ETSI TR 101 790 V1.1.1 (2001-09)

Technical Report

Digital Video Broadcasting (DVB); Interaction channel for Satellite Distribution Systems; Guidelines for the use of EN 301 790

European Broadcasting Union

Union Européenne de Radio-Télévision

EBU·UER

DVB

Digital Video Broadcasting

ETSI

Reference

DTR/JTC-DVB-97

Keywords

DVB, broadcast, satellite, TV

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, send your comment to:
editor@etsi.fr

Copyright Notification

Contents

Intellectual Property Rights

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

The attention of ETSI has been drawn to the Intellectual Property Rights (IPRs) listed below which are, or may be, or may become, Essential to the present document. The IPR owner has undertaken to grant irrevocable licences, on fair, reasonable and non-discriminatory terms and conditions under these IPRs pursuant to the ETSI IPR Policy. Further details pertaining to these IPRs can be obtained directly from the IPR owner.

The present IPR information has been submitted to ETSI and 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.

IPRs:

Project	Company	Title	Country of Registration	Application n°	Countries Applicable
DVB-RCS	SES-ASTRA	Apparatus and method for generating a carrier frequency	European Patent Office	PCT/EP00/01037	WO
DVB-RCS	SES-ASTRA	Apparatus and method for generating a carrier frequency	European Patent Office	EP 99107496.4	EP
DVB-RCS	SES-ASTRA	Apparatus and method for generating a carrier frequency	European Patent Office	HK 00106169.2	HK

IPR Owner: Société Européenne des Satellites S.A. (SES-ASTRA)
L-6815 Château de Betzdorf
Luxembourg

Contact: Mr. Martin Halliwell
Director of Communications Technology Department
Tel: +352 710725 1
Fax: +352 710725 227

Foreword

This Technical Report (TR) has been produced by the Joint Technical Committee (JTC) 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.

Introduction

The present document gives the first guidelines for the implementation of Digital Video Broadcasting (DVB) interaction channel for Satellite Distribution System (also known as DVB-RCS: DVB Return Channel via Satellite).

The present document describes the main features of the DVB-RCS system and gives guidelines for setting up DVB-RCS networks.

Updates to the present document will be produced when more results from DVB-RCS deployment experience (from field trials, …) become available.

Objective

- The present document describes the DVB-RCS specification for geostationary satellite interactive system [1]. It tries to draw attention to the technical questions that need to be answered in setting up a DVB-RCS network and offers some guidance in finding answers to them.

Target readers

- The present document is aimed at the technical department of satellite broadcasting organizations that are considering implementing satellite interactive networks as well as manufacturers of terminals/gateways. It assumes that readers are familiar with digital broadcasting techniques.

Outline of the present document

The present document provides some examples of implementation details related either with the physical (guard times, preambles, code performance, typical frames, link budget, …) or the medium access control (use of capacity request categories, …) layers.

The document also provides extensive details about Return Channel Satellite Terminal (RCST) implementation guidelines. For example:

- The interface between the RCST indoor unit (IDU) and outdoor unit (ODU) is described.
- The optional Simple Network Management Protocol (SNMP) Management Information Base (MIB) is provided.
- The interaction with SMATV, LAN and IHDN are considered.

Examples of incorporation of DVB-RCS into an existing digital television platform as well as typical DVB-RCS networks are also addressed.

It shall be noted that from up to Clause 9, the clause numbering is similar to the one used in [1] in order to ease the use of the present document.

1 Scope

The present document provides the first guidance to manufacturers, network operators and service providers on equipment design for and operations of a geostationary Satellite Interactive Network, where a large number of Return Channel Satellite Terminals (RCST) are controlled by a central Network Control Centre (NCC).

It is applicable to satellite systems as defined in [1]. In such a system the RCSTs receive a Forward Link based on the DVB-S specifications [2]. The Return Link signal transmitted from the RCST is received by one or more Gateways which also interact with the NCC.

The system as defined in [1] may be used in all frequency bands allocated to FSS or BSS services, and the first expected implementations are in the bands listed in Annex E.

Information concerning the most relevant international regulations and recommendations (ITU, ETSI, DVB, etc.) which in some cases could be applicable to the DVB-RCS terminals is included in Clause 2.

The present Guideline Document (GD), as well as the normative document [1], covers two RCST profiles, Type A which is able to support IP services only, and Type B which shall be able to operate as RCST Type A and also to support native ATM protocols by encapsulating ATM cells within an MPEG2 Transport Stream on the forward link.

The present document should not be used to justify the fulfilment of the essential requirements under Article 3.2 of the R&TTE Directive [3]. Requirements for ElectroMagnetic Compatibility (EMC) under Article 3.1b of the R&TTE Directive [3] are given in EN 300 673 [36] or EN 301 489-12 [37]. Harmful interference is limited by requiring a minimum set of Control and Monitoring Functions (CMF) as well as specifying limits for on-axis radiation, off-axis spurious radiation, carrier suppression, off-axis EIRP emission density and pointing accuracy. These specifications are in general depending on the transmit frequency. For system transmitting at Ku band frequencies EN 301 428 [12] applies. Limits for Ka band systems are given in EN 301 459 [11].

2 References

For the purposes of this Technical Report (TR), the following references apply:

[1] ETSI EN 301 790 (V1.2.2): "Digital Video Broadcasting (DVB); Interaction channel for Satellite Distribution Systems". (also known as the 'DVB-RCS' specification).

[2] C. Berrou and M. Jézéquel: "Non binary convolutional codes for turbo coding", *Elect. Letters*, Vol. 35, N° 1, pp. 39-40, January 1999. http://www-elec.enst-bretagne.fr/publication/publi.shtml

[3] L.R. Bahl, J. Cocke, F. Jelinek, J. Raviv: "Optimal decoding of linear codes for minimizing symbol error rate". *IEEE Trans. Information Technology*, IT-20, pp.284-287, March 1974.

[4] C. Heegard and S. B. Wicker: "*Turbo coding*". Kluwer Academic Publishers, Dordrecht, 1999. http://www.nativei.com/heegard/papers/turbocoding.html

[5] A. J. Viterbi: "An intuitive justification and simplified implementation of MAP decoder for convolutional codes", *IEEE Journal on Select. Areas in Comm.*, vol. 16, pp. 260-264, February 1998. http://sunsite.informatik.rwth-aachen.de/dblp/db/indices/a-tree/v/Viterbi:Andrew_J=.html

[6] A. Dingninou, F. Raouafi et C. Berrou: "Organisation de la mémoire dans un turbo décodeur utilisant l'algorithme SUB-MAP" GRETSI, Vannes, France, September 1999. http://www-elec.enst-bretagne.fr/publication/publi.shtml

[7] Proceedings of the 2nd Symposium on Turbo codes and related topics. 4 - 7 September 2000, http://www-elec.enst-bretagne.fr.

[8] IETF RFC 1483: "Multiprotocol Encapsulation over ATM Adaptation Layer 5".

[9] IETF RFC 1901: "Introduction to Community-based SNMPv2".

[10] IETF RFC 2865: "Remote Authentication Dial In User Service (RADIUS).

[11] 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 to 30,0 GHz frequency bands covering essential requirements under article 3.2 of the R&TTE Directive".

[12] ETSI EN 301 428: "Satellite Earth Stations and Systems (SES); Harmonized EN for Very Small Aperture Terminal (VSAT); Transmit-only, transmit/receive or receive-only satellite earth stations operating in the 11/12/14 GHz frequency bands covering essential requirements under article 3.2 of the R&TTE directive".

[13] IEEE 802.3 (2000): "IEEE Standard for Information technology - Local and metropolitan area networks - Part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications".

[14] EN 50083 series: "Cable networks for television signals, sound signals and interactive services".

[15] EN 61319-1: "Interconnections of satellite receiving equipment - Part 1: Europe".

[16] ETSI ES 200 800 (V1.2.1): "Digital Video Broadcasting (DVB); DVB interaction channel for Cable TV distribution systems (CATV)".

[17] ETSI TR 101 196 (V1.1.1): "Digital Video Broadcasting (DVB); Interaction channel for Cable TV distribution systems (CATV); Guidelines for the use of ETS 300 800".

[18] ETSI TR 101 201 (V1.1.1): "Digital Video Broadcasting (DVB); Interaction channel for Satellite Master Antenna TV (SMATV) distribution systems; Guidelines for versions based on satellite and coaxial sections".

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

[20] ETSI TS 101 224: "Digital Video Broadcasting (DVB); Home Access Network (HAN) with an active Network Termination (NT)".

[21] "DiSEqC Bus Specification" (Version 4.2), EUTELSAT.

[22] ISO 8824: "Information technology - Open Systems Interconnection - Specification of Abstract Syntax Notation One (ASN.1)".

[23] IETF RFC 2579: "Textual Conventions for SMIv2".

[24] IETF RFC 1321: "The MD5 Message-Digest Algorithm".

[25] IETF RFC 2401: "Security Architecture for the Internet Protocol".

[26] IETF RFC 2486: "The Network Access Identifier".

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

[28] IETF RFC 1701: "Generic Routing Encapsulation (GRE)".

[29] IETF RFC 1702: "Generic Routing Encapsulation over IPv4 networks".

[30] IETF RFC 1905: "Protocol Operations for Version 2 of the Simple Network Management Protocol (SNMPv2)".

[31] CEPT/ERC/DEC/(00)03: "Licensing of Satellite (SIT) terminals in 10.7-12.75 GHz/ 29.5-30.0 GHz".

[32] CEPT/ERC/DEC/(00)04: "Licensing of Satellite (SUT) terminals in 19.7-20.2 GHz/ 29.5-30.0 GHz".

[33] CEPT/ERC/DEC/(00)05: "Licensing of Satellite (VSAT) terminals in 14.0-14.25 GHz/ 12.4-12.75 GHz".

[34] IETF RFC 1907: "Management Information Base for Version 2 of the Simple Network Management Protocol (SNMPv2)".

[35] IETF RFC 1213: "Management Information Base for Network Management of TCP/IP-based internets:MIB-II".

[36] ETSI EN 300 673: "Electromagnetic compatibility and Radio spectrum Matters (ERM); ElectroMagnetic Compatibility (EMC) standard for Very Small Aperture Terminal (VSAT), Satellite News Gathering (SNG), Satellite Interactive Terminals (SIT) and Satellite User Terminals (SUT) Earth Stations operated in the frequency ranges between 4 GHz and 30 GHz in the Fixed Satellite Service (FSS)".

[37] ETSI EN 301 489-12: "Electromagnetic compatibility and Radio spectrum Matters (ERM); ElectroMagnetic Compatibility (EMC) standard for radio equipment and services; Part 12: Specific conditions for Very Small Aperture Terminal, Satellite Interactive Earth Stations operated in the frequency ranges between 4 GHz and 30 GHz in the Fixed Satellite Service (FSS)".

3 Definitions, symbols and abbreviations

3.1 Definitions

Void.

3.2 Symbols

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

PER Packet Error Rate. An errored packet is a decoded packet containing at least one bit in error.

E_b/N_o The ratio between total power used for transmission divided by the number of information bits per second and the noise power density.
Typically, E_S/N_0 measurements will be made at the reception site. To get the corresponding E_b/N_0 values [that are required for CMT], a simple calculation has to be made. Annex D gives an example for such a calculation.

3.3 Abbreviations

For the purposes of the present document, the abbreviations defined in [1] and the following apply:

ABR	Available Bit Rate
ACK	Acknowledgement
ACQ	Acquisition burst
AGC	Automatic Gain Control
ANT	Antenna subsystem
ASIC	Application Specific Integrated Circuit
ASN	Abstract Syntax Notation
ATM	Asynchronous Transfer Mod
AVBDC	Absolute volume Based Dynamic Capacity
AWGN	Added White gaussian noise
BER	Bit Error Rate
BSS	Broadcast Satellite Service
CBR	Continuous Bit Rate
CHAP	Challenge-Handshake Authentication Protocol
CMOS	Complementary Metal Oxide Semiconductor
CMT	Correction Message Table
CR	Capacity Request
CRA	Continuous Rate Assignment
CRC	Cyclic Redundancy Check
CS	Contention Slot
CSC	Common Signalling Channel
CSYNC	Contention based SYNC
DC	Direct Current

DULM	Data Unit Labelling method
DVB	Digital Video Broadcast
DVB-S	Digital Video Broadcast via Satellite
EIRP	Equivalent Isotropic Radiated Power
ERC	The European Radio Communications Committee
FCA	Free Capacity Assignment
FCT	Frame Composition Table
FEC	Forward Error Correction
FLS	Forward Link Signalling
FPGA/CPLD	Field-Programmable Gate Array/Complex Programmable Logic Device
FSS	Fixed Satellite Service
GPS	Global Positioning System
GT	Guard Time
HED	Head End Device
HPA	High Power Amplifier
HTTP	HyperText Transfer Protocol
HW	Hardware
IANA	Internet Assigned Numbers Authority
ID	Identity
IDU	Indoor Unit
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering task Force
IFL	Interfacility link
IGMP	Internet Group Management Protocol
IHDN	In-Home Digital Network
IP	Internet Protocol
ISP	Internet Service Provider
JT	Jitter Tolerant
LAN	Local Area Network
LANE	Local Area Network Emulation
LHM	Local Hub Manager
LNB	Low Noise Block converter
LO	Local Oscillator
MAC	Medium Access Control
MAP	Maximum Aposteriori Probability
MECH	Mechanical subsystem
MF-TDMA	Multi Frequency Time Division Multiple Access
MIB	Management Information Base
MPEG	Motion Pictures Expert Group
MSL	Minimum Scheduler Latency
NACK	Negative Acknowledgement
NAS	Network Access Server
NCC	Network Control Centre
NCR	Network Clock Reference
NIT	Network Information Table
NIU	Network Interface Unit
OAM	Operation, Administration and Maintenance
OBO	Output Back off
ODU	Outdoor Unit
OID	Object Identification number
PAP	Password Authentication Protocol, used with PPP.
PAT	Program Association Table
PCR	Program Clock Reference
PDU	Protocol Data Unit
PER	Packet Error Rate
PID	Packet Identifier
PLL	Phase Lock Loop
PMT	Program Map Table
PPP	Point to Point Protocol
PSI	Program Specific Information
PSU	Power Supply Unit
PWD	PassWorD

PWK	Pulse Width Keying
QoS	Quality of Service
R&TTE	Radio Equipment and Telecommunications Terminal Equipment Regulations
RADIUS	Remote Authentication Dial-In User Service.
RBDC	Rate-Based Dynamic Capacity
RCS	Return Channel via Satellite
RCST	Return Channel via Satellite Terminal
RFC	Request For Comments
RMS	Root Mean Square
RMT	RCS Map Table
RT	Real Time
SA	Security Association
SCT	Superframe Composition Table
SI	Service Information
SISO	Soft In/Soft Out module
SMATV	Satellite Master Antenna Television
SMI	Structure of Management Information
SMS	Subscriber Management System
SNMP	Simple Network Management Protocol
SPT	Satellite Position Table
SSP	Satellite Service Provider
SSPA	Solid State Power Amplifier
SUT	Satellite User Terminal
SW	SoftWare
SYNC	Synchronization
tbc	To Be Confirmed
tbd	To Be Defined
TBTP	Terminal Burst Time Plan
TCT	Time-slot Composition table
TDMA	Time Division Multiple Access
TIM	Terminal Information Message
TM	Traffic Manager
TRF	Traffic
TRX	Transceiver
TS	Transport Stream
TS	Transport Stream
TTL	Time to Live
TVRO	TeleVision Receive Only
TWTA	Travelling Wave Tube Amplifier
TxD	Transmission Disabled
U/C	Up Converter
uimsbf	unsigned integer most significant bits first
UIU	User Interface Unit
UW	Unique Word
VBDC	Volume-Based Dynamic Capacity
VC	Virtual Circuit
VCC	Virtual Channel Connection
VCI	Virtual Circuit Identifier
VR	Variable Rate
VSAT	Very Small Antenna Terminal

4 Reference model

4.1 Guidelines for Network Topology

The Reference Model for an interactive satellite network depicted in [1] includes all interconnections among Network Control Centre, Traffic Gateway(s), Feeder(s) and Terminals, which are possible from a functional viewpoint.

In practice not all these interconnections will be implemented. Also, some functional blocks may be co-located. This clause describes therefore the network architectures that are more likely to be implemented for the service provision.

The simplest architecture is an interactive satellite network with a single Traffic Gateway and a single Feeder co-located in an Earth Station (see Figure 4-1). The Network Control Centre is possibly also co-located.

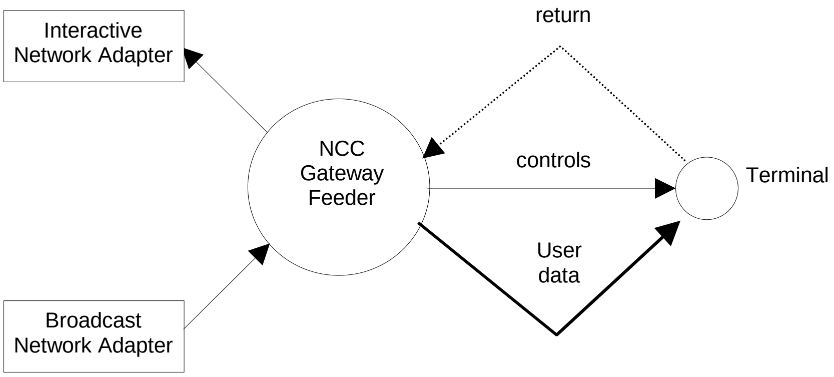


Figure 4-1: architecture with a single gateway and feeder (co-located)

This Earth Station has both an Interactive Network Adapter and a Broadcast Network Adapter. It generates the forward link signal, including user data and the control and timing signals needed for the operation of the Satellite Interactive Network. It receives the RCST return signals, provides interactive services and/or connections to external service providers and networks and it provides monitoring, accounting and billing functions.

When more Feeders exist in the interactive satellite network, the terminals should be able to switch from one to another, without loosing network synchronization (Figure 4-2). In order to achieve this, the following network architecture is envisaged.

Terminals are provided with at least two receivers. One receiver is continuously tuned to the DVB-S MPEG transport stream emitted from a "primary" Feeder, the one which includes the control and timing signals and which provides monitoring, accounting and billing. The other receiver(s) can be tuned to different signals transmitted by "secondary" feedes to receive user data. The capability of the ODU to receive separate signals is the only limitation.

In this configuration, terminals tuned to different "primary" Feeders (most likely belonging to different networks), might receive information from the same "secondary" Feeder(s).

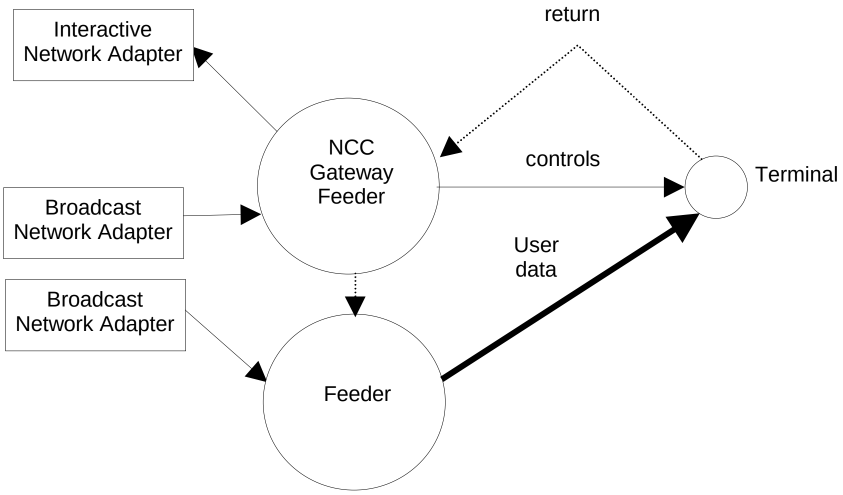

Interactive Network Adapter
return
NCC Gateway Feeder
controls
Terminal
Broadcast Network Adapter
User data
Broadcast Network Adapter
Feeder

Figure 4-2: Architecture with more than one feeder

5 Forward link

Figure 5-1 shows one way to implement the forward signalling data from the NCC (the SI-information for RCS) to an existing DVB-S system. The SI-tables with signalling data to the RCST from the NCC are represented as binary data (for example in binary files). These binary data are sent to the Gateway and put into a PSI/SI-inserter, together with the PSI/SI binary signalling data for DVB-S. There, each of the binary-represented signalling tables are associated with a PID-value and multiplexed into the TS. Then, the RCST must be capable of reading PID-values and its associated data. The RCST know which PID-values represent the SI-tables from the PMT, except the RMT whose linkage-descriptor lies in the NIT (through the PAT). With this "PID-reading-functionality", the RCST is able to extract all signalling information from the NCC.

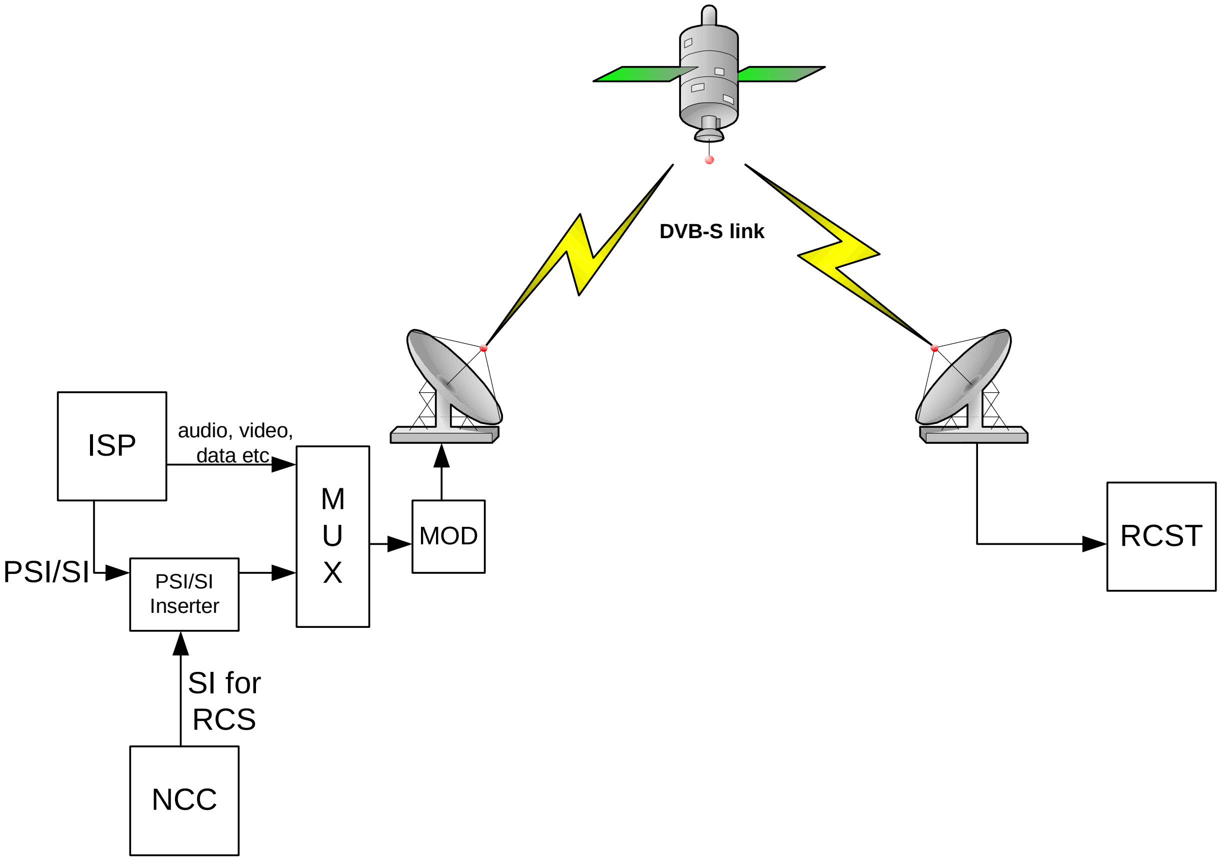


Figure 5-1: Implementation of SI signalling from NCC in DVB-S

6 Return link

6.1 RCST synchronization

For administrative and technical reasons the location of an RCST must be known to the network operator.

An RCS system can be designed assuming an accuracy of the location (longitude, latitude and altitude) of the RCST of no more than a few kilometres. Some network operators may require a better accuracy.

It is recommended that commonly available high precision location systems (e.g. GPS) be used during the RCST installation or any re-installation.

The normalized carrier frequency accuracy RMS value shall be better than 10^{-8}.

The corresponding maximum error value should be 0,06 ppm.

The residual relative frequency offset between any two RCSTs arises due to Doppler on the Satellite-to-RCST downlink and the RCST-to-Satellite uplink. If possible the NCC shall correct for satellite translation error and Doppler introduced on the NCC-to-Satellite uplink and the Satellite-to-NCC downlink. The residual relative frequency offset shall also be compensated for by the NCC. The RCST will not correct satellite transponder translation error.

6.2 Burst format

Contention is possible with ATM TRF bursts. There is no standardized mechanism defined for contention access with the optional MPEG TRF bursts, but it can for example be done by PID assignment to individual terminals.

It is recommended that sync slots are formed by subdivision of traffic slots.. For example, in the single ATM cell sized slot, this slot can be divided into 2 or more sync slots.

All SYNC bursts assigned to the RCSTs must be transmitted. If the terminal does not have a message to send, it must use the 'No message' in the M&C field. If the terminal does not have any capacity request to send, it must send a VBDC request with an amount of 0.

An example for the frequency sequence in the transmission of an ACQ slot could have the format as: B79A A5B7 625D F39F 8A07 09E8 AE86 F1FA E063 045B 9125 AA61 2.

Example values for the preambles could be as given in Table 1.

Table 1: Example preamble values

Preamble_length	Example preamble (hexadecimal notation, msb sent first, gray-coded mapping)
32	0347 F657 1528 E590
48	67FC 8EE8 A8D3 F6CB 4612 B784

Both the preamble and frequency sequences are provided to the RCSTs in the forward link tables (TCT). For the ACQ bursts, the data to send is the concatenation of the preamble and frequency sequence.

When using the optional MPEG2 traffic bursts, the number of packets is implicitly given in the TCT. The RCSTs can apply the following procedure to find this number.

1) **Step1:** Convert timeslot_duration (**tsd**) and burst_start_offset (**bso**) from upcrmsf into uimsbf:

   An n-bit τ of PCR counts in upcrmsf is converted to uimsbf by:

   $\delta(\tau)$ (uimbsf) = base(τ) x 300 + ext(τ)

   Where ext(τ) is the uimbsf defined by the 9 least significant bits of τ and base(τ) is the uimsbf defined by the (n-9) most significant bits.

2) **Step 2:** Convert δ(**tsd)** and δ(**bso)** from PCR counts to modulation symbols

   A duration of δ PCR counts is converted into (uimsbf) modulation symbols by

   d(δ) (uimsbf) symbols = {δ PCR counts} x symbol_rate / 27 000 000

3) **Step 3:** Compute the number of channel bits of **one encoded packet** and divide it by 2 to get (uimsbf) modulation symbols (QPSK assumed).

4) **Step 4:** The number of packets in the time slot is:

   Number of packets = Ψ [(d(**tsd**) – d(**bso**) – d(**preamble**))/ d(**one encoded packet**)]

   Where Ψ[.] is the integer operator

   This method supposes that the guard time of an MPEG TRF burst is shorter than one half MPEG2 TS packet.

6.3 Randomization for energy dispersal

A complementary non self-synchronizing de-randomizer is used in the receiver to recover the data randomized as described in [1]. The de-randomizer shall be enabled after detection of the preamble.

6.4 Coding

6.4.1 CRC error detection code

The CRC code is mandatory on turbo coded CSC bursts.

6.4.1.1 CRC coding example

CRC coding of a CSC burst is taken as an example. The clear, randomized and CRC coded sequences are listed in Table 2 from left to right in order of transmission, in hexadecimal format, with msb sent first. CRC coding is applied after randomization.

Since [1] defines CRC computation as a polynomial division, the CRC *encoder* register is set to initial value 0x0000 before processing the first bit (0) of the first byte 0x43 of the randomized sequence. Finally, after processing the last bit (0) of the last byte 0xd6 the CRC *encoder* register content becomes 0x2f6d. In the absence of errors, the *decoder* CRC register content becomes 0x0000 after processing the complete burst inclusive CRC bits. A circuit as the one shown on Figure 6-1 can compute the CRC word.

Table 2: CRC coding example

Part	**Capabilities**			**MAC address**						**reserved bits**					**CRC**	
byte number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
clear	40	01	21	b1	cd	a1	73	4f	e1	ff	ff	ff	ff	ff		
randomized	43	f7	29	85	fd	19	d0	dc	28	97	48	8c	4c	d6		
CRC coded	43	f7	29	85	fd	19	d0	dc	28	97	48	8c	4c	d6	2f	6d

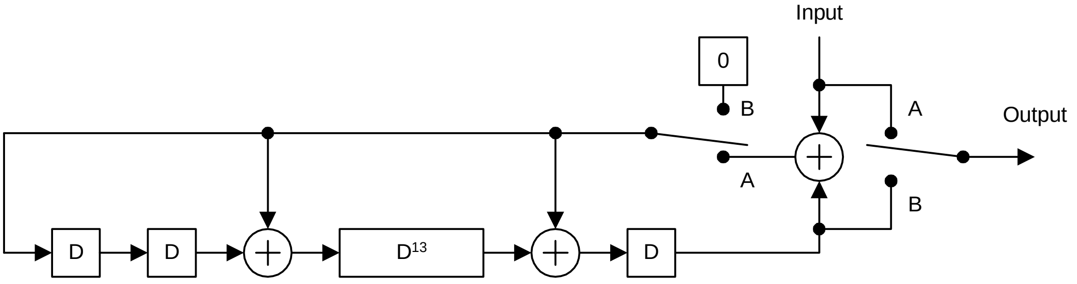


Figure 6-1: CRC calculation

The shift register cells are 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. In the receiver, the CRC is computed in an identical manner on the received data, and compared to the received CRC word. A discrepancy indicates that one or more transmission errors have occurred.

6.4.2 Reed Solomon outer coding

As in the Normative document [1].

6.4.3 Convolutional inner coding

The convolutional encoder should be flushed after each MPEG block.

6.4.4 Turbo code

6.4.4.1 General principles of coding and decoding

The DVB-RCS turbo code puncturing patterns are recalled in Table 3, these puncturing matrixes are to be applied in a repetitive manner. Figure 6-2 and Figure 6.3 show how the encoded bits are to be punctured.

Table 3: Puncturing patterns for Turbo codes. "1"=keep

$$1/3\ \begin{array}{c} Y \\ W \end{array}\begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad 2/5\ \begin{array}{c} Y \\ W \end{array}\begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \quad 1/2\ \begin{array}{c} Y \\ W \end{array}\begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad 2/3\ \begin{array}{c} Y \\ W \end{array}\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \quad 3/4\ \begin{array}{c} Y \\ W \end{array}\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$4/5\ \begin{array}{c} Y \\ W \end{array}\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad 6/7\ \begin{array}{c} Y \\ W \end{array}\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

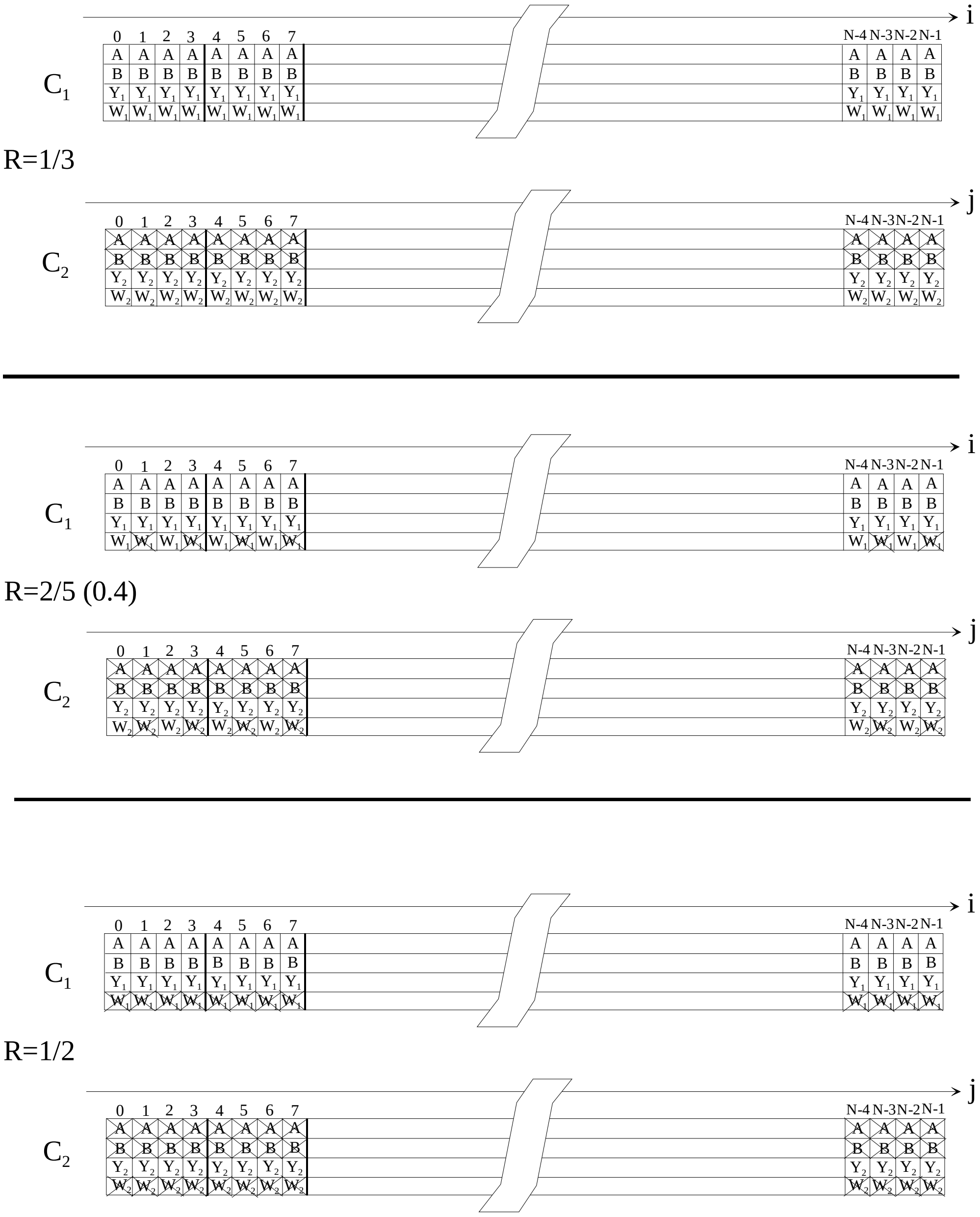


Figure 6-2: Puncturing patterns for rates 1/3, 2/5 and 1/2. Crossed symbols are not transmitted

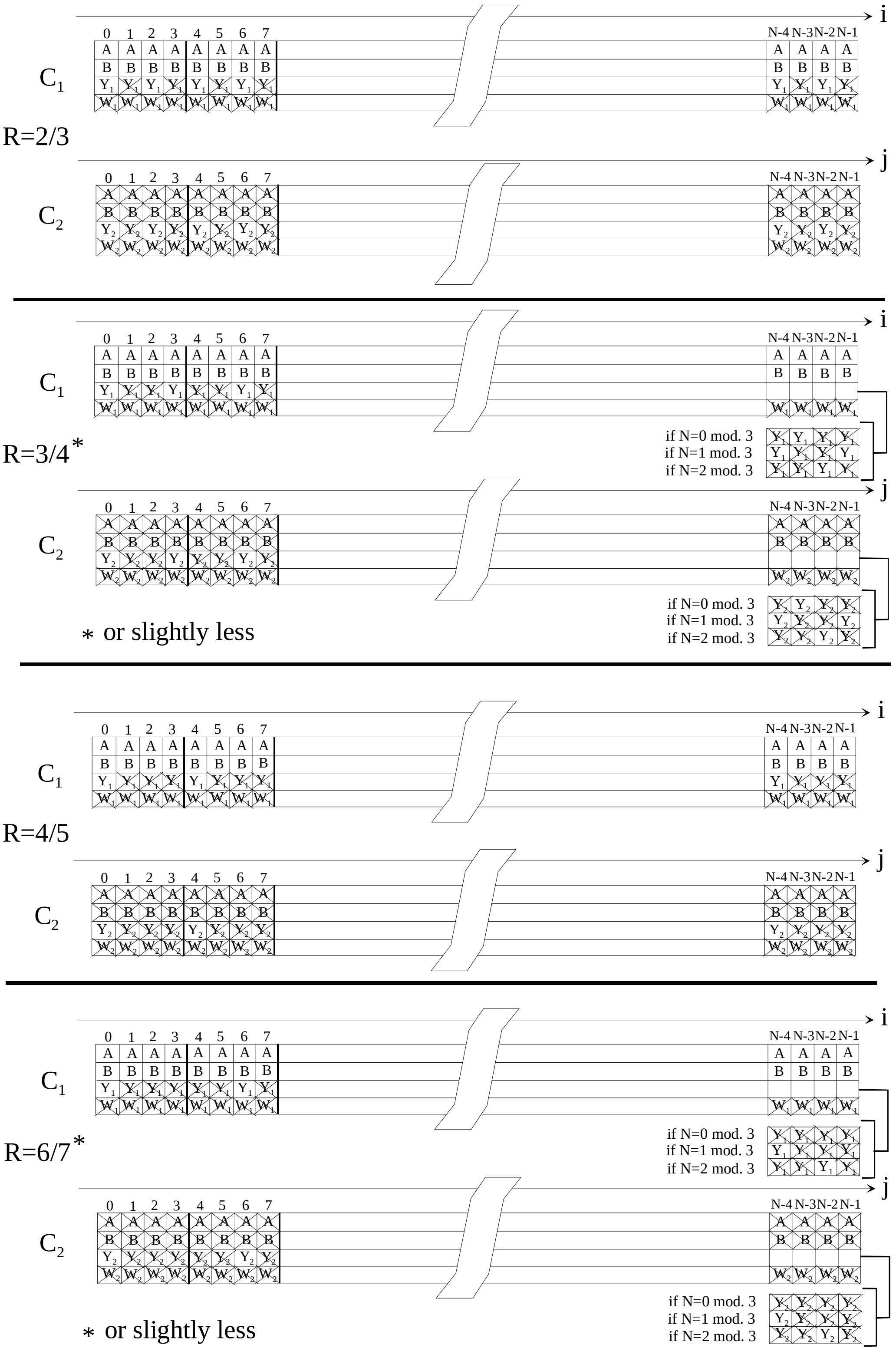


Figure 6-3: Puncturing patterns for rates 2/3, 3/4, 4/5 and 6/7. Crossed symbols are not transmitted

Figure 6-4 depicts the general principle of the Turbo encoder specified in Clause 6.4.4 of EN 301 790 [1]. It is a parallel concatenation of two double-binary, circular, recursive and systematic convolutional encoders (C1 and C2 in Figure 6-4).

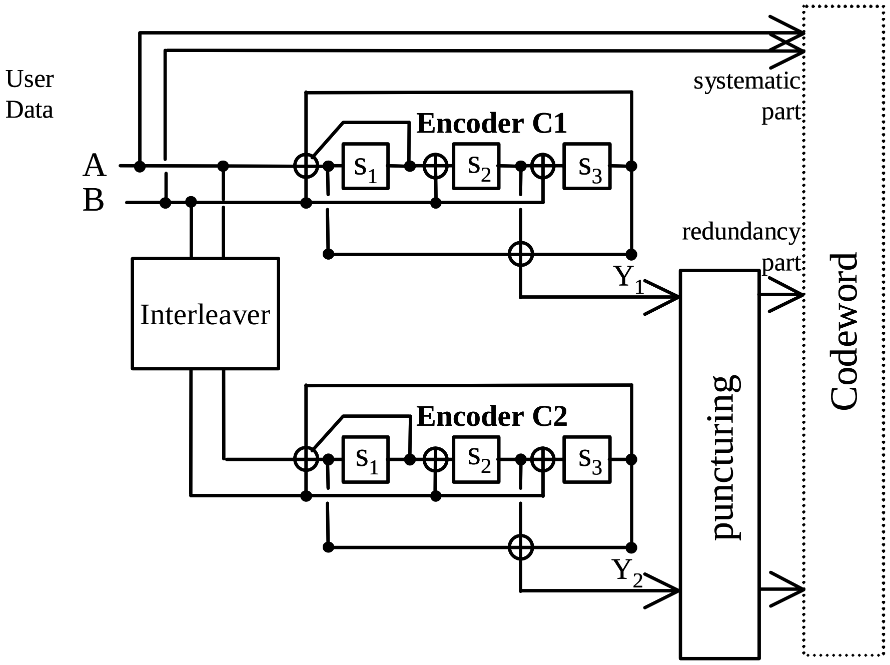


Figure 6-4: The principle of the turbo encoder (rates <1/2 not shown)

The component convolutional encoders are identical. Their associated trellis has 8 states, accepts 2 input bits (A and B in Figure 6-4) at a time and correspondingly has 4 paths branching out of each state. The benefits of using non-binary convolutional encoders can be found in [2]. The permutations (i.e. interleavers) between the component convolutional codes is based on simple algebraic laws, avoiding the use of memory-consuming look-up tables for the permutations. The laws are independent of the code rates and have been fine-tuned for each block size to avoid flattening of the error curve for BER well below 10^{-9}.

