_N(1+\alpha) \\
H(f) &= 0 && \text{for} \quad |f| > f_N(1+\alpha)
\end{aligned}
$$

where $f_N = \dfrac{1}{2T_s} = \dfrac{R_s}{2}$ is the Nyquist frequency and $\alpha$ is the roll-off factor.

At the RCST antenna output (using a large output back-off), the allowed spectrum and group delay variation shall be according to the mask given in EN 300 421 [1] for every return link symbol rate supported by the terminal.

## 6.5.3   EIRP control

The RCST shall be capable of adjusting the transmit EIRP in steps of nominally 0,5 dB over the operating range specified by the manufacturer. Over this range, the terminal output power change shall reflect a power adjust command to within 0,5 dB or to within 20 % in dB of the requested adjustment, whichever is less stringent (see clauses 8.5.5.9 and 8.5.5.10.3).

## 6.5.4   Guard time

Each burst is surrounded by a guard time which allows for RCST power switch-off transient and system timing errors. The guard time is allocated by the NCC as an implicit element of the TCT. The guard time is network dependant and is determined by overall system requirements. On optional MPEG2-TS TRF bursts, the guard time shall be shorter than half an MPEG2-TS packet.

## 6.5.5   Spectrum spreading in the return link (Mobile Opt)

The DVB-RCS return link MF-TDMA transmission can optionally be spread in bandwidth using the provisions in this clause. Spectrum spreading provisions for continuous-carrier return link transmissions are defined in clause 10.4.2.

The spectral spreading can be achieved by two means. The first consists of the use of π/2-BPSK modulation. This is equivalent to spreading a QPSK modulated signal by a factor 2. The signal can also be spread by burst repetition. The resulting signal shall be equivalent to that obtained by the processing flow shown in figure 23 and described further in this clause.
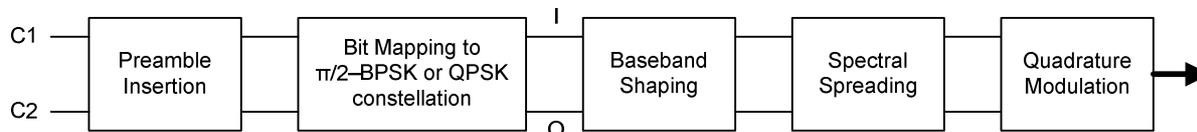


**Figure 23: Post-encoder processing with spectrum spreading**

When π/2-BPSK modulation is used, it shall employ absolute mapping (no differential coding). The bit-to-symbol mapping shall follow figure 24. If the normalization factor $1/\sqrt{2}$ is applied to the $I$ and $Q$ components, the corresponding average energy per symbol becomes equal to 1. C1 and C2 are alternatively transmitted on BPSK constellations that are π/2 shifted. The order of transmission is C1 first, C2 second, C1 third, etc.
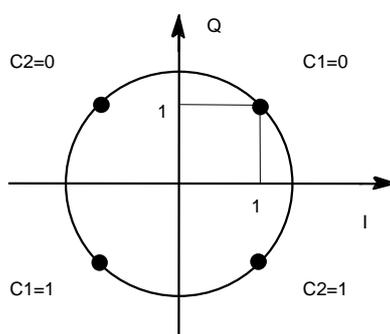


**Figure 24: Bit mapping to π/2-BPSK constellation for MF-TDMA transmissions**

Spectrum spreading by repetition shall be achieved in a manner equivalent to the following: After baseband shaping, the signal is organised in bursts containing $N$ complex samples $X_i$. This includes guard interval and ramp-up and ramp-down of the square root raised cosine filter. The input symbol rate is $R_s$. Each burst is repeated $F$ times in the time domain. The spread burst contains $F \times N$ complex samples $Y_j$, defined as follows:

$$Y_j = X_{j \bmod N}$$

At the output, the spread burst shall have the same duration as the non-spread burst. Therefore, the symbol rate is now equal to $F \times R_s$. It is this symbol rate which is signalled in the TCT table.

The spread burst is transmitted in one time slot as shown in figure 25. The composition of the time slot containing spread bursts is signalled in the Timeslot Composition Table (clause 8.5.5.4).
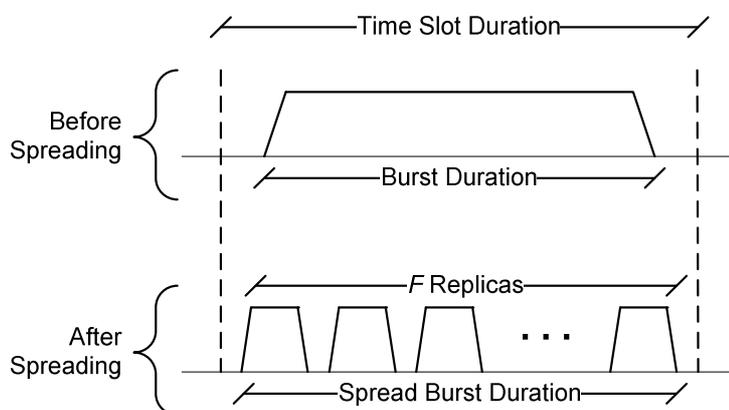


**Figure 25: Spectrum spreading by burst repetition**

# 6.6       MAC messages

All methods described below can be used by RCSTs for capacity requests and M&C messages. The four first methods are based on the Satellite Access Control (SAC) field and the last one allows encapsulation in ATM or MPEG-2 Data Units called Data Unit Labelling Method (DULM). One or more of the methods may be employed in a Satellite Interactive Network. For the particular implementation, the RCSTs are configured at the time of logon by the logon initialize descriptor (see clause 8.5.5.10.4) or mesh logon initialize descriptor (see clause 8.5.5.10.22) that is transmitted in a TIM.

## 6.6.1     Methods based on the Satellite Access Control (SAC) field

### 6.6.1.1       SAC field composition

The SYNC, the optional prefix attached to ATM TRF bursts and the optional MPEG TRF burst may contain the Satellite Access Control (SAC) field composed of signalling information added by the RCST for the purpose of requesting capacity on the session, or other additional MAC information.

The SAC is composed of optional sub-fields that are defined in table 18.

The SAC field configuration for specific SYNC (see clauses 6.6.1.3 and 6.6.1.4) or ATM TRF (see clause 6.6.1.2) time slots is signalled by the NCC in the TCT as described in clause 8.5.5.4.

The optional SAC field configuration of MPEG bursts is signalled directly within the TS packet of the specific MPEG TRF burst as defined in clause 6.6.1.5.

   NOTE:      The symbols and abbreviations, and the method of describing syntax used in the present document are the
                    same as those defined in clauses 2.2 and 2.3 of ISO/IEC 13818-1 [7].

**Table 18: Syntax of the SAC field**

| Syntax (see note) | No. of bits | | Information Mnemonic |
|---|---|---|---|
| | Reserved | Information | |
| SAC_field() { | | | |
|   if (Route_ID_flag == 0) | | | |
|     Route_ID | | 16 | uimsbf |
|   if (request_flag == 1) | | | |
|     for (i=0; i<=capacity_requests_number; i++) { | | | |
|       Capacity_Request { | | | |
|         Scaling_Factor | | 1 | bslbf |
|         Capacity_Request_Format | | 1 | bslbf |
|         Capacity_Request_Type | | 2 | bslbf |
|         If (Capacity_Request_Format == 0) { | | | |
|           Channel_ID | | 4 | uimsbf |
|           Capacity_Request_Value | | 8 | uimsbf |
|         } | | | |
|         If (Capacity_Request_Format == 1) { | | | |
|           Channel_ID | 4 | 8 | uimsbf |
|           Capacity_Request_Value | | 8 | uimsbf |
|         } | | | |
|       } | | | |
|     } | | | |
|     CR_Pad_Byte | | See text | uimsbf |
|   if (M_and_C_flag == 1) | | | |
|     M_and_C_Message | | 16 | bslbf |
|   if (Group_ID_flag == 1) | | | |
|     Group_ID | | 8 | uimsbf |
|   if (Logon_ID_flag == 1) | | | |
|     Logon_ID | | 16 | uimsbf |
|   if (ACM_flag == 0) { | | | |
|     ACM { | | | |
|       CNI | | 8 | uimsbf |
|       MODCOD_RQ | | 8 | bslbf |
|     } | | | |
|   } | | | |
|   if (MOB_flag ==0) | | | |
|     Mobility_Control_Message | | 32 | bslbf |
|   Pad_Bytes | | see text | uimsbf |
| } | | | |
| NOTE:  For SYNC and ATM TRF bursts, the sub-fields used in test statements (Route_ID_Flag, request_flag, capacity_requests_number, M_and_C_flag, Group_ID_flag, Login_ID_flag and ACM_flag) refer to the subfields of the specific timeslot of the TCT for which the RCST has to transmit a SAC field as defined in clause 8.5.5.4.<br>For MPEG TRF bursts carrying a SAC field, sub-fields used in test statements refer to the subfields of the SAC_composition of the Adaptation Field Private Data as defined in clause 6.6.1.5. | | | |

Semantics for the SAC field:

- **Route_ID:** This 16-bit field defines a destination forward (downlink) link that may be used for the prefixed payload in a regenerative system or to indicate a connectivity channel used in association with dynamic connectivity / QoS optimisation. Values are system dependent.

- **Capacity_Request:** Each capacity request is composed of the following sub-fields:

  - **Scaling_Factor:** This 1-bit sub-field defines the scaling factor of the Capacity_Request_Value sub-field (see table 19).

**Table 19: Scaling_Factor**

| Value | Scaling factor |
|-------|----------------|
| 0 | 1 |
| 1 | 16 |

- **Capacity_Request_Format:** This flag is set to 0 if the 4-bit Channel_ID format is used and to 1 if the 8-bit Channel_ID format is used.

- **Capacity_Request_Type:** This is a 2-bit sub-field specifying the category of capacity request (see table 20). The capacity categories are described in clause 6.8.

**Table 20: Capacity_Request_Type**

| Capacity_Request_Type Value | Capacity category | Units for Capacity_Request_Value |
|---|---|---|
| 00 | VBDC | Unit of payload size * scaling factor |
| 01 | RBDC | Unit of 2kbits/s * scaling factor |
| 10 | AVBDC | Unit of payload size * scaling factor |
| 11 | Reserved | |
| NOTE: | The payload size is either 53 bytes or 188 bytes according to the encapsulation mode defined at logon. | |

- **Channel_ID:** This is a 4-bit field when Capacity_Request_Format=0 and an 8-bit field when Capacity_Request_Format=1. It indicates the channel for which the associated capacity request is being issued. The value 0000 is the default and indicates that the request is applied to any channel, if not specified otherwise for the system. Other values are system dependent.

- **Capacity_Request_Value:** This 8-bit unsigned integer defines the volume units of payload size or the bit rate in 2 kbits/s of the capacity request as defined in table 20. A scaling factor as defined in table 19 may be applied.

- **CR_Pad_Byte:** If the space for capacity requests as given by the SAC field specification is larger than what can be utilized, this field contains the necessary number of 8-bit fields with the value "0" to fill up the unused space.

If the RCST does not have any capacity request to send, it shall send a VBDC request with an amount of 0.

- **M_and_C_Message:** This 16-bit sub-field defines M&C messages (see table 21).

**Table 21: M_and_C_Message**

| M_and_C_ Message value | Meaning |
|---|---|
| 0x0000 | No Message |
| 0x0001 | Fine synchronization achieved |
| 0x0002 | Log-off request |
| 0x0003 - 0x7FFF | Reserved |
| 0x8000 - 0xFFFF | Echo Reply |

The "No Message" is used by the RCST to indicate that it has no particular M&C message to send.

The "Fine synchronization achieved" message is used by the RCST to indicate that it has completed fine uplink synchronization on the return link and is ready to transmit traffic.

The "Log-off request" message is used by the RCST to initiate logoff.

The "Echo Reply" is a maintenance feature. The NCC can request the RCST to echo back a predetermined sequence in this SAC for trouble shooting purposes. The NCC would put this request in a TIM (see clause 8.5.5.10.8).

- **Group_ID:** This 8 bit field defines which Group ID the RCST is assigned to at logon, as identified by the Terminal Information Message (TIM). The Group_ID and Logon_ID sub-fields enable the NCC to identify the RCST that has transmitted a burst in a contention timeslot.

- **Logon_ID:** This 16 bit field defines which Logon ID the RCST is assigned to at logon, as identified by the Terminal Information Message (TIM). The Group_ID and Logon_ID sub-fields enable the NCC to identify the RCST that has transmitted a burst in a contention timeslot.

- **ACM:** The ACM sub-field allows the RCST to communicate the quality of forward link reception to the NCC for enabling adaptive coding and modulation as defined in clause 5. It is composed of the following two sub-fields:

  - **CNI:** This 8-bit sub-field defines Carrier to Noise plus Interference ratio as defined in clause 5.

  - **MODCOD_RQ:** This 8-bit sub-field defines the Modulation type and coding request as defined in clause 5.

- **Mobility_Control_Message:** The Mobility_Control_Message sub-field allows the RCST to communicate requests and status information related to mobility management to the NCC. The contents of the Mobility_Control_Message are defined in clause 6.6.1.1.1. Support for this sub-field is mandatory for RCSTs that indicate support for mobility in the CSC burst.

- **Pad_bytes:** If the SAC field length (as given by Burst.SAC_Length) is larger than the sum of its sub-field lengths, then this field contains as many 8-bit sub-fields with the value "0" as required to match the SAC field length.

When several of the SAC sub-fields are present, they appear in the order defined in table 18.

### 6.6.1.1.1          Mobility Control Message field composition (Mobile Req)

The Mobility_Control_Message allows the RCST to communicate requests and status messages related to mobility management to the NCC.The format of this message is defined in table 22.

**Table 22: Syntax of the Mobility_Control_Message field**

| Syntax (see note) | No. of bits | | Information Mnemonic |
|---|---|---|---|
| | Reserved | Information | |
| Mobility_Control_Message () { | | | |
|   Message_Type | | 3 | uimsbf |
|   if (Message_Type == 0) | | | |
|     Reserved | | 29 | uimsbf |
|   if (Message_Type == 1\|2\|3) { | | | |
|     Reserved | | 1 | bslbf |
|     Current_beam_ID | | 16 | uimsbf |
|     Candidate_beam_ID_1 | | 4 | uimsbf |
|     Candidate_beam_ID_2 | | 4 | uimsbf |
|     Candidate_beam_ID_3 | | 4 | uimsbf |
|   } | | | |
|   if (Message_Type == 4) { | | | |
|     Exclusion_Zone_Action_Request | | 3 | uimsbf |
|     Current_Beam_ID | | 16 | uimsbf |
|     Exclusion_Zone_ID | | 10 | uimsbf |
|   } | | | |
|   else if (Message_Type == 5) { | | | |
|     Back_Off | | 5 | uimsbf |
|     Azimuth_Pointing_Error | | 8 | tcimsbf |
|     Elevation_Pointing_Error | | 8 | tcimsbf |
|     Orientation_Error | | 8 | tcimsbf |
|   } | | | |
|   else if (Message_Type == 6) { | | | |
|     Position_Report_Valid | | 1 | bslbf |
|     Position_Report_Part | | 1 | bslbf |
|     Position_Report_Sequence_Number | 1 | 3 | uimsbf |
|     if (Position_Report_Part == 0) { | | | |
|       Position_Latitude | 1 | 18 | tcimsbf |
|       Altitude_Base | | 4 | uimsbf |
|     } | | | |
|     else { | | | |
|       Position_Longitude | | 19 | tcimsbf |
|       Altitude_Extension | | 4 | uimsbf |
|     } | | | |
|   } | | | |
| } | | | |

Semantics for Mobility_Control_Message:

- **Message_Type:** This 3-bit sub-field defines the type of message conveyed, as defined in table 23.

**Table 23: Mobility message type**

| Message Type | Value |
|---|---|
| No message | 000 |
| Forward and return link handover request | 001 |
| Forward link handover request | 010 |
| Return link handover request | 011 |
| Exclusion zone entry | 100 |
| Transmitter status report | 101 |
| Position report | 110 |
| Reserved | 111 |

- **Current_Beam_ID:** This 16-bit sub-field identifies the beam number of the satellite carrying the link for which the handover is requested. When Message_Type==1, this field identifies a beam used for both forward and return link.

- **Candidate_Beam_ID_1:** This 4-bit sub-field identifies the first-choice candidate handover destination beam relative to the current beam. A value of "1111" indicates that no first-choice candidate has been identified.

- **Candidate_Beam_ID_2:** This 4-bit sub-field identifies the second-choice candidate handover destination beam relative to the current beam. A value of "1111" indicates that no second-choice candidate has been identified.

- **Candidate_Beam_ID_3:** This 4-bit sub-field identifies the third-choice candidate handover destination beam relative to the current beam. A value of "1111" indicates that no third-choice candidate has been identified.

- **Exclusion_Zone_Action_Request:** This 3-bit sub-field indicates the action requested by the RCST upon entering the exclusion zone, as defined in table 24.

**Table 24: Exclusion zone action request**

| Action Requested | Value |
|---|---|
| No specific request | 000 |
| Log off | 001 |
| Change frequency | 010 |
| Adapt Transmission Parameters | 011 |
| Reserved | 100 to 111 |

NOTE 1: The "Adapt Transmission Parameters" request can entail any combination of changes to power, data rate, coding scheme and spreading factor deemed appropriate by the NCC.

- **Exclusion_Zone_ID:** This 10-bit sub-field identifies the exclusion zone, relative to the current return link beam, that the RCST is about to enter.

- **Back_Off:** This 5-bit sub-field indicates the current back-off from the maximum transmit power, in units of 0,5 dB. The value "11111" shall represent a back off of 16,5 dB or greater.

- **Azimuth_Pointing_Error:** This 8-bit sub-field indicates the current antenna pointing offset from the current return link satellite along the geostationary arc, in units of 0,1 degree. Pointing to the east of the satellite shall be indicated as a positive number. The value 0x80 shall represent an undetermined pointing error.

- **Elevation_Pointing_Error:** This 8-bit sub-field indicates the current antenna pointing offset from the current return link satellite perpendicular to the geostationary arc, in units of 0,1 degree. Pointing to the north of the satellite shall be indicated as a positive number. The value 0x80 shall represent an undetermined pointing error.

- **Orientation_Error:** This 8-bit sub-field indicates the current return link antenna pattern rotation from its nominal orientation with its long axis parallel to the geostationary arc, in units of 2 degrees. Counter-clockwise rotation as seen from the RCST shall be indicated as a positive number. The value 0x80 shall represent an undetermined orientation error.

- **Position_Report_Valid:** This 1-bit sub-field indicates whether the position report is valid. It is set to "1" if the report is valid and to "0" otherwise.

NOTE 2: This flag can also be used to indicate a refusal to provide a position report, for example for security reasons.

- **Position_Report_Part:** This 1-bit sub-field indicates which of the two parts of the position report defined in the table is being transmitted. A complete position report consists of both parts, each transmitted in one Mobility_Control_Message.

- **Position_Report_Sequence Number:** This 3-bit sub-field holds a sequence number of the position report. The sequence number should be incremented for each transmitted report and wrap to 0 when the maximum value is reached. The sequence number shall be the same for each of the two parts of a position report.

- **Position_Latitude:** This 18-bit sub-field indicates the current latitude of the terminal in the WGS84 [22] datum, in units of 0,001 degrees. Northern latitudes shall be stated as a positive number, southern latitudes as negative.

- **Altitude_Base:** This 4-bit sub-field indicates the terminal's altitude above the WGS84 [22] reference surface, in units of 1 600 m. Altitudes below the reference surface shall be indicated as 0.

- **Position_Longitude:** This 19-bit sub-field indicates the current longitude of the terminal in the WGS84 [22] datum, in units of 0,001 degrees. Eastern longitudes shall be stated as a positive number, western longitudes as negative.

- **Altitude_Extension:** This 4-bit sub-field indicates the terminal's altitude above the value provided in Altitude_Base, in units of 100 m. Altitudes below the WGS84 [22] reference surface shall be indicated as 0.

NOTE 3:  The overall reported altitude is $100 \times (16 \times \text{Altitude\_Base} + \text{Altitude\_Extension})$.

## 6.6.1.2     Prefix method mechanism

This mechanism is based on an optional $N_{p,atm}$ bytes prefix attached to ATM traffic bursts. If used, the prefix carries control and management information from the RCSTs to the NCC. This mechanism is supported by the SAC route_ID and request sub-fields (see clause 6.6.1.1) when appended to ATM Traffic bursts (see clause 6.2.1.1).

## 6.6.1.3     Mini-slot method

This mechanism is based on a periodic assignment to logged-on RCSTs of bursts smaller than traffic timeslots. It carries control and management information from the RCSTs to the NCC and is used also for maintaining RCST synchronization. This mechanism is supported by the SAC request sub-field (see clause 6.6.1.1) used in SYNC bursts (see clause 6.2.2.1).

## 6.6.1.4     Contention based mini-slot method

As per the method described in the previous clause, but the mini-slot can be accessed by a group of RCSTs on a contention basis. This mechanism is supported by the SAC request, Group_ID and Logon_ID sub-fields (see clause 6.6.1.1) used in SYNC bursts (see clause 6.2.2.1).

## 6.6.1.5     MPEG Adaptation Field Method (MPAF) (option)

This method is based on using the private data bytes of the MPEG layer adaptation field to carry a SAC message. The adaptation field, if used, carries control and management information from the RCST to the NCC. The availability of the method is not signalled in the TCT but directly in the MPEG header of the MPEG TRF burst. The functionality shall be configurable in the RCST to ensure compatibility with gateways not implementing the option.

The format of TS packet carrying a SAC in the adaptation field shall be according to ISO/IEC 13818-1 [7], clause 2.4.3. Specifically, the fields shall be coded as follows:

- sync_byte: as defined in [7];

- transport_error_indicator: as defined in [7];

- payload_unit_start_indicator: set to "1" if the TS packet includes user data in the payload section following the adaptation field, or set to "0" if the TS packet carries no payload section;

- transport_priority: set to "0";

- PID: set to the assigned PID used for return link traffic;

- transport_scrambling_control: as defined in [7];

- adaptation_field_control: set to "10" whenever a SAC message only is being sent (no payload) and set to "11" when the adaptation field is followed by a payload section;

- adaptation_field(): as defined below.

When the TS packet does not carry a SAC, the adaptation field is coded as per ISO/IEC 13818-1 [7], clause 2.4.3.4 with the following field values:

- adaptation_field_length: set to "2 + transport_private_data_length";

- discontinuity_indicator: set to "0";

- random_access_indicator: set to "0";

- elementary_stream_priority_indicator: set to "0";

- PCR_flag: set to "0";

- OPCR_flag: set to "0";

- splicing_point_flag: set to "0";

- transport_private_data_flag: set to "1";

- adaptation_field_extension_flag: set to "0";

- transport_private_data_length: set to "number of private_data_bytes that follow";

- private_data_byte: coded as shown in the table blow.

The private data bytes of the adaptation field shall be coded as defined in table 25.

**Table 25: Syntax of the MPEG Adaptation Field Private Data field**

| Syntax | No. of bits | | Information Mnemonic |
|---|---|---|---|
| | Reserved | Information | |
| Adaptation_Field_Private_data() { | | | |
|   SAC_composition { | | | |
|     Route_ID_flag | | 1 | bslbf |
|     ACM_flag | | 1 | bslbf |
|     MOB_flag | | 1 | bslbf |
|     SAC_length | | 5 | uimsbf |
|     request_flag | | 1 | bslbf |
|     M_and_C_flag | | 1 | bslbf |
|     Group_ID_flag | | 1 | bslbf |
|     Logon_ID_flag | | 1 | bslbf |
|     capacity_requests_number | | 3 | bslbf |
|     Reserved | | 1 | bslbf |
|   } | | | |
|   SAC_field as defined in clause 6.6.1.1 | | n | |
| } | | | |

Semantics for the Adaptation Field Private Data field:

- **SAC_composition:** the semantics for the sub-fields (Route_ID_flag, ACM_flag, MOB_flag, SAC_length, request_flag, M_and_C_flag, Group_ID_flag, Logon_ID_flag, capacity_requests_number) in the adaptation field private data is equal to the corresponding fields in the TCT (see clause 8.5.5.4).

- **SAC_field:** as defined in clause 6.6.1.1.

## 6.6.2    Data Unit Labelling Method (DULM)

The Data Unit Labelling Method (DULM) is a "message-based" method that allows RCSTs to transmit control and/or management information to the NCC in the payload of the Data Units already assigned to them in TRF bursts. These Data Units are either ATM cells or MPEG TS packets, depending on the encapsulation mode of the RCST, as described in clause 6.2.1.

The sequence of these "special" TRF bursts from an RCST to the NCC constitutes a dedicated virtual channel named CTRL/MNGM. This virtual channel can thus be utilized to support message-based delivery of control (e.g. bandwidth requests) and/or management information, possibly in combination with other methods described in this clause.

The use of the CTRL/MNGM virtual channel may be at the initiative of the NCC or of the RCST. For instance, an active RCST may insert a bandwidth request in an already assigned timeslot. The NCC may request status information from an RCST, and allocate timeslots to a terminal for the CTRL/MNGM channel.

## 6.6.2.1      DULM with ATM-formatting

For RCSTs, using ATM TRF bursts as defined in clause 6.2.1.1, the CTRL/MNGM virtual channel shall be identified by a unique value in the Data Unit header, with the MSB of the PT field set to 1 so as to allow discrimination from normal traffic information (see clause 8.1.1).

A DULM message shall be composed of an integer number (between 1 and 64) of Information Elements (IE).

Each IE shall have the format described in figure 26, and be made of 2 bytes of header (IE type and IE length) plus a body of n bytes, with n between 1 and 512. A given DULM message may be composed of IEs of different lengths.
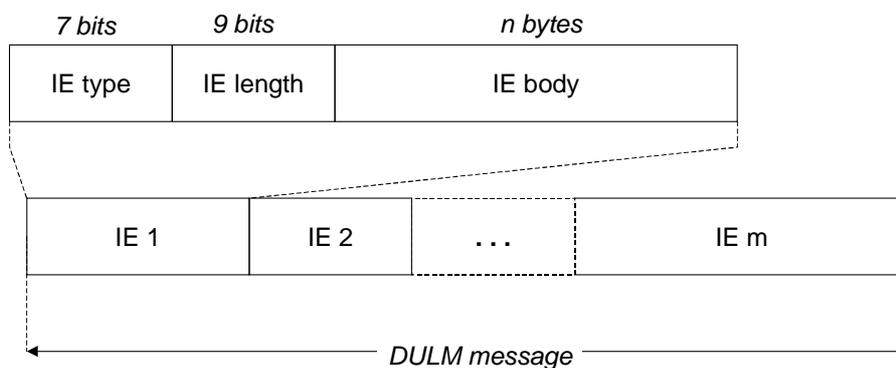


**Figure 26: DULM message with ATM TRF bursts**

The DULM message shall then be transmitted using standard AAL5 mechanisms, as specified in ITU-T Recommendation I.363-5 [15].

The Information Elements shall be as defined in table 26.

**Table 26: IEs for the ATM TRF case**

| IE type | | IE length (see note 1) | IE body |
|---|---|---|---|
| 0x00 | Capacity Request | 2 bytes | As per clause 6.6.1.1 |
| 0x01 | M&C | 2 bytes | As per clause 6.6.1.1 |
| 0x02 | Group_and_Logon_ID | 3 bytes | As per clause 6.6.1.1 (see note 2) |
| 0x03 | Message Header | 4 bytes | See description below |
| 0x04 | Cause | 2 bytes | See description below |
| 0x05 | Channel_ ID | 1 byte | See description below |
| 0x06 | Source Address | 6 bytes | See description below |
| 0x07 | Destination Address | 6 bytes | See description below |
| 0x08 | Forward Stream Identifier | 3 bytes | See description below |
| 0x09 | Return Stream Identifier | 3 bytes | See description below |
| 0x0A | Type | 1 byte | See description below |
| 0x0B | Forward Profile | 3 bytes | See description below |
| 0x0C | Return Profile | 3 bytes | See description below |
| 0x0D | Security Sign-on Response | 8 bytes | As per clause 9.4.9.2 |
| 0x0E | Route_ID | 2 bytes | See description below |
| 0x0F - 0x1E | Reserved for connection control | As defined in [23] | |
| 0x1F | Wait | As per clause 9.4.9.9 | |
| 0x20 - 0x30 | Reserved | | |
| 0x31 | Main Key Exchange Response | As per clause 9.4.9.4 | |
| 0x32 | Reserved | | |
| 0x33 | Quick Key Exchange Response | As per clause 9.4.9.6 | |
| 0x34 | Reserved | | |
| 0x35 | Explicit Key Exchange Response | As per clause 9.4.9.8 | |
| 0x36 | ACM | 2 bytes | as per clause 6.6.1.1 |
| 0x37 | Mobility_Control_Message | 4 bytes | as per clause 6.6.1.1 |
| 0x38 | Continuous Carrier Control | As per clause 10.4.4 | |
| 0x39 - 0x5F | Reserved | | |
| 0x60 - 0x7F | User defined | See note 3 | |
| NOTE 1: "IE length": length (in bytes) of the IE body. | | | |
| NOTE 2: Group_and_Logon_ID: concatenation of the 1-byte Group_ID and the 2-byte Logon_ID fields, in this order. | | | |
| NOTE 3: User defined values can be used for system-specific IEs. | | | |

IE type description:

- Message Header: it identifies the type of message, sets the total length in byte of the message and identifies the connection affected by the connection control signalling.

- Cause: it conveys the reason for the reject of a previous request.

- Channel_ID: defined and utilized according to the present document.

- Source Address: it is the address of the calling end point.

- Destination Address: it is the address of the called end point(s).

- Forward stream identifier: it identifies a single forward information flow pertaining to the connection or to an aggregation of connections; it is either one {VPI, VCI} pair or a single PID, depending on the ATM or MPEG-2 nature of the information flow.

- Return stream identifier: it identifies a single return information flow pertaining to the connection or to an aggregation of connections; it is either one {VPI, VCI} pair or a single PID, depending on the ATM or MPEG-2 nature of the information flow.

- Type: it describes the connection configuration in terms of direction and casting.

- Forward Profile: it describes the priority and the overall amount of resources of the forward streams of the connection.

- Return Profile: it describes the priority and the overall amount of resources of the return streams of the connection.

- Route_ID: see clause 6.6.1.1.

## 6.6.2.2     DULM with MPEG-formatting

For the RCSTs using the optional MPEG TRF bursts, a CTRL/MGNM PID shall be used in the header of CTRL/MNGM bursts. This PID is obtained by the RCST during the logon procedure (see clause 8.5.5.10.4).

DULM messages shall use the transport mechanism described below, and illustrated in figure 27.
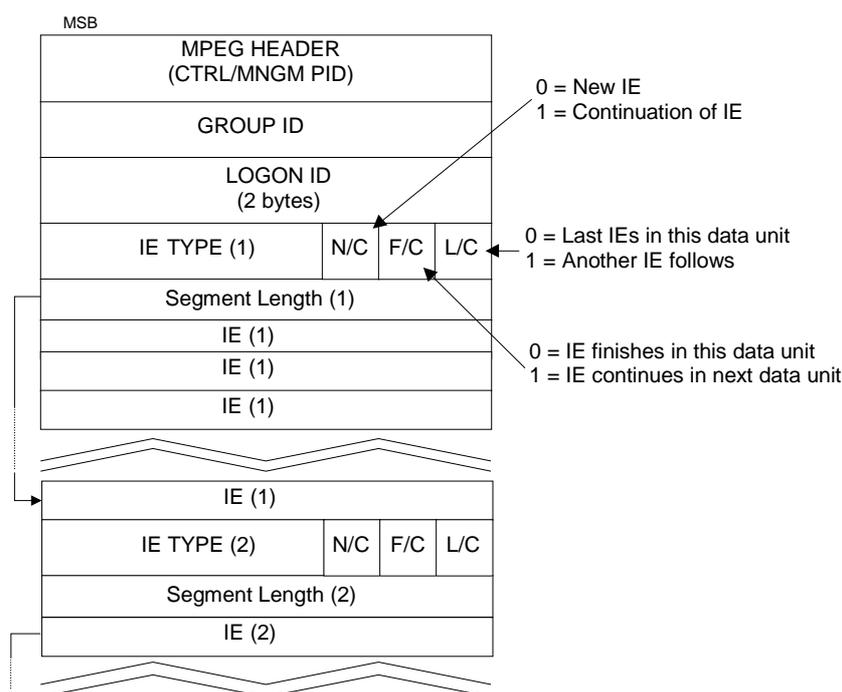


**Figure 27: DULM over optional MPEG packets**

The first 3 bytes of each Data Unit payload shall contain the RCST Group_ID and Logon_ID. Then one or more Information Elements (IE) follow with the structure described hereafter.

The first byte of an IE contains 4 fields:

- the [IETYPE] field is contained in the first 5 bits. It contains the IE type, as defined in table 27;

- the [N/C] is contained in the 6th bit. If [N/C] = 0 the IE is starting in this Data Unit, if [N/C] = 1 the IE is continued from the previous CTRL/MNGM Data Unit from the same RCST;

- the [F/C] is contained in the 7th bit. If [F/C] = 0 the IE will finish in this CTRL/MNGM Data Unit, if [F/C] = 1 the IE will continue in the next CTRL/MNGM Data Unit from the same RCST;

- the [L/C] is contained in the 8th bit. If [L/C] = 0 the IE is the last of the CTRL/MNGM Data Unit, if [L/C] = 1 another IE follows in the same CTRL/MNGM Data Unit.

The second byte contains the [SLENG] field. It indicates the segment length in number of bytes, defined as the part of the IE contained in the current Data Unit.

The following bytes contain the variable-length fragment.

   NOTE:     If the IE spans over several CTRL/MNGM Data Units, the IE header (IETYPE, [N/C], [F/C], [L/C], segment_length) is duplicated on all Data Units.

Padding bytes set to all "0"s shall be appended to the last IE of a CTRL/MNGM data unit.

**Table 27: IEs for the MPEG TRF case**

| IE type (MPEG) | | IE length | IE body |
|---|---|---|---|
| 0×00 | Capacity Request | 2 bytes | As per clause 6.6.1.1 |
| 0x01 | M&C | 2 bytes | As per clause 6.6.1.1 |
| 0x02 | Reserved | | |
| 0x03 | Message Header | 4 bytes | See description below |
| 0x04 | Cause | 2 bytes | See description below |
| 0x05 | Channel_ID | 1 byte | See description below |
| 0x06 | Source Address | 6 bytes | See description below |
| 0x07 | Destination Address | 6 bytes | See description below |
| 0x08 | Forward Stream Identifier | 3 bytes | See description below |
| 0x09 | Return Stream Identifier | 3 bytes | See description below |
| 0x0A | Type | 1 byte | See description below |
| 0x0B | Forward Profile | 3 bytes | See description below |
| 0×0C | Return Profile | 3 bytes | As described in [i.3] |
| 0x0D | Security Sign-on Response | 8 bytes | As per clause 9.4.9.2 |
| 0x0E | Route_ID | 2 bytes | See description below |
| 0x0F - 0x10 | Reserved | | |
| 0x11 | Main Key Exchange Response | As per clause 9.4.9.4 | |
| 0x12 | Reserved | | |
| 0x13 | Quick Key Exchange Response | As per clause 9.4.9.6 | |
| 0x14 | Reserved | | |
| 0x15 | Explicit Key Exchange Response | As per clause 9.4.9.8 | |
| 0x16 | ACM | 2 bytes | as per clause 6.6.1.1 |
| 0x17 | Mobility_Control_Message | 4 bytes | as per clause 6.6.1.1 |
| 0x18 | Continuous Carrier Control | As per clause 10.4.4 | |
| 0x19 - 0x1B | Reserved | | |
| 0x1C - 0x1D | User defined | System dependent | |
| 0x1E | Extended IE type for connection control | See description below | |
| 0x1F | Wait | As per clause 9.4.9.9 | |

IE type description:

- Message Header: it identifies the type of message, sets the total length in byte of the message and identifies the connection affected by the connection control signalling.

- Cause: it conveys the reason for the reject of a previous request.

- Channel_ID: defined and utilized according to the present document.

- Source Address: it is the address of the calling end point.

- Destination Address: it is the address of the called end point(s).

- Forward stream identifier: it identifies a single forward information flow pertaining to the connection or to an aggregation of connections; it is either one {VPI, VCI} pair or a single PID, depending on the ATM or MPEG-2 nature of the information flow.

- Return stream identifier: it identifies a single return information flow pertaining to the connection or to an aggregation of connections; it is either one {VPI, VCI} pair or a single PID, depending on the ATM or MPEG-2 nature of the information flow.

- Type: it describes the connection configuration in terms of direction and casting.

- Forward Profile: it describes the priority and the overall amount of resources of the forward streams of the connection.

- Return Profile: it describes the priority and the overall amount of resources of the return streams of the connection.

- Route_ID: see clause 6.6.1.1.

- Extended IE type for connection control: as defined in [23].

## 6.7 Multiple access

The multiple-access capability is either fixed or dynamic slot MF-TDMA. RCSTs shall indicate their capability by using the MF-TDMA field present on the CSC burst (see clause 6.2.3).

In addition, the optional provisions for continuous carrier access defined in clause 10 may be applied.

### 6.7.1 MF-TDMA

The satellite access scheme is Multi-Frequency Time Division Multiple Access (MF-TDMA). MF-TDMA allows a group of RCSTs to communicate with a gateway using a set of carrier frequencies, each of which is divided into time-slots. The NCC will allocate to each active RCST a series of bursts, each defined by a frequency, a bandwidth, a start time and a duration.

#### 6.7.1.1 Fixed MF-TDMA

In Fixed-Slot MF-TDMA, the bandwidth and duration of successive traffic slots used by an RCST is fixed, as illustrated in figure 28 where the arrow indicates a typical sequence of slots assigned by the NCC to one RCST. In this case, TCT parameters (see clause 8.5.5.4) defining the burst parameters (symbol_rate, inner_code_type, inner_code_ordering, outer_coding, inner_code_puncturing, modulation and baseband shaping) of a superframe are fixed. A fixed MF-TDMA RCST can send a mix of SYNC and single size TRF bursts provided that the burst parameters fulfil the previous requirement. If the NCC requests a change in these parameters, then they will apply to a new superframe with a delay as described in clause 6.7.2.3.
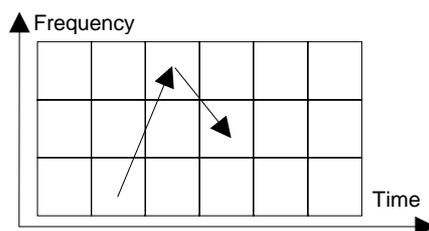


**Figure 28: Fixed-slot MF-TDMA**

#### 6.7.1.2 Dynamic MF-TDMA (Optional)

Dynamic-Slot MF-TDMA uses additional RCST flexibility to vary the bandwidth and duration of successive slots allocated to an RCST. In addition to changing carrier frequency and burst duration, the RCST may also change transmission rate and coding rate between successive bursts. The advantage of the more flexible RCST is more efficient adaptation to the widely varying transmission requirements typical of multimedia. The basic principle of the flexible RCST is illustrated in figure 29, where the arrows show an RCST using successive slots with different bandwidths and durations.
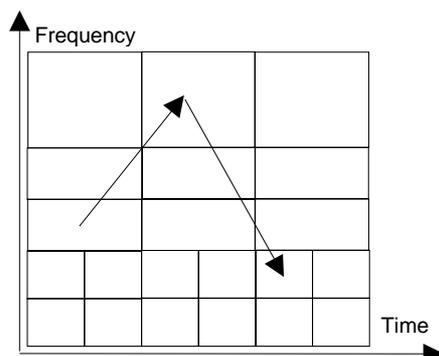


**Figure 29: Optional dynamic-Slot MF-TDMA, using a flexible RCST**

## 6.7.1.3 Frequency range

RCSTs have a specific frequency range for the frequency hopping from time-slot to time-slot. This frequency hopping range is communicated from an RCST to the NCC in a CSC burst during logon procedure (see clause 6.2.3). The frequency hopping capability of individual RCSTs shall be at the manufacturer's discretion and shall be at least 20 MHz (i.e. ±10 MHz around centre frequency).

The frequency agility is specified in terms of short term frequency hopping and long term frequency tuning.

Frequency hopping covers changes of frequency between adjacent slots in time. The settling time between bursts shall not exceed the guard interval defined in clause 6.5.4. The settling time is the period required to achieve the frequency tolerance given in clause 6.1.2. As an option, a manufacturer can declare that the terminal needs a full TRF slot between transmissions on different carrier frequencies within that range. The selected option shall be signalled in the CSC burst.

Frequency tuning covers the change in centre frequency of hop ranges. The settling time shall not exceed 1 s.

## 6.7.2 Segmentation of the return link capacity

In a Satellite Interactive Network, the timeslots of the return link are organized and numbered so that the network is able to allocate them to individual RCSTs.

## 6.7.2.1 Superframes

A superframe is a portion of time and frequency of the return link.

Within a Satellite Interactive Network, a Superframe_ID identifies the return link resources accessed by a given set of RCSTs. Figure 29 shows a typical example whereby superframe_IDs are indeed separate sets of carrier frequencies.
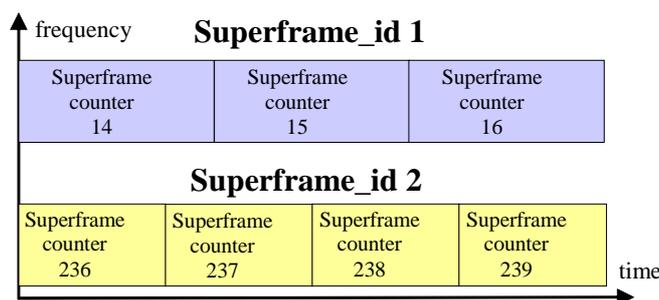


**Figure 30: Typical example of superframes of a Satellite Interactive Network**

In a Satellite Interactive Network, the global return link capacity may be segmented amongst sets of RCSTs, and the network will then separately manage several Superframe_IDs. In the following, we only consider one Superframe_ID.

As shown by figure 30, the consecutive superframes of a given Superframe_ID are contiguous in time. Each occurrence of a superframe in time is labelled with a number called "superframe_counter".