For each block of user bits each component encoder computes a state, denoted the circular state, in such a way that the trellis "tail-bites" itself, making the trellis circular: by initializing the encoder's shift register with this circular state, the shift register terminates in the same state when all user bits have been fed into the encoder.

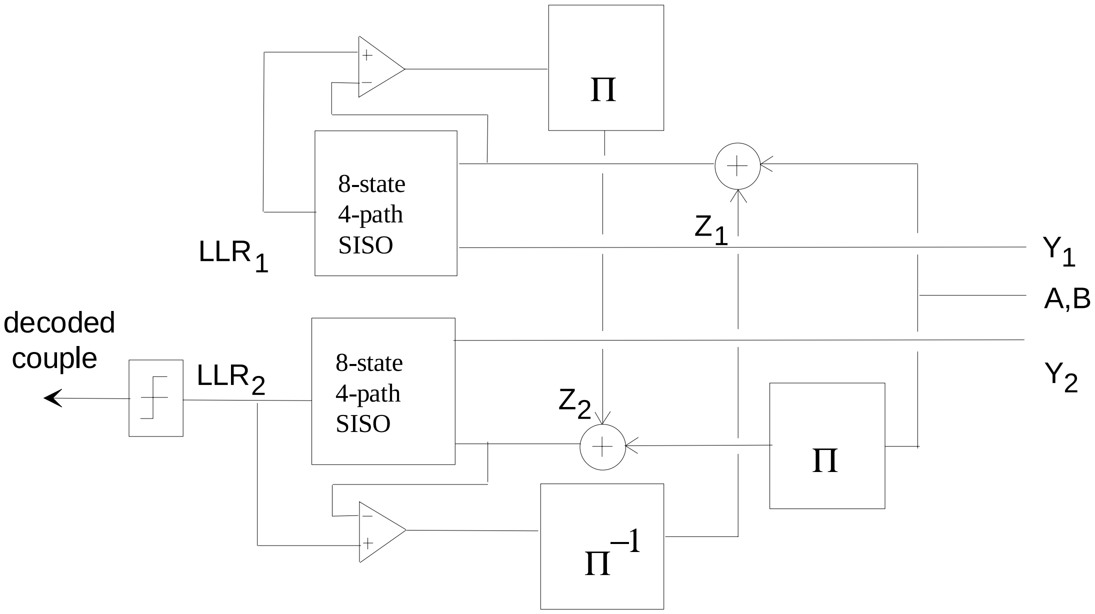


Figure 6-5: The principle of the turbo decoding

Figure 6-5 gives the generic processing engine of an associated turbo decoder. This engine is built around two soft-in/soft-out modules (SISO). The SISO are identical in structure, however, as inputs, one receives data in the natural order and the other one in the interleaved order. The outputs of one SISO, after proper scaling and after reordering, are used by its dual SISO in the next step.

6.4.4.1.1 Puncturing examples for DVB-RCS Turbo Code

The following examples of turbo encoded data can be used for validating a turbo encoder. The order of transmission is the natural order.

The 16-byte packet to be encoded (payload data) is formed by the **repetition of the h87 byte**:

Bit	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
	A_0	**B_0**	**A_1**	**B_1**	**A_2**	**B_2**	**A_3**	**B_3**
H87=	1	0	0	0	0	1	1	1

1) Payload data sequence (A_0B_0 A_1B_1 …….$A_{63}B_{63}$):

```
10 00 01 11 10 00 01 11 10 00 01 11 10 00 01 11 10 00 01 11 10 00 01 11 10 00 01
11 10 00 01 11 10 00 01 11 10 00 01 11 10 00 01 11 10 00 01 11 10 00 01 11 10 00
01 11 10 00 01 11 10 00 01 11
```

2) Rate 1/3

"Y" parity sequence ($Y1_0Y2_0$ $Y1_1Y2_1$ …….$Y1_{63}Y2_{63}$):

```
00 01 11 10 00 01 11 10 00 01 11 10 00 01 11 10 00 01 11 10 00 01 11 10 00 01 11
10 00 01 11 10 00 01 11 10 00 01 11 10 00 01 11 10 00 01 11 10 00 01 11 10 00 01
11 10 00 01 11 10 00 01 11 10
```

"W" parity sequence ($W1_0W2_0$ $W1_1W2_1$ …….$W1_{63}W2_{63}$):

```
01 10 01 10 01 10 01 10 01 10 01 10 01 10 01 10 01 10 01 10 01 10 01 10 01 10 01
10 01 10 01 10 01 10 01 10 01 10 01 10 01 10 01 10 01 10 01 10 01 10 01 10 01 10
01 10 01 10 01 10 01 10 01 10
```

3) Rate 2/5

"Y" parity sequence ($Y1_0Y2_0$ $Y1_1Y2_1$ …….$Y1_{63}Y2_{63}$):

```
00 01 11 10 00 01 11 10 00 01 11 10 00 01 11 10 00 01 11 10 00 01 11 10 00 01 11
10 00 01 11 10 00 01 11 10 00 01 11 10 00 01 11 10 00 01 11 10 00 01 11 10 00 01
11 10 00 01 11 10 00 01 11 10
```

"W" parity sequence ($W1_0W2_0$ $W1_1W2_1$ …….$W1_{31}W2_{31}$):

```
01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01
01 01 01 01 01
```

4) Rate 1/2

"Y" parity sequence ($Y1_0Y2_0$ $Y1_1Y2_1$ …….$Y1_{63}Y2_{63}$):

00 01 11 10 00 01 11 10 00 01 11 10 00 01 11 10 00 01 11 10 00 01 11 10 00 01 11 10 00 01 11 10 00 01 11 10 00 01
11 10 00 01 11 10 00 01 11 10 00 01 11 10 00 01 11 10 00 01 11 10 00 01 11 10

5) Rate 2/3

"Y" parity sequence ($Y1_0Y2_0$ $Y1_1Y2_1$ …….$Y1_{31}Y2_{31}$):

```
00 11 00 11 00 11 00 11 00 11 00 11 00 11 00 11 00 11 00 11 00 11 00 11 00 11 00
11 00 11 00 11
```

6) Rate 3/4

"Y" parity sequence ($Y1_0Y2_0$ $Y1_1Y2_1$ …….$Y1_{21}Y2_{21}$):

```
00 10 11 01 00 10 11 01 00 10 11 01 00 10 11 01 00 10 11 01 00 10
```

7) Rate 4/5

"Y" parity sequence ($Y1_0Y2_0$ $Y1_1Y2_1$ …….$Y1_{15}Y2_{15}$):

```
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

8) Rate 6/7

"Y" parity sequence ($Y1_0Y2_0$ $Y1_1Y2_1$ ....$Y1_{10}Y2_{10}$):

```
00 11 00 11 00 11 00 11 00 11 00
```

6.4.4.2 Implementation trade-offs

For the SISO computation, a Maximum-A-Posteriori (MAP) principle as popularized by the Bahl-Cocke-Jelinek-Raviv algorithm [3,4] gives the best performance but its complexity is rather prohibitive, with today's technology. A good trade-off is to use the Sub-MAP algorithm, also called Max-Log-MAP or Dual Viterbi [5]. The sub-optimality of this algorithm is about 0,2-0,5 dB (depending on the block size of the codeword).

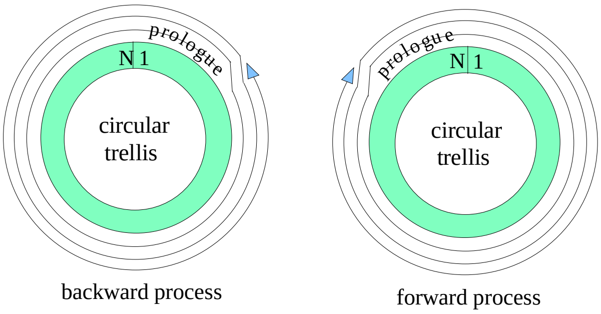


Figure 6-6: Processing a circular code by the backward-forward algorithm

The Sub-MAP performs calculations in opposite directions: backward and forward (Figure 6-6). *N* is the number of couples (A,B) in the block of information bits. The two circular states can be obtained by the decoder by starting the forward and backward some steps ahead of the circular states, called the "prologue" part (Figure 6-6). In practice, a prologue of 32 steps is sufficient to converge to the right circular state.

It has been found in one implementation, using Sub-MAP and at Packet Error Ratio (PER) in the region of 10^{-5}, that a 3-bit quantization entails a penalty of 0,2 dB compared to a 4-bit quantization. It can be further shown, by simulation, that increasing from 4 bits to 5 bits results in no more than 0,1 dB additional coding gain.

Measurements, using samples from a real demodulator with Automatic Gain Control, have put into evidence that the position of the "sample clouds" within the quantizer range could be optimized for Turbo decoding. One implementation uses the following rule-of-thumb: "multiply the analog samples of the demodulator by a constant so that after 4-bit quantization, the average of the unsigned values is equal to Rate*8, Rate being the Turbo code rate (6/7, 4/5, 3/4, etc.)".

There are three types of memories in a turbo decoder:

Input buffer memory: The input buffer is organized in two blocks, the first to hold the quantized I/Q samples of the codeword being processed, the second one to hold the samples of the next incoming codeword. 4-bit quantization should be sufficiently accurate to represent the I/Q samples as discussed above.

Metrics memory: This memory is to store the cumulated metrics calculated during the backward direction (this is comparable to the path memory in a Viterbi decoder). Since the trellis has 8 states, the memory needed is *N* x 8 metrics. Each metric should be coded with 8 bits. There are various methods to reduce the memory requirement for these metrics at the expense of a slight performance degradation (see for instance [5], [6]). Splitting the trellis into sub-blocks (or windows), while storing the starting metrics calculated in the previous iteration for each sub-block, is one such solution. A length of some tens of bits have been found to be a good trade-off in one implementation of this solution.

Extrinsic information memory: This memory is used for transferring the *N* pieces of extrinsic information (the Z data in Figure 6-5) from one decoding step to the next. The code being quaternary, *N* x 4 values must be memorized. These values should be coded with 5 bits. A single memory is sufficient, either for reading or writing. Indeed, when processing either trellis associated with C1 or C2, both in backward or forward direction, the processor refers to the natural order address *i* for C1, and through $i = \Pi(j)$ for C2. Thus, it is not required to have the close-form formulae for the inverse permutation.

6.4.4.3 Typical silicon requirements

With today's technology, with the above implementation trade-offs, Turbo decoding of the DVB-RCS code requires modest silicon resource.

At the "2nd Symposium on Turbo codes and related topics", 4-7 Sept 2000, Brest, [7] , 2 implementations, one on FPGA/CPLD and one on an ASIC qualified for space-born applications, were reported. The following data can be noted:

	FPGA/CPLD	ASIC (space qualified)
Input quantization	4 bits	4 bits
Number of inputs and guaranteed user rate(s)	One input of 4 Mbit/s (6 iterations)	One 6,3 Mbit/s input or three asynchronous inputs of 2,1 Mbit/s each (12 iterations)
Number of input buffer blocks	2	6 (see note)
Number of processing engines	1	3 (see note)
Number of kgates	200	524
Typical use of silicon	Memory: about 100 % RAM cells Algorithm: about 6000 Logic Elements	Input memory < 6 x 16,5 = 100 kgates Turbo decoders (including other memories) < 3 x75 = 225 kgates
NOTE: This architecture is due to the particular requirement of the target application that the ASIC must be capable of processing in parallel **three asynchronous** bit streams each one-third of the nominal bit rate of 6,3 Mbit/s.		

6.4.5 Preferred coding combinations

6.4.5.1 Concatenated coding scheme

The preferred coding combinations for concatenated-coded systems are summarized in Table 4.

Table 4: Preferred coding combination for concatenated-coded systems

Burst Type	Randomization	CRC	Reed-Solomon	Convolutional
TRF	Yes	No	Yes	Yes
SYNC	Yes	No/Yes	Yes	Yes
CSC	Yes	Yes	No	Yes
ACQ	N/A	N/A	N/A	N/A

The rationale for these choices includes the following:

- Randomization is required on all transmissions in order to ensure sufficient energy dispersal. ACQ bursts however contain no information, so the fixed bit sequence should be chosen directly to have adequate spectral properties.

- CRC check is not required on TRF transmissions, because there is no physical layer ARQ procedure in place. CRC is used mainly on CSC bursts, in order to allow collision detection. CRC can be used for SYNC bursts if the contention-based mini-slot method is employed, but is not required for systems using assigned mini-slots.

- CSC bursts may be transmitted for example at full power, or with power that increases until successful reception. The Reed-Solomon code can therefore be left off the CSC transmission – additional transmit power can be used instead. This allows the CSC time slot to be shortened, while preserving the necessary guard intervals.

It should be noted that, when ATM-like TRF bursts are used, all the cells carried in one burst are encoded as a single entity. When MPEG transport is employed, each MPEG packet is encoded separately.

6.4.5.2 Turbo coded systems

The preferred coding combinations for Turbo coded systems are summarized in Table 5.

Table 5: Preferred coding combination for Turbo coded systems

Burst Type	Randomization	CRC	Turbo Code
TRF	Yes	No	Yes
SYNC	Yes	No/Yes	Yes
CSC	Yes	Yes	Yes
ACQ	N/A	N/A	N/A

The rationale for these choices includes the following:

Randomization is required on all transmissions in order to ensure sufficient energy dispersal. ACQ bursts however contain no information, so the fixed frequency sequence can be chosen directly to have adequate spectral properties.

CRC check is not required on TRF transmissions, because there is no physical layer ARQ procedure in place. CRC is used mainly on CSC bursts, in order to allow collision detection. CRC can be used for SYNC bursts if the contention-based mini-slot method is employed, but is not required for systems using assigned mini-slots.

It should be noted that, when ATM-like TRF bursts are used, all the cells carried in one burst are encoded as a single entity. When MPEG transport is employed, each MPEG packet is encoded separately.

6.5 Modulation

The guard time is a "silent" time interval that separates one burst from the following one.

Typical guard time intervals associated with each type of burst are given in Table 6. The guard interval consists of two parts; a Nyquist guard interval based on the symbol rate of the transmission and timing portion based on the propagation delay uncertainties. The combined guard intervals (Nyquist + timing) are applied at the beginning and end of each burst.

Table 6: Examples of guard Time Interval per Burst Type

Burst Type	Nyquist Guard Interval	Absolute Timing Guard Interval	Symbol Independent Timing Guard Interval (network dependent)
TRF	1,5 - 3,5 symbols	0,5 symbol	≈ 1 µs
CSC	1,5 - 3,5 symbols	0,5 symbol	≈ 15 µs (see note)
ACQ	1,5 - 3,5 symbols	0,5 symbol	≈ 15 µs (see note)
SYNC	1,5 - 3,5 symbols	0,5 symbol	≈ 1 µs (see note)

NOTE: For the SYNC, ACQ and CSC slot types, the times quoted are minimum values.

The "burst_start_offset" parameter of the TCT table allows RCST burst transmit time to be offset from the start of the corresponding timeslot. This parameter is related to the timeslot guard time as shown in Figure below.

Thus, the timeslot guard time may be expressed in the following ways:

guard time = time_slot_duration – (preamble length + payload length)

guard time ≥ burst_start_offset

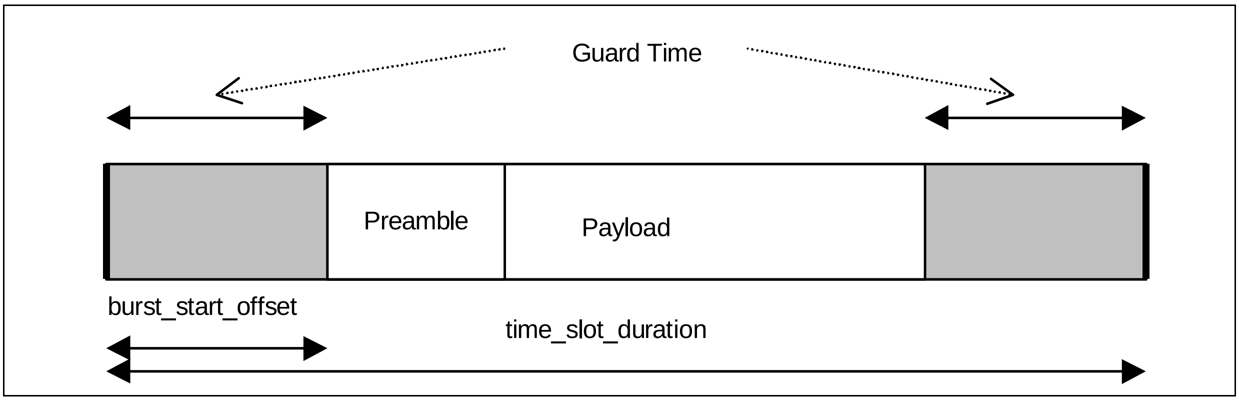


Figure 6-7: Relationship between parameter "burst_start_offset" in TCT and guard time

The "burst_start_offset" parameter may be used to allocate a different amount of guard time to different burst types – e.g. a CSC burst with a potentially large timing error will typically be assigned to a time slot with a larger "burst_start_offset" value than that of a TRF burst. In practice, it seems sensible to try and centre the transmitted burst within its designated timeslot. In this case, we have:

guard time = 2 × burst_start_offset

The OBO can be controlled independently by the RCST through measuring the RF output power and reducing the power if required. Alternatively the OBO could be controlled in conjunction with the NCC by using for example the optional SNMP mechanism. The NCC determines from time to time the operating point on the output power versus input power curve of the HPA. This can be realized for example by requesting RF power changes and monitoring the actual received RF power. During normal operation the NCC requests only power levels that ensure sufficient OBO. For this method the NCC needs knowledge of the relation between operating point and spectrum degradation. Therefore, the RCST communicates this characteristic during the configuration parameter exchange.

The burst should not introduce degradations to adjacent bursts. A ramp up/ down characteristic such as the one of Figure 6-8 shall be fulfilled.

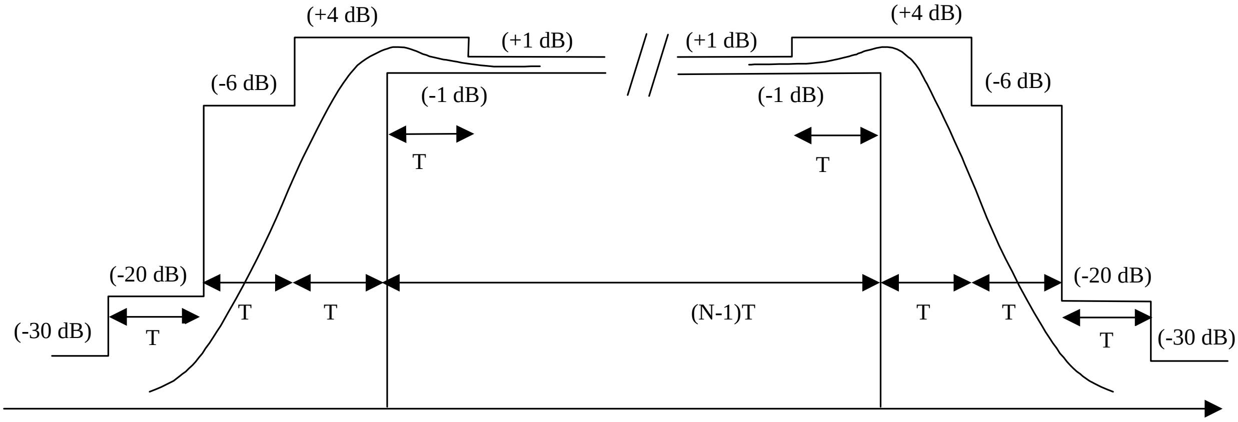


Figure 6-8: burst ramp up/down for a N symbol burst transmitted by the RCST

6.6 MAC messages

The DULM mechanism does not use RFC 1483 [8].

6.6.1 Multiple channels per terminal

In this clause channels are defined as independent communications paths, to which different capacity may be dynamically allocated (just like capacity allocation to distinct terminals) and separately managed, according to different traffic and network profiles.

The rationales for allowing terminals with multiple "channels" are:

- provision of differentiated QoS services – having for example one channel for "best effort" traffic and one channel for "1st class" traffic in each RCST would allow the network to allocate the uplink time-slots with weighted priorities
- provision of connection-oriented services (such as ATM Virtual Connections) where there may be per-VC service guarantees
- provision of simultaneous communications between one RCST and several Gateways

A functional representation of this feature is depicted below. The figure shows there is at least one queue per channel ID.

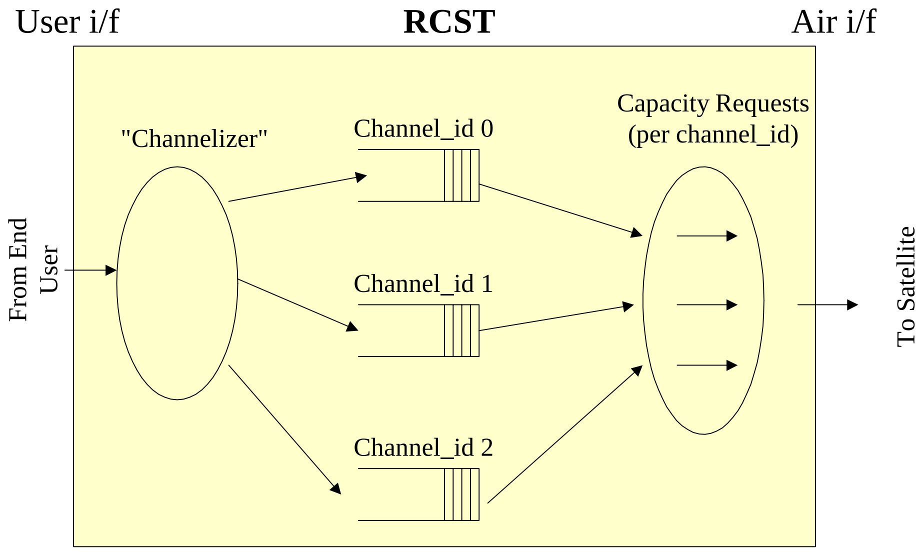


Figure 6-9: Functional representation

In all cases, it must be understood that the number and usage of different channels per terminal is left to the network operator decision. Signalling messages for the creation, modification or deletion of such "channel IDs", and associating them with traffic flows will be clarified at the next revision.

6.7 Multiple access

Wherever possible the frame and the superframe should be organized according to the following conditions (see also Clause 8.2 RCST Addressing):

- each frame to be the same length in terms of NCR clock periods,
- a frame on a given frequency to carry homogeneous traffic, that is, the bit rate, the code rate and the burst length shall be constant,
- carriers to be ordered according to burst length, bit rate, code rate and frequency,

these conditions can be observed within frequency bands if the transponder needs to be divided into different regions (perhaps to support different beams for example).

For the sake of uplink resource efficiency, the superframe duration is usually envisaged to be in the order of a few tens or hundreds of milliseconds.

6.7.1 Example for segmentation of return link capacity

For an RCST using the fixed MF-TDMA mode, no bursts are to be sent when the RCST is asked to reconfigure its transmission parameters.

6.7.1.1 ATM traffic time slots

In order to facilitate the usage of the normative document [1] an example for segmentation of return link capacity is given here. The segmentation follows the definition of Fixed MF-TDMA that can be found in Clause 6.7.1.1 of the normative document [1]. Four peak information bit rates are available:

- 144 kbit/s
- 384 kbit/s
- 1 024 kbit/s
- 2 048 kbit/s

Each RCST is assigned to a specific bit rate depending on its capabilities and the local conditions.

Traffic bursts, which are transmitted in traffic time slots, contain one ATM cell. All information bit rates apply to the 53 bytes contained in the ATM cell. Therefore, the information bit rate is available to the user in the case of an RCST Type B. In the case of an RCST Type A some overhead is needed for encapsulating IP datagrams. Error correction coding and the preamble of traffic bursts depend on link budget including demodulator performance. They lead to specific time slot duration for traffic bursts and traffic slots. CSC and acquisition slots have the same duration as traffic slots (although the actual bursts may be shorter than the slots). The duration of synchronization slots is half of the duration of traffic slots.

A frame consists of a number of time slots on a number of carriers. Each frame has a duration of 26,5 ms. The number and composition of time slots per frame is determined by the information bit rate to be supported by the frame. Table 7 shows the frame composition depending on the peak information bit rate. Either CSC and acquisition slots or synchronization slots are available on a carrier in a frame, as shown by the two lines for each data rate.

Table 7: Frame composition example

Peak information bit rate	Slots per carrier and per frame				Peak symbol rate in kbit/s	Carriers per frame
	traffic slots	CSC/acquisition slots	synchronization slots	total of traffic slot durations		
144 kbit/s	9	1	0	10	238	60
		0	2			
384 kbit/s	24	2	0	26	618	23
		0	4			
1 024 kbit/s	64	4	0	68	1 618	9
		0	8			
2 048 kbit/s	128	8	0	136	3 237	4
		0	16			

Traffic capacity is assigned on a frame basis. This means that the repetition rate of the TBTP is equal to the frame period, thus the TBTP shall be distributed every 26,5 ms. The number of traffic slots for each of the information bit rates allows generating bit rates that are multiples of 16 kbit/s. With a CRA or RBDC assignment of n time slots in every consecutive frame the RCST gets the bit rate equal to n times 16 kbit/s assigned.

The overhead slots, i.e. CSC, ACQ and SYNC slots, are aligned across the carriers and at the beginning of the frame. This is shown in Figure 6-10.

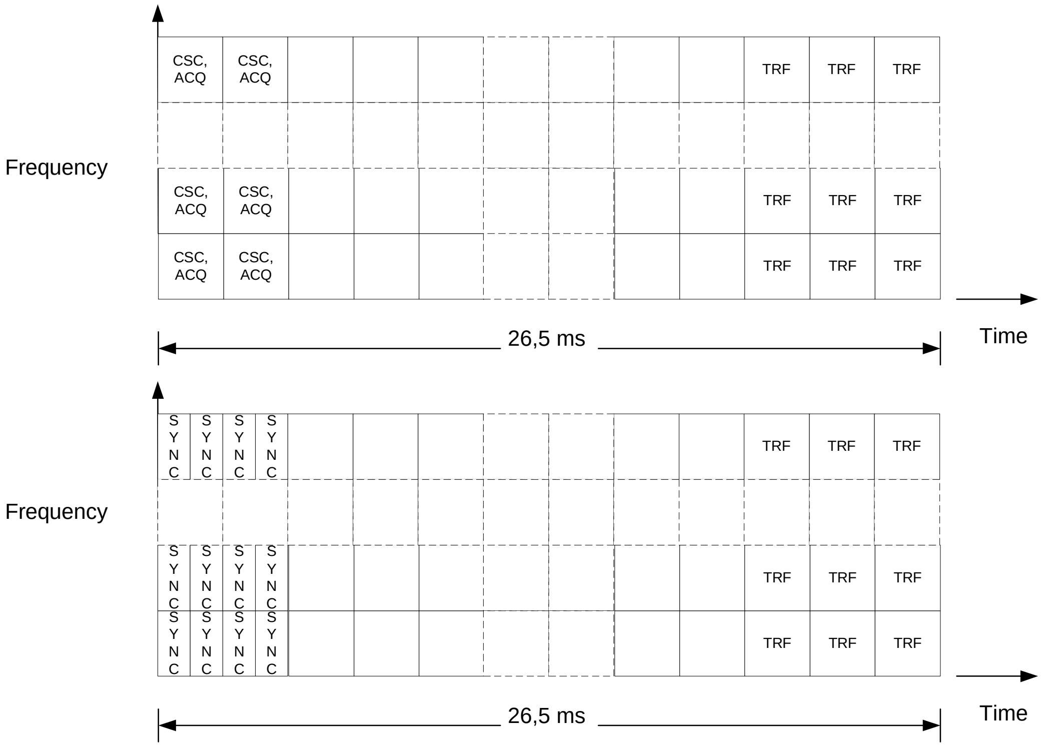


Figure 6-10: Example frame composition principle

The bandwidth of a frame is less or equal to the frequency hopping range of RCSTs, which is 20 MHz in this example. The number of carriers in this bandwidth can be derived from the symbol rate, which depends on the error correction coding and the preamble configuration. Example values for symbol rate and number of carriers are given in Table 7.

SYNC slots are assigned with a 32 frames period. Therefore, each terminal transmits a SYNC burst every 848 ms.

6.7.1.2 Optional MPEG traffic time slots

There are broadly two families of segmentation of return link capacity, one based on ATM traffic time slots and the other based on optional MPEG TS time slots. The selection of one or the other depends on many considerations pertaining to the network operator's strategy.

Reasons for using MPEG time slots can be:

- Better error protection thanks to longer blocks of information bits.

- Better efficiency for encapsulating broadband IP traffic, especially for streaming and for multicast traffic in the return direction.

The Gateway and the Feeder is co-located and some traffic from some RCSTs are intended to be transferred to the forward link. This traffic may contain pure MPEG streams, IP/DVB streams, or both. For IP/DVB streams, the operation of decapsulation-encapsulation can be avoided when there is no need to perform such operation (no IP filtering at the Gateway-Feeder interface for the said traffic).

Use of DVB-RCS air interface in satellite with future regenerative on-board payloads, as such payloads are likely based on DVB-S on the downlink.

The following is an example of a simple return link segmentation based on MPEG time slots.

The reference design is based on a symbol rate of 270 ksymbol/s. The choice of 270 ksymbol/s as basic rate is motivated by its simple relationship with the NCR clock (27 MHz). This does not preclude the use of other symbol rates if other criteria dictates so.

Other design objectives are given below:

- Traffic slots accommodate 1 MPEG packet (i.e. 752 modulation symbols plus preamble, guard-time and a FEC redundancy part depending on FEC rate).
- The equivalent of one traffic slot per frame is used to carry signalling. For simplicity, the beginning of the frame is used and it is divided into mini-slots.
- The frame length should be about 45 ms (to keep latency low), while the total time allocated to mini-slots should be kept below 15 % of a frame.
- Mini-slots should accommodate indifferently CSC, ACQ and SYNC (for simplicity).
- Short bursts are always of 16-byte long, including 2-byte CRC and are always Turbo coded with rate 1/2.
- The TDMA Preamble inclusive of UW is equal to 48 Symbols, for all burst types. This does not preclude the use of shorter preambles, depending on demodulator performance and the ability of the RCST to maintain synchronization.
- TDMA Guard-time should be equivalent to a terminal-to-satellite range uncertainty of 4 km for TRF slots and greater than 50 km for other slots.

These design objectives can be fulfilled with the segmentation given below:

Table 8: FRAME COMPOSITION FOR Symbol rate = 270 ksymbol/s

FEC	TRF_symb: TRF slot length in symbols 52+752+R	No of mini-slots in one TRF slot	Mini-slot length (symbols) GT+48+64	GT length for mini-slots (symb)	Resulting Terminal-to-satellite distance uncertainty allowed by GT	No. of equiv. TRF slots per frame	Resulting Frame length	Signalling overhead
1/2	1 556	7	222	46	51 km	8	47,6 ms	14 %
2/3	1 180	5	236	60	66 km	10	43,7 ms	10 %
3/4	1 055	5	211	35	39 km (see note)	12	46,9 ms	9 %
4/5	992	4	248	72	80 km	12	44,1 ms	9 %
6/7	932	4	233	57	63 km	13	44,9 ms	8 %

R = number of modulation symbols carrying Turbo redundancy bits.
GT = variable Guard-Time of mini-slots in symbols (for TRF slots a fixed 4-symbol guard-time is assumed).
NOTE: If required, the number of mini-slots can be reduced to 4 to increase the GT for this case.

The allocation of the pool of mini-slots to CSC, SYNC and CSYNC bursts is dependent on the network operator optimization strategy. For example, in the cases of 4 mini-slots per frame (TRF slot FEC=4/5 and 6/7), a network operator may apply the following rules:

First mini-slot: assign to periodic SYNC of "low activity" terminals, any such terminal getting a grant of one mini-slot every 256 frames (period of approximately 12 s. Note that such a long period may be incompatible with up path power control).

Second mini-slot: assign to CSC and SYNC for terminals attempting to log-on, with a periodicity of 16 frames (approximately 720 ms).

Third mini-slot: assign to CSYNC (contention-based SYNC).

Fourth mini-slot: assign to periodic SYNC of "high activity" terminals, any such terminal getting a grant of one mini-slot every 32 frames (approximately 1,4 s).

The classification of terminals into *high activity* and *low activity* groups will be dependent on the network operator strategy. As a simple example of such classification: a high activity terminal not making any successful request over a period of consecutive 256 frames will be "downgraded" to low activity. A low activity terminal, having made two or more successful requests over a period of consecutive 32 frames will be "elevated" to high activity status. The NCC keeps tracks of the terminals' status, terminals do not need to know their status. Under the above scheme, there can be 256 idle terminals and 32 active terminals sharing the bandwidth of the TDMA frame of one 270 ksymbols channel, or a total of 288 log-on terminals. An idle terminal has one mini-slot opportunity every frame in contention mode but only one pre-assigned mini-slot every 256 frames (or 12 s). An active terminal has one mini-slot opportunity every frame in contention mode and one pre-assigned mini-slot every 32 frames (or 1,4 s) for signalling purposes.

The user information rate per traffic slot at the 188-byte level (i.e. the 4 bytes MPEG header is included as user bytes) can be computed using the following formula:

$$\text{User rate per TRF slot} = 270\times2\times (752/\text{TRF\_symb})\times1/\text{Ns}$$

Where, TRF_symb is number of symbols in a traffic slot and Ns is the number of equivalent traffic slots per frame.

The maximum user rate, when all traffic slots except the signalling slot is allocated to one terminal is given by:

$$\text{Max. user rate} = (\text{Ns-1}) * (\text{User rate per slot})$$

Table 9: USER RATES FOR Symbol rate = 270 ksymbol/s

FEC	Number of traffic slots	Length of traffic slots (TRF_symb)	User rate at 188-byte level per traffic slot (kbit/s)	Max. user rate when all traffic slots are granted to one terminal (kbit/s)
1/2	7	1 556	32,62	228,34
2/3	9	1 180	34,41	309,72
3/4	11	1 055	32,08	352,83
4/5	11	992	34,11	375,24
6/7	12	932	33,52	402,19

Therefore, if each terminal gets one traffic slot per frame then from 7 to 12 active terminals can be accommodated simultaneously, each with a bit rate of about 32 kbit/s (CRA allocation). If 32 active terminals dynamically share the overall bandwidth then each will get on average between 7 kbit/s (FEC=1/2) and 12,6 kbit/s (FEC=6/7) with the maximum "peak rate" between 228 kbit/s (FEC=1/2) and 402 kbit/s (FEC=6/7).

6.8 Capacity request categories

Hereinafter, MSL refers to the minimum scheduler (i.e. the entity which generates the TBTP) latency time. For example, if the structure described in Clause 6.7.1 is applied, the MSL (in frames) can be defined as the minimum time from the beginning of the frame in which a request is sent until the frame in which a corresponding assignment will apply. The MSL corresponds to the worst case round trip propagation delay from any RCST to scheduler and back again, plus any on-board delay at the satellite, plus scheduler processing delays, rounded up to a whole number of frames.

The sum of affected or requested traffic for any given RCST shall not exceed the maximum transmit capability of that RCST or the maximum allowed transmit capability whichever is less.

The mapping between source traffic type and capacity category depends on the types of service provided, on the transmission protocols used and on constraints imposed by the satellite orbit. For these reasons, the following suggested mapping are only provided as examples.

6.8.1 Continuous Rate Assignment (CRA)

CRA should be used for traffic which requires a fixed guaranteed rate, with minimum delay and minimum delay jitter, such as the CBR class of ATM traffic.

This category is also preferred for variable rate traffic which cannot tolerate the MSL delay. An example of such traffic for a GEO satellite could be the ATM Variable Bit Rate - real time (VBR-rt) class.

6.8.2 Rate Based Dynamic Capacity (RBDC)

RBDC should be used for variable rate traffic which can tolerate the MSL delay. A typical application for RBDC over a GEO satellite could be the ATM Available Bit Rate (ABR) class.

CRA and RBDC can be used in combination, with CRA providing a fixed minimum capacity per frame and RBDC giving a dynamic variation component on top of the minimum. A typical application could be the ATM Variable Bit Rate - non real time (VBR-nrt) class.

6.8.3 Volume Based Dynamic Capacity (VBDC)

VBDC should be used only for traffic that can tolerate delay jitter, such as the Unspecified Bit Rate (UBR) class of ATM traffic or standard IP traffic.

VBDC and RBDC can also be used in combination for ABR traffic, with the VBDC component providing a low priority capacity extension above the guaranteed limit in the RBDC category.

6.8.4 Absolute volume Based Dynamic Capacity (AVBDC)

AVBDC is similar to VBDC and should be used instead of VBDC when the RCST senses that a VBDC request might be lost. This might happen when requests are sent on contention bursts or when the channel conditions (PER, Eb/No) are degraded. Traffic supported by AVBDC is similar to the VBDC one.

6.8.5 Free Capacity Assignment (FCA)

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.

6.9 Queuing Strategy

An RCST may queue all traffic arriving from the user interface, using separate queues for traffic which is subject to different transmission priorities. As an example, one queue shall be provided for each of the following priorities, where implemented:

Real Time (RT) priority, corresponding to traffic carried using the CRA capacity category. Such traffic typically represents emulated circuit switched operation with tight constraint on end to end jitter build-up.

Variable Rate (VR) priority, corresponding to traffic carried using the RBDC capacity category. Two VR traffic sub-priorities are possible: jitter sensitive (VR-Real Time or VR-RT) or jitter tolerant (VR-Jitter Tolerant or VR-JT). Where an RCST is required to support traffic with separate VR-RT and VR-JT components, then at least one queue shall be provided for each component with the VR-RT queue being the higher priority.

Jitter Tolerant (JT) priority, corresponding to all other traffic i.e. that carried using the VBDC/AVBDC capacity category.

Separate queues could also be implemented in association with different values of channel ID field as defined in Clause 6.6.

Queue lengths are a function of several factors including traffic profile, total system loading and congestion control methods. The queuing strategy for traffic classes using a combination of the above categories is not considered in the present document. However, it is likely that it requires a further queue per circuit source to allow context specific transmit processing. More queues may be required to meet network management constraints, such as the congestion control strategy. For example, the ATM explicit rate control for ABR traffic may require one queue per Virtual Circuit (VC).

6.10 Requesting Strategy

The traffic requesting strategy used by Type B RCSTs shall depend on the traffic priority. These strategies are defined below for the case where no congestion control is applicable.

It shall be noted that the terminal does not apply requests based on anticipation of future traffic.

a) RT priority traffic

The RCST shall not issue any requests for RT priority traffic, or for the RT priority component of mixed priority traffic. The capacity assigned to the RCST will be the CRA capacity.

b) VR priority traffic

VR priority traffic can be sent only where the RCST has negotiated a non-zero RBDC category limit with the NCC. Such traffic requires a request for RBDC capacity to be sent which matches the current demand.

The RCST shall calculate the total VR request required as the sum of the jitter sensitive component (VR-RT) and the jitter tolerant component (VR-JT). The VR-RT component shall be the amount of VR-RT traffic required to be sent in the frame being requested (i.e. one MSL in the future), and corresponds to that traffic of this class which was received during the prior frame period, less any part which is already allowed for in the RT priority traffic (CRA) capacity as a minimum capacity. If the resulting value is negative, then it shall be set to 0. The VR-JT component shall be the size of the current total VR-JT queue, after allowing for assignments in the current frame, less the total of pending VR-JT requests. A pending VR-JT request is defined as a request transmitted to the scheduler (or left active where no VR request update was sent) within the last MSL frames, i.e. the request or associated assignment is either in transit to/from the scheduler or being processed by the scheduler. If the resulting value is negative, then it shall be set to 0.

The total VR request shall be limited to the maximum RBDC rate. The resulting request shall be transmitted if it satisfies any of the following criteria:

- it is not equal to the last RBDC request,
- the time since the last RBDC request was sent is approaching the time-out value.

Since RBDC requests are non-cumulative, a duplicated transmission of each request may be advisable where the probability of a request loss is unacceptable in guaranteeing the QoS of the associated traffic class.

For VR-RT traffic, the assignment strategy must ensure ready availability of at least one CR opportunity for that RCST in each uplink frame. Where this is not implicitly guaranteed by other means, the simplest way of ensuring this is to always use a combined RBDC+CRA approach with the CRA component giving one or two slots per frame minimum assignment. A similar provision may also be needed for VR-JT traffic, where the QoS guarantees a given minimum latency.

To avoid a potential loss of VR communication, the requesting strategy shall ensure that no single loss of an RBDC request will trigger the time-out mechanism.

c) JT priority traffic

For JT priority traffic, the RCST shall calculate the total JT request required as the sum of a JT traffic component and a network management messaging component.

The JT traffic component shall be the size of the current total JT queue, after allowing for assignments in the current frame, less the pending JT request. The pending JT request is defined as the rolling sum of all JT requests previously transmitted to the scheduler less the JT component of assignments already received i.e. it represents requests and assignments which are either in transit to/from the scheduler or stored in the scheduler. If the resulting JT component value is negative, then it shall be set to 0.

The network management messaging component is the number of cells required for the network management messaging defined above. If the resulting total JT request is negative or zero, then no JT request shall be transmitted. Otherwise the RCST shall use as many VBDC/AVBDC CR as needed to transmit the total JT request, subject to availability, given that each transmitted request is limited to a maximum size and that such requests are cumulative/absolute. In the event of conflict between a need to transmit both RBDC and VBDC requests, then priority shall be given to RBDC.

6.11 Assignment/Allocation Processing

6.11.1 Assignment/Allocation Extraction

Each frame, the RCST shall process the TBTP message from the scheduler for its allocation area, to extract the assignment count and slot allocations for its next uplink frame transmissions.

6.11.2 Mapping Assignment/Allocation to Queues

The assignment signalled by the scheduler is the sum of all components (CRA + RBDC + VBDC +AVBDC + FCA), with no indication of the relative amount of each component. In principle, the RCST is free to use this capacity as wished to transmit any queued traffic in the scheduled frame, however, the following approach is suggested:

For all traffic types, the RCST shall qualify the "normal" volume of cells loaded (as described in the remaining points below) by any overriding congestion control limit which applies to each individual circuit. Note that this allows any queued data for a non-congested circuit to replace circuit data which is limited by congestion control.

For traffic using a combination of two or more categories, the limits below apply to the component of the traffic associated with that category.

CRA category traffic shall be loaded for transmission first, with the RT cell quantity defined by the amount queued for the transmit frame. The total amount loaded (including any CRA component of a combined CRA + other category traffic) shall not exceed the authorized CRA capacity.

RBDC traffic shall be the next loaded for transmission in priority order VR-RT then VR-JT. The VR-RT cell quantity shall be that defined by the amount queued for the transmit frame. The sum of VR traffic shall not exceed the authorized RBDC maximum.

VBDC/AVBDC traffic shall be used to complete the allocation, within the limits imposed by its queue.

6.12 Procedure for contention resolution

When capacity requests (CR) are sent in contention slots (CS), there is a definite chance of a collision, which typically results in the capacity request being lost. In such situations, the RCST will want to reissue its capacity request. Figure 6-11 illustrates the procedure an RCST should follow before reissuing its capacity request.

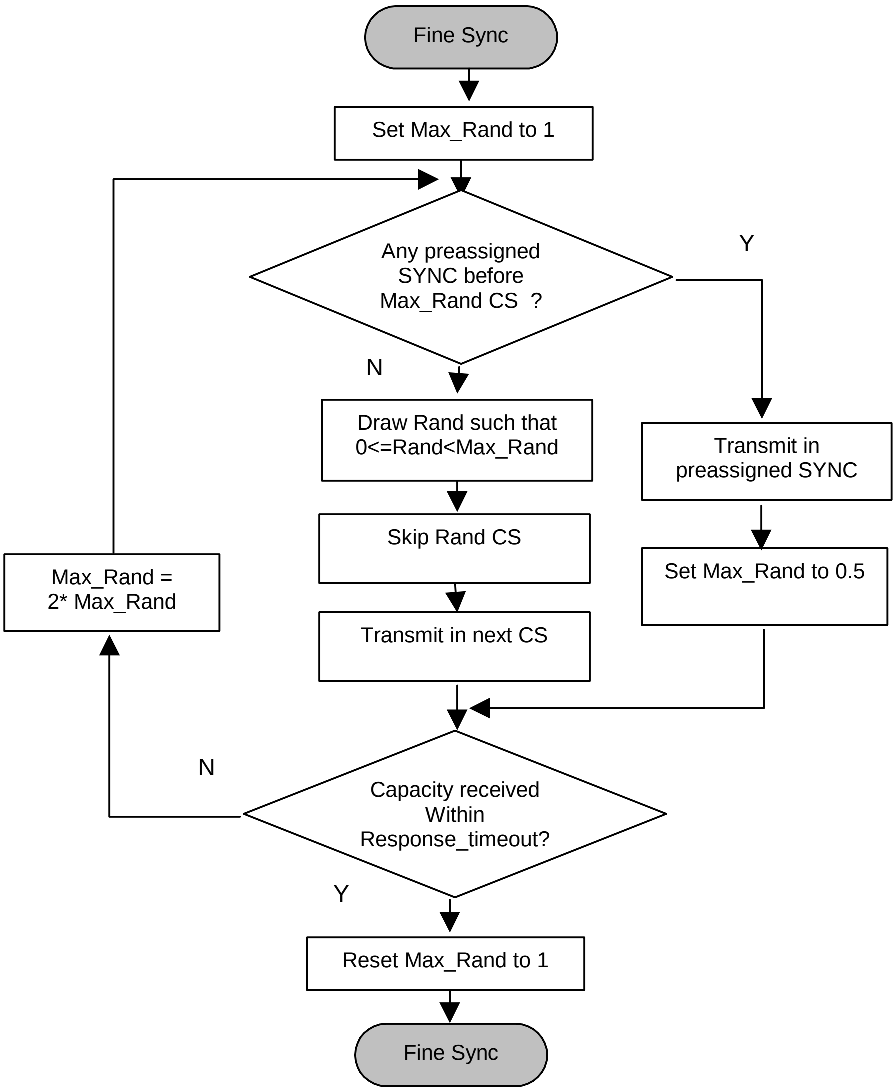

Fine Sync
Set Max_Rand to 1
Any preassigned SYNC before Max_Rand CS ?
Y
N
Draw Rand such that 0<=Rand<Max_Rand
Skip Rand CS
Transmit in next CS
Transmit in preassigned SYNC
Set Max_Rand to 0.5
Max_Rand = 2* Max_Rand
N
Capacity received Within Response_timeout?
Y
Reset Max_Rand to 1
Fine Sync

Figure 6-11: Protocol for handling lost capacity requests (CRs) sent in contention slots (CS)

The contention resolution algorithm described is a variation of the one used in the Ethernet standard (IEEE 802.3 [13]), also known as the "truncated exponential back off" resolution. The present algorithm is governed by two parameters:

> **Max_Rand**: is a *real* number, the integer value of which defines the random domain in number of Contention Slots (CS) assigned to the RCST's Group_ID. It has value 1 at the beginning and the end of the contention resolution procedure.
>
> **Rand**: is a random integer drawn with uniform probability in the random domain [0, Max_Rand] in each contention resolution attempt.

Before selecting a contention slot to transmit in, the RCST first checks if a preassigned SYNC slot is within the random domain (i.e. occurs earlier than the next Max_Rand number of CS). If such a preassigned SYNC does not exist in the random domain, the RCST selects a contention slot drawn at random in the random domain and transmit in that CS. On the other hand, if a preassigned SYNC slot is within the random domain, the RCST shall wait and transmit in the preassigned SYNC slot.

When a transmission is sensed as lost, by expiration of the "response_timeout", the RCST doubles the parameter Max_Rand and reiterates the same algorithm. This is called the back-off loop. The doubling of Max_Rand is to avoid congestion in the network (as in the Ethernet standard, except that the procedure described here also seizes the opportunity offered by a preassigned SYNC slot).

Note that after a successful transmission or after a transmission in a preassigned SYNC slot, the random domain is reset to [0,1], its initial value.

The parameter 'response_timeout' in Figure 6-11 to that of the Contention Control descriptor detailed in Clause 8.5.5.10.14 of the normative document.

If after sending a request in contention mode, a pre-assigned SYNC slot occurs before "response_timeout", then the RCST should systematically send a duplicata of the request and reset the random domain as well as the timer for "response_timeout". The NCC shall systematically discard a request when it is received from a pre-assigned slot and it follows a request made in a contention slot, the two received requests being separated in less than "response_timeout".

RCSTs should also abort the procedure when the back-off loops has been executed more than 8 times, as this is indicative of a prolonged outage, due to propagation or hardware failure. In that way, in the absence of transmission errors, a successful exit of the procedure is ensured in less than the period between two pre-assigned SYNC slots.

Note particularly that there is no explicit way for an RCST to know whether its capacity request has been granted, other than it has to deduce from the TBTP whether it did indeed receive the capacity it asked for.

7 Synchronization procedures

7.1 Overall events sequencing

Terminals shall know what the initial power level is that they can use to enter the network. The knowledge about this power level shall be gathered during the installation procedure. Terminals that support the RCST MIB will keep the information about this power level in the rtnLk subgroup. The MIB parameter RCSTConfigLinesAirIfRtnLkDefIfLevel will provide the default transmitted IF power level, specified in tenth of dBm, out of the IDU for sending a CSC burst at RCST reboot or power on. The value for the parameter is derived from RCSTConfigLinesAirIfRtnLkFirstIflevel, by adding a specific offset (according to the rain margin) to the IF level at which the RCST received a first response during installation when sending CSC bursts repeatedly with increasing level. During normal operation, the terminal should send CSCs with the same symbol rate as used during installation, or adapt the power level.

[1] explicitly defines that the terminal should stay in HOLD state also after a power switch off/reset in order to avoid that the end user circumvents the NCC Transmit_Disable simply by switching the RCST off and on again. This implies that RCST stores the HOLD state in non-volatile memory. The only way that the RCST can exit the HOLD state is by receiving a TIM message containing a Transmit_Disable flag set to 0.

When the RCST enters the hold state, it shall cease transmission and release all assigned logon session parameters (i.e. logon_id, group_id, timeslot allocations).

7.2 Initial synchronization procedure

The same clause in the normative document [1] contains all details.

7.3 Logon procedure

The normative document defines the logon procedure in Clause 7.3 by a flow chart. Parameters for the procedure are defined in Clause 8.5.5.10.14 as fields of a descriptor. The flow chart given in Figure 7-1 is derived from the one of the normative document, but expresses parameters by the field names of the descriptor.

The normative document defines that after exceeding a number of unsuccessful CSC attempts the RCST shall give up the logon procedure. For refining the definition a description is given here how the RCST continues operation and when it resumes logon. In order to improve system performance by avoiding network overload an increasing wait period is introduced. As shown in Figure 7-1 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.

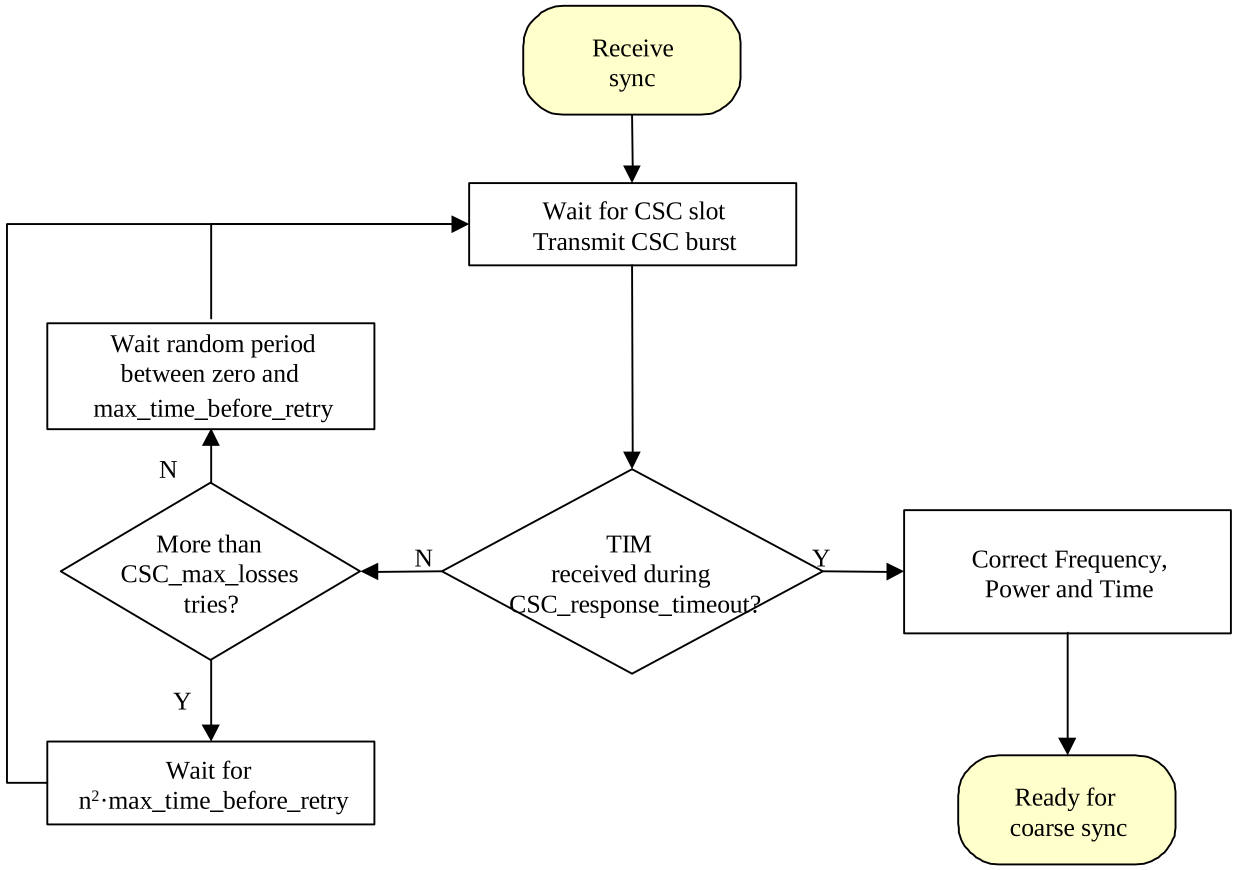

Receive sync
Wait for CSC slot
Transmit CSC burst
Wait random period between zero and max_time_before_retry
N
More than CSC_max_losses tries?