For each superframe (of a given Superframe_ID), allocation of timeslots is communicated to the RCSTs via the TBTP table (see clause 8.5.5.7). An RCST is allowed to transmit bursts only in timeslots which were allocated to it ("dedicated access"), or on random-access timeslots ("contention access").

NOTE 1: Some timeslots (like the ACQ and the SYNC bursts, see clauses 6.2.2.1 and 6.2.2.2) can be assigned to RCSTs on the basis of a period which is much longer than the superframe via individual TIM messages (see clauses 8.5.5.10.5 and 8.5.5.10.6). The period for these timeslots will be system dependent, but typically in the order of one second.

The superframe duration is therefore the elementary period of time for the assignment of resources to terminals.

NOTE 2: Some timeslots (like the SYNC bursts, see clause 6.2.2.1) can be assigned to RCSTs on the basis of a period which is much longer than the superframe duration, in the order of one second (system-dependent).

### 6.7.2.2        Frames

A superframe is composed of frames, themselves composed of timeslots. The frame is at an intermediate level between the superframe and the timeslots. It is introduced for reasons of signalling efficiency (on the forward link signalling). The frame duration is not used as the basis of any timeslot allocation process.
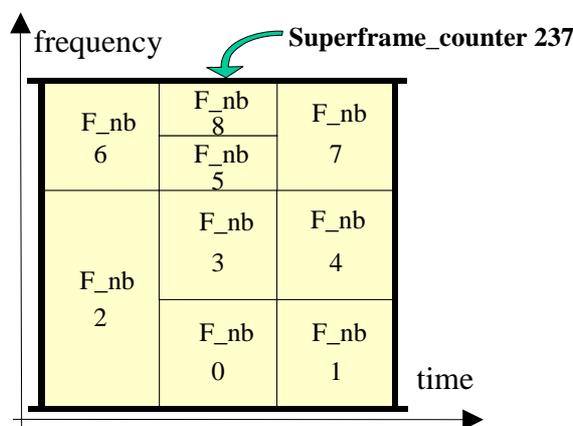


**Figure 31: Example of superframe composition**

In a superframe, frames are numbered from 0 (lowest frequency, first in time) to N (highest frequency, last in time), ordered in time then in frequency as shown in figure 31. N shall be less than or equal to 31.

Frames of a superframe may not all have the same duration, bandwidth and timeslot composition.

Frames and superframes may all have the same duration, in which case frames can be seen as frequency sub-bands of the superframe. Anyway this property is not mandatory, and figure 31 shows an example of one superframe lasting 3 times more than each of its frames.

### 6.7.2.3        Timeslots

A frame is composed of timeslots (see the Frame Composition Table, clause 8.5.5.3). A "frame_id" identifies a particular arrangement of timeslots. For example, frame_id = 1 could identify a sequence of 10 "user traffic" timeslots on the same carrier, and frame_id = 2 a sequence of 4 "control" timeslots followed by 8 "user traffic" timeslots, all on the same carrier.
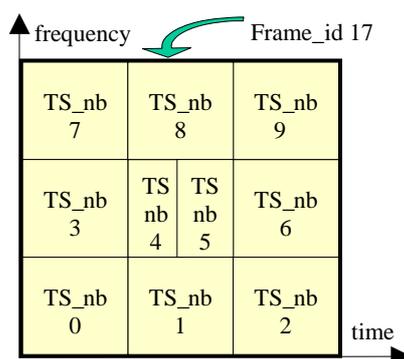


**Figure 32: Example of frame composition**

A frame may span over several carrier frequencies. In a frame, timeslots are numbered from 0 (lowest frequency, first in time) to M (highest frequency, last in time), ordered in time then in frequency as shown in figure 32. M shall be less than or equal to 2 047.

For the purpose of allocation, each timeslot is uniquely identified by its Superframe_ID, Superframe_counter, Frame_number and Timeslot_number.

The RCST shall process the TBTP message from the NCC for its allocation area, to extract the assignment count and timeslot allocations for its next uplink transmissions. The latency time from the arrival of the TBTP message at the RCST until the RCST is ready to transmit the bursts assigned by that TBTP shall not exceed 90 ms for MF-TDMA transmissions. The latency to be ready to transmit shall not exceed 2secondsfor the optional continuous carrier transmissions.

# 6.8        Capacity request categories

The timeslot allocation process shall support five capacity categories:

- Constant Rate Assignment (CRA).

- Rate Based Dynamic Capacity (RBDC).

- Volume Based Dynamic Capacity (VBDC).

- Absolute Volume Based Dynamic Capacity (AVBDC).

- Free Capacity Assignment (FCA).

Capacity request mechanisms for the optional continuous carrier access method are defined in clause 10.

## 6.8.1      Continuous Rate Assignment (CRA)

CRA is rate capacity which shall be provided in full for each and every superframe while required. Such capacity shall be negotiated directly between the RCST and the NCC.

## 6.8.2      Rate Based Dynamic Capacity (RBDC)

RBDC is rate capacity which is requested dynamically by the RCST. RBDC capacity shall be provided in response to explicit requests from the RCST to the NCC, such requests being absolute (i.e. corresponding to the full rate currently being requested). Each request shall override all previous RBDC requests from the same RCST, and shall be subject to a maximum rate limit negotiated directly between the RCST and the NCC.

To prevent a terminal anomaly resulting in a hanging capacity assignment, the last RBDC request received by the NCC from a given terminal shall automatically expire after a time-out period whose default value is 2 superframes, such expiry resulting in the RBDC being set to zero rate. The time-out can be configured between 1 and 15 superframes (if set to 0 the time out mechanism is disabled) by the optional mechanism of clause 8.4.2.

CRA and RBDC can be used in combination, with CRA providing a fixed minimum capacity per superframe and RBDC giving a dynamic variation component on top of the minimum.

## 6.8.3      Volume Based Dynamic Capacity (VBDC)

VBDC is volume capacity which is requested dynamically by the RCST. VBDC capacity shall be provided in response to explicit requests from the RCST to the NCC, such requests being cumulative (i.e. each request shall add to all previous requests from the same RCST). The cumulative total per RCST shall be reduced by the amount of this capacity category assigned in each superframe.

## 6.8.4      Absolute Volume Based Dynamic Capacity (AVBDC)

AVBDC is volume capacity which is requested dynamically by the RCST. This VBDC capacity shall be provided in response to explicit requests from the RCST to the NCC, such requests being absolute (i.e. this request replaces the previous ones from the same RCST). The AVBDC is used instead of VBDC when the RCST senses that the VBDC request might be lost (for example in the case of contention minislots).

## 6.8.5 Free Capacity Assignment (FCA)

FCA is volume capacity which shall be assigned to RCSTs from capacity which would be otherwise unused. Such capacity assignment shall be automatic and shall not involve any signalling from the RCST to the NCC. It shall be possible for the NCC to inhibit FCA for any RCST or RCSTs.

FCA should not be mapped to any traffic category, since availability is highly variable. Capacity assigned in this category is intended as bonus capacity which can be used to reduce delays on any traffic which can tolerate delay jitter.

# 7 Synchronization procedures

This clause defines the procedures to allow an RCST to logon to the satellite interactive network and also how the terminal can logoff from (or be logged off by) the network.

Additional provisions for the optional continuous carrier access method are provided in clause 10.

The period of time from the terminal logon to the terminal logoff is called a **session**.

## 7.1 Overall events sequencing

In order to be able to proceed to logon, the RCST shall be in the **Receive Synchronization state**, which is reached following the **Initial synchronization procedure** described in clause 7.2.

The entry of an RCST into the system is then achieved through the following four phases:

- **Logon procedure:** the RCST requests initial access to the network and gets initial logon information from the network (or alternatively the logon request may be rejected by the network). See clause 7.3.

- **Acquisition coarse synchronization procedure (optional):** the RCST improves its physical synchronization (frequency, time, and power adjustments). See clause 7.4.

- **Fine synchronization procedure (optional):** the RCST completes its physical synchronization. See clause 7.5.

- **Synchronization maintenance procedure:** the RCST maintains its physical synchronization during the entire session. See clause 7.6.

Corresponding to the *procedures*, the RCST can be in one of the following *states*:

- **Hold:** the RCST is in hold mode. The RCST Status field in the TIM contains a flag called Transmit_Disable (see clause 8.5.5.8). A terminal that receives a TIM with this flag set to 1 shall cease transmission and release all assigned logon session parameters (i.e. Logon_ID, Group_ID, timeslot allocations) and enter the Hold state. This can happen when the terminal is in the Receive sync state, Ready for coarse sync state, Ready for fine sync state or the Fine Sync state. A terminal that is in the Hold state shall remain there after a power off or reset. A terminal goes from the Hold state to the Receive sync state only when it receives a TIM with the Transmit_Disable flag set to 0.

- **Inactive Off/Stand-by:** the RCST is not powered or on a stand-by mode or has lost synchronization.

- **Receive sync:** the RCST has acquired the forward link.

- **Ready for coarse sync:** the RCST has been detected by the NCC, and may initiate a coarse synchronization procedure.

- **Ready for fine sync:** the RCST has been detected by the NCC, and may initiate a fine synchronization procedure.

- **Fine sync:** the RCST is synchronized and can send traffic.

The logoff procedure described in clause 7.7 allows the RCST to leave the network.

Figures 25 to 30 give an overview of the sequence, where ⬭ represent states and ▭ represent procedures. Figure 33 shows the RCST synchronization state diagram.
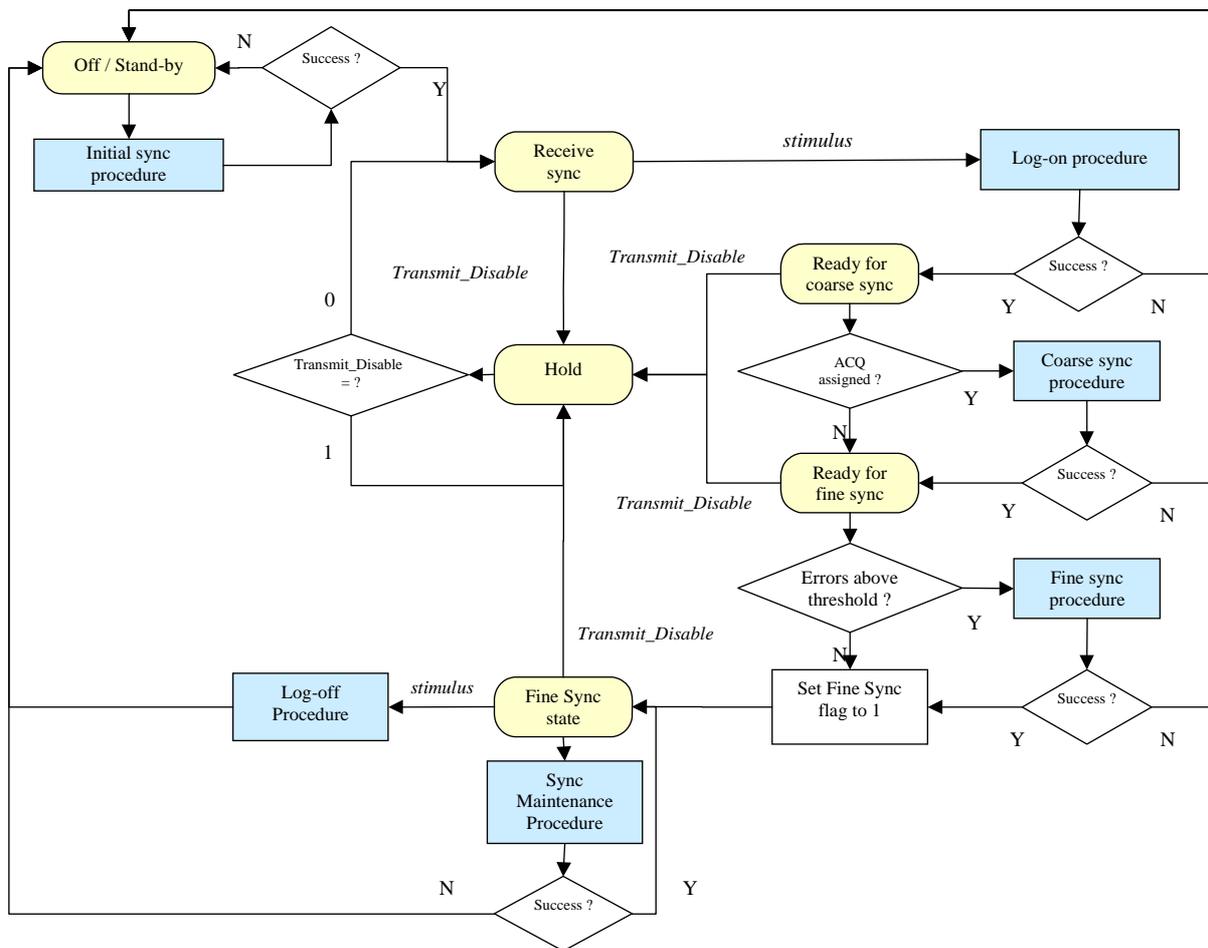


**Figure 33: RCST synchronization state diagram**

All the states, events, conditions and procedures are further described in this clause.

The RCST is not allowed to transmit TRF bursts until it has reached the "Fine Sync" state.

**Signalling Messages**

The exchange of signalling messages between RCST and NCC, including optional messages, is illustrated in figure 34.

This illustrates the normal flow of events in the case that the optional coarse and fine synchronization procedures are used.
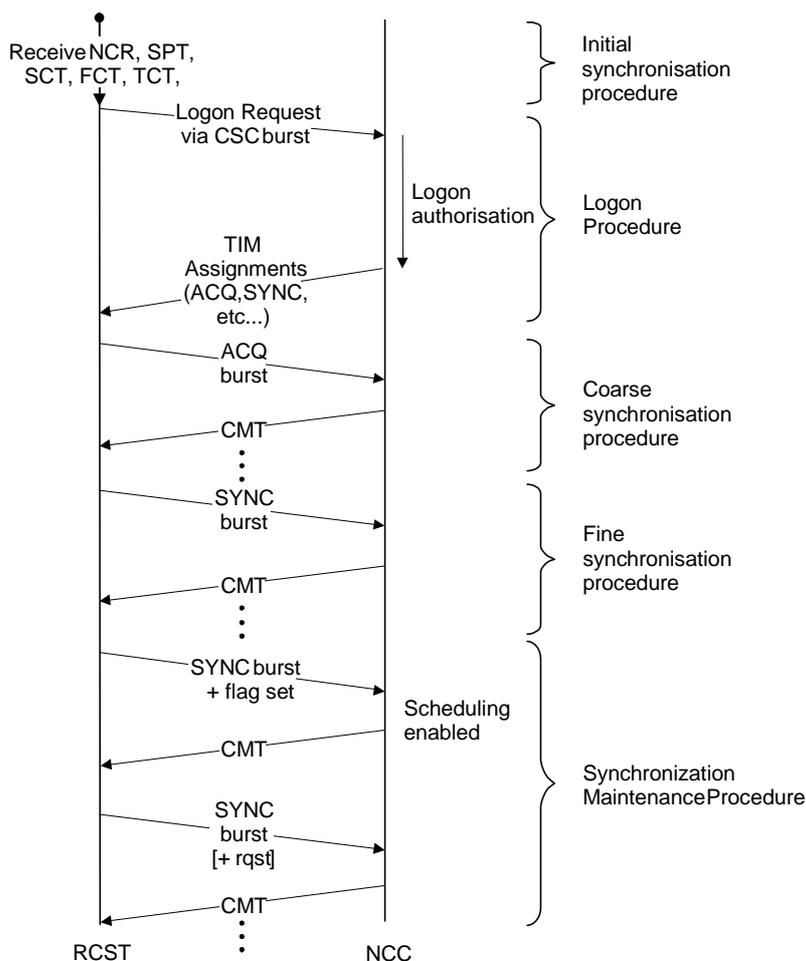
**Figure 34: Example of RCST network entry signalling flow**

# 7.2    Initial synchronization procedure

Following the power-up, the RCST shall proceed as detailed below:

- The RCST shall first follow the procedures described in clause 8.5.5.11 to find all necessary control information related to the operation of the RCS network. This includes NCR synchronization, through which the RCST initiates its internal clock, by tracking the NCR which is transmitted by the NCC on the forward link.

- The RCST shall then calculate the satellite ranges for both forward and return links using the satellite ephemeris data contained within the Satellite Position Table (SPT) plus a knowledge of its own location (latitude, longitude and height above sea level). It shall use these ranges to calculate the corresponding satellite to RCST and RCST to satellite propagation delays. The nominal satellite position, which can be found in the NIT, shall be used in the case that the NCC does not transmit a SPT.

- The RCST shall continue to receive the NCR throughout the session. In the event that NCR synchronization is lost, the RCST ceases transmission (see clause 7.7.3) and shall re-start the initial synchronization procedure. Similarly, any failure of the RCST during one of the later-described procedures takes the RCST back to the initial synchronization procedure.

- The RCST shall receive the burst time plan transmitted by the NCC at regular intervals. The BTP is contained in the Forward link Signalling, and is made of the Superframe, Frame and Timeslot Composition Tables. All these tables are described in clause 8. The RCST shall also acquire the broadcast TIM.

After following these steps, the RCST shall enter the Receive sync state.

# 7.3      Logon procedure

After the RCST has received all SI tables related to the structure of the satellite interactive network, it is ready to initiate a network logon, in order to be admitted into the system and be ready to handle traffic. The RCST can decide to move from the "Receive sync" state e.g. because it is booting up or because it wants to transmit data and is no longer logged-on after a long period of inactivity. Alternatively, the network may trigger the logon procedure by sending a "Wake up" signal to the RCST in a unicast TIM as described in clause 8.5.5.8.

The logon procedure is illustrated in figure 35 and detailed below.

The RCST sends a logon request in a CSC timeslot using Slotted-Aloha random access. This request contains the RCST MAC address and a field indicating the capabilities of the terminal (see clause 6.2.3). If it is received correctly, the NCC will proceed with the next step. In the absence of a reply from the NCC in due time, the RCST shall assume that there was a collision between multiple simultaneous requests and shall retry after a maximum, randomly-selected interval. If applicable, upper limits for repetition rate and duration of the logon request are specified in [5]. Parameters for this retransmission scheme are retrieved by the terminal from the Contention Control descriptor (see clause 8.5.5.10.14), acquired during the Initial synchronization procedure.

The NCC verifies that transmission resources are available (ACQ and SYNC bursts) and checks if the administrative aspects are satisfied (e.g. account is valid, account is paid, etc.). If all conditions are met, the NCC proceeds with the next step. As shown in figure 35 the RCST shall wait for $n^2$ times the value of max_time_before_retry with n being the number of passes through this loop.

The NCC sends a TIM message to the RCST as an acknowledgement. This "logon" TIM shall contain the information detailed in clauses 8.5.5.8 and 8.5.5.10.1.

Subsequently, an ATM VPI/VCI shall be used by RCSTs sending ATM TRF bursts. For RCSTs sending optional MPEG packets in TRF bursts, a PID is assigned. VPI/VCI or PID values are assigned in the TIM Logon Initialize descriptor (see clause 8.5.5.10.4) or mesh logon initialize descriptor (see clause 8.5.5.10.22).

NOTE:      Under exceptional conditions such as a severe network failure, access to the CSC bursts may be dedicated rather than random.
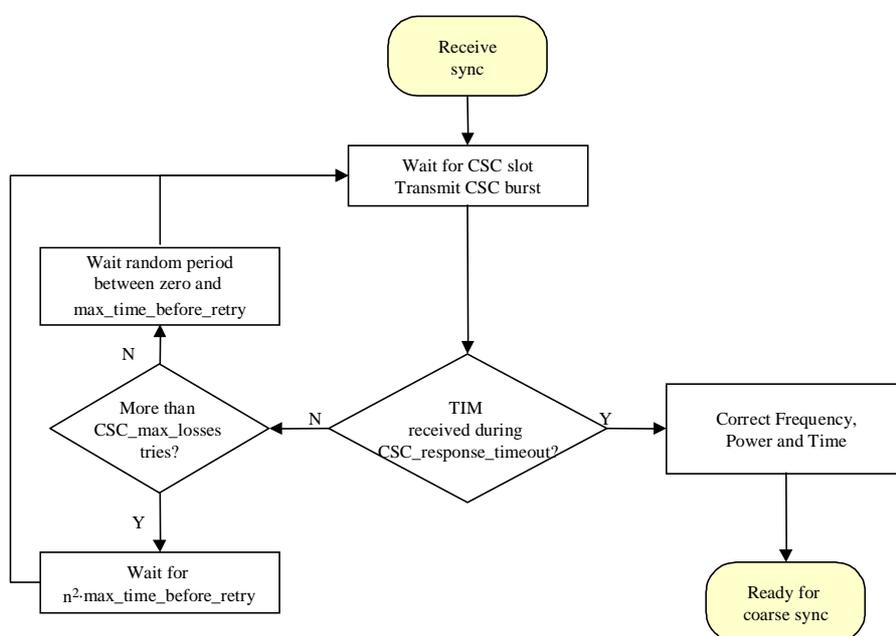


**Figure 35: Logon procedure**

## 7.3a     Logon in the presence of a large timing uncertainty (Mobile Req)

The provisions in this clause may be used to allow logon in situations where the RCST-to-satellite delay is not known with an accuracy that is sufficient to ensure that the CSC burst can be transmitted such that it is received within the boundaries of a CSC time slot.

In this situation, the NCC can transmit the optional Correction_message_extension descriptor in the logon TIM, to identify the time slot in which the CSC burst was received. When computing the overall timing correction, the RCST shall combine the "coarse" correction conveyed in this manner with the "fine" correction conveyed in the Correction_message_descriptor.

Support for this optional feature shall be indicated in the forward link signalling by setting the Large_Timing_Uncertainty_Flag in the corresponding Superframe Composition Section to "0". An RCST requiring the use of this method shall not attempt to logon unless the support is thus signalled.

When using this method, the RCST shall locate the longest available sequence of consecutive CSC slots in the superframe and shall attempt to transmit the CSC burst such that it is received either:

- In the central slot if the number of consecutive CSC slots is odd.

- In either of the two slots closest to the central time instant of the sequence, if the number of consecutive CSC slots is even.

## 7.4     Coarse synchronization procedure (optional)

In a network where all RCSTs are locked to the NCR on the forward link, the NCC can correct all frequency and timing errors other than differential Doppler between the RCST and the NCC. Initial burst time errors can be low when the satellite and terminal position are known. Provided the NCC/gateway receivers can handle these residual errors, which are small for a satellite maintained in a tight "box", there is no requirement for the RCST to perform the ranging process of the coarse synchronization procedure. In this case the TIM will not contain an ACQ assign descriptor (see clause 8.5.5.10.5), and the RCST shall enter directly the "Ready for fine synchronization" state. Otherwise, if the descriptor is contained then the RCST shall perform the coarse synchronization procedure given here.

After the RCST has logged on and been given the authorization to proceed, it shall commence the acquisition phase to achieve timing and frequency synchronization and power adjustment. The procedure is illustrated in figure 36 and detailed below.

The RCST is assigned ACQ bursts via the TIM. The RCST sends an ACQ burst at its reserved time slot. The NCC measures the timing, frequency and power error of the ACQ burst, relative to the system reference, and sends this information back in the Correction Message Table (CMT, see clause 8.5.5.9). The RCST in turn adjusts its transmission parameters, and retries. This process continues until the accuracy is within the "coarse sync thresholds" indicated to the RCSTs in the ACQ Assign descriptor (see clause 8.5.5.10.5). The same descriptor provides means to limit the number of loops in this procedure ("Max tries exceeded"). Parameters for this procedure are contained in the correction_control_descriptor (see clause 8.5.5.10.15).
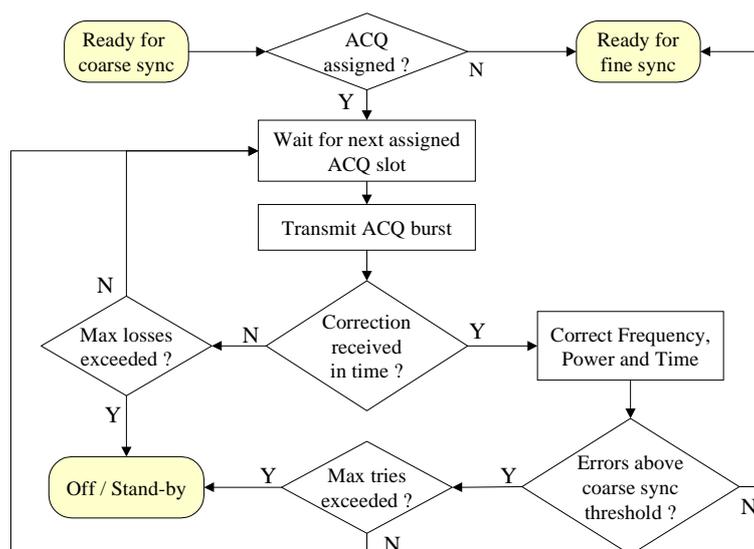
**Figure 36: Coarse synchronization procedure (optional)**

## 7.5      Fine synchronization procedure (optional)

This procedure is quite similar in principle to the coarse synchronization procedure described in clause 7.4, but it uses dedicated SYNC slots instead of ACQ bursts. This procedure is only performed if the errors indicated in the latest correction message (which was either included in the "logon" TIM received right after CSC, or in a CMT during the optional ACQ procedure) are larger than the "fine sync thresholds" indicated in the SYNC Assign descriptor of the "logon" TIM (see clause 8.5.5.10.6).

The fine synchronization procedure is shown in figure 37. Parameters for this procedure are contained in the correction_control_descriptor (see clause 8.5.5.10.15).
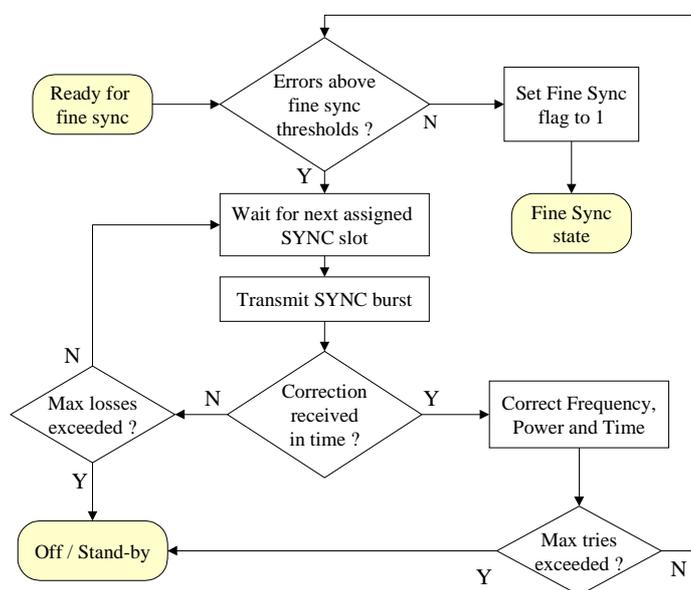


**Figure 37: Fine synchronization procedure (optional)**

NOTE:      It may be necessary to average a number of error measurements in order to achieve sufficient accuracy.

## 7.6 Synchronization maintenance procedure

Upon achieving fine synchronization, the RCST is allowed to transmit TRF bursts. It shall in parallel proceed to maintain synchronization. This procedure is carried out continuously for the duration of the session as shown in figure 38. However, in case the NCC has indicated to the RCST that a security sign-on was required by the network (see the Logon Initialize descriptor clause 8.5.5.10.4 or mesh logon initialize descriptor clause 8.5.5.10.22), the RCST shall wait for the security handshake to occur (see clause 9.4). Once it has returned a "Security Sign-on Response" message, the RCST is allowed to transmit user traffic in its allocated TRF timeslots.

Loop counts, timeouts and thresholds are indicated to the RCST through the SYNC_Assign descriptor embedded in a TIM (see clause 8.5.5.10.6).

Parameters for this procedure are contained in the correction_control_descriptor (see clause 8.5.5.10.15).
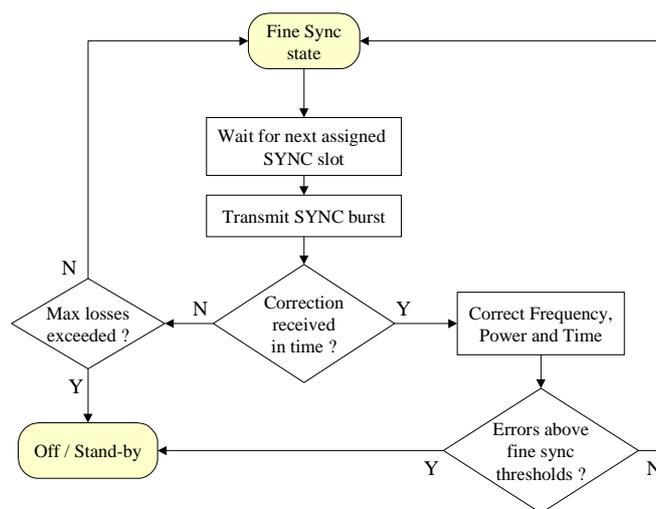


**Figure 38: Synchronization maintenance procedure**

## 7.7 Logoff procedure

### 7.7.1 General

The RCST logoff procedure described in this clause applies when the RCST is in the Fine Sync state.

When the RCST logs-off, it shall cease all transmission and its logical address and SYNC timeslot shall be removed from the active list and made available to other RCSTs that may join the network. The logoff procedure is initiated as a result of a session termination (normal) or of a failure (abnormal) as described in the following clauses.

### 7.7.2 Normal

A normal logoff can be initiated either automatically or manually by the user at the end of a session. For an RCST initiated logoff, a logoff request message shall be transmitted in the M&C clause. For an NCC initiated logoff, the logoff message shall be carried by a Terminal Information Message (TIM) addressed to the RCST (see clause 8.5.5.8).

### 7.7.3 Abnormal

An RCST shall logoff in the following conditions:

- loss of receive synchronization, i.e.:
    - NCR not received for several consecutive seconds;
    - CMT burst correction not received for several consecutive SYNCs.

# 8 Control and management

This clause defines the messages to allow an RCST to logon to the satellite interactive network. These will be used to co-ordinate an identification of the calling RCST, a process to adjust the power of the RCST, and a logon procedure which gives an identification to the RCST that can be used to transmit meta-signalling to request traffic connections. The following clauses detail the protocol stacks used in the forward and return link and each one of the messages exchanged between the NCC/Gateway and RCST and vice-versa.

As a minimum set of requirements the RCST shall comply with the Control and Monitoring Functions (CMF) specified in [5] if applicable. Among others, the present document requires that the RCST is only allowed to transmit, when it receives its control correctly.

## 8.1 Protocol stack

On the return link the protocol stack is based on ATM cells or optional MPEG2-TS packets mapped onto TDMA bursts. For transmission of IP datagrams, the protocol stacks used on the return link are as follows:

- ATM based return link: IP/AAL5/ATM, as defined in [6].

- Optional MPEG return link: multiprotocol over MPEG2 Transport Streams encapsulation as defined in [i.2] and [4].

In the forward link the protocol stack is based on the DVB/MPEG2-TS standard (see TR 101 154 [i.4]). For transmission of IP datagrams, the protocols stacks used on forward link are as follows:

- IP over multiprotocol encapsulation over MPEG2 Transport Streams, as defined in [i.2] and [4].

- optionally IP/AAL5/ATM/MPEG-TS in data piping mode as defined in [i.3] so as to enable direct terminal to terminal communications in regenerative satellite systems.

### 8.1.1 RCST Type A (IP)

The RCST Type A shall be able to support IP services only.

ATM cells are used on the return link to benefit from the AAL5 segmentation and re-assembly (SAR) function. The mechanism to carry IP over ATM AAL5 shall be as specified in RFC 2684 [6], see clause 4.1 (Payload Format for Routed IP PDUs). From the two multiplexing methods defined there the one called "VC based multiplexing of Routed Protocols", shall be applied. The ATM cells shall be mapped into MF-TDMA traffic bursts as defined in clause 6.2.1.1. The 53 bytes of the ATM cell are made of 5 bytes of header and 48 bytes of payload. ATM cells are used either for user traffic or for control and/or management traffic (handled by upper layers). For type A RCSTs, the 5-byte header of an ATM cell shall follow the ATM UNI cell format, as represented on figure 39.
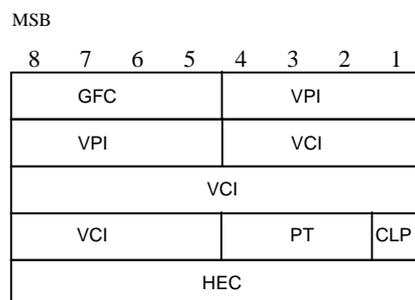


**Figure 39: ATM cell header format**

The use of the different fields shall be as follow:

- GFC: reserved. This field has another meaning when the optional security mechanism is used (see clause 9.4.6.3).

- VPI: set to the VPI value transmitted by the network to the terminal after logon (see clause 8.5.5.10.4), or a VPI value obtained through upper layer signalling.

- VCI: set to the VCI value transmitted by the network to the terminal after logon (see clause 8.5.5.10.4), or a VCI value obtained through upper layer signalling.

A given VPI/VCI pair shall not be allocated by the network to more than one RCST at a given time.

- PT: This field shall be used as follows:

  - bit 2 of octet 4 shall be used for AAL5 Segmentation and Reassembly process (AUU bit), as defined in ITU-T Recommendation I.363-5 [15];

  - bit 3 of octet 4 is normally used by ATM signalling to indicate if the cell has experienced congestion. This bit shall be set to 0;

  - bit 4 of octet 4 (normally used by ATM signalling to indicate a traffic cell). This bit shall be set to 0 for traffic cells and to 1 for CTRL/MNGM bursts (see clause 6.6.2).

- CLP: Cell Loss Priority. This bit is used to indicate the cells which would be first deleted in case of buffer congestion in one of ATM network nodes. This bit shall be set to 0.

- HEC: this field shall be generated as described in ITU-T Recommendation I.432 [10].

**Optional data piping mode:**

User Traffic is sent between the User Device and the Gateway or optionally between two user devices. In systems using an ATM based return link, sending traffic between two user devices in regenerative satellite systems implies that terminals will need to support the data piping mode as defined in [i.3].

**Optional MPEG mode:**

Alternatively the IP SAR function can be provided by the DVB multiprotocol encapsulation method according to TR 101 202 [i.2] using MPEG2 Transport Stream packets as the container. The packets shall be mapped into MF-TDMA traffic bursts as defined in clause 6.2.1.2.

The PID(s) for user traffic and, if applicable, CTRL/MNGM messages (see clause 6.6.2) to be used in the return link MPEG TS packets are set by the logon initialize descriptor (see clause 8.5.5.10.4) or mesh logon initialize descriptor (see clause 8.5.5.10.22), or obtained through upper layer signalling.

User traffic is sent between the User Device and a Gateway, or optionally between two user devices, whereas signalling is sent only between the NIU and the NCC. Figure 40 shows an example of a protocol stack for user traffic and signalling.
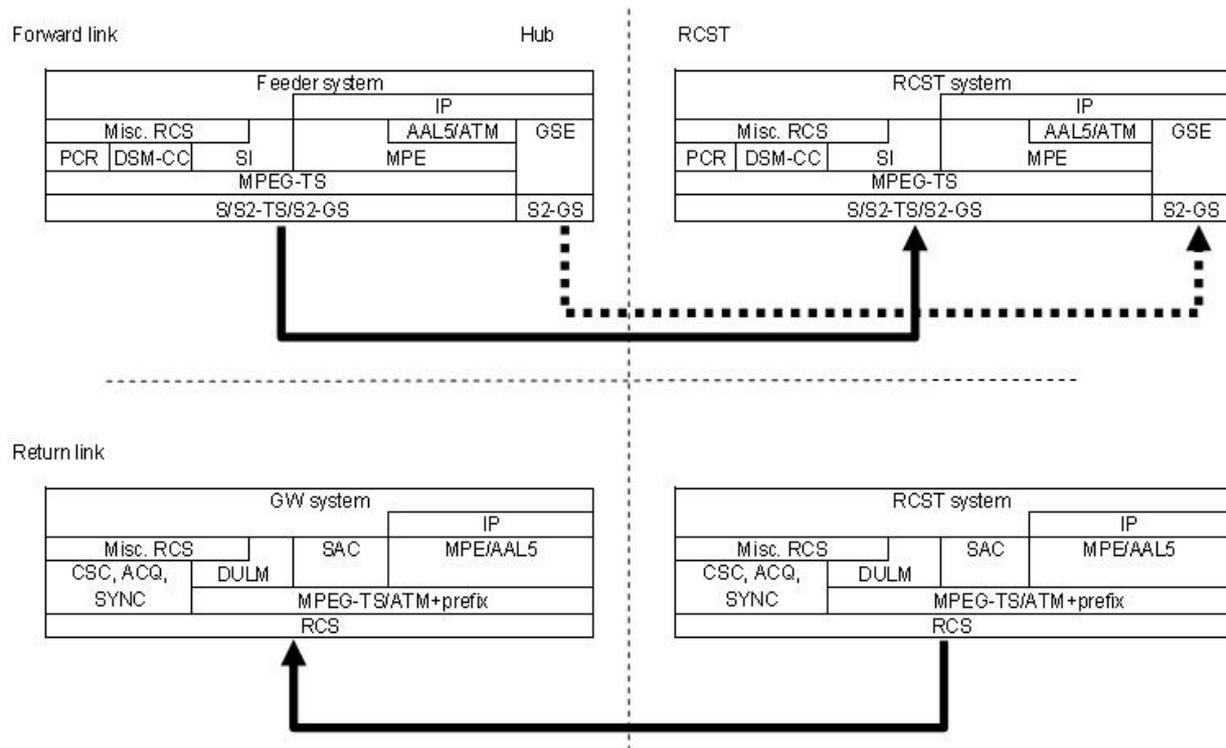
**Figure 40: Example of protocol stack with Type A RCST**
**(IP/AAL5/ATM/MPEG2/DVBS is optional in the forward link)**

## 8.1.2 Optional RCST Type B (Native ATM)

The RCST Type B shall be able to operate as RCST Type A and shall also be able to support native ATM protocols by encapsulating ATM cells within an MPEG2 Transport Stream on the forward link, as defined in TR 100 815 [i.3]. On the return link ATM cells shall be mapped into MF-TDMA traffic bursts as defined in clause 6.2.1.1.

The RCST can support Permanent Virtual Circuits (PVCs) and Switched Virtual Circuits (SVCs) on the forward- and return link as an UNI. Standard compliant signalling according to ITU-T Recommendation Q.2931 [8] shall be used. In difference to a normal ATM environment there is a shared medium between the terminal and the gateway. During the Logon procedure a VPI/VCI is assigned to the RCST by the Logon initialize descriptor or mesh logon initialize descriptor that is carried in the TIM, see clauses 8.5.5.10.4, 8.5.5.10.22 and 8.5.5.8. The assigned VPI/VCI shall overwrite the values 0/5 that are normally used for ITU-T Recommendation Q.2931 [8] signalling.

ATM connectivity in RCSTs is optional.

## 8.2 RCST addressing

On the Forward Link, RCSTs shall be uniquely identified by a physical MAC address and a logical address, except that, when the optional provisions of clause 6.4.5 are applied, the MAC address shall be replaced by the Non-line-of-sight RCST identifier defined in clause 6.4.5.5.

The *MAC address* is a physical address stored in non-volatile memory. It corresponds to a unique RCST physical identifier. It shall follow the IEEE 802.3 [9] standard and shall consist of 48 bits. The value 0xFFFFFFFFFFFF shall be reserved for broadcasting to all RCSTs. The MAC address shall be used inside a CSC burst and in DSM-CC private data sections that carry TIMs. It will also be used to encapsulate IP datagrams into MPEG2-TS frames TR 101 202 [i.2].

The logical address is composed of two fields, the Group_ID and Logon_ID which are assigned to each RCST during logon. They are used for addressing individual RCSTs until logoff.

The *Group_ID* corresponds to a group of logged-on RCSTs. It shall consist of 8 bits. The value 0xFF shall be reserved for system use.

The *Logon_ID* uniquely identifies the RCST within a group identified with the Group_ID. The Logon_ID shall consist of 16 bits. The value 0xFFFF shall be reserved for system use (contention mode on the return link).

For a Type A RCST any data (user traffic) that is destined to a specific RCST shall be transmitted using the RCST MAC address. Any data (user traffic) that is destined to all Type A RCSTs shall be transmitted using the broadcast MAC address (0xFFFFFFFFFFFF).

Independently, each protocol used at higher layers may impose its own addressing scheme, e.g. IP addresses, etc.

# 8.3     Forward link signalling

DVB defines a set of tables built upon the MPEG PSI tables to provide detailed information regarding the broadcast network. Such DVB tables are referred as the Service Information (SI) tables. In a two-way Satellite Interactive Network, consisting of a forward and return link via satellite, medium access control information and other signalling are communicated through the forward link and shall be transmitted in a DVB compliant manner. Thus, the specifications for Service Information (SI) in DVB systems EN 300 468 [3] shall apply. The forward link signalling consists of general SI tables, carrying information about the structure of the satellite interactive network, and RCST specific messages sent to individual RCSTs, private data fields defined for standard DVB-SI tables, special Transport Stream packets (PCR Insertion) and descriptors, including private descriptors for standard DVB-SI tables.

## 8.3.1     General SI tables

General SI data describing the Satellite Interactive Network organization are structured as six types of table. These tables are broadcast in private sections (see ISO/IEC 13818-1 [7]). Where applicable the use of descriptors allows a flexible approach to the organization of the tables and allows for future compatible extensions. The precise definition of the table content is given in clause 8.5.5.

### 8.3.1.1     Superframe Composition Table (SCT)

This table describes the sub-division of the entire satellite interactive network into superframes and frames. The table contains for each superframe, a superframe identification, a centre frequency, an absolute start time expressed as an NCR value and a superframe count. Each superframe is further divided into frames. Each type of frame is identified by a frame_id. The frame position within a superframe is given by a frame number used for timeslot assignments in the TBTP. The frame numbering convention is described in clause 6.7.2. The frames are positioned relative to the centre frequency and start time of the associated superframe.