N
TIM received during CSC_response_timeout?
Y
Correct Frequency, Power and Time
Y
Wait for n²·max_time_before_retry
Ready for coarse sync

Figure 7-1: Logon procedure

7.4 Coarse synchronization procedure (optional)

The same clause in the normative document [1] contains all details.

7.5 Fine synchronization procedure (optional)

The same clause in the normative document [1] contains all details.

7.6 Synchronization maintenance procedure

The same clause in the normative document [1] contains all details.

7.7 Logoff procedure

Values for the RCST log-off conditions could be as:

Loss of receive synchronization, i.e.:

- NCR not received for 6 consecutive seconds,
- CMT burst correction not received for 3 consecutive SYNCs.

8 Control and management

For optional Type B RCSTs, if capacity request mechanisms can be in conflict with ATM features, then these mechanisms can be disabled by the NCC and not used on the satellite network.

8.1 Protocol stack

The same clause in the normative document [1] contains all details.

8.2 RCST addressing

The Group_ID is probably the same on two consecutive sessions.

Group_ID: The Group_ID was introduced to reduce the signalling load on the forward link. A unique TBTP is applicable to a group of terminals. The network operator may perform Group_ID grouping based on the subscriber profiles and criteria such as:

- RCSTs of similar transmission capabilities are grouped together (i.e. same maximum transmission data rate, RCSTs of like ACQ capabilities)
- Terminals belonging to the same SMATV installation (These terminals will share the ODU and, thus, the return (uplink) frequency. NCC must handle these terminals as a group.)
- RCSTs belonging to different Service Providers are given different Group_IDs
- RCSTs having similar subscriber profiles
- RCSTs of similar capacity needs and expected traffic patterns

Interactive Network ID and Population ID: during installation the installer enters for the Interactive Network ID and the Population ID values that the network operator has assigned. The terminal uses the values for accessing the forward link signalling as defined in the normative document. At a later point in time the values can be changed by an authorized installer or remotely by the NCC using the optional SNMP mechanism.

During logon the RCST is assigned to a specific group and superframe by the field Group_ID of the Logon Initialize Descriptor and the field Superframe_ID of the Satellite Return Link Descriptor. The TBTP, which assigns time slots to terminals, is sorted by Group_ID but not by Superframe_ID. Time slot assignment is on a superframe basis and valid for a specific repetition of the superframe. This gives the impression that all RCSTs of a group must belong to the same superframe. This is not the case if superframes and their repetitions have the same duration and start time. Then the superframe count in the TBTP applies to all the superframes that RCSTs of a group belong to.

8.3 Forward link signalling

Note that if the NCC calculates time offset due to RCST-to-satellite range the SPT need not be transmitted.

The accessing of the forward link signalling is described in Clause 8.5.5.11 in [1]. It is recommended to keep Figure 35 (Example of the forward Link Location) by hand when reading the extra explanations that are provided here.

Information about the Satellite Interactive Network shall be conveyed in a RCS Map Table (RMT). One RMT will be available per satellite network (orig_network_id). The terminal will be able to find the RMT by looking for a linkage descriptor of linkage type RCS Map (0x07) in the NIT on the start-up transport stream. Like all linkage descriptors, it contains the TS_id and the service_id for the service (RCS Map) that it links to. The NIT will contain a loop over transport streams, where for each TS multiple descriptors can be included. The satellite delivery system descriptor allows the terminal to tune to the transport stream that carries the RMT by giving information on the physical properties of the stream (such as frequency, polarization and symbol rate). On this stream, the terminal will find the PID that carries the RMT by using the PAT and PMT with the service_id from the RCS Map linkage descriptor.

The RMT has the same syntax as the NIT, which means that it can provide linkage descriptors and a loop over transport streams. Of course, the RMT will **not** use the default NIT PID (0x0010). The terminal will look for linkage descriptors of linkage type RCS FLS (0x81) to find the TS_id and service_id for the Forward Link Signalling service it should use. When the network uses multiple RCS FLS, the terminal has to search for its population_id in the private data part of the RCS FLS linkage descriptors. The terminal shall select the RCS FLS linkage descriptor that contains its population_id. In the loop over transport streams, the terminal shall then extract all relevant TS related information. This includes the satellite forward link and satellite return link descriptor. The satellite forward link descriptor is an enhanced satellite delivery descriptor, that also includes some extra information about the network's usage of the forward link (see Clause 8.5.5.10.11 in EN 301 790). In early implementations, where terminals will not support the parallel reception of multiple forward links, there will be only one such descriptor (link_usage = '000' - combined signalling/data link). The satellite return link descriptor provides such information as the superframe_id and the transmit centre frequency offset, that the terminal needs to know in order to be able to use the return link efficiently.

The terminal will then finally tune to the transport stream that carries its data and signalling. To find the RCS specific signalling, it will find the PMT for the RCS specific signalling by using the PAT and PMT. In the FLS PMT, the terminal will be told in what PIDs the RCS tables and TIMs are carried and which PID is used for the NCR. The terminal can then load all relevant information from SCT, FCT, TCT and SPT, extract the 27 MHz reference from the NCR and select a suitable CSC in the superframe that matches the superframe_id from the satellite return link descriptor. When the CSC is received by the NCC, and the terminal is authenticated successfully, the terminal will get a TIM that contains among other things the group_id and logon_id. From that time on, the terminal can get slot allocations and correction messages through TBTP and CMT.

More information on frame definitions and the use of group_id and logon_id can be found in Clauses 6.7 and 8.2 of the present document.

The slot type used in the CMT table and correction message descriptor is intended to indicate on which type of burst the measure was performed. First implementations of this mechanism are expected to use only the SYNC bursts.

The sync_repeat_period of the sync_assign_descriptor is intended to indicate the number of superframes between 2 SYNC assignments. 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, a SYNC_repeat_period = 2 means that two superframes containing the SYNC slot assignment are separated by 2 superframes that do not have the SYNC slot assigned.

8.3.1 Repetition Rates

The normative document [1] states that the TBTP shall be updated every superframe, or equivalently, for each increment of the 'superframe_count' parameter.

When the superframe duration is long (its definition allows superframes to be as long as 93,2 s), an update rate of once per superframe would lead to unacceptable response times for the DVB-RCS system. This potential problem is remedied by observing that the definition of the TBTP does not preclude it from being transmitted several times per increment of the 'superframe_count' parameter. In such a scheme, consecutive TBTPs contain the same 'superframe_count' value, but different 'frame_number' values. For example, the first TBTP could use 'frame_number' values 1-8, the second TBTP could use frame numbers from 9-16 and so on. Note, however, that the 'frame_number' values do not necessarily have to be consecutive numbers.

Similarly, if the frames themselves are long in duration, they too could possibly be split up into multiple TBTP assignments. In this case, the TBTPs would contain the same 'superframe_count' and 'frame_number' parameter values, but would differ in their 'start_slot' parameter values.

Although difficult to pinpoint an exact value, a general guideline is to issue the TBTP associated with a particular 'superframe_id' multiple times per second, so that this latency does not noticeably degrade the system response time.

8.3.2 DVB RCS SI table updates

An update of the RCS SI tables is indicated by the use of the current_next_indicator and version_number fields of the SI section header. The Table Update Descriptor may be used to inform the RCST that there is an upcoming change to one of these tables. For the SI tables used for frame composition, SCT, FCT and TCT, the start time parameters of the SCT (superframe_start_time_base, and superframe_start_time_ext) shall be used to invoke the transition from current to new tables.

8.4 Return Link Signalling

Described in EN 301 790 [1].

8.5 Coding of SI for forward link signalling

8.5.1 Table definition

The TBTP shall be transmitted in the proper order (consecutive frames are sent one after each other). TBTP shall not be sent fragmented (for example a part of frame n, then a part of frame m, then again a part of frame n and so on).

8.5.1.1 Timeslot Composition Table (TCT)

Preamble_length: For the CSC, SYNC and TRF bursts, this parameter is used to unambiguously communicate the length of the preamble sequence that precedes the encoded information bits. When used with the ACQ burst, however, this parameter indicates the length in symbols of both the preamble and the special frequency sequence, as illustrated in Figure 6-5 of the normative document.

For an RCST, there is no particular need to be able to differentiate between the two segments of the ACQ burst. To a Traffic Gateway, on the other hand, the situation is different. Thus, in order to fully exploit the capabilities an ACQ burst offers, the Traffic Gateway should know the length of the individual segments of the ACQ burst. If applicable – i.e. if the ACQ burst is used – then this information should be communicated from the NCC to the Traffic Gateway. This can be done efficiently in any number of ways, but is not detailed further here, as it falls outside the scope of the present document to describe NCC-Traffic Gateway communication to this level of detail.

8.6 SNMP (optional)

8.6.1 Introduction

The DVB-RCS normative document provides interoperability between RCSTs and NCC and allows basic operation. Functionality that goes beyond that can be realized using Simple Network Management Protocol (SNMP) in conjunction with a specific Management Information Base (MIB), as it is mentioned in Clauses 8.4.2 and 8.4.3 of the DVB-RCS normative document [1]. In the following a specification for the RCST MIB, its use, the supported network topology and the application of the standard MIB II is given.

In the framework of the SNMP option the word "shall" is used in the following when referring to functionality that is required to guarantee integrity of the option.

SNMP version 2c, as defined in [9], shall be used.

The MIB definition is provided in Annex F.

8.6.2 Definitions

The following definitions apply in conjunction with the RCST MIB:

Session: A session represents the period of time during which a RCST is connected to the RCS network.

Traffic manager (TM): The TM is the real-time NCC subsystem which provides the connection management for RCST connectivity, by co-ordinating and updating various servers and databases, controlling the return link bandwidth and resource availability, and updating route tables.

Local Hub Manager (LHM): NCC's element and network management system (non real-time functions).

Subscriber Management System (SMS): The SMS is the NCC subsystem, which maintains the RCSTs profiles. The SMS enables the RCS Service Administrator to update data associated with a particular subscriber. Examples of such data include subscriber name, service location, service features, and feature parameters.

Super User: A person that can access a higher level of RCST functionality than the normal user. This can be thought of as a local RCST administrator. The Super User is local to the RCST Domain.

Installer: A person authorized by the network and/or service provider(s) to install a RCST according to operational and legal requirements. The installer is the only authorized person having access to all installation related parameters referenced in the RCSTSystem objects in the MIB.

Hub: The Network Control Centre (NCC). The word hub is used in some names that do not originate from the DVB-RCS specification and guideline documents.

8.6.3 RCST operational status

The DVB-RCS normative document [1] defines in Clause 7.1 states and procedures for RCST synchronization with the network. In conjunction with SNMP an operational status is introduced in order to enable additional functionality. Status values are related to RCST states. The status has one of the following values:

Idle: Corresponds to the Off/Stand-by state of the DVB-RCS normative document [1].

Initialized: Corresponds to the Receive Sync state of the DVB-RCS normative document [1].

Hold: Corresponds to the Hold state of the DVB-RCS normative document [1].

OAM Active: Similar to the Fine Sync state of the DVB-RCS normative document [1]. The difference is that in the OAM Active status the RCST is allowed to transmit OAM data, but not to transmit traffic data.

Active: Similar to the Fine Sync state of the DVB-RCS normative document [1]. The difference is that the RCST uses different VCCs for OAM and traffic data.

Fault: A fault condition has occurred and user intervention is required.

OAM data consists in particular of SNMP messages between the NCC and the RCST, using the MIBs defined in Annex F. Authentication and FTP based software download belong also to OAM.

An RCST signals its SNMP capability by the SNMP field of the RCST Capability field carried in the CSC burst that is sent during logon. An NCC that wants to use SNMP with the Active and OAM Active statuses for access control replies with a TIM that contains a Network Layer Info Descriptor. The Message_body field of the Network Layer Info Descriptor can take either a long version or a short version. The RCST should distinguish between the two versions by means of the length. In the short version it contains the following information (Table 10), which enables transmitting OAM data:

Table 10: Message body field of the Network Layer Info Descriptor

Syntax	No. of bits	Mnemonic
Message_body {		
TM_IP_Address	32	uimsbf
OAM_VPI	8	uimsbf
OAM_VCI	16	uimsbf
Random_Number	16	uimsbf
}		

TM_IP_Address: IP address of the traffic manager, used for SNMP messages.

OAM_VPI: VPI used for OAM data

OAM VCI: VCI used for OAM data

Random_Number: Random number used for RADIUS [10] request in conjunction with user authentication.

In the long version the message body is defined as follows.

The Message_body of the Network Layer Info descriptor is a datagram that will take the form of an SNMP message. The Message Body shall be formatted according to IETF RFC 1901 [9] and RFC 1905 [30] and the PDU type shall be a SetRequest PDU. Several Variable Bindings with objects defined in the RCST MIB may be included in the Variable Bindings list of Network Layer Info Descriptor, dependent on the capabilities of the RCST, and the service requested by the RCST / supported by the DVB RCS gateway. As described in [1] the length of the Network Layer Info descriptor shall not exceed 255 bytes and preferably fit within a single TS packet.

Examples of RCST MIB objects that may be included in the Variable Bindings of the Network Layer Info Descriptor:

Table 11: Message body field of the Network Layer Info Descriptor

OBJECT NAME	SYNTAX	Support	DESCRIPTION
RCSTConfigNetworkTrfIpAddr	IpAddress	Optional	RCST IP address assigned dynamically through TIM.
RCSTConfigNetworkOamIpAddr	IpAddress	Optional	OAM IP address assigned dynamically.
RCSTConfigNetworkAAAPriIpAddr	IpAddress	Optional	Primary IP address to use for AAA. (Authentication, Authorization and Accounting)
RCSTConfigAccessPolicyIndex	Integer32 value 1	Optional	Index for Traffic manager IP address.
RCSTConfigAccessPolicyIpAddr	IpAddress	Optional	Primary Traffic manager IP address.

After achieving fine synchronization (and while maintaining it) the RCST is in the OAM Active status and can use SNMP to communicate with the NCC. If it has to transmit traffic data, then it has to go to the Active status by using objects of the callCntlTrap subgroup of the RCST MIB. Other objects of the callCntlTrap and callCntl subgroups allow leaving the OAM status or OAM Active status.

8.6.4 MIB-II

The MIB-II standard shall be implemented on each RCST that provides the SNMP option. The major groups shall be supported. It is worth pointing out that in order to be supported, a MIB-II sub tree (or a MIB sub tree, in general) has to be entirely implemented, i.e. all its objects have to be defined.

The following MIB-II groups shall be supported: system (1), interfaces (2), ip (4), icmp (5), tcp (6), udp (7), transmission (10) (especially dot3 subgroup), snmp (11). Note that in case of future development requesting the implementation of new SNMP objects hosted in other MIB-II subgroups, it shall be mandatory to implement the whole corresponding subgroup. This shall be done in order to be SNMP compliant with the MIB-II.

8.6.5 Private Enterprise RCST MIB

The private enterprise MIB structure, presented in Annex F, shall be implemented on each RCST that provides the SNMP option. The private enterprise RCST MIB is defined under the private.enterprises.dvb MIB tree. IANA has allocated the number 2 696 to it. Different system specific subtrees will be defined there. The subtree RCS for DVB-RCS systems has got the number 2 . The private enterprise RCST MIB is located under RCS with the name RCSTMIB and with number 1. A potential private enterprise NCC MIB would also be located here, as shown in the figure.

Due to some syntax rules of ASN0.1, the structure of the MIB defined in an ASCII file may show some differences (compliance group, identity module, etc). Hence, one shall take the figure as a simplified structure that reflects the core structure of the RCST MIB.

Figure 8-1 shows that the RCST MIB is made of six subgroups: *RCSTSystem, RCSTConfig, RCSTLife, RCSTActions, RCSTCallCnt and RCSTPrivatel*. The philosophy that has brought this splitting is the following. The RCST is a system and it has to be described. Moreover, it shall be controllable. And finally, it is linked to the outside world via a special interface: the Air Interface. Some of the RCST characteristics are defined during its installation, other are defined in order to get access to the system and finally several are dynamically changing and adapted during the session.

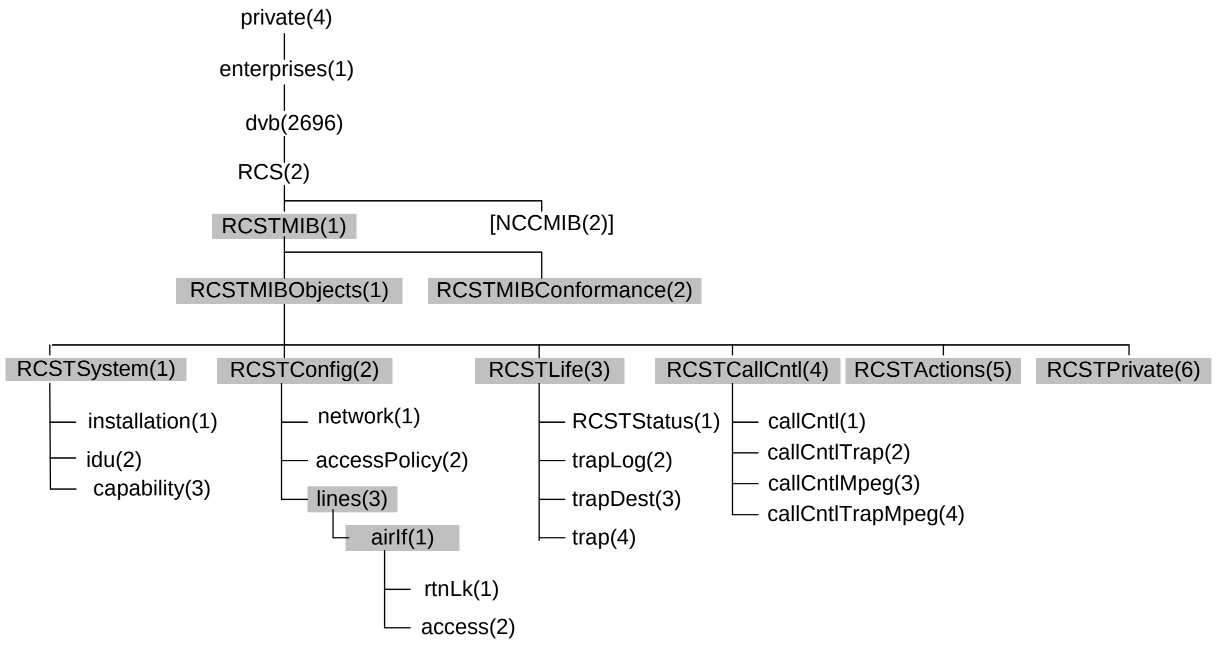


Figure 8-1: Private enterprises RCST MIB Structure

The RCST MIB has to be accessible through the following three ways: from the NCC via the OAM IP address, from the Internet (or anywhere outside/inside of the RCS network) via the NCC and the TRAFFIC IP address of the RCST and, finally, from a possible LAN behind the RCST. Note that the access via the TRAFFIC IP address is only applicable to the "wakeable" RCST. Indeed, this one has a static user defined.

8.6.5.1 RCSTSystem Subgroup

Amongst a population of RCSTs, each entity is identifiable separately. Each RCST shall be recognizable at least via the different parameters that are defined during its installation. A group called *RCSTSystem* collecting all these parameters has been created. These latter shall not change during the RCST life except in case of re-location or hardware/software updates. Hence, they can be considered as a mean to unambiguously distinguish a particular RCST in the whole population.

In this group three subgroups have been defined: capabilities, *installation* and *idu*. The first subgroup, *installation*, shall gather all the installation-related information. Those parameters shall be defined during the installation by the installer and shall not change afterwards, except in case of re-location for example. A table which size will be defined by the installer contains strings of 256 characters in order to define a (vendor specific) reference for each RCST components and provide some version number (SW, HW and the like).

The *idu* subgroup gathers physical SNMP IDU related objects, while the capability subgroup defines objects that have information which shall be given in the RCST capability field of the CSC burst, and capacity request types supported by the RCST.

To summarize, this *RCSTSystem* group can be seen as the group that hosts the static parameters, whose change is linked to either a physical modification or a functional modification of the system. These objects can also be seen as characterizing the system.

8.6.5.2 RCSTConfig subgroup

This subgroup shall be considered as gathering any data that is useful to get access, maintain and enter a session. Hence, things like synchronization, access management, interface parameters, network settings, user information and the likes are part of this group. Some aspects have been grouped under a common title, *airIf*, in the *lines* group in order to create an interface-dependent subgroup. If another kind of interface would have to be defined and used between the RCST and NCC, all its related parameters shall be defined in a specific group under the *lines* group.

A subgroup *rtnLk*, which addresses the return link, has been defined under *airIf*.

The user interface of the RCST will not be taken into account in the RCST MIB. Indeed, the different subgroups defined under MIB-II shall be used in order to describe and define this interface. In summary, the RCST user interface characteristics are considered as being captured in the *ip* and *interfaces* Groups of MIB-II.

The group, called *access,* is concerned with the critical login procedure to the Satellite Interactive network. This group lists the different timeouts and some statistics/performances about CSC slots and connection attempts. Moreover, both "normal" and "recovery-from-disaster" situations are considered. This group really gives an idea of the login process health.

Moreover, it has been considered that the network definition would basically not change that much during a normal session. Hence, the related objects (IP configuration, MAC address and the like) shall be defined under the *network* subgroup.

And finally, the *accessPolicy* subgroup shall define the access rights to the different subgroups of the RCST MIB as far as the different system entities are concerned (private users, TM, LHM, NCC…). This group is part of the *RCSTConfig* group because it shall be defined before any session can be opened and shall not be changed afterwards.

8.6.5.3 RCSTLife subgroup

Once the RCST is in session, some parameters evolve dynamically according to both RCST and users behaviour. The *RCSTLife* group shall contain parameters that characterize the dynamic state of the RCST and its components.

The most important parameters in this group are those describing the status of the RCST and its components.

Note that three subgroups – *trapLog, trapDest and trap* – are gathering all the information about notifications as far as the status of the RCST is concerned. A log table defined in *trapLog* gives the history of the 255 last notifications. The subgroup *trapDest* defines a destination (network entity) for each particular trap. And finally, the RCST status-related traps are provided in the *trap* subgroup.

8.6.5.4 RCSTCallCntl subgroup

This subgroup is composed of four smaller subgroups: *callCntl, callCntlTrap, callCntlMpeg and callCntlTrapMpeg.*

The *callCntl* group contains objects to support the call control interface between the RCST and the NCC. This group contains common objects for terminals supporting ATM and MPEG, objects applicable only to ATM and objects for end user logoff.. The MIB file for these objects will not be generally distributed as they instrument an internal control interface. Also for security reason, the NCC is the only SNMP client, which is allowed to access (remotely) objects in this group.

On the other side, the *callCntlTrap* subgroup shall collect the SNMP objects defining all the call control Traps supported by the RCST and sent to the Traffic Manager (TM) in the NCC.

The callCntlMpeg group contains objects for call control between RCSTs supporting MPEG and the NCC. The PID assignment for different types of connections is described in the group.

The callCntlTrapMpeg group defines additional call control Traps for RCSTs supporting MPEG.

8.6.5.5 RCSTActions subgroup

This MIB group contains objects a network manager can use to invoke actions and tests supported by the RCST agent and to retrieve the action/test results.

Ping, software upgrade (download) as well as others are some of the actions that can be performed from a remote platform for testing purpose, for example.

8.6.6 Harmonization and Generalization

The MIB-II *interfaces* and *ip* subgroups define the different interfaces as well as their addresses. However, in the RCST MIB, there are two kinds of IP addresses – OAM and Traffic – that could be allocated a single interface. Hence, some objects have to be defined in the RCST MIB in order to differentiate the two kinds of IP addresses.

Although the *interfaces* and *ip* tables of MIB-II are linked to each other, there is no way to differentiate between control and user traffic. Additionally, there are no counters defined other than the generic ones given in the MIB-II group. In MIB-II, the *ipAdEntIfIndex* object of *ip* (mib-II) *ipAddrTable* matches *ifIndex* of *interfaces* (mib–II) *ifTable*. These objects give a relation between an interface physical address and its IP address in the MIB-II scope.

Using both Traffic and OAM IP addresses provides a way to differentiate the two kinds of flow. Through the IP address value, a direct link can be done between MIB-II *ip* subgroup and the private enterprise RCST MIB. Then using both *ipAdEntIfIndex* (*ip* MIB-II subgroup) and *ifIndex* (*interfaces* MIB-II subgroup) values, it is possible to link any IP address to a physical interface. To do so, two SNMP objects have been defined: *RCSTConfigNetworkTrafficIpAddr* (Traffic IP address) and *RCSTConfigNetworkOamIpAddr* (OAM IP address).

Only one TRF IP address and one OAM IP address covers both the air interfaces (RT and FW) of the RCST. And in certain cases, the TRF IP address will even cover the user interface of the RCST. The two objects providing the index value for both kind of traffic allow to determine what kind of traffic is passing through which interface and has which IP address. They can be considered, as some pointers defined to map traffic types to interfaces, and to map interfaces to IP addresses.

Another feature that shall be implemented through this private MIB is the following. For each interface, it shall be possible to trace through the SNMP objects, which protocols are used, how many packets are received/transmitted, which port numbers and interfaces are used.

8.7 Forward Interaction Path Descriptor

The DVB standards for interactive services require a Forward Interaction Path and a Return Interaction Path, which carry unicast traffic. The Forward Interaction Path Descriptor provides parameters that an RCST needs for finding on the forward link its unicast traffic of the Forward Interaction Path.

The Forward Interaction Path Descriptor, which is defined in Table 12, 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 12: Forward Interaction Path Descriptor

Syntax	No. of bits		Information Mnemonic
	Reserved	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**
}			
}			
}			

*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. 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. Here, the value is 0xAD.

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 unicast traffic of the Forward Interaction Path.

9 Security, identity and encryption

Clause 9.4 of the normative document [1] describes an optional security mechanism, which provides link layer security. An example for a security and authentication concept, which takes into account the common practise in the Internet environment, can be found in Annex G.

An example of end user authentication using RADIUS between RCST and NCC/ISP can be found in Annex H.

10 RCST implementation guidelines

10.1 Architecture

The RCST should comply with the architecture outlined in Figure 10-1. An RCST conceptually consists of the Outdoor Unit (ODU), the Interfacility-Link (IFL), and the Indoor Unit (IDU).

The ODU is composed of the following subsystems: Antenna Subsystem (ANT), Transceiver (TRX), and Mechanical Subsystem (MECH). The Interfacility Link (IFL) is a cable assembly, which interconnects the IDU with the ODU.

The ANT consists out of the reflector(s) and a combined transmit/receive feed. Optionally the ANT may also include an additional receive feed for receiving from a different orbital location. The receive (Rx) part of the TRX includes the Low Noise amplifier(s), frequency downconversion and polarization as well as frequency band selection. The transmit part (Tx) of the TRX performs frequency upconversion as well as power amplification. The MECH attaches the ODU to a firm structure and provides means for accurate pointing.

The IDU consists of the following subsystems: Network Interface Unit (NIU), User Interface Unit (UIU), Power Supply Unit (PSU) and Packaging. These subsystems can be implemented e.g. in a standalone IDU, within a desktop PC or Set-Top Box.

The UIU is the interface between all receive/transmit elements of the IDU and the user device.

The NIU is constituted of at least one forward link receiver for reception of the forward link signalling (and the Traffic sent on the same Transport Stream), a transmit chain for transmission of Traffic and forward link signalling to the ODU, and all the necessary controlling elements. If only one forward link receiver is available Traffic and forward link signalling must be received from the same Transport Stream. Additional forward link receivers allow the transmission of Traffic and forward link signalling on different transport streams. This results in significant improvement of operational flexibility and should be the preferred solution. The number of available forward link receivers is a parameter exchanged between the RCST and the NCC during RCST logon.

The conceptual split between IDU and ODU functionality as described above, and specified in the present document represents only one possible separation of functions. There may be also different approaches providing the same overall functionality.

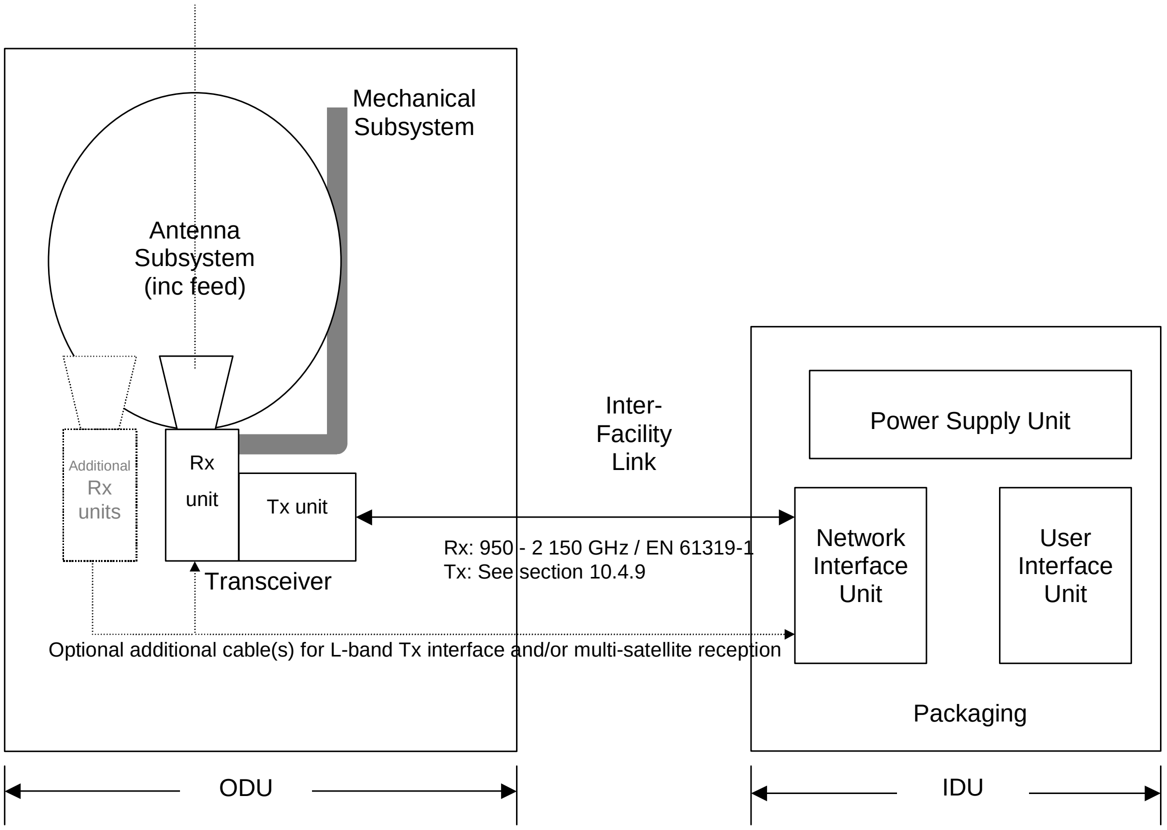


Figure 10-1: Conceptual RCST Architecture

10.2 System performance

10.2.1 RF/IF Performance

The RF parameters have been selected to comply with the conditions identified in the ERC decisions related to Exemption from Individual Licensing of Satellite User Terminals (SUTs), Satellite Interactive Terminals (SIT) and Very Small Aperture Terminal (VSAT). These ERC decisions then make reference to the Harmonized Standards as follows:

- CEPT/ERC/DEC/(00)03 [31], ERC Decision of 27 March 2000 on Exemption from Individual Licensing of Satellite Interactive Terminals (SITs) operating within the Frequency Bands 10,70 - 12,75 GHz Space-to-Earth and 29,50 – 30,00 GHz Earth-to-Space. Refer to EN 301 459 [11]

- CEPT/ERC/DEC/(00)04 [32], ERC Decision of 27 March 2000 on Exemption from Individual Licensing of Satellite User Terminals (SUTs) operating within the Frequency Bands 19,70 - 20,20 GHz space-to-Earth and 29,50 - 30,00 GHz Earth-to-space. Refer to EN 301 459 [11]

- CEPT/ERC/DEC/(00)05 [33], ERC Decision of 28 March 2000 on Exemption from Individual Licensing of Very Small Aperture Terminals (VSAT) operating in the frequency bands 14,0 - 14,25 GHz Earth-to-space and 12,5 - 12,75 GHz space-to-Earth. Refer to EN 301 428 [12]

In addition, these ERC decisions add the following requirements:

- an EIRP up to 50 dBW,

- max. 2 Watt at the antenna horn,

- 500 m from airport perimeter fences.

The nominal EIRP capability of the RCSTs should be typically in the range of 40 .... to 50 dBW.

When operating at the nominal EIRP, the spectral regrowth shall not exceed –20 dB. Spectral regrowth is defined as the ratio of the power in an adjacent channel of bandwidth 1,35 x symbol rate to the power in an equivalent bandwidth centred on the transmit carrier.

NOTE: The frequency separation between the adjacent channel and transmit channel is system dependent.

In addition to the requirements given in [1] the RCST shall operate within the requirements outlined in Table 13. In this table the overall requirements for the RCST have been split by IDU/ODU where relevant. However, since this split is heavily implementation dependent, the split is made for the three different options identified in Clause 10.3 and Table 18. In all cases the sum of the IDU/ODU contributions must be within the overall RCST requirements

Table 13: RCST Transmit Performance Requirements

Item	Description	Overall RCST requirement	ODU/IDU design Recommendations		
1	Transmit Frequency Step Size	10 Hz	not applicable		
2	Transmit Frequency Settling time	Within the hopping range this parameter shall be within the burst guard time interval. When the hopping range is exceeded the settling time shall be below 1 s.			
3	SSB Phase Noise (Tx signal)		Option 1 ODU/IDU	Option 2a ODU/IDU	Option 2b ODU/IDU
	10 Hz	≤ -16 dBc/Hz	-16 / -28*	-16 / -28*	-22 / -22*
	100 Hz	≤ -54 dBc/Hz	-54 / -66 *	-54 / -66 *	-60 / -60*
	1 kHz	≤ -64 dBc/Hz	-64 / -76 *	-64 / -76 *	-72 / -69*
	10 kHz	≤ -74 dBc/Hz	-74 / -86 *	-74 / -86 *	-82 / -75*
	100 kHz	≤ -89 dBc/Hz	-89 / -101*	-89 / -101*	-94 / -91*
	> 1 MHz	≤ -106 dBc/Hz	-106/ -118*	-106 / -118*	-109/ -109*
	NOTE: It is assumed that the combined effect of other phase noise sources in the transmission path (hub and satellite included) is at least 10 dB better.		* dBc/Hz	* dBc/Hz	* dBc/Hz
4	Amplitude Variation (Tx signal)		Option 1 ODU/IDU	Option 2a ODU/IDU	Option 2b ODU/IDU
	In any 200 kHz band	< 0,7 dB p-p		<0,2 / <0,5	<0,2 / <0,5
	In any 2 MHz band	< 0,9 dB p-p		<0,4/<0,5	<0,4/<0,5
	In any 3 MHz band	< 1,0 dB p-p		<0,5/<0,5	<0,5/<0,5
	In any 20 MHz band	< 2,0 dB p-p		<1,2/<0,8	<1,2/<0,8
	In any 40 MHz band	< 3,0 dB p-p		<1,5 /<1,5	<1,5 /<1,5
5	Group Delay Variation (Tx Signal) In any bandwidth equal to the highest symbol rate	< 0,2/(max. Symbol rate) p-p			

The I/Q amplitude imbalance shall be < 0,5 dB . The maximum misalignment between I and Q symbols shall be 5 % of a symbol period .

Over Ka band return channels, with at least one SYNC per second (used mainly for power monitoring control), an RCST shall meet the indicated transmit performance specifications.

Within the transmit band the RCST shall meet the spurious radiation such that for each spurious signal that it transmits outside the nominated bandwidth, the total EIRP of each spurious signal shall not exceed a level of 60 dB below the total EIRP of the transmitted carrier (modulated or unmodulated). Within the transmit band the transmission enabled RCST shall not generate a Noise EIRP density (dBW/Hz) exceeding:

Nominal EIRP (dBW) – 122 dBHz

outside the nominated bandwidth. In the transmission disabled state the limits shall be 30 dB more stringent.

Outside the transmit band the RCST shall meet the requirements specified either in EN 301 459 [11] or EN 301 428 [12].

10.2.2 AWGN code performance

The code performance given here below is related to the case where the coder and decoder simulation models are placed back to back (in baseband) together with an additive white Gaussian noise (AWGN) channel. Neither the satellite channel nor the RF impairments are considered.

10.2.2.1 Concatenated coding performance

Table 14 provides examples of indicative values of the performance achievable with the concatenated coding. The values are applicable to a system that employs soft-in/hard-out Viterbi decoding of the inner code, with 3-bit input soft decisions, and an algebraic decoder of the outer code. The indicated values do not include any allowances for modem synchronization effects or implementation imperfections.

The term E_b/N_0 refers to the energy per information bit (440 and 1 504 bits, respectively, for the two packet sizes indicated).

The PER (Packet Error Ratio) is the fraction of the transmitted packets (bursts) that contain at least one information bit in error after decoding. The BER (Bit Error Ratio) is the fraction of all information bits that are in error after decoding. The following approximate relationship exists between PER and BER:

For 55-byte packet: BER ~ PER/13

For 188-byte packet: BER ~ PER/40

Table 14: Concatenated code performance

Inner code rate	PER	E_b/N_0 (55 bytes)	E_b/N_0 (188 bytes)
1/2	10^{-3}	3,8 dB	3,4 dB
	10^{-5}	4,4 dB	4,0 dB
2/3	10^{-3}	4,6 dB	4,1 dB
	10^{-5}	5,3 dB	4,8 dB
3/4	10^{-3}	5,3 dB	4,8 dB
	10^{-5}	6,1 dB	5,5 dB

10.2.2.2 Turbo code performance

The performance given below can be achieved with 4-bit quantization, 6 iterations and a SUB-MAP decoding algorithm. Further gain can be achieved by increasing the number of iterations at the expense of reduced information bit rate, at constant system clock. For the 16 bytes case, the performance is given without applying a CRC code.

The BER (Bit Error Rate) can be derived from the PER, for PER<1E-5, using the following empirical formulae:

For 53-byte packet: BER < ~ PER/70

For 188-byte packet: BER < ~ PER/300

Table 15: Performance for PER=10^{-5}

FEC	Eb/No (188 bytes)	Eb/No (53 Bytes)	Eb/No (16 Bytes)
1/3	1,9 dB	2,2 dB	3,2 dB
2/5	2,1 dB	2,4 dB	3,4 dB
1/2	2,5 dB	2,8 dB	3,7 dB
2/3	3,1 dB	3,7 dB	4,9 dB
3/4	3,8 dB	4,5 dB	5,6 dB
4/5	4,4 dB	5,3 dB	
6/7	5,1 dB	6,0 dB	

Table 16: Performance for PER=10^{-7}

FEC	Eb/No (188 bytes)	Eb/No (53 Bytes)
1/3	2,5 dB	2,9 dB
2/5	2,7 dB	3,1 dB
½	3,2 dB	3,6 dB
2/3	4,0 dB	4,6 dB
¾	4,6 dB	5,4 dB
4/5	5,3 dB	6,3 dB
6/7	6,0 dB	7,0 dB

Typical link budgets are provided in Annex C.

10.3 Interfaces

This clause describes the bi-directional communications used in the IDU/ODU IFL.

This IFL usually takes the form of a coaxial section whose nominal impedance shall be 75 Ω.

To meet different application requirements two different options for an IFL are described. The IFL protocol description is provided in Annex B.

Option 1 comprises a low TX-IF solution suited for transparent SMATV systems with hardware implementations for a return channel in the 20 MHz to 60 MHz band where S-band distribution is difficult to install. This option requires a frequency agile ODU.

Option 2 comprises a S-Band TX-IF solution which uses block upconversion. All frequency agility is done by the IDU and fast frequency hopping within 500 MHz is possible. . Within this option there are two possibilities:

- option 2a which uses a 10 MHz reference and
- option 2b which uses a 108 MHz reference (see Clause 10.3.9).

A single cable design can be used with both options. Several examples for different applications can be found in Clause 11.

In order to facilitate the use of RCST for individual or collective installation, the signals and the frequencies supporting those communications shall be compliant with the EN 50083 [14] series and EN 61319-1 [15] as far as applicable.

10.3.1 Broadcast channel and forward interaction channel

The broadcast channel and the forward interaction channel (embedded or not into the broadcast channel) are carried down the coaxial section in the standardized intermediate satellite frequency range: 950 MHz to 2 150 MHz.

The signal level at the input of the IDU shall comply with the required signal level at system outlet (Clause 5.2.1 of EN 50083-7 [14]).

10.3.2 Return interaction channel

Option 1

As mentioned in the architecture Clause 10.1, the return channel frequency agility is functionally split in two parts: a slow frequency agility within a wide frequency range in the ODU and a fast frequency hopping within a narrow frequency range in the IDU.

The return interaction channel is carried up the coaxial section within the frequency range 20 MHz to 60 MHz.

The signal level at the input of the ODU shall be automatically set to provide the required EIRP via the IDU level control and according to the information received from the NCC in the forward interaction channel.

Option 2

The S-Band 2,5 GHz to 3,0 GHz is used for the IF signal for return interaction channel. The system is open for applications with up to 500 MHz fast frequency hopping signals produced in the IDU only.

10.3.3 ODU control signal

To allow an identification of the ODU as well as some adjustments during operation an ODU control signal is used.

The ODU control signal is the same for both options and is based on EN 61319-1 [15]. The same modulation and framing structure is used.

In some systems a 22 kHz control signal may conflict with the existing TVRO equipment using 22 kHz signalling and/or a higher data rate is required. In these cases a 10,7 MHz FSK signal can be used instead. In that case the RX is controlled by the signal defined in EN 61319-1 [15] and all other ODU functions are controlled by the 10,7 MHz.

10.3.3.1 Concept of the 22 kHz Pulse Width Keying (PWK) Bus

The low data rate communication between the IDU and the ODU is based on a 22 kHz PWK signal as used by DiSEqC (ref). The impedance of the bus at 22 kHz shall be 15 Ω. A parallel inductor of 270 µH can be used to support a DC. power supply current. In this case a capacitor to ground should be supplied to shape the 22 kHz signal. The DC feeding point is grounded for 22 kHz with a capacitor. If a DC is not needed for powering peripheral devices, then in order to maintain correct operation of the DiSEqC bus, there should be a minimum of 10 V bias applied, but the inductor and capacitor can be omitted.

The control signal from every device on the bus is produced by a 43 mA current shunt producing a 650 mV signal which is monitored by every device. This amplitude of the DiSEqC carrier tone on the bus is normally too small to detect directly on a "TTL" or "CMOS" compatible pin on a microcontroller, so usually a "comparator" input, or a simple external (one-transistor) amplifier, is required. In any case, it is important not to make the input too sensitive to small-amplitude signals which may be "noise" or interference. It is recommended that the smallest amplitude normally detected is about 200 mV peak-peak. This can be achieved either with hysteresis (positive feedback applied around the comparator/amplifier) or with a DC bias offset (equivalent to about 100 mV) applied to the input of the amplifier/comparator. Hysteresis (if symmetrical) can maintain a reasonably constant 50 % duty cycle for the detected carrier tone, whilst the DC offset method may generate a less desirable asymmetric (pulse) waveform when the carrier amplitude approaches the lower limit.

All devices are connected in parallel on the bus and shall therefore have a high impedance.

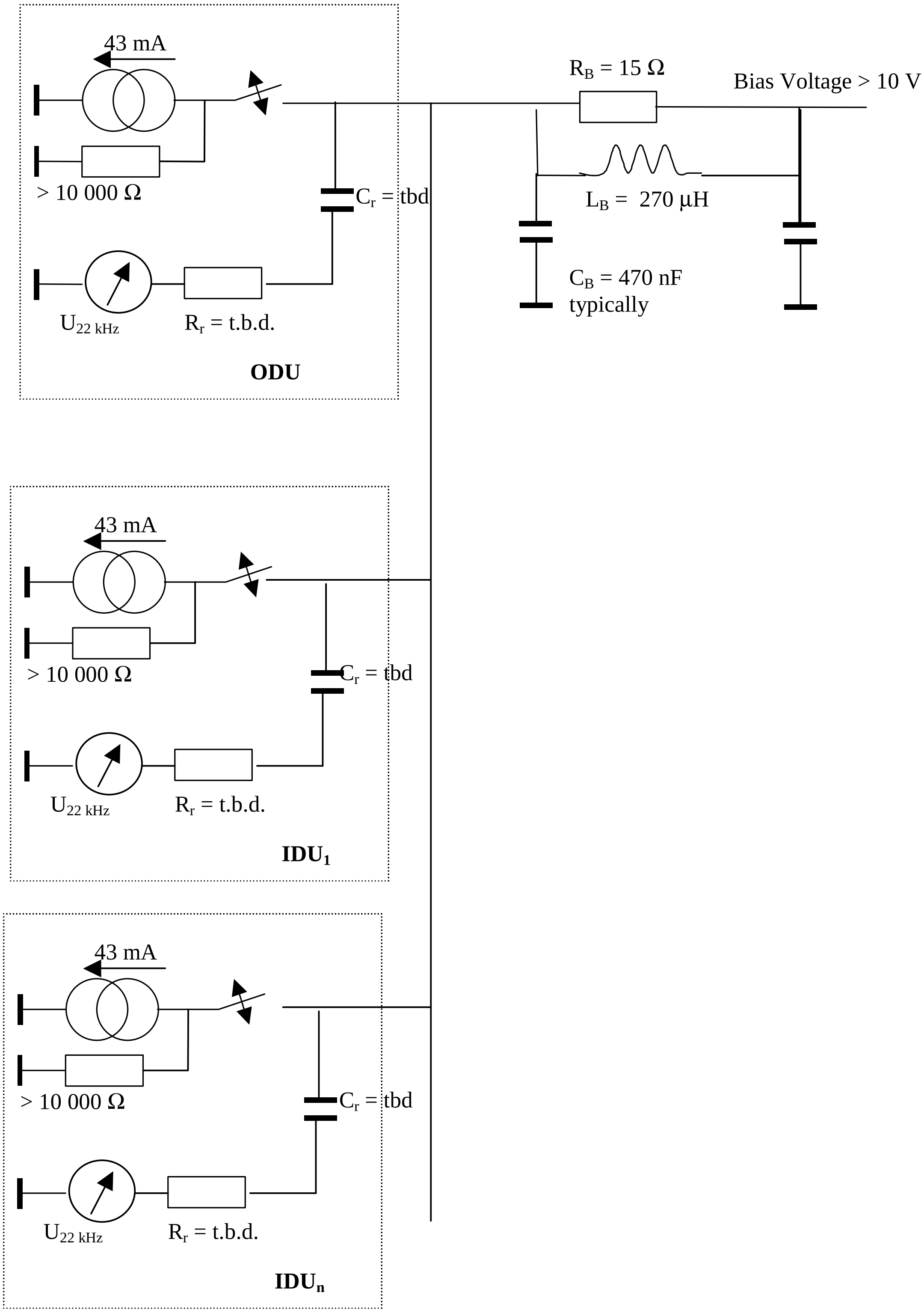


Figure 10-1: 22 kHz PWK bus concept

Specifications

carrier frequency	22 kHz ± 20 %
Bus load impedance R	5 Ω ± 5 %

DC supply

Bus load inductance L_B	270 µH ± 5 %
Bus load capacitance C_B	typically 470 nF

Current source

current amplitude	43 mA ± 10 %
source impedance	> 10 kΩ

22 kHz carrier detection device resistance R_r	typically between 5 and 10 kΩ.
DC block capacitor	typically a few nF, but depends on the value R_r, $_{Rr}$, it should be chosen so as to give a time constant of around 100 µS

bit definition

timing base	0,5 ms ± 0,1 ms
bit length	1,5 ms
"0"	1,0 ms burst + 0,5 ms pause
"1"	0,5 ms burst + 1,0 ms pause

10.3.3.2 Receive control signal via separate cable

To support applications where the transmit part and the receive part of an ODU are connected via separate cables the IDU shall be able to send a control signal to the receive part of the ODU (LNB) or to a multiswitch SMATV by using the standardized interface described in the CENELEC publication EN 61319-1 [15]. A bus concept as described in 10.4.3.1 is only needed if the extended communication protocol is used and/or multi satellite reception is required using DiSEqC switching.

Incidentally, this standardized interface also ensures the powering of the receive part (LNB) of the ODU.

10.3.3.3 Higher data rate control signal

The higher data rate communication channel is based on 2-FSK modulation of a single carrier, which is shared by all the devices of the network. The access to this carrier is based on an Aloha scheme with no collision avoidance, the communication protocol will be able to work with or without ACK.

In the communication between IDU and ODU and/or Head End Devices (HED) a master-slave approach is used where the IDU is the master. For this reason an ODU or HED can only transmit a message after a request from an IDU.

Main characteristics

Frequency:	f_0 = 10,7 MHz
Modulation:	2-FSK; Δf = 67 kHz
Bit-rate:	[up to] 100 kbit/s

NOTE: Possible harmonization with the SCTE (Society of Cable Telecommunication Engineers, USA) proposed specification will be considered.

Bit definition

"0"	$f_0 - \Delta f$	length = 10,0 µs
"1"	$f_0 + \Delta f$	length = 10,0 µs

Tx. max power level

IDU	98 dB(µV) (75 Ω)
ODU	108 dB(µV) (75 Ω)

Rx. min. power level

IDU	53 dB(µV) (75 Ω)
ODU	43 dB(µV) (75 Ω)
Tx. spectral mask:	≤ -60 dB @ f_0- 2 MHz ≤ f ≤ f_0 + 2 MHz

10.3.4 Control functions from the IDU

The following functions shall be available:

- SSPA ON/OFF (relates to disabled state as described in EN 301 459 [11])
- Tx Unit power off
- LNB control (in the case the LNB or a part of the LNB is controlled by the ODU control bus)
- Full Reset
- Software Reset
- Password Reset

The following functions may be available:

- Frequency tuning within the wide frequency range of the slow frequency agility (if applicable)
- Software update of the ODU
- Tx Frequency band selection (select different Local Oscillators)
- Modulation ON/OFF (transmit Continuous Wave)
- Set Transmit output power level
- Get ODU location data (latitude and longitude)

The transmit control signal level at the output of the IDU shall comply with the Clause 5.3.2 of EN 50083-10 [14].

10.3.5 Monitoring functions (from ODU on request)

The following functions shall be available:

- SSPA ON/OFF status
- Phase lock oscillator status
- Power supply status
- Microprocessor status
- Authentication information exchange between the ODU and the IDU
- EUI-64 of the ODU

NOTE: EUI-64 is a trademark of IEEE, it is intended to identify a single unique device independent of its functionality. The rules for allocation can be obtained from the IEEE.

- ODU manufacturing information

The following functions may be available:

- ODU temperature information for compensation
- ODU RF parameters

- LNB status (in the case the LNB or a part of the LNB is controlled by the ODU control bus)
- Get ODU location data (latitude and longitude)

10.3.6 Control and Monitoring Protocol Description

The protocol description is provided in Annex B.

10.3.7 Reference frequency

For measurement and frequency correction a reference frequency is required. This reference frequency may be generated either in the IDU or ODU and can be either 10 or in the case of option 2b 108 MHz. This reference frequency may be used for frequency measurement only or as a LO reference, in which case the phase noise requirements are more constrained (see below).

10.3.7.1 108 MHz

Frequency stability vs. temperature range and ageing: ±25 ppm.

In the case the reference is used as a LO reference then the phase noise required is as follows:

10 Hz	-65 dBc/Hz
100 Hz	-103 dBc/Hz
1 kHz	-113 dBc/Hz
10 kHz	-123 dBc/Hz
100 kHz	-138 dBc/Hz

In the case the reference is used for frequency measurements only the phase noise required is as indicated in Table 17:

Table 17: RCST RF Allan variance

Gate time T (s)	Point	Allan Variance sigma (T)
0,3	A	1E-19
3	B	1E-18

10.3.7.2 10 MHz

Frequency stability vs. temperature range and ageing: ±25 ppm.

Signal level: 108 dBμV ± 5 dB.

In the case the reference is used as a LO reference then the phase noise required is as follows:

10 Hz	-86 dBc/Hz
100 Hz	-124 dBc/Hz
1 kHz	-134 dBc/Hz
10 kHz	-144 dBc/Hz
100 kHz	-159 dBc/Hz

In the case the reference is used for frequency measurements only the phase noise required is as indicated in Table 17.

10.3.8 Powering the ODU

The supply DC voltage at the input of the transmit part shall be in the range 18 V to 30 V.

10.3.9 Overview of the different options

The following table provides an overview of the different options.

Table 18: Overview of the different interface options

<table>
<tr><td colspan="2"></td><td>Option 1</td><td colspan="2">Option 2</td></tr>
<tr><td colspan="2">RX-IF [MHz]</td><td colspan="3">950 - 2 150</td></tr>
<tr><td colspan="2">TX- IF [MHz]</td><td>20 – 60</td><td colspan="2">2 500 - 3 000 (see note)</td></tr>
<tr><td colspan="2"></td><td>Option 1</td><td>Option 2a</td><td>Option 2b</td></tr>
<tr><td colspan="2">Reference Frequency [MHz]</td><td>108</td><td>10</td><td>108</td></tr>
<tr><td colspan="2">Reference frequency signal level</td><td>70 – 80 dBµV</td><td>108 dBµV</td><td>99 dBµV</td></tr>
<tr><td rowspan="2">Phase noise of reference</td><td>reference is used for frequency measurement only</td><td>see 10.3.7.1</td><td>see 10.3.7.2</td><td>see 10.3.7.1</td></tr>
<tr><td>reference is used as LO reference</td><td>see 10.3.7.1</td><td>see 10.3.7.2</td><td>see 10.3.7.1</td></tr>
<tr><td rowspan="2">Control and Monitoring Signal</td><td>version 1</td><td>Not applicable</td><td colspan="2">extended 22 kHz PWK adapted from EN 61319 [15]</td></tr>
<tr><td>version 2</td><td colspan="3">RX: 14/18 V 0/22 kHz or 22 kHz PWK
(EN 61319-1 [15])
RX&TX: 10,7 MHz FSK</td></tr>
<tr><td colspan="2">RX Control and Monitoring Signal for separate RX-connections</td><td colspan="3">14/18 V 0/22 kHz or 22 kHz PWK (EN 61319-1 [15])</td></tr>
<tr><td colspan="5">NOTE: This range may be extended to 950 MHz to 3 000 MHz to allow for a L-band TX interface using two cable solution.</td></tr>
</table>

10.4 ODU Environmental Conditions

10.4.1 Operational Environment

There shall be no degradation of the specified system and subsystem performance when operating under the following environmental conditions:

Temperature:	-30°C to +50°C
Solar Radiation:	500 W/m² max.
Humidity:	0 - 100 % (condensing)
Rain:	up to 40 mm/h
Wind:	up to 45 km/h

The ODU mechanical construction shall make sure there are no random vibrations during wind conditions which cause any performance degradation.

10.4.2 Survival Conditions

The ODU shall be allowed to degrade in performance, but shall maintain pointing accuracy and shall suffer no permanent degradation under the following environmental conditions:

Temperature:	-40 to +60°C
Solar Radiation:	1 000 W/m²
Humidity:	0 - 100 % (condensing)
Precipitation:	up to 100 mm/h of rain or 12 mm/h of freezing rain or 50 mm/h of snowfall
Static load:	25 mm of ice on all surfaces
Wind:	up to 120 km/h

Storage & Transportation:

Temperature: -40 to +70°C

Shock and Vibration: as required for handling by commercial freight carriers

11 User network guidelines

11.1 RCST interaction with cable and non-transparent SMATV

This clause describes a possible design for the interaction of the RCST with cable networks or non-transparent SMATV installations. Figure 11-1 depicts an example of such an interaction.

An RCST is connected to the head-end of a cable system via an IP network (typically Ethernet). At the end-user's premises, the cable network is terminated by a cable modem.

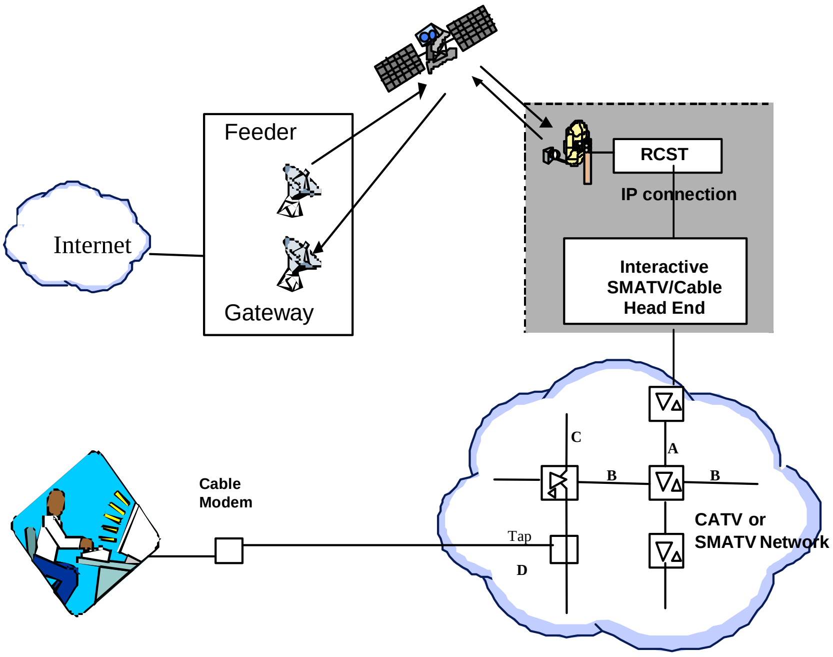


Figure 11-1: RCST interaction with a CATV or SMATV network

The cable modem can be integrated in an interactive set-top box or can be used with a PC. The interactive IP traffic will typically be combined with the distribution of TV broadcast signals on the cable distribution network. In the case of Satellite Master Antenna TV (SMATV) the TV signal is also received via satellite.

The local network will require its own network management. The management of e.g. the set top boxes and cable modems is not the task of a DVB-RCS NCC. Some services might have to be offered by the network management system in order to allow the correct operation of the network.

DVB has standardized the DVB interaction channel for Cable TV distribution systems (CATV) in ES 200 800 [16] (previously ETS 300 800) and also provides guidelines for the use of such systems in TR 101 196 17 [17]. A DVB interaction channel via satellite in non-transparent SMATV networks is described in TR 101 201 [18].

In principle, any locally managed network that offers interactive communication can be connected to an RCST over an IP connection.

11.2 Transparent SMATV

This clause describes the implementation of very cost effective solutions matched to the SMATV interactive networks.

Transparent SMATV means: no transmodulation for the forward signals, nor for the return signals; the same modulation is carried through the whole SMATV network.

Transparent SMATV means also: compliance of frequencies, bandwidths and levels with the SMATV European standard EN 50083 [14] and SMATV control channel specifications .

This concept of transparency allows the use of the same models of IDU in collective installations as in individual installations.

The SMATV transparency is already achieved for the forward signals in existing SMATV-IF installations; indeed, in such installations, the same satellite receivers can be used as in individual satellite reception. So, the solutions described in this clause are only some extensions of the SMATV-IF existing solutions.

The ODU used for SMATV-IF installations could be equipped with a 4 output LNB powered by a head-end component.

There are two ways to distribute the satellite channels in SMATV-IF installations:

The first way is to use passive components (splitter, tap-off) in the common parts of the network. Those components are connected between themselves with only one cable. In order to maximize the number of attractive channels, 99 % of those installations are equipped with IF/IF channels reorganizers at the head-end.

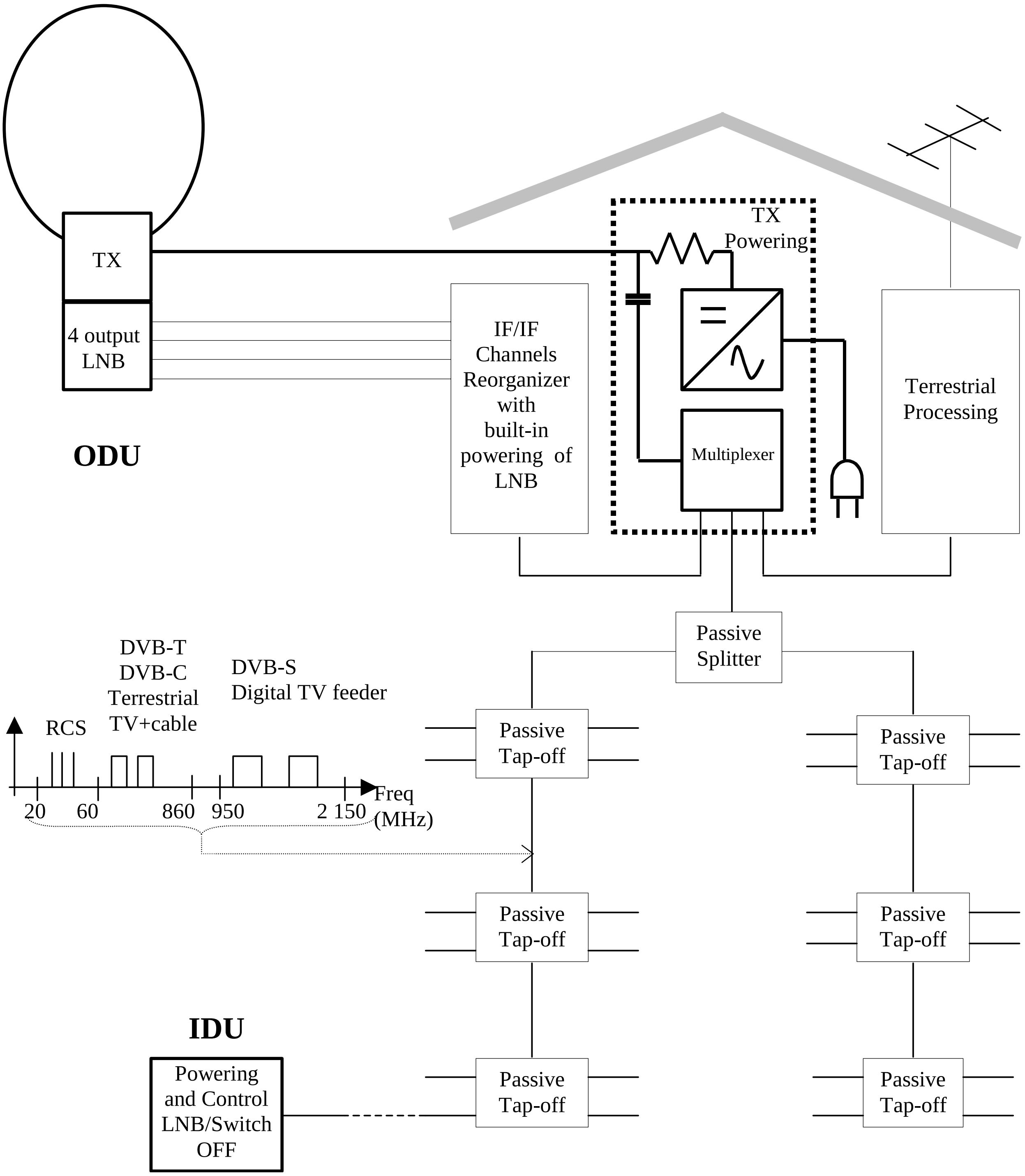
The second way is to use multiswitches in the common parts of the network. Those multiswitches are connected between themselves with five cables (nine in some cases). One of those cables carries the terrestrial channels (with or without return path implementation) and the others, connected to a four output LNB (two in case of nine cables) carry the satellite channels. Each end-user is connected to a multiswitch with only one cable.

It is foreseen that a CATV return channel network and a DVB-RCS return channel scheme will not coexist on the same SMATV installation (same cables). They can coexist with one more cable in the SMATV installation.

11.2.1 Interactive "one cable" SMATV-IF installation

This type of installation is schematized in Figure 11-2. Most of those installations are already installed with passive components (splitter and tap-off) working from 5 MHz to 2 150 MHz.

To up-grade such an installation for satellite interactivity, it suffices to replace the existing dish by an ODU and to add a component connected to the head-end for filtering the signals and powering the ODU.



NOTE: In case of existing SMATV installation which have to be up-graded for satellite interactivity, the components to be replaced or added are drawn in bold lines.

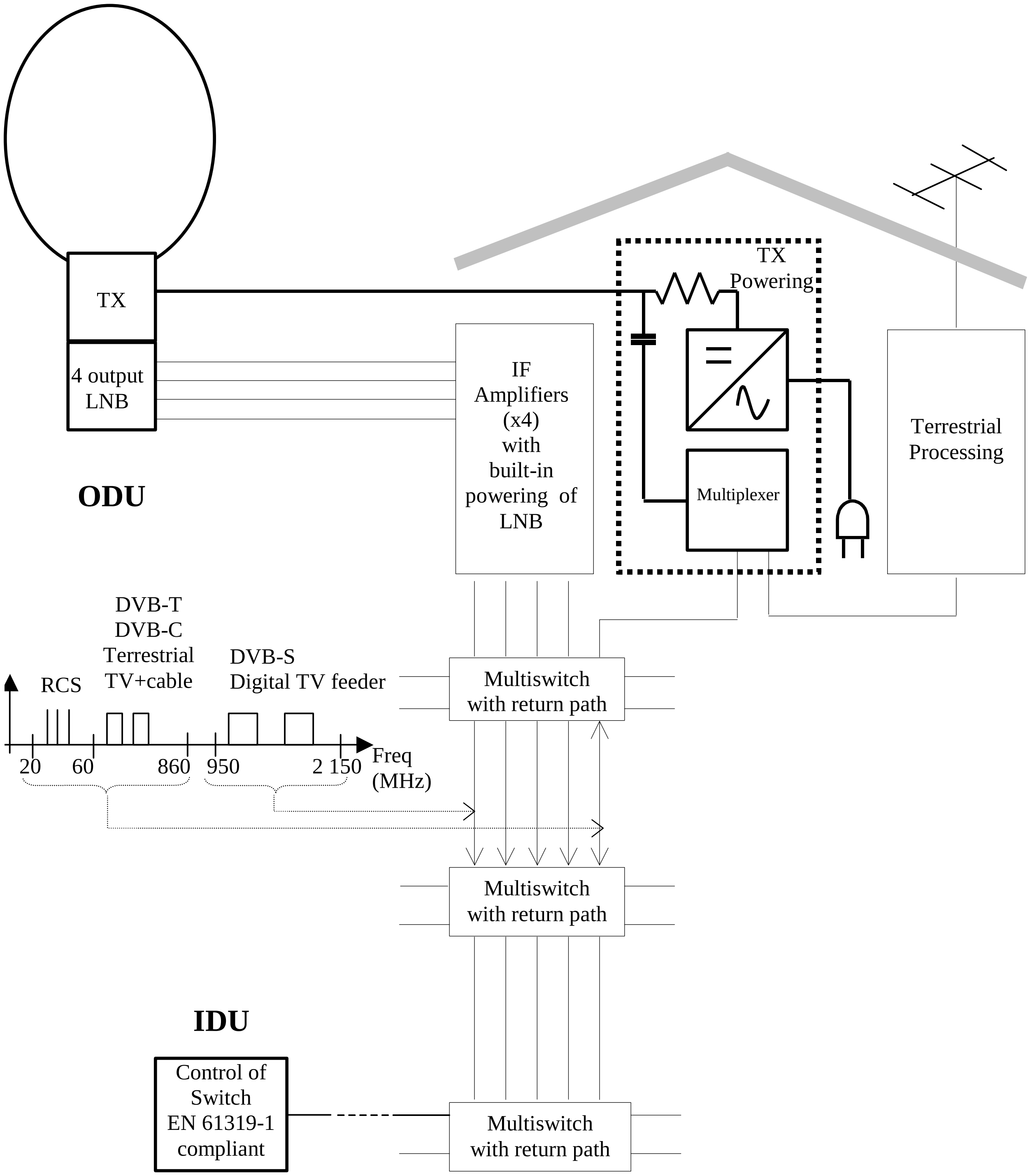
Figure 11-2: Interactive "one cable" SMATV-IF installation

11.2.2 Interactive "multiswitches equipped" SMATV-IF installation

Two main types of multiswitches are used for those installations: with and without built-in 5-65 MHz return path.

The installation using the first type (built-in return path) is shown in Figure 11-4.

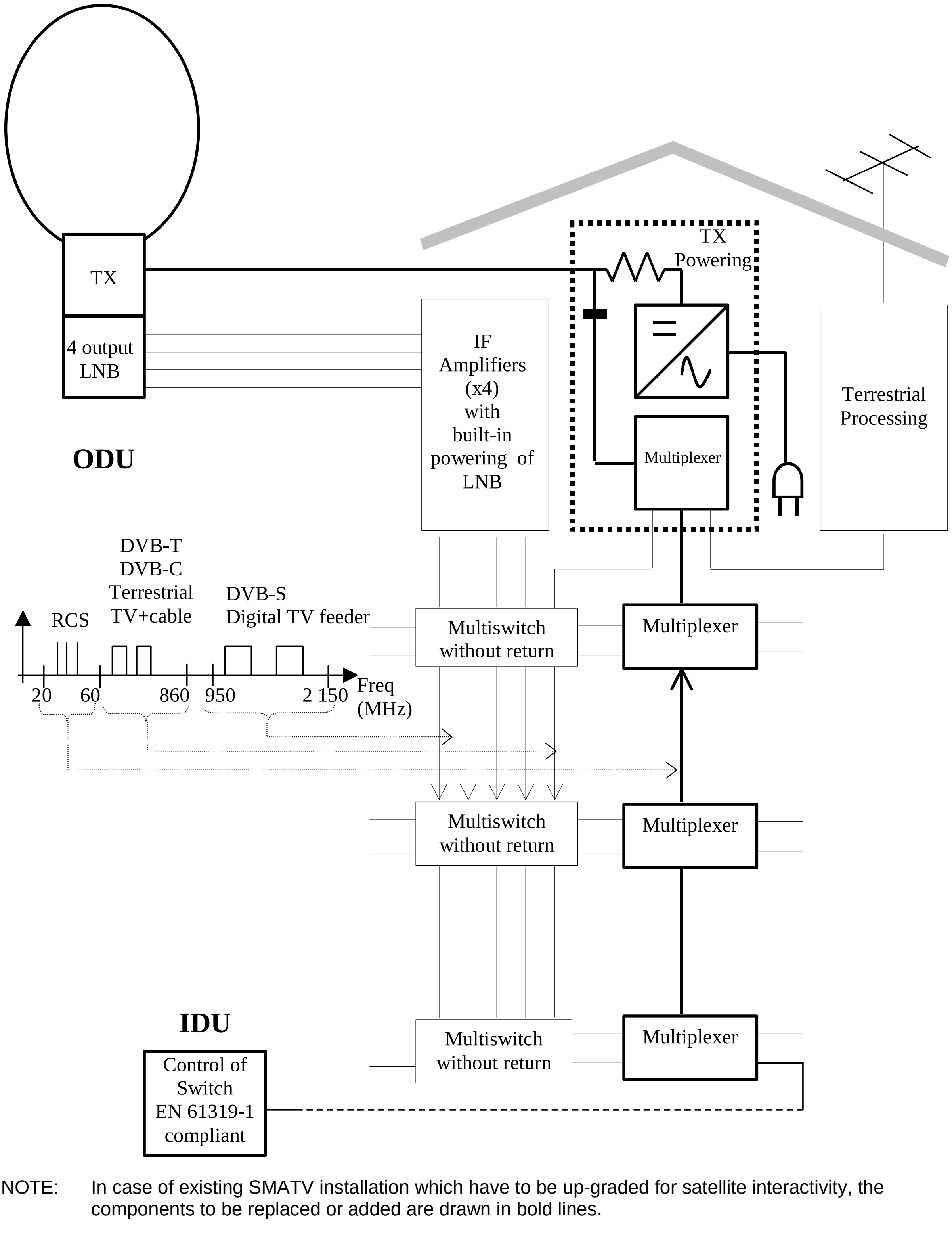
To up-grade such an installation for satellite interactivity, the existing dish could be replaced by an ODU, and a new component for filtering the signals and powering the ODU could be added at the head of the network.


TX
4 output LNB
ODU
IF Amplifiers (x4) with built-in powering of LNB
TX Powering
Multiplexer
Terrestrial Processing
DVB-T
DVB-C
Terrestrial TV+cable
DVB-S
Digital TV feeder
RCS
20
60
860
950
2 150
Freq (MHz)
Multiswitch with return path
Multiswitch with return path
IDU
Control of Switch EN 61319-1 compliant
Multiswitch with return path

NOTE: In case of existing SMATV installation which have to be up-graded for satellite interactivity, the components to be replaced or added are drawn in bold lines.

Figure 11-3: Interactive "multiswitches equipped" SMATV-IF installation (multiswitches with built-in return path)

The installation using multiswitches without built-in return path is schematized in Figure 11-3. More additional components are needed to up-grade such an installation than the previously described installation, particularly some multiplexers connected between the multiswitch outputs and the end-user connection.



NOTE: In case of existing SMATV installation which have to be up-graded for satellite interactivity, the components to be replaced or added are drawn in bold lines.

Figure 11-4: Interactive "multiswitches equipped" SMATV-IF installation (multiswitches without built-in return path)

11.3 RCST interaction with local area networks

This clause describes the interworking between a RCST and a Local Area Network (LAN). An RCST has to include an interface to the user network. The complexity of this interface depends on what kind of technology is used in the LAN. Figure 11-5 shows the general system.

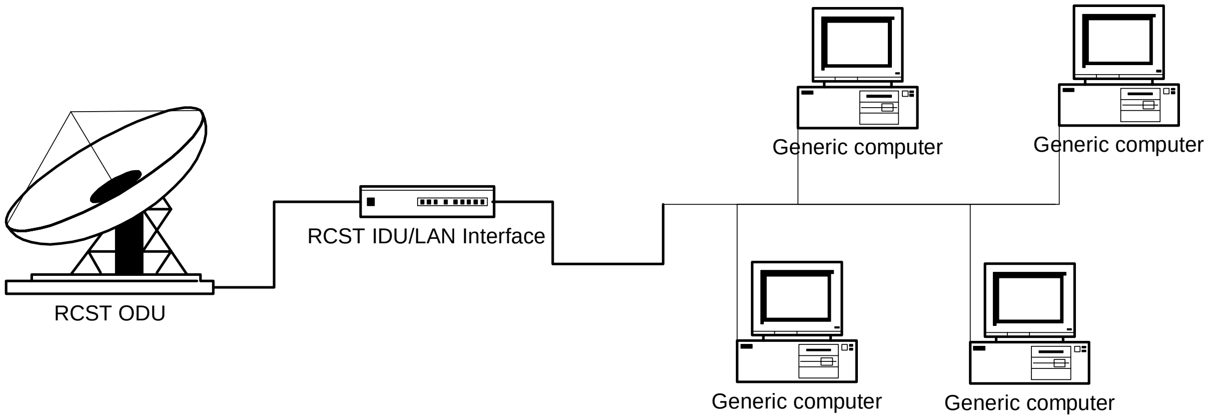


Figure 11-5: General system

The DVB-RCS system supports two types of data traffic, IP (required) and native ATM (optional). In short, conventional IP traffic is most widespread and is probably the cheapest option to implement in this system. On the other hand - native ATM provides services that an IP-based network does not support. The RCST indicates what kind of data traffic it supports during logon using the ATM-connectivity field in the CSC-burst. NCC confirms data transport mode in TIM.

Conventional IP mode is illustrated in Figure 11-6. On the forward link, data in IP-packets are encapsulated and sent in MPEG2-TS packets. The IDU/LAN-interface decodes the packets and sends them as Ethernet frames addressed to the different user entities on the network. On the return link, the user entities send IP data in Ethernet frames to the RCST. The RCST has the choice either sending data encapsulated in ATM TRF burst using the AAL5 segmentation and reassemble function, or optional MPEG2-TS –packets.

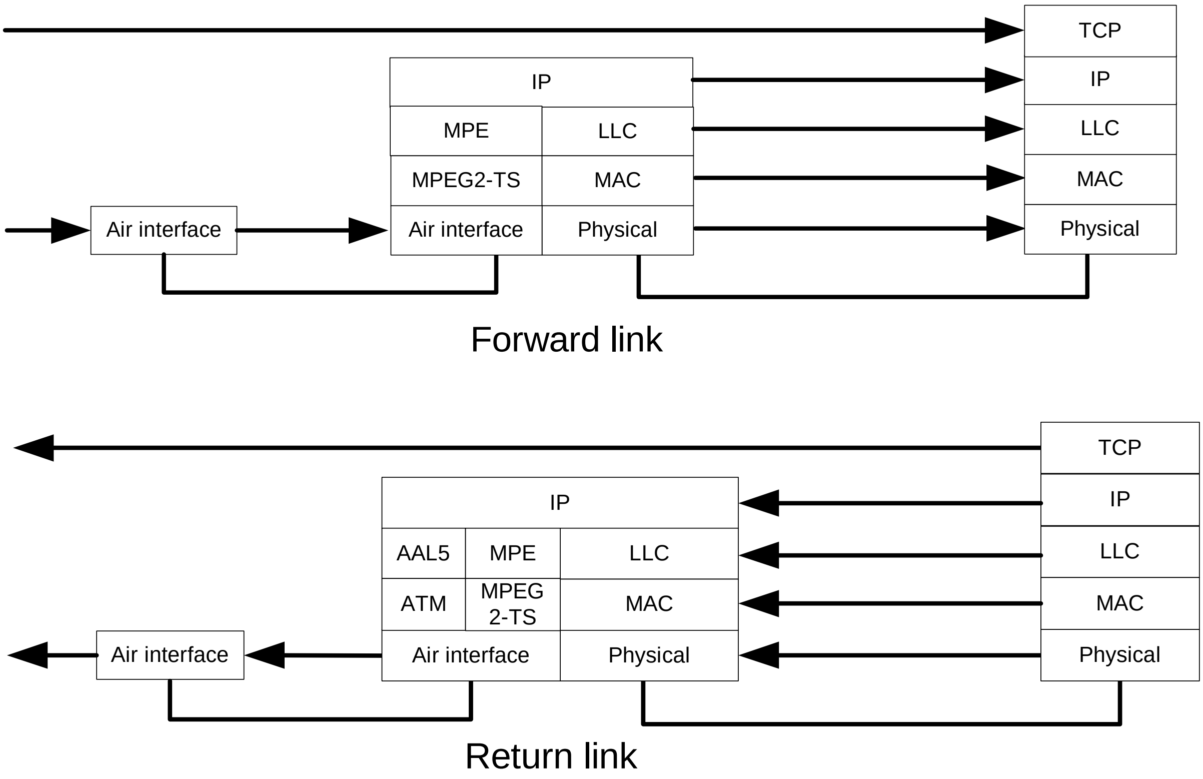


Figure 11-6: Conventional IP mode

The optional native ATM support enables a genuine ATM connection from an end-point on the network to the outside world. Thus, the special capabilities in the ATM system, like end-to-end QoS, may be utilized. During logon to the RCS-network, NCC assigns VPI/VCI identificators in TIM to the RCST for signalling purposes. These indicators replace the standard values of 0/5 so signalling to and from each RCST may be identified. To smooth the signalling process between the user entities and the RCS gateway, the RCST must translate the VPI/VCI signalling identifiers on the user network (0/5) to and from the values assigned by NCC. The user entities may then set up VCCs to the outside world using ATM signalling.

Figure 11-7 illustrates a network with native ATM support. On the forward link, ATM-cells are encapsulated in MPEG2-TS-packets according to TR 100 815 [19]. The RCST/network interface decodes the packets, strips off the MPEG-header and transmits the ATM-cells directly to the recipient. On the return link, the user entities send ATM cells to the RCST, which transmits them directly through the RCS-network using ATM TRF bursts.

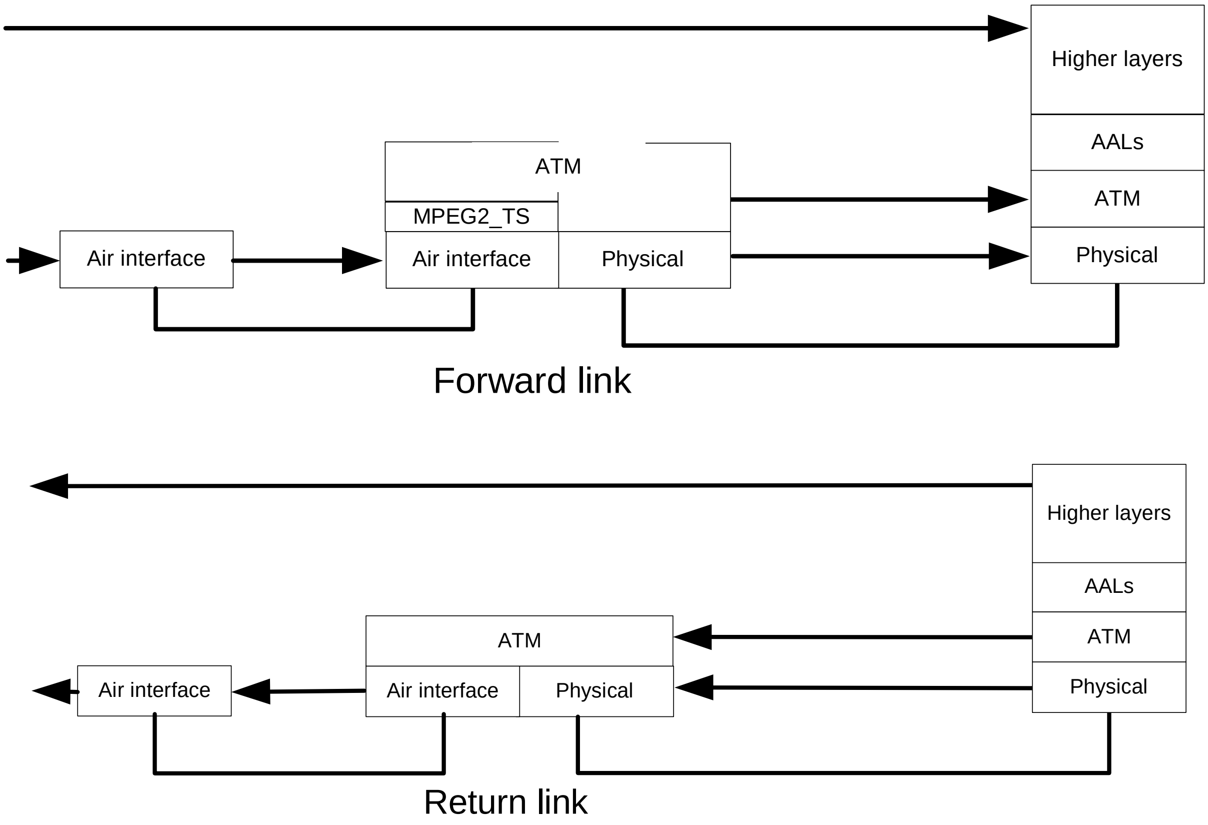


Figure 11-7: Native ATM support

In practice, there are not many user networks supporting the ATM network protocol. The current trend is that ATM is increasingly becoming a technology for backbone networks and WANs. Therefore, this capability will probably not be used to a large degree when interfacing towards a user network. To exploit the ATM support of an RCST on a traditional IP-based LAN in a beneficial way, overlay technology is required. One example is LAN Emulation (LANE) as described in [13] and illustrated in Figure 11-8. LANE provides an interoperable transition from existing LANs to ATM. Ethernet frames are divided into ATM cells and are transmitted through the ATM network. Using this kind of technology requires a LAN emulation configuration server (LECS), a LAN emulation server (LES) and a Broadcast and unknown server (BUS) to be built into the RCS-gateway. These components handle tasks such as logon to the ATM network, broadcasting and multicasting and address mapping. Each RCST supporting LANE on the RCS-network requires a LAN emulation client (LEC). This client sets up control connection to the LAN emulation servers and maps MAC addresses to ATM addresses.

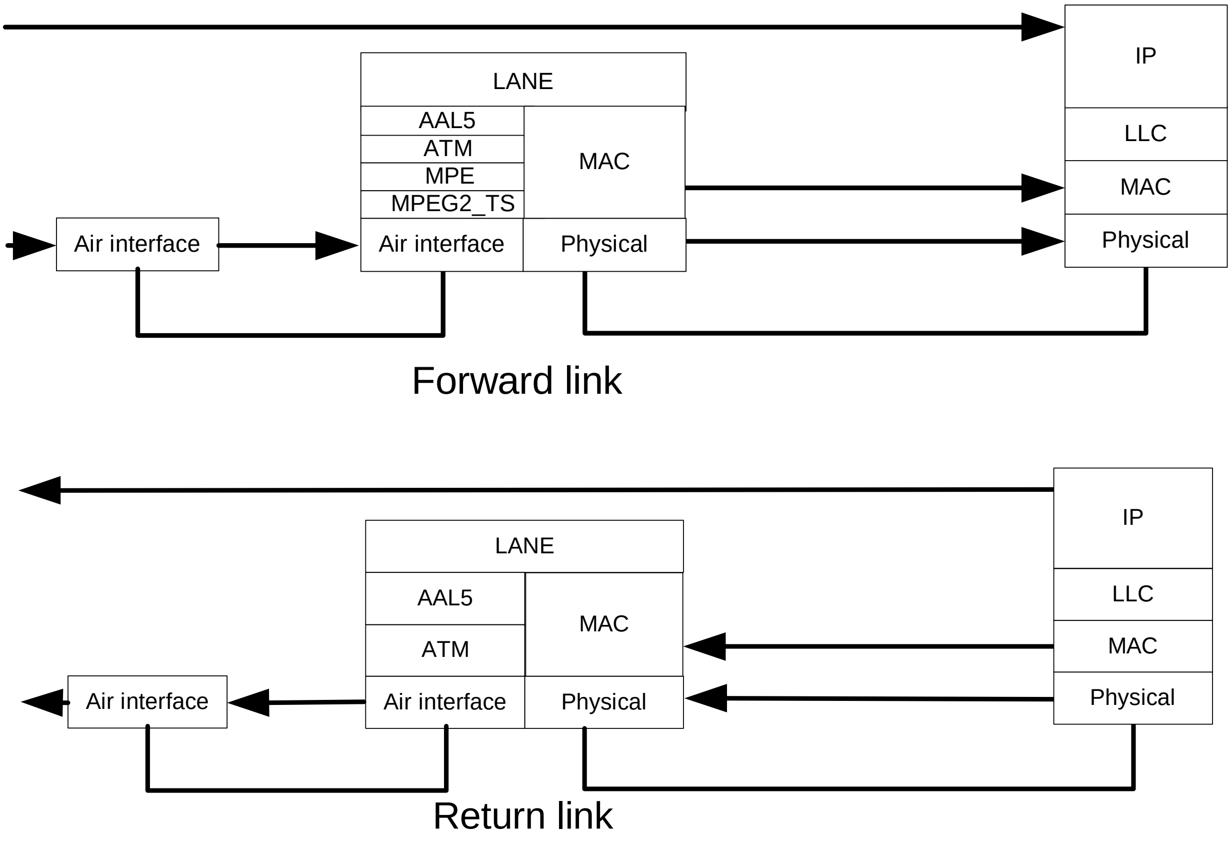


Figure 11-8: LAN emulation example

One important design issue in the interface is the buffer sizing. It is important for the ease of operation to design the network interface in a way that makes the satellite network transparent for the users of the LAN. Timing data transmission to fit in available time-slots should therefore be a task for the RCST. Thus, if two stations on the network transmit data simultaneously, data from one of the station has to be put in a buffer at the interface until the RCST has the opportunity to send a new burst. This applies both when using IP or ATM as data transport.

11.4 RCST interaction with IHDN

The DVB Home Local Network specification [20] defines how to implement an IEEE 1394 based home network in conjunction with DVB distribution and interaction networks. In conjunction with DVB-RCS the RCST is part of a residential gateway.

Annex A: Examples of incorporation of satellite based return channel into a digital television platform

This clause describes on a high level one way of integrating the DVB-RCS return channel into an already DVB compliant digital television platform. Figure A-1 shows such a system. Data connections are marked with solid lines, and control (Ethernet) connections are dashed lines.

Feeder/Gateway. As far as the existing platform is concerned, this will be another TV (MPEG-2) channel. The feed to the multiplexers will be e.g. a PC connected to the Internet through a number of high-speed connections. The maximum number of requests to an external network (Internet, say) is assumed to be implementation dependent. Capacity assignments for the individual RCST will have to be taken into account when, say IP-packets, are packed inside the MPEG-2 transport stream. In addition the total forward bit rate for this unit will be controlled from the NCC and in the last instance from the SYSTEM CONTROLLER. The total forward bit rate should be specified as a constant or minimum amount to allow for a guaranteed bit rate for an individual RCST.

Receiver/Demodulator. RX can hopefully use the same antenna as the forward data, and provide synchronization before the signal is demodulated. Both these tasks are controlled from the NCC.

NCC. Controls the timing, synchronization and other control parameters for the whole return channel system. This is again assumed to be under supervisory control from the SYSTEM CONTROLLER, which has direct control over the bit rate for all feeders. Exactly how this interaction is going to be realized is not clear at this stage. The NCC and SYSTEM CONTROLLER may be integrated when and if the system is realized, and this is partly done by assuming the Gateway/Feeder unit to be controlled by the NCC via the SYSTEM CONTROLLER. The NCC is thought to control the generation of SI data through the connection to the SI generator.

Since this is an MPEG system it implies that any IP traffic is transmitted inside 188-byte MPEG frames.

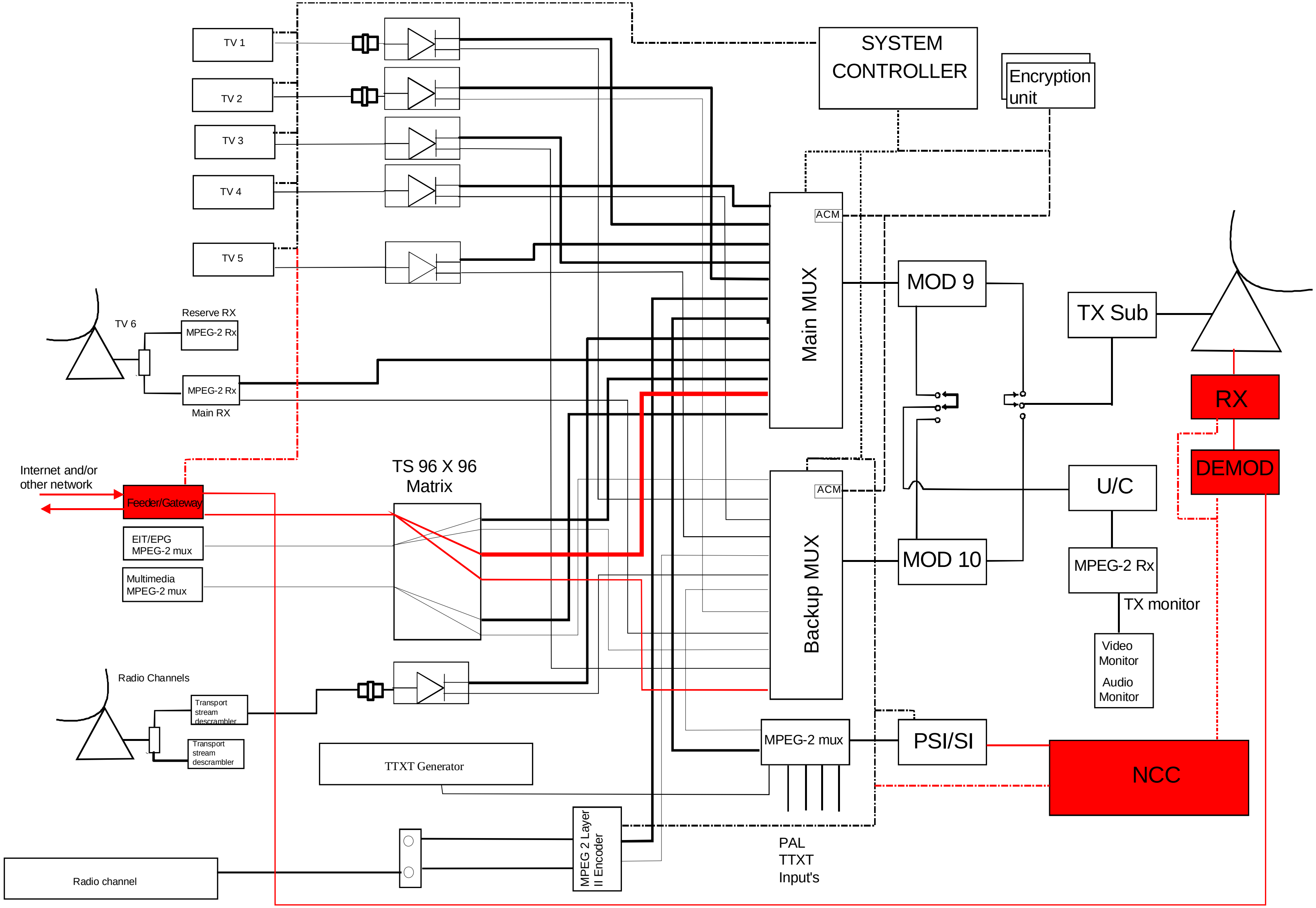


Figure A-1: Example of incorporation of satellite based return channel into a digital television platform

Annex B: RCST IDU/ODU IFL protocol description

The protocol between the IDU and ODU will be Master/Slave. The IDU shall act as the Master and the ODU shall act as the Slave. This protocol is based on an extension of the DiSEqC™ bus specification Version 4.2 which will be updated (to Version 5.0 in due course) to include the new commands described in the present document.

B.1 Command and Request Processing

Only one command or status request can be processed at a time. Once the IDU has issued a command or status request to the ODU, a new command cannot be issued until the IDU has received a valid response (ACK or NACK) or the command has timed out. In the case of either a NACK or time-out, the IDU may issue a given command up to three times before declaring a fault on the interface.

B.2 Alarms

When a hardware alarm occurs within the ODU, the ODU shall:

1) disable the SSPA to inhibit transmission by removing power to the Tx circuit,
2) disable the frequency reference signal provided to the IDU (if implemented),
3) buffer the fault indication until read or cleared by the IDU.

After detecting the ODU fault (via loss of reference, hub report of abnormal logoff or time-out of command request), the IDU shall send a status request message to the ODU to identify the type of alarm. This request has a higher priority than other message traffic. For instance, if an ODU alarm is detected, and the IDU is about to issue a command, the command is delayed and the status request issued.

The IDU may ignore the Frequency Reference shut down during RCST installation.

B.3 Dynamic behaviour

Unless otherwise specified the subsequent clauses, the ODU shall respond to a command or request received from the IDU within the allowable timeout period (TOODU) of 150 ms. In general, to force the ODU to respond immediately to a command/request from the IDU, the DiSEqC command 0x01 may be sent after any power-on or (re-) initialization procedure.

NOTE: DiSEqC devices will default to a random response time between 15 ms to 115 ms, and even 135 ms in case of collision avoidance.

The TOODU may be disabled during installation.

B.4 Error recovery mechanism

When the IDU does not receive its expected answer (no answer or NACK), it may re-send the message twice, after which an alarm shall be raised. From the ODUs point of view, there will be no limitations on the number of times that the IDU can attempt to send a message. If the ODU keeps receiving messages in error, it will continually respond with the error code. If there is an invalid password, it will "lock-up" after the sixth attempt (see Clause B.5.4).

B.5 Message level description

The Monitoring and Control Protocol message is depicted in Figure B-1. A dedicated protocol will be used for extended messages longer than 8 bytes (e.g., software downloads). It is described in detail in Clause B.5.5. Bytes are transmitted MSB first, and each byte is followed by an odd parity bit.

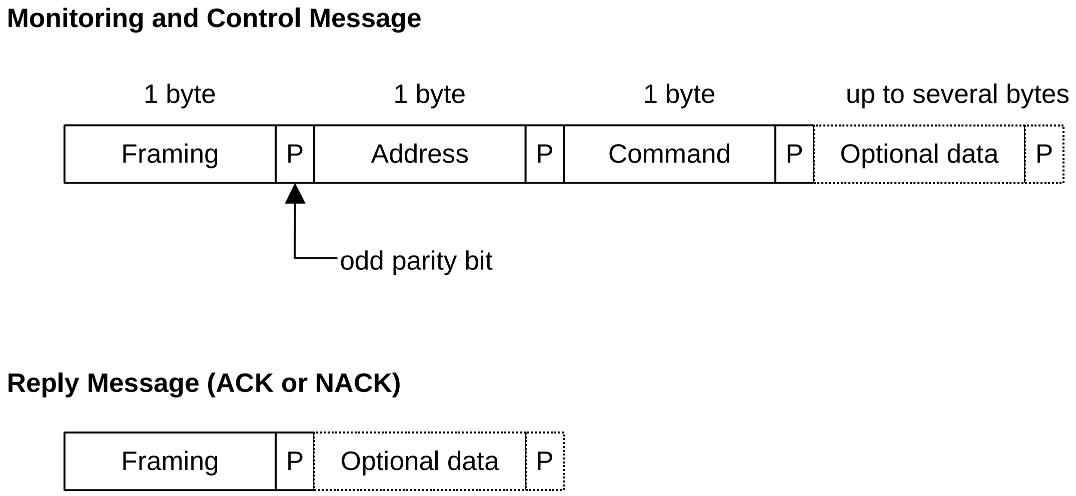


Figure B-1: Message Format

"Framing", "Address" and "Command" fields are detailed in following clauses.

B.5.1 Framing field description

The Framing Field is described in Table B.1.

Table B.1: Framing Definitions

Hex Byte	Binary	Framing byte Function
0xE2	1110 0010	Command from Master, Reply required, First transmission
0xE4	1110 0100	Reply from Slave, "OK", no errors detected.
0xE5	1110 0101	Reply from Slave, Command not supported by slave.
0xE6	1110 0110	Reply from Slave, Parity Error detected - Request repeat.
0xE7	1110 0111	Reply from Slave, message format not recognized - Request repeat.
0xE8	1110 1000	Extended Command from Master, Reply required only after last message block, First transmission
0xE9	1110 1001	Extended Command from Master, Reply required only after last message block, Repeated transmission
0xEA	1110 1010	Extended Command from Master, Reply required after each message block, First transmission
0xEB	1110 1011	Extended Command from Master, Reply required after each message block, Repeated transmission
0xEC 00	1110 1100 0000 0000	Reply from Slave, command understood, task not yet completed, unknown time to execute
0xEC nn	1110 1100 nn	Reply from Slave, command understood, check if task completed after nn seconds (1 to 127 binary)
0xED nn		Please repeat block nn (where nn is between 01 and 2C)
0x ED E1		Reply from Slave, EUI-64 of IDU not valid
0xED Fp	1110 1101 1111 pppp	Reply from Slave, relating to password commands, where p:
0xED F0	1110 1101 1111 0000	Password in the incoming string not valid (1st to nth attempts – unidentified number of non-critical attempts)
0xED Fn	1110 1101 1111 nnnn	Password in the incoming string not valid (identified number of non-critical attempts)
0xED FE	1110 1101 1111 1110	Password in the incoming string not valid, pen-ultimate attempt (e.g. 5th)
0xED FF	1110 1101 1111 1111	ODU Locked (installer required – e.g. 6th to infinite use of wrong password)
0xEE 00	1110 1110	Reply from Slave, CRC not valid (no additional information)
0xEF	1110 1111	Reply from Slave, additional blocks to follow

NOTE: The framing commands are grouped in pairs, where the value of the 2nd LSB of the first bytes gives an indication whether a further response is expected ("1") or not ("0"), although this is not a "hard" rule this should assist with low level detection software.

A positive acknowledgement (0xE4) shall be used to reply that the message from the Master has been successfully received. Any data requested by the IDU (defined by the original command from the IDU) will be sent directly after the positive acknowledgement byte. If the reply is more than 8 bytes in total then it must use the extended message structure (see Clause B.5.5.3).

Negative acknowledgements shall use values 0xE5, 0xE6 0xE7 (no additional data is permitted) as defined in Table B.1

0xE5 shall additionally be used when a command is not supported or cannot be implemented due to a functional problem in the ODU. In this case the ODU shall also flag an alarm to the IDU using the mechanism described in B2.

0xE6 Parity error detected (this will, in practice, occur as the result of transmission error), parity check is performed on each byte of each command. Notice that in case of CRC error, the reply is 0xEE.

0xE7 shall be used to flag an incompatibility between a command and any other field rendering execution of that command impossible for example incorrect message structure, wrong number of bits or bytes.

In cases where the password is used in the command, the ODU may reply 0xEB, 0xEC, 0xED or 0xEE.

B.5.2 Address field description

This field (encoded in one byte) specifies the destination subsystems for each message according to definitions in Table B.2 below.

Table B.2: Address Definitions

Hex Byte	Binary	Address byte Function
0x80	1000 0000	Any RCST (and VSAT)
0x81	1000 0001	IDU of RCST
0x82	1000 0010	ODU of RCST
0x83	1000 0011	Other (Future extension)

B.5.3 Command field description (IDU → ODU)

This field (encoded in one byte) specifies the required action for the addressed subsystem according to definitions in Table B.3.

Table B.3: Command Definitions

Hex. Byte	Use pass-word	Command	Status	Action
00	No	Reset	M	Reset all the ODU functions (same as power down reset)
0A	No	Soft Reset	M	ODU Software Reset
12	No	Monitoring	M	Request the general status of the ODU
5C	No	Manufacturer's ID	M	Request Manufacturer's Identification
5D	No	Product ID	M	Request the PProduct’s Identification
C1	No	Download start	O	Allow the ODU to enter into download mode
C2	No	Download data	O	Download software data
C3	No	Download abort	O	Abort the download process
C4	No	Download valid	O	Grab the "new" software into a non volatile memory
C5	No	Download toggle	O	Toggle between current software version and previous one
C6	Yes	SSPA ON	M	Enable the ODU amplification output
C7	No	SSPA OFF	M	Disable the ODU amplification output
C8	Yes	Set power level	O	Set SSPA power level
C9	No	Mod ON	O	Normal operation
CA	Yes	Mod OFF	O	Modulation Off, transmit Continuous Wave (CW)
CB	Yes	Change passwd	M	Enable the ODU password modification
CC	Yes	Validate passwd	M	Activate the new password
CD	Yes	Reset ODU locked	M	Reset the ‘Faulty password counter’ and back to default password
CE	No	Transmitter Disable	M	ODU to power down transmitter
CF	Yes	Transmitter Enable	M	ODU to power up transmitter (SSPA remains off)
D0	No	Get calibration table	O	Get the ODU calibration data for temperature and frequency variations
D1	No	Get Temperature	O	Report temperature of the ODU
D2	No	Get power output value	O	Get the ODU measured power output
D3	No	Get Location	O	Get the ODU geographical location (latitude, longitude, altitude)
D4	Yes	Set Location	O	Set the ODU location (when stored in ODU)
D5	No	Serial Number	M	Request the ODU serial number
D6	No	Firmware version	M	Request the ODU firmware version
D7 to DF	-	-	-	Reserved for future DVB-RCST commands

NOTE: M = Mandatory, O = Optional (in case function is not supported)

B.5.4 Password Description

Passwords are required for some commands in order to avoid inadvertent transmission or unapproved use of the ODU. The password will consist of 4 bytes and must be used with each of the designated commands shown in Table B.3. In these cases the password will immediately follow the relevant command byte, if there is any associated data used by these commands, it will be sent in a following block (see extended messages clause below).

The ODU shall refuse commands if the password does not correspond to its last valid password. After 5 consecutive erroneous passwords, the ODU shall warn that only 1 try remains. After the 6th faulty password, ODU shall refuse all commands except the status request, transmitter disable command and the download commands. These last commands allow an expert to reset the 'Faulty password counter' and reset the default password. The default password is system dependent.

When the ODU becomes locked due to 6 consecutive incorrect passwords, the SSPA shall be disabled and the transmitter powered down.

In the following clauses, the password is noted 'PWD' in the commands description.

B.5.5 Extended message format

In order to maintain backward compatibility with existing DiSEqC processors (typically 8 bit microprocessors) it is not possible to have more than 8 bytes of continuous code without the risk of potentially crashing existing devices. Therefore, to allow for the transmission of much longer messages, these will be subdivided into blocks of 8 bytes. Between each block there must be a short pause (T_b) of between 5 ms and 10 ms to allow existing microprocessors, and systems with small hardware buffers, to process each block without a data overflow.

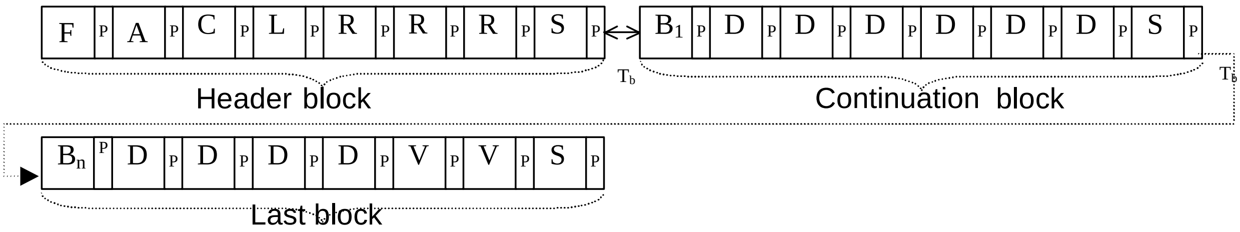
B.5.5.1 Extended messages for commands (IDU → ODU)

The structure of the first block will always be a standard DiSEqC message which has a framing, address and command byte, and does NOT contain any of the subsequent data which is to be error protected (e.g. CRC verified). This block will identify that the subsequent blocks are mostly data and will have a different structure, namely the first byte will be a block identifier which increments in each block, and the last byte will again be reserved for error protection. The framing byte (0x/E2/E8/EA) of the first block defines whether a reply is required to THIS initial block (before the data is transmitted), only after the last block or to all blocks. Also within the first block it will be possible to define how many blocks there are in total. An advantage of the optional reply here is that the slave can be given some time to 'prepare' itself for the main data processing task (e.g. clearing a block of memory), and could delay the reply for (say) up to 100 ms, if it needed to (assuming the master has asked for a reply). If not all the subsequent blocks are to be replied to, then the LAST block could then have a reply of the form 'E4' (OK), or 'ED nn [nn]' (Please repeat block number[s] nn).

All subsequent (continuation) blocks would be of the form: 'Ax dd dd dd dd dd dd [pp]' where A is A, B or C indicating the high nibble of the block count, x is the low nibble of the block count, d are data nibbles and pp is a simple (optional) checksum of the 6 bytes in the block. 'A0' will be reserved as a 'wildcard' block number for applications where it is unnecessary to update the block identifier byte for each block.

The last block contains data (or 'stuffed' bytes if appropriate) AND the 16-bit CRC. The reason for this is mainly that the CRC is processed in exactly the same way as the data bits, and then if the result is 0000 the data is valid. In this way, with 6 bytes per data block, this fits 256 data bytes (+ 2 CRC bytes) neatly into exactly 43 blocks (plus the initial 'header' block) which would be carried in the range of 0xA1 to 0xCB.

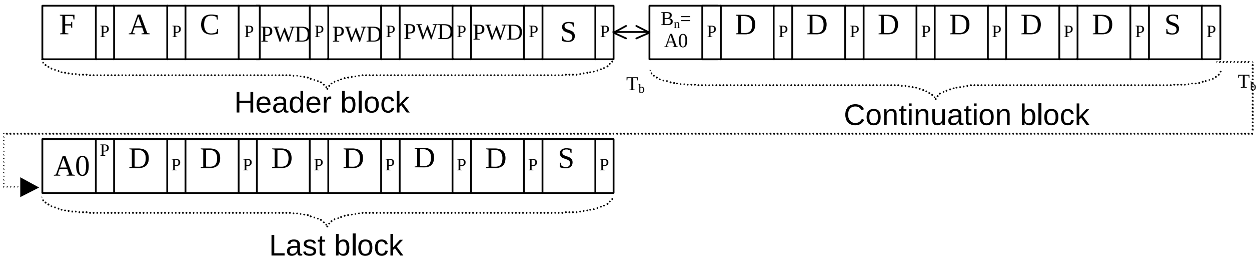
The extended message structure is shown below:


F P A P C P L P R P R P R P S P
B1 P D P D P D P D P D P D P S P
Tb
Tb
Header block
Continuation block
Bn P D P D P D P D P V P V P S P
Last block

F = Framing byte, P = Parity bit, A = Address byte, C = Command byte, L = Length of message, R = Reserved byte (for reply strategy etc.), B_n = Block identifier, D = Data byte, S = checkSum (optional),V = Verification (CRC as described in the clause below), 5 ms < T_b < 10 ms

B.5.5.2 Simplified structure for short fixed length extended messages (IDU → ODU)

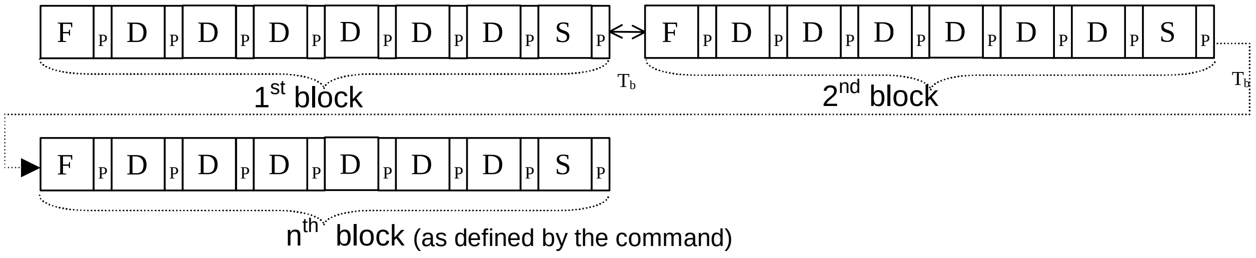