### 8.3.1.2     Frame Composition Table (FCT)

This table describes the partitioning of the frames into time-slots. The table contains for each frame_id (i.e. for each frame type) a frame duration, the total number of timeslots contained in the frame, the start times and frequency offsets for the timeslots. The transmission parameters (such as symbol rate, code rate, preamble, etc.) for each timeslot are referred by a time-slot identifier (timeslot_id) to a description conveyed in the TCT.

### 8.3.1.3     Time-Slot Composition Table (TCT)

This table defines the transmission parameters for each time-slot type identified by the time-slot identifier. It provides information about the timeslot properties such as symbol rate, code rate, preamble, payload content (TRF with ATM cells, TRF with MPEG2 TS packets, CSC, ACQ, SYNC) and others.

### 8.3.1.4     Satellite Position Table (SPT)

This table contains the satellite ephemeris data required to update the burst position at regular intervals. The table shall contain this data for those satellites that form an active part of a particular Satellite Interactive Network.

### 8.3.1.5        Correction Message Table (CMT)

The NCC sends the Correction Message Table to groups of RCSTs. The purpose of the CMT is to advise the logged-on RCSTs what corrections shall be made to their transmitted bursts. The CMT provides correction values for burst frequency, timing and amplitude. The CMT contains the corrections for the RCSTs with the most recently measured ACQ and SYNC bursts.

### 8.3.1.6        Terminal Burst Time Plan (TBTP)

This message is sent by the NCC to a group of terminals. The group is addressed by a logical Group_ID, while each individual terminal is addressed by a logical Logon_ID. Both Group_ID and Logon_ID are notified to the terminal at logon time. It contains one or more entries for each RCST, with each entry defining an assignment of a contiguous block of timeslots. Each traffic assignment is described by the number of the start timeslot in the block and a repetition factor giving the number of consecutive timeslot allocations.

The TBTP allows timeslots to be allocated once or continuously. In the latter case a mechanism is provided to add or remove time slot allocations from the terminal burst time plan.

## 8.3.2       Terminal Information Message (TIM)

This message is sent by the NCC either to an individual RCST addressed by its MAC address (uni-cast message) or broadcast to all RCSTs using a reserved broadcast MAC address and contains static or quasi static information about the forward link such as configuration.

This message may also be used to facilitate the handing over of an RCST to a different group or network group or network or to switch a group of RCSTs to a different forward link signalling service on another MPEG2-TS for example. This message is sent in a DSM-CC private data section (see ISO/IEC 13818-6 [14]).

## 8.3.3       PCR Insertion TS Packet

The PCR Insertion TS Packet shall be used for inserting the NCR value used for return link synchronization. The PID shall be assigned via the PMT of the Forward Link Signalling service (see clause 8.5.5.11). The optional payload may contain information about the delay between all relevant satellite to NCC and satellite to Gateway combinations.

## 8.3.4       Summary

Figure 41 gives an overview of the protocol stack for forward link signalling. This protocol stack corresponds to the protocol shown in figure 40.
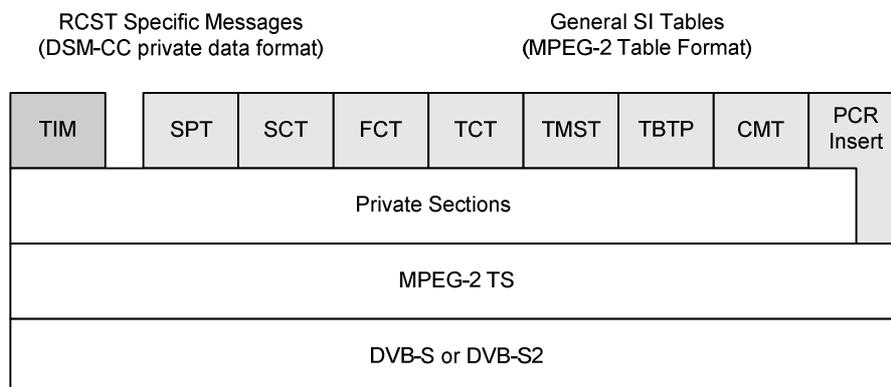


**Figure 41: Protocol stack for forward signalling**

## 8.3.5    Repetition rates

All sections of the SCT, FCT, TCT, SPT, TMST and broadcast TIM shall be transmitted at least every 10 seconds to allow newly activated RCSTs to acquire the necessary start-up state. In addition, the TIM shall be updated as required to reflect system status changes requiring immediate notification of the RCSTs.

The TBTP shall be updated every superframe.

The update rate of the NCR value in the PCR Insertion TS Packet shall be between 200 times per second and 10 times per second. If used, the optional PCR Insertion TS packet payload section shall be transmitted at least once a second.

The uni-cast TIM shall be updated as needed to reflect changes affecting a given RCST.

# 8.4    Return link signalling

## 8.4.1    RCST synchronization and Identification messages

The NCC manages RCST synchronization. The synchronization process is a set of signalling messages exchanged between the NCC and the RCST on reserved timeslots, which allow fine tuning of all synchronization parameters, timing, frequency, and power (see clause 7).

The messages described below are used for the synchronization of the RCST. The ACQ and SYNC bursts are special bursts containing symbols dedicated for the NCC to be able to measure frequency and timing offset.

The NCC requests an RCST to transmit a sequence of ACQ bursts via the TIM by assigning a BTP ACQ time slot for a limited number of repeats. In systems which can tolerate the required signalling bandwidth, the NCC requests an RCST to transmit an ACQ bursts as and when required by an ACQ burst assignment signalled via the TBTP.

In systems using periodic SYNC transmissions, the NCC also requests an RCST to periodically transmit a SYNC burst via the TIM by assigning a SYNC time slot once every N superframes. In systems using ad hoc (non periodic) SYNC transmissions, the NCC requests an RCST to transmit a SYNC burst as and when required by a SYNC timeslot assignment signalled via the TBTP.

The NCC determines power, frequency and burst time error of the RCST. Corrections for frequency and burst time are sent in a CMT for correction by the RCST.

| | |
|---|---|
| **Forward:** | **Message/DSM-CC and SI Section/MPEG2-TS/(DVB-S or DVB-S2)** |
| **Return:** | **Special bursts/Air Interface** |
| **Messages used:** | **TIM (forward) - [DSM-CC] or TBTP [SI]** |
| | **CMT (forward) - [SI]** |
| | **CSC (return)** |
| | **ACQ (return)** |
| | **SYNC (return)** |

## 8.4.2　Configuration parameters between RCST and NCC (optional)

The NCC uses the configuration parameters exchanged between RCST and NCC to identify the functional capability of the RCST and therefore what transmission characteristics it can demand from that particular RCST. In addition the NCC can obtain information such as manufacturer identification, RCST version (number of forward link receivers, RCST type (amplifier power), user identification, hardware version, software version, RCST position (latitude, longitude, altitude), Outdoor Unit characteristics (power, antenna size and antenna gain), type of RCST connection (SMATV, single user, multiple user), installer identification, postal code of the area, date and time of installation and more). A private Management Information Base (MIB) in the RCST stores the configuration parameter values in variables. The NCC uses SNMP commands to obtain the current configuration parameter values from the RCST MIB. The NCC sets a flag in the RCST MIB to acknowledge receipt of the configuration parameter values. An SNMP agent in the RCST responds to commands from an SNMP client in the NCC. This exchange of configuration parameters is optional, the RCST indicates in the CSC burst (see clause 6.2.3) if it implements SNMP.

**Forward:**　　　　**SNMP/UDP/IP/DSM-CC/MPEG2-TS/(DVB-S or DVB-S2)**

**Return:**　　　　**SNMP/UDP/IP/Traffic bursts/Air Interface**

**Messages:**　　　　**get-request [MIB variable] (forward)**

　　　　　　　**get-next-request [MIB variable] (forward)**

　　　　　　　**get-response [MIB variable, value] (return)**

　　　　　　　**set-request [acknowledgement flag] (forward)**

NOTE:　Private MIB variables are out of the scope of the present document.

## 8.4.3　Other messages for network management (optional)

The NCC and RCST send other SNMP messages whenever they are needed for network management. Such messages implement installation procedures, software upgrades, transmit authorization or prohibition, individual/group control and traffic forward link assignment, RCST status enquiries, and RCST or NCC requests to leave the network. The NCC queries MIB values to determine RCST status and stores values in the MIB variables to trigger RCST actions. The RCST sends *trap* messages to notify the NCC when it has accomplished triggered actions or to issue requests. For instance, the NCC sends a *set-request* message with a MIB variable value to authorize or prohibit transmission. Similarly, the RCST sends a *trap* message to the NCC to request to leave the network. This exchange of messages is optional, the RCST indicates in the CSC burst (see clause 6.2.3) if it implements SNMP.

**Forward:**　　　　**SNMP/UDP/IP/DSM-CC/MPEG2-TS/(DVB-S or DVB-S2)**

**Return:**　　　　**SNMP/UDP/IP/Traffic bursts/Air Interface**

**Messages:**　　　　**get-request [MIB variable] (forward)**

　　　　　　　**get-next-request [MIB variable] (forward)**

　　　　　　　**get-response [MIB variable] (return)**

　　　　　　　**set-request [MIB variable, value] (forward)**

　　　　　　　**trap [MIB variable value, value] (return)**

NOTE:　Private MIB variables are out of the scope of the present document.

## 8.4.4    Burst time plan exchange

The Burst Time Plan (BTP) is sent to all the RCSTs affected by using the Terminal Burst Time Plan message (TBTP). This information is the basis for the RCSTs to know when to transmit their bursts. Capacity Requests (CR) messages are sent by the RCSTs to the NCC for all volume based connections depending on the data present in the queue. For constant bit rate connections, the TDMA scheduler at the NCC will automatically update the BTP according to the set-up parameters.

**Forward:**          **Message/SI Table/MPEG2-TS/(DVB-S or DVB-S2)**

**Return:**           **Capacity requests (CR)/Air Interface**

**Messages:**        **TBTP (forward)**

                     **CR (return)**

# 8.5    Coding of SI for forward link signalling

## 8.5.1    Introduction

Forward Link Signalling for Satellite Interactive Network is divided into five parts:

- general SI Tables (SCT, FCT, TCT, SPT, TBTP and CMT);

- RCST Specific Messages (TIM);

- Private Data fields defined for standard DVB-SI tables;

- Special Transport Stream packets (PCR Insertion);

- descriptors, including private descriptors for standard DVB-SI tables.

The following clauses describe the coding and definition of these tables, messages and descriptors.

## 8.5.2    SI table mechanism

SI Tables for Satellite Interactive Network are transmitted in private sections as defined in ISO/IEC 13818-1 [7], Systems.

The syntax and semantics of SI tables defined in EN 300 468 [3] for the segmentation of tables in one or more sections, the section length, the section identification, the mapping of sections into Transport Streams, the repetition rates and the random access shall also be applicable to SI tables for Satellite Interactive Network defined in the present document.

If required, the tables defined in the present document may be scrambled to prohibit traffic analysis. If the tables are scrambled, the same mechanisms as defined in EN 300 468 [3] for scrambled SI tables shall apply.

## 8.5.3    DSM-CC section mechanism

RCST Specific Messages for Satellite Interactive Network are transmitted in DSM-CC private data sections as defined in ISO/IEC 13818-6 [14] in general and in EN 300 468 [3] for DVB.

If required, RCST Specific Messages defined in the present document may be scrambled to prohibit traffic analysis. They may be scrambled on an individual basis.

## 8.5.4    Coding of PID and table_id fields

**Table 28: PID and table_id allocation SI Tables**

| Table and private data sections defined in the present document | PID | Table_id |
|---|---|---|
| Reserved | | 0x00 to 0x40 |
| RMT | Assigned (see note) | 0x41 |
| Reserved | | 0x42 to 0x9F |
| SCT | Assigned | 0xA0 |
| FCT | Assigned | 0xA1 |
| TCT | Assigned | 0xA2 |
| SPT | Assigned | 0xA3 |
| CMT | Assigned | 0xA4 |
| TBTP | Assigned | 0xA5 |
| PCR packet payload | Assigned | 0xA6 |
| Reserved | | 0xA7 |
| Reserved | | 0xA8 |
| Reserved | | 0xA9 |
| Transmission Mode Support Table | Assigned | 0xAA |
| Reserved for future use | | 0xAB to 0xAF |
| TIM | Assigned | 0xB0 |
| LL_FEC_parity_data_table | Assigned | 0xB1 |
| Reserved for future use | | 0xB2 to 0xBF |
| User defined | | 0xC0 to 0xFE |
| NOTE: This PID shall be defined to be a given value across all interactive networks in a given system. | | |

Table 28 lists the PID and table_id values which shall be used for the TS packets which carry SI tables and RCST Specific Messages defined in the present document.

## 8.5.5    Table definitions

The following clauses describe the syntax and semantics of the different types of table.

NOTE:    The symbols and abbreviations, and the method of describing syntax used in the present document are the same as those defined in clauses 2.2 and 2.3 of ISO/IEC 13818-1 [7].

The mnemonics defined in clause 2.2.6 of ISO/IEC 13818-1 [7] are used in the tables of the present document. For convenience, the present document defines a further three mnemonics to cover frequently used parameter formats, as follows:

- spfmsbf = single precision floating point value, which is a 32 bit value formatted in accordance with ANSI/IEEE Standard 754 [13]. The most significant bit (i.e. the most significant bit of the exponent) is first.

- upcrmsf = unsigned PCR count value. The coding of this type of parameter shall be identical to the coding of the program_clock_reference (PCR) described in ISO/IEC 13818-1 [7], comprising a base field of up to 33 bits and a 9 bit extension field. Where the number of bits is less than the full 42 bit PCR format, then the least significant 9 bits shall correspond to the extension field and the remaining bits shall correspond to the least significant bits of the base field. The most significant bit is first.

- flagmsf = multi-bit boolean flags field (e.g. status byte/word), with the most significant bit first. Each flag is asserted on a logic "1". In an extension to the ISO/IEC 13818-1 [7] syntax, an individual flag is referenced using standard object reference "dot" notation, i.e. the "foo" flag in parameter "status" is referenced as "status.foo" and takes the value TRUE if the foo flag bit is "1".

### 8.5.5.1    Standard section headers

The following standard headers have been tailored for forward link signalling use, and represent a specific subset of the more general formats specified in the DVB and ISO standards.

8.5.5.1.1          SI section header

**Table 29: Standard SI section header**

| Syntax | No. of bits | Mnemonic |
|---|---|---|
| table_id | 8 | uimsbf |
| section_syntax_indicator | 1 | bslbf |
| reserved_for_future_use | 1 | bslbf |
| reserved | 2 | bslbf |
| section_length | 12 | uimsbf |
| interactive_network_id | 16 | uimsbf |
| reserved | 2 | bslbf |
| version_number | 5 | uimsbf |
| current_next_indicator | 1 | bslbf |
| section_number | 8 | uimsbf |
| last_section_number | 8 | uimsbf |

The standard header for a SI section occupies a total of 64 bits, and shall be as defined in table 29.

Semantics for the standard SI section header:

- table_id: This 8 bit field identifies the table. See table 28 for the table_id values.

- section_syntax_indicator: The section_syntax_indicator is a 1-bit field which shall be set to "1".

- section_length: This is a 12-bit field, the first two bits of which shall be "00". It specifies the number of bytes of the section, starting immediately following the section_length field and including the CRC. The section_length shall not exceed 1 021 so that the entire section has a maximum length of 1 024 bytes.

- interactive_network_id: This is a 16-bit field, which serves as a label to identify the Satellite Interactive Network, to which the table shall apply.

- version_number: This 5-bit field is the version number of the sub_table. The version_number shall be incremented by 1 when a change in the information carried within the sub_table occurs. When it reaches value 31, it wraps around to 0. When the current_next_indicator is set to "1", then the version_number shall be that of the currently applicable sub_table defined by the table_id and interactive_network_id_mask. When the current_next_indicator is set to "0", then the version_number shall be that of the next applicable sub_table defined by the table_id and interactive_network_id.

- current_next_indicator: This 1-bit indicator, when set to "1" indicates that the sub_table is the currently applicable sub_table. When the bit is set to "0", it indicates that the sub_table sent is not yet applicable and shall be the next sub_table to be valid.

- section_number: This 8-bit field gives the number of the section. The section_number of the first section in the sub_table shall be "0x00". The section_number shall be incremented by 1 with each additional section with the same table_id and interactive_network_id.

- last_section_number: This 8-bit field specifies the number of the last section (that is, the section with the highest section_number) of the sub_table of which this section is part.

8.5.5.1.2        DSM-CC private section header

**Table 30: Standard DSM-CC private section header**

| Syntax | No. of bits | Mnemonic |
|---|---|---|
| table_id | 8 | uimsbf |
| section_syntax_indicator | 1 | bslbf |
| private_indicator | 1 | bslbf |
| reserved | 2 | bslbf |
| section_length | 12 | uimsbf |
| MAC_address_6 | 8 | uimsbf |
| MAC_address_5 | 8 | uimsbf |
| reserved | 2 | bslbf |
| payload_scrambling_control | 2 | bslbf |
| address_scrambling_control | 2 | bslbf |
| LLC_SNAP_flag | 1 | bslbf |
| current_next_indicator | 1 | bslbf |
| section_number | 8 | uimsbf |
| last_section_number | 8 | uimsbf |
| MAC_address_4 | 8 | uimsbf |
| MAC_address_3 | 8 | uimsbf |
| MAC_address_2 | 8 | uimsbf |
| MAC_address_1 | 8 | uimsbf |

The standard header for a DSM-CC private section occupies a total of 96 bits, and shall be as defined in table 30.

Semantics for the standard DSM-CC private section header:

- table_id: This 8 bit field identifies the table. See table 28 for the table_id values.

- section_syntax_indicator: The section_syntax_indicator is a 1-bit field which shall be set to "1" to denote that a CRC32 check field is used at the end of the section.

- private_indicator: The private_indicator is a 1 bit field that shall be set to the complement of the section_syntax_indicator (i.e. to "0").

- section_length: This is a 12-bit field, the first two bits of which shall be "00". It specifies the number of bytes of the section, starting immediately following the section_length field and including the CRC. The section_length shall not exceed 1 021 so that the entire section has a maximum length of 1 024 bytes.

- MAC_address_[1 to 6]: This 48 bit field contains the MAC address of the destination. The MAC address is fragmented in 6 fields of 8 bits, labelled MAC_address_1 to MAC_address_6. The MAC_address_1 field contains the most significant byte of the MAC address, while MAC_address_6 contains the least significant byte.

NOTE:      The order of the bits in the byte is not reversed, and the MSB of each byte is still transmitted first.

- payload_scrambling_control: This 2 bit field defines the scrambling mode of the payload section (see table 31). This includes the payload starting after the MAC_address_1 but excludes the CRC32 field. The scrambling method applied is user private. If the optional security mechanism described in clause 9.4 is active, this field shall be as defined in clause 9.4.6.3.

**Table 31: Coding of the payload_scrambling_control field**

| value | payload scrambling control |
|---|---|
| 00 | unscrambled |
| 01 | defined by service |
| 10 | defined by service |
| 11 | defined by service |

- address_scrambling_control: This 2 bit field defines the scrambling mode of the MAC address section (see table 32). This field enables a dynamic change of MAC addresses. The scrambling method applied is user private.

**Table 32: Coding of the address_scrambling_control field**

| value | address scrambling control |
|-------|---------------------------|
| 00 | unscrambled |
| 01 | defined by service |
| 10 | defined by service |
| 11 | defined by service |

- LLC_SNAP_flag: This 1 bit flag shall be set to "0" to indicate that the payload does not use LLC/SNAP encapsulation.

- current_next_indicator: This 1-bit field shall be set to "1".

- section_number: This 8-bit field gives the number of the section. The section_number of the first section in the message shall be "0x00". The section_number shall be incremented by 1 with each additional section for the same message.

- last_section_number: This 8-bit field specifies the number of the last section (that is, the section with the highest section_number) of the message of which this section is part.

## 8.5.5.2 Superframe Composition Table (SCT)

The SCT (see table 33) conveys information relating to the organization of the satellite interactive network, in particular the sub-division of the framing structure. The combination of the interactive_network_id and the superframe_id allows each superframe to be uniquely identified. To each satellite interactive network is assigned an individual interactive_network_id which serves as a unique identification code. The interactive_network_id shall be unique throughout a given network.

The SCT shall be segmented into superframe composition sections using the syntax described in EN 300 468 [3]. Any sections forming part of an SCT shall be transmitted in TS packets with a PID value assigned in the PMT. Any sections of the SCT shall have the table_id value as defined in table 28.

**Table 33: Syntax of superframe composition section**

| Syntax | No. of bits Reserved (see note) | No. of bits Information | Information Mnemonic |
|--------|--------------------------------|-------------------------|---------------------|
| Superframe_composition_section(){ | | | |
| SI_private_section_header | | 64 | - |
| superframe_loop_count | | 8 | uimsbf |
| for(i=0;i<=superframe_loop_count;i++){ | | | |
| superframe_id | | 8 | uimsbf |
| Large_Timing_Uncertainty_Flag | | 1 | bslbf |
| uplink_polarization | 5 | 2 | bslbf |
| superframe_start_time_base | | 33 | uimsbf |
| superframe_start_time_ext | 6 | 9 | uimsbf |
| superframe_duration | | 32 | upcrmsf |
| superframe_centre_frequency | | 32 | uimsbf |
| superframe_counter | | 16 | uimsbf |
| frame_loop_count | 3 | 5 | uimsbf |
| for(j=0;j<=frame_loop_count;j++) { | | | |
| frame_id | | 8 | uimsbf |
| frame_start_time | | 32 | upcrmsf |
| frame_centre_frequency_offset | | 24 | tcimsbf |
| } | | | |
| } | | | |
| CRC_32 | | 32 | rpchof |
| } | | | |
| NOTE: Reserved bits are of type bslbf, and shall precede the Information bits on the same line. | | | |

Semantics for the superframe_composition_section:

- SI_private_section_header: This is the standard SI private section header defined in table 29, and occupies a total of 64 bits.

- superframe_loop_count: This is an 8 bit field indicating one less than the number of iterations in the loop that follows. A zero count indicates one loop.

- superframe_id: This is an 8-bit field which serves as a label for identification of this superframe from any other superframe within the satellite interactive network.

- Large Timing Uncertainty_Flag: This flag when set to "0" indicates that the large timing uncertainty is supported on the related Super_Frame and that the Correction_Message_Extension_Descriptor will be sent together with the Correction_Message_Descriptor.

- uplink_polarization: This is a 2-bit field specifying the polarization of the transmitted signal (see table 34).

**Table 34: Polarization definition**

| Polarization | Value |
|---|---|
| linear - horizontal | 00 |
| linear - vertical | 01 |
| circular - left | 10 |
| circular - right | 11 |

- superframe_start_time_base and superframe_start_time_ext: These two fields give the absolute time of the beginning of the superframe identified by the superframe_id. The coding of the fields shall be identical to the coding of the program_clock_reference (PCR) described in ISO/IEC 13818-1 [7], with the two fields corresponding to the base and extension parts of the PCR respectively.

NOTE: The separation of base and extension by 6 reserved bits matches the format widely used in other DVB tables for PCR values.

- superframe_duration: This 32-bit field gives the duration of the superframe identified by the superframe_id, in terms of PCR count intervals. The 32 bits correspond to a maximum duration of 93,2 s.

- superframe_centre_frequency: This 32-bit field gives the absolute centre frequency of the superframe identified by the superframe_id. The frequency is given in multiples of 100 Hz.

- superframe_counter: This 16 bit field gives the superframe count value, modulo 65 536. This is used to avoid ambiguity in the processing of the TBTP message.

- frame_loop_count: This 5 bit field indicates one less than the number of iterations in the loop that follows. A zero count indicates one loop. The frame numbers follow the numbering convention defined in clause 6.7.2.

- frame_id: Gives the frame type identifier for the $j^{th}$ frame, corresponding to a frame type defined in the FCT.

- frame_start_time: This 32 bit field gives the start time of the $j^{th}$ frame relative to the superframe start time, in terms of PCR count intervals. The 32 bits correspond to a maximum duration of 93,2 s. For continuous carrier operation this parameter is only used as timing reference for carrier assignment control; it does not represent a property of the carrier.

- frame_centre_frequency_offset: This 24-bit field gives the signed offset of the centre frequency of the $j^{th}$ frame relative to the superframe_centre_frequency parameter (SCT). The frequency is given in multiples of 100 Hz.

- CRC_32: This is a 32-bit field that contains the CRC value that gives a zero output of the registers in the decoder defined in annex B of EN 300 468 [3] after processing the entire section.

## 8.5.5.3        Frame Composition Table (FCT)

The FCT conveys information describing the different frame types. This table defines the structure in the frequency/time space for each frame type.

The FCT shall be as defined in table 35. It shall be segmented into frame composition sections using the syntax described in EN 300 468 [3]. Any sections forming part of an FCT shall be transmitted in TS packets with a PID value assigned in the PMT. Any sections of the FCT shall have the table_id value as defined in table 28.

**Table 35: Syntax of frame composition section**

| Syntax | No. of bits Reserved (see note) | Information | Information Mnemonic |
|---|---|---|---|
| Frame_composition_section(){ | | | |
|   SI_private_section_header | | 64 | - |
|   frame_ID_loop_count | | 8 | uimsbf |
|   for(i=0;i<=frame_ID_loop_count;i++){ | | | |
|     frame_id | | 8 | uimsbf |
|     frame_duration | | 32 | upcrmsf |
|     total_timeslot_count | 5 | 11 | uimsbf |
|     start_timeslot_number | 5 | 11 | uimsbf |
|     timeslot_loop_count | | 8 | uimsbf |
|     for(j=0;j<=timeslot_loop_count;j++){ | | | |
|       timeslot_frequency_offset | | 24 | tcimsbf |
|       timeslot_time_offset | | 32 | upcrmsf |
|       timeslot_id | | 8 | uimsbf |
|       repeat_count | | 8 | uimsbf |
|     } | | | |
|   } | | | |
|   CRC_32 | | 32 | rpchof |
| } | | | |
| NOTE:    Reserved bits are of type bslbf, and shall precede the Information bits on the same line. | | | |

Semantics for the frame_composition_section:

- SI_private_section_header: This is the standard SI private section header defined in table 29, and occupies a total of 64 bits.

- frame_ID_loop_count: This is a 8-bit field indicating one less than the number of iterations of the frame loop that follows. A zero count indicates one loop.

- frame_id: This 8-bit field serves as a label for identification of the ith frame type from any other frame type.

- frame_duration: This 32-bit field gives the duration of the ith frame identified by the frame_id, in terms of PCR count intervals. The 32 bits correspond to a maximum duration of 93,2 s. For continuous carrier operation this parameter is only used as timing reference for carrier assignment control; it does not represent a property of the carrier.

- total_timeslot_count: This 11 bit field defines the total number of timeslots in the ith frame.

- start_timeslot_number: This 11 bit field defines the number of the first timeslot of the ith frame defined in this section, following the numbering scheme defined in clause 6.7.2. This simplifies the partitioning of a frame across sections, since the definition of a non-homogenous frame format can exceed the capacity of one section.

- timeslot_loop_count: This is a 8-bit field indicating one less than the number of iterations of the timeslot loop that follows. A zero count indicates one loop. The order follows the numbering scheme defined in clause 6.7.2, starting with the start_timeslot_number.

- timeslot_frequency_offset: This 24-bit field gives the signed value of the offset of the centre frequency of the jth timeslot group relative to the frame_centre_frequency parameter (FCT). The frequency is given in multiples of 100 Hz.

- timeslot_time_offset: This 32 bit field gives the time offset of the jth timeslot group relative to the frame start time, in terms of PCR count intervals. The 32 bits correspond to a maximum duration of 93,2 s. For continuous carrier operation this parameter is only used as timing reference for carrier assignment control; it does not represent a property of the carrier.

- timeslot_id: This 8 bit field identifies the type of timeslot for the jth timeslot group, and corresponds to a timeslot_id defined in the TCT.

- repeat_count: This 8 bit field value is the number of repeats of the preceding timeslot_id. The value is one less than the total number of successive timeslots of the same type. E.g. a value of 0 indicates no repeats (1 occurrence only), while a value of 2 indicates 2 further repeats for a total of 3. All repeats shall have the same timeslot_frequency_offset value, but the timeslot_time_offset value shall be increased from the prior timeslot by the timeslot_duration value in the TCT for that timeslot_id. The value of repeat_count shall be set to 0 when defining instances of time slots representing continuous carriers.

- CRC_32: This is a 32-bit field that contains the CRC value that gives a zero output of the registers in the decoder defined in annex B of EN 300 468 [3] after processing the entire section.

## 8.5.5.4 Timeslot Composition Table (TCT)

The TCT (see table 36) conveys information relating to the properties of the types of timeslot, such as TRF with ATM cells, TRF with MPEG2-TS packets, CSC, SYNC and ACQ. The timeslot_id allows each kind of timeslot to be uniquely identified. Only timeslot properties are described within this table, the time and frequency co-ordinates of a particular timeslot are contained in the SCT/FCT.

**Table 36: Timeslot composition section**

| Syntax | No. of bits | | Information Mnemonic |
|---|---|---|---|
| | Reserved (see note) | Information | |
| Timeslot_composition_section(){ | | | |
|    SI_private_section_header | | 64 | - |
|    timeslot_loop_count | | 8 | uimsbf |
|    For(i=0;i <= timeslot_loop_count;i++){ | | | |
|      timeslot_id | | 8 | uimsbf |
|      symbol_rate | | 24 | uimsbf |
|      timeslot_duration | | 24 | upcrmsf |
|      burst_start_offset | | 16 | upcrmsf |
|      inner_code_type | | 1 | bslbf |
|      inner_code_ordering | | 1 | bslbf |
|      outer_coding | | 2 | bslbf |
|      inner_code_puncturing | | 4 | bslbf |
|      modulation | | 5 | bslbf |
|      baseband_shaping | | 3 | bslbf |
|      timeslot_payload_type | | 8 | uimsbf |
|      Route_ID_flag | | 1 | bslbf |
|      ACM_flag | | 1 | bslbf |
|      MOB-flag | | 1 | bslbf |
|      SAC_length | | 5 | bslbf |
|      request_flag | | 1 | bslbf |
|      M_and_C_flag | | 1 | bslbf |
|      Group_ID_flag | | 1 | bslbf |
|      Logon_ID_flag | | 1 | bslbf |
|      capacity_requests_number | | 3 | bslbf |
|      New_permutation | | 1 | bslbf |
|      If((Inner_code_type == 1) and (New_permutation == 1)) { | | | |
|        P0 | 3 | 5 | uimsbf |
|        P1 | 6 | 10 | uimsbf |
|        P2 | 6 | 10 | uimsbf |
|        P3 | 6 | 10 | uimsbf |
|      } | | | |
|      preamble_length | | 8 | uimsbf |
|      for (j=0;j<preamble_length;j++){ | | | |
|        preamble_symbol | | 2 | bslbf |
|      } | | | |
|      while (!bytealigned){ | | | |
|        stuffing_bit } | | 1 | bslbf |
|    } | | | |
|    CRC_32 | | 32 | rpchof |
| } | | | |
| NOTE: Reserved bits are of type bslbf, and shall precede the Information bits on the same line. | | | |

The TCT shall be segmented into timeslot composition sections using the syntax described in EN 300 468 [3]. Any sections forming part of a TCT shall be transmitted in TS packets with a PID value assigned in the PMT. Any sections of the TCT shall have the table_id as defined in table 28.

Semantics for the timeslot_composition_section:

- SI_private_section_header: This is the standard SI private section header defined in table 29, and occupies a total of 64 bits.

- timeslot_loop_count: This is a 8-bit field indicating one less than the number of iterations of the timeslot loop that follows.

- timeslot_id: This is an 8-bit field which serves as a label for identification of the ith timeslot including all properties of the slot.

- symbol_rate: This 24 bit field gives the symbol rate in symbol/s.

- timeslot_duration: This 24 bit field defines the duration of the timeslot, in terms of PCR count intervals. The 24 bits corresponds to a maximum duration of 364 ms. The value of timeslot_duration shall be set to 0 for timeslots representing continuous carriers.

- burst_start_offset: This 16 bit field defines the burst start offset from the slot start, in terms of PCR count intervals. The 16 bits corresponds to a maximum offset of 1,42 ms.

- inner_code_type: This bit specifies the inner coding type to use. A "0" means the K = 7 convolutional code (see clause 6.4.3) and a "1" means the Turbo Code (see clause 6.4.4).

- inner_code_ordering: This bit specifies the order of transmission of the encoded block for Turbo Code. A "0" means the natural order, a "1" the reverse order (see clause 6.4.4.4). This bit is reserved for the K = 7 convolutional code.

- outer_coding: This is a 2-bit field specifying the outer coding mode to be used. The value "01" indicates that the RS code, as specified in clause 6.4.2, is switched on. The value "10" indicates that the CRC error detection code, as described in clause 6.4.1 is used. The value "00" indicates that both the RS and CRC codes are used. The value "11" indicates that neither the CRC nor the RS are used.

- inner_code_puncturing: This is a 4-bit field (see table 37) specifying the inner coding rates to be used in combination with the selected inner coding. When using continuous carrier for the return link, this is the inner coding to be used unless the RCST is allowed to use other inner coding as indicated in the optional Return Transmission Mode descriptor.

**Table 37: Inner coding**

| Code Rate | Value 0000 0001 | K = 7 0011 0100 | Turbo 1111 | Contin. Carrier (Note 2) | Contin. Carrier (Note 4) |
|---|---|---|---|---|---|
| 1/2 | 0000 | x | x | x | x |
| 2/3 | 0001 | x | x | x | Not used |
| 3/4 | 0010 | x | x | x | x |
| 5/6 | 0011 | x | Not used | x | Not used |
| 7/8 | 0100 | x | Not used | Not used | Not used |
| 1/3 | 0101 | Not used | x | x | Not used |
| 2/5 | 0110 | Not used | x | x | Not used |
| 4/5 | 0111 | Not used | x | x | Not used |
| 6/7 | 1000 | Not used | x | Not used | Not used |
| 1/4 | 1001 | Not used | Not used | x | x |
| 3/5 | 1010 | Not used | Not used | x | Not used |
| 8/9 | 1011 | Not used | Not used | x | Not used |
| 9/10 | 1100 | Not used | Not used | x (Note 3) | Not used |
| Reserved for future use | 1101 to 1110 | | | | |
| Inner code is omitted | 1111 | x | x | x | |
| NOTE 1: x Means supported.<br>NOTE 2: DVB-S2 with Normal and Short frames.<br>NOTE 3: DVB-S2 with Normal frames.<br>NOTE 4: DVB-S2 extension with Very Short frames. | | | | | |

- modulation: This is a 5-bit field which serves as an identifier of the modulation scheme.as defined in table 38. When using return link spectrum spreading, the first bit (MSB) is equal to 1, the second bit is equal to 0 if the modulation is $\pi/2$-BPSK and 1 if the modulation is QPSK. When using return link spectrum spreading, the last three bits (LSBs) determine the spreading factor $F$ as defined in table 39. When using continuous carrier operation in the return link, this is the modulation scheme to be used unless the RCST is allowed to use other modulation schemes as indicated in the optional Return Transmission Mode descriptor.

**Table 38: Modulation scheme**

| Modulation | Value | MF-TDMA | Continuous Carrier |
|---|---|---|---|
| Not defined | 00000 | | |
| QPSK | 00001 | Yes | Yes |
| 8PSK | 00010 | | Yes |
| 16APSK | 00011 | | Yes |
| 32APSK | 00100 | | Yes |
| π/2-BPSK (No spreading) | 00101 | Mobile | Yes |
| Reserved for future use | 00110 to 01111 | | |
| π/2-BPSK with spreading | 10000 to 10111 | Mobile | Yes |
| QPSK with spreading | 11000 to 11111 | Mobile | Yes |

**Table 39: Return link spreading factors**

| Spreading Factor | Modulation Scheme LSBs |
|---|---|
| 2 | 000 |
| 3 | 001 |
| 4 | 010 |
| 6 | 011 |
| 8 | 100 |
| 10 | 101 |
| 13 | 110 |
| 16 | 111 |

- baseband_shaping: This is a 3-bit field which serves as an identifier of the used baseband square-root cosine roll-off shaping as defined in table 40.

**Table 40: Return link baseband shaping**

| Roll off factor | MF-TDMA | Continuous Carrier |
|---|---|---|
| 0.35 | 000 | 000 |
| 0.25 | Not Used | 001 |
| 0.20 | Not Used | 010 |
| Reserved for future use | 011 to 111 | |

- timeslot_payload_type: This is an 8-bit field which serves as a label for identification of the type of timeslot. The values are assigned in table 41. The range of "user defined" values can e.g. be used to signal the system-dependent use of continuous carriers with transmission schemes other than DVB-S2. When using continuous carrier for the return link, this indicates the format to be used unless the RCST is allowed to use other formats as indicated in the optional Return Transmission Modes descriptor.

**Table 41: Timeslot types**

| Timeslot type | Value |
|---|---|
| Reserved | 0x00 |
| TRF with one ATM cell | 0x01 |
| TRF with two ATM cells | 0x02 |
| Reserved | 0x03 |
| TRF with four ATM cells | 0x04 |
| TRF with MPEG2-TS packet(s) | 0x05 |
| CSC | 0x06 |
| ACQ | 0x07 |
| SYNC | 0x08 |
| DVB-S2 normal frame with pilots | 0x09 |
| DVB-S2 normal frame without pilots | 0x0A |
| DVB-S2 short frame with pilots | 0x0B |
| DVB-S2 short frame without pilots | 0x0C |
| DVB-S2 extension very short frame with pilots | 0x0D |
| Reserved for future use | 0x0E to 0x7F |
| User defined | 0x80 to 0xFF |

- Route_ID_flag: The value "0" indicates that the SAC field of this time slot contains a route_ID sub-field as described in clause 6.6.1.1. The value "1" indicates that there is no route_ID sub-field.

- ACM_flag: The value "0" indicates that the SAC field of this time slot contains an ACM sub-field as described in clause 6.6.1.1. The value "1" indicates that there is no ACM sub-field.

- MOB_flag: The value "0" indicates that the SAC field of this time slot contains a Mobility_Control_Message as described in 6.6.1.1. The value "1" indicates that there is no Mobility_Control_Message sub-field.

- SAC_length: This is a 5-bit field specifying the length of the SAC field in bytes not including an optional CRC.

- Request_flag: The value "1" indicates that the SAC field of this time slot contains at least one Request sub-field as described in clause 6.6.1.1. In this case, the number of requests is indicated with the capacity_requests_number field described here below. The value "0" indicates that there is no Request sub-field.

- M_and_C_flag: The value "1" indicates that the SAC field of this time slot contains an M&C sub-field as described in clause 6.6.1.1. The value "0" indicates that there is no M&C sub-field.

- Group_ID_flag: The value "1" indicates that the SAC field of this time slot contains a Group_ID sub-field as described in clause 6.6.1.1. The value "0" indicates that there is no Group_ID sub-field.

- Logon_ID_flag: The value "1" indicates that the SAC field of this time slot contains a Logon_ID sub-field as described in clause 6.6.1.1. The value "0" indicates that there is no Logon_ID sub-field.

- capacity_requests_number: This 3-bit field indicates one less than the number of capacity_requests allowed in the SAC as described in clause 6.6.1.1. In networks using the TRF-prefix signalling method (capacity requests appended to TRF bursts), this field would normally have a value of 0 ("000"). The number of CRs does not need to be indicated for the prefix method. The prefix must be of 4 bytes to allow one 3-byte CR. When using 4 byte prefix only for CR, either one 3-byte CR or two 2-byte CRs will fit in the 4-byte space. If the maximum number of CRs is indicated (different from zero), it defines the maximum number of 2-byte CRs that can be fitted. This field is reserved if the request flag is equal to "0".

- New_permutation: When set to 0, this bit specifies that the turbo code permutation to be used is the default one as defined in table 9. When set to 1, this bit indicates that the permutation parameters are defined in the sequel of the TCT. This bit is reserved if the turbo code is not used (inner_code_type==0).

- P0, P1, P2 and P3: Define the set of permutation parameters to be used for the specific burst type.

- preamble_length: This is an 8-bit field specifying the preamble length in symbols. The loop which follows is aligned however on byte boundaries. This means that for example if the preamble length is 14 Symbols, the loop over the preamble bytes will consist of 4 bytes with the last 4 bits being merely stuffing bits.

- preamble_symbol: This is a 2-bit field specifying a QPSK symbol of the preamble. Symbols are given in the order they shall be transmitted. The first bit specifies the I bit and the second bit the Q bit of a QPSK symbol. As the preamble is not scrambled, a proper sequence shall be selected in order to comply with ETSI requirements concerning off-axis EIRP.

- stuffing_bit: Since the preamble description is byte aligned, stuffing bits are present until the next byte boundary. The stuffing bits may take any value and shall be discarded by the RCST.

- CRC_32: This is a 32-bit field that contains the CRC value that gives a zero output of the registers in the decoder defined in annex B of EN 300 468 [3] after processing the entire section.

## 8.5.5.5       Satellite Position Table (SPT)

The SPT conveys information about the positions of all the satellites used for the forward and return links. In the case where the NCC computes the nominal RCST-to-satellite range in 27 MHz clock periods at RCST registration, the SPT table needs not be transmitted.

The SPT shall be segmented into satellite position sections using the syntax described in EN 300 468 [3]. Any sections forming part of an SPT shall be transmitted in TS packets with a PID value assigned in the PMT. Any sections of the SPT shall have the table_id value as defined in table 28. The SPT is defined in table 42.

**Table 42: Syntax of satellite position table section**

| Syntax | No. of bits | | Mnemonic |
|---|---|---|---|
| | Reserved (see note) | Information | |
| satellite_position_section(){ | | | |
| SI_private_section_header | | 64 | - |
| satellite_loop_count | | 8 | uimsbf |
| for(i=0;i<=satellite_loop_count;i++){ | | | |
| satellite_id | | 8 | uimsbf |
| x_coordinate | | 32 | spfmsbf |
| y_coordinate | | 32 | spfmsbf |
| z_coordinate | | 32 | spfmsbf |
| } | | | |
| CRC_32 | | 32 | rpchof |
| } | | | |
| NOTE:      Reserved bits are of type bslbf, and shall precede the Information bits on the same line. | | | |

Semantics for the satellite_position_section:

- SI_private_section_header: This is the standard SI private section header defined in table 29, and occupies a total of 64 bits.