To simplify structure for short fixed messages of two or three blocks, for example password protected commands, it is possible to drop the data verification (CRC) since the likelihood of errors is much lower. As the message length (number of 8 byte blocks) is fixed and is defined by the command itself, byte "L is not required,. in this case the subsequent block identifier(s) is set to "A0". To give an example for the case of a password protected command the structure could be as follows:


F P A P C P PWD P PWD P PWD P PWD P S P
Bn= A0 P D P D P D P D P D P D P S P
Tb
Tb
Header block
Continuation block
A0 P D P D P D P D P D P D P S P
Last block

F = Framing byte, P = Parity bit, A = Address byte, C = Command byte, PWD = Password byte, B_n = Block identifier set to A0, D = Data byte, S = checkSum (optional), 5 ms < T_b< 10 ms

B.5.5.3 Extended messages for replies (ODU → IDU)

For certain commands, the replies have additional data attached. If the total number of data bytes expected in the reply (as defined by the originating command) is more than 6 bytes then it is necessary to use the extended message structure shown below. The framing byte will usually be "E4" and the last byte is reserved for a checksum (whether it is used or not). This gives a "payload" of 6 bytes per block.


F P D P D P D P D P D P D P S P
F P D P D P D P D P D P D P S P
Tb
Tb
1st block
2nd block
F P D P D P D P D P D P D P S P
nth block (as defined by the command)

F = Framing byte, P = Parity bit, D = Data byte, S = checkSum (optional), 5 ms < T_b = 10 ms

B.5.6 CRC Definition

Some commands require a CRC (see Figure B.2) at the end of the payload in order to secure the communication. The framing, destination address, command and other bytes of the first block are not included within the calculation. Only the data bytes in the subsequent blocks are processed to calculate the CRC). The CRC used is:

$$\text{CRC} = x^{16} + x^{12} + x^{9} + x^{5} + x + 1$$

Figure B-2: CRC calculation

B.5.7 General implementation of functions

In this clause the breakdown of each function into message exchanges between IDU and ODU is shown. These commands are not used during transmissions to avoid generating any spurious noise in the ODU.

B.5.7.1 Reset Status and Parameter Request

B.5.7.1.1 ODU reset (0x0A)

Reset of all software ODU functions (reload PLL divider, reset register status, alarms…). Note that the 'Faulty password counter' will not be reset.

This command (see Table B.4) shall not be sent if the SSPA is On.

Table B.4: ODU reset

Direction	Message	Comment
IDU → ODU	E2 82 0A	IDU sends reset command to the ODU
ODU → IDU	E5	Command rejected, not supported by ODU (should never occur)
ODU → IDU	E6	Command rejected, parity error during transmission
ODU → IDU	E7	Command rejected, message format not recognized
ODU → IDU	ED FF	Command rejected, ODU locked (due to use of faulty password in at a previous stage)
ODU → IDU	E4	Command accepted. ODU will perform a complete reset (software reset). Faulty_passwd_counter will not be reset

Note that "reset command" will be rejected if the ODU is locked. In fact the only way to download new software is to perform an ODU hard reset (cycling power). The IDU shall wait at least 10 s after an "ODU reset" to send any command (ODU loader boot time).

B.5.7.1.2 ODU Status (0x12)

This command requests the ODU status (see Table B.5). The ODU returns the general status information to the IDU. The ODU shall reply to this command even if it is in locked state. Means that the 0xED answer is not possible to this command. Alarms are buffered until the IDU reads the status register or until the IDU performs an ODU reset (0x0A).

Table B.5: ODU status

Direction	Message	Comment
IDU → ODU	E2 82 12	IDU sends status command to the ODU
ODU → IDU	E5	Command rejected, not supported by ODU (should never occur)
ODU → IDU	E6	Command rejected, parity error during transmission
ODU → IDU	E7	Command rejected, message format not recognized
ODU → IDU	E4 aa bb cc	Command accepted. ODU will give its status with 3 bytes (aa bb cc).

When the Status request command is launched while the ODU is in download mode, only "Software Download alarm" and "ODU main" fields are relevant.

B.5.7.1.2.1 aa byte status description: Alarms

Table B.6: Alarms (ODU status)

bit	Status name	Values
7	Self test alarm	0 : No Self test alarm 1 : Self test alarm
6	PLL status	0 : Lock PLL 1: Unlock PLL
5	Power supply status	0 : No Power supply Alarm 1 : Power supply Alarm
4	Faulty password counter	[0.. 6] faulty password(s) (bit 4 is msb)
3		
2		
1	Software Download alarm	0 : No CRC alarm on downloaded file 1 : CRC (see note) alarm on downloaded file
0		0 : No other download error 1 : Other download error

NOTE: This CRC corresponds to the CRC of the whole downloaded program (it does not refer to the CRC performed on each data packet –refer to 0xC2 command).

B.5.7.1.2.2 bb byte status description: ODU state

Table B.7: State (ODU status)

bit	Status name	Values
7	Reserved	0
6	Reserved	0
5	Reserved	0
4	Reserved	0
3	SSPA Status	0 : Off 1 : On (see note)
2	ODU main	0 : Not in Running state 1 : Running state
1		Reserved : 0
0		0 : Not in Software download state 1 : Software download state

NOTE: This CRC corresponds to the CRC of the whole downloaded program (it does not refer to the CRC performed on each data packet –refer to 0xC2 command).

B.5.7.1.2.3 cc byte status description: Reserved for future use

Reserved, all bits set to zero.

B.5.7.1.3 ODU Identification (0x54, 0x55, 0x56, 0xD5)

These commands (see Table B.8) allows the factory or an authorized installer to collect the different ODU product information: Manufacturer's information (using EUI64 standard from IEEE), ODU software & hardware version/release, ODU type and ODU serial number, etc.

After power up or reset, the IDU needs to issue this command to the ODU to move to on-line mode. The IDU shall wait at least 10 s after an ODU power cycling to send this command (ODU boot time).

Table B.8: ODU manufacturer's identification (0x5C)

Direction	Message	Comment
IDU → ODU	E2 82 5C	IDU sends Manufacturer's identification command to the ODU
ODU → IDU	E5	Request rejected, not supported by ODU (should never occur)
ODU → IDU	E6	Request rejected, parity error during transmission
ODU → IDU	E7	Request rejected, message format not recognized
ODU → IDU	ED FF	Command rejected, ODU locked (due to use of faulty password in at a previous stage, ≥ 6 faulty passwords used)
ODU → IDU	E4 gg gg gg	Request accepted. ODU shall return the Manufacturer's ID, first three bytes of EUI64

Table B.9: ODU product identification (0x5D)

Direction	Message	Comment
IDU → ODU	E2 82 5D	IDU sends Product identification command to the ODU
ODU → IDU	E5	Request rejected, not supported by ODU (should never occur)
ODU → IDU	E6	Request rejected, parity error during transmission
ODU → IDU	E7	Request rejected, message format not recognized
ODU → IDU	ED FF	Command rejected, ODU locked (due to use of faulty password in at a previous stage, ≥6 faulty passwords used)
ODU → IDU	E4 hh hh hh hh hh	Request accepted. ODU shall return the Product ID, remaining 5 bytes of EUI64

Table B.10: ODU firmware version (0xD6)

Direction	Message	Comment
IDU → ODU	E2 82 D6	IDU sends firmware version command to the ODU
ODU → IDU	E5	Request rejected, not supported by ODU (should never occur)
ODU → IDU	E6	Request rejected, parity error during transmission
ODU → IDU	E7	Request rejected, message format not recognized
ODU → IDU	ED FF	Command rejected, ODU locked (due to use of faulty password in at a previous stage, ≥ 6 faulty passwords used)
ODU → IDU	E4 aa bb cc dd ff	Request accepted. ODU shall return the ODU serial number

Table B.11: ODU serial number (0xD5)

Direction	Message	Comment
IDU → ODU	E2 82 D5	IDU sends serial number command to the ODU
ODU → IDU	E5	Request rejected, not supported by ODU (should never occur)
ODU → IDU	E6	Request rejected, parity error during transmission
ODU → IDU	E7	Request rejected, message format not recognized
ODU → IDU	ED FF	Command rejected, ODU locked (due to use of faulty password in at a previous stage, ≥ 6 faulty passwords used)
ODU → IDU	EF ee ee ee ee ee ee CS EF ee ee ee ee ee ee CS E4 ee ee ee ee ee ee CS	Request accepted. ODU shall return the ODU serial number

NOTE: CS = Check Sum

All the following values are considered hexadecimally coded.

Table B.12: ODU identification codes

Bytes	Bits	Status name	Values
aa	7..4	Current Software Major version	0..F
	3..0	Current Software Minor version	0..F
bb	7..4	Backup Software Major version	0..F
	3..0	Backup Software Minor version	0..F
cc	7..4	Hardware Major version	0..F
	3..0	Hardware Minor version	0..F
dd	7..3	Reserved	0
	0..2	ODU type	Gives the ODU type (1 to 4, depending on transmit symbol rate).
ee ee ee ee ee ee ee ee ee ee ee ee ee ee ee ee	127..0	ODU Serial Number	0..FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
ff	7..0	ODU Boot Firmware version	0..F
gg gg gg	63 .. 40	Company ID of the Manufacturer allocated by IEEE	Identifies Manufacturer
hh hh hh hh hh	39 .. 0	Unique Product ID allocated by Manufacturer according to IEEE guidelines	0 .. FF FF FF FF FF

B.5.7.2 Operational Commands

B.5.7.2.1 SSPA ON (0xC6)

This command forces the ODU to enable its amplification output.

Table B.13: SSPA on

Direction	Message	Comment
IDU → ODU	E2 82 C6 PWD	IDU sends the SSPA output enabling command to the ODU
ODU → IDU	E5	Request rejected, not supported by ODU (should never occur)
ODU → IDU	E6	Request rejected, parity error during transmission
ODU → IDU	E7	Request rejected, message format not recognized
ODU → IDU	ED Fn	Request rejected, password used not valid (< 5 faulty passwords used)
ODU → IDU	ED FE	Request rejected, 5 consecutive faulty passwords used
ODU → IDU	ED FF	Command rejected, ODU locked (due to use of faulty password in at a previous stage, ≥ 6 faulty passwords used)
ODU → IDU	E4	Command accepted. ODU shall turn on the SSPA

Restrictions: this command shall be send by the IDU if and only if no alarm (PLL unlock, Power supply, Self test fail) is present. Furthermore, the current password has to be different than the default one (factory one) in the ODU otherwise the command shall be refused. If the command and password are correct, the ODU returns the acknowledgement immediately. Nevertheless, the amplification stage may not be directly ready to transmit. The IDU has to implement a timer to know when the ODU will be ready to transmit (given time after receiving SSPA ON command acknowledgement.

B.5.7.2.2 SSPA OFF (0xC7)

This command forces the ODU to disable its amplification output.

Table B.14: SSPA off

Direction	Message	Comment
IDU → ODU	E2 82 C7	IDU sends the SSPA output disabling command to the ODU
ODU → IDU	E5	Request rejected, not supported by ODU (should never occur)
ODU → IDU	E6	Request rejected, parity error during transmission
ODU → IDU	E7	Request rejected, message format not recognized
ODU → IDU	ED FF	Command rejected, ODU locked (due to use of faulty password in at a previous stage)
ODU → IDU	E4	Command accepted. ODU shall turn off the SSPA

. By default, after a power on or a reset, the SSPA shall be turned off by the ODU.

B.5.7.2.3 Transmitter Disable (0xCE)

This command forces the ODU to power down the transmitter circuitry. This command can be issued to the ODU when it is locked due to previously incorrect passwords or when signalled to do so via the hub.

Table B.15: Transmitter Disable

Direction	Message	Comment
IDU → ODU	E2 82 CE	IDU sends the transmitter disable command to the ODU
ODU → IDU	E5	Request rejected, not supported by ODU (should never occur)
ODU → IDU	E6	Request rejected, parity error during transmission
ODU → IDU	E7	Request rejected, message format not recognized
ODU → IDU	E4	Command accepted. ODU shall disable the transmitter

This command is issued by the IDU whenever the RCST is put in Hold State. The transmitter can only be re-enabled by the IDU when the Hold State is removed. In case of error (including internal fault conditions such as PLL unlock and/or DC powering problem) or alarm, the ODU shall automatically disable the SSPA; the ODU shall be unconditionally stable.

B.5.7.2.4 Transmitter Enable (0xCF)

This command allows the IDU to re-enable the transmitter when the RCST Hold State is removed. At the completion of this command, the transmitter is again powered on, but the SSPA is still in the off state.

Table B.16: Transmitter Enable

Direction	Message	Comment
IDU → ODU	E2 82 CF PWD	IDU sends the transmitter enable command to the ODU
ODU → IDU	E5	Request rejected, not supported by ODU (should never occur)
ODU → IDU	E6	Request rejected, parity error during transmission
ODU → IDU	E7	Request rejected, message format not recognized
ODU → IDU	EA	Command rejected, ODU locked (due to use of faulty password in at a previous stage, ≥ 6 faulty passwords used)
ODU → IDU	E4	Command accepted. ODU shall power on transmitter

B.5.7.2.5 Set Power Level (0xC8)

This command allows the IDU to adjust the output power level of the ODU in at least 1 dB or less steps. The command instructs the ODU by indicating how many "steps" up or down encoded by one signed data byte PWR_ADJ (±128 steps), this byte will follow in a separate block.

Table B.17: Set Power Level

Direction	Message	Comment
IDU → ODU	E2 82 C8 PWD PWR_ADJ	IDU sends the Set Power Level command to the ODU followed by the value in the PWR_ADJ byte
ODU → IDU	E5	Request rejected, not supported by ODU (optional function)
ODU → IDU	E6	Request rejected, parity error during transmission
ODU → IDU	E7	Request rejected, message format not recognized
ODU → IDU	EA	Command rejected, ODU locked (due to use of faulty password in at a previous stage, ≥ 6 faulty passwords used)
ODU → IDU	E4	Command accepted. ODU shall change power level

B.5.7.2.6 Mod ON (0xC9)

In the case when the modulation is applied within the ODU and a co-axial IFL is still used, then this command allows the ODU to re-enable the modulation (for future implementations).

Table B.18: Modulation On

Direction	Message	Comment
IDU → ODU	E2 82 C9	IDU sends the Mod On command to the ODU
ODU → IDU	E5	Request rejected, not supported by ODU (optional function)
ODU → IDU	E6	Request rejected, parity error during transmission
ODU → IDU	E7	Request rejected, message format not recognized
ODU → IDU	EA	Command rejected, ODU locked (due to use of faulty password in at a previous stage, ≥ 6 faulty passwords used)
ODU → IDU	E4	Command accepted. ODU shall modulation on

B.5.7.2.7 Mod OFF (0xCA)

In the case when the modulation is applied within the ODU and a co-axial IFL is still used, then this command allows the ODU to disable the modulation (for future implementations).

Table B.19: Modulation Off

Direction	Message	Comment
IDU → ODU	E2 82 CA PWD	IDU sends the Mod OFF command to the ODU
ODU → IDU	E5	Request rejected, not supported by ODU (optional function)
ODU → IDU	E6	Request rejected, parity error during transmission
ODU → IDU	E7	Request rejected, message format not recognized
ODU → IDU	EA	Command rejected, ODU locked (due to use of faulty password in at a previous stage, ≥6 faulty passwords used)
ODU → IDU	E4	Command accepted. ODU shall switch modulation off

When the modulation is switched off the ODU will transmit a "continuous wave" i.e. a clean carrier.

B.5.7.3 Download Commands

B.5.7.3.1 Download start (0xC1)

This command allows the ODU to enter into download mode. This command can only be issued after and ODU power cycle and prior to identification status request. The IDU shall wait at least 10 s after an ODU power cycling to send this command (ODU loader boot time). The DL_FL_SIZE corresponds to the Download File Size expressed in bytes on 24 bits.

Table B.20: Download start

Direction	Message	Comment
IDU → ODU	E2 82 C1 DL_FL_SIZE	IDU sends the Download start command to the ODU. It includes the number of bytes of the complete software to download and the CRC on the file size
ODU → IDU	E5	Request rejected, not supported by ODU (this answer may occur if the download start command is sent out of the allowed time after the ODU power ON event)
ODU → IDU	E6	Request rejected, parity error during transmission
ODU → IDU	E7	Request rejected, message format not recognized
ODU → IDU	E4	Command accepted. ODU shall enter into download mode immediately and store the download file size
	EC nn	Command accepted. ODU shall enter into download mode, please check status after nn seconds (1 to 127 binary) and store the download file size

Note that the DL_FL_SIZE may handle a value of 0 (zero).

If the password given is the default one, the download start command shall be refused except if the ODU is locked. In this case, the ODU shall enter the download mode so that a new software version can be loaded to clear the faulty password counter. This command can take up to 6,5 s to execute. IDU timeouts must account for this delay.

B.5.7.3.2 Download data (0xC2)

This command allows the IDU to transfer ODU program bytes to the ODU divided in 256 bytes per command (message) in 43 blocks of 8 bytes. If the program code is longer than 256 bytes than multiple messages each starting with 0xC2 will be used.

Table B.21: Download data

Direction	Message	Comment
IDU → ODU	E2 82 C2 L 248 data bytes	IDU sends the length of message in terms of 6 byte block of data, up to 258 bytes including 2 byte CRC – i.e. max. number of data blocks is 43. Alternative framing byte (E8 or EA) in this first block will indicate the exact reply strategy implemented
ODU → IDU	E5	Request rejected, not supported by ODU (should never occur)
ODU → IDU	E6	Request rejected, parity error during transmission
ODU → IDU	E7	Request rejected, message format not recognized
ODU → IDU	E4	Command accepted (first block OK) continue with data download
IDU → ODU	Block identifier + 6 data bytes + CheckSum	L x blocks of 8 bytes (see Clause B.5.5)
ODU → IDU	E4	Command accepted. ODU shall store the checked data until the download validation

Any failed packet shall be ignored by the ODU.

The complete program shall be stored into not sensitive memory until the validation of the complete downloaded software.

The IDU timeout period shall be increased to at least 500 ms to allow for complete ODU processing of the command prior to sending the next message.

B.5.7.3.3 Download abort (0xC3)

This command allows the IDU to abort the downloading process when communication problems have occurred or on major trouble.

Table B.22: Download abort

Direction	Message	Comment
IDU → ODU	E2 82 C3	IDU sends the Download abort command to the ODU
ODU → IDU	E5	Request rejected, not supported by ODU (should only occur if software downloading is not supported)
ODU → IDU	E6	Request rejected, parity error during transmission
ODU → IDU	E7	Request rejected, message format not recognized
ODU → IDU	E4	Command accepted. ODU shall remove the previous downloaded data bytes and exit the download mode in order to restore the normal running mode

Once the abort command has been acknowledged, the IDU has to perform an ODU reset (reset or power cycling). The current software shall still be active. The IDU timeout for this message must be increased to 3,5 s.

B.5.7.3.4 Download validate (0xC4)

This command allows the ODU to check the received software and store it if the received data is correct. This command may be shown as indicating the end of the download procedure.

Table B.23: Download validate

Direction	Message	Comment
IDU → ODU	E2 82 C4	IDU sends the Download validate command to the ODU.
ODU → IDU	E5	Request rejected, not supported by ODU (should never occur)
ODU → IDU	E6	Request rejected, if the new software is not valid (wrong CRC file) or the ODU is not able to store the new downloaded software or parity error during transmission of the latter command
ODU → IDU	E7	Request rejected, message format not recognized
ODU → IDU	E4	Command accepted. ODU shall check the complete program validity and store it in order to activate this new software as the current software before acknowledging the command - if completed within 115 ms, otherwise:
ODU → IDU	EC nn	Command accepted. ODU shall check the complete program validity and store it in order to activate this new software as the current software before acknowledging the command, please check status after nn seconds (1 to 127 binary) if validation is complete (e.g. IDU resends 0XC4 command until E4 is received)

Before responding positively the command, the ODU shall:

- Check the new software validity (CRC).
- Save the current software into the backup section.
- Save the new received software into the current software section.
- Restore the running bit into the main ODU status field.

The new program shall be active only after a reset command or a power off & on. The timeout for the ODU response must be increased to as much as 8 s to accommodate required processing. The SW version will be updated in the ODU status register once the reset has been launch by the IDU. The IDU has to check the ODU status and ODU identification registers to be aware of the software download result.

B.5.7.3.5 Download toggle (0xC5)

This command allows the IDU to toggle to the previous software. This command shall be send only if the ODU is in download mode. To do so, the IDU shall use the "start download" command with a DL_FL_SIZE set to 0. The current software shall be transfer to the backup non-volatile memory and the 'old' program becomes the current one.

Table B.24: Download toggle

Direction	Message	Comment
IDU → ODU	E2 82 C5	IDU sends the Download revert command to the ODU.
ODU → IDU	E5	Request rejected, not supported by ODU (should never occur)
ODU → IDU	E6	Request rejected, parity error during transmission
ODU → IDU	E7	Request rejected, message format not recognized
ODU → IDU	E4	Command accepted. ODU shall toggle the current software with the previous one (in the backup section) if completed within 115 ms, otherwise:
ODU → IDU	EC nn	Command accepted. ODU shall toggle the current software with the previous one (in the backup section), please check status after nn seconds (1 to 127 binary) if reversion is complete (e.g. IDU resends 0XC5 command until E4 is received)

Before responding positively the command, the ODU shall switch current and 'old' software program. It has to be noticed that if this command is sent twice, the ODU status will not be affected. The 'old' program shall be active only after a reset command or a power off & on. The SW version will be updated in the ODU status register once the reset has been launch by the IDU (this status reflects the version of the effective running software). The timeout for the ODU response can be as large as 12 s to support the processing of this command.

B.5.7.4 Password Commands

The procedure to change a password is divided into 2 parts: the password change command (using the current password (PWD_cur)' and the new one (PWD_new)) and the password validate command. Immediately after the acknowledgement of the password validate command, the new password becomes the current valid one. If any other command or request is inserted between the 2 password commands, the password error has to be raised, increasing the 'Faulty password counter'. Furthermore, the password modification procedure will have to be re-initialized.

B.5.7.4.1 Change password (0xCB)

This command enables the ODU password modification. This function only changes the password but does NOT change the 'current' password validity. This command required the message to be split into two blocks as shown below.

Table B.25: Change password

Direction	Message	Comment
IDU → ODU	E2 82 CB PWD PWD_new CRC	IDU sends the change password command to the ODU with the current password value in the first block. New password calculated with CRC is sent in the second block
ODU → IDU	E5	Request rejected, not supported by ODU (should never occur)
ODU → IDU	E6	Request rejected, parity error during transmission
ODU → IDU	E7	Request rejected, message format not recognized
ODU → IDU	ED Fn	Request rejected, password (current) not valid (< 5 faulty passwords used)
ODU → IDU	ED FE	Request rejected, 5 consecutive faulty passwords used
ODU → IDU	ED FF	Command rejected, ODU locked (due to use of faulty password in at a previous stage, ≥ 6 faulty passwords used)
ODU → IDU	E4	Command accepted. ODU shall store the new password and wait to the next password command: validate password

The old password is still valid at this point. As already noticed, the passwords are coded on 4 bytes.

B.5.7.4.2 Validate password (0xCC)

This changes the current password to use the new password.

Table B.26: Validate password

Direction	Message	Comment
IDU → ODU	E2 82 CC PWD_new CRC	IDU sends the validate password command to the ODU in the first block The new password calculated with the CRC is sent in the second block
ODU → IDU	E5	Request rejected, not supported by ODU (should never occur)
ODU → IDU	E6	Request rejected, parity error during transmission
ODU → IDU	E7	Request rejected, message format not recognized. This reply is the one sent by the ODU if the "validate password command" is not sent immediately after the "change password command" (see note). This should never occur. The IDU is in charge of sending the right command sequence
ODU → IDU	ED Fn	Request rejected, password (old) not valid (< 5 faulty passwords used)
ODU → IDU	ED FE	Request rejected, 5 consecutive faulty passwords used
ODU → IDU	ED FF	Command rejected, ODU locked (due to use of faulty password in at a previous stage, ≥6 faulty passwords used)
ODU → IDU	E4	Command accepted. ODU has compared the 2 new passwords and shall activate the new password

NOTE: This reply is also used if the "change password" command is sent twice consecutively, meant that 0xCA command will be followed by 0xCA command. If the "validate password" command is not sent immediately after the "change password" command, an error is generated by the ODU, and the 'faulty password counter' will be incremented. The process to modify the password exits.

The new password is valid if and only if the 2 commands "change password" and "validate password" are correctly sent with the current password and the new password. If the current password in the "change password command" or the new password in the "validate password command" is not correct, the faulty password counter shall be incremented. This command shall be sent immediately (consecutively) after the acknowledgement of the "change password command", otherwise the "change password command" shall be discarded by the ODU (the process to modify the password exits).

This command can take up to 3,5 s to execute. The IDU timeouts must account for this delay.

B.5.7.4.3 Reset ODU locked (0xCD)

This command allows authorized personnel to reset the 'Faulty password counter' and reset the default password.

Table B.27: Reset password

Direction	Message	Comment
IDU → ODU	E2 CD PWD_dft	IDU sends the Default Password to the ODU which resets the faulty password counter to zero and sets PWD_cur = PWD_disable
ODU → IDU	E5	Request rejected, not supported by ODU (should never occur)
ODU → IDU	E6	Request rejected, parity error during transmission
ODU → IDU	E7	Request rejected, message format not recognized

One implementation of this command could be as follows:

- ODU is delivered with both current_password and default password set to 0000. This password MUST be changed before the ODU will transmit.
- During installation the Hub/IDU forces installer/user to enter the first "user" password.
- Hub records this first_user_password and ODU changes default_password and the current_password to this value.
- All subsequent changes to the current_password by the user are not recorded by the hub nor change default_password in ODU.

- When ODU becomes locked:
out of band request (e.g. by telephone call to hub) for reset
Hub authorizes IDU to send reset command "CD" using default_password, faulty password counter is reset and ODU changes current_password to and default_password to 0000 (i.e. unable to transmit)
Hub either forces installer/user or itself to change password, new value recorded as default_password (both at hub and in ODU) and current_password.

- ODU is unlocked.

NOTE: For added security the hub can at any time change the default_password in the ODU by using the reset command.

B.5.7.5 Other functions

B.5.7.5.1 ODU calibration table (0xD0)

This request allows the IDU to retrieve the ODU calibration matrix following the frequency and temperature curve. The format of the calibration matrix will be system and ODU dependent to account for frequency differences (e.g., Ka vs. Ku-Band), temperature variations, etc. This command is optional depending upon the implementation of the ODU.

Table B.28: ODU calibration table

Direction	Message	Comment
IDU → ODU	E2 82 D0	IDU sends the calibration matrix request to the ODU
ODU → IDU	E5	Request rejected, not supported by ODU (optional function)
ODU → IDU	E6	Request rejected, parity error during transmission
ODU → IDU	E7	Request rejected, message format not recognized
ODU → IDU	ED FF	Command rejected, ODU locked (due to use of faulty password in at a previous stage)
ODU → IDU	E4 aa …	Request accepted. ODU shall return the output power calibration matrix

NOTE: No generic table has been defined to date, for a manufacturer specific table of less than 7 bytes the reply can use the simple message structure.

B.5.7.5.2 ODU measured temperature (0xD1)

This command allows the IDU to obtain the measured temperature of the IDU.

Table B.29: Measured Temperature

Direction	Message	Comment
IDU → ODU	E2 82 D1	IDU request measured temperature from the ODU
ODU → IDU	E5	Request rejected, not supported by ODU (optional function)
ODU → IDU	E6	Request rejected, parity error during transmission
ODU → IDU	E7	Request rejected, message format not recognized
ODU → IDU	EA	Request rejected, ODU locked
ODU → IDU	E4 aa	Command accepted. ODU provides internal temperature in degrees Celsius (2's complement encoded on 1 byte)

B.5.7.5.3 ODU output power Level (0xD2)

This request allows the IDU to retrieve the measured output power of the ODU.

Table B.30: ODU output power Level

Direction	Message	Comment
IDU → ODU	E2 82 D2	IDU sends the output power level request to the ODU
ODU → IDU	E5	Request rejected, not supported by ODU (optional function)
ODU → IDU	E6	Request rejected, parity error during transmission
ODU → IDU	E7	Request rejected, message format not recognized
ODU → IDU	ED FF	Command rejected, ODU locked (due to use of faulty password in at a previous stage)
ODU → IDU	E4 bb	Request accepted. ODU shall return the output power level (encoded on one byte)

B.5.7.5.4 ODU Location (0xD3)

This command allows the IDU to get the location information from the ODU.

Table B.31: Get Location data

Direction	Message	Comment
IDU → ODU	E2 82 D3	IDU request to get geographical location data
ODU → IDU	E5	Request rejected, not supported by ODU (optional function)
ODU → IDU	E6	Request rejected, parity error during transmission
ODU → IDU	E7	Request rejected, message format not recognized
ODU → IDU	EF xx xx xx xx, yy yy CS E4 yy yy, zz zz zz zz CS	Command accepted. ODU sends back its position co-ordinates as defined below

NOTE 1: CS = Check Sum

- x_co-ordinate: This 32 bit field defines the x co-ordinate of the RSCT location in metres;

- y_co-ordinate: This 32 bit field defines the y co-ordinate of the RSCT location in metres;

- z_co-ordinate: This 32 bit field defines the z co-ordinate of the RSCT location in metres.

NOTE 2: The position of the satellites will be expressed as Cartesian co-ordinates 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 metre level.

NOTE 3: These 32 bit fields are encoded in the same way as the satellite position data as spfmsbf = single precision floating point value, which is a 32 bit value formatted in accordance with ANSI/IEEE Standard 754 (1985). The most significant bit (i.e. the most significant bit of the exponent) is first.

B.5.7.5.5 Set ODU Location (0xD4)

This command allows the IDU to send the location information to the ODU in the case it is stored in the ODU.

Table B.32: Set Location data

Direction	Message	Comment
IDU → ODU	E2 82 D3 PWD A0 xx xx xx xx yy yy CS A0 yy yy zz zz zz zz CS	IDU command to set geographical location data
ODU → IDU	E5	Request rejected, not supported by ODU (should never occur)
ODU → IDU	E6	Request rejected, parity error during transmission
ODU → IDU	E7	Request rejected, message format not recognized
ODU → IDU	E4	Command accepted. ODU stores geographical location data

NOTE: CS = Check Sum

B.5.8 Command compatibility when SSPA ON

IDU/ODU communications should not be performed during actual return channel transmissions by the RCST to avoid the introduction of spurious signals on the transmitted carrier. In addition, some commands are not available when the SSPA is powered on. The compatibility of commands with the SSPA on is shown in the table below.

Table B.33: Command activity when transmitting

Hex. Byte	Command	SSPA ON
00	Reset	Not compatible
0A	Soft reset	Not compatible
12	Monitoring	Compatible
5C	Manufacturer's ID	Compatible
5D	Product ID	Compatible
C1	Download start	Not Compatible
C2	Download data	Not Compatible
C3	Download abort	Not Compatible
C4	Download valid	Not Compatible
C5	Download toggle	Not Compatible
C6	SSPA ON	--
C7	SSPA OFF	Compatible
C8	Set power level	Compatible
C9	Mod ON	Compatible
CA	Mod OFF, transmit Continuous Wave (CW)	Not Compatible
CB	Change password	Not Compatible
CC	Validate password	Not Compatible
CD	Reset ODU locked	Not Compatible
CE	Transmitter Disable	Compatible
CF	Transmitter Enable	Not compatible
D0	Get calibration data	Not Compatible
D1	Get Temperature	Compatible
D2	Get power output value	Compatible
D3	Get Location	Compatible
D4	Set Location	Not Compatible
D5	Serial Number	Compatible
D6	Firmware version	Compatible

B.5.9 Use of extended message structures

For commands with more than 4 bytes of additional data sent immediately after the command it is necessary to use either the fixed extended message structure (see Clause 5.5.2) or for very long messages (e.g. software downloading) the full extended structure (see Clause 5.5.1).

For replies with more than 7 bytes of data then it is necessary to use the extended message structure (see Clause 5.5.3).

The number of data bytes and which extended message structure to use is indicated in the table below:

Table B.34: Data bytes and use of message structures

Commands				Reply	
Hex. Byte	**Description**	**No of Data Bytes**	**Message structure**	**No of Reply Data Bytes**	**Message structure**
00	Reset	0	simple	0	simple
0A	Soft reset	0	simple	0	simple
12	Status	0	simple	3	simple
5C	Manufacturer's ID	0	simple	8	simple
5D	Product ID	0	simple	5	simple
C1	Download start	3	simple	1	simple
C2	Download data	up to 256	full extended	0	simple
C3	Download abort	0	simple	0	simple
C4	Download valid	0	simple	1	simple
C5	Download toggle	0	simple	1	simple
C6	SSPA ON	4	simple	0	simple
C7	SSPA OFF	0	simple	0	simple
C8	Set power level	5	fixed extended	0	simple
C9	Mod ON	0	simple	0	simple
CA	Mod OFF, transmit Continuous Wave (CW)	4	simple	0	simple
CB	Change password	10	fixed extended	0	simple
CC	Validate password	10	fixed extended	0	simple
CD	Reset ODU locked	4	simple	0	simple
CE	Transmitter Disable	0	simple	0	simple
CF	Transmitter Enable	4	simple	0	simple
D0	Get calibration data	0	simple	< 7	simple
D1	Get Temperature	0	simple	1	simple
D2	Get power output value	0	simple	1	simple
D3	Get Location	0	simple	12	fixed extended
D4	Set Location	16	fixed extended	0	simple
D5	Serial Number	0	simple	18	fixed extended
D6	Firmware version	0	Simple	5	simple

Annex C: Reference link budgets

The following link budgets are provided as examples and are not binding to a system implementation.

C.1 EIRP realization: implementation example

Table C.1 provides an example for the realization of an EIRP value of 45 dBW.

Table C.1: Example for the realization of an EIRP of 45 dBW

RCST TX characteristics		
Antenna diameter	m	0,80
Antenna efficiency		0,65
Tx frequency	GHz	29,70
Antenna peak gain	dBi	46,04
TX power/carrier	W	1,00
Output back-off	dB	0,54
Coupling losses	dB	0,50
EIRP	dBW	45,00

C.2 DVB-RCS Return Link-Budget

In the following tables, examples of link budgets for RCSTs, transmitting at different information rates are shown.

Important input parameters are in ***bold italic***.

Normal input parameters are in *italic*.

Formulas and normal results are in normal text.

Important results are in **bold**.

Table C.2 concerns the uplink section of the link budget. Table C.3 focuses on the downlink and overall link budget.

Table C.2: Up-link part

DVB-RCS Return Link Budget	Up-Link	128 kbit/s	384 kbit/s	1 024 kbit/s	2 048 kbit/s
RCST TX characteristics					
Information rate	***kbit/s***	***128,00***	***384,00***	***1 024,00***	***2 048,00***
Total coding rate		***0,5000***	***0,5000***	***0,5000***	***0,5000***
Total coding rate	dB	-3,01	-3,01	-3,01	-3,01
Channel rate	kbit/s	256,0	768,0	2 048,0	4 096,0
Roll-off factor		*0,35*	*0,35*	*0,35*	*0,35*
Occupied bandwidth	kHz	172,8	518,4	1 382,4	2 764,8
Tx frequency	*GHz*	*29,70*	*29,70*	*29,70*	*29,70*
C/I (see note 1)	*dB*	*19,50*	*19,50*	*19,50*	*19,50*
EIRP	***dBW***	***42,00***	***45,00***	***47,50***	***50,00***
Pointing losses	*dB*	*1,00*	*1,00*	*1,00*	*1,00*
EIRP effective	dBW	41,00	44,00	46,50	49,00
RCST -> Sat propagation					
Range	*km*	*38 039,81*	*38 039,81*	*38 039,81*	*38 039,81*
Path loss	dB	213,50	213,50	213,50	213,50
Atmospheric attenuation (w/o rain)	*dB*	*0,90*	*0,90*	*0,90*	*0,90*
Rain attenuation	***dB***	***0,00***	***0,00***	***0,00***	***0,00***
Additional attenuation	dB	0,00	0,00	0,00	0,00
Total attenuation	dB	214,40	214,40	214,40	214,40
Power flux density	dBW/m^2	-122,50	-119,50	-117,00	-114,50
Sat reception					
G/T towards SIT	***dB/K***	***13,00***	***13,00***	***13,00***	***13,00***
Transponder bandwidth	*MHz*	*400,00*	*400,00*	*400,00*	*400,00*
Boltzmann constant	dBW/K-Hz	-228,60	-228,60	-228,60	-228,60
Up-link results					
C/No up-link	**dBHz**	**68,20**	**71,20**	**73,70**	**76,20**
C/Io up-link ***(see note 2)***	**dBHz**	**69,98**	**74,75**	**79,01**	**82,02**
Ebch/No up-link	dB	14,12	12,35	10,59	10,08
Eb/No up-link	**dB**	**17,13**	**15,36**	**13,60**	**13,09**

NOTE 1: Value of C/I computed for a typical Ka-band SSPA at 1 dB compression point; C represents the total in-band power, while I represents twice the power of the first side-lobe.

NOTE 2: In the computation of C/Io up-link, Io is interfering power density generated by two adjacent RCSTs operating at their maximum powers (i.e. only subject to free-space loss).

Table C.3: Downlink and overall link budget

DVB-RCS Return Link Budget	Down-Link	128 kbit/s	384 kbit/s	1 024 kbit/s	2 048 kbit/s
Sat TX characteristics					
Tx Frequency	*GHz*	*18,50*	*18,50*	*18,50*	*18,50*
EIRP at saturation	***dBW***	***56,00***	***56,00***	***56,00***	***56,00***
Single carrier OBO *(see note 3)*	dB	38,24	33,01	29,21	26,20
Total OBO	dB	5,23	5,23	5,23	5,23
Single carrier EIRP *(see note 4)*	dBW	17,76	22,99	26,79	29,80
C/I (see note 5)	***dB***	***20,00***	***20,00***	***20,00***	***20,00***
C/Io down-link	**dBHz**	**72,38**	**77,15**	**81,41**	**84,42**
Sat -> HUB propagation					
Range	*km*	*38 460,53*	*38 460,53*	*38 460,53*	*38 460,53*
Path loss	dB	209,48	209,48	209,48	209,48
Atmospheric attenuation (w/o rain)	*dB*	*0,60*	*0,60*	*0,60*	*0,60*