- satellite_loop_count: This 8 bit field defines one less than the number of satellites belonging to the system. A zero count indicates one loop.

- satellite_id: This 8 bit field defines a system assigned satellite identifier.

- x_coordinate: This 32 bit field defines the x co-ordinate of the satellite ephemeris in meters.

- y_coordinate: This 32 bit field defines the y co-ordinate of the satellite ephemeris in meters.

- z_coordinate: This 32 bit field defines the z co-ordinate of the satellite ephemeris in meters.

- NOTE:      The position of the satellites will be expressed as Cartesian coordinates x, y, z in the geodetic reference frame ITRF96 (IERS Terrestrial Reference Frame). This system coincides with the WGS84 (World Geodetic System 84) reference system at the one meter level.

- CRC_32: This is a 32 bit field that contains the CRC value that gives a zero output of the registers in the decoder defined in annex B of EN 300 468 [3] after processing the entire section.

### 8.5.5.6          PCR Insertion Transport Stream packet

The PCR Insertion Transport Stream (TS) packet is used to insert the NCR count for return link synchronization purpose.

### 8.5.5.6.1          TS packet format

The format of the TS packet shall be as defined in ISO/IEC 13818-1 [7], clause 2.4.3. The value of the fields in the TS packet (ISO/IEC 13818-1 [7], table 2.3) shall be as follows:

- sync_byte: as defined in [7].

- transport_error_indicator: set to "0".

- payload_unit_start_indicator: if the payload of the TS packet contains the PCR Insertion TS packet payload section, the payload_unit_start_indicator value shall be "1", indicating that the first byte of the payload of this Transport Stream packet carries a pointer field. This pointer field is an 8-bit field whose value shall be the number of bytes, immediately following the pointer field until the first byte of the section that is present in the payload of the Transport Stream packet (so a value of 0x00 in the pointer field indicates that the section starts immediately after the pointer field).

- transport_priority: set to "0".

- PID: set to the assigned PID value.

- transport_scrambling_control: as defined in [7], depending on whether or not transport stream scrambling is used.

- adaptation_field_control field: set to "10" whenever a PCR value only is being sent (no payload) and set to "11" to indicate that the packet contains an adaptation field followed by payload to indicate that an update to one or more of the forward or return link combinations data is present.

- adaptation_field(): present - see adaptation field definition below (see clause 8.5.5.6.2).

- data_byte [loop]: present if adaptation_field_control field set to "11" - see payload definition below (see clause 8.5.5.6.3).

### 8.5.5.6.2          Adaptation field

The adaptation field shall be coded as per ISO/IEC 13818-1 [7] clause 2.4.3.4, with field values as follows:

- adaptation_field_length: set to 7;

- discontinuity_indicator: set to "0";

- random_access_indicator: set to "0";

- elementary_stream_priority_indicator: set to "0";

- PCR_flag: set to "1";

- OPCR_flag: set to "0";

- splicing_point_flag: set to "0";

- transport_private_data_flag: set to "0";

- adaptation_field_extension_flag: set to "0";

- program_clock_reference_base; program_clock_reference_extension: set to the inserted NCR value, as defined in [7].

The remaining fields of the adaptation field are not applicable.

### 8.5.5.6.3          Optional payload field

The optional payload of the TS packet may contain the delays between NCCs and satellites as well as between Traffic Gateways and satellites. The corresponding syntax shall be conforming to the one defined in the standard ISO/IEC 13818-1 [7] for the definition of private sections. The section shall be entirely contained in the payload of the TS packet containing the NCR.

If the payload is used to transmit delay information, the syntax in table 43 shall be used.

**Table 43: Syntax of the optional PCR Insertion TS packet payload section**

| Syntax | No of bits | | Information mnemonic |
|---|---|---|---|
| | Reserved | Information | |
| PCR Insertion TS packet payload section (){ | | | |
| SI_private_section_header | | 64 | - |
| forward_link_combinations | | 8 | uimsbf |
| for (i=0; i<forward_link_combinations; i++){ | | | |
|     satellite_id | | 8 | uimsbf |
|     NCC_id | | 8 | uimsbf |
|     propagation_delay } | | 32 | upcrmsf |
| return_link_combinations | | 8 | uimsbf |
| for (i=0; I<return_link_combinations; i++) { | | | |
|     satellite_id | | 8 | uimsbf |
|     gateway_id | | 8 | uimsbf |
|     propagation_delay} | | 32 | upcrmsf |
| descriptor_length | | 8 | uimsbf |
| for(i=0; i<N; i++) { | | | |
|     descriptor()} | | | |
| CRC_32 | | 32 | rpchof |
| } | | | |

- SI_private_section_header: This is the standard SI private section header defined in table 29, and occupies a total of 64 bits.

- forward_link_combinations: This 8 bit field defines the number of all applicable NCC to satellite combinations on the forward link.

- satellite_id: This 8 bit field defines the identifier of the satellite end of one combination. The satellite_id is assigned by the system administrator. The RCST shall match this satellite_id with the satellite_id provided in the Satellite Forward Link Descriptor defined in clause 8.5.5.10.11.

- NCC_ id: This 8 bit field defines the identifier of the NCC end of one combination. The NCC_id is assigned by the system administrator. The RCST shall match this NCC_id with the NCC_id provided in the Satellite Forward Link Descriptor defined in clause 8.5.5.10.11.

- propagation_delay: This 32 bit field defines the propagation_delay between NCC and satellite as a PCR count. For the forward link it is the delay from NCC to satellite while for the return link it is the delay from satellite to Gateway. The 32 bits corresponds to a maximum delay of 93,2 s. RCSTs may use this information to compute delays. When the optional PCR TS packet payload section is present, and the forward link combination matches the satellite_id and NCC_id in use, the RCST shall add the value of the delay included to the result of the delay calculation based on the RCST position and the Satellite Position Table defined in clause 8.5.5.5; i.e. the transmission instant shall be advanced by an amount equal to the value of the forward link propagation_delay, compared to that determined without considering the propagation_delay. The value of the delay included in the optional PCR Insertion TS packet payload section is dependent on the implementation of the NCC. When the optional PCR TS packet payload section is present, and the return link combination matches the satellite_id and gateway_id in use, the RCST shall add the value of the delay included to the delay calculation based on the RCST position and the Satellite Position Table. When applicable values for both forward and return link are present, the correction shall be the sum of the two values.

NOTE:    If the values provided correspond to the actual delays between NCC and satellite and between satellite and gateway respectively, the corresponding correction will place the system timing reference plane at the gateway. If zero values are transmitted, or if no applicable correction values are provided, the reference plane will be located at the satellite. The location of the reference plane is a system design decision.

- return_link_combinations: This 8 bit field defines the number of all applicable satellite to Gateway combinations on the return link.

- Gateway_ id: This 8 bit field defines the identifier of the Gateway end of one combination. The Gateway_id is assigned by the system administrator. The RCST shall match this gateway_id with the gateway_id provided in the Satellite Return Link Descriptor defined in clause 8.5.5.10.12.

- descriptor_length: This 8 bit field gives the total length in bytes of future descriptors.

- CRC_32: This is a 32 bit field that contains the CRC value that gives a zero output of the registers in the decoder defined in annex B of EN 300 468 [3] after processing the entire section.

## 8.5.5.7      Terminal Burst Time Plan (TBTP)

The TBTP defines the dynamic timeslot assignments generated every superframe.

The TBTP shall be as defined in table 44. It shall be segmented into TBTP sections using the syntax described in EN 300 468 [3]. Any sections forming part of a TBTP shall be transmitted in TS packets with a PID value assigned in the PMT. Any sections of the TBTP shall have the table_id as defined in table 28.

**Table 44: Terminal Burst Time Plan section**

| Syntax (see note 2) | No. of bits | | Mnemonic |
|---|---|---|---|
| | Reserved (see note 1) | Information | |
| Terminal_burst_time_plan(){ | | | |
| SI_private_section_header | | 64 | - |
|    Group_ID | | 8 | uimsbf |
|    Superframe_count | | 16 | uimsbf |
|   frame_loop_count | 3 | 5 | uimsbf |
|   for (i=0;i<=frame_loop_count;i++) { | | | |
|     frame_number | 3 | 5 | uimsbf |
|     BTP_loop_count | 5 | 11 | uimsbf |
|     for (j=0;j<= BTP_loop_count;j++){ | | | |
|       Logon_ID | | 16 | uimsbf |
|       Multiple_channels_flag | | 1 | bslbf |
|       Assignment_type | | 2 | bslbf |
|       VBDC_queue_empty_flag | | 1 | *bslbf* |
|       Start_slot | 1 | 11 | uimsbf |
|       If (Multiple_channels_flag == 1) | | | |
|         If (8-bit Channel_ID is *not* supported) | | | |
|           Channel_ID | 4 | 4 | uimsbf |
|         If (8-bit Channel_ID *is* supported) | | | |
|           Channel_ID | | 8 | See text |
|       Assignment_count} | | 8 | uimsbf |
|     } | | | |
|    CRC_32 | | 32 | rpchof |
| } | | | |
| NOTE 1:   Reserved bits are of type bslbf, and shall precede the Information bits on the same line. | | | |
| NOTE 2:   The use of the 8-bit format of the Channel_ID is restricted to situations where this format is known to be supported by the specific RCST. If this is not explicitly known, the 4-bit format of the Channel_ID shall be used towards the specific RCST. | | | |

Semantics for the terminal_burst_time_plan:

- SI_private_section_header: This is the standard SI private section header defined in table 29, and occupies a total of 64 bits.

- Group_ID: This is an 8 bit field that identifies the service group for which the message applies. See clause 8.2.

- Superframe_count: This 16 bit field identifies the modulo 65,536 superframe count to which the BTP applies.

- Frame_loop_count: This 5 bit field specifies one less than the number of superframe frame loops that follow. A zero count indicates one loop. Each entry in the loop corresponds to the definition of one frame in the superframe.

- frame_number: This 5 bit field specifies the frame number within the superframe, using the frame numbering as defined in clause 6.7.2.

- BTP_loop_count: This 11-bit field specifies one less than the number of the frame BTP entry loops that follow. A zero count indicates one loop. Each entry in the loop provides a definition of a block of timeslots allocated. Where a given terminal is assigned multiple blocks, there will be one entry per block. The entries for a given terminal may not be consecutive in the section.

- Logon_ID: This 16 bit field gives the identifier assigned to the terminal at logon time. This number is limited in scope to within the Group_ID (i.e. the same Logon_ID can be given to a different terminal within a different Group_ID). See clause 8.2.

- Multiple_channels_flag: This 1-bit flag indicates the presence of a Channel_ID field for the current assignment. A value of 1 corresponds to the Channel_ID field being present. If the field is equal to 0, channel_id=0 shall be assumed by the RCST.

- Assignment_Type: This 2 bit field defines the nature of the assignment, as defined in table 45.

**Table 45: Assign type**

| Assign type for MF-TDMA time slots | Assign type for time slots defining continuous carriers | Value |
|---|---|---|
| one time assignment | carrier assignment | 00 |
| repeating assignment | permanent carrier assignment | 01 |
| assignment release | carrier release | 10 |
| Reserved | permanent carrier release | 11 |

VBDC_queue_empty_flag: This 1 bit flag is set to "1" when the VBDC queue for that RCST is empty after the assignments in the table, and is set to "0" when the VBDC queue is not empty. If the Multiple_channels_flag is set to "1", then the queue is that for the channel identified by the Channel_ID field;

- Start_slot: This 11 bit field gives the number of the first timeslot in the block, relative to the frame (numbering as defined in clause 6.7.2).

- Channel_ID: This 4-bit or 8-bit field indicates the channel to which timeslots are being assigned. Values are managed by the NCC. This field is present if the multiple_channels_flag is set to "1". The 4-bit field is used if the 8-bit field is not supported; it carries Channel_ID values in the range 0-15. The 8-bit field is used if it is supported; it is a uimsbf field that carries Channel_ID values in the range 0-255, coded as {(Channel_ID + 240) modulo 256}. Support for the 8-bit Channel_ID is indicated at logon time in the CSC burst, as defined in clause 6.2.3.

- Assignment_count: This 8 bit field gives one less than the number of timeslots assigned in the block. These timeslots will be consecutively numbered in ascending order starting with the start_slot.

- CRC_32: This is a 32 bit field that contains the CRC value that gives a zero output of the registers in the decoder defined in annex B of EN 300 468 [3] after processing the entire section.

## 8.5.5.8    Terminal Information Message (TIM)

The TIM is used to transmit static or quasi-static information about the forward and return links, such as configuration parameters. The format supports two variants of this message:

- A terminal specific (uni-cast) message containing information for a specific terminal. This variant may be encrypted for security reasons. It is sent during logon initialization, and when a parameter changes. When a TIM contains multiple descriptors, the terminal shall process all descriptors before changing its configuration.

- A broadcast message giving general information applicable to all terminals. This variant is unlikely to be encrypted. It will need to be transmitted sufficiently often that newly powered terminals can acquire any necessary information within a reasonable time. This matches a similar requirement for the SCT, FCT and

TCT signals, suggesting the same repeat interval (see clause 8.3.5). It is recommended that the RCST is pre-programmed with suitable default values for parameters normally contained in the descriptors in the broadcast TIM, in particular those in the Contention Control Descriptor (see clause 8.5.5.10.14) and the Correction Control Descriptor (see clause 8.5.5.10.15). Selection of suitable default values will ordinarily be made by the manufacturer or network operator, prior to installation of the RCST. Values contained in the broadcast table shall override the defaults.

**Table 46: Terminal Information Message section**

| Syntax | No. of bits | | Information Mnemonic |
|---|---|---|---|
| | Reserved (see note) | Information | |
| terminal_information_message_section() { | | | |
|    DSM-CC_private_section_header | | 96 | |
|    If MAC_address == RCST MAC (not broadcast) { | | | |
|      RCST_Status } | | 8 | flagmsf |
|    else { | | | |
|      Network_status } | | 8 | flagmsf |
|    descriptor_loop_count | | 8 | uimsbf |
|    for (i= 0; i<= descriptor_loop_count; i++) { | | | |
|      descriptor() } | | | |
|    If MAC_address == RCST MAC (not broadcast) { | | | |
|      Pad_bytes } | | see text | bslbf |
|    CRC_32 | | 32 | rpchof |
| } | | | |
| NOTE: Reserved bits are of type bslbf, and shall precede the Information bits on the same line. They shall be ignored by the RCST. For an encrypted uni-cast TIM, the bit values shall be varied in a random manner to avoid encryption spoofing. | | | |

The TIM shall be as defined in table 46. It shall be segmented into terminal information message sections using the syntax described in EN 300 468 [3]. Any sections forming part of a TIM shall be transmitted in TS packets with a PID value assigned in the PMT. Any sections of the TIM shall have the table_id as described in table 28.

Semantics for the terminal_information_message_section:

- DSM-CC_private_section_header: This is the standard DSM-CC private section header defined in table 30, and occupies a total of 96 bits.

- RCST_Status: This 8 bit field gives status flags defining the network state of the RCST (see table 47).

**Table 47: RCST status**

| Bit | Identifier |
|---|---|
| (MSB) 7 | ID_encrypt |
| 6 | Logon_fail_(busy) |
| 5 | Logon_denied |
| 4 | Log_off |
| 3 | Transmit_Disable |
| 2 | Rain_Fade_release |
| 1 | Rain_Fade_detect |
| (LSB) 0 | Wake_up |

Semantics for the flag bits are as follows, where a logic "1" asserts the condition defined:

- ID_encrypt: Indicates that the terminal shall use TBTP logon ID encryption.

- Logon_fail_(busy): Indicates that the terminal cannot enter the network because of lack of resources.

- Logon_denied: Indicates that the terminal is not authorized to enter the network.

- Log_off: Indicates that the network has enforced a terminal logoff. The terminal shall immediately cease transmission and enter the logged off state.

- Transmit_Disable: Indicates that the terminal shall immediately cease transmission and enter the hold state, and shall not resume until it receives a uni-cast TIM with this bit reset to "0".

- Rain_Fade_release: Indicates that the NCC is performing a reconfiguration procedure to restore settings following cessation of a rain fade event.

- Rain_Fade_detect: Indicates that the NCC has detected a rain fade event and is performing a reconfiguration procedure to establish rain fade settings.

- Wake_up: Indicates that the NCC wants to wake up the RCST.

- Network_Status: This 8 bit field gives status flags defining the network state for RCSTs within the scope of the broadcast MAC address. The flag bits shall be as defined in table 48.

**Table 48: Network status**

| Bit | Identifier |
|---|---|
| (MSB) 7 | ID_encrypt |
| 6 | Reserved |
| 5 | Reserved |
| 4 | CSC_link_failure_rec |
| 3 | Link_failure_recovery |
| 2 | Return_link_failure |
| 1 | NCC_Receive_Failure |
| (LSB) 0 | Scheduler_Failure |

Semantics for the flag bits are as follows, where a logic "1" asserts the condition defined:

- ID_encrypt: Indicates that terminals shall use TBTP logon ID encryption.

- CSC_link_failure_rec: Indicates that the system is recovering from a failure of a forward or return link. The NCC affects CSC bursts in the TBTP. Terminals shall wait for their reserved timeslot or use one of the CSC contention timeslots.

- Link_failure_recovery: Indicates that the system is recovering from a failure of a forward or return link. Terminals shall follow a pre-defined procedure for large outage recovery until a new broadcast TIM is received with this bit reset to "0".

- Return_link_failure: Indicates that the NCC has detected a failure of the return link. All RCSTs shall cease transmission until a new broadcast TIM is received with this bit reset to "0".

- NCC_Receive_Failure: Indicates that the NCC has a receive channel failure. Affected terminals shall suspend transmission until a new broadcast TIM is received indicating that the failure is no longer present.

- Scheduler_Failure: Indicates that the NCC has a scheduler failure. Affected terminals shall suspend transmission of all traffic, but not SYNC bursts, until a new broadcast TIM is received indicating that the failure is no longer present. SYNC bursts transmitted while the fault is present shall have the request field of the SAC set to "0".

- descriptor_loop_count: This 8 bit field defines one less than the number of descriptors in the following loop. A zero count indicates one loop.

- descriptor(): The descriptors that may be inserted into the TIM are defined in clause 8.5.5.10. It should be noted that the allowed descriptors are different between the unicast and broadcast versions of the TIM (as defined by the scope of the MAC address).

- Pad_bytes: Each Pad_byte shall be an 8 bit field. Sufficient Pad_bytes are inserted so as to pad the encrypted portion of the message to the encryption algorithm boundary. The content of these fields is undefined and shall be randomized to prevent code spoofing. The encrypted portion of the message starts immediately following the MAC Address fields and finishes immediately prior to the CRC_32 field.

- CRC_32: This is a 32 bit field that contains the CRC value that gives a zero output of the registers in the decoder defined in annex B of EN 300 468 [3] after processing the entire section.

### 8.5.5.9        Correction Message Table (CMT)

The CMT shall be as shown in table 49. It shall be segmented into terminal information message sections using the syntax described in EN 300 468 [3]. Any sections forming part of a CMT shall be transmitted in TS packets with a PID value assigned in the PMT. Any sections of the CMT shall have the table_id value as defined in table 28. This message provides closed loop feedback to a number of terminals to allow them to adjust the transmit power level, frequency and burst timing as required to maintain link integrity, as well as other error condition feedback messages.

**Table 49: Correction Message Table section**

| Syntax | No. of bits | | Information Mnemonic |
|---|---|---|---|
|  | Reserved (see note) | Information | |
| correction_message_table_section(){ | | | |
|    SI_private_section_header | | 64 | - |
|    Entry_loop_count | | 8 | uimsbf |
|    for (i = 0;i <= Entry_loop_count; i++) { | | | |
|      Group_ID | | 8 | uimsbf |
|      Logon_ID | | 16 | uimsbf |
|    Time_correction_flag | | 1 | bslbf |
|    Power_correction_flag | | 1 | bslbf |
|    Frequency_correction_flag | | 1 | bslbf |
|      Slot_Type | | 2 | bslbf |
|      Burst_time_scaling | | 3 | uimsbf |
|    If (Time_correction_flag == 1) | | | |
|      Burst_time_correction | | 8 | tcimsbf |
|    If (Power_correction_flag == 1) | | | |
|      Power_control_flag | | 1 | bslbf |
|    If(Power_control_flag==1) | | | |
|      Power_correction | | 7 | tcimsbf |
|    else | | | |
|      EsN0 | | 7 | tcimsbf |
|    If (Frequency_correction_flag == 1) | | | |
|      Frequency_correction} | | 16 | tcimsbf |
|    CRC_32 | | 32 | rpchof |
| } | | | |
| NOTE:      Reserved bits are of type bslbf, and shall precede the Information bits on the same line. | | | |

Semantics for the correction_message_table_section:

- SI_private_section_header: This is the standard SI private section header defined in table 29, and occupies a total of 64 bits.

- Entry_loop_count: This field specifies one less than the number of correction message loops that follow. A zero count indicates one loop.

- Group_ID: This 8 bit field defines which Group ID the RCST is assigned to, as identified by the Terminal Information Message (TIM). This matches the Group_ID used in the TBTP sections (see clause 8.5.5.7).

- Logon_ID: This 16 bit field identifies the assigned terminal logon identifier, as identified by the TIM. Nominally, this is the same as used in the TBTP sections. When scrambling is used, the Logon ID used in the CMT will be the unscrambled version, to avoid a possible compromise to the TBTP security.

- Time_correction_flag; Power_correction_flag; Frequency_correction_flag; Slot_type; Burst_time_scaling; Burst_time_correction; Power_control_flag; Power_correction; EsN0; Frequency_correction: These fields are identical to the matching fields of the Correction_message_descriptor, and define one measurement correction set. See clause 8.5.5.10.3 for a definition of the descriptor.

   NOTE:    The CMT does not use the descriptor directly, since the added descriptor_tag and descriptor_length parameters would represent a substantial, and unnecessary, overhead.

- CRC_32: This is a 32 bit field that contains the CRC value that gives a zero output of the registers in the decoder defined in annex B of EN 300 468 [3] after processing the entire section.

### 8.5.5.10 Descriptor coding

#### 8.5.5.10.1 Descriptor identification and location

Table 50 lists the descriptors defined within the present document, giving the descriptor-tag values and the intended placement within the tables.

**Table 50: Descriptor tags and locations**

| descriptor | Tag value | PMT | RMT | TIM (see note 1) Broadcast | Unicast |
|---|---|---|---|---|---|
| Reserved | 0x00 - 0x49 | | | | |
| Reserved | 0x4B - 0x9F | | | | |
| Network_layer_info_descriptor | 0xA0 | | | | X |
| Correction_message_descriptor | 0xA1 | | | | X |
| Logon_initialize_descriptor | 0xA2 | | | | X |
| ACQ_assign_descriptor | 0xA3 | | | | X |
| SYNC_assign_descriptor | 0xA4 | | | | X |
| Encrypted_Logon_ID_descriptor | 0xA5 | | | | X |
| Echo_value_descriptor | 0xA6 | | | X | X |
| Linkage_descriptor (private data) (see note 2) | 0x4A | | X | | |
| RCS_content_descriptor | 0xA7 | X | | | |
| Satellite_forward_link_descriptor | 0xA8 | | X | | X |
| Satellite_return_link_descriptor | 0xA9 | | X | | X |
| table_update_descriptor | 0xAA | | | X | |
| Contention_control_descriptor | 0xAB | | | X | |
| Correction_control_descriptor | 0xAC | | | X | |
| Forward_interaction_path_descriptor | 0xAD | | | | X |
| Return_interaction_path_descriptor | 0xAE | | | | X |
| Connection_control_descriptor | 0xAF | | | | X |
| Mobility control descriptor | 0xB0 | | | | X |
| Correction_message_extension_descriptor | 0xB1 | | | | X |
| Return_Transmission_Modes_descriptor | 0xB2 | | | X | X |
| Mesh_Logon_initialize_descriptor | 0xB3 | | | | X |
| Reserved | 0xB4 | | | | X |
| Implementation_type_descriptor | 0xB5 | | | X | |
| LL_FEC_identifier_descriptor | 0xB6 | | | X | X |
| Reserved | 0xB7 - 0xDF | | | | |
| User defined; can be employed in system-dependent extensions | 0xE0 - 0xFE | | | | |
| NOTE 1: For the TIM message sections, the definition of intended placement differs between the unicast and broadcast versions of that message. | | | | | |
| NOTE 2: Private extension to existing DVB descriptor. | | | | | |

After logon, the unicast TIM contains at least the following descriptors: Correction_message_descriptor, either the Logon_initialize_descriptor or the Mesh_Logon_initialize_descriptor, SYNC_assign_descriptor and Satellite_return_link_descriptor. Support of the Mesh_ Logon_initialize_descriptor is optional.

#### 8.5.5.10.2 Network Layer Info descriptor (optional)

The Network Layer Info descriptor provides a mechanism by which network level information, such as the meta-signalling VPI/VCI and base IP address, can be passed to the Management Plane of the RCST during, or prior to, the start-up configuration phase of logon. As such, the message content is passed transparently through the lower layers covered by the present document and is not defined here. The descriptor is defined in table 51.

**Table 51: Network Layer Info descriptor**

| Syntax | No. of bits | Mnemonic |
|---|---|---|
| Network_layer_info_descriptor(){ | | |
| descriptor_tag | 8 | uimsbf |
| descriptor_length | 8 | uimsbf |
| Message_body | see text | |
| } | | |

Semantics for the Network_layer_info_descriptor:

- descriptor_tag: The descriptor tag is an 8 bit field which identifies each descriptor. Its value is given in the Tag value column of table 50.

- descriptor_length: The descriptor length is an 8 bit field specifying the number of bytes of the descriptor immediately following the descriptor_length field.

- Message_body: This variable length field shall contain a datagram for passing to the target application. The length of the message body shall not exceed 255 bytes, and should preferably be limited such that the section containing this descriptor fits within a single TS packet. This datagram will take the form of an SNMP message. The messages that can be passed by this method are beyond the scope of the present document. Basic functionalities related with the network are provided with the logon initialize descriptor (see clause 8.5.5.10.4) or mesh logon initialize descriptor (see clause 8.5.5.10.22). The network layer info descriptor is optional, the RCST indicates in the CSC burst (see clause 6.2.3) if it implements SNMP.

### 8.5.5.10.3        Correction Message descriptor

The Correction Message descriptor defines a transmit parameter correction set for one terminal measurement. It shall be as defined in table 52.

**Table 52: Correction Message descriptor**

| Syntax | No. of bits | | Information Mnemonic |
|---|---|---|---|
| | Reserved (see note) | Information | |
| Correction_message_descriptor(){ | | | |
| descriptor_tag | | 8 | uimsbf |
| descriptor_length | | 8 | uimsbf |
| Time_correction_flag | | 1 | bslbf |
| Power_correction_flag | | 1 | bslbf |
| Frequency_correction_flag | | 1 | bslbf |
| Slot_Type | | 2 | bslbf |
| Burst_time_scaling | | 3 | uimsbf |
| If (Time_correction_flag == 1) | | | |
| Burst_time_correction | | 8 | tcimsbf |
| If (Power_correction_flag == 1) | | | |
| Power_control_flag | | 1 | bslbf |
| If(Power_control_flag==1) | | | |
| Power_correction | | 7 | tcimsbf |
| else | | | |
| EsN0 | | 7 | tcimsbf |
| If (Frequency_correction_flag == 1) | | | |
| Frequency_correction | | 16 | tcimsbf |
| } | | | |
| NOTE:    Reserved bits are of type bslbf, and shall precede the Information bits on the same line. For encrypted message types, the value of these bits is undefined. | | | |

Semantics for the Correction_message_descriptor:

- descriptor_tag: The descriptor tag is an 8 bit field which identifies each descriptor. Its value is given in the Tag value column of table 50.

- descriptor_length: The descriptor length is an 8 bit field specifying the number of bytes of the descriptor immediately following the descriptor_length field.

- Time_correction_flag, Power_correction_flag, Frequency_correction_flag: These three bits are used to indicated the presence of time, power and frequency correction fields, respectively, in the remainder of the descriptor.

- Slot_type: This 2 bit field identifies the type of burst being measured, as defined in table 53.

**Table 53: Slot type**

| Value | Identifier |
|-------|------------|
| 00    | TRF        |
| 01    | CSC        |
| 10    | ACQ        |
| 11    | SYNC       |

- Burst_time_scaling: This 3 bit field gives the power-of-2 scaling to apply to the Burst_time_correction parameter, i.e. a value of 2 indicates a scaling factor of 4 (= shift left 2 bits). In case there is no time correction in this descriptor, i.e. the Time_correction_flag is equal to 0, the Burst_time_scaling field shall be set to 000.

- Burst_time_correction: This 8 bit field gives the required correction to burst timing as a two's complement binary PCR clock count (i.e. in counts of the 27 MHz PCR clock) that shall be scaled according to the Burst_time_scaling field above. To minimize truncation errors, the N LSB bits of the scaled value shall be set to an approximate mid-range value of "1" followed by "0"s, with N being the value of the Burst_time_scaling field. For example, with N = 2, the resulting clock count value is "dd dddd dd10".

- Power_control_flag: This 1 bit field defines how uplink power control for the return link is carried out. The value "1"indicates that the NCC transmits a power correction value for the RCST and the value "0" indicates that the NCC transmits a measured $E_s/N_0$ value instead.

- Power_correction: This 7 bit field gives the required correction to uplink power on the return link in 0,5 dB steps as a two's complement integer value.

- EsN0: This 7 bit field gives the measured $E_s/N_0$ value on the return link in 0,5 dB steps as two's complement integer value. This value can be used to control the uplink power, as an alternative to the Power_correction value. $E_s/N_0$ is the energy per transmitted symbol, divided by the spectral density of noise and interference.

- Frequency_correction: This 16 bit field gives the required correction to frequency in 10 Hz steps, as a two's complement integer value. A negative value indicates that the terminal shall reduce frequency. For systems not implementing frequency correction, this field shall be set to all 0 s.

8.5.5.10.4        Logon Initialize descriptor

This descriptor provides parameters needed for initial logon (see table 54).

**Table 54: Logon Initialize descriptor**

| Syntax | No. of bits | | Information Mnemonic |
|---|---|---|---|
| | Reserved (see note) | Information | |
| Logon_Initialize_descriptor() { | | | |
|    descriptor_tag | | 8 | uimsbf |
|    descriptor_length | | 8 | uimsbf |
|    Group_ID | | 8 | uimsbf |
|    Logon_ID | | 16 | uimsbf |
|    Continuous_carrier | 2 | 1 | bslbf |
|    Security_handshake_required | | 1 | bslbf |
|    Prefix _flag | | 1 | bslbf |
|    Data_unit_labelling_flag | | 1 | bslbf |
|    Mini_slot_flag | | 1 | bslbf |
|    Contention_based_mini_slot_flag | | 1 | bslbf |
|    Capacity_type_flag | 1 | 1 | bslbf |
|    Traffic_burst_type | | 1 | bslbf |
|      If (Traffic_burst_type == 0) { | | | |
|        Connectivity | | 1 | bslbf |
|        If (Connectivity == 0) { | | | |
|          Return_VPI | 4 | 8 | uimsbf |
|          Return_VCI | | 16 | uimsbf |
|        } | | | |
|        Else { | | | |
|          Return_signalling_VPI | 4 | 8 | uimsbf |
|          Return_ signalling_VCI | | 16 | uimsbf |
|          Forward_ signalling_VPI | 8 | 8 | uimsbf |
|          Forward_ signalling_VCI | | 16 | uimsbf |
|        } | | | |
|      } | | | |
|      Else { | | | |
|        Return_TRF_PID | | 13 | uimsbf |
|        Return_CTRL_MNGM_PID | 3 | 13 | uimsbf |
|      } | | | |
|    if (Capacity_type_flag == 0) { | | | |
|      CRA_level | | 24 | uimsbf |
|      VBDC_max | 5 | 11 | uimsbf |
|      RBDC_max | | 24 | uimsbf |
|      RBDC_timeout | | 16 | uimsbf |
|    } | | | |
| } | | | |
| NOTE:        Reserved bits are of type bslbf, and shall precede the Information bits on the same line. They shall be ignored by the RCST. For an encrypted uni-cast TIM, the bit values shall be varied in a random manner to avoid encryption spoofing. | | | |

Semantics for the Logon_initialize_descriptor:

- descriptor_tag: The descriptor tag is an 8 bit field which identifies each descriptor. Its value is given in the Tag value column of table 50.

- descriptor_length: The descriptor length is an 8 bit field specifying the number of bytes of the descriptor immediately following the descriptor_length field.

- Group_ID: This 8 bit field defines which Group ID the terminal is assigned to. This matches the Group_ID used in the TBTP and CMT sections.

- Logon_ID: This 16 bit field identifies the assigned terminal logon identifier, which is used in the TBTP and CMT sections.

- Continuous_carrier: This flag indicates whether the descriptor applies to the MF-TDMA air interface (value "1") or the continuous carrier air interface.(Value "0").

- Security_handshake_required: The value "1" indicates that the security handshake as described in clause 9.4 is to be used. Otherwise it is set to "0".

- Prefix_flag: The value "1" indicates that the Prefix Method according to clause 6.6.1.2 is implemented in the network and the value "0" indicates that it is not implemented.

- Data_unit_labelling_flag: The value "1" indicates that the Data Unit Labelling Method according to clause 6.6.2 is implemented in the network and the value "0" indicates that it is not implemented.

- Mini_slot_flag: The value "1" indicates that the Mini-slot Method according to clause 6.6.1.3 is implemented in the network and the value "0" indicates that it is not implemented.

- Contention_based_mini_slot_flag: The value "1" indicates that the Contention-based Mini-slot Method according to clause 6.6.1.4 is implemented in the network and the value "0" indicates that it is not implemented.

- Capacity_type_flag: A value of "0" indicates that fields defining settings and limits for capacity requests are present. A value of "1" indicates that these fields are absent.

- Traffic_burst_type: This 1 bit field defines the traffic burst type to be used on the return link. The value "0" indicates ATM TRF according to clause 6.2.1.1 and the value "1" indicates optional MPEG2-TS TRF according to clause 6.2.1.2.

- Connectivity: This 1 bit field defines the connectivity to be used. The value "0" indicates IP connectivity according to clause 8.1.1 (Type A RCST) and the value "1" indicates optional ATM connectivity according to clause 8.1.2 (Type B RCST). In the case of MPEG2-TS TRF the connectivity is always IP and therefore not signalled.

- Return_VPI, Return_VCI: These fields define the VPI/VCI that the RCST shall use in ATM cells on the return link.

- Return_signalling_VPI, Return_signalling_VCI: These fields define the VPI/VCI that is used on the return link for ITU-T Recommendation Q.2931 [8] signalling instead of the normal value 0/5. The signalling is used to set up connections for traffic. These parameters can be the same for all RCSTs.

- Forward_signalling_VPI, Forward_signalling_VCI: These fields define the VPI/VCI that is used on the forward link for ITU-T Recommendation Q.2931 [8] signalling instead of the normal values 0/5. The signalling is used to set up connections for traffic.

- Return_TRF_PID: This 13 bit field defines the PID that the RCST shall use in optional MPEG2 TS packets on the return link for traffic information. This parameter can be the same for all RCSTs.

- Return_CTRL_MNGM_PID: This 13 bit field defines the PID that the RCST shall use in optional MPEG2 TS packets on the return link for CTRL/MNGM information. This parameter can be the same for all RCSTs.

- CRA_level: The CRA assignment to the terminal, in bits/s.

- VBDC_max: The maximum number of payload units per superframe allocated to the RCST through A/VBDC. This is not necessarily an absolute value for each superframe; it can be an average over several superframes.

- RBDC_max: The maximum allowed RBDC data rate, in bits/s.

- RBDC_timeout: The RBDC timeout, in superframes. A value of "0" indicates that the timeout is disabled.

### 8.5.5.10.5    ACQ Assign descriptor

This descriptor (table 55) defines the assignment of Acquisition bursts for the optional coarse synchronization procedure at logon. If used, it will only be present following detection of a valid CSC from the terminal. Some systems may choose to assign acquisition timeslots via the TBTP instead of using this descriptor.

**Table 55: ACQ Assign descriptor**

| Syntax | No. of bits | | Information Mnemonic |
|---|---|---|---|
| | Reserved (see note) | Information | |
| ACQ_assign_descriptor() { | | | |
| descriptor_tag | | 8 | uimsbf |
| descriptor_length | | 8 | uimsbf |
| ACQ_achieved_time_threshold | | 8 | uimsbf |
| ACQ_achieved_frequency_threshold | | 16 | uimsbf |
| ACQ_repeats | 2 | 6 | uimsbf |
| ACQ_start_superframe | | 16 | uimsbf |
| ACQ_frame_number | 3 | 5 | uimsbf |
| ACQ_repeat_period | 2 | 6 | uimsbf |
| ACQ_slot_number | 5 | 11 | uimsbf |
| } | | | |
| NOTE:    Reserved bits are of type bslbf, and shall precede the Information bits on the same line. They shall be ignored by the RCST. For an encrypted uni-cast TIM, the bit values shall be varied in a random manner to avoid encryption spoofing. | | | |

Semantics for the ACQ_assign_descriptor:

- descriptor_tag: The descriptor tag is an 8 bit field which identifies each descriptor. Its value is given in the Tag value column of table 50.

- descriptor_length: The descriptor length is an 8 bit field specifying the number of bytes of the descriptor immediately following the descriptor_length field.

- ACQ_achieved_time_threshold: This 8 bit field gives the magnitude of the burst timing error threshold value to be used for the transition from the coarse synchronization procedure to the "ready for fine sync" state, see clause 7.1. The value is scaled by the same amount as the Burst_time_correction value of the Correction Message Table and Correction Message descriptor, allowing a direct magnitude comparison with that value. Transition to the "ready for fine sync" state occurs when the magnitude of the measured Burst_time_correction value is less than, or equal to, the ACQ_achieved_time_threshold.

- ACQ_achieved_frequency_threshold: This 16 bit field gives the magnitude of the carrier frequency error threshold value to be used for the transition from the coarse synchronization procedure to "ready for fine sync" state, see clause 7.1. Transition to the "ready for fine sync" state occurs when the magnitude of the measured Frequency_correction value is less than, or equal to, the ACQ_achieved_frequency_threshold.

NOTE:    If both ACQ_achieved_time_threshold and ACQ_achieved_frequency_threshold have non-zero values, transition to the "ready for fine sync" state occurs when both criterion are fulfilled.

- ACQ_repeats: This 6 bit field defines the maximum number of times that the ACQ shall be repeated during the Coarse Synchronization procedure. A special value of "00 0000" is reserved to indicate that the terminal is being given a sustained acquisition burst. This special value is reserved for use during terminal alignment procedures, typically at the time of installation.

- ACQ_start_superframe: This 16 bit field gives the modulo 65 536 superframe number at which the ACQ assignment starts (see SCT Superframe_counter parameter in clause 8.5.5.2).

- ACQ_frame_number: This 5 bit field identifies which frame number in the superframe contains the ACQ burst. This number matches the frame numbering defined in clause 6.7.2.

- ACQ_repeat_period: This 6 bit field gives the number of superframes between ACQ repeats.

- ACQ_slot_number: This 11 bit field identifies the timeslot number to use for ACQ. This number matches the numbering defined in clause 6.7.2.

#### 8.5.5.10.6        SYNC Assign descriptor

This descriptor (table 56) provides the static BTP assignment of SYNC bursts. It is typically sent following the detection of a valid CSC from the terminal, but may also be sent when it is necessary to modify the SYNC assignment.

**Table 56: SYNC Assign descriptor**

| Syntax | No. of bits | | Information Mnemonic |
|---|---|---|---|
| | Reserved (see note) | Information | |
| SYNC_assign_descriptor() { | | | |
| descriptor_tag | | 8 | uimsbf |
| descriptor_length | | 8 | uimsbf |
| SYNC_achieved_time_threshold | | 8 | uimsbf |
| max_SYNC_tries | | 8 | uimsbf |
| SYNC_achieved_frequency_threshold | | 16 | uimsbf |
| SYNC_start_superframe | | 16 | uimsbf |
| SYNC_frame_number | 3 | 5 | uimsbf |
| SYNC_repeat_period | | 16 | uimsbf |
| SYNC_slot_number | 5 | 11 | uimsbf |
| } | | | |
| NOTE:    Reserved bits are of type bslbf, and shall precede the Information bits on the same line. They shall be ignored by the RCST. For an encrypted uni-cast TIM, the bit values shall be varied in a random manner to avoid encryption spoofing. | | | |

Semantics for the SYNC_assign_descriptor:

- descriptor_tag: The descriptor tag is an 8 bit field which identifies each descriptor. Its value is given in the Tag value column of table 50.

- descriptor_length: The descriptor length is an 8 bit field specifying the number of bytes of the descriptor immediately following the descriptor_length field.

- SYNC_achieved_time_threshold: This 8 bit field gives the magnitude of the burst timing error threshold value to be used for the transition from the fine synchronization procedure to the "fine sync" state, see clause 7.1. The value is scaled by the same amount as the Burst_time_correction value of the Correction Message Table and Correction Message descriptor, allowing a direct magnitude comparison with that value. Transition to the "fine sync" state occurs when the magnitude of the measured Burst_time_correction value is less than, or equal to, the SYNC_achieved_time_threshold.

- max_SYNC_tries: This 8 bit field gives the maximum number of bursts that the RCST can send to achieve fine synchronization, from the start of the "ready for fine sync" state (see clause 7.1).

- SYNC_achieved_frequency_threshold: This 16 bit field gives the magnitude of the carrier frequency error threshold value to be used for the transition from the fine synchronization procedure to the "fine sync" state, see clause 7.1. Transition to the "fine sync" state occurs when the magnitude of the measured Frequency_correction value is less than, or equal to, the SYNC_achieved_frequency_threshold.

  NOTE:    If both SYNC_achieved_time_threshold and SYNC_achieved_frequency_threshold have non-zero values, transition to the "fine sync" state occurs when both criterion are fulfilled.

- SYNC_start_superframe: This 16 bit field gives the modulo 65 536 superframe number at which the SYNC assignment starts (see SCT Superframe_counter parameter in clause 8.5.5.2).

- SYNC_frame_number: This 5 bit field identifies which frame number in the superframe contains the SYNC burst. This number matches the frame numbering defined in clause 6.7.2.