Rain attenuation	***dB***	***0,00***	***0,00***	***0,00***	***0,00***
Additional attenuation	*dB*	*0,00*	*0,00*	*0,00*	*0,00*
Total attenuation	dB	210,08	210,08	210,08	210,08
Power flux density	dBW/m^2	-107,29	-107,29	-107,29	-107,29
HUB reception					
Antenna diameter	***m***	***6,00***	***6,00***	***6,00***	***6,00***
Antenna efficiency		*0,55*	*0,55*	*0,55*	*0,55*
Antenna gain	dB	58,71	58,71	58,71	58,71
Sky temperature	*K*	*26,00*	*26,00*	*26,00*	*26,00*
Pointing error	*deg*	*0,03*	*0,03*	*0,03*	*0,03*
3dB beamwidth	deg	0,19	0,19	0,19	0,19
Pointing loss	dB	0,30	0,30	0,30	0,30
Coupling losses	*dB*	*0,50*	*0,50*	*0,50*	*0,50*
LNA noise temperature	***K***	***150,00***	***150,00***	***150,00***	***150,00***
Equivalent system noise temperature	K	228,47	228,47	228,47	228,47
G/T	dB/K	35,12	35,12	35,12	35,12
Boltzmann constant	*dBW/K/Hz*	*-228,60*	*-228,60*	*-228,60*	*-228,60*
Down-link results					
C/No down-link	**dBHz**	**70,60**	**75,83**	**79,63**	**82,64**
Number of frequency slots		***2000***	***600***	***250***	***125***
Total capacity	**Mb/s**	**256,00**	**230,40**	**256,00**	**256,00**
Total channel rate	Mb/s	512,00	460,80	512,00	512,00
Channel spacing		*1,00*	*1,00*	*1,00*	*1,00*
Occupied bandwidth	MHz	345,60	311,04	345,60	345,60
Ebch/N0 down-link	dB	16,52	16,98	16,52	16,52
Eb/No down-link	**dB**	**19,53**	**19,99**	**19,53**	**19,53**
Full return link results					
C/No up-link	**dBHz**	**68,20**	**71,20**	**73,70**	**76,20**
C/Io up-link	**dBHz**	**69,98**	**74,75**	**79,01**	**82,02**
C/No down-link	**dBHz**	**70,60**	**75,83**	**79,63**	**82,64**
C/Io down-link	**dBHz**	**72,38**	**77,15**	**81,41**	**84,42**
C/No total	**dBHz**	**64,02**	**68,10**	**71,35**	**74,05**
Implementation Losses	*dB*	*2,00*	*2,00*	*2,00*	*2,00*
BER degradation (phase noise. tracking. etc.)	*dB*	*1,00*	*1,00*	*1,00*	*1,00*
Ebch/No total	dB	6,93	6,25	5,23	4,93
Eb/No total	**dB**	**9,94**	**9,26**	**8,25**	**7,94**
Req. Eb/No for FER=10^(-7) (see note 6)	***dB***	***3,20***	***3,20***	***3,20***	***3,20***
Additional margin		**6,74**	**6,06**	**5,05**	**4,74**

NOTE 3: Value of Output Back-Off for the useful carrier at the TWTA output; values are computed for a typical 20 GHz TWTA.

NOTE 4: Value of EIRP for the useful carrier at the antenna output.

NOTE 5: Value of C/I computed for a typical 20 GHz TWTA (40-60 W class) at about 5 dB OBO.

NOTE 6: Value when using the turbo coding with rate 0,5 and MPEG packets.

Annex D: Deriving E_b/N_0 from E_S/N_0 - an example

The following provides a definition for the relation between E_b/N_0 and E_S/N_0, based on the RS/Viterbi coding case.

$$\frac{E_b}{N_0} = \frac{1}{Coderate_{RS}} \cdot \frac{1}{Coderate_{Conv}} \cdot \frac{Symbol}{\#Bit} \cdot \frac{E_s}{N_0}$$

As an example for a traffic burst, the following values could be used for the calculation:

$Coderate_{RS} = \frac{53 + 2}{53 + 2 + 16} = \frac{55}{71}$	ATM cell 53 bytes, SAC field 2 bytes Reed Solomon parity: 16 bytes
$Coderate_{Conv} = \frac{1}{2}$	convolutional code,
$\frac{\#\ Bit}{Symbol} = 2$	QPSK
$\frac{E_s}{N_0}$	Signal to Noise Ratio (AWGN Channel)

NOTE: Theoretically the convolutional code rate is slightly smaller than ½ because of the six flushing bits. If we consider the code rate to be the ratio between input bits and output bits we have:

$$Coderate_{Conv} = \frac{N_i}{2(N_i + 6)}$$

where N_i is the bits at the input to the convolutional encoder. For one ATM cell with a 2 byte SAC and 16 RS parity bytes, N_i = 71 bytes or 568 bits. Therefore:

$$Coderate_{Conv} = \frac{568}{2(568+6)} = \left(\frac{1}{2}\right) \cdot \frac{568}{574} = \left(\frac{1}{2}\right) \cdot 0{,}98955$$

This results in a negligible difference of less than 0,05 dB, if the six bits are not considered.

Annex E: Example of used frequency bands

The first deployment of RCSTs is expected to use the following frequency ranges:

- Reception is in one or several of the frequency bands of the Fixed Satellite Service (FSS) or Broadcast Satellite Service:

  10,70 GHz - 11,70 GHz

  11,70 GHz - 12,50 GHz

  12,50 GHz - 12,75 GHz

  17,70 GHz - 19,70 GHz

  19,70 GHz - 20,20 GHz

  21,40 GHz - 22,00 GHz

- Transmission is in one of the frequency bands allocated to FSS:

  14,00 GHz - 14,25 GHz

  27,50 GHz - 29,50 GHz

  29,50 GHz - 30,00 GHz

Other bands are also envisaged. Regulation of usage of frequency bands is covered by other bodies.

Linear or circular polarization is used for transmission or reception.

Annex F:
MIB definition

This clause defines the different SNMP objects that compose the private enterprise RCST MIB. Subgroup after subgroup, each object is reviewed according to the ASN definition structure [22] in order to create a compiled MIB.

MIB-II has been said to be necessary and shall be implemented. However, not all the groups shall be implemented. This MIB is entirely defined in RFCs and described in several textbooks. Hence, the present document provides only the required group definition.

F.1 Information Modules

An "information module" is an ASN.1 module defining information relating to network management. The SMI describes how to use an adapted subset of ASN.1 [22] to define an information module. Further, additional restrictions are placed on "standard" information modules. It is strongly recommended that "enterprise-specific" information modules also adhere to these restrictions.

Typically, there are three kinds of information modules:

- MIB modules, which contain definitions of inter-related managed objects, make use of the OBJECT-TYPE and NOTIFICATION-TYPE macros;
- compliance statements for MIB modules, which make use of the MODULE-COMPLIANCE and OBJECT-GROUP macros [1]; and
- capability statements for agent implementations which make use of the AGENT-CAPABILITIES macros [1] (not used in our case).

The RCST MIB includes definitions of managed objects (first part) and a compliance statement (last part).

F.2 Access Rights

The MAX-ACCESS clause defines whether it makes "protocol sense" to read, write and/or create an instance of the object, or to include its value in a notification. This is the maximal level of access for the object. (This maximal level of access is independent of any administrative authorization policy.)

These values are ordered, from least to greatest: "not-accessible", "accessible-for-notify", "read-only", "read-write", "read-create".

If any columnar object in a conceptual row has "read-create" as its maximal level of access, then no other columnar object of the same conceptual row may have a maximal access of "read-write". (Note that "read-create" is a superset of "read-write".)

The write and read access rights of any SNMP object are defined/identified according to the different users/entities. In the RCST MIB definition within the present document, the following notations are used in the scope of the access rights:

- 'W' stands for 'Write' access
- 'R' stands for 'Read' access
- 'C' stands for 'Create' access
- 'N' stands for 'Not-Accessible' access
- 'H' stands for 'Hub' and therefore means NCC (either SMS or HLM or TM)
- 'I' stands for 'Installer and Service'

- 'A' stands for 'Local Administrator' defined as the 'Super User'
- 'S' stands for 'ISP and SSP'

The access rights to a particular SNMP object are defined cross-checking both the maximum level of access of that SNMP object and the access rights granted to the entity according to its community name.

Table F.1 describes the relationship between SNMPv2 MIB MAX-ACCESS Value and Protocol Access Mode.

Table F.1: relationship between SNMPv2 MIB MAX-ACCESS Value and Protocol Access Mode

	SNMPv2 Protocol Operation	
MAX-ACCESS **Value**	READ-ONLY	READ-WRITE
read-only	Available for get and trap operations	
read-write	for get and trap operations	Available for get, set, and trap operations
read-create	Available for get and trap operations	Available for get, set, create, and trap operations
accessible-for-notify	Available for trap operations	
not-accessible	Unavailable	

The table shall be understood as follows. Defining an SNMP object, one has to give it a maximum access. This access is described by the different rows of the table. But every object belongs to the different views defined for each community. Moreover, its access rights are defined in relation with this community. This can be read in the columns of the table.

The intersection of each row and each column defines a kind of 'availability' of the SNMP object as far as the SNMP actions (get, set, trap, etc) are concerned. Hence, when the view access rights of an SNMP object in a particular view are defined as 'Read-Only' while its maximum access rights are 'Read-Write', this means that the object is somehow available to GET and TRAP operations. Indeed, this object is at most readable and, hence, no SET action can be done.

F.3 SNMP Objects Syntax

Each SNMP object is of a specific type. There exist numerous types and those are defined in different RFCs. The following comments emanate from those RFCs.

The Integer32 type represents integer-valued information between -2^31 and 2^31-1 inclusive (-2147483648 to 2147483647 decimal). This type is indistinguishable from the INTEGER type. Both the INTEGER and Integer32 types may be sub-typed to be more constrained than the Integer32 type.

The INTEGER type (but not the Integer32 type) may also be used to represent integer-valued information as named-number enumerations. In this case, only those named-numbers so enumerated may be present as a value. Note that although it is recommended that enumerated values start at 1 and be numbered contiguously, any valid value for Integer32 is allowed for an enumerated value and, further, enumerated values need not be contiguously assigned.

Note that the 'RowStatus' type is a textual convention defined in RFC 2579 [23] and shall be implemented as such. This syntax is mainly used to declare dynamic tables.

The TimeTicks type represents a non-negative integer which represents the time, modulo 2^32 (4294967296 decimal), in hundredths of a second between two epochs. When objects are defined which use this ASN.1 type, the description of the object identifies both of the reference epochs.

The TimeStamp textual convention is defined in [23] and is based on the TimeTicks type. With a TimeStamp, the first reference epoch is defined as the time when sysUpTime (MIB-II system SNMP object) was zero, and the second reference epoch is defined as the current value of sysUpTime.

F.4 Private Enterprise RCST MIB

This MIB shall be defined under the SNMP Network Management Private Enterprise number assigned to SES by IANA. The number allocated by the IANA for this MIB is 2696 and the name is "dvb" In this proprietary subtree, different system specific subtrees will be defined. The subtree RCS for DVB-RCS systems has got the number 2. The private enterprise RCST MIB is located under RCS with the name RCSTMIB and with number 1. A potential private enterprise NCC MIB would also be located here, as shown in the figure.

The following clauses define each subgroup of the private enterprise RCST MIB group.

F.4.1 RCSTSystem group

Basically, the SNMP variables of this group shall gather some basic information that would allow anyone to trace the history – the life – of the RCST as well as to get a complete description of its constitution on the component point of view.

Lots of parameter will be defined at installation. Hence, a subgroup has been allocated to those.

F.4.1.1 installation subgroup

This subgroup contains all the information related to the RCST installation and the technical staffs that performed this installation. These parameters are believed to stay unchanged once it has been defined during installation. Modification of hardware equipment, maintenance operations and geographical re-location may require an update of those SNMP objects. They are defined in Table F.2.

Note that *RCSTSysInstallLocation* object gives the location of the ODU antenna, which is needed for network operation, while the *system.sysLocation* (MIB-II SNMP OID) provides the location of the IDU unit, which can not be used for the same purpose.

The definition and use of domain names are ruled by the *RCSTSysInstallAllowedDomainTable* table defined in this subgroup and two others SNMP objects – namely *RCSTActDefaultDomainName* and *RCSTActDefaultDomainStatus* – defined in the RCSTActions subgroup (because of access right differences).

The rules about the domain names are the following:

The *RCSTSysInstallAllowedDomainTable* table will list **all** the allowed domains. If this table is populated, any "user@domain" login attempt will be tested against this list, and will not be transmitted to the NCC by the RCST if it is bearing a non-listed "@domain". This list is restrictive. Example: "kiosk" service that allows connectivity to restricted set of ISPs. If there is no entry in this table, there is no restriction on the domain name. If there is a unique entry in this table, this "@domain" will de facto be the enforced "@domain". Example: RCSTs subsidized by an ISP.

One (and only one) of these *RCSTSysInstallAllowedDomainNames* found in the *RCSTSysInstallAllowedDomainTable* table can be set by the Super User as being the *RCSTActDefaultDomainName*, the RCST will then automatically append this domain name to the user login attempts that have no "@domain" explicitly specified. If there is no entry in the *RCSTSysInstallAllowedDomainTable* table, there is no restriction on the *RCSTActDefaultDomainName*. Example: users will not have to type in the complete "user@domain_name.com" string each time they log in.

Table F.2: RCSTSystem.installation subgroup definition

OID	Name	Syntax	Access	Definition / Description
1	**RCSTSysInstallOwner**	DisplayString Size(0..255)	$R_{HIA}W_{I}$	Identifies the technical staff that performed the RCST installation and commission of both ODU and IDU. It also provides contact information and is defined locally by RCST Administrator or Installation Team. It shall only be modifiable by the installer.
2	**RCSTSysInstallDate**	DisplayString Size(0..255)	$R_{HIA}W_{I}$	Identifies the installation date, Work Order number, and other relevant installation information. It shall be defined locally by RCST Administrator or Installation Team and shall only be modifiable by the Maintenance (Update)/Installation Team.
3	**RCSTSysInstallLocation**	DisplayString Size(0..255)	$R_{HIA}W_{I}$	Physical location of the **ODU antenna** expressed as Cartesian co-ordinates 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 metre level.) The Installation Team shall assign the location value to this variable. Floating-point notation shall be used. Format: **x** <space> **y** <space> **z**. Values in metre. The system.sysLocation object of MIB-II provides physical location of the **IDU unit**.
4	**RCSTSysInstallSSPId**	DisplayString Size(0..255)	$R_{HIA}W_{I}$	Satellite Service Provider (SSP) Identifier.
5	**RCSTSysInstallRCSTMacAddr**	MacAddress	R_{HIA}	RCST MAC Address of the Air Interface.
6	**RCSTSysInstallOduAntennaSize**	Integer32	$R_{HIA}W_{I}$	This object gives the diameter of the antenna. This value shall be given in centimetre. Defined at installation. The object can be used in conjunction with environmental requirements.
7	**RCSTSysInstallOduSSPA**	Integer32	$R_{HIA}W_{I}$	This field describes the SSPA installed in the ODU and shall be defined by the installer. The power shall be given in tenth of Watt for more flexibility, i.e. 0,5 W will be represented by 5, 1 W by 10 and 2 W by 20. Defined at installation.
8	**RCSTSysInstallAllowedDomainNextIndex**	Integer32	R_{HIA}	Provides the next available index in the dynamic table *RCSTSysInstallAllowedDomainTable*. The row-creation algorithm shall use this value.
9	**RCSTSysInstallAllowedDomainTable**	SEQUENCE OF RCSTSysInstall AllowedDomain Entry	N	This feature gives the possibility for the RCST to reject logins to any domain not specified as allowable in the RCST without sending a user login request to the NCC. The installer is the only entity that can enable/disable this functionality and enter/modify the allowed @domain names. The allowed @domain names will be permanently stored in the RCST.

9.1	**RCSTSysInstallAllowedDomainEntry**	SEQUENCE	N	SEQUENCE OF {
				RCSTSysInstallAllowedDomainIndex,
				RCSTSysInstallAllowedDomainName,
				RCSTSysInstallAllowedDomainStatus
				}

9.1.1	**RCSTSysInstallAllowedDomainIndex**	Integer32	N	Uniquely identifies each authorized domain name.
9.1.2	**RCSTSysInstallAllowedDomainName**	DisplayString Size(0..255)	$R_{HIA}C_{I}$	Allowed domain name, permanently stored in the RCST and enabled/disabled/defined by the Installer.
9.1.3	**RCSTSysInstallAllowedDomainStatus**	RowStatus	$R_{HIA}C_{I}$	This object enables the table to be dynamic. Moreover, it provides some information about the validity of each row. It also enables creation/deletion of rows.
10	**RCSTSysInstallCompNextIndex**	Integer32	R_{HIA}	Next index available in the table *RCSTSysInstallCompDescrTable*. The row creation algorithm shall use it.

OID	Name	Syntax	Access	Definition / Description
11	**RCSTSysInstallCompDescrTable**	SEQUENCE OF RCSTSysInstallCompDescrEntry	N	Description of the different component of the RCST. This table shall be defined at installation and, though it is defined as a dynamic table, once defined by the installer, it shall never increase nor decrease. It is the installer that will definitively fix its size. This latter may vary from one manufacturer to the other.

11.1	**RCSTSysInstallCompDescrEntry**	SEQUENCE	N	SEQUENCE OF {
				RCSTSysInstallCompDescrIndex,
				RCSTSysInstallCompDescrString,
				RCSTSysInstallCompDescrStatus
				}

11.1.1	**RCSTSysInstallCompDescrIndex**	Integer32	N	Uniquely identifies each row of the table.
11.1.2	**RCSTSysInstallCompDescrString**	DisplayString Size(0..255)	$R_{HIA}C_{I}$	256-character string that gives a thorough description of each component. It shall contain a name reference, a version (HW, SW) and the like.
11.1.3	**RCSTSysInstallCompDescrStatus**	RowStatus	$R_{HIA}C_{I}$	This object enables the table to be dynamic. Moreover, it provides some information about the validity of each row. It also enables creation/deletion of rows.

F.4.1.2 idu subgroup

IDU-related SNMP objects describing physical parameters are defined in this group and defined in Table F.3. They provide some basic and general information about the RCST system.

Note that the shortening "IduShut" refers to "IDU Shutdown".

Table F.3: RCSTSystem.idu subgroup definition

OID	Name	Syntax	Access	Definition / Description
1	**RCSTSysIduShutTxAGCRateThresh**	Integer32	$R_{HIA}W_{IH}$	AGC variation rate threshold, specified in tenth of dB/s, above which the IDU must disable its modulator output.
2	**RCSTSysIduShutTxAGCWindow**	Integer32	$R_{HIA}W_{IH}$	Integration window of the AGC variation rate measurement, specified in seconds.
3	**RCSTSysTargetEbN0**	Integer32	$R_{HIA}W_{IH}$	This value describes the wanted Eb/N0 value that enables operation of the return link with the required error performance. The values shall be given in tenth of dB and the initial value shall be equal to 7 dB. The range shall be from 0 to 31,5 dBm with a precision of 0,1 dB.
4	**RCSTSysLoopGainD**	Integer32 Default = 5	$R_{HIA}W_{IH}$	This value defines the loop gain of the ULPC parameter will affect the dynamic behaviour of the system and determine its properties regarding acquisition and error smoothing performance. The value shall be given in tenths (i.e. 15 stands for 1,5). The initial value shall be equal to 0,5. The range shall be from 0 to 2 with a precision of 0,1.
5	**RCSTSysEbN0Range**	Integer32 Default = 80	R_{HIA}	This value describes the possible range of Eb/N0 variation that can be compensated by the system. It shall be identical to the possible rain-fade in dB and corresponds also to the uplink power dynamic range the RCST can cover. The values shall be given in tenth of dB and the initial value shall be equal to 8 dB. The range shall be from 0 to 20 dB with a precision of 0,1 dB.

F.4.1.3 capability subgroup

This subgroup contains objects that describe the capabilities of the IDU, which shall also be given in the RCST capability field of the CSC burst. In addition this group contains objects defining the capacity request categories supported by the RCST.

Table F.4: RCSTSystem.capability subgroup definition

OID	Name	Syntax	Access	Definition / Description
1	**RCSTSysCapabilitySecurity**	Integer32	$R_{HIA}W_{IH}$	'1' for RCST implementing security mechanism as described in [1], '0' otherwise
2	**RCSTSysCapabilitySNMP**	Integer32	$R_{HIA}W_{IH}$	'1' for RCST supporting SNMP, '0' otherwise
3	**RCSTSysCapabilityATMconnectivity**	Integer32	$R_{HIA}W_{IH}$	'1' for RCST capable of ATM connectivity, '0' for not capable
4	**RCSTSysCapabilityMPEG2TSTRF**	Integer32	$R_{HIA}W_{IH}$	'1' for RCST capable of MPEG2-TS TRF, '0' for not capable
5	**RCSTSysCapabilityRCSTboards**	Integer32	$R_{HIA}W_{I}$	Number of RCST forward link receivers: '00' for 1 receiver, '01' for 2, '10'for more than two, '11' reserved
6	**RCSTSysCapabilityRCSTACQ**	Integer32	$R_{HIA}W_{IHA}$	'0' for RCST not requiring ACQ burst, '1' for ACQ required
7	**RCSTSysCapabilityMultiIdu**	Integer32	$R_{HIA}W_{I}$	'0' for single indoor unit/single outdoor unit configuration, '1' when two or more IDUs are connected to a single ODU
8	**RCSTSysCapabilitySWVersion**	Integer32	$R_{HIA}W_{IHA}$	System dependent. Can be used to define the RCST software version. (bit 7 bit 0)
9	**RCSTSysCapabilityFreqHoppRange**	Integer32	R_{HIA}	Defines the RCST burst to burst frequency hopping range capability: '00' for 20 MHZ. Other patterns system dependent
10	**RCSTSysCapabilityMFTDMA**	Integer32	$R_{HIA}W_{I}$	'1' for RCST supporting dynamic MF-TDMA, '0' for RCST supporting fixed MF-TDMA
11	**RCSTSysCapabilityRCSTClass**	Integer32	$R_{HIA}W_{IH}$	System dependent (2 lsb)
12	**RCSTSysCapabilityRCSTMode**	Integer32	$R_{HIA}W_{IH}$	'000' for Installation mode, '001' for Operational mode, '010' for Reference RCST mode. Other patterns reserved
13	**RCSTSysCapabilityCapacityReqCRA**	Integer32	$R_{HIA}W_{IH}$	'1' CRA supported, '0' CRA not supported
14	**RCSTSysCapabilityCapacityReqRBDC**	Integer32	$R_{HIA}W_{IH}$	'1'RBDC supported, '0' RBDC not supported
15	**RCSTSysCapabilityCapacityReqVBDC**	Integer32	$R_{HIA}W_{IH}$	'1'VBDC supported, '0' VBDC not supported
16	**RCSTSysCapabilityCapacityReqAVBDC**	Integer32	$R_{HIA}W_{IH}$	'1'AVBDC supported, '0' AVBDC not supported
17	**RCSTSysCapabilityCapacityReqFCA**	Integer32	$R_{HIA}W_{IH}$	'1'FCA supported, '0' FCA not supported
18	**RCSTSysCapabilityCapacityReqDCRA**	Integer32	$R_{HIA}W_{IH}$	'1'DCRA supported, '0' DCRA not supported

F.4.2 RCSTConfig group

This subgroup shall be considered as gathering any data that is useful to get access, maintain and enter a session. Hence, things like synchronization, access management, interface parameters, network settings, user information and the likes are part of this group. Some aspects have been grouped under a common title, *airIf*, in the *lines* group in order to create an interface-dependent subgroup.

F.4.2.1 network subgroup

This subgroup shall contain all the SNMP objects related to network parameters, i.e. MAC addresses, IP addresses and the like. Note that this group shall also host a way to link the different interfaces defined in the *Interfaces* and *Ip* MIB-II groups with the kind of traffic they handle. The different SNMP objects are defined in Table F.5.

Note that some parameters have two versions: the "default" one and the current one. When starting, both versions shall be equal. Once the process or the session is ongoing some dynamic procedure/process may ask to modify this value. Then the default value would be saved in the 'default' version while the new value would be allocated to the 'current' variable. Doing so, the default value can be recovered at any time.

In this subgroup, two objects have been defined in order to differentiate between control and user traffic and associate them with a physical interface. Both *RCSTConfigNetworkTrafficIpAddr* (Traffic) and *RCSTConfigNetworkOamIpAddr* (OAM) provide the value of the IP address of, respectively, the user traffic and the control flow. These IP addresses provide a link to MIB-II via the SNMP object *ipAdEntAddr* defined in MIB-II *ip* subgroup. This object represents the IP address of a flow and is uniquely linked to a physical interface designated by the *ipAdEntIfIndex* SNMP object. Indeed, this latter object is equal to a particular value of *ifIndex*, SNMP object of MIB-II *interfaces* subgroup that uniquely identifies each physical interface.

It is necessary to use the IP addresses of both flows because both OAM and Traffic addresses could be allocated to a single physical interface and in this case, it would not be possible to differentiate both flows based on the interface index, for example.

Table F.5: RCSTConfig.network subgroup definition

OID	Name	Syntax	Access	Definition / Description
1	**RCSTConfigNetworkOamIpAddr**	IpAddress	$R_{HIA}W_{HI}$	OAM IP Address of the RCST. This object used with both *ip* and *interfaces* MIB-II subgroups determines uniquely the interface through which OAM traffic is passing through. Note that the OAM IP address may be statically or dynamically assigned. It is system dependent whether the OAM IP address and the Traffic IP address are the same address.
2	**RCSTConfigNetworkOamIpNetworkMask**	IpAddress	$R_{HIA}W_{HI}$	Network Mask for the OAM IP Address.
3	**RCSTConfigNetworkTrfIpAddr**	IpAddress	$R_{HIA}W_{HI}$	Current TRAFFIC IP Address of the RCST. This object used with both *ip* and *interfaces* MIB-II subgroups determines uniquely the interface through which user traffic is passing through. By default, it is equal to the same address as in OID *RCSTConfigNetworkDefaultTrfIpAddr*. This OID, *RCSTConfigNetworkDefaultTrfIpAddr* OID and *RCSTConfigNetworkTrfIpAddrAssign* OID are linked together.
4	**RCSTConfigNetworkDefaultTrfIpAddr**	IpAddress	$R_{HIA}W_{HI}$	**Default** TRAFFIC IP Address of the RCST. This object shall be the initial address used in case the initial TRAFFIC IP address given by OID *RCSTConfigNetworkTrfIpAddr* would change while in session in order to keep track of the "default" address. Its value shall never change while in session. *RCSTConfigNetworkTrfIpAddr* OID, this OID and *RCSTConfigNetworkTrfIpAddrAssign* OID are linked together.
5	**RCSTConfigNetworkTrfIpAddrAssign**	INTEGER	$R_{HIA}W_{HI}$	Identifies whether the TRAFFIC IP address is statically ('static' – 1) or dynamically ('dynamic' – 2) assigned. *RCSTConfigNetworkTrfIpAddr* OID and *RCSTConfigNetworkDefaultTrfIpAddr* OID are linked to this object: when statically assigned, both OIDs represent the same IP address. In case the IP address is dynamically assigned, *RCSTConfigNetworkTrfIpAddr* OID will be the dynamic IP address while *RCSTConfigNetworkDefaultTrfIpAddr* OID remains the static IP address.
6	**RCSTConfigNetworkTrfIpNetworkMask**	IpAddress	$R_{HIA}W_{HI}$	Network Mask for the Traffic IP Address of the RCST.
7	**RCSTConfigNetworkDefaultTrfIpNetworkMask**	IpAddress	$R_{HIA}W_{HI}$	Network Mask for the default Traffic IP Address of the RCST.
8	**RCSTConfigNetworkOamReleaseTimeout**	Integer32	$R_{HIA}W_{HI}$	Time, specified in milliseconds, after which the RCST shall released the Return Link and OAM Resources due to no OAM traffic (from Hosts or RCST) bound for the Return Link, while one or more entries remain in the Host Authentication table.
9	**RCSTConfigNetworkOamIpAddrAssign**	INTEGER	$R_{HIA}W_{HI}$	Identifies whether the OAM IP address is statically ('static' – 1) or dynamically ('dynamic' – 2) assigned.
10	**RCSTConfigNetworkAuthPriIpAddr**	IpAddress	$R_{HIA}W_{HI}$	IP address within the NCC, to which the RCST shall send Auth messages.
11	**RCSTConfigNetworkAuthSecIpAddr**	IpAddress	$R_{HIA}W_{HI}$	IP address within the NCC, to which the RCST shall send Auth messages.

F.4.2.2 accessPolicy subgroup

This group contains objects to support the access policy of the MIB. The access control policy supported by the RCST is described F.4.2.3.

For security reason, an installer or a network operator using a non-SNMP interface shall configure this table. The installer accesses the MIB in order to define it via HTTP.

Table F.6: RCSTConfig.accessPolicy subgroup definition

OID	Name	Syntax	Access	Definition / Description
1	**RCSTConfigAccessPolicyNextIndex**	Integer32	R_{HIA}	Provides the next available index in the dynamic table *RCSTConfigAccessPolicyTable*. The row-creation algorithm shall use this value.
2	**RCSTConfigAccessPolicyTable**	Sequence of RCSTConfigAccessPolicyEntry	N	The entries in this table map domain IP subnets to community names and management entity names to IP addresses. The community names are used by the RCST to determine the kind of access right the originator of an SNMP request has.

2.1	**RCSTConfigAccessPolicyEntry**	Sequence	N	SEQUENCE OF {
				RCSTConfigAccessPolicyIndex,
				RCSTConfigAccessPolicyIpAddr,
				RCSTConfigAccessPolicyNetMask,
				RCSTConfigAccessPolicyCommunityName,
				RCSTConfigAccessPolicyEntity,
				RCSTConfigAccessPolicyStatus
				}

2.1.1	**RCSTConfigAccessPolicyIndex**	Integer32	N	Its value is used to index this table.
2.1.2	**RCSTConfigAccessPolicyIpAddr**	IpAddress	$R_{HIA}C_{HI}$	IP address of a network management entity. Note that this IP address can be any address belonging to the subnet defined using the OID 2.1.3 *RCSTConfigAccessPolicyNetMask*.
2.1.3	**RCSTConfigAccessPolicyNetMask**	IpAddress	$R_{HIA}C_{HI}$	Network mask for the management entity referred to in OID 2.1.5 of this table. Note that this network mask can be set to 255.255.255.255 so that it maps to a unique IP address.
2.1.4	**RCSTConfigAccessPolicyCommunityName**	DisplayString size(0..255)	$R_{HIA}C_{HI}$	Corresponding community name.
2.1.5	**RCSTConfigAccessPolicyEntity**	DisplayString size(0..255)	$R_{HIA}C_{HI}$	Management entity name.
2.1.6	**RCSTConfigAccessPolicyStatus**	RowStatus	$R_{HIA}C_{HI}$	This object enables the table to be dynamic. Moreover, it provides some information about the validity of each row. It also enables creation/deletion of rows.
3	**RCSTConfigAccessPolMibViewNextIndex**	Integer32	R_{HIA}	Provides the next available index in the dynamic table *RCSTConfigAccessPolMibViewTable*. The row-creation algorithm shall use this value.
4	**RCSTConfigAccessPolMibViewTable**	Sequence of RCSTConfigAccessPolMibViewEntry	N	The entries in this table describe the mapping of community names, accessible object groups and the access privileges. An entry contains the community name, the OID branch accessible by members of this community and the level of access privilege.

4.1	**RCSTConfigAccessPolMibViewEntry**	Sequence	N	SEQUENCE OF {
				RCSTConfigAccessPolMibViewIndex,
				RCSTConfigAccessPolMibViewCommunityName,
				RCSTConfigAccessPolMibViewPrefix,
				RCSTConfigAccessPolMibViewAccessRight,
				RCSTConfigAccessPolMibViewStatus
				}

OID	Name	Syntax	Access	Definition / Description
4.1.1	**RCSTConfigAccessPolMibViewIndex**	Integer32	N	Its value is used to index this table.
4.1.2	**RCSTConfigAccessPolMibViewCommunityName**	DisplayString size(0..255)	$R_{HIA}C_{HI}$	Community name of a network management entity.
4.1.3	**RCSTConfigAccessPolMibViewPrefix**	OBJECT IDENTIFIER	$R_{HIA}C_{HI}$	Branch OID of the object groups which members of this community can access.
4.1.4	**RCSTConfigAccessPolMibViewAccessRight**	INTEGER	$R_{HIA}C_{HI}$	Level of access right the community has. The supported values shall be "read-only" ("1"), "read-write" ("2"), and "not-accessible" ("3").
4.1.5	**RCSTConfigAccessPolMibViewStatus**	RowStatus	$R_{HIA}C_{HI}$	This object enables the table to be dynamic. Moreover, it provides some information about the validity of each row. It also enables creation/deletion of rows.

F.4.2.3 Description of the accessPolicy Subgroup

The access policy for all the supported MIB groups are set up using the RCSTConfigAccessPolicyTable and the RCSTConfigAccessPolMibViewTable tables. For security reason, entries to these tables are pre-configured through a non-SNMP interface as part of the commissioning process. When an SNMP request arrives, the validity of the packet's source IP address + community name combination is checked in the *RCSTConfigAccessPolicyTable* table. This community name together with the object ID(s) in the SNMP request determine the access right to the information being requested. This second look up process makes use of the information in the RCSTConfigAccessPolMibView table. Entries in these 2 tables are shown below. Note that the RCSTConfigAccessPolMibView table stands only for first delivery SNMP objects.

Here is a description of the process the RCST shall follow when receiving an SNMP set/get message:

1) The RCST checks in the *RCSTConfigAccessPolicyTable* table if the SNMP request is coming from a valid IP subnet (note that the network mask can be set to 255.255.255.255 so that it maps to a unique IP address) and if this subnet is associated with the given community string.

2) The RCST checks if this specific request is authorized for that community string using the MIB view corresponding to the community name and defined in the *RCSTConfigAccessPolMibViewTable* table.

3) The request is performed.

The process that shall be followed by the RCST when sending a trap is the following:

1) The RCST parses the *RCSTLifeTrapDest* table based on the trap OID. A "Trap Destination Management entity" is associated to each occurrence (in the *RCSTLifeTrapDestTable* table) of this trap OID.

2) The RCST parses the *RCSTConfigAccessPolicyTable* table based on the "Trap destination management entities" ("Management Entity Name"). A *RCSTConfigAccessPolicyIpAddr* is associated to each occurrence (in the *RCSTConfigAccessPolicyTable* table) of these "Trap Destination Management Entities" ("Management Entity Name").

3) Traps are sent to these IP addresses.

Note that not all *RCSTConfigAccessPolicyEntity* entries have to be filled in the *RCSTConfigAccessPolicyTable* table, as this field is only used when traps have to be emitted. When multiple *RCSTConfigAccessPolicyEntity* entries have the same value (e.g. "TM"), this means that the traps will be sent to all associated IP addresses. This method offers extended granularity.

Authentication shall be based on the community names as defined in [12]. The community names shall reflect the different users of a network. The "Service" community name applies to the manufacturer who needs access to the MIB in the scope of some testing issues.

The RCST is said to have three kinds of interface address: LAN IP Address, Traffic IP Address and OAM IP Address. The MIB management system shall be implemented in such a way that any SNMP RCST MIB object shall be accessible using one of these three kinds of IP address (according to the access rights, of course). However, all SNMP objects that have write access for the NCC shall only be accessible/modifiable via the OAM IP address. In a sense, the access rights of any SNMP object can be considered as interface-dependent.

Table F.7: The RCSTConfigAccessPolicyTable

RCSTConfigAccessPolicyIndex	RCSTConfigAccessPolicyIpAddr	RCSTConfigAccessPolicyCommunityName
1	Primary TM IP address	HUBManager
2	Backup TM IP address	HUBManager
3	Primary SMS IP address	HUBManager
4	Backup SMS IP address	HUBManager
5	RCST IP address	SuperUser
6	Service Station IP Address	Service
7	Installer host IP address	Installer
8	ISP or SSP IP address	ISP_SSP
9	Other IP address	public

Next table represents the Access Policy MIB View table that shall be defined by the installer. Note that it shall be modifiable by the NCC.

Table F.8: The RCSTConfigAccessPolMibViewTable

RCSTConfigAccessPolMibViewIndex	RCSTConfigAccessPolMibViewCommunityName	RCSTConfigAccessPolMibViewPrefix	RCSTConfigAccessPolMibViewAccessRight
1	HUBManager	RCSTSysInstall	read-only
2	SuperUser	RCSTSysInstall	read-only
3	Installer	RCSTSysInstall	read-write
4	Service	RCSTSysInstall	read-write
5	Public	RCSTSysInstall	not-accessible
6	ISP_SSP	RCSTSysinstall	not-accessible
7	HUBManager	RCSTSysIdu	read-write
8	SuperUser	RCSTSysIdu	read-only
9	Installer	RCSTSysIdu	read-write
10	Service	RCSTSysIdu	read-write
11	Public	RCSTSysIdu	not-accessible
12	ISP_SSP	RCSTSysIdu	not-accessible
13	HUBManager	RCSTConfigNetwork	read-write
14	SuperUser	RCSTConfigNetwork	read-only
15	Installer	RCSTConfigNetwork	read-write
16	Service	RCSTConfigNetwork	read-write
17	Public	RCSTConfigNetwork	not-accessible
18	ISP_SSP	RCSTConfigNetwork	not-accessible
19	HUBManager	RCSTAccessPol	read-write
20	SuperUser	RCSTAccessPol	read-only
21	Installer	RCSTAccessPol	read-write
22	Service	RCSTAccessPol	read-write
23	Public	RCSTAccessPol	not-accessible
24	ISP_SSP	RCSTAccessPol	not-accessible
25	HUBManager	RCSTConfigLinesAirIfRtnLk	read-write
26	SuperUser	RCSTConfigLinesAirIfRtnLk	not-accessible
27	Installer	RCSTConfigLinesAirIfRtnLk	read-write
28	Service	RCSTConfigLinesAirIfRtnLk	read-write
29	Public	RCSTConfigLinesAirIfRtnLk	not-accessible
30	ISP_SSP	RCSTConfigLinesAirIfRtnLk	not-accessible
31	HUBManager	RCSTConfigLinesAirIfAccess	read-write
32	SuperUser	RCSTConfigLinesAirIfAccess	read-only
33	Installer	RCSTConfigLinesAirIfAccess	read-write
34	Service	RCSTConfigLinesAirIfAccess	read-write
35	Public	RCSTConfigLinesAirIfAccess	not-accessible
36	ISP_SSP	RCSTConfigLinesAirIfAccess	not-accessible
37	HUBManager	RCSTLifeRCSTStatus	read-only
38	SuperUser	RCSTLifeRCSTStatus	read-only
39	Installer	RCSTLifeRCSTStatus	read-only
40	Service	RCSTLifeRCSTStatus	read-only
41	Public	RCSTLifeRCSTStatus	not-accessible
42	ISP_SSP	RCSTLifeRCSTStatus	not-accessible
43	HUBManager	RCSTLifeTrapLog	read-only
44	SuperUser	RCSTLifeTrapLog	read-only

RCSTConfigAccessPolMibViewIndex	RCSTConfigAccessPolMibViewCommunityName	RCSTConfigAccessPolMibViewPrefix	RCSTConfigAccessPolMibViewAccessRight
45	Installer	RCSTLifeTrapLog	read-only
46	Service	RCSTLifeTrapLog	read-only
47	Public	RCSTLifeTrapLog	not-accessible
48	ISP_SSP	RCSTLifeTrapLog	not-accessible
49	HUBManager	RCSTLifeTrapDest	read-write
50	SuperUser	RCSTLifeTrapDest	read-write
51	Installer	RCSTLifeTrapDest	read-write
52	Service	RCSTLifeTrapDest	read-write
53	Public	RCSTLifeTrapDest	not-accessible
54	ISP_SSP	RCSTLifeTrapDest	not-accessible
55	HUBManager	RCSTLifeTrap	not-accessible
56	SuperUser	RCSTLifeTrap	not-accessible
57	Installer	RCSTLifeTrap	not-accessible
58	Service	RCSTLifeTrap	not-accessible
59	Public	RCSTLifeTrap	not-accessible
60	ISP_SSP	RCSTLifeTrap	not-accessible
61	HUBManager	RCSTAct	read-write
62	SuperUser	RCSTAct	read-write
63	Installer	RCSTAct	read-write
64	Service	RCSTAct	read-write
65	Public	RCSTAct	not-accessible
66	ISP_SSP	RCSTAct	not-accessible
67	HUBManager	RCSTCallCntl	read-write
68	SuperUser	RCSTCallCntl	read-only
69	Installer	RCSTCallCntl	read-only
70	Service	RCSTCallCntl	read-only
71	Public	RCSTCallCntl	not-accessible
72	ISP_SSP	RCSTCallCntl	not-accessible
73	HUBManager	RCSTCallCntlTrap	not-accessible
74	SuperUser	RCSTCallCntlTrap	not-accessible
75	Installer	RCSTCallCntlTrap	not-accessible
76	Service	RCSTCallCntlTrap	not-accessible