- SYNC_repeat_period: This 16 bit field gives the number of superframes between SYNC repeats, for example, SYNC_repeat_period=0 means that the SYNC slot is assigned on each superframe, SYNC_repeat_period=1 means that two superframes containing the SYNC slot assignment are separated by 1 superframe that does not have the SYNC slot assigned, and so on.

- SYNC_slot_number: This 11 bit field identifies the timeslot number to use for SYNC. This number matches the numbering defined in clause 6.7.2.

### 8.5.5.10.7 Encrypted Logon ID descriptor

This descriptor (see table 57) is provided as one possible method of TBTP key management, when the TBTP is encrypted. It requires that the TIM message itself be encrypted with an RCST specific key.

**Table 57: Encrypted Logon ID descriptor**

| Syntax | No. of bits | | Information Mnemonic |
| --- | --- | --- | --- |
|  | Reserved (see note) | Information |  |
| Encrypted_Logon_ID_descriptor() { |  |  |  |
|     descriptor_tag |  | 8 | uimsbf |
|     descriptor_length |  | 8 | uimsbf |
|     ID_start_time |  | 16 | uimsbf |
|     ID_update_period |  | 16 | uimsbf |
|     ID_loop_count |  | 8 | uimsbf |
|     for (i = 0; i<= ID_loop_count; i++) { |  |  |  |
|     Encrypted_Logon_ID } |  | 16 | uimsbf |
| } |  |  |  |
| NOTE: Reserved bits are of type bslbf, and shall precede the Information bits on the same line. They shall be ignored by the RCST. The value of these bits is undefined and shall be varied in a random manner to avoid encryption spoofing. | | | |

Semantics for the Encrypted_Logon_ID_descriptor:

- descriptor_tag: The descriptor tag is an 8 bit field which identifies each descriptor. Its value is given in the Tag value column of table 50.

- descriptor_length: The descriptor length is an 8 bit field specifying the number of bytes of the descriptor immediately following the descriptor_length field.

- ID_Start_time: This 16 bit field gives the superframe count value at which the first Encrypted_Logon_ID parameter in the following loop takes effect. This corresponds to the superframe counter defined in the SCT and TBTP.

- ID_update_period: This 16 bit field gives the period, in terms of superframe counts, at which successive Encrypted_Logon_ID parameters take effect following the first such parameter.

- ID_loop_count: This 8 bit field defines one less than the number of encrypted logon IDs that follow. A zero count indicates one loop. The IDs follow in sequential time order, earliest first.

- Encrypted_Logon_ID: This 16 bit value gives the value for a future encrypted logon ID.

### 8.5.5.10.8 Echo Value descriptor

This descriptor (table 58) supports a simple loop-back RCST diagnostic test.

**Table 58: Echo Value descriptor**

| Syntax | No. of bits | | Information Mnemonic |
| --- | --- | --- | --- |
|  | Reserved (see note) | Information |  |
| Echo_value_descriptor() { |  |  |  |
|     descriptor_tag |  | 8 | uimsbf |
|     descriptor_length |  | 8 | uimsbf |
|     Echo_Value |  | 16 | "1" + bslbf |
| } |  |  |  |
| NOTE: Reserved bits are of type bslbf, and shall precede the Information bits on the same line. They shall be ignored by the RCST. For an encrypted uni-cast TIM, the bit values shall be varied in a random manner to avoid encryption spoofing. | | | |

Semantics for the Echo_value_descriptor:

- descriptor_tag: The descriptor tag is an 8 bit field which identifies each descriptor. Its value is given in the Tag value column of table 50.

- descriptor_length: The descriptor length is an 8 bit field specifying the number of bytes of the descriptor immediately following the descriptor_length field.

- Echo_value: This 16 bit field defines the value to be echoed back (for example in the 2 M&C bytes of the return link SAC field). The MSB shall be set to "1".

### 8.5.5.10.9        Linkage descriptor (private data)

This is an extension to the standard DVB Linkage descriptor (see EN 300 468 [3], clause 6.2.11), occupying the private data bytes provision at the end of that descriptor. It allows the RCST to identify which interactive network services its population group. The Linkage descriptor shall be as EN 300 468 [3], table 47, with the changes as highlighted in bold font in table 59.

**Table 59: Linkage descriptor - private data**

| Syntax | No. of bits | | Information Mnemonic |
|---|---|---|---|
| | Reserved (see note) | Information | |
| linkage_descriptor(){ | | | |
| descriptor_tag | | 8 | uimsbf |
| descriptor_length | | 8 | uimsbf |
| transport_stream_id | | 16 | uimsbf |
| original_network_id | | 16 | uimsbf |
| service_id | | 16 | uimsbf |
| linkage_type | | 8 | uimsbf |
| Interactive_Network_ID | | 16 | uimsbf |
| Population_ID_loop_count | | 8 | uimsbf |
| for (i=0; i<=Population_ID_loop_count; i++) { | | | |
| population_ID_base | | 16 | uimsbf |
| population_ID_mask | | 16 | uimsbf |
| } | | | |
| for (i=0;i<N;i++) { | | | |
| private_data_byte | | 8 | bslbf |
| } | | | |
| } | | | |
| NOTE:    Reserved bits are of type bslbf, and shall precede the Information bits on the same line. They shall be ignored by the RCST. For an encrypted uni-cast TIM, the bit values shall be varied in a random manner to avoid encryption spoofing. | | | |

Semantics for the private data part of the linkage_descriptor:

- descriptor_tag: The descriptor tag is an 8 bit field which identifies each descriptor. Its value is defined in EN 300 468 [3] and is recalled in table 50.

- descriptor_length: The descriptor length is an 8 bit field specifying the number of bytes of the descriptor immediately following the descriptor_length field.

- transport_stream_id: This is a 16-bit field which identifies the TS containing the information service indicated.

- original_network_id: This 16-bit field gives the label identifying the network_id of the originating delivery system of the information service indicated.

- service_id: This is a 16-bit field which uniquely identifies an information service within a TS. The service_id is the same as the program_number in the corresponding program_map section.

- linkage_type: This is an 8-bit field specifying the type of linkage. Its value is 0x81 for "RCS FLS".

- Interactive_Network_ID: This 16 bit field gives the label identifying the network_ID for the interactive network that services the population_ID's following.

- Population_ID_loop_count: This 8 bit field indicates one less than the number of population_ID ranges in the following list.

- population_ID_base and population_ID_mask: These two 16 bit values, in combination, define a range of population_ID's associated with this linkage descriptor/interactive network. The population_ID is the identifier for the population group that the RCST belongs to. This is a value assigned by the network operator, and is a configuration parameter known to the RCST prior to forward link acquisition. The population_ID_base parameter defines the fixed bit pattern part of the population_ID range, while the population_ID_mask parameter defines those bit positions of the population ID that are "not significant". A "1" value in a bit of the mask indicates that the corresponding bit of the RCST population_ID may be "1" or "0". A "0" value in a bit of the mask indicates that the corresponding bit of the RCST population_ID shall match the value of that bit in the base parameter.

- private_data_byte: This is an 8 bit field, the value of which is privately defined. It retains the functionality of the original linkage_descriptor for further extensions.

### 8.5.5.10.10      RCS content descriptor

The RCS content descriptor provides the definition of the PID assignments to the RCS specific tables, and is shown in table 60. This descriptor is used in the second loop of the Program Map Table (PMT), defined in ISO/IEC 13818-1 [7], clause 2.4.4.8/table 2-28. Each descriptor defines the General RCS SI Tables and RCS Specific Messages associated with one elementary_PID.

The use of the RCS content descriptor for the RMT is optional.

NOTE:      The PCR Insertion packet PID is signalled earlier in the PMT section as the PCR_PID parameter.

**Table 60: RCS content descriptor**

| Syntax | No. of bits | | Information Mnemonic |
|---|---|---|---|
| | Reserved (see note) | Information | |
| RCS_content_descriptor() { | | | |
|    descriptor_tag | | 8 | uimsbf |
|    descriptor_length | | 8 | uimsbf |
|    for (i=0; i<N; i++) { | | | |
|       Table_id | | 8 | uimsbf |
|    } | | | |
| } | | | |
| NOTE:      Reserved bits are of type bslbf, and shall precede the Information bits on the same line. They shall be ignored by the RCST. For an encrypted uni-cast TIM, the bit values shall be varied in a random manner to avoid encryption spoofing. | | | |

Semantics for the RCS_content_descriptor:

- descriptor_tag: The descriptor tag is an 8 bit field which identifies each descriptor. Its value is given in the Tag value column of table 50.

- descriptor_length: The descriptor length is an 8 bit field specifying the number of bytes of the descriptor immediately following the descriptor_length field.

- table_id: This 8 bit field gives the table_id value for a General RCS SI Table or RCS Specific Message, as defined in table 28.

8.5.5.10.11      Satellite forward link descriptor

The satellite forward link descriptor defines the forward link, and is used in place of the Satellite delivery system descriptor of EN 300 468 [3] for RCS systems. It is shown in table 61.

**Table 61: Satellite forward link descriptor**

| Syntax | No. of bits | | Information Mnemonic |
|---|---|---|---|
| | Reserved (see note) | Information | |
| Satellite_forward_link_descriptor() { | | | |
| descriptor_tag | | 8 | uimsbf |
| descriptor_length | | 8 | uimsbf |
| satellite_ID | | 8 | uimsbf |
| beam_ID | | 16 | uimsbf |
| NCC_ID | | 8 | uimsbf |
| link_usage | | 3 | bslbf |
| local_link_ID | | 5 | uimsbf |
| frequency | | 32 | uimsbf |
| orbital_position | | 16 | bslbf |
| west_east_flag | | 1 | bslbf |
| Polarization | | 2 | bslbf |
| transmission_standard | | 2 | uimsbf |
| if (transmission_standard == 0) { | | | |
| "001" | | 3 | bslbf |
| } | | | |
| else if ((transmission_standard == 1) or (transmission_standard == 2)) { | | | |
| scrambling sequence selector | | 1 | bslbf |
| roll_off | | 2 | uimsbf |
| } | | | |
| symbol_rate | | 24 | uimsbf |
| if (transmission_standard == 0){ | | | |
| FEC_inner | | 4 | bslbf |
| Reserved | 4 | | bslbf |
| } | | | |
| else if ((transmission_standard == 1) or (transmission_standard == 2)) { | | | |
| Input_Stream_Identifier | | 8 | uimsbf |
| if (scrambling_sequence_selector == 0) | | | |
| spreading_code_selector | | 3 | |
| scrambling_sequence_index | 3 | 18 | uimsbf |
| } | | | |
| for (i=0; i<N; i++) { | | | |
| private_data_byte | | 8 | bslbf |
| } | | | |
| } | | | |
| NOTE:    Reserved bits are of type bslbf, and shall precede the Information bits on the same line. They shall be ignored by the RCST. For an encrypted uni-cast TIM, the bit values shall be varied in a random manner to avoid encryption spoofing. | | | |

Semantics for the Satellite_forward_link_descriptor:

- descriptor_tag: The descriptor tag is an 8 bit field which identifies each descriptor. Its value is given in the Tag value column of table 50.

- descriptor_length: The descriptor length is an 8 bit field specifying the number of bytes of the descriptor immediately following the descriptor_length field.

- satellite_ID: This 8 bit field identifies which satellite is carrying the forward link, and corresponds to the satellite_ID field in the SPT/PCR Insertion packet payload.

- beam_ID: This 16 bit field identifies the beam number of the satellite carrying the forward link.

- NCC_ID: This 8 bit field identifies which NCC is transmitting the forward link, and corresponds to the NCC_ID field in the PCR Insertion packet payload (see clause 8.5.5.6).

- link_usage: This 3 bit field allows link discrimination for RCSTs that can operate on multiple forward links simultaneously. The usage codes are shown in table 62.

**Table 62: Forward link usage codes**

| Usage code | Value |
|---|---|
| 000 | Combined signalling/data link |
| 001 | Signalling link only |
| 010 | Data link only |
| 111 | Release data link |
| 011 to 110 | Reserved for future use |

The first two codes are mutually exclusive in any Transport Stream ID (TS_id) entry of the RMT (there can be only one signalling link for a given population_ID). The signalling link only value can only be used where all RCSTs covered by that TS_id are capable of receiving multiple forward links simultaneously.

The release data link value allows a previously assigned data link to be released, without logging off the terminal.

For RCSTs that can operate on multiple forward links simultaneously, there are two possible modes of defining links during logon:

- the RMT defines only the signalling link, and the TIM defines any data only links. This is the most flexible method.

- the RMT uses multiple descriptors, one for each forward link defined. Only one of these may be a signalling link.

NOTE 1:  In DVB terms, this mode "stretches" the definition of a transport stream to actually cover a group of inter-related transport streams.

For both modes, the TIM can command changes to the initial configuration defined via the RMT.

NOTE 2:  Changing a signalling link may cause service interruption and is not recommended.

RCSTs capable only of single carrier operation shall use only the combined signalling/data link descriptor and shall ignore descriptors for other link_usage codes:

- local_link_ID: This 5 bit field is used to simplify changes to the definition of a link for RCSTs that can operate on multiple forward links simultaneously, and is a RCS local value defined by the interactive network operator. It allows the NCC to indicate which of the forward links is being created, changed or released. The last two operations can only be performed via a TIM message.

- frequency: This 32-bit field gives the frequency value. The frequency is given in multiples of 100 Hz.

- orbital_position: The orbital_position is a 16 bit field giving the 4-bit BCD values specifying 4 characters of the orbital position in degrees where the decimal point is after the third character (e.g. 019,2°).

- west_east_flag: The west_east_flag is a 1 bit field indicating if the satellite position is in the western or eastern part of the orbit. A value "0" indicates the western position and a value "1" indicates the eastern position.

- polarization: The polarization is a 2 bit field specifying the polarization of the transmitted signal (see table 63);

**Table 63: Forward link polarization**

| Polarization | Value |
|---|---|
| linear - horizontal | 00 |
| linear - vertical | 01 |
| circular - left | 10 |
| circular - right | 11 |

- transmission_standard: 0 for DVB-S, 1 for DVB-S2 using CCM, 2 for DVB-S2 using ACM and 3 reserved.

- scrambling sequence selector: Value 1 means default DVB-S2 physical layer scrambling sequence of index 0, value 0 means that the scrambling sequence to be used is specified using the scrambling_sequence_index field. The value 0 must be used if spectrum spreading with a spreading factor greater than 1 is employed.

- roll_off: 0 for not defined, 1 for 20 %, 2 for 25 %, 3 for 35 %.

- symbol_rate: The symbol_rate is a 24 bit field giving the symbol rate in multiples of 100 symbols/s.

- FEC_inner: The FEC_inner is a 4 bit field specifying the inner FEC scheme used, as per table 64.

**Table 64: Inner FEC scheme**

| Code Rate | Value |
|---|---|
| 1/2 | 0000 |
| 2/3 | 0001 |
| 3/4 | 0010 |
| 5/6 | 0011 |
| 7/8 | 0100 |
| Inner code is omitted | 1111 |
| Reserved for future use | 0101 to 1110 |

- Input_Stream_Identifier: As defined in clause 5.1.6 of [16].

- spreading_code_selector: This 3-bit field defines the chip sequence used to achieve spectrum spreading, in accordance with table 65.

**Table 65: Forward link spreading sequences**

| Value | Spreading factor | Chip sequence |
|---|---|---|
| 000 | 2 | 1, 1 |
| 001 | 2 | 1, -1 |
| 010 | 3 | 1, 1, 1 |
| 011 | 4 | 1, 1, 1, 1 |
| 100 | 4 | 1, 1, -1, -1 |
| 101 | 4 | 1, -1, 1, -1 |
| 110 | 4 | 1, -1, -1, 1 |
| 111 | 1 (no spreading) | 1 |

- scrambling_sequence_index: DVB-S2 physical layer scrambling sequence index as defined in clause 5.5.4 of [16].

- private_data_byte: This is an 8 bit field, the value of which is privately defined. It can be used, for example, to indicate system specific NCC information.

8.5.5.10.12        Satellite return link descriptor

**Table 66: Satellite return link descriptor**

| Syntax | No. of bits | | Information Mnemonic |
|---|---|---|---|
| | Reserved (see note) | Information | |
| Satellite_return_link_descriptor() { | | | |
|    descriptor_tag | | 8 | uimsbf |
|    descriptor_length | | 8 | uimsbf |
|    satellite_ID | | 8 | uimsbf |
|    beam_ID | | 16 | uimsbf |
|    gateway_ID | | 8 | uimsbf |
|    Reserved | 8 | | bslbf |
|    orbital_position | | 16 | bslbf |
|    west_east_flag | 7 | 1 | bslbf |
|    Superframe_ID | | 8 | uimsbf |
|    Tx_frequency_offset | | 24 | tcimsbf |
|    for (i=0; i<N; i++) { | | | |
|       private_data_byte | | 8 | bslbf |
|    } | | | |
| } | | | |
| NOTE:    Reserved bits are of type bslbf, and shall precede the Information bits on the same line. They shall be ignored by the RCST. For an encrypted uni-cast TIM, the bit values shall be varied in a random manner to avoid encryption spoofing. | | | |

The satellite return link descriptor defines the characteristics of the return link and is shown in table 66.

Semantics for the Satellite_return_link_descriptor:

- descriptor_tag: The descriptor tag is an 8 bit field which identifies each descriptor. Its value is given in the Tag value column of table 50.

- descriptor_length: The descriptor length is an 8 bit field specifying the number of bytes of the descriptor immediately following the descriptor_length field.

- satellite_ID: This 8 bit field identifies which satellite is carrying the return link, and corresponds to the satellite_ID field in the SPT/PCR Insertion packet payload.

- beam_ID: This 16 bit field identifies the beam number of the satellite carrying the return link.

- gateway_ID: This 8 bit field identifies which Gateway is receiving the return link, and corresponds to the gateway_ID field in the PCR Insertion packet payload.

- orbital_position: The orbital_position is a 16 bit field giving the 4-bit BCD values specifying 4 characters of the orbital position in degrees where the decimal point is after the third character (e.g. 019,2°).

- west_east_flag: The west_east_flag is a 1 bit field indicating if the satellite position is in the western or eastern part of the orbit. A value "0" indicates the western position and a value "1" indicates the eastern position.

- superframe_ID: This 8 bit field identifies which superframe the terminal is to use. This identifier matches an entry in the SCT.

- Tx_frequency_offset: This 24-bit field gives the signed offset of the RCST transmit centre frequency relative to the Superframe_centre_frequency parameter (SCT). The frequency is given in multiples of 100 Hz. This parameter shall be ignored by the RCSTs when it is used in broadcast messages.

- private_data_byte: This is an 8 bit field, the value of which is privately defined. It can be used, for example, to indicate system specific NCC information.

### 8.5.5.10.13        Table Update descriptor

The Table Update descriptor (see table 67) provides a mechanism for notifying RCSTs when there is an upcoming change to one of the RCS Tables. In normal operation, these tables are expected to change infrequently. This descriptor provides a method for notifying the RCSTs on a particular interactive network when an update to one or more of these tables is imminent. This allows the RCSTs to only scan for table changes at this time, so freeing up processing resources. An RCST that monitors the composition tables continually may ignore this descriptor. Use of this descriptor does not modify the normal update control method implemented by the version_number and current_next_indicator fields of the section header. In addition, the Table Update descriptor may be utilized to indicate an update to any General SI Table defined in the present document or DVB-SI table, such as NIT, BAT, SDT, EIT, etc.

> NOTE:    In the case of the SCT, the Superframe start time (parameters Superframe_start_time_base and Superframe_start_time_ext) for each superframe definition, changes in a systematic fashion to reflect the regular period of the superframe. No update will be signalled where this is the only change and where the change has no material effect on superframe timing (i.e. the new start time equals a previous start time plus an integer number of superframe periods for the superframe in question).

**Table 67: Table Update descriptor**

| Syntax | No. of bits | | Information Mnemonic |
|---|---|---|---|
| | Reserved (see note) | Information | |
| table_update_descriptor() { | | | |
| descriptor_tag | | 8 | uimsbf |
| descriptor_length | | 8 | uimsbf |
| interactive_network_id | | 16 | uimsbf |
| for (i=0; i<n; i++) { | | | |
| table_id | | 8 | uimsbf |
| new_version | 3 | 5 | uimsbf |
| } | | | |
| } | | | |
| NOTE:       Reserved bits are of type bslbf, and shall precede the Information bits on the same line. They shall be ignored by the RCST. For an encrypted uni-cast TIM, the bit values shall be varied in a random manner to avoid encryption spoofing. | | | |

Semantics for the table_update_descriptor:

- descriptor_tag: The descriptor tag is an 8 bit field which identifies each descriptor. Its value is given in the Tag value column of table 50.

- descriptor_length: The descriptor length is an 8 bit field specifying the number of bytes of the descriptor immediately following the descriptor_length field.

- interactive_network_id: This 16 bit field indicates which interactive network the tables belong to, and matches the field of the same name in the table section header.

- table_id: This 8 bit field indicates a change to the associated table. The table_id values are defined in table 28 in the present document for the General SI Tables and in EN 300 468 [3] for the standard DVB-SI tables.

- new_version: This 5 bit field defines the new version number of the table after the change, and matches the version_number field in the table section header.

### 8.5.5.10.14    Contention control descriptor

**Table 68: Contention control descriptor**

| Syntax | No. of bits | | Information Mnemonic |
|---|---|---|---|
|  | Reserved (see note) | Information |  |
| Contention_Control_descriptor(){ |  |  |  |
|     descriptor_tag |  | 8 | uimsbf |
|     descriptor_length |  | 8 | uimsbf |
|     Superframe_ID |  | 8 | uimsbf |
|     CSC_response_timeout |  | 32 | upcrmsf |
|     CSC_max_losses |  | 8 | uimsbf |
|     Max_time_before_retry |  | 32 | upcrmsf |
| } |  |  |  |
| NOTE:    Reserved bits are of type bslbf, and shall precede the Information bits on the same line. For encrypted message types, the value of these bits is undefined. | | | |

The Contention control descriptor (see table 68) defines the necessary retransmission parameters for the CSC bursts.

Semantics for the contention_control_descriptor:

- descriptor_tag: The descriptor tag is an 8 bit field which identifies each descriptor. Its value is given in the Tag value column of table 50.

- descriptor_length: The descriptor length is an 8 bit field specifying the number of bytes of the descriptor immediately following the descriptor_length field.

- Superframe_ID: This is an 8-bit field which serves as a label for identification of the relevant superframe to which this descriptor applies.

- CSC_response_timeout: This 32-bit field gives the value of the timeout after which the RCST shall consider that there was a collision on its previous CSC burst. It is expressed in terms of PCR count intervals. The 32 bits correspond to a maximum duration of 93,2 s.

- CSC_max_losses: This 8-bit field specifies the maximum number of unsuccessful CSC attempts before the RCST gives up its logon procedure.

- Max_time_before_retry: This 32-bit field gives the upper bound before retransmission of a CSC burst, expressed in terms of PCR count intervals, after the CSC_response_timeout has expired. In other words, upon expiry of the CSC_response_timeout and if the CSC_max_losses has not been reached, the RCST shall wait a random time between 0secondand this upper bound before transmitting a new CSC burst. The 32 bits correspond to a maximum duration of 93,2 s.

### 8.5.5.10.15    Correction Control descriptor

The Correction control descriptor (see table 69) defines the necessary timeouts for the coarse synchronization, fine synchronization and synchronization maintenance procedures.

**Table 69: Correction control descriptor**

| Syntax | No. of bits | | Information Mnemonic |
|---|---|---|---|
|  | Reserved (see note) | Information |  |
| Correction_Control_descriptor(){ |  |  |  |
|     descriptor_tag |  | 8 | uimsbf |
|     descriptor_length |  | 8 | uimsbf |
|     ACQ_response_timeout |  | 32 | upcrmsf |
|     SYNC_response_timeout |  | 32 | upcrmsf |
|     ACQ_max_losses |  | 8 | uimsbf |
|     SYNC_max_losses |  | 8 | uimsbf |
| } |  |  |  |
| NOTE:    Reserved bits are of type bslbf, and shall precede the Information bits on the same line. | | | |

Semantics for the correction_control_descriptor:

- descriptor_tag: The descriptor tag is an 8 bit field which identifies each descriptor. Its value is given in the Tag value column of table 50.

- descriptor_length: The descriptor length is an 8 bit field specifying the number of bytes of the descriptor immediately following the descriptor_length field.

- ACQ_response_timeout: This 32-bit field gives the value of the timeout after which the RCST shall assume there was a problem on its previous ACQ burst. It corresponds to the "Correction received in time" test of the coarse sync procedure. It is expressed in terms of PCR count intervals. The 32 bits correspond to a maximum duration of 93,2 s.

- SYNC_response_timeout: This 32-bit field gives the value of the timeout after which the RCST shall assume there was a problem on its previous SYNC burst. It corresponds to the "Correction received in time" test of the fine sync and sync maintenance procedures. It is expressed in terms of PCR count intervals. The 32 bits correspond to a maximum duration of 93,2 s.

- ACQ_max_losses: This 8-bit field specifies the maximum number of consecutive ACQs with no correction received before expiry of ACQ_response_timeout. When this number is reached, the RCST shall return to the "Off/Stand-by" state. It corresponds to the "max losses exceeded" test of the coarse sync procedure (see clause 7.4).

- SYNC_max_losses: This 8-bit field specifies the maximum number of consecutive SYNCs with no correction received before expiry of SYNC_response_timeout. When this number is reached, the RCST shall return to the "Off/Stand-by" state. It corresponds to the "max losses exceeded" test of the fine sync procedure and of the sync maintenance procedure (see clauses 7.5 and 7.6).

### 8.5.5.10.16    Forward Interaction Path descriptor

The DVB standards for interactive services require a Forward Interaction Path and a Return Interaction. The Forward Interaction Path descriptor provides parameters that an RCST needs for finding on the forward link its traffic of the Forward Interaction Path. This descriptor is delivered to and utilized by an RCST when the RCST does not make use of the optional Connection Control descriptor.

The Forward Interaction Path descriptor is carried in a TIM. It is valid until it is overwritten by a new version. ("Wake-able" RCSTs in specific network architectures may continue to receive traffic on the assigned Forward Interaction Path after log off). Unicast TIMs allow distributing the RCSTs over the available forward link capacity. The NCC can assign one or more transport streams (i.e. transponders) to an RCST, depending on the number of receivers that the RCST contains. Therefore, the descriptor contains a loop over transport streams. On each transport stream the traffic can be carried in one or more PIDs, which are listed in a second loop.

**Table 70: Forward Interaction Path descriptor**

| Syntax | No. of bits | | Information Mnemonic |
|---|---|---|---|
| | Reserved (see note) | Information | |
| Forward_interaction_path_descriptor() { | | | |
| descriptor_tag | | 8 | Uimsbf |
| descriptor_length | | 8 | Uimsbf |
| for (i=0; I<n; I++} | | | |
| original_network_id | | 16 | Uimsbf |
| transport_stream_id | | 16 | Uimsbf |
| PID_loop_count | 4 | 4 | Uimsbf |
| For (k=0; k<=PID_loop_count; k++) { | | | |
| PID | 3 | 13 | Uimsbf |
| } | | | |
| } | | | |
| } | | | |
| NOTE:    Reserved bits are of type bslbf, and shall precede the Information bits on the same line. They shall be ignored by the RCST. For an encrypted unicast TIM, the bit values shall be varied in a random manner to avoid encryption spoofing. | | | |

Semantics for the Forward Interaction Path descriptor:

- descriptor_tag: The descriptor tag is an 8-bit field that identifies each descriptor. Its value is given in the Tag value column of table 50.

- descriptor_length: The descriptor length is an 8-bit field specifying the number of bytes of the descriptor immediately following the descriptor_length field.

- original_network_id: This 16-bit field gives the label identifying the network_id of the originating delivery system containing the Forward Interaction Path.

- transport_stream_id: This is a 16-bit field which identifies the transport stream containing the Forward Interaction Path.

- PID_loop_count: This field gives one less than the number of PIDs that follow.

- PID: This 13-bit field gives a PID that carries traffic of the Forward Interaction Path.

### 8.5.5.10.17    Return Interaction Path descriptor

The DVB standards for interactive services require a Forward Interaction Path and a Return Interaction Path. The Return Interaction Path descriptor provides parameters that an RCST needs for encapsulating its return traffic when the initial PID allocated at logon (see Logon Initialize descriptor and Mesh Logon Initialize Descriptor) is not sufficient. This descriptor is delivered to and utilized by an RCST when the RCST does not make use of the optional Connection Control descriptor.

The Return Interaction Path descriptor is carried in a TIM. It is valid until it is overwritten by a new version.

**Table 71: Return Interaction Path descriptor**

| Syntax | No. of bits | | Information Mnemonic |
|---|---|---|---|
| | Reserved (see note) | Information | |
| Return_interaction_path_descriptor() { | | | |
| descriptor_tag | | 8 | uimsbf |
| descriptor_length | | 8 | uimsbf |
| Continuous_carrier | 3 | 1 | bslbf |
| Network_Routing_Label_loop_Count | | 4 | uimsbf |
| For (i=0; i<= Network_routing_Label_loop_Count; i++) { | | | |
| Allocation_Desallocation_flag | 6 | 1 | blsbf |
| PID_flag | | 1 | blsbf |
| If (PID_flag  == 1) { | | | |
| PID_loop_count | | 8 | uimsbf |
| For (j=0; j<= PID_loop_Count; j++) { | | | blsbf |
| PID | 3 | 13 | uimsbf |
| } | | | |
| } | | | |
| VPI/VCI_flag | 7 | 1 | blsbf |
| If (VPI/VCI_flag  == 1) { | | | |
| VPI/VCI_loop_count | | 8 | uimsbf |
| For (k=0; k<= VPI/VCI_loop_Count; k++) { | | | |
| VPI | | 8 | uimsbf |
| VCI | | 16 | uimsbf |
| } | | | |
| } | | | |
| Route_ID_flag | 7 | 1 | blsbf |
| If (Route_ID_flag  == 1) { | | | |
| Route_ID_loop_count | | 8 | uimsbf |
| For (l=0; l<= Route_ID_loop_Count; l++) { | | | |
| Route_ID | | 16 | uimsbf |
| } | | | |
| } | | | |
| Channel_ID | 4 | 4 | uimsbf |
| } | | | |
| } | | | |
| NOTE:  Reserved bits are of type bslbf, and shall precede the Information bits on the same line. They shall be ignored by the RCST. For an encrypted unicast TIM, the bit values shall be varied in a random manner to avoid encryption spoofing. | | | |

Semantics for the Return Interaction Path descriptor:

- descriptor_tag: The descriptor tag is an 8-bit field that identifies each descriptor. Its value is given in the Tag value column of table 50.

- descriptor_length: The descriptor length is an 8-bit field specifying the number of bytes of the descriptor immediately following the descriptor_length field.

- Continuous_carrier: This flag indicates whether the descriptor applies to the MF-TDMA air interface (value "1") or the continuous carrier air interface.(Value "0").

- Network_Routing_Label_Loop_count: This field indicating one less than the number of iteration of the frame loop that follows. A zero count indicates one loop.

- Allocation_Desallocation_flag: This one-bit field indicates whether this is a Network_Routing_labels assignment to the RCST (the flag should then be set to 1) or a de-assignment (the flag should then be set to 0).

- PID_flag: The value "1" indicates that the descriptor contains PIDs. The value "0" indicates that the descriptor does not contains PIDs.

- PID_Loop_count: This field indicating one less than the number of PID that follows. A zero count indicates one PID.

- PID: This 13-bit field gives a PID that the RCST shall use in optional MPEG2 TS traffic packets on Return Interaction Path.

- VPI/VCI_flag: The value "1" indicates that the descriptor contains VPI/VCIs. The value "0" indicates that the descriptor does not contains VPI/VCIs.

- VPI/VCI_Loop_count: This field indicating one less than the number of VPI/VCI that follows. A zero count indicates one VPI/VCI pair.

- VPI: This 8-bit field gives a VPI that the RCST shall use in ATM traffic packets on Return Interaction Path.

- VCI: This 16-bit field gives a VCI that the RCST shall use in ATM traffic packets on Return Interaction Path.

- Route_ID_flag: The value "1" indicates that the descriptor contains route_IDs. The value "0" indicates that the descriptor does not contains route_IDs.

- Route_ID__Loop_count: This field indicating one less than the number of route_ID that follows. A zero count indicates one route_ID.

- Route_ID: This 16-bit field gives a route_ID that the RCST shall use in prefixed ATM traffic packets on Return Interaction Path.

- Channel_ID: This 4-bit field indicates the channel with which the Network_Routing_labels value shall be associated in the RCST. Values are managed by the NCC. This value is used to provide differentiated QoS and/or for connectivity purposes.

### 8.5.5.10.18    Connection Control descriptor (optional)

The Connection Control descriptor provides a mechanism by which signalling information are passed to the Control Plane of a RCST during connection control phases. As such, the message content is passed transparently through the lower layers covered by the present document and is not defined here. The descriptor is defined in table 72.

**Table 72: Connection Control descriptor**

| Syntax | No. of bits | Mnemonic |
|---|---|---|
| Connection_Control_descriptor(){ | | |
|     descriptor_tag | 8 | uimsbf |
|     descriptor_length | 8 | uimsbf |
|     Message_body | see text | |
| } | | |

Semantics for the Connection_Control_descriptor:

- descriptor_tag: The descriptor tag is an 8 bit field which identifies each descriptor. Its value is given in the Tag value column of table 50.

- descriptor_length: The descriptor length is an 8 bit field specifying the number of bytes of the descriptor immediately following the descriptor_length field.

- message_body, as defined in [23]: This variable length field shall contain a connection control signalling message for passing to the target connection control entity. The length of the message body should not exceed 255 bytes and should preferably be limited such that the section containing this descriptor fits within a single TS packet.

### 8.5.5.10.19    Mobility Control descriptor (Mobile Req)

This descriptor (table 73) supports conveying of mobility-related commands to the RCST.

**Table 73: Mobility Control descriptor**

| Syntax | No. of bits | | Information Mnemonic |
|---|---|---|---|
| | Reserved (see note) | Information | |
| Mobility_control_descriptor() { | | | |
| descriptor_tag | | 8 | uimsbf |
| descriptor_length | | 8 | uimsbf |
| Command_value | | 16 | uimsbf |
| Command_parameter | | 16 | See text |
| } | | | |
| NOTE:    Reserved bits are of type bslbf, and shall precede the Information bits on the same line. They shall be ignored by the RCST. For an encrypted uni-cast TIM, the bit values shall be varied in a random manner to avoid encryption spoofing. | | | |

Semantics for the Mobility_control_descriptor:

- descriptor_tag: The descriptor tag is an 8 bit field which identifies each descriptor. Its value is given in the Tag value column of table 50.

- descriptor_length: The descriptor length is an 8 bit field specifying the number of bytes of the descriptor immediately following the descriptor_length field.

- Command_value: This 16 bit field specifies a mobility-related command to be carried out by the RCST, as defined in table 74.

- Command_parameter: This 16 bit field allows the specification of parameters pertaining to specific values of Command_value, as defined in table 74.

**Table 74: Mobility command and parameter values**

| Action Requested | Value | Parameter |
|---|---|---|
| No command | 0x0000 | Reserved |
| Execute forward and return link handover | 0x0001 | Reserved |
| Execute forward link handover | 0x0002 | Reserved |
| Execute return link handover | 0x0003 | Reserved |
| Reserved | 0x0004 | Reserved |
| Send transmitter status report | 0x0005 | Reserved |
| Send position report | 0x0006 | Reserved |
| Maximum NCR absence time without enforcing initial synchronisation procedure | 0x0007 | Maximum allowed time; see note |
| Reserved | 0x0008-0x8FFF | |
| User defined | 0x9000-0xFFFF | |
| NOTE:    The time is indicated in seconds; the format is uimsbf. A value of 0 indicates that the RCST must unconditionally carry out the initial sync procedure before logging on following a log-off. | | |

### 8.5.5.10.20    Correction Message Extension descriptor (Mobile Req)

This descriptor (see table 75) supports conveying to the RCST of the identity of the time slot in which a CSC burst is received.

**Table 75: Correction Message Extension descriptor**

| Syntax | No. of bits | | Information Mnemonic |
|---|---|---|---|
| | Reserved (see note) | Information | |
| Correction_Message _Extension_descriptor(){ | | | |
| descriptor_tag | | 8 | uimsbf |
| descriptor_length | | 8 | uimsbf |
| Superframe-ID | | 8 | uimsbf |
| Superframe_Count | | 16 | uimsbf |
| Frame _Number | 3 | 5 | uimsbf |
| Slot_Number | 5 | 11 | uimsbf |
| } | | | |
| NOTE:     Reserved bits are of type bslbf, and shall precede the Information bits on the same line. They shall be ignored by the RCST. For an encrypted uni-cast TIM, the bit values shall be varied in a random manner to avoid encryption spoofing. | | | |

Semantics for Correction_Message _Extension descriptor:

- descriptor_tag: The descriptor tag is an 8 bit field which identifies each descriptor; Its value is given in the Tag value column of table 50.

- descriptor_length: The descriptor length is an 8 bit field specifying the number of bytes of the descriptor immediately following the descriptor_length field.

- superframe_id: This is an 8-bit field identifies the superframe in which the CSC burst was detected and to which the Correction_Message_descriptor applies.

- Superframe_count: This 16 bit field identifies the modulo 65 536 superframe count when the CSC burst was detected and to which the Correction_Message_descriptor applies.

- frame_number: This 5 bit field specifies the frame number within the superframe, in which the CSC burst was detected and to which the Correction_Message_descriptor applies.

- slot_number: This 11 bit field identifies the timeslot number where the burst was detected and to which the Correction_Message_descriptor applies. This number matches the numbering defined in clause 6.7.2.

### 8.5.5.10.21        Return Transmission Modes (RTM) descriptor (Mobile Opt)

The Return Transmission Modes descriptor (table 76) indicates the different scrambling sequences, modulation schemes, FEC code rates and formats that the RCST may use in the return link when operating in continuous transmission mode.

**Table 76: Return Transmission Mode descriptor**

| Syntax | No. of bits | | Information Mnemonic |
|---|---|---|---|
| | Reserved | Information | |
| return_transmission_modes_descriptor(){ | | | |
|    descriptor_tag | | 8 | uimsbf |
|    descriptor_length | | 8 | uimsbf |
|    superframe_id_loop_count | | 8 | uimsbf |
|    for(i=0; i<superframe_id_loop_count; i++){ | | | |
|       superframe_id | | 8 | uimsbf |
|       scrambling_sequence_index | 6 | 18 | uimsbf |
|       transmission_mode_count | | 8 | uimsbf |
|       for(j=0; j<transmission_mode_count; j++){ | | | |
|          inner_code_puncturing | 4 | 4 | bslbf |
|          modulation | 3 | 5 | bslbf |
|          timeslot_payload_type | | 8 | uimsbf |
|          EsNo_threshold_flag | | 1 | bslbf |
|          EsN0_threshold | | 7 | tcimsbf |
|          } | | | |
|       } | | | |
|    } | | | |

Semantics for the return_transmission_modes_descriptor:

- descriptor_tag: The descriptor tag is an 8 bit field which identifies each descriptor; Its value is given in the Tag value column of table 50.

- descriptor_length: The descriptor length is an 8 bit field specifying the number of bytes of the descriptor immediately following the descriptor_length field.

- superframe_id_loop_count: This is the number of superframe_id's that are assigned transmission modes by this descriptor. Each iteration describes the transmission modes assigned to one superframe_id.

- superframe_id: This identifies the superframe_id for which the transmission modes in the iteration of the superframe_id loop apply.

- scrambling_sequence_index: DVB-S2 physical layer scrambling sequence index as defined in clause 5.5.4 of [16].

- transmission_mode_count: This is the number of iterations in the loop that follows. Each iteration describes one transmission mode that is supported by the receiver of the respective superframe_id.

- inner_code_puncturing; This 4-bit field specifies an inner code puncturing that may be used as an alternative to that given in the TCT entry. The coding of the field is as defined for the field of the same name in clause 8.5.5.4. Only values applicable to continuous carrier operation are allowed.

- modulation: This 5-bit field specifies a modulation/spreading combination that may be used as an alternative to that given in the TCT entry. The coding of the field is as defined for the field of the same name in clause 8.5.5.4. Only values applicable to continuous carrier operation are allowed.

- timeslot_payload_type: This 8-bit field indicates a timeslot type that may be used as an alternative to that given in the TCT entry. The coding of the field is as defined for the field of the same name in clause 8.5.5.4. Only values applicable to continuous carrier operation are allowed.

- EsN0_threshold_flag: This 1-bit flag indicates whether the EsN0_threshold field contains a valid threshold (value "1") or not (value "0").

- EsN0_threshold: This 7-bit field specifies the lowest $E_s/N_0$ value at the demodulator that allows demodulation of the transmission mode with adequate quality, in 0,5 dB steps as a two's complement integer value. If EsNo_threshold_flag is "0", the value of this field is unspecified.

### 8.5.5.10.22    Mesh logon initialize descriptor (Optional)

This descriptor provides parameters needed for initial logon of a mesh-capable RCST (see table 77), replacing the logon_initialize_descriptor defined in clause 8.5.5.10.4 for such RCSTs. This descriptor is a superset of the Logon Initialize Descriptor, hence it allows the terminal to operate in both star and mesh modes simultaneously.

**Table 77: Mesh Logon Initialize descriptor**

| Syntax | No. of bits | | Information Mnemonic |
|---|---|---|---|
| | Reserved (see note) | Information | |
| Mesh_logon_Initialize_descriptor() { | | | |
|   descriptor_tag | | 8 | uimsbf |
|   descriptor_length | | 8 | uimsbf |
|   Group_ID | | 8 | uimsbf |
|   Logon_ID | | 16 | uimsbf |
|   Continuous_carrier | 2 | 1 | bslbf |
|   Security_handshake_required | | 1 | bslbf |
|   Prefix _flag | | 1 | bslbf |
|   Data_unit_labelling_flag | | 1 | bslbf |
|   Mini_slot_flag | | 1 | bslbf |
|   Contention_based_mini_slot_flag | | 1 | bslbf |
|   Mesh_connectivity_flag | | 1 | |
|   Capacity_type_flag | | 1 | bslbf |
|   Traffic_burst_type | | 1 | bslbf |
|     If (Traffic_burst_type == 0) { | | | |
|       Connectivity | | 1 | bslbf |
|       If (Connectivity == 0) { | | | |
|         Return_VPI | 4 | 8 | uimsbf |
|         Return_VCI | | 16 | uimsbf |
|       } | | | |
|       Else { | | | |
|         Return_signalling_VPI | 4 | 8 | uimsbf |
|         Return_ signalling_VCI | | 16 | uimsbf |
|         Forward_ signalling_VPI | 8 | 8 | uimsbf |
|         Forward_ signalling_VCI | | 16 | uimsbf |
|       } | | | |
|     } | | | |
|     Else { | | | |
|       Return_TRF_PID | | 13 | uimsbf |
|       Return_CTRL_MNGM_PID | 3 | 13 | uimsbf |
|     } | | | |
|   if (Capacity_type_flag == 0) { | | | |
|     CRA_level | | 24 | uimsbf |
|     VBDC_max | 5 | 11 | uimsbf |
|     RBDC_max | | 24 | uimsbf |
|     RBDC_timeout | | 16 | uimsbf |
|   } | | | |
|   If (Mesh_connectivity_flag == 0) { | | | |
|     CRA_mesh_level | | 24 | uimsbf |
|     VBDC_mesh_max | 5 | 11 | uimsbf |
|     RBDC_mesh_max | | 24 | uimsbf |
|     RBDC_mesh_timeout | | 16 | uimsbf |
|   } | | | |
| } | | | |
| NOTE:    Reserved bits are of type bslbf, and shall precede the Information bits on the same line. They shall be ignored by the RCST. For an encrypted uni-cast TIM, the bit values shall be varied in a random manner to avoid encryption spoofing. | | | |

Semantics for the Mesh_Logon_Initialize_descriptor:

- descriptor_tag: The descriptor tag is an 8 bit field which identifies each descriptor. Its value is given in the Tag value column of table 50.

- descriptor_length: The descriptor length is an 8 bit field specifying the number of bytes of the descriptor immediately following the descriptor_length field.

- Group_ID: This 8 bit field defines which Group ID the terminal is assigned to. This matches the Group_ID used in the TBTP and CMT sections.

- Logon_ID: This 16 bit field identifies the assigned terminal logon identifier, which is used in the TBTP and CMT sections.

- Continuous_carrier: This flag indicates whether the descriptor applies to the MF-TDMA air interface (Value "1") or the continuous carrier air interface.(Value "0").

- Security_handshake_required: The value "1" indicates that the security handshake as described in clause 9.4 is to be used. Otherwise it is set to "0".

- Prefix_flag: The value "1" indicates that the Prefix Method according to clause 6.6.1.2 is implemented in the network and the value "0" indicates that it is not implemented.

- Data_unit_labelling_flag: The value "1" indicates that the Data Unit Labelling Method according to clause 6.6.2 is implemented in the network and the value "0" indicates that it is not implemented.

- Mini_slot_flag: The value "1" indicates that the Mini-slot Method according to clause 6.6.1.3 is implemented in the network and the value "0" indicates that it is not implemented.

- Contention_based_mini_slot_flag: The value "1" indicates that the Contention-based Mini-slot Method according to clause 6.6.1.4 is implemented in the network and the value "0" indicates that it is not implemented.

- Mesh_connectivity_flag: A value of "0" indicates that fields defining settings and limits for mesh capacity requests are present. A value of "1" indicates that these fields are absent.

- Capacity_type_flag: A value of "0" indicates that fields defining settings and limits for STAR capacity requests are present. A value of "1" indicates that these fields are absent.

- Traffic_burst_type: This 1 bit field defines the traffic burst type to be used on the return link. The value "0" indicates ATM TRF according to clause 6.2.1.1 and the value "1" indicates optional MPEG2-TS TRF according to clause 6.2.1.2.

- Connectivity: This 1 bit field defines the connectivity to be used. The value "0" indicates IP connectivity according to clause 8.1.1 (Type A RCST) and the value "1" indicates optional ATM connectivity according to clause 8.1.2 (Type B RCST). In the case of MPEG2-TS TRF the connectivity is always IP and therefore not signalled.

- Return_VPI, Return_VCI: These fields define the VPI/VCI that the RCST shall use in ATM cells on the return link.

- Return_signalling_VPI, Return_signalling_VCI: These fields define the VPI/VCI that is used on the return link for ITU-T Recommendation Q.2931 [8] signalling instead of the normal value 0/5. The signalling is used to set up connections for traffic. These parameters can be the same for all RCSTs.

- Forward_signalling_VPI, Forward_signalling_VCI: These fields define the VPI/VCI that is used on the forward link for ITU-T Recommendation Q.2931 [8] signalling instead of the normal values 0/5. The signalling is used to set up connections for traffic.

- Return_TRF_PID: This 13 bit field defines the PID that the RCST shall use in optional MPEG2 TS packets on the return link for traffic information. This parameter can be the same for all RCSTs.

- Return_CTRL_MNGM_PID: This 13 bit field defines the PID that the RCST shall use in optional MPEG2 TS packets on the return link for CTRL/MNGM information. This parameter can be the same for all RCSTs.

- CRA_level: The CRA assignment to the terminal for all STAR communications, in bits/s.

- VBDC_max: The maximum number of payload units per superframe allocated to the RCST through A/VBDC for STAR communications. This is not necessarily an absolute value for each superframe; it can be an average over several superframes.

- RBDC_max: The maximum allowed RBDC data rate, in bits/s for all STAR communications.

- RBDC_timeout: The RBDC timeout, in superframes for all STAR communications. A value of "0" indicates that the timeout is disabled.

- CRA_mesh_level: The CRA assignment to the terminal for all mesh communications, in bits/s.

- VBDC_mesh_max: The maximum number of payload units per superframe allocated to the RCST through A/VBDC for mesh communications. This is not necessarily an absolute value for each superframe; it can be an average over several superframes.

- RBDC_mesh_max: The maximum allowed RBDC data rate, in bits/s for all mesh communications.

- RBDC_mesh_timeout: The RBDC timeout, in superframes for all mesh communications. A value of "0" indicates that the timeout is disabled.

### 8.5.5.10.23    Implementation type descriptor (Optional)

This descriptor (table 78) supports indication of the generic implementation type of the hub. The indicated options can be interpreted by the RCST without additional information. The information provided can assist the RCST in choosing a way to operate that will work. The descriptor supports system specific extensions.

**Table 78: Implementation type descriptor**

| Syntax | No. of bits | | Information Mnemonic |
|---|---|---|---|
| | Reserved (see note) | Information | |
| Implementation_type_descriptor() { | | | |
| descriptor_tag | | 8 | uimsbf |
| descriptor_length | | 8 | uimsbf |
| ncc_protocol version | | 8 | uimsbf |
| location_update_allowed_flag | 2 | 1 | bslbf |
| rbdc_accepted_flag | | 1 | bslbf |
| vbdc_accepted_flag | | 1 | bslbf |
| avdbc_accepted_flag | | 1 | bslbf |
| timing_offset_flag | | 1 | bslbf |
| timing_reference_flag | | 1 | bslbf |
| C2P_protocol_version | 5 | 3 | uimsbf |
| hub_type_id | | 24 | uimsbf |
| hub_sw_id | | 24 | uimsbf |
| user_options_count | | 8 | uimsbf |
| For(i=0; i < user_options_count; i++){ | | | |
| user_options_byte | | 8 | uimsbf |
| } | | | |
| For(i=0; i < n; i++){ | | | |
| reserved_byte | | 8 | uimsbf |
| } | | | |
| } | | | |
| NOTE:    Reserved bits are of type bslbf, and shall precede the Information bits on the same line. They shall be ignored by the RCST. For an encrypted uni-cast TIM, the bit values shall be varied in a random manner to avoid encryption spoofing. | | | |

Semantics for the Implementation_type_descriptor:

- descriptor_tag: The descriptor tag is an 8 bit field which identifies each descriptor. Its value is given in the Tag value column of table 50.

- descriptor_length: The descriptor length is an 8 bit field specifying the number of bytes of the descriptor immediately following the descriptor_length field. This descriptor is specifically foreseen to be extended at the end with more specific bytes in future revisions, to provide space for more information about the generic implementation type. If the descriptor_length specifies fewer bytes than is needed to hold the information specified in table 78, the descriptor is truncated at the corresponding point. This way, the information can be safely interpreted from the first byte following the header with the interpretation known by the specific RCST, skipping any unknown content. A specific system may use a short or long descriptor to convey as much information as desired. However, if a non-zero value of user_options_count is specified, the descriptor_length shall be sufficient to accommodate the specified number of user_options_bytes.

- ncc_protocol_version: This 8 bit field indicates the RCS protocol version implemented by the NCC, coded as specified in table 79.

**Table 79: Standard version implemented by the NCC**

| Value | Version number |
|-------|----------------|
| 2-255 | Reserved |
| 1 | Version 1.5 |
| 0 | Version 1.4 or earlier version |

- location_update_allowed_flag: A "1" indicates that the hub generally allows location updates issued by the RCST in the CSC slot as specified in section "CSC burst format". A "0" indicates that location update by use of the CSC slot is generally prohibited, but may be specifically allowed for the RCST in question according to system specific methods.

- rbdc_accepted_flag: A "1" indicates that the hub honours RBDC requests on the default channel. A "0" indicates that the hub may unconditionally discard RBDC requests on the default channel.

- vbdc_accepted_flag: A "1" indicates that the hub honours VBDC requests on the default channel. A "0" indicates that the hub may unconditionally discard VBDC requests on the default channel.

- avbdc_accepted_flag: A "1" indicates that the hub honours AVBDC requests on the default channel. A "0" indicates that the hub may unconditionally discard AVBDC requests on the default channel.

- timing_offset_flag: A "1" indicates that the hub requires that the RCST offsets its timing from the native NCR packet source reference point as indicated in the NCR payload field. A "0" indicates that an RCST may attempt to log on using the native timing of the system even if it discards the optional NCR payload.

- timing_reference_flag: A "1" indicates that the hub applies the nominal position of the return link satellite for each return path as the native NCR packet source reference point, referring to the point of time when the PCR packet with the NCR value entered the channel interleaver and FEC encoder at the nominal link rate, or to the applicable start of the frame signal (SOF) as specified for DVB-S2 ACM/VCM. A "0" indicates that the hub may apply a system specific native NCR packet source reference point.

- C2P_Protocol_version: This 3-bit field defines the version of the connection control protocol defined in [23] that is supported by the NCC. The coding of the field is defined in table 80.

**Table 80: Connection control protocol version implemented by the NCC**

| Value | Version number |
|-------|----------------|
| 0 | No connection control protocol supported |
| 1 | Version 1 |
| 2 - 7 | Reserved |

- hub_type_id: A 24 bit parameter that is used to identify the type of hub. The value used can be an OUI value registered by the hub vendor in the IEEE Registration Authority as a company id. If such an OUI is not applicable for the hub, the field must be set to 0xFFFFFF to unconditionally indicate an unknown hub type.

- hub_sw_id: A 24 bit parameter that identifies the SW running on the hub in the context of a known hub type. It should be discarded if the hub_type_id is unknown.

- user_options_count: The number of bytes used to indicate user defined options.

- user_option_byte: The concatenated user_option_byte field contains user defined options. These options must be interpreted in the context of a known hub_type_id, and may have to be interpreted in the context of the hub_sw_id. The content shall be discarded if the hub_type_id is unknown.

- reserved_byte: This content shall be discarded by the receiver.

### 8.5.5.10.24       LL_FEC_identifier_descriptor (optional)

This descriptor (table 81) defines the characteristics of one or more link layer FEC frames. The FEC frames shall not employ time slicing.

**Table 81: LL_FEC identifier descriptor**

| Syntax | No. of bits | | Information Mnemonic |
|---|---|---|---|
| | Reserved (see note) | Information | |
| fec_identifier_descriptor() { | | | |
| descriptor_tag | | | |
| descriptor_length | | | |
| loop_count | | 8 | uimsbf |
| for (i=0; i < loop_count; i++) { | | | |
| link_direction | | 1 | bslbf |
| encapsulation_type | | 1 | bslbf |
| if (encapsulation_type == "0") { | | | |
| elementary_stream_id | 1 | 13 | uimsbf |
| } else { | | | |
| gse_fec_id | | 14 | uimsbf |
| } | | | |
| ll_fec | | 2 | uimsbf |
| frame_size | 2 | 3 | uimsbf |
| buffer_timeout | | 3 | uimsbf |
| dscp | | 6 | uimsbf |
| reserved for future use | 32 | | uimsbf |
| } | | | |
| } | | | |
| NOTE:       Reserved bits are of type bslbf, and shall precede the Information bits on the same line. They shall be ignored by the RCST. For an encrypted uni-cast TIM, the bit values shall be varied in a random manner to avoid encryption spoofing. | | | |

Semantics for LL-FEC identifier descriptor:

- **descriptor_tag:** Shall be set to value of 0xB6.

- d**escriptor_length:** This 8-bit field specifies the number of bytes of the descriptor immediately following this field.

- **loop_count:** This is an 8 bit field indicating the number of iterations in the loop that follows.

- **link_direction:** This 1-bit field indicates whether the referenced stream is transmitted in the forward link (set to "0") or in the return link (set to "1").

- **encapsulation_type:** This 1-bit field indicates whether the referenced stream is a elementary stream (set to "0") or a GSE-FEC stream (set to "1").

- **elementary_stream_id:** This field identifies the 13-bit PID of the elementary stream carrying the LL-FEC frames.

- **gse_fec_id:** This 14-bit field carries a unique value that identifies the FEC process. It is used to associate individual LL_FEC frames with the process.

- **ll_fec:** This 2-bit field indicates whether the referenced elementary stream uses LL-FEC, and which algorithm is used. Coding of this field is according to table 82.

**Table 82: LL_FEC algorithm**

| value | LL-FEC | Algorithm |
|-------|--------|-----------|
| 00 | LL-FEC not used | n/a |
| 01 | LL-FEC used | Reed-Solomon code |
| 10 | LL-FEC used | Raptor code |
| 11 | reserved for future use | reserved for future use |

- **frame_size:** This 3-bit field indicates the exact number of rows in each LL-FEC Frame. The coding of the field is according to table 83.

**Table 83: LL_FEC frame size coding**

| Value | LL-FEC Frame rows (RS) | LL-FEC Frame rows (Raptor) | Address Granularity (Raptor) | Max LL-FEC ADT Size (Raptor, Informative) |
|-------|------------------------|----------------------------|------------------------------|-------------------------------------------|
| 0x00 | 256 | 256 | 2 | 16 777 216 bits = 16 Mbits |
| 0x01 | 512 | 512 | 4 | 33 554 432 bits = 32 Mbits |
| 0x02 | 768 | 768 | 6 | 50 331 648 bits = 48 Mbits |
| 0x03 | 1 024 | 1 024 | 8 | 67 108 864 bits = 64 Mbits |
| 0x04 | reserved for future use | 64 | 1 | 4 194 304 bits = 4 Mbits |
| 0x05 | reserved for future use | 2 048 | 16 | 134 217 728 bits = 128 Mbits |
| 0x06 | reserved for future use | 4 080 | 32 | 267 386 880 bits = 255 Mbits |
| 0x07 | reserved for future use | reserved for future use | reserved for future use | reserved for future use |
| NOTE: The address granularity is 1 for all Reed-Solomon code options. | | | | |

- **buffer_timeout:** This 3-bit field indicates the time in milliseconds for the maximum interval between the transmission of the first section with a given fec_frame_number (in general a data section) and the transmission of last section with the same fec_frame_number (in general a parity section). The field is coded in accordance with table 84.

**Table 84: Coding of buffer_timeout**

| Value | buffer timeout (ms) |
|-------|---------------------|
| 0x00 | 10 |
| 0x01 | 40 |
| 0x02 | 160 |
| 0x03 | 640 |
| 0x04 | 2 560 |
| 0x05 | 10 240 |
| 0x06 | not specified |
| 0x07 | reserved for future use |

- **dscp:** This 6-bit field can be used to signal a Differentiated Services Code Point (DSCP) associated with LL-FEC Frame or any other type of Quality-of-Service label. The value is system-dependent.

- **reserved_for_future_use:** This 32-bit field shall be set to "0xFFFFFFFF".

### 8.5.5.11    Accessing of the forward link signalling

Service Information for Forward Link Signalling (FLS) shall be transmitted in one or more Forward Link Signalling services. Being a DVB data service, the FLS service shall be referenced like any other DVB service in the Program Association Table (PAT), Program Map Table (PMT) and optionally Service Description Table (SDT). If contained in the SDT, the service_type shall be set to 0x0F for "RCS FLS". The PMT of the FLS service shall contain the PIDs for the NCR, the General SI tables and RCST specific messages as defined in the present document.

Additional information about the Satellite Interactive Network shall be conveyed in a RCS Map Table (RMT). This table having the same syntax as the NIT - but being transmitted on a different PID - shall contain one or more linkage descriptors pointing to one or more FLS services. Each linkage descriptor shall contain a list of population_ids, which are being used by the RCST to select its appropriate FLS service. For this purpose the RCST has a default population_id for the installation process which may be changed at a later stage and be memorized for following network logons. The population_id values are unique on the satellite network, except for mobile applications where a population_ID value may be used on several beams and in some cases on several TS_IDs.

An interactive network is defined by a group of terminals serviced by a single NCC. In the simplest case, each NCC handles a single interactive network, however a large NCC logically splits its capacity into several interactive_networks. A population is a subset of a particular interactive network.

For the purposes of the present document, the forward link acquisition procedure is concerned with acquiring the forward link that carries the Forward Link Signalling service. This forward link acquisition procedure comprises the following steps:

- physical link synchronization;

- forward link location;

- NCR synchronization;

- DVB-RCS specific table loading.

*Physical link synchronization* is the process of tuning to the required transponder, recovering the transmitted signal and synchronizing to the DVB transport stream formatting on that signal. This shall be done each time an RCST is required to move to a different transponder.

*Forward link location* is the procedure for identifying which transponder carries the Forward Link Signalling service, and for identifying the PIDs. This procedure uses three standard DVB table types as follows:

- Network Information Table (NIT);

- Program Allocation Table (PAT);

- Program Map Table (PMT).

The detailed procedure for forward link location shall be as defined below in this section. It should be noted that this procedure may involve tuning to a maximum of three transponders in sequence, requiring a physical link synchronization for each transponder.

An outline of the forward link location procedure flow is shown in figure 42. This procedure requires that the following parameters be stored at the RCST as power-up configuration data:

- Location details for forward link start-up Transport Stream (TS). This can be any Transport Stream in the network. The parameters correspond to those defined in the satellite_delivery_system_descriptor of EN 300 468 [3].

- Population_ID value.

The RCST shall tune to the start-up Transport Stream and scan the copy of the NIT to locate the Transport Stream carrying the RMT. This shall be identified by locating the linkage descriptor containing the linkage_type code (0x07) for the RCS Map service. It shall use the TS_id parameter from that linkage descriptor to locate the satellite_delivery_system_descriptor for that Transport Stream (in the second loop of the NIT), and hence the tuning details for the Transport Stream, which carries the RMT.

The RCST shall then re-tune (if necessary) to the Transport Stream carrier that carries the RMT and shall load the RMT using the assigned PID. It shall scan the RCS Map Table for all linkage_descriptors containing the RCS FLS service linkage_type code 0x81, to find the descriptor containing its population_id (in the private data extension to that descriptor type). This descriptor will also contain the Interactive_network_id to be used by that RCST, together with the TS_id for the Transport Stream to be used by the RCST for logon and the service_id to be used by the PAT later.

The RCST shall then locate the entry for that TS_id in the second loop of the RCS Map Table and extract the satellite forward link and satellite return link descriptors. Where there are multiple instances of these, then it shall extract the signalling satellite forward link and the primary satellite return link as a minimum, and use these for the balance of the procedure. These descriptors contain initial parameters required by the subsequent Return Link Acquisition process, namely the satellite_id for both forward and return links, the gateway_id for the return link, the NCC_id for forward link, plus the superframe_id and transmit centre frequency offset for the return link. The RCST shall then re-tune (if necessary) to the Transport Stream carrier carrying its forward link signalling, using the parameters defined in the signalling satellite forward link descriptor, and locate the PAT entry with prg_nbr parameter equal to the service_id. This defines the PID for the associated PMT.

From the PMT, it shall extract the PID for the PCR Insertion packet (PCR_PID), and the PIDs for the RCS specific signalling tables. These PIDs are identified by locating those stream definitions containing the RCST content descriptor (defined in clause 8.5.5.10.10), and associating the table_id's for RCS tables with the elementary_PID value defined for that stream.

The result of this forward link location procedure shall be the following:

- identification of the NCC and satellite for the Forward Link Signalling service;

- identification of the Gateway and satellite for the return link;

- definition of the PIDs for the DVB-RCS forward signalling messages;

- definition of the superframe to be used for the return link logon procedure.

*NCR synchronization* is the procedure for locking the RCST timing to the NCR count at the NCC. This shall be done by receiving the PCR Insertion Transport Stream Packet (see clause 8.5.5.6) at the transmitted interval and deriving a synchronous PCR count and PCR clock locally in the RCST that are locked to the PCR count in that packet. Upon successful acquisition of the receive NCR, the RCST shall continue to monitor it. As a matter of safety if the reception of the forward link is interrupted, the RCST shall cease transmission if conditions defined in clause 7.7.3 are fulfilled.

*DVB-RCS specific table loading*. The RCST shall load the following tables, using the PIDs identified in the forward link location procedure:

- SCT, FCT, and TCT to identify the available superframe and frame structures;

- SPT to obtain the satellite ephemeris data for the forward link signalling satellite and return link satellite;

- TIM (broadcast version) to identify possible link anomalies.

In addition, the RCST shall monitor for unicast TIM messages addressed to the RCST MAC address for possible wake-up messages.

An RCST is capable of receiving DVB-S only, DVB-S2 only, or both. In a network that shall support the first two kinds, the RCS Map Table must be transmitted simultaneously on a transponder with DVB-S and a transponder with DVB-S2. Different Population IDs for the three kinds link each RCST to an FLS and forward link traffic that it can receive.

transponder 88

transponder 71

any transponder on satellite network

PAT        PID=0x0000
table_id=0x00
TS_id (=1088)
 prg_nbr=0 network_PID=0x0010 [NIT]
 prg_nbr(=27050) PMT_PID(=512) [TV channel 1]
 prg_nbr(27051) PMT_PID(=513) [TV channel 2]
 prg_nbr(=28153) PMT_PID(=514) [Data Carrousel]
 prg_nbr(=1303) PMT_PID(=524) [IP DATA in MPE 7]
 prg_nbr(=1304) PMT_PID(=525) [IP DATA in MPE 8]
 prg_nbr(=1304) PMT_PID(=525) [DATA in ATM 4]
 prg_nbr(=802) PMT_PID(=802) **[Forward signalling 2]**
 prg_nbr(=803) PMT_PID(=803) [Forward signalling 3]

RCS Map Table        PID(=Y) [assigned through PAT,
PMT]
table_id=0x41
network_id
network

linkage_descr [Forward signalling 1]
 TS_id(=1071)

linkage_descr [Forward signalling 2]
TS_id(=1088)
orig_network_id (=X)
**service_id(=802)**
**linkage_type=0x81 [RCS FLS service]**
 private data
            { interactive_network_id
             list of population_ids }

NIT        PID=0x0010
table_id=0x40
network_id
network_name_descr

linkage_descr [information service]

linkage_descr [CA repl. service]

linkage_descr [EPG]

linkage_descr [information service]
TS_id(=1071)
orig_network_id(=X)
service_id(=10)
linkage_type(=0x01) [info service]

5

PMT        PID(=802)
table_id=0x02
prg_nbr(=802)
**PCR_PID(=0x1600) [PCR Insertion TS Packet]**
CA_descr [if required]
PDSD [private_data_specifier_descr] (optional)
private_descr [spare] (optional)

stream_type=0x05 [private section]
**elementary_PID(=820)**
RCS content descriptor
 table_id=0xA0 [SCT]
 table_id=0xA1 [FCT]
 table_id=0xA2 [TCT]
 table_id=0xA3 [SPT]

   table_id=0xB0 [TIM]

     table_id=0xA5 [TBTP]
     table_id=0xA4 [CMT]

3

TS_id(=1071)

**TS_id(=1088)**
orig_network_id (=X)

service_list_descr (optional)
srvc_id(=27050) srvc_type(=0x01)
srvc_id(=27051) srvc_type(=0x01)
srvc_id(=28153) srvc_type(=0x0C)
srvc_id(=802) **srvc_type=0x0F [RCS FLS]**

satellite_forward_link_descr
satellite_id(=0x23)
NCC_id
frequency(12,10000)
orbital_position
west_east_flag; polarization;
modulation; symbol_rate; FEC_inner

satellite_return_link_descr
satellite_id(=0x06)
gateway_id
orbital_position
west_east_flag
Superframe_ID(=0x42)
Tx_frequency_offset(=0)

2

linkage_descr [RCS Map 1] (optional)

linkage_descr [RCS Map 1]
TS_id(=1071)
orig_network_id(=X)
service_id(=810)
**linkage_type=0x07 [RCS Map service]**
 private data [spare]

1

TS_id(=2089)

TS_id(=1089)

**TS_id(=1071)**
orig_network_id (=X)

service_list_descr (optional)

satellite_delivery_descr
frequency(=011,80000)
orbital_position
west_east_flag; polarization;
modulation; symbol_rate; FEC_inner

4

6

SPT        PID=820
table_id=0xA3

TCT        PID=820
table_id=0xA2

FCT        PID=820
table_id=0xA1

SCT        PID=820
table_id=0xA0
{
table content
}

TIM        PID=830
table_id=0xB0
 MAC address
 message_type
 { message content }

CMT        PID=831
table_id=0xA4
{
 TBTP        PID=831
 table_id=0xA5
 {
 table content
 }

1 = seek tuning parameters for RCS Map Table
2 = access RCS Map Table
3 = seek tuning parameters for Forward Signaling Service
     ("Forward signalling 2") with matching
     interactive_network_id and population_id
4 = do tuning (if necessary) and look for PAT
5 = access PMT of selected Forward Signaling Service
6 = get PIDs for SCT, FCT, TCT, TIM, TBTP, CMT, SPT,
etc.

**Figure 42: Example of forward link location**

## 8.5.5.12    RCS Map Table

The RCS Map Table enables to link different populations of RCSTs to specific FLS. The syntax is based on the syntax of the DVB Network Information Table that is defined in EN 300 468 [3].

The RCS Map Table shall contain one or multiple linkage descriptors each pointing to one FLS service. Each FLS service shall carry a set of signalling tables (SCT, TCT, FCT, SPT, TBTP, CMT) and TIMs for a defined RCST population. The linkage descriptor is defined in clause 8.5.5.10.9. An RCST selects its FLS service by comparing its own Population ID with the Population IDs listed in each of the linkage descriptors.

The RCS Map Table shall contain a loop over MPEG Transport Streams that carry FLS. For each of these MPEG Transport Streams it shall contain a Satellite Forward Link descriptors, which is defined in clause 8.5.5.10.11, and a Satellite Return Link descriptor, which is defined in clause 8.5.5.10.12.

The RCS Map Table shall be segmented into sections using the syntax of table 85.

**Table 85: RCS Map Table**

| Syntax | No. of bits | | Information Mnemonic |
|---|---|---|---|
| | Reserved | Information | |
| RCS_map_section(){ | | | |
|     table_id | | 8 | uimsbf |
|     section_syntax_indicator | | 1 | bslbf |
|     reserved_future_use | | 1 | bslbf |
|     Reserved | | 2 | bslbf |
|     section_length | | 12 | uimsbf |
|     network_id | | 16 | uimsbf |
|     Reserved | | 2 | bslbf |
|     version_number | | 5 | uimsbf |
|     current_next_indicator | | 1 | bslbf |
|     section_number | | 8 | uimsbf |
|     last_section_number | | 8 | uimsbf |
|     reserved_future_use | | 4 | bslbf |
|     network_descriptors_length | | 12 | uimsbf |
|     for(i=0;i<N;i++){ | | | |
|         descriptor() | | | |
|     } | | | |
|     reserved_future_use | | 4 | bslbf |
|     transport_stream_loop_length | | 12 | uimsbf |
|     for(i=0;i<N;i++){ | | | |
|         transport_stream_id | | 16 | uimsbf |
|         original_network_id | | 16 | uimsbf |
|         reserved_future_use | | 4 | bslbf |
|         transport_descriptors_length | | 12 | uimsbf |
|         for(j=0;j<N;j++){ | | | |
|         descriptor() | | | |
|         } | | | |
|     } | | | |
|     CRC_32 | | 32 | rpchof |
| } | | | |

Semantics for the RCS Map Table:

- table_id: This 8 bit field identifies the table. The value is 0x41.

- section_syntax_indicator: The section_syntax_indicator is a 1-bit field which shall be set to "1".

- section_length: This is a 12-bit field, the first two bits of which shall be "00". It specifies the number of bytes of the section, starting immediately following the section_length field and including the CRC. The section_length shall not exceed 1 021 so that the entire section has a maximum length of 1 024 bytes.

- network_id: This is a 16-bit field which serves as a label to identify the delivery system, to which the table shall apply.

- version_number: This 5-bit field is the version number of the sub_table. The version_number shall be incremented by 1 when a change in the information carried within the sub_table occurs. When it reaches value 31, it wraps around to 0. When the current_next_indicator is set to "1", then the version_number shall be that of the currently applicable sub_table defined by the table_id and _network_id_mask. When the current_next_indicator is set to "0", then the version_number shall be that of the next applicable sub_table defined by the table_id and _network_id.

- current_next_indicator: This 1-bit indicator, when set to "1" indicates that the sub_table is the currently applicable sub_table. When the bit is set to "0", it indicates that the sub_table sent is not yet applicable and shall be the next sub_table to be valid.

- section_number: This 8-bit field gives the number of the section. The section_number of the first section in the sub_table shall be "0x00". The section_number shall be incremented by 1 with each additional section with the same table_id and network_id.

- last_section_number: This 8-bit field specifies the number of the last section (that is, the section with the highest section_number) of the sub_table of which this section is part.

- network_descriptors_length: This 12-bit field gives the total length in bytes of the following network descriptors. Network descriptors shall contain the Network Name descriptor from EN 300 468 [3] and one or more linkage descriptors containing private data bytes as defined in clause 8.5.5.10.9. Additional descriptors from the NIT definition in EN 300 468 [3] are optional and the RCST does not need to use them.

- transport_stream_loop_length: This is a 12-bit field specifying the total length in bytes of the Transport Stream loops that follow, ending immediately before the first CRC-32 byte.

- transport_stream_id: This is a 16-bit field which serves as a label for identification of this Transport Stream from any other multiplex within the delivery system.

- original_network_id: This 16-bit field gives the label identifying the network_id of the originating delivery system.

- transport_descriptors_length: This is a 12-bit field specifying the total length in bytes of Transport Stream descriptors that follow. Transport Stream descriptors shall contain the Satellite Forward Path descriptor, which is defined in clause 8.5.5.10.11, and the Satellite Return Path descriptor, which is defined in clause 8.5.5.10.12. Additional descriptors from the NIT definition in EN 300 468 [3] are optional and the RCST does not need to use them.

- CRC_32: This is a 32 bit field that contains the CRC value that gives a zero output of the registers in the decoder defined in annex B of EN 300 468 [3] after processing the entire section.

## 8.5.5.13    Transmission Mode Support Table

The Transmission Mode Support Table defines the DVB-S2 transmission modes supported by the network for forward link transmission. If the forward link is transmitted using DVB-S2, then this table must be transmitted as part of the forward link signalling.

The table contains a loop over transmission mode definitions. MODCOD, pilot symbols and FECFRAME are defined in [16]. Use of this table is defined in clause 5.

The Transmission Support Mode Table shall be segmented into sections using the syntax defined in table 86.

**Table 86: Transmission Mode Support Table**

| Syntax | No. of bits | | Information Mnemonic |
|---|---|---|---|
| | Reserved | Information | |
| transmission_mode_support_section(){ | | | |
|    SI_private_section_header | | 64 | - |
|    transmission_mode_count | | 8 | uimsbf |
|    for(i=0; i<transmission_mode_count; i++){ | | | |
|       frame_length | | 2 | bslbf |
|       pilot_symbols | | 1 | |
|       MODCOD | | 5 | bslbf |
|    } | | | |
|    CRC_32 | | 32 | rpchof |
| } | | | |

Semantics for the transmission_mode_support_section:

- SI_private_section_header: This is the standard SI private section header defined in table 29, and occupies a total of 64 bits.

- transmission_mode_count: This is the number of iterations in the loop that follows. Each interation describes one DVB-S2 transmission mode that is supported by the network.

- frame length: This field indicates the possible FECFRAME length applied with the transmission mode described by the iteration. "01" means short frames only, "10" means long frames only, "11" means both short and long frames, "00" is reserved.

- pilot_symbols: This field indicates the use of pilot symbols for the transmission mode described by the iteration. "1" means that pilot symbols are used, "0" means that they are not used.

- MODCOD: This field indicates the modulation scheme for the transmission mode described by the iteration. The definition of values is the same as for the MODCOD parameter in the DVB-S2 standard [16].

- CRC_32: This is a 32-bit field that contains the CRC value that gives a zero output of the registers in the decoder defined in annex B of EN 300 468 [3] after processing the entire section.

## 8.5.5.14    LL-FEC parity data table

The LL_FEC parity data table enables the transport of parity data for the systematic codes used to overcome channel interruptions and other imperfections encountered by mobile terminals. LL_FEC parity sections are carried in the same elementary stream as the corresponding application data.

The LL_FEC parity data table shall be as defined in table 87. It shall be segmented into sections using the syntax described in EN 300 468 [3].

NOTE:    The encapsulation_type meta-variable is not carried explicitly in this table. The pertinent value is defined in the LL_FEC_identifier_descriptor entry (clause 8.5.5.10.24) that applies to the LL-FEC data being transported.

**Table 87: LL_FEC Parity DataTable**

| Syntax | No. of bits | | Information Mnemonic |
|---|---|---|---|
| | Reserved | Information | |
| LL_FEC_parity_data_section(){ | | | |
| table_id | | 8 | uimsbf |
| section_syntax_indicator | | 1 | bslbf |
| private_indicator | | 1 | bslbf |
| reserved | 2 | | bslbf |
| section_length | | 12 | uimsbf |
| if (encapsulation_type == "0") { | | | |
| padding_columns | 5 | 8 | uimsbf |
| } | | | |
| else { | | | |
| no_adt_columns | | 13 | uimsbf |
| } | | | |
| reserved for future use | | 17 | bslbf |
| if (encapsulation_type == "0") { | | | |
| section_number | | 8 | uimsbf |
| last_section_number | | 8 | uimsbf |
| } | | | |
| else { | | | |
| encoding_symbol_id | | 16 | uimsbf |
| } | | | |
| rcs_real_time_parameters() | | 26 | see semantics |
| for (i=0; i<N; i++) { | | | |
| parity_data_byte | | 8 | uimsbf |
| } | | | |
| CRC_32 | | 32 | rpchof |
| } | | | |

Semantics for LL_FEC_parity_data_section:

- table_id: This 8 bit field identifies the table. The value is 0xB1.

- section_syntax_indicator: This field shall be set to 1 and be interpreted as defined by ISO/IEC 13818-6 [14].

- private_indicator: This field shall be set to 0 and be interpreted as defined by ISO/IEC 13818-6 [14].

- reserved: Shall be set to "11".

- section_length: This 8-bit field specifies the number of remaining bytes in the section immediately following this field up to the end of the section, including the CRC-32.

- padding_columns: This 8-bit field indicates the number of full columns of the Application data table of the actual LL-FEC Frame filled with padding bytes only; i.e. it is equal to 191 minus the number of columns that contain Application data. The value indicated shall be from 0 to 190. The value may vary frame by frame.

- no_adt_columns: This 13-bit field indicates the number of columns of the Application data table of the actual FEC Frame. The value indicated shall be from 0 to 8 191. The value may vary frame by frame.

- reserved_for_future_use: This 17- bit field shall be set to "0x1FFFF".

- section_number: This 8-bit field gives the number of the section. The section_number of the first section carrying RS data of an LL-FEC Frame shall be "0x00". The section_number shall be incremented by 1 with each additional section carrying RS data of the corresponding LL-FEC Frame.

- last_section_number: This 8-bit field shall indicate the number of the last section that is used to carry the RS data of the current LL-FEC Frame.

- encoding_symbol_id: This 16-bit field specifies the ESI of the first encoding symbol of this section. The ESI shall be greater than or equal to *no_adt_columns* for the LL-FEC Frame.

- rcs_real_time_parameters: This 26-bit field carries RCS Real-time parameters for the FEC parity data. The details are specified in clause 6.4.5.6.

- parity_data_byte: Contain the parity data delivered. The number of parity_data_bytes carried in the section shall be a multiple of the number of rows in the corresponding LL-FEC Frame.

- CRC_32: This is a 32-bit field that contains the CRC value that gives a zero output of the registers in the decoder defined in annex B of EN 300 468 [3] after processing the entire section.

# 8.6        Mobility management (Mobile Req)

Beam handover management shall rely on three distinct processes: handover detection/recommendation, handover decision and handover execution. They involve a number of events/activities taking place in RCST and NCC.

- Beam handover management shall be based on a distributed approach, with the detection/recommendation taking place in RCST and the handover decision in NCC. The NCC shall also control the overall handover execution process.

NOTE 1:  The NCC can derive the location of the RCST at log-on time (i.e. the RL transponder/beam) from the Superframe_ID - Beam_ID table in its database, using as entry the Superframe_ID of the acquired return link.

- The RCST shall detect the need for handover and shall signal this by means of transmitting one or more Mobility_Control_Message SAC subfields containing handover requests and prioritised recommendations for target beams.

- Each handover request can refer to the forward link, the return link or to both. In situations where handover is requested for both links but these are operated through separate beams, individual requests for forward and return link handover shall be issued.

- The RCST shall attempt to transmit the initial handover request at least 10 seconds prior to the time at which it is predicted that the link would be lost without a handover.

- The RCST may send multiple copies of the set of handover requests until their receipt by the NCC is acknowledged (by the transmission of a TIM with the handover command, see below).

- The decision algorithm in the NCC shall accept multiple such handover recommendations from an RCST and respond with a unique handover decision.

- The NCC shall be capable of including the relevant signalling in the DVB-RCS tables distributed in the beam in which the RCST finds itself at a given time. In particular, the NCC shall be capable of generating and transmitting a TIM in the current beam with information pertinent to the resources in the target beam.

- The handover execution shall start with the transmission of the handover command to RCST, together with all the information needed for operation in the target beam.

- The NCC shall transmit all information needed for operation in the target beam while the RCST is still operating in the current beam. This includes the distribution of the descriptions of all relevant forward and return links (RMT, SCT, FCT, TCT).

- The handover command and all the information related to the target beam shall be sent by using a unicast TIM with the following descriptors:

    - Network Layer Information Descriptor, carrying any changes to the signalling PID values.

    - Mobility Control Descriptor, including the relevant handover command(s).

    - Satellite Forward Link Descriptor, containing the new beam ID and the physical characteristics (frequency, polarisation, modulation/coding) of the new forward link.

    - Satellite Return Link Descriptor, containing the new beam ID and new Superframe ID with the associated Tx Frequency Offset.

NOTE 2:   The new Superframe ID is associated with a new frame structure, defined by the composition tables (SCT, FCT, TCT) corresponding to the new beam/transponder.

-   SYNC Assign descriptor, which gives the location (i.e. slot number), frame and start superframe of a new static SYNC assignment (applicable to the new frame structure), together with the repeat period. This is not required if SYNC slots are assigned through the TBTP.

-   Optionally an ACQ Assign descriptor, which gives the location (i.e. slot number), frame and start superframe of a temporarily assigned ACQ slot (applicable to the new frame structure), together with the number of repeats and the repeat period.

-   Forward Interaction Path Descriptor, containing the traffic PID values (if subject to changes).

-   Logon Initialize descriptor or Mesh Logon Initialize descriptor, containing the new Group ID, Logon ID.

- Upon the reception of the handover command the RCST shall extract and store the information needed for operation in the target beam.

- The RCST shall be capable of handling multiple sets of tables, physical and logical resources (corresponding to different beams) and of switching from one set to another as triggered by pre-defined events.

- The RCST synchronisation scheme shall be capable of coping with discontinuities in PCR distribution during handover.

- The RCST shall acquire burst synchronisation in the target beam by using the DVB-RCS mechanisms (including the use of SYNC bursts and CMT replies, and optional use of ACQ bursts).

- After the transmission of the handover command and during the acquisition by the RCST of the forward and return synchronisation in the target beam, the NCC shall stop forwarding traffic and signalling to the RCST.

- After activating the composition tables for the target beam, the RCST shall cease sending SYNC bursts to the old beam.

- An RCST in a logged-off state may send a CSC burst with the "location update" bit set to 0 in order to inform the NCC of its location. Reception of such a CSC burst shall not in itself cause the NCC to initiate a logon process.

# 8.7        Interference avoidance for mobile terminals (Mobile Req)

The mobile RCST shall comply with the applicable set of requirements for the Control and Monitoring functions (CMF) specified in the applicable ETSI MSS standards for aeronautical, maritime and land mobile stations [17], [18], [19], [20] and [24], covering the essential requirements under article 3.2 of the R&TTE directive. Article 3.2 concerns the use of spectrum under the R&TTE Directive [21]. The sections below specify the minimum required functions to meet the stated CMF requirements, to control interference and otherwise faulty operation for:

- off-axis EIRP emission density into adjacent FSS satellites;

- power flux density of received interference signal at specified terrestrial stations;

- fault conditions.

## 8.7.1      Off-axis EIRP emission density

The NCC shall control the level of off-axis EIRP emission density to comply with the regulatory requirements in [17], [18], [19], [20] and [24] by adjusting the corresponding on-axis EIRP density level. In order to permit the NCC to determine permissible transmit levels, the RCST shall support the Mobility_Control_Message SAC field and the Mobility Control descriptor in the unicast TIM. When requested, the RCST shall send the Transmission Status Report form of the Mobility_Control_Message at the earliest opportunity.

## 8.7.2    Power Flux Density at the surface of the earth

The NCC shall control the mobile terminal EIRP density to ensure that the received power flux density at the earth stations of the FS, RAS, SRS and any other applicable services does not exceed the relevant limits prescribed in [17], [18], [19] , [20] and [24].

The RCST shall be able to determine when such interference may occur in the near future. Upon detecting this situation, the RCST shall send the Exclusion Zone Entry form of the Mobility_Control_Message SAC field at the earliest opportunity. This message shall be sent at least twice.

The Mobility_Control_Message SAC field may contain a request for a particular type of remedial action to avoid :harmful interference. The decision shall be taken and implemented by the NCC. Possible remedial actions include

- Log off.

- Change frequency to a band with no or less restrictive limits.

- Change transmission parameters to comply with limits.

If the RCST is forced to log off in response to the transmission of the Exclusion Zone Entry form of the Mobility_Control_Message SAC field, it shall not resume transmission until it has determined that it has left the exclusion zone.

## 8.7.3    Fault conditions

The NCC shall periodically monitor and detect any fault conditions that may lead to malfunction of normal operation, including faults that result in both intra-system and intersystem interference, and shall take appropriate remedial actions. The fault conditions shall include the general requirements specified in [17], [18], [19], [20] and [24].

# 9        Security, identity, encryption

Security is intended to protect the user identity including its exact location, the signalling traffic to and from the user, the data traffic to and from the user and the operator/user against use of the network without appropriate authority and subscription. Three levels of security can be applied to the different layers:

- DVB common scrambling in the forward link (could be required by the service provider);

- satellite interactive network individual user scrambling in the forward and return link;

- IP or higher layer security mechanisms (could be used by the service provider, the content provider).

DVB Common Scrambling does not provide a security service per terminal and does not provide a two way security service (only forward link).

IP or higher layer security mechanisms are end-to-end solutions proposed by the service provider and/or the content provider.

DVB-RCS privacy (presented in clause 9.4) provides a security system at the data link layer so that the system is inherently secure on the satellite section without recourse to additional measures. This security system can be applied on both forward and return links and for both unicast and multicast data streams.

## 9.1      Authentication

Authentication may be implemented by request for a user name or password on the client device. In the case of a PC used as the client device, the RCST does not need to carry any special implementation. However, if the RCST contains a proxy client, then the proxy may be able to authenticate itself to the NCC. This means that an authentication server may be implemented at the NCC, which manages the authentication of each user.

Authentication could also be replaced by a Smart Card on the RCST, also used for the link layer individual control word encryption.

## 9.2        Forward link

DVB Common Scrambling could be required in the forward link.

Individual scrambling may be implemented at the section level, but the MAC address of the user may remain in the clear, since the RCST uses the MAC address to filter messages.

## 9.3        Return link

The client device may handle IPsec, so the router at the Gateway may be able to handle IPsec.

Individual layer 2 scrambling may also be implemented.

## 9.4        Security (optional)

The optional security mechanism described here is derived from the one used in [11].

The RCSTs indicates in the CSC bursts if it implements this option. The NCC indicates in the logon initialize descriptor or mesh logon initialize descriptor if this mechanism is to be used. This security mechanism consists of two separate sub-systems:

- a set of MAC messages used for authentication and key-agreement between NCC and RCST. These messages are used for key negotiation during a session setup as well as for on-the-fly update of keys (see clause 9.4.7);

- on-the-fly encryption and decryption of payload data streams.

When a session is being setup, one of three request/response MAC message-pairs is used to generate session keys specific to the payload streams associated with the session.

A session key is a shared secret between the NCC and the RCST: even if every MAC message is intercepted, the cryptographic properties of the protocol ensure that an eavesdropper cannot determine the session key value.

This is achieved by using a public-key protocol, which requires an up-front shared secret, or a simpler protocol based on a long-term shared secret between NCC and RCST called a cookie. The cookie is 160 bits long. It can be used by the NCC to authenticate the RCST logon.

Each RCST will store its own cookie in non-volatile storage, whereas the NCC will maintain a data-base of the cookie values of the RCSTs on its network. Cookie values will be updated occasionally as de-stated by security policy, but they are less vulnerable than session keys: a successful brute-force attack on a session key reveals nothing about the cookie value, nor any other session key.

The MAC messages also implement a defence against clones: an RCST that is a physical copy of an existing RCST and attempts to operate on the network under the cloned identity (when the cloned RCST itself is not registered on the network). The anti-cloning measure is a simple non-volatile 8-bit counter that is incremented synchronously at the NCC and RCST over time: if a clone RCST engages in traffic with the NCC, this will be detected the next time the cloned RCST connects because the counter value will be out of synchronization.

If the clone attempts to operate concurrently with the cloned unit, there will be an immediate break-down of functionality for both units, due to confusion within the MAC protocol. This amounts to a denial-of-service attack, and the NCC should be prepared for this kind of protocol failure.

### 9.4.1        Cryptographic primitives

The key exchange protocols and data stream encryption is based on a set of well-established primitive cryptographic functions. The functions and their associated key sizes can be changed in the future, in case crypt-analytic or brute-force attacks become a realistic threat.

The specific set of functions and key sizes are negotiated between NCC and RCST at sign-on time. The functions supported at the present time are Diffie-Hellman, HMAC-SHA1, DES and AES. Check current cryptographic literature for any updates regarding their security and use.

The following sections give a brief overview of the cryptographic primitives, and details on how they are used in the protocol. Later sections describe the exact field layout of the MAC messages.

The protocol parameters are described in terms of byte strings. Integer quantities are represented as base-256 byte strings.

Big-endian byte-ordering is used, that is, the most significant byte comes first. If necessary to reach a fixed length, the string is padded with zeros at the most significant end.

## 9.4.1.1 Public key exchange

A public key exchange primitive is used to allow the NCC and RCST to agree on a secret, although communicating in public. The Diffie-Hellman scheme is based on unsigned integer arithmetic and works as follows.

The NCC chooses two public values, a large prime number $m$, and a (small) number $g$ which is a generator modulo $m$ (that is, $g^a \mod m$ will generate all number from 0 to $m-1$ for varying a). The NCC also chooses a secret number $x < m$, and sends the following three values to the RCST: $m, g, X = g^x \mod m$.

The RCST chooses a secret value $y < m$, and responds to the NCC with the value $Y = g^y \mod m$.

The RCST now calculates $s = X^y \mod m = (g^x)^y \mod m = g^{(x*y)} \mod m$, whereas the NCC calculates $Y^x \mod m = (g^y)^x \mod m = g^{(y*x)} = s$, so the NCC and RCST now agree on the value $s$.

The value of $s$ is a secret shared between NCC and RCST. To determine its value from the publicly communicated values $m, g, X$, and $Y$, an eavesdropper shall determine $x$ or $y$ by solving an equation of the form $Z = g^z \mod m$ for unknown $z$. This is known as the discreet logarithm problem and is computationally infeasible with current algorithms for sufficiently large values of $m$.

The parameter size supported are 512 bits for the prime number $m$, and hence also for the remaining values since all arithmetic is modulo $m$.

In the applicable MAC messages, the unsigned integer quantities $m, g, X$, and $Y$ are encoded into fixed-size fields (64 bytes, 96 bytes or 128 bytes) using big-endian byte-ordering.

## 9.4.1.2 Hashing

The protocol makes use of a keyed hash function that computes secure checksums which can only be verified with the possession of a secret key. The function has the one-way property, meaning that it is computationally infeasible to find an input value that maps to a given output value.

The hash function is also used to generate derived secret material based on a master secret. Because of the one-way property, the master secret is protected even if the derived secret is discovered.

In generic terms, the keyed hash function takes two byte strings as input, the key and a data string, and produces another string of bytes, the digest:

- digest = H (key, data).

The H function shall accept key and data parameters of any size, whereas the protocol is designed to accept digests of any size.

The specification currently supports the HMAC-SHA1 function defined in RFC 2104 [12]. It produces a 20-byte digest.

## 9.4.1.3 Encryption

Payload data is encrypted and decrypted using symmetric-key block ciphers.

The specification currently supports two particular modes of operation to be associated to those symmetric-key block ciphers:

- Cipher Block Chaining (CBC) mode.

- Counter (CTR) mode.

**Cipher Block Chaining Mode**

In generic terms, the encryption and decryption functions take three byte strings as input, the key, an Initialization Vector (IV) and a data block, and produce as output another data block of the same length. The Cipher Block Chaining mode requires input plaintext blocks to be an exact multiple of the cipher block size. Padding operations are then necessary:

- ciphertext = E (key, IV, plaintext);

- D (key, IV, ciphertext) = plaintext.

The key length, IV length and block length is given by the chosen cipher, and the payload stream processing logic will apply it as appropriate to data units of various sizes.

The IV is an explicit non secret random binary value used as the initializing input algorithm for the encryption of the plaintext. A new IV value is calculated for every plaintext block to cipher. The IV must be unpredictable.

The IV value is conveyed in its clear form with the ciphertext. It precedes the protected (encrypted) payload.

The specification currently recommends AES algorithm in CBC mode implying a 16 bytes random value (IV) to be generated for every plaintext block and transmitted with the resulting ciphertext.

The specification also supports DES algorithm in CBC mode with an initialization vector set to zero.

**Counter Mode**

Inputs of the encryption and decryption functions for a counter mode approach are identical to the CBC mode except the Initialization vector to be replaced by a counter parameter:

- ciphertext = E (key,counter, plaintext);

- D (key,counter, ciphertext) = plaintext.

The key length and block length are given by the chosen cipher, and the payload stream processing logic will apply it as appropriate to data units of various sizes.

Counter characteristics (length, generation and manipulation) are also context specific. General requirement is that a counter value must not be used more than once associated to a given session key.

The specification currently support AES algorithm in CTR mode with the following counter usage:

- Counter length is 16 bytes.

Counter is made of three parts noted as nonce (4 bytes), counter (8 bytes), and block counter (4 bytes). Sub counter part values are big-endian integer values.

**Nonce**

The nonce value is a random value to be associated to a given session key. The value is transmitted with each session key during EKE exchanges (see clause 9.4.4). The same nonce value is then used for a given exchanged key.

**Master counter**

The master counter is a dynamic counter value to be associated to a session key.

It is managed by the senders this way:

- The master counter is set to 0 whenever beginning ciphering with a new session key.

- The master counter is then incremented by one for each plaintext to encrypt.

The master counter value is conveyed in its clear form with the ciphertext. It precedes the protected (encrypted) payload.

**Block Counter**

The block counter field is the least significant 4 bytes of the counter block. The bloc counter is set up to the value one for each plaintext to cipher (It will be incremented internally to generate subsequent portions of the key stream).

Two remarks are to be associated to such a management and format of the counter parameter:

- Re-keying operations must be conducted taking into account that the counter value has to be unique for a given session key. Situation authorizes no more than $2^{64}$ plaintext blocks to be ciphered with the same key.

- Due to the size of the block counter part of the counter, the maximum length of a plaintext block to be ciphered is 68 719 476 720 octets.

## 9.4.1.4       Pseudo-random numbers

The protocols used for generating secret values depend on the availability of a pseudo-random, that is, practically unpredictable, endless string of bytes. This will typically be produced with a Pseudo-Random Number Generator, PRNG, algorithm.

The random bytes are used to generate the secret Diffie-Hellman values, $x$ and $y$, and for nonce values used during key exchange. The unpredictable nature of the random input ensures that different secret values are produced each time, that a necessary randomness is associated to plaintext in order to limit cryptanalysis tasks and also prevents replay of old intercepted messages.

The present document does not require any particular algorithm, only that the NCC and RCST each choose one that is well-established and cryptographically analysed.

The hardest aspect of using a PRNG is to initialize it with an unpredictable seed value. The seed should contain multiple high-granularity device-dependent time-samples, as well as any other available pseudo-random material, like file allocation tables, etc. These random source values are then hashed together to squeeze out the entropy for the seed value.

## 9.4.1.5       Padding

Modes of operation related to cipher algorithms may require input plaintext blocks to be an exact multiple of the cipher block size. Padding operations are then necessary.

Corresponding padding scheme to use is:

- The padding bytes are initialized with a series of (unsigned, 1-byte) integer values. The first padding byte appended to the plaintext is numbered 1, with subsequent padding bytes making up a monotonically increasing sequence: 1, 2, 3, etc.

- Whenever the plaintext block size is a multiple of the cipher block size, a complete block of padding equal to the size of the cipher block size and following the previous scheme is to be added.

Previous rules allow the receiving part to unambiguously remove padding data after the deciphering operation.

The specification currently supports:

- AES algorithm in CBC which uses a block size of 16 octets. Padding operations are then required to maintain a 16-octet block size.

- DES algorithm in CBC which uses a block size of 8 octets. Padding operations are then required to maintain a 8-octet block size.

It is to note that AES in CTR does not require padding data.

## 9.4.2      Main Key Exchange (MKE)

Main Key Exchange (MKE) uses Diffie-Hellman to develop a shared secret between the NCC and RCST, which is independent of the cookie value. Furthermore, it uses the cookie value to authenticate the RCST to the NCC. It optionally uses the newly developed shared secret to update the cookie value. Finally, it derives a shared secret key used for the security context that is used to process payload stream data.

The exchange is initiated by the NCC sending a message containing the Diffie-Hellman values, `m`, `g`, `X`, and a random nonce string, `nonce1`. The RCST responds with a message containing its Diffie-Hellman value, `Y`, a random nonce string, `nonce2`, and an authentication string, `auth`.

The NCC and RCST each use the same formula to calculate the authentication string:

- `auth = H (cookie, nonce1 ~ nonce2);`

which is communicated by the RCST and checked by the NCC. This proves the identity of the RCST, since it requires knowledge of the cookie to calculate the correct value of `auth`.

The RCST and NCC each use the Diffie-Hellman values (see clause 9.4.1.1) to arrive at the same secret value, `s`:

- `s = g^(x*y) mod m.`

This unsigned integer value is encoded as a byte string, of length specified by the Diffie-Hellman parameter size, using big-endian byte ordering. It is then used to calculate a temporary shared secret string, `temp`:

- `temp = H (encode (s), nonce2 ~ nonce1).`

If the cookie is to be updated, the new value is computed in sections for `n = 1, 2, …`:

- `newcookie(n) = H (temp ~ (unsigned char)1 ~ (unsigned char)n, "").`

These string values are computed and concatenated until the total length matches or exceeds the length of the cookie. The cookie is then obtained by taking the first 20 bytes out of the concatenated sections, starting from the beginning.

The session key used for payload stream encryption is likewise computed in sections:

- `key(n) = H (temp ~ (unsigned char) 2 ~ (unsigned char) n, "").`

Where, again, a sufficient number of sections are calculated to produce enough bytes to cover the length of the key. The session key is obtained "in the same manner as the cookie" by taking the required number of bytes out of the concatenated sections, starting from the beginning.

## 9.4.3      Quick Key Exchange (QKE)

Quick Key Exchange (QKE) uses the existing cookie value to authenticate the RCST to the NCC, and then derive a shared secret key used for the security context that is used to process payload stream data.

The exchange is initiated by the NCC sending a message containing a random nonce string, `nonce1`. The RCST responds with a message containing a random nonce string, `nonce2`, and an authentication value, `auth`.

The value of `auth` is calculated in the same way as for Main Key Exchange, and can be used to verify the identity of the RCST (see clause 9.4.2).

The RCST and NCC then each calculate a temporary shared secret string, `temp`:

- `temp = H (cookie ~ (unsigned char)3, nonce2 ~ nonce1).`

This value is used to produce the payload encryption key in the same way as for Main Key Exchange (see clause 9.4.2).

## 9.4.4        Explicit Key Exchange (EKE)

Explicit Key Exchange (EKE) is used by the NCC to deliver a pre-determined session key to the RCST. The session key is encrypted under a temporary key derived from the cookie value, and is used for the security context that is used to process payload stream data.

The delivery is performed by the NCC sending a message containing a random nonce string, `nonce1`, and a byte string value, `encryptedkey`, which has the same length as a key used for payload encryption. The RCST responds with a message containing a random nonce string, `nonce2`, and an authentication value, `auth`.

The value of `auth` is calculated in the same way as for Main Key Exchange, and can be used to verify the identity of the RCST (see clause 9.4.2).

Both the NCC and RCST calculate a temporary shared secret string, `temp`:

  • `temp = H (cookie ~ (unsigned char)4, nonce1)`.

which is used to produce sections of a temporary key, in the same way as for Main Key Exchange (see clause 9.4.2).

The NCC uses these temporary `key` string sections to XOR with the session key to obtain the `encryptedkey` value, and the RCST performs a second XOR operation to decrypt the session key value.

For normal DES, 8 bytes of raw key data are delivered, which are used to derive the actual key with the appropriate number of effective bits, as described below (see clause 9.4.2).

For AES, 16 bytes of raw key are delivered, which are used to derive the actual key with the appropriate number of effective bits, as described below (see clause 9.4.2).

## 9.4.5        Key derivation

The actual key value used for processing payload data is derived from the `key` sections developed during key exchange.

For DES, 8 bytes of raw key data is required, so a single 20-byte section, `key(1)`, computed by HMAC-SHA1 is sufficient.

For AES, 16 bytes of raw key data is required, so a single 20-byte section, key(1), computed by HMAC-SHA1 is sufficient. In each byte, the least significant bit is not used (it can be used as an odd-parity bit of the remaining 7 bits), bringing the effective key size down to 56 bits. Furthermore, when DES is used in 40-bit mode, the two most significant bits of each byte in the key are zeroed.

## 9.4.6        Data stream processing

Security can be applied to various payload data streams selectively. The elementary unit is called a security context, which contains two session keys used for encrypting and decrypting a stream of payload data. Only one of the keys is used to process any particular payload unit. Each key can be used for processing both upstream and downstream payload data.

Having two keys allows negotiation of a new key to take place while payload data is processed using the old one, and then do an immediate switch-over once the new key is agreed upon, without interrupting payload traffic. The NCC initiates the key exchanges, and can start using a session key for downstream traffic encryption once the key exchange is complete. For upstream traffic encryption, the RCST should use whichever key was used by the NCC in the most recent encrypted payload unit.

### 9.4.6.1        Payload streams

A payload stream is identified by either of:

  • a 24-bit (UNI) ATM virtual circuit VPI/VCI;

  • a 48-bit MAC-address: this is used for DVB Multiprotocol Encapsulation downstream payload data.

When a payload stream is secured, the RCST and the NCC will have matching security contexts, which are used to encrypt/decrypt both upstream and downstream traffic. For unsecured payload streams there is no security context, and payload data is not encrypted.

To support encrypted multi-cast traffic, the same security context will be created for each member using EKE (see clause 9.4.4), so that each RCST can decrypt the common payload data stream.

## 9.4.6.2        Data encryption

Within a payload data stream, data is carried in individual units at the various protocol layers. Encryption is applied at the lowest layer possible, consistent with the payload stream:

- The unit of encryption is the payload field of an AAL5 CPCS-PDU. This payload is encrypted using the security context implied by the stream.

- DVB Multiprotocol Encapsulation payload streams: the unit of encryption is a single DVB Multiprotocol Encapsulation section. The datagram_data_bytes (between the MAC-address and the CRC/checksum) are encrypted using the security context implied by the stream. As defined in clause 9.4.1.5, the DVB Multiprotocol Encapsulation payload to be encrypted will be adjusted to have a length of $n \times 8$ bytes (n is an integer) for the DES algorithm, $n \times 16$ bytes for the AES algorithm by adding an appropriate amount of stuffing bytes before the CRC/checksum according to EN 300 468 [3], annex B after processing the entire section. The CRC/checksum is calculated on the encrypted datagram bytes, while higher-level protocol layers see only unencrypted datagrams.

## 9.4.6.3        Encryption flags

There are flags in the header of each encryption unit specifying which of the two sessions keys of the security context is used.

The receiver will use the security context of the payload stream to see if decryption shall be done.

AAL5 payload: Bit 0 and 1 of the CPCS-UU field contained within the CPCS-PDU Trailer of any AAL5 CPCS-PDU is used to convey security information as illustrated below.
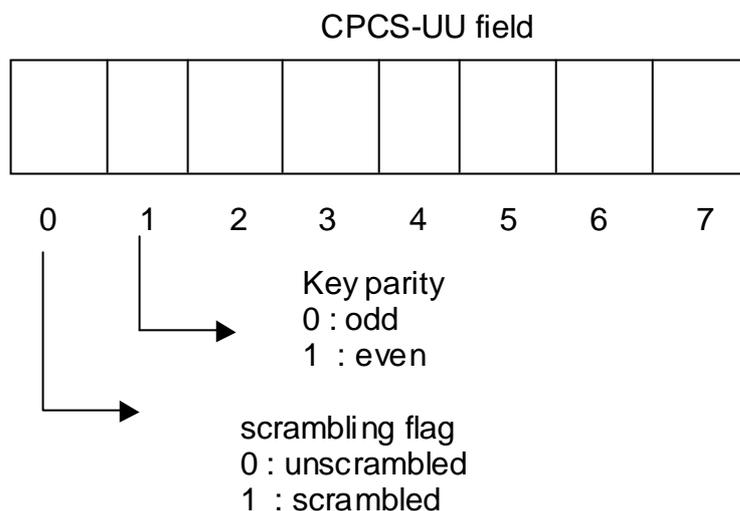


**Figure 43: CPCS-UU field**

- DVB Multiprotocol Encapsulation sections, according to EN 301 192 [4], the 2-bit payload_scrambling_control field in the section header is used:

    - 00: not encrypted;

    - 01: reserved;

    - 10: encrypted using session key 0;

- 11: encrypted using session key 1.

The 2-bit address_scrambling_control field in the section header is 00 all the time (the address is not scrambled).

# 9.4.7 Security establishment

Security control operations enforce the authentication service and prepare the NCC and the RCST to handle encryption by allowing them to agree on the streams to be encrypted as well as on the algorithms and keys to be used. To do so, NCC and RCST shall handle some security messages and respect a security signalling protocol.

Globally, the control operations are divided in the security connection of terminals and the re-keying of cryptographic contexts.

The security connection procedure occurs after the RCST passed the DVB-RCS logon, and allows the NCC and the RCST to:

- Negotiate security parameters; this is the negotiation step of the security connection procedure.

- Agree on initial session keys and authenticate the RCST; this is the key exchange step of the security connection procedure.

Re-keying of cryptographic contexts is periodically launched by the NCC which updates the session keys used in the interactive network in order to enhance the protection against cryptanalyze based attacks.

As mentioned before, the DVB-RCS security mechanisms work on data streams and cryptographic contexts. A data stream is identified either by a MPE MAC address or by an ATM VPI/VCI couple, and cryptographic contexts are identified by KeyIDs;

KeyIDs are a numbering of the cryptographic contexts, this numbering being local to a given RCST. The KeyID 0 (zero) is reserved for the unicast cryptographic contexts when the other values are reserved for the multicast streams.

The mapping of data streams on cryptographic contexts is done during the negotiation step of the security connection procedure, while key agreement for cryptographic contexts is done during the key exchange step or when proceeding to a re-keying procedure.

The authenticity of the RCST can be checked by the NCC during the key exchange step and during a re-keying procedure.

**Security connection of a RCST**

From the beginning here is how the security issues are handled at connection time:

- When an RCST registers on the network it will first go through the normal logon procedure as described in clause 7. It can then do negotiation handshakes with the NCC to defined the protected streams and to establish the level of security support, in particular the cryptographic algorithms and key sizes to be used subsequently. During logon, the RCST indicates its encryption/security capability to the NCC on the CSC burst (see clause 6.2.3) and the NCC communicates in the logon initialize descriptor (see clause 8.5.5.10.4) or mesh logon initialize descriptor (see clause 8.5.5.10.22) whether security shall be used.

- If security is used, the NCC then starts the security connection procedure with the negotiation step by sending as many Sign On request messages as the data streams required by the RCST. For Sign On request messages related to multicast forward MPE streams, the NCC selects only one encryption algorithm in its proposal, as the cryptographic context will be shared by all the RCSTs of the network.
The RCST shall then reply to each message issuing a Sign On response message including its choices.

A failure during the negotiation step causes the RCST to be logged off from the DVB-RCS network.

- Once the negotiation step is completed, the NCC shall proceed with the key exchange step by sending EKE request messages for each cryptographic context handled in the previous step. The NCC can request for the RCST to increment its clone counter. The RCST replies to each EKE request message with an EKE response message including its clone counter as well as an "authenticator" value. When receiving an EKE response message, the NCC can authenticate the terminal based on the clone counter and the "Authenticator" value. If the RCST has to increment its clone counter, it transmits the value of the clone counter before incrementation. A failure during the key exchange step of the connection causes the RCST to be logged off from the DVB-RCS network.

**Re-keying**

The NCC can start a re-keying procedure to update the session key of a cryptographic context. To transmit a new key, the NCC shall send EKE request messages, as for the key exchange step of the connection operation. For multicast cryptographic contexts, the NCC shall transmit an EKE request message to each concerned RCST.

The RCSTs shall extract session keys from incoming EKE request messages and reply with EKE response messages. Clone counter values cannot be updated during re-keying operations. While a cryptographic context is being updated, the old key shall still be used. For downlink streams, the gateway can start ciphering with the new key only upon reception of the EKE response message (or all the EKE response messages for a multicast cryptographic context). For an uplink stream, the ST can switch to the new key as soon as it receives the EKE request message.

A failure during this stage of the protocol causes the RCST to be logged off.

## 9.4.8    Persistent state variables

To facilitate authentication, key exchange, and clone detection, the RCST has a set of state variables whose values are retained across registrations and power cycles.

**Table 88: Persistent RCST variables**

| Name | Function | Size |
|------|----------|------|
| Cookie | authentication cookie | 160 bits |
| Cookie_SN | cookie sequence number | 1 bit |
| Clone_Counter | clone detection counter | 8 bits |
| Clone_Counter_SN | clone counter sequence number | 1 bit |

The sequence numbers are used to ensure that the NCC and RCST can stay synchronized even in case the RCST drops off the net in the middle of a protocol exchange.

### 9.4.8.1    Guaranteed delivery

The NCC will ensure that a protocol exchange is complete before proceeding. If it does not receive a response MAC message within a given time-interval, it will re-transmit the original message unchanged. The RCST will do likewise in situations where it requires a response. If the number of re-transmissions exceeds three, the protocol fails.

Due to race conditions, superfluous re-transmissions may be generated by both NCC and RCST. They shall discard such messages after the first message has in fact been received.

If the RCST is not ready to respond within the specified time-out, it can send **<MAC>Wait** messages (see clause 9.4.9.9) to extend the time it has available to generate a proper response. Upon receiving the wait message, the NCC will restart its timer and reset the retry count.

The protocol time-out values can be set by the <**MAC**> **Default Configuration Message** as defined in clause 5.5.3.2 of [11], otherwise the following default values apply.

**Table 89: Protocol time-out values**

| Code | Protocol stage | Default Value |
|------|----------------|---------------|
| 0xD | Security Sign-On | 700 |
| 0xE | Main Key Exchange | 1 200 |
| 0xF | Quick Key Exchange<br>Explicit Key Exchange | 900 |

The Unit for the timeouts is ms.

## 9.4.9    Security MAC messages

The following presents the security messages defined in the DVB-RCS framework. Two pairs of request/response messages are defined, one aiming at negotiating security parameters, the other allowing to agree on session keys as well as to authenticate RCSTs.

The following table details the roles played by each pair of messages.

**Table 90: Security MAC messages roles**

| Pair of messages | Role |
|------------------|------|
| **Sign On request** and<br>**Sign On response** | At the negotiation step of the connection procedure:<br>Allows to associate cryptographic contexts to data streams.<br>Allows to negotiate algorithms and parameters. |
| **EKE request** and<br>**EKE response** | At the key exchange step of the connection procedure and during re-keying procedures:<br>Allows the NCC to provide the RCSTs with session keys.<br>Allows the NCC to authenticate RCSTs. |

Request messages are sent by the NCC to the RCSTs and response messages are sent by the RCSTs to the NCC.

### 9.4.9.1      <MAC>Security Sign-On

As part of the registration process when an RCST performs the logon, the NCC and RCST will negotiate the specific set of cryptographic algorithms and parameters used in the key exchange protocols and for payload encryption.

The selections are global, and apply to all subsequent security exchanges for as long as the RCST is registered on the network.

The selections affect the layout of the subsequent key exchange messages, since they have fields that vary in size according to the choice of algorithms and parameters.

The NCC indicates which algorithms and parameters it supports by setting the appropriate bits in the **<MAC>Security Sign-On** message. There are four classes of algorithms, and the NCC will set one or more bits in each of the four fields to indicate which specific choices it supports.

The Sign On request message is presented in table 91. One Sign On request message associates a cryptographic context, indicated by the **Key_Id** field, to a data stream indicated by the **Flow_Id** field.

Negotiation applies to the specified keyId and involves four classes: public key algorithm (used for the Main Key Exchange), hashing algorithms, encryption algorithms, and Nonce sizes. The Sign On message carries the propositions made by the NCC to the RCST in the fields Public_Key_Alg, Hash_Alg, Encryption_Alg and Nonce_Size as follow: for each class, the NCC proposes to the RCST a possible value by setting the corresponding bit to "1".

**Table 91: Security Sign-On message structure**

| Security_Sign-On (){ | Bits | Bytes | Bit Number/Description | Parameter bytes |
|---|---|---|---|---|
| **Public_Key_Alg** | | 1 | **Public key algorithm choices:** | $P_{pka}$: |
| PKA_Reserved | 7 | | 7..1: Reserved, shall be 0 | 64 |
| PKA_DH_512 | 1 | | 0:(yes/no) Diffie-Hellman, 512 bits | |
| **Hash_Alg** | | 1 | **Hash algorithm choices:** | $P_{ha}$: |
| HA_Reserved | 7 | | 7..1: Reserved, shall be 0 | 20 |
| HA_HMACSHA1 | 1 | | 0:(yes/no) HMAC-SHA1 | |
| **Encryption_Alg** | | 1 | **Encryption algorithm choices:** | $P_{ea}$: |
| EA_Reserved | 4 | | 7..4: Reserved, shall be 0 | |
| EA_AES_CTR_128 | 1 | | 3: AES in counter mode, 128 bits key | 24 |
| EA_AES_CBC_IV_128 | 1 | | 2: AES in CBC mode, with a random IV, | 16 |
| EA_DES_56 | 1 | | 128 bits key | |
| EA_DES_40 | 1 | | 1:(yes/no) DES, 56 bit key | 8 |
| | | | 0:(yes/no) DES, 40 bit key | |
| **Nonce_Size** | | 1 | **Nonce size choices:** | $P_{ns}$: |
| NS_Reserved NS_64 | 7 | | 7..1: Reserved, shall be 0 | 8 |
| NS_64 | 1 | | 0: (yes/no) 8 random bytes | |
| **Security_Ctxt_Version_Flow** | | 4 | | |
| | 1 | | 32: the version of the security protocol. Shall be set to zero. | |
| Id_Type | 1 | | 31: The type of stream designated by the message. If "Id_Type" is 0, then the 23 lowest bits of Flow_Id are the 23 lowest bits of the stream's MAC address. Else Flow_Id is an ATM VPI/VCI couple. | |
| Flow_Id | 24 | | 7..30: identify the data stream | |
| Key_Id | 6 | | 1..6: the keyID of the cryptographic context | |
| } | | | | |

If the security option is supported, the minimum subset to support is PKA_DH_512, HA_HMACSHA1, EA_DES_40, and NS_64. EA_DES_56 is optional.

The Sign On request message indicates the data stream with the **Flow_Id** field. The meaning of this field depends on the value of Id_Type:

If Id_Type is set to "0", the data stream is a MPE stream, and Flow_Id is set as follow:

- If the stream is a the unicast stream of the RCST, Flow_Id is set to zero, as well as Key_Id.

- If the stream is a multicast MPE stream, the 23 lowest bits of Flow_Id are the 23 lowest bits of the stream's MAC address, as specified in RFC 1112 [i.6].

If Id_Type is set to "1", the data stream is an ATM PVC, and Flow_Id is the VPI/VCI of the PVC as follows:

| VPI (8 bits) | VCI (16 bits) |
|---|---|

**Figure 44: Flow_Id when Id_Type is set to "1"**

## 9.4.9.2     <MAC>Security Sign-On Response

In its security sign-on response, the RCST indicates which specific algorithms and parameters to use. It does so by choosing one of the suggestions offered by the NCC within each of the four classes.

The fields of the response message have the same definition as the message from the NCC, except that exactly one bit will be set in each field.

If the RCST is unable to support any of the suggested algorithms for any class, it shall return an all-zero field value, and the NCC will revert to non-secure communication or re-issue the **<MAC>Security Sign-On** message with different choices. In this case, the NCC will logoff the RCST.

**Table 92: Security Sign-On Response message structure**

| Security_Sign-On_Response() { | Bits | Bytes | Bit Number/Description | Parameter bytes |
|---|---|---|---|---|
| **Public_Key_Alg** | | **1** | **Public key algorithm choices:** | $P_{pka}$: |
| PKA_Reserved | 7 | | 7..1: Reserved, shall be 0 | 64 |
| PKA_DH_512 | 1 | | 0:(yes/no) Diffie-Hellman, 512 bits | |
| **Hash_Alg** | | **1** | **Hash algorithm choices:** | $P_{ha}$: |
| HA_Reserved | 7 | | 7..1: Reserved, shall be 0 | 20 |
| HA_HMACSHA1 | 1 | | 0:(yes/no) HMAC-SHA1 | |
| **Encryption_Alg** | | **1** | **Encryption algorithm choices:** | $P_{ea}$: |
| EA_Reserved | 4 | | 7..4: Reserved, shall be 0 | |
| EA_AES_CTR_128 | 1 | | 3: AES in counter mode, 128 bits key | 24 |
| EA_AES_CBC_IV_128 | 1 | | 2: AES in CBC mode, with a random IV, 128 | 16 |
| EA_DES_56 | 1 | | bits key | |
| EA_DES_40 | 1 | | 1:(yes/no) DES, 56 bit key | 8 |
| | | | 0:(yes/no) DES, 40 bit key | |
| **Nonce_Size** | | **1** | **Nonce size choices:** | $P_{ns}$: |
| NS_Reserved | 7 | | 7..1: Reserved, shall be 0 | 8 |
| NS_64 | 1 | | 0: (yes/no) 8 random bytes | |
| **Security_Ctxt_Version_Flow** | | 4 | | |
| | 1 | | 32: the version of the security protocol. Shall be set to zero. | |
| Id_Type | 1 | | 31: The type of stream designated by the message. If "Id_Type" is 0, then the 23 lowest bits of Flow_Id are the 23 lowest bits of the stream's MAC address. Else Flow_Id is an ATM VPI/VCI couple. | |
| Flow_Id | 24 | | 7..30: identify the data stream | |
| Key_Id | 6 | | 1..6: the keyID of the cryptographic context | |
| } | | | | |

### 9.4.9.3        <MAC>Main Key Exchange

The Main Key Exchange message is used to start a cookie-independent key exchange with the RCST, and also instructs the RCST whether to update its cookie value and clone counter value. The connection_id is not used and shall be set to all "0".

**Table 93: Main Key Exchange message structure**

| Main_Key_Exchange () { | Bits | Bytes | Bit Number/Description |
|---|---|---|---|
| **Connection_ID** | 32 | 4 | **MAC connection identifier** |
| **Flags** | | 1 | |
| Reserved | 4 | | 7..4: shall be 0 |
| FL_Initializing | 1 | | 3:(yes/no) first ever key exchange |
| FL_Update_Cookie | 1 | | 2:(yes/no) make new cookie value |
| FL_Update_Counter | 1 | | 1:(yes/no) increment clone counter |
| FL_Session_Key | 1 | | 0: select session key 0 or 1 |
| **Reserved** | 8 | 1 | **Reserved for future use, shall be 0** |
| **Nonce** | | $P_{ns}$ | **Random string** $nonce1$ |
| **DH_Modulus** | | $P_{pka}$ | **Diffie-Hellman modulus** m |
| **DH_Generator** | | $P_{pka}$ | **Diffie-Hellman generator** g |
| **DH_Public_X** | | $P_{pka}$ | **Diffie-Hellman public value** X |
| } | | | |

The FL_Session_Key bit specifies which session key of the security context to update.

If the FL_Update_Counter bit is set, it instructs the RCST to increment its clone detection counter.

If the FL_Update_Cookie bit is set, it instructs the RCST to generate a new cookie value to be used for future authentications and key exchanges, and to reset the clone detection counter to zero.

If the FL_Initializing bit is set, it tells the RCST that the Authenticator field in the response will be ignored.

The sizes of the multi-byte fields are determined by the parameters of the algorithms selected during security sign-on (see clause 9.4.9.1).

The NCC will use its own private Diffie-Hellman value, x, together with the fields of the response message from the RCST to derive the new session key value, as well as any new value for the cookie (see clause 9.4.2).

### 9.4.9.4        <MAC>Main Key Exchange Response

The Main Key Exchange Response message authenticates the RCST and completes the cookie-independent key exchange with the NCC. It also contains the current value of the clone detection counter. The connection_id is not used and shall be set to all "0".

**Table 94: Main Key Exchange Response message structure**

| Main_Key_Exchange_Response () { | Bits | Bytes | Bit Number/Description |
|---|---|---|---|
| **Connection_ID** | 32 | 4 | **MAC connection identifier** |
| **Flags** | | 1 | |
| Reserved FL_Cookie_SN | 6 | | 7..2: shall be 0 |
| FL_cookie_SN | 1 | | 1: cookie sequence number |
| FL_Counter_SN | 1 | | 0: clone counter sequence number |
| **Clone_Counter** | 8 | 1 | **Current clone counter value** |
| **Nonce** | | $P_{ns}$ | **Random string** $nonce2$ |
| **Authenticator** | | $P_{ha}$ | **Authentication value** auth |
| **DH_Public_Y** | | $P_{pka}$ | **Diffie-Hellman public value** Y |
| } | | | |

The `FL_Counter_SN` bit is the current sequence number of the clone detection counter. The **Clone_Counter** field is the current value of the counter. A clone collision has been detected if the NCC finds a mismatch from the expected value.

The `FL_Cookie_SN` bit is the sequence number of the cookie used for authentication.

If the `FL_Update_Cookie` bit was set by the NCC, the RCST will generate a new cookie value and complement the cookie sequence number bit. It will also reset the clone counter value to zero and clear the clone counter sequence number bit.

If the `FL_Update_Counter` bit was set by the NCC, the RCST will increment the value of the clone counter (modulo 256) and complement the clone counter sequence number bit.

Any updates to the cookie, clone counter, or their associated sequence number bits do not take effect, and shall not be committed to non-volatile storage, until the following <**MAC**>**Connect Confirm** message is received by the RCST.

The RCST uses its private Diffie-Hellman value, $y$, together with the message fields to derive the new session key value, as well as any new value for the cookie (see clause 9.4.2).

## 9.4.9.5      <MAC>Quick Key Exchange

The Quick Key Exchange message is used to start a cookie-dependent key exchange with the RCST, and also instructs the RCST whether to update its clone counter value. The connection_id is not used and shall be set to all "0".

**Table 95: Quick Key Exchange message structure**

| Quick_Key_Exchange () { | Bits | Bytes | Bit Number/Description |
|---|---|---|---|
| Connection_ID | 32 | 4 | MAC connection identifier |
| Flags | 8 | 1 | |
|    Reserved | 6 | | 7..2: shall be 0 |
|    FL_Update_Counter | 1 | | 1:(yes/no) increment clone counter |
|    FL_Session_Key | 1 | | 0: select session key 0 or 1 |
| Reserved | 8 | 1 | Reserved for future use, shall be 0 |
| Nonce | | $P_{ns}$ | Random string `nonce1` |
| } | | | |

The `FL_Session_Key` bit specifies which session key of the security context to update.

If the `FL_Update_Counter` bit is set, it instructs the RCST to increment its clone detection counter.

The NCC will use its knowledge of the cookie value together with the fields of the response message from the RCST to derive the session key value (see clause 9.4.3).

## 9.4.9.6      <MAC>Quick Key Exchange Response

The Quick Key Exchange Response message authenticates the RCST and completes the cookie-dependent key exchange with the NCC. It also contains the current value of the clone detection counter. The connection_id is not used and shall be set to all "0".

**Table 96: Quick Key Exchange Response message structure**

| Quick_Key_Exchange_Response () { | Bits | Bytes | Bit Number/Description |
|---|---|---|---|
| Connection_ID | 32 | 4 | MAC connection identifier |
| Flags | | 1 | |
|    Reserved | 6 | | 7..2: shall be 0 |
|    FL_Cookie_SN | 1 | | 1: cookie sequence number |
|    FL_Counter_SN | 1 | | 0: clone counter sequence number |
| Clone_Counter | 8 | 1 | Current clone counter value |
| Nonce | | $P_{ns}$ | Random string `nonce2` |
| Authenticator | | $P_{ha}$ | Authentication value `auth` |
| } | | | |

The `FL_Cookie_SN` bit is the sequence number of the cookie used for authentication.

The `FL_Counter_SN` bit is the current sequence number of the clone detection counter. The **Clone_Counter** field is the current value of the counter. A clone collision has been detected if the NCC finds a mismatch from the expected value.

If the `FL_Update_Counter` bit was set by the NCC, the RCST will increment the value of the clone counter (modulo 256) and complement the clone counter sequence number bit.

The RCST uses the cookie value together with the message fields to derive the session key value (see clause 9.4.3).

## 9.4.9.7        <MAC>Explicit Key Exchange

The Explicit Key Exchange message is used to securely deliver an existing session key value to the RCST, and also instructs the RCST whether to update its clone counter value.

Its layout, presented in the following table, is determined by the results of the negotiation step. This is indicated by parameters Pns and Pea.

The delivered session key applies to the cryptographic context specified by the `Key_Id` field.

The connection_id is not used and shall be set to all "0".

**Table 97: Explicit Key Exchange message structure**

| Explicit_Key_Exchange (){ | Bits | Bytes | Bit Number/Description |
|---|---|---|---|
| Connection_ID | 32 | 4 | MAC connection identifier |
| Key_Id | 6 | 1 | 2..7: The KeyID to which the message refers to |
| **Flags**<br>        FL_Update_Counter<br>    FL_Session_Key | <br>6<br>1<br>1 | **1** | <br>7..2: shall be 0<br>1:(yes/no) increment clone counter<br>0: select session key 0 or 1 |
| **Nonce** | | $P_{ns}$ | **Random string** nonce1 |
| **Encryptedkey** | | $P_{ea}$ | **Encrypted session key** |
| } | | | |

The `FL_Session_Key` bit specifies which session key of the security context to update.

If the `FL_Update_Counter` bit is set, it instructs the RCST to increment its clone detection counter.

The NCC has used its knowledge of the cookie value to encrypt the session key value (see clause 9.4.4).

According to the cipher algorithm and usage, additional information may also be part of the transmission of the `Encryptedkey`. This is the case when using an AES CTR scheme. Nonce part of the counter must precede the protected (encrypted) session key. As the same session key may be used when bi-directional ciphering is used, two nonces are conveyed with the session key; one to be used by the NCC when ciphering (and the RCST when deciphering), the other to be used by the RCST when ciphering (and the NCC when deciphering).

When key material is only used for uni-directional ciphering, the useless nonce information is set to 0.
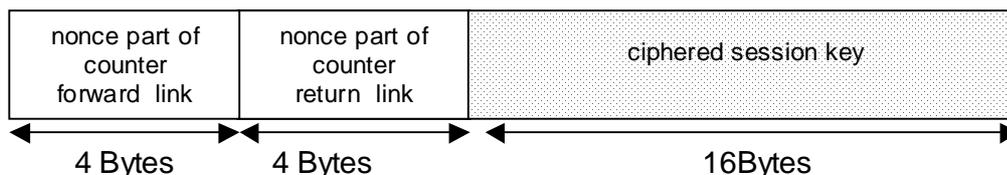


**Figure 45: "Encrypted key" field format when using AES-CTR**

### 9.4.9.8        <MAC>Explicit Key Exchange Response

The Explicit Key Exchange Response message authenticates the RCST and acknowledges receipt of the delivered key. It also contains the current value of the clone detection counter. The connection_id is not used and shall be set to all "0".

**Table 98: Explicit Key Exchange Response message structure**

| Explicit_Key_Exchange_Response () { | Bits | Bytes | Bit Number/Description |
|---|---|---|---|
| Connection_ID | 32 | 4 | MAC connection identifier |
| Key_Id | 6 | 1 | 2..7: The KeyID of the association to which the message refers to |
| Flags | | | |
| FL_Cookie_SN | 1 | | 1: cookie sequence number |
| FL_Counter_SN | 1 | | 0: clone counter sequence number |
| Clone_Counter | 8 | 1 | Current clone counter value |
| Nonce | | $P_{ns}$ | Random string nonce2 |
| Authenticator | | $P_{ha}$ | Authentication value auth |
| } | | | |

The FL_Cookie_SN bit is the sequence number of the cookie used for authentication and session key decryption. If the NCC determines that it has used the wrong cookie for session key encryption it will re-issue the **<MAC>Explicit Key Exchange** using the old cookie value.

The FL_Counter_SN bit is the current sequence number of the clone detection counter. The **Clone_Counter** field is the current value of the counter. A clone collision has been detected if the NCC finds a mismatch from the expected value.

If the FL_Update_Counter bit was set by the NCC, the RCST will increment the value of the clone counter (modulo 256) and complement the clone counter sequence number bit.

The RCST uses the cookie value together with the message fields to decrypt the session key value (see clause 9.4.4).

### 9.4.9.9        <MAC>Wait

The Wait message is used by the RCST to extend the time the NCC waits for a reply to a given message. Upon receiving it, the NCC will reset its time-out value and retry count (see clause 9.4.8.1).

**Table 99: Wait message structure**

| Wait () { | Bits | Bytes | Bit Number/Description |
|---|---|---|---|
| Connection_ID | 32 | 4 | MAC connection identifier |
| Message_Type | 8 | 1 | Type of message from NCC |
| Reserved | 8 | 1 | Reserved for future use, shall be 0 |
| } | | | |

The **Message_Type** field is the message type value of the message received from the NCC being processed. The connection_id is not used and shall be set to all "0". The RCST indicates that it is currently unable to send a reply to the message.

## 9.5        Transport of security messages (optional)

The MAC security messages transmitted over the air interface can be transported:

- Either using DULM (see clause 6.6.2) over the return path and using a dedicated and well-known PID over the forward path. "Security enhanced" RCSTs will thus have to MAC-filter this PID.

- Or using the same IP communication stack carrying the user and management traffic in the DVB-RCS network as shown below.

| UDP |  |
|---|---|
| IP |  |
| AAL5 | MPE |
| ATM |  |
| MPEG2 |  |
| DVB-S |  |

| UDP |  |
|---|---|
| IP |  |
| AAL5 | MPE |
| ATM | MPEG2 |
| DVB-RCS |  |

Forward link                                        Return link
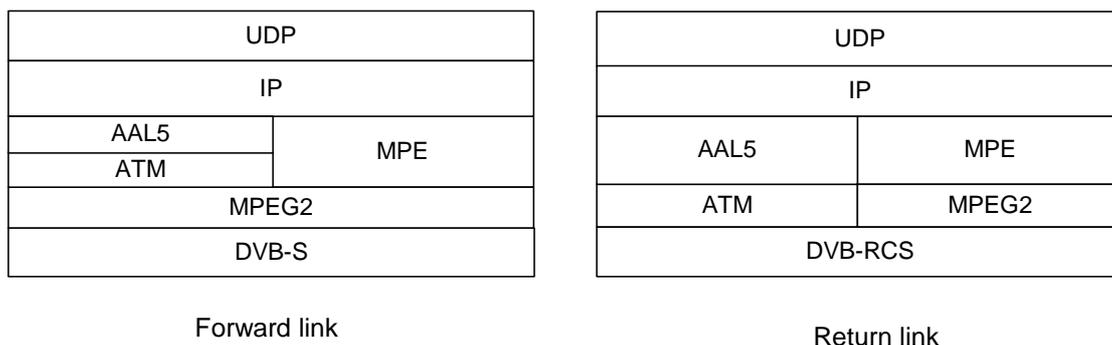
**Figure 46: IP protocol stack for transporting security messages**

Security messages are inserted in UDP datagrams with TLV descriptors, with the possibility to have several security messages per datagram as follows.
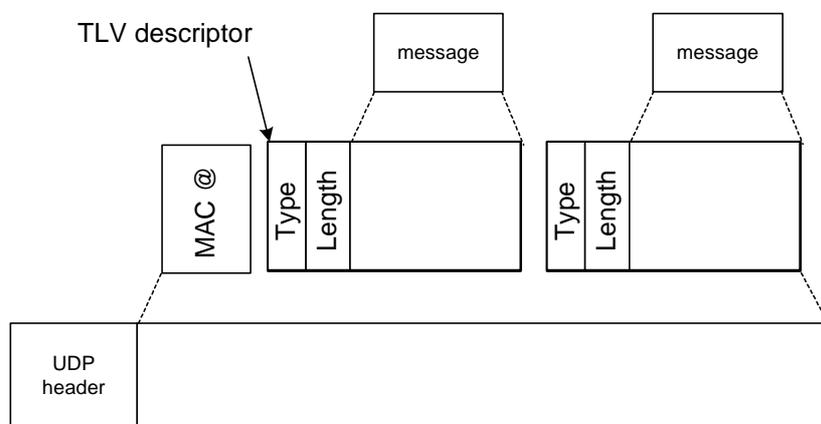


**Figure 47: Several security messages per datagram**

For message transfers from a RCST to the NCC, the RCST's MAC address shall be inserted at the beginning of the UDP payload, as illustrated in figure 100. This MAC address allows the NCC to know from which RCST the messages are coming.

TLV descriptors allow to identify the type of the security messages with the following syntax.

**Table 100: TLV descriptor structure**

| Security_TLV_descriptor { | Bytes | Parameter |
|---|---|---|
| Type | 1 |  |
| Length | 1 | Mlen |
| If (Type == 0x0d) {<br>    EKE message<br>} | Mlen |  |
| if (Type == 0x35) {<br>    Sign On message<br>} | Mlen |  |

In the table above, the **EKE message** field can be either an EKE request message or an EKE response message, depending which entity sends it (NCC or RCST). The same remark applies to the **Sign On message** field.

The **Length** field is the length in bytes of the **EKE message** or **Sign On message** field.

# 10       Continuous carrier operation (Mobile Opt)

The RCST can optionally employ a continuous carrier mode of transmission in accordance with the provisions in this clause. The ability to operate in this manner shall be signalled in the CSC burst. An RCST declaring support for basic continuous carrier operation shall be capable of transmitting either a continuous carrier or an MF-TDMA signal, but will normally not be able to transmit both carriers simultaneously. An RCST declaring support for enhanced continuous carrier operation shall be able to transmit both types of signal simultaneously.

## 10.1      Modes of operation

Steady-state operation of the RCST shall be characterised by one of four operational modes:

- Off/Standby: As defined for MF-TDMA operation.

- Fine Sync: As defined for MF-TDMA operation.

- Continuous: Operation with a continuous carrier only.

- Continuous + Fine Sync (Optional): Simultaneous operation of continuous carrier and MF-TDMA.

Irrespective of the presence of continuous carriers, all MF-TDMA transmissions shall be in accordance with the provisions of clause 6.

Support for the "Continuous + Fine Sync" mode is optional. The operational states and the events that define transitions between them are illustrated in figure 48 for the basic RCST and in figure 49 for the enhanced RCST that supports simultaneous MF-TDMA and continuous operation.
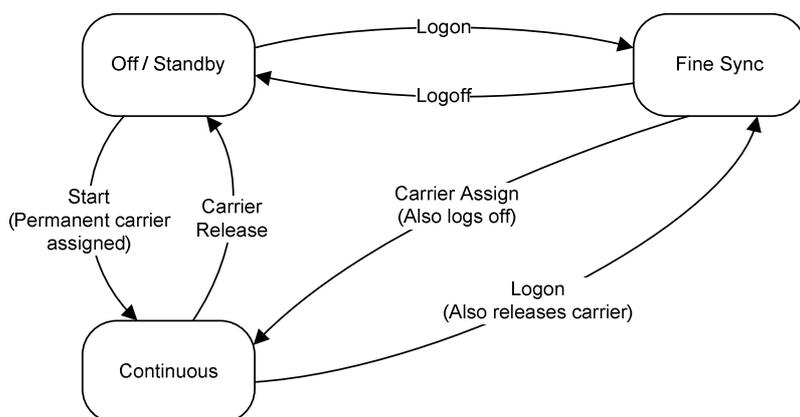


**Figure 48: State diagram for basic continuous carrier operation**
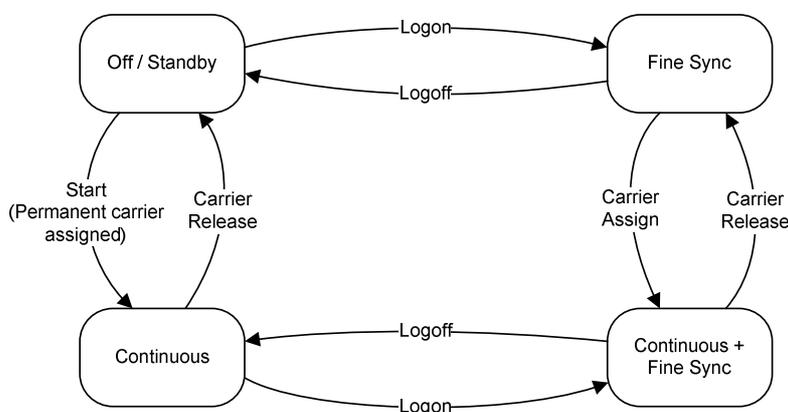


**Figure 49: State diagram for enhanced continuous carrier operation**

The following transitions apply to both types of RCST:

- From the Off/Standby state, successful execution of a logon process results in the RCST being in fine Sync state (MF-TDMA operation).

- From the Fine Sync state, successful execution of a logoff process results in the RCST being in the Off/Standby state.

- From the Off/Standby state, successful initialisation of an RCST with a permanently-assigned carrier results in the RCST being in the Continuous state. This transition shall not rely on reception of the forward link signalling defining the carrier plan.

- From the Continuous state, release of all assigned carriers results in the RCST being in the Off/Standby state.

The following transitions apply to a basic RCST without support for the Continuous + Fine Sync mode:

- From the Fine Sync state, successful assignment of a continuous carrier results in the RCST being logged off and in the Continuous state.

- From the Continuous state, successful execution of a logon process results in release of the assigned carrier and the RCST being in Fine Sync state (MF-TDMA operation).

The following transitions apply to an enhanced RCST with support for the Continuous + Fine Sync mode:

- From the Fine Sync state, successful assignment of a continuous carrier results in the RCST being in the Continuous + Fine Sync state.

- From the Continuous + Fine Sync state, release of all assigned carriers results in the RCST being in the Fine Sync state (MF-TDMA operation).

- From the Continuous + Fine Sync state, successful execution of a logoff process results in the RCST being in the Continuous state.

- From the Continuous state, successful execution of a logon process results in the RCST being in Fine Sync state (MF-TDMA operation).

The logon and logoff processes shall be as defined for MF-TDMA operation in clause 7. Assignment and release of continuous carriers are defined in this clause.

## 10.2    Forward link

The forward link for operation in Continuous and Continuous + Fine Sync modes shall be as defined in clause 5, possibly modified for mobile operation as defined in the present document.

Forward Link signalling associated with continuous carrier return link operation shall be carried in accordance with the provisions of clause 8.3.

## 10.3    RCST synchronisation

Carrier Synchronisation in Continuous and Continuous + Fine Sync modes shall be as defined in clause 6.1.2.

## 10.4    Return link

The baseline transmission scheme for the continuous return link carriers is DVB-S2 [16]. The present document shall be used in accordance with the provisions in the following sub-clauses. The control mechanisms additionally allow management of return link carriers operating using proprietary transmission schemes. Properties of such carriers are not addressed.

## 10.4.1    Modulation and coding

### 10.4.1.1    Carrier operation

Return link DVB-S2 carriers can be operated in CCM mode. The RCST shall support the mandatory MODCOD and frame size combinations defined for the Broadcast Services profile in [16].

Alternatively, return link DVB-S2 carriers can be operated in ACM mode. The RCST can signal its ability to control the transmission mode by setting the Continuous_ACM flag in the CSC burst. When this flag is set to zero, the RCST is capable of selecting a suitable transmission mode for the return link traffic when operating in continuous mode. The RCST may then select another transmission mode than indicated in the TBTP from within the range given by the Return Transmission Modes descriptor. When this flag is set to one, or if no applicable Return Transmission Modes descriptor is received, the RCST shall use the transmission mode indicated by the TBTP for all continuous mode return link traffic on the corresponding carrier.

NOTE:    VCM/ACM operation completely under the NCC's control can be achieved by defining multiple continuous carrier types in the TCT that differ only in the modulation/coding parameters and correspondingly defining multiple continuous carriers in the FCT that occupy the same frequency space but have different modulation/coding parameters. Change of the modulation/coding parameters is carried out through the TBTP, using the assignment/release mechanism for continuous carriers. It is the responsibility of the NCC to ensure that duplicate assignments of the same frequency space are avoided.

The characteristics of the continuous return link carriers supported in the system shall be signalled through the timeslot composition table, in which dedicated "timeslot types" represent continuous carriers. It is the responsibility of the NCC not to assign such timeslots to RCSTs that do not support continuous carrier operation.

ACM mode shall not be used in combination with $\pi/2$-BPSK modulation and rate-1/4 coding.

### 10.4.1.2    Bit mapping to π/2-BPSK constellation

A continuous return link carrier can be transmitted using $\pi/2$-BPSK modulation. When this is used, it shall employ absolute encoding (no differential coding). The bit-to-symbol mapping shall follow figure 50. Values from sub-constellations b1 and b2 are alternately transmitted, starting with the MSB of the BBHEADER, which shall be transmitted as b1. The normalised average energy per symbol shall be equal to 1.
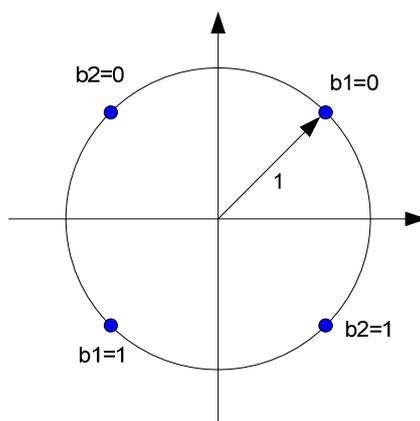


**Figure 50: Bit mapping to π/2-BPSK constellation for continuous return link carrier**

### 10.4.1.3    Physical layer scrambling

The physical-layer scrambling sequence to be used for the DVB-S2 return link continuous carrier transmisions can be signalled through the use of the Return Transmission Modes descriptor. If the Return Transmission Modes descriptor is not used for the applicable superframe, the default physical layer scrambling sequence shall be used.

### 10.4.1.4        Physical layer header

The SOF field of the physical layer header shall be in accordance with clause 5.5.2.1 of [16].

The MODCOD field of the physical layer header shall be coded in accordance with clause 5.5.2.2 of [16], except when π/2-BPSK modulation is used. In that case, the MODCOD value shall be set to "11111".

The TYPE field of the physical layer header shall be in accordance with clause 5.5.2.3 of [16], except when very short frames are used, in which case the TYPE field shall be set to "00". Pilots shall always be used when very short frames are employed.

### 10.4.1.5        Very short frame size

A return link continuous transmission can use a very short frame size by employing the provisions in this clause. This very short frame is an extension of DVB-S2 and shall follow the provisions of [16] except as stated in this clause. Signalling for definition of carriers using very short frames is carried in the Timeslot Composition Table (see clause 8.5.5.4).

Mode adaptation and stream adaptation shall be in accordance with clauses 5.1 and 5.2 of [16] respectively, except that the values of $K_{BCH}$ shall be those given in annex C. FEC encoding shall be in accordance with annex C.

Bit interleaving as defined in [16] shall not be applied.

Bit mapping into the modulation constellation shall be in accordance with clause 5.4 of [16] for QPSK, 8PSK, 16APSK and 32APSK and in accordance with clause 10.4.1.2 for π/2-BPSK. For 8PSK, 2 stuffing bits shall be appended to the end of the FECFRAME in order to obtain an integer number of symbols. For 32APSK, 4 such stuffing bits shall be added. No stuffing bits shall be added for π/2-BPSK, QPSK and 16APSK.

Physical layer framing shall be in accordance with clause 5.5 of [16], except that the last slot in each frame shall contain $k$ information-carrying symbols, so that the total number of such symbols in the frame, i.e. excluding the physical layer header and pilots, is $90 \times S + k$. The values of $S$ and $k$ are given in table 101. The remaining $90-k$ symbols in the last slot of each frame shall be filled with pilot symbols. Each pilot shall be an unmodulated symbol with the value $I=Q=(1/\sqrt{2})$. The total number of symbols in each frame, including physical layer header and pilots, thus becomes $90 \times (S+2) + P \times \lfloor (S-1)/16 \rfloor$, where $P=36$ and $\lfloor x \rfloor$ denotes the largest integer not greater than $x$.

**Table 101: Number of slots for very short XFECFRAMES**

| Modulation | S | k |
|---|---|---|
| π/2-BPSK | 45 | 46 |
| QPSK | 22 | 68 |
| 8PSK | 15 | 16 |
| 16APSK | 11 | 34 |
| 32APSK | 9 | 10 |
| NOTE: S is the number of complete 90-symbol slots in the frame that carry information symbols. | | |

The physical layer header shall be in accordance with clause 10.4.1.4. Physical layer scrambling shall be in accordance with clause 10.4.1.3. Pulse shaping and modulation shall be in accordance with [16].

## 10.4.2      Spectrum spreading for continuous return link carriers

A return link continuous transmission can be spread in bandwidth using the provisions in this clause. Such spreading is applied in two stages: spreading and scrambling. The first operation, spreading, multiplies every $(I+jQ)$ symbol by a sequence of chips to enlarge the bandwidth of the signal. The number of chips per symbol is called the Spreading Factor ($SF$). When $SF = 1$, the transmission is equivalent to a conventional DVB-S2 signal.

The second operation, scrambling, applies a scrambling code to the spread signal. The processing is similar to that applied to spread DVB-S2 forward links and is illustrated in figure 5.

Spreading shall be applied on a PLFRAME basis. Each symbol in a PLFRAME, including the PLHEADER and pilot symbols if used, shall be spread by a repetition of a real-valued spreading code $C(i)$. $C(i) = 1$ for $0 \leq i < SF$, so the output of the spreading for each symbol on the $I$ and $Q$ branches is a sequence of $SF$ repetitions of the original symbol value.

If $\{d[k]\}$, $k = 0, 1, \ldots, N_{PLFRAME}-1$, represents the $(I+jQ)$ symbols of the PLFRAME, where $N_{PLFRAME}$ is the number of symbols in one PLFRAME, then the spreading operation yields the spread sequence $s(i)$:

$$s(i) = d\left(\lfloor i / SF \rfloor\right) \quad \text{for } i = 0, 1,\ldots, (N_{PLFRAME} \times SF)\text{-}1$$

The spreading factor is signalled in the Timeslot Composition Table (clause 8.5.4).

In terms of the reference modulator signal flow defined in DVB-S2 [16], the spreading shall be performed immediately prior to the physical layer scrambling.

The second operation, scrambling, shall replace that defined for physical layer scrambling in clause 5.5.4 of [16] and shall be achieved through the use of the same method, except that the length of the scrambling sequence is here equal to $N_{PLFRAME} \times SF$, rather than $N_{PLFRAME}$. The scrambling sequence shall be aligned with the PLFRAME epoch, and it shall be re-initialized at the beginning of each PLFRAME.

The sequence of complex valued chips shall be scrambled (complex chip-wise multiplication) by the complex-valued scrambling code, $w(i)$, defined in [16], clause 5.5.4, when $SF$ is greater than 1. The Spread PLFRAME duration depends on the selected modulation and the adopted spreading factor. The scrambled symbols, $z(i)$, shall be obtained by directly multiplying the spread symbols, $s(i)$, by the scrambling sequence, $w(i)$, as follows:

$$z(i) = s(i) \times w(i \text{ modulo } 66\,420), \quad i=0,1,2,\ldots,(N_{PLFRAME} \times SF)\text{-}1$$

After scrambling, the signal $\{z(i)\}$ shall be square root raised cosine filtered as described in [16], clause 5.6.

## 10.4.3    Stream format and data encapsulation

The stream format of the continuous carrier shall be a single, packetized generic stream.

Packetizations using the ATM profile shall be supported. The VPI/VCI values to be used shall be signalled to the RCST using unicast signalling.

Packetisation using the MPEG profile shall be supported for RCSTs that support this profile in MF-TDMA mode. The PID values to be used shall be signalled to the RCST using unicast signalling. The sync byte value shall be 0x47 for all packets.

   NOTE:    Although the stream is reminiscent of a transport stream when the MPEG profile is employed, it lacks certain properties of transport streams and is therefore still characterised as a generic stream.

## 10.4.4    Return link signalling

Return link signalling specific to the continuous-carrier operation shall be carried using DULM, using the assigned signalling VPI/VCI or PID. When the MPEG profile is used, the logon_ID and group_ID contained in the DULM message shall be ignored.

The RCST shall issue carrier release requests for all non-permanently assigned continuous carriers prior to ceasing operation, unless prevented from doing so by an overriding cause such as equipment malfunction or power interruption.

The Continuous Carrier Control Information Element (CCC-IE) shall be used to convey the related return link signalling. The format of this information element is given in table 102.

**Table 102: Continuous carrier control information element**

| Syntax | No. of bits | | Information Mnemonic |
|---|---|---|---|
| | Reserved (see note) | Information | |
| Continuous_carrier_control_element() { | | | |
|    message_type | | 8 | uimsbf |
|    if (message_type == 0) { | | | |
|       assign_carrier_id | 8 | 8 | uimsbf |
|    } | | | |
|    else if (message_type == 1) { | | | |
|       release_carrier_superframe | | 8 | uimsbf |
|       release_carrier_frame | 3 | 5 | uimsbf |
|       release_carrier_timeslot | 5 | 11 | uimsbf |
|    } | | | |
|    else if (message_type == 2) { | | | |
|       replace_carrier_superframe | | 8 | uimsbf |
|       replace_carrier_frame | 3 | 5 | uimsbf |
|       replace_carrier_timeslot | 5 | 11 | uimsbf |
|       replace_carrier_id | | 8 | uimsbf |
|    } | | | |
|    else if (message_type == 3) { | | | |
|       permanent_assign_carrier_id | 8 | 8 | uimsbf |
|    } | | | |
|    else if (message_type == 4) { | | | |
|       permanent_release_carrier_superframe | | 8 | uimsbf |
|       permanent_release_carrier_frame | 3 | 5 | uimsbf |
|       permanent_release_carrier_timeslot | 5 | 11 | uimsbf |
|    } | | | |
|    else if (message_type == 5) { | | | |
|       assign_accept | 7 | 1 | bslbf |
|       assign_carrier_superframe | | 8 | uimsbf |
|       assign_carrier_frame | 3 | 5 | uimsbf |
|       assign_carrier_timeslot | 5 | 11 | uimsbf |
|    } | | | |
| } | | | |
| NOTE: Reserved bits are of type bslbf, and shall precede the Information bits on the same line. They shall be ignored by the RCST. For an encrypted uni-cast TIM, the bit values shall be varied in a random manner to avoid encryption spoofing. | | | |

Semantics for the Continuous carrier control information element:

- message_type: This 8-bit field defines the type of message contained in the information element, in accordance with table 103.

**Table 103: Continuous carrier message types**

| Value | Message |
|---|---|
| 0x00 | Request carrier assignment |
| 0x01 | Release assigned carrier |
| 0x02 | Request replacement carrier |
| 0x03 | Request permanent carrier assignment |
| 0x04 | Request release of permanent carrier |
| 0x05 | Carrier assignment acknowledgement |
| 0x06 - 0x7F | Reserved |
| 0x80 - 0xFF | User defined |

- assign_carrier_id: This 8-bit field defines the type (timeslot_id) of carrier requested.

- release_carrier_superframe: This 8-bit field defines the superframe_id of the superframe containing the carrier that is requested to be released.

- release_carrier_frame: This 5-bit field defines the frame number within the superframe of the frame containing the carrier that is requested to be released.

- release_carrier_timeslot: This 11-bit field defines the timeslot number within the frame defining the carrier that is requested to be released.

- replace_carrier_superframe: This 8-bit field defines the superframe_id of the superframe containing the carrier that is requested to be replaced.

- replace_carrier_frame: This 5-bit field defines the frame number within the superframe of the frame containing the carrier that is requested to be replaced.

- replace_carrier_timeslot: This 11-bit field defines the timeslot number within the frame defining the carrier that is requested to be replaced.

- replace_carrier_id: This 8-bit field defines the type (timeslot_id) of the carrier requested as replacement.

- permanent_assign_carrier_id: This 8-bit field defines the type (timeslot_id) of carrier requested for permanent assignment.

- permanent_release_carrier_superframe: This 8-bit field defines the superframe_id of the superframe containing the permanently assigned carrier that is requested to be released.

- permanent_release_carrier_frame: This 5-bit field defines the frame number within the superframe of the frame containing the permanently assigned carrier that is requested to be released.

- permanent_release_carrier_timeslot: This 11-bit field defines the timeslot number within the frame defining the permanently assigned carrier that is requested to be released.

- assign_accept: This 1-bit field indicates whether the RCST can accommodate the requested carrier assignment; "1" indicates acceptance, "0" rejection.

- assign_carrier_superframe: This 8-bit field defines the superframe_id of the superframe containing the assigned carrier that is being confirmed.

- assign_carrier_frame: This 5-bit field defines the frame number within the superframe of the frame containing the assigned carrier that is being confirmed.

- assign_carrier_timeslot: This 11-bit field defines the timeslot number within the frame defining the assigned carrier that is being confirmed.

## 10.4.5   Carrier assignment and release

The NCC can assign specific carriers to an RCST and can subsequently modify or revoke such assignments. Such actions can be in response to a request from the RCST. The operations are communicated using the TBTP.

Continuous-carrier related commands issued by the NCC can cause the RCST to change its operational mode as defined in clause 10.1.

The commands shall have the following effects:

- Carrier assignment: The RCST shall issue an acknowledgement in a CCC-IE and may start using the assigned carrier for return link traffic if it is able to do so. The RCST shall not start transmissions before the notional start time of the assigned time slot. If the RCST supports Fine Sync + Continuous mode, it may continue to use the MF-TDMA air interface as well. If this mode is not supported, the RCST shall issue a logoff command and cease using the MF-TDMA air interface.

- Carrier release: The RCST shall cease using the carrier within 2 seconds of the notional start time of the time slot used to carry the revocation command.

- Permanent carrier assignment: This shall have the same effect as a regular carrier assignment, except that, upon the next initialisation, the RCST shall not attempt to logon but shall instead use the permanently assigned carrier.

- Permanent carrier release: This shall have the same effect as a regular carrier release command. In addition, the RCST shall not attempt to use the carrier upon the next initialisation.

# Annex A (informative):
# Compliance table

**Table A.1: RCST compliance table**

| PROFILE NAME | Baseline | ATM (option) | MPEG2 (option) | Baseline DVB-S2 | ATM DVB-S2 | MPEG2 DVB-S2 | Mobile |
|---|---|---|---|---|---|---|---|
| Access scheme | | | | | | | |
| Fixed MF-TDMA | • | • | • | • | • | • | • |
| Dynamic MF-TDMA | o | o | o | o | o | o | o |
| Traffic Burst Format | | | | | | | |
| ATM | • | • | • | • | • | • | • |
| MPEG2 | | | • | | | • | • |
| Connectivity | | | | | | | |
| IP | • | • | • | • | • | • | • |
| Native ATM | | • | o | | • | o | |
| Dynamic (C2P) | o | o | o | o | o | o | o |
| Channel Coding | | | | | | | |
| RS | • | • | • | • | • | • | |
| Convolutional | • | • | • | • | • | • | |
| Turbo | • | • | • | • | • | • | • |
| CRC | • | • | • | • | • | • | • |
| Capacity Requests and management information | | | | | | | |
| Prefix | • | • | • | • | • | • | • |
| Data Unit Labelling | • | • | • | • | • | • | • |
| Mini-Slots | • | • | • | • | • | • | • |
| Contention Mini-Slot | • | • | • | • | • | • | • |
| Security mechanism | o | o | o | o | o | o | o |
| RCST forward link receivers | | | | | | | |
| Single DVB-S | • | • | • | | | | o |
| Multiple DVB-S | o | o | o | | | | |
| Single DVB-S2 | | | | • | • | • | • |
| Multiple DVB-S2 | | | | o | o | o | o |
| Mobility | | | | | | | |
| Mobility management | | | | | | | • |
| Spectrum spreading | | | | | | | o |
| Continuous carrier operation | | | | | | | o |
| Large uncertainty logon | | | | | | | • |
| NLOS Channel countermeasures | | | | | | | o |
| •: Minimum Compliance Requirement for RCST. o: Optional Compliance Point (statement by manufacturer). | | | | | | | |

# Annex B (informative):
# Bibliography

ITU-T Recommendation I.363-3: "B-ISDN ATM Adaptation Layer specification; Part 3: Type 3/4 AAL".

IETF RFC 4326: "Unidirectional Lightweight Encapsulation (ULE) for Transmission of IP Datagrams over an MPEG-2 Transport Stream (TS)".

# Annex C (normative):
# Channel coding for very short frames

The coding scheme defined in this clause replaces that of the DVB-S2 standard for use with very short frames in the return link of DVB-RCS systems. Refer to the DVB-S2 standard (EN 302 307 [16]) for definitions and terminology associated with this process.

The input to the encoding process shall be a BBFRAME and the output shall be a FECFRAME. Each BBFRAME ($K_{BCH}$ bits) shall be processed by the FEC coding subsystem, to generate a FECFRAME ($n_{ldpc}$ bits). The coding consists of an outer BCH code and an inner LDPC code.

The parity check bits (BCHFEC) of the systematic BCH code shall be appended after the BBFRAME, and the parity check bits (LDPCFEC) of the LDPC encoder shall be appended after the BCHFEC field, as shown in figure C.1. Table C.1 gives the coding parameters for the very short FECFRAME.
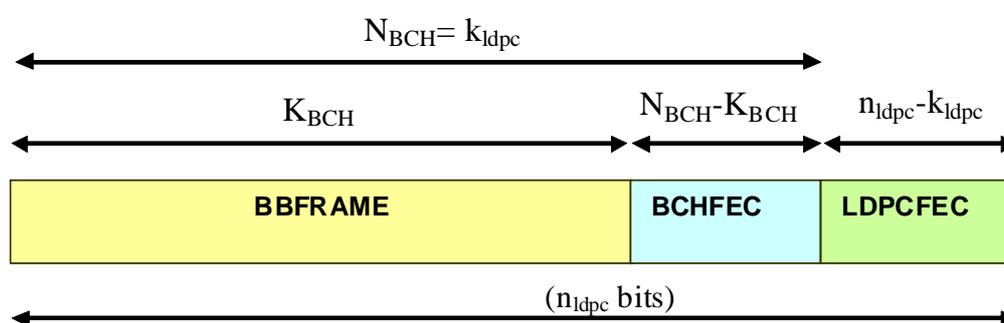


**Figure C.1: Format of data for a very short FECFRAME**

**Table C.1: Coding parameters for very short FECFRAME**

| LDPC code | BCH uncoded Block $K_{BCH}$ | CRC coded block $N_{BCH}$ LDPC Uncoded Block $k_{ldpc}$ | LDPC Coded Block $n_{ldpc}$ |
|---|---|---|---|
| ¼ | 993 | 1 024 | 4 096 |
| ½ | 1 992 | 2 048 | 4 096 |
| ¾ | 3 000 | 3 072 | 4 096 |

# C.1 BCH encoding

A t-error correcting BCH ($N_{bch}$, $K_{bch}$) coding shall be applied to each BBFRAME ($K_{bch}$) to generate an error protected packet. The BCH code parameters are given in table C.2.

BCH encoding of information bits $\boldsymbol{m} = (m_{k_{bch}-1}, m_{k_{bch}-2}, ..., m_1, m_0)$ onto a codeword $\boldsymbol{c} = (m_{k_{bch}-1}, m_{k_{bch}-2}, ..., m_1, m_0, d_{n_{bch}-k_{bch}-1}, d_{n_{bch}-k_{bch}-2}, ..., d_1, d_0)$ is achieved as follows:

- Multiply the message polynomial $m(x) = m_{k_{bch}-1}x^{k_{bch}-1} + m_{k_{bch}-2}x^{k_{bch}-2} + ... + m_1 x + m_0$ by $x^{n_{bch}-k_{bch}}$.

- Divide $x^{n_{bch}-k_{bch}} m(x)$ by $g(x)$, the generator polynomial. Let $d(x) = d_{n_{bch}-k_{bch}-1}x^{n_{bch}-k_{bch}-1} + ... + d_1 x + d_0$ be the remainder.

- Set the codeword polynomial to $c(x) = x^{n_{bch}-k_{bch}} m(x) + d(x)$.

The BCH codes for use with LDPC rates 1/4 and 1/2 has native values of $N_{BCH}$ of 1 023 and 2 047, respectively. For these code rates, the BCH code shall be extended by one bit, computed as the modulo-2 sum of the bits in $(m_{k_{bch}-1}, m_{k_{bch}-2}, ..., m_1, m_0, d_{n_{bch}-k_{bch}-1}, d_{n_{bch}-k_{bch}-2}, ..., d_1, d_0)$. The extension bit shall be transmitted as the last bit of the BCHFEC field.

**Table C.2: BCH code specification for very short FECFRAME**

| Outer code for the (4 096,3 072) LDPCC | |
|---|---|
| Shortened BCH | (3 072, 3 000, t=6) from shortening of a (4 095,4 023) BCH |
| g(x) coefficients (g₀…g₇₂) | 1011101011010010100111001100010011001100100010111101000100111010010001001 |
| **Outer code for the (4 096,2 048) LDPCC** | |
| Extended BCH | (2 048,1 992,t=5) |
| g(x) coefficients (g₀…g₅₅) | 101100110111100110001001000000000010000101010010101011 |
| **Outer code for the (4 096,1 024) LDPCC** | |
| Extended BCH | (1024,993, t=3) |
| g(x) coefficients (g₀…g₃₀) | 11001000100010001001010100000101 |

# C.2    LDPC encoding

The LDPC encoder systematically encodes an information block $i$ of size $k_{ldpc}$, $i = (i_0, i_1, ..., i_{k_{ldpc}-1})$ onto a codeword $c$ of size $n_{ldpc}$, $c = (i_0, i_1, ..., i_{k_{ldpc}-1}, p_0, p_1, ... p_{n_{ldpc}-k_{ldpc}-1})$. The transmission of the codeword starts in the given order from $i_0$ and ends with $p_{n_{ldpc}-k_{ldpc}-1}$.

LDPC code parameters $(n_{ldpc}, k_{ldpc})$ are given in table C.1. The task of the encoder is to determine $n_{ldpc} - k_{ldpc}$ parity bits $(p_0, p_1, ..., p_{n_{ldpc}-k_{ldpc}-1})$ for every block of $k_{ldpc}$ information bits, $(i_0, i_1, ..., i_{k_{ldpc}-1})$. The procedure is as follows:

* Initialize $p_0 = p_1 = p_2 = ... = p_{n_{ldpc}-k_{ldpc}-1} = 0$;

* Accumulate the first information bit, $i_0$, at parity bit addresses specified in the first row of the appropriate table, selected from tables C.4 to C.6 according to the code rate. All additions are carried out in the Galois field GF(2). For example, for rate 1/2 (table C.5):

$$p_{69} = p_{69} \oplus i_0$$

$$p_{440} = p_{440} \oplus i_0$$

$$p_{588} = p_{588} \oplus i_0$$

$$p_{847} = p_{847} \oplus i_0$$

$$p_{1520} = p_{1520} \oplus i_0 ;$$

* For the next 127 information bits $i_m, m = 1, 2, ..., 127$, accumulate $i_m$ at parity bit addresses $\{x + m \bmod 128 \times q\} \bmod (n_{ldpc} - k_{ldpc})$, where $x$ denotes the address of the parity bit of the accumulator corresponding to the first bit $i_0$, and $q$ is a code rate dependent constant specified in table C.3. Continuing with the example, $q=16$ for rate 1/2. So, for example, for information bit $i_1$, the following operations are performed:

$$p_{85} = p_{85} \oplus i_1$$

$$p_{456} = p_{456} \oplus i_1$$

$$p_{604} = p_{604} \oplus i_1$$

$$p_{863} = p_{863} \oplus i_1$$

$$p_{1536} = p_{1536} \oplus i_1 ;$$

- For the 129th information bit $i_{128}$, the addresses of the parity bit accumulators are given in the second row of table C.5. In a similar manner the addresses of the parity bit accumulators for the following 127 information bits $i_m, m = 129, 130, ..., 255$ are obtained using the formula $\{x + m \bmod 128 \times q\} \bmod (n_{ldpc} - k_{ldpc})$, where $x$ denotes the address of the parity bit accumulator corresponding to the information bit $i_{128}$, i.e. the entries in the second row of table C.5;

- In a similar manner, for every group of 128 new information bits, a new row from table C.5 is used to find the addresses of the parity bit accumulators.

After all of the information bits are exhausted, the final parity bits are obtained as follows:

- Sequentially perform the following operations starting with $i=1$

$$p_i = p_i \oplus p_{i-1}, \qquad i = 1, 2, ..., n_{ldpc} - k_{ldpc} - 1;$$

- Final content of $p_i$, $i = 0, 1, .., n_{ldpc} - k_{ldpc} - 1$ is equal to the parity bit $p_i$.

**Table C.3: Values of increment parameter *q* for very short FECFRAME**

| Code Rate | *q* |
|---|---|
| ¼ | 24 |
| ½ | 16 |
| ¾ | 8 |

**Table C.4: Parity bit addresses for rate-1/4 coding in very short FECFRAME**

```
165 340 720 1362 1624 2190 2219 2696 2810 2821 2919 3022
226 404 422 551 753 1265 1567 1891 2309 2676 2739 2809
231 1611 2299 2791
78 518 1274 1506
1084 1432 1796 2529
953 1709 2973 3008
850 1717 2185 2902
588 923 1295 1440
```

**Table C.5: Parity bit addresses for rate-1/2 coding in very short FECFRAME**

```
69 440 588 847 1520
542 724 737 1001 2013
496 611 1510 1583 1707
50 663 942 1601 1674
41 637 805 1146 1858
404 571 1320 1356 1475
96 938 1638 1785 1934
129 807 939 984 1919
206 789 972 1606 1730
212 1325 1343 1463 1859
461 824 1013 1275 1888
265 289 900 1354 1932
394 1171 1279 1569 1879
418 566 654 875 1104
217 247 1267 1508 1869
140 902 1496 1602 1957
```

**Table C.6: Parity bit addresses for rate-3/4 coding in very short FECFRAME**

```
142 200 476 615 818
255 465 792 859 981
516 541 754 793 846
60 107 269 295 840
187 454 754 945 959
252 352 470 610 893
41 399 427 509 688
3 137 420 590 666
329 568 666 765 822
10 71 396 765 785
152 423 466 893 979
179 793 828 991 1014
56 147 234 341 972
101 226 409 563 694
36 331 336 775 806
334 519 753 928 1004
327 346 496 620 638
1 95 224 852 875
59 81 354 581 918
203 876 887 928 965
141 518 667 746 761
254 650 687 772 984
113 227 712 743 1021
438 609 836 845 954
```

# History

<table>
<tr><th colspan="3">Document history</th></tr>
<tr><td>V1.1.1</td><td>September 2001</td><td>Publication as TR 101 790</td></tr>
<tr><td>V1.2.1</td><td>January 2003</td><td>Publication as TR 101 790</td></tr>
<tr><td>V1.2.2</td><td>December 2000</td><td>Publication</td></tr>
<tr><td>V1.3.1</td><td>March 2003</td><td>Publication</td></tr>
<tr><td>V1.4.1</td><td>September 2005</td><td>Publication</td></tr>
<tr><td>V1.3.1</td><td>September 2006</td><td>Publication as TR 101 790</td></tr>
<tr><td>V1.5.1</td><td>January 2009</td><td>One-step Approval Procedure       OAP 20090510: 2009-01-10 to 2009-05-11</td></tr>
<tr><td>V1.5.1</td><td>May 2009</td><td>Publication</td></tr>
</table>