77	Public	RCSTCallCntlTrap	not-accessible
78	ISP_SSP	RCSTCallCntlTrap	not-accessible
79	HUBManager	RCSTCallCntlMpeg	read-write
80	SuperUser	RCSTCallCntlMpeg	read-only
81	Installer	RCSTCallCntlMpeg	read-only
82	Service	RCSTCallCntlMpeg	read-only
83	Public	RCSTCallCntlMpeg	not-accessible
84	ISP_SSP	RCSTCallCntlMpeg	not-accessible
85	HUBManager	RCSTCallCntlTrapMpeg	not-accessible
86	SuperUser	RCSTCallCntlTrapMpeg	not-accessible
87	Installer	RCSTCallCntlTrapMpeg	not-accessible
88	Service	RCSTCallCntlTrapMpeg	not-accessible
89	Public	RCSTCallCntlTrapMpeg	not-accessible
90	ISP_SSP	RCSTCallCntlTrapMpeg	not-accessible
91	HUBManager	RCSTSysCapability	read-write
92	SuperUser	RCSTSysCapability	not-accessible
93	Installer	RCSTSysCapability	read-write
94	Service	RCSTSysCapability	read-write
95	Public	RCSTSysCapability	not-accessible
96	ISP_SSP	RCSTSysCapability	not-accessible

F.4.2.4 lines subgroup

F.4.2.4.1 airIf subgroup

This subgroup contains parameters that enable the NCC as well as the Super User and the different users to have access to data about the forward and return paths. Some information about the synchronization and medium access process are also provided. Those data are managed by the NCC and shall not be accessible to other parties except for some particular SNMP objects (mainly used for statistics) that shall be modifiable (reset) by the Super User.

F.4.2.4.1.1 rtnLk subgroup

This clause lists all the parameters that characterize the return link.

Up to now, the RCST is only able to deal with one return link at a time. Hence, there was no need to define a table to collect the different SNMP objects, as it will be done for the forward.

Table F.9: RCSTConfig.lines.airIf.rtnLk subgroup definition

OID	Name	Syntax	Access	Definition / Description
1	**RCSTConfigLinesAirIfRtnLkMaxEirp**	Integer32	$R_{HI}W_{HI}$	Maximum allowed EIRP in tenth of dBm on the return link.
2	**RCSTConfigLinesAirIfRtnLkFreqHopRange**	Integer32	R_{HI} W_{HI}	Gives the frequency hopping range of the RCST on the return channel. The value shall be given in kHz. Up to now, 20 000 kHz and 40 000 kHz are the only supported values. Refer to [1] for more details – RCST Capability 24-bit field.
3	**RCSTConfigLinesAirIfRtnLkDefIfLevel**	Integer32	$R_{HI}W_{HI}$	**Default** transmitted IF power level, specified in tenth of dBm, out of the IDU for sending a CSC burst at RCST reboot or power on.
4	**RCSTConfigLinesAirIfRtnLkFirstIflevel**	Integer32	$R_{HI}W_{HI}$	IF level at which the RCST received a first response during installation when sending CSC bursts repeatedly with increasing level. (By adding a specific offset to this level the RCST determines during installation the default value of *RCSTSysInstallRtnLkDefIFLevel*.) Its value shall be given in tenth of dBm.

F.4.2.4.1.1.1 access subgroup

This clause describes the different parameters characterizing the access procedure. As there are two different mode of access process: the normal mode and the abnormal mode used in case of disaster recovery. Hence, a two-row table has been defined. In this group, thresholds and counters are defined giving information about the access procedure health status.

Table F.10: RCSTConfig.lines.airIf.access subgroup definition

OID	Name	Syntax	Access	Definition / Description
1	**RCSTConfigLinesAirIfAccessMode**	INTEGER	R_{HIA}	Indicates whether the RCST is performing a "normal" ('1') logon procedure or an "abnormal" ('2') access procedure (i.e. a disaster recovery process).
2	**RCSTConfigLinesAirIfAccessTable**	Sequence of RCSTAccessEntry.	N	This table gathers all the different times/delays that are relevant to an access procedure whatever the mode: normal or abnormal.

2.1	**RCSTConfigLinesAirIfAccessEntry**	Sequence	N	Sequence of {
				RCSTConfigLinesAirIfAccessIndex,
				RCSTConfigLinesAirIfAccessTinitial,
				RCSTConfigLinesAirIfAccessTto,
				RCSTConfigLinesAirIfAccessTresendMin,
				RCSTConfigLinesAirIfAccessTresendMax,
				RCSTConfigLinesAirIfAccessCscMax,
				RCSTConfigLinesAirIfAccessNMax,
				RCSTConfigLinesAirIfAccessTBackOffN
				}

2.1.1	**RCSTConfigLinesAirIfAccessIndex**	Integer32	N	A unique value characterizes each access mode. This matches the *RCSTConfigLinesAirIfAccessMode* value, in fact.

OID	Name	Syntax	Access	Definition / Description
2.1.2	**RCSTConfigLinesAirIfAccessTinitial**	Integer32	$R_{HIA}W_{HI}$	This object defines the maximum delay the RCST will wait before it issues the first CSC slot to get connected. This value shall be set to 0 when the RCST is in the normal connection mode. It is part of the abnormal login process. Value shall be defined in multiples of superframe. Refer to [1] for more details – abnormal login procedure.
2.1.3	**RCSTConfigLinesAirIfAccessTto**	Integer32	$R_{HIA}W_{HI}$	This variable represents a **fixed** time-out delay the RCST has to wait in the random access scheme after having sent a CSC slot. This time shall be defined in multiples of superframe. Refer to the [1] for more details.
2.1.4	**RCSTConfigLinesAirIfAccessTresendMin**	Integer32 (Default = 0)	$R_{HIA}W_{HI}$	This object defines the minimum delay the RCST shall wait before retrying to send a CSC slot when no reply was received after *RCSTConfigLinesAirIfAccessTto*. In [1], this value is equal to 0. Refer to [1] for more details – Tresend parameter.
2.1.5	**RCSTConfigLinesAirIfAccessTresendMax**	Integer32	$R_{HIA}W_{HI}$	This object defines the maximum delay the RCST shall wait before retrying to send a CSC slot when no reply was received after *RCSTConfigLinesAirIfAccessTto*. In [1], this parameter is referred to as "Tresend". Refer to [1] for more details – Tresend parameter.
2.1.6	**RCSTConfigLinesAirIfAccessCscMax**	Integer32	$R_{HIA}W_{HI}$	This object gives the maximum number of CSC slots / connection attempts a RCST is allowed to issue. Once this threshold has been reached, the RCST shall wait for *RCSTConfigLinesAirIfAccessTBackOffn* time before retrying the complete login process. Moreover, every time this amount is exceeded, *RCSTConfigLinesAirIfAccessN* shall be incremented by 1. Refer to [1] for more details – normal login **procedure.**
2.1.7	**RCSTConfigLinesAirIfAccessNMax**	Integer32	$R_{HIA}W_{HI}$	This threshold shall define the maximum value for an internal “AccessN" counter in the terminal. Once this threshold has been reached, the RCST shall wait for an operator intervention before retrying a complete login procedure. In other words, this parameter represents the maximum number of time acquisition is retried before the IDU must disable its modulator output. Refer to [1] for more details – normal login procedure.
2.1.8	**RCSTConfigLinesAirIfAccessTBackOffN**	Integer32	$R_{HIA}W_{HI}$	This defines the time a RCST has to wait before retrying the complete login process once it has reached the maximum amount of CSC slots it is allowed to use for this purpose. Note that this delay is defined according to an exponential back-off process, which is function of the internal “AccessN" counter in the terminal.
3	**RCSTConfigLinesAirIfAccessNetId**	Integer32	$R_{HIA}W_{IH}$	Interactive network identifier.
4	**RCSTConfigLinesAirIfAccessPopId**	Integer32	$R_{HIA}W_{IH}$	Population identifier.
5	**RCSTConfigLinesAirIfAccessStartTranspFreq**	Integer32	$R_{HIA}W_{HI}$	Frequency of the start transponder carrying a Network Information Table to which any RCST shall trigger to acquire forward link. Its value shall be given in multiple of 100 kHz.

OID	Name	Syntax	Access	Definition / Description
6	**RCSTConfigLinesAirIfAccess StartTranspOrbPos**	Intege32	$R_{HIA}W_{HI}$	Orbital position of the satellite casting the start transponder that carries an Network Information Table to which any RCST shall trigger to acquire forward link. Its value shall be given in tenth of degree, i.e. '1234' would represent the 123,4 degrees position. The installer shall define this parameter.
7	**RCSTConfigLinesAirIfAccess StartTranspPolar**	Integer32	$R_{HIA}W_{HI}$	2-bit field giving the polarization of the start transponder carrying an Network Information Table to which any RCST shall trigger to acquire forward link: linear and horizontal ('00'), linear and vertical ('01'), circular left ('10'), circular right ('11')
8	**RCSTConfigLinesAirIfAccess StartTranspFec**	Integer32	$R_{HIA}W_{HI}$	Specifies the Forward Error Correction used on the start transponder carrying a NIT to which any RCST shall trigger to acquire forward link.
9	**RCSTConfigLinesAirIfAccess StartTranspSymbRate**	Integer32	$R_{HIA}W_{HI}$	Specifies the symbol rate on the start transponder carrying an Network Information Table to which any RCST shall trigger to acquire forward link. Its value shall be given in multiple of 100 symbol/s.

F.4.3 RCSTLife group

F.4.3.1 RCSTStatus subgroup

This clause shall describe the operational and network status of the RCST at any time while the terminal is up and running.

Table F.11: *RCSTLife.RCSTStatus* subgroup definition

OID	Name	Syntax	Access	Definition / Description
1	**RCSTLifeRCSTStatusMode**	INTEGER	R_{HIA}	Indicates the current operational mode of the RCST. Supported values are: 'commissioning_mode' ('1'), 'normal_operational_mode' ('2'), 'reference_RCST_mode' ('3'). Other values are reserved for future use Refer to [1] for more details – RCST Capability 24-bit field.
2	**RCSTLifeRCSTStatusCurrent**	INTEGER Default = 1	R_{HIA}	Defines the current RCST operational status: 'idle' ('1'), 'initialized' ('2'), 'hold' ('3'), 'oam_active' ('4'), 'active' ('5'), 'fault ('6') Other values shall be reserved for future use.
3	**RCSTLifeRCSTStatusCurrentTimestamp**	TimeStamp	R_{HIA}	Timestamp indicating at what time the current operational status has been reached by the RCST. Every time the RCST status changes, the value of this object shall be assigned to *RCSTLifeRCSTStatusPreviousTimeStamp* and then updated to be equal to the *sysUpTime* MIB-II object value.
4	**RCSTLifeRCSTStatusPrevious**	INTEGER Default = 1	R_{HIA}	Defines the previous RCST operational status: 'idle' ('1'), 'initialized' ('2'), 'hold' ('3'), 'oam_active' ('4'), 'active' ('5'), fault ('6') Other values shall be reserved for future use.
5	**RCSTLifeRCSTStatusPreviousTimestamp**	TimeStamp	R_{HIA}	Timestamp indicating at what time the previous operational status has been reached by the RCST. This object receives the old value of *RCSTLifeRCSTStatusCurrentTimestamp* every time the RCST status changes.
6	**RCSTLifeRCSTStatusRebootStatus**	INTEGER	R_{HIA}	Provide some information about the result of the reboot process. The following values shall be supported: success ('1') failure ('2')
7	**RCSTLifeRCSTStatusDownloadStatus**	INTEGER	R_{HIA}	Provide some information about the result of the download process. The following values shall be supported: success ('1') failure ('2')
8	**RCSTLifeRCSTStatusCurrentImageId**	Integer32	R_{HIA}	Current image of the Software Download material. RCST Images will be distributed to the RCST in one binary file. In the Software download the NCC is the FTP server, the RCST is the FTP client and the transfer shall be made in binary mode.

OID	Name	Syntax	Access	Definition / Description
9	**RCSTLifeRCSTStatusCurrentImageValidFlag**	INTEGER	R_{HIA}	Status of the image ID given by RCSTLifeRCSTStatusCurrentImageId. The following values are supported: 'valid' (1) 'not_valid' (2)
10	**RCSTLifeRCSTStatusAlternateImageId**	Integer32	R_{HIA}	Alternate image of the Software Download material. RCST Images will be distributed to the RCST in one binary file. In the Software download the NCC is the FTP server, the RCST is the FTP client and the transfer shall be made in binary mode.
11	**RCSTLifeRCSTStatusAlternateImageValidFlag**	INTEGER	R_{HIA}	Status of the image ID given by RCSTLifeRCSTStatusAlternateImageId. The following values are supported: 'valid' (1) 'not_valid' (2)

F.4.3.2 trapLog subgroup

This subgroup acts as a log for all the traps sent by the RCST. It can be seen as a history of the past 255 traps sent to the remote manager. It also provides a mechanism for the remote manager to detect any lost trap and to recover information, which may otherwise be lost.

Table F.12: RCST MIB *RCSTLife.trapLog* subgroup

OID	Name	Syntax	Access	Definition / Description
1	**RCSTLifeTrapLogTable**	Sequence of RCSTLifeTrapLogEntry	N	This table contains a history of the past 255 traps this RCST has sent to the remote manager. It provides a mechanism for the remote manager to detect any lost trap and to recover information, which may otherwise be lost. A trap is logged in this table before it is sent. In the event that a trap gets lost, the remote manager can detect the loss by a gap in the index value of the variables included in the trap. It can then choose to retrieve the information, which were sent in the trap. Note that this table does not log the traps sent to the TM manager.

1.1	**RCSTLifeTrapLogEntry**	Sequence	N	sequence of {
				RCSTLifeTrapLogIndex,
				RCSTLifeTrapLogNumber,
				RCSTLifeTrapLogEventTime
				}

1.1.1	**RCSTLifeTrapLogIndex**	Integer32 (0..255)	N	Table index, which uniquely identifies an entry in this table. Note that even though this table will have a maximum number of entries, say 255, it should be implemented as a wrap-around FIFO queue with a monotonously increasing index number up to the maximum representable by the processor to facilitate the detection of lost traps.
1.1.2	**RCSTLifeTrapLogNumber**	OBJECT IDENTIFIER	R_{HIA}	The trap OID.
1.1.3	**RCSTLifeTrapLogEventTime**	TimeStamp	R_{HIA}	Identifies the time of occurrence of the event that generated the current Trap. The time shall be given according to the *sysUpTime* SNMP MIB-II value.

F.4.3.3 trapDest subgroup

In this subgroup, a table has been defined in order to allocate a destination to each trap that will be sent from the RCST. Together with the Access Policy table defined in the access policy subgroup, this table gives a relation between any trap and an IP address, being its destination.

Table F.13: RCST MIB *RCSTLife.trapDest* subgroup

OID	Name	Syntax	Access	Definition / Description
1	**RCSTLifeTrapDestTableNextIndex**	Integer32	R_{HIA}	The next available index to be used for a new row creation in the trap destination table.
2	**RCSTLifeTrapDestTable**	Sequence of RCSTLifeTrapDestEntry	N	This table defines the destination entities for traps generated by the RCST SNMP agent. Typical configuration for the RCST is to send network management traps to the SMS, and call control traps to the TM in the NCC.

2.1	**RCSTLifeTrapDestEntry**	Sequence	N	SEQUENCE OF {
				RCSTLifeTrapDestIndex,
				RCSTLifeTrapDestEntity,
				RCSTLifeTrapDestOid,
				RCSTLifeTrapDestStatus
				}

2.1.1	**RCSTLifeTrapDestIndex**	Integer32	N	A value that uniquely identifies this entry.
2.1.2	**RCSTLifeTrapDestEntity**	DisplayString Size(0..255)	$R_{HIA}C_{HIA}$	Trap destination management entity.
2.1.3	**RCSTLifeTrapDestOid**	OBJECT IDENTIFIER	$R_{HIA}C_{HIA}$	The OID of the trap to be sent.
2.1.4	**RCSTLifeTrapDestStatus**	RowStatus	$R_{HIA}C_{HIA}$	The entity configuring this destination entry should set this variable to Active once it has provided all the necessary information in this table and is ready to receive traps. If the destination community address and subnet have not been configured, this set will fail. Traps will only be sent to those destinations whose status is Active. To suspend receiving traps, the owner of the destination can set this variable to notInService. To remove an entry from this table, set the status to Destroy.

F.4.3.4 trap subgroup

RCST shall support the following SNMPv2 Enterprise Traps sent to the RCST Management System (SMS) in the NCC.

Table F.14: RCST MIB *RCSTLife.trap* subgroup – Notification definition table

OID	Name	Syntax	Access	Definition / Description
1	**RCSTLifeTrapRCSTDownloadStatus**	**Notification**	N	Sent by RCST to SMS upon software download completion in order to inform the SMS about the success or failure of the process. Contains the following varbinds: RCSTSysInstallRCSTMacAddr RCSTSysInstallSSPId RCSTLifeRCSTStatusDownloadStatus (failure or success)RCSTLifeTrapLogEventTime
2	**RCSTLifeTrapRCSTRebootStatus**	**Notification**	N	Sent by RCST to SMS in the scope of the RCST Reboot action (part of Software Upgrade/Download process). This event is sent in order to provide the NCC with information about the success or the failure of the action. Contains the following varbinds: RCSTSysInstallRCSTMacAddr RCSTSysInstallSSPId RCSTLifeRCSTStatusCurrentImageId RCSTLifeRCSTStatusRebootStatus (failure or success) RCSTLifeTrapLogEventTime
3	**RCSTLifeTrapPingStatus**	**Notification**	N	Always sent after completion of the Ping Device command requested by the SMS. Contains the following varbinds: RCSTSysInstallRCSTMacAddr, RCSTSysInstallSSPId, RCSTLifeTrapLogEventTime, RCSTActPingResult Value, RCSTActPingTableIndex

F.4.4 RCSTCallCntl subgroup – RCST-TM Interface MIB

F.4.4.1 callCntl Group

The *callCntl* group contains objects to support the call control interface between the RCST and the TM in the NCC. The MIB file for these objects will not be generally distributed as they instrument an internal control interface. Also for security reason, the TM is the only SNMP manager, which is allowed to access objects in this group.

Table F.15: RCST-TM Interface MIB *callCntlObject* Group Objects

OID	Name	Syntax	Access	Definition / Description
1	**RCSTCallCntlTmRCSTId**	PhysAddress	R_{HIA}	This object is used by the Traffic Manager to identify the RCST sessions it controls. Each SNMP trap exchange between a RCST and the TM must contain this identifier. The default value for this variable is the RCST's MAC address which is the one sent in the CSC burst and will be initialized to such by the RCST.
2	**RCSTCallCntlAtmConnTableNextIndex**	Integer32	R_{HIA}	The next available index to be used for a new row creation in the trap destination table.
3	**RCSTCallCntlAtmConnTable**	Sequence of RCSTCallCntlAtmConnEntry	N	This table contains information about the ATM connections between the NCC and the RCST.

3.1	**RCSTCallCntlAtmConnEntry**	SEQUENCE	N	SEQUENCE OF {
				RCSTCallCtnlAtmConnIndex,
				RCSTCallCtnlAtmConnVpi,
				RCSTCallCtnlAtmConnVci,
				RCSTCallCtnlAtmConnType,
				RCSTCallCntlAtmConnCraMax,
				RCSTCallCntlAtmConnVbdcMax,
				RCSTCallCtnlAtmConnStatus
				}

3.1.1	**RCSTCallCntlAtmConnIndex**	Integer32	N	This is the table index, which uniquely identifies an entry in this table. To avoid lengthy negotiation between the TM and the RCST on which index to use, the following implementation should be followed: index = 1 is reserved for the OAM VCC; index = 2 is reserved for the traffic VCC which the TM autonomously creates for the RCST at start up.
3.1.2	**RCSTCallCntlAtmConnVpi**	Integer32	$R_{HIA}C_H$	The VPI of an ATM connection between the NCC and the RCST.
3.1.3	**RCSTCallCntlAtmConnVci**	Integer32	$R_{HIA}C_H$	The VCI of an ATM connection between the NCC and the RCST.
3.1.4	**RCSTCallCntlAtmConnType**	INTEGER	$R_{HIA}C_H$	Indicates whether the connection is used to carry OAM traffic or user traffic. Supported values shall be oam_traffic (1) and user_traffic (2).
3.1.5	**RCSTCallCntlAtmConnCraMax**	Integer32	$R_{HIA}C_H$	Maximum number of CRA bytes per second allocated to the corresponding VCC. This object is useful for the queuing mechanism. This object shall be defined through the TIM message for the OAM traffic. While for the user traffic, it shall be set via SNMP Set message issued by the TM
3.1.6	**RCSTCallCntlAtmConnVbdcMax**	Integer32	$R_{HIA}C_H$	Maximum number of VBDC time slots per frame allocated to the corresponding VCC. This object is useful for the queuing mechanism. This object shall be defined through the TIM message for the OAM traffic. While for the user traffic, it shall be set via SNMP Set message issued by the TM

OID	Name	Syntax	Access	Definition / Description
3.1.7	**RCSTCallCntlAtmConnStatus**	RowStatus	$R_{HIA}C_H$	The entries in the connection table are typically created or deleted either by the RCST installer or by the TM. When the connection has been successfully created, this variable should be set to active (1). To delete a connection, set this variable to destroy (6). This variable is initialized to notInService (2) when a row is created. Protocol: When the RCST wishes to obtain traffic transport capacity from the NCC (i.e. to transition from the OAM_active to the active state), it sends a trap to the TM to make a VCC request. The TM should utilize this table to communicate the allocated resource to the RCST. In the event that allocation has failed, set the VPI and VCI to 0 and this variable to notInService (4). As a response to the RCST's traffic VCC release request trap, the TM should set the corresponding *RCSTCallCntlAtmConnStatus* to destroy so that the RCST SNMP agent can clean up the table entry. In the case of the OAM VCC, the RCST will delete the entry after it has sent out the *RCSTCallCntlTrapOamReleaseRequest* trap.

4	**RCSTCallCntlRCSTHoldCommand**	INTEGER:
		clear (1);
		hold (2)
		Default = 1
			$R_{HIA}W_H$	This variable is used by the Traffic Manager to issue a Hold command to the RCST. The Hold command implies a logoff and hold if the RCST is not already in the Initialized state. It will disable the Return Link and the RCST will transition from the Initialized to the Hold state. This variable will be reset when RCST Enable Command comes back.
				If RCST is already in Hold state then no response will be sent to any new Hold Command.

5	**RCSTCallCntlRCSTLogoffConfirm**	INTEGER: not_confirmed (1); confirmed (2)	$R_{HIA}W_H$	This variable is used by the Traffic Manager as a response to the RCST's *RCSTCallCntlTrapRCSTLogoffRequest* trap (OID 5 Table F.16) (TM confirms RCST Log-off to the RCST). Setting this variable to Confirmed (2) will trigger the RCST to do its own cleanup.

6	**RCSTCallCntlRCSTLogoffCommand**	INTEGER:
		clear (1);
		logoff (2)
		Default = 1
			$R_{HIA}W_H$	This variable is used by the Traffic Manager to issue a RCST Logoff command to the RCST. (Sent by TM to RCST through an SNMP Set Request)
				This variable will be set to Clear (1) when the RCST has successfully re-logged in, i.e. successfully getting OAM VCC assigned by the TM.

7	**RCSTCallCntlUserLogoffTableNextIndex**	Integer32	R_{HIA}	The next available index to be used for a new row creation in the trap destination table.

OID	Name	Syntax	Access	Definition / Description
8	**RCSTCallCntlUserLogoffTable**	Sequence of RCSTCallCntlUserLogoffEntry	N	This table is used to track the status of the User logoff request to the TM. The RCST issues the user logoff request to the TM using an SNMP trap. When the TM has finished cleaning the user resources it will set the Logoff status variable to indicate that logoff is complete. Entries in this table are therefore temporary. When the logoff process has been completed, its corresponding entry will be removed. This table will only contain Username / Host IP address entries that are also present in the RCST's User Authentication table. The RCST User table could have easily supported this functionality. The reason to have a separate table is to cleanly separate this control interface from the network management interface at the cost of some small duplication.

8.1	**RCSTCallCntlUserLogoffEntry**	SEQUENCE	N	SEQUENCE OF {
				RCSTCallCntlUserLogoffIndex,
				RCSTCallCntlUserName,
				RCSTCallCntlUserHostIpAddr,
				RCSTCallCntlUserLogoffStatus,
				RCSTCallCntlUserStatus
				}

8.1.1	**RCSTCallCntlUserLogoff Index**	Integer32	N	Index of entry in the User Logoff Request Table.
8.1.2	**RCSTCallCntlUserName**	DisplayString size(0..255)	R_{HIA}	Name of the user who has requested the logoff.
8.1.3	**RCSTCallCntlHostIpAddr**	IpAddress	R_{HIA}	IP address of Host from which User has logged in.
8.1.4	**RCSTCallCntlUserLogoffStatus**	INTEGER: pending (1); complete (2) failed (3)	$R_{HIA}C_H$	Status of the User Logoff Request. Initially set to 'pending' (1) when a new entry is created by the RCST SNMP agent in this table. This is to identify that a User Logoff Request has been issued to the TM but is pending. When the Logoff process has been completed, the TM will set this variable to 'complete' (2). This has the cascading effect of deleting the corresponding entry of the same Username from the User table and also deleting this particular entry from this table. In case there would be no reply from the TM, a time-out (the same as the one used at login – Login Timer) would be raised and the variable would be set to "failed" (3). The user would then receive a notification about the logoff failure and would have to issue a new trial manually. Note that the NCC shall also be able to set this object to 'failed' in case something would go wrong at the NCC side.
8.1.5	**RCSTCallCntlUserStatus**	RowStatus	$R_{HIA}C_H$	This object enables the table to be dynamic. Moreover, it provides some information about the validity of each row. It also enables creation/deletion of rows.

F.4.4.2 callCntlTrap Group

The RCST shall support the following call control Traps, for ATM, sent to the Traffic Manager (TM) in the NCC:

Table F.16: RCST-TM Interface MIB *callCntlTrap* Group Objects

OID	Name	Syntax	Access	Definition / Description
1	**RCSTCallCntlTrapTrafficVCCRequest**	Notification	N	Sent by the RCST to the TM to request a Traffic VCC as part of a re-login process, to transition from the OAM Active state to the Active state and acquire a Traffic VCC to support Host traffic. Contains the following varbinds: RCSTCallCntlTmRCSTId RCSTCallCntlAtmConnTableNextIndex The TM should communicate the response to this request by creating an entry in the *RCSTCallCntlAtmConnTable* using the provided index.
2	**RCSTCallCntlTrapOamReleaseRequest**	Notification	N	Sent by the RCST to the TM to request the release of the current OAM RL resources, to transition from the OAM Active state to the Initialized state. Contains the following varbinds: RCSTCallCntlTmRCSTId.
3	**RCSTCallCntlTrapTrafficReleaseRequest**	Notification	N	Sent by the RCST to the TM to request the release of the current Traffic RL resources, but keep the current OAM RL resources, to transition from the Active state to the OAM Active state. Contains the following varbinds: RCSTCallCntlTmRCSTId RCSTCallCntlAtmConnIndex The TM should communicate the response to this request by destroying an entry in the *RCSTCallCntlAtmConnTable* using the provided index.
4	**RCSTCallCntlTrapUserLogoffRequest**	Notification	N	Sent by the RCST to the TM to request the release of resources for a specific User, i.e. User Logoff. Contains the following varbinds: RCSTCallCntlTmRCSTId RCSTCallCntlUserLogoffIndex RCSTCallCntlUserName RCSTCallCntlHostIpAddr The TM should communicate the response to this request by setting the corresponding *RCSTCallCntlUserLogoffStatus* variable to Complete (2).
5	**RCSTCallCntlTrapRCSTLogoffRequest**	Notification	N	Sent by the RCST to the TM (for RCST log-off) to request the release of all current Return Link resources, to transition from the Active or OAM Active state to the Initialized or Hold state. Contains the following varbinds: RCSTCallCntlTmRCSTId The protocol is such that when the TM has finished its processing of this request, it will set the *RCSTCallCntlRCSTLogoffConfirm* variable, so that the RCST can do its own cleanup.

F.4.4.3 callCntlMpeg group

The *callCntlMpeg* group contains objects to support the call control interface between the RCST and the TM in the NCC. The MIB file for these objects will not be generally distributed as they instrument an internal control interface. Also for security reason, the TM is the only SNMP manager, which is allowed to access objects in this group.

Table F.17: RCST – TM Interactive MIB callCntlMpeg group objects

OID	Name	Syntax	Access	Definition / Description
1	**RCSTCallCntlMpegConnTableNextIndex**	Integer32	R_{HIA}	The next available index to be used for a new row creation in the MPEG Conn. Table.
2	**RCSTCallCntlMpegConnTable**	Sequence of RCSTCallCntlMpegConnEntry	N	This table contains information about the MPEG connections between the NCC and the RCST.

2.1	**RCSTCallCntlMpegConnEntry**	SEQUENCE	N	SEQUENCE OF {
				RCSTCallCtnlMpegConnIndex,
				RCSTCallCtnlMpegConnType,
				RCSTCallCtnlMpegConnPid,
				RCSTCallCtnlMpegEntryStatus
				}

2.1.1	**RCSTCallCntlMpegConnIndex**	Integer32	N	This is the table index, which uniquely identifies an entry in this table. Index = 1 is reserved for the CTRL_MNGM PID (DULM), index = 2 is reserved for the traffic PID which the TM autonomously creates for the RCST at start up, they must be inserted by the RCST SNMP agent after a successful RCST logon.
2.1.2	**RCSTCallCntlMpegConnType**	Integer32	$R_{HIA}C_{H}$	Type of connection to which the PID is associated: 1- CTRL_MNGM (DULM traffic) 2- TRF (user traffic) 3- Native Mpeg (traffic) 4- Additional TRF PIDs
2.1.3	**RCSTCallCntlMpegConnPid**	Integer32	$R_{HIA}C_{H}$	The PID used for this type of connection between the RCST and the NCC. Bit 13-31 unused.
2.1.4	**RCSTCallCntlMpegEntryStatus**	RowStatus	$R_{HIA}C_{H}$	The entries in the MPEG connection table are typically created or deleted either by the RCST SNMP agent or by the NCC network manager. If the entry is present, entry status shall be set to (1) active.

F.4.4.4 callCntlTrapMpeg group

The RCST shall support the following call control Traps, for MPEG, sent to the Traffic Manager (TM) in the NCC:

Table F.18: RCST – TM Interactive MIB CallCntlTrapMpeg trap definition

OID	Name	Syntax	Access	Definition / Description
1	**RCSTCallCntlTrapMpegPidRequest**	Notification	N	This trap is used by the RCST to request a PID cf RCSTCallCntlMpegConnType 3 or 4. Contains the following varbinds: RCSTCallCntlTmRCSTId, RCSTCallCntlMpegConnTableNextIndex, RCSTCallCntlMpegConnType The TM shall respond to this request by creating an entry in the RCSTCallCntlMpegConnTable using the provided index. The RCSTCallCtnlMpegEntryStatus variable sent from TM is set to 4 (create and go)
2	**RCSTCallCntlTrapMpegPidReleaseRequest**	Notification	N	This trap is used to release a PID for MPEG. Contains the following varbinds: RCSTCallCntlTmRCSTId, RCSTCallCntlMpegConnIndex The TM shall respond to this request by removing an entry from the RCSTCallCntlMpegConnTableusing the provided index. The RCSTCallCtnlMpegEntryStatus variable sent from TM is set to 6 (destroy)

F.4.5 RCSTActions group

This MIB group contains objects a network manager can use to invoke actions and tests supported by the RCST agent and to retrieve the action/test results.

Table F.19: *RCST.RCSTActions* subgroup

OID	Name	Syntax	Access	Definition / Description
1	**RCSTActDownloadUrl**	DisplayString Size(0..255)	$R_{HIA}W_{HIA}$	This object provides the complete URL that is used by the download process. This URL is made up of a identification (login), a password as well as the complete file path.
2	**RCSTActDownloadOwner**	DisplayString Size(0..255)	$R_{HIA}W_{HIA}$	Identifies the entity that has initiated the download process.
3	**RCSTActDownloadOwnerIpAddr**	IpAddress	$R_{HIA}W_{HIA}$	Provides the IP address of the entity that has initiated the download process. Note that this entity is named in *RCSTActDownloadOwner.*
4	**RCSTActPingTableNextIndex**	Integer32	R_{HIA}	The next available table index for new row creation. (Suggestion: it may be agreed on not to support this variable. The manager can just use some random number for the table index)
5	**RCSTActPingTable**	Sequence of RCSTActPingEntry	N	This table supports the Ping command issued from a network management entity to the RCST.

5.1	**RCSTActPingEntry**	SEQUENCE	N	SEQUENCE OF {
				RCSTActPingTableIndex,
				RCSTActPingDestination,
				RCSTActPingFrom,
				RCSTActPingSize,
				RCSTActPingResultTimeStamp,
				RCSTActPingResult,
				RCSTActPingTime,
				RCSTActPingOwnerIp,
				RCSTActPingRowStatus
				}

5.1.1	**RCSTActPingTableIndex**	Integer32	N	An index uniquely identifying an entry in this table.
5.1.2	**RCSTActPingDestination**	IpAddress	$R_{HIA}C_{HIA}$	Identifies the device to be pinged by the RCST using its IP address.
5.1.3	**RCSTActPingFrom**	INTEGER	$R_{HIA}C_{HIA}$	Identifies the RCST interface to issue the ping command from, either the OAM IP address ('oam_ip_address' "1") or the Traffic IP address (traffic_ip_address' "2").
5.1.4	**RCSTActPingSize**	Integer32	$R_{HIA}C_{HIA}$	Specifies the number of bytes to include in the ping command.
5.1.5	**RCSTActPingResultTimeStamp**	TimeStamp	R_{HIA}	The value of *sysUpTime* of the last Ping result.
5.1.6	**RCSTActPingResult**	INTEGER	R_{HIA}	Result of the last ping command. The supported values shall be: 'none' (1), 'pass' (2), 'failed' (3), 'inProgress' (4).
5.1.7	**RCSTActPingTime**	Integer32	R_{HIA}	Time response, in milliseconds, of the last ping command. Set to (0) if ping command has failed.
5.1.8	**RCSTActPingOwnerIpAddr**	IpAddress	$R_{HIA}C_{HIA}$	Identifies the entity that has initiated the Ping. It can be used to avoid collision in a multi-manager environment.

OID	Name	Syntax	Access	Definition / Description
5.1.9	**RCSTActPingRowStatus**	RowStatus	$R_{HIA}C_{HIA}$	The supported RowStatus values are: - active (1) notInService (2) notReady (3) createAndGo (4) createAndWait (5) - destroy (6) When this variable is set to Active, it triggers the start of the Ping test. If all the necessary information has not been provided, this set will fail. When the Ping command has been executed, the agent will set this variable to notInService. For more information about this textual convention, one shall refer to [10].
6	**RCSTActRebootStatus TrapFlag**	INTEGER	$R_{HIA}W_{HIA}$	Identifies whether the RCST is expected to send the SNMP Enterprise Trap (RCST Reset/Reboot Status) after resetting/rebooting. Set ("1" – 'send_ trap') by the SMS prior to issuing the Reset/Reboot command if it wishes to receive a reset/reboot completion trap. (Note to RCST designer: this flag needs to be in non-volatile memory.) Cleared ("2" – 'no_trap') by the RCST after sending the Trap. This is done so that if the RCST resets/reboots due to reasons other than the command from the SMS, a trap will not be sent there.
7	**RCSTActRebootCommand**	INTEGER Default = 2	$R_{HIA}W_{HIA}$	This variable shall force the RCST to reboot when set to 1 – 'reboot'. Default value shall be 2 – 'idle'.
8	**RCSTActDownloadCommand**	INTEGER Default = 2	$R_{HIA}W_{HIA}$	This variable shall initiate a RCST download process when set to 1 – 'download'. Default value shall be 2 – 'idle'.
9	**RCSTActDropSACommand**	INTEGER Default = 2	$R_{HIA}W_{HIA}$	This variable shall force the RCST to drop its SA (in order to support IPSec failure, for example) when set to 1 – 'drop_SA'. Default value shall be 2 – 'idle'.
10	**RCSTActBootImage**	Integer32	$R_{HIA}W_{HIA}$	This variable shall trigger a boot from the current image if set to 1 or from the alternate image if set to 2. No action shall be taken in case another value would be assigned to this SNMP object. Note that when/before booting from the alternate image it is not required to check if the alternate image is valid via an SNMP Set message.
11	**RCSTActValidateCurrentImageCommand**	INTEGER Default = 2	$R_{HIA}W_{HIA}$	This variable shall force the RCST to validate the current image when set to 1 – 'validate_current_image', i.e. the one that it is currently running. When not in used, it shall be equal to 2 – 'idle' – the default value. This command shall be available to the Installer, the Super User as well as the NCC.
12	**RCSTActTrafficReleaseTimeout**	Integer32	$R_{HIA}W_{HIA}$	Time, specified in milliseconds, after which the RCST shall released its Traffic VCC due to no traffic from Hosts bound for the Return Link, while one or more entries remain in the Host Authentication table.

OID	Name	Syntax	Access	Definition / Description
13	**RCSTActDefaultDomainName**	DisplayString Size(0..255)	$R_{HIA}W_{HIA}$	Default domain name. The Super User can enable/disable this mandatory feature and set the default @domain name. The default @domain name will be permanently stored in the RCST. The RCST will append the default @domain name to all unqualified logins. Some explanations about this object can be found in the *installation* subgroup section of this MIB specification.
14	**RCSTActDefaultDomainStatus**	INTEGER enabled(1), disabled(2)	$R_{HIA}W_{HIA}$	Default domain name feature enabled (1) or disabled (2) by the Super User.

F.4.6 Applications group

Not mentioned in the presented structure because no SNMP objects have been defined so far to be set in this group.

F.5 MIB-II

F.5.1 Supported MIB–II Groups

The following MIB-2 Groups are applicable to the management of the RCST and shall be supported (if possible) by the RCST. The supported Objects are specified in the following clauses:

system Group

interfaces Group

ip Group

icmp Group

tcp Group

udp Group

transmission.dot3 Group

snmp Group

F.5.2 Objects Not Supported

The MIB-I and MIB-II specifications dictate that for an implementation to claim support for a group, it must support all of the objects in a group. It is certainly permissible for an agent to support only some of the objects in a group, but in that case the vendor cannot claim that the group is supported.

The correct way to handle this Situation is for the agent to return the error code *noSuchName* and for the vendor to admit that this particular group is not supported.

F.5.3 MIB-II Groups Specifications

Objects of the following MIB-2 groups shall be supported as specified.

ID and *Name* identify objects contained in the definition tables. The *ID* specifies the object location in the MIB hierarchy under the group branch; e.g. object *sysContact* in group system has ID 4, and this corresponds to the last digit in the OID string 1.3.6.1.2.1.1.4 specifying the object location.

Individual objects can be accessed in one of the following ways, as defined in the tables' Access column: Read Only (R); Read-Write (RW); Write Only (W); or Not Accessible (N). Write-access in this context is understood to mean from an SNMP Manager or Agent.

Objects that are Write-accessible are identified as Write-accessible by the Super User ($W_{L,}$ which may include the RCST installer), by the NCC ($W_{H,}$ which may either be the SMS or the TM), or any combination of the above, e.g. Installer and NCC (W_{LH}).

F.5.3.1 system Group

All objects of the MIB-2 system group shall be supported by the RCST. These objects contain basic system information and are described in Table F.20.

Table F.20: MIB-2 *system* Group Objects

OID	Name	Syntax	Access	Description / Definition
1	**sysDescr**	DisplayString Size(0..255)	R	A text description, which should include some very generic information about the device: type of system (RCST), manufacturer, and date.
2	**sysObjectID**	Object Identifier	R	An authoritative identifier assigned to this product by its vendor.
3	**sysUpTime**	TimeTicks	R	The time (in 1/100th of a second) since the network management portion of the system was last re-initialized.
4	**sysContact**	DisplayString Size(0..255)	RW$_{IS}$	A person responsible for the node, along with information such as phone number. Defined at installation. In the scope of the RCS system, this object shall provide all necessary information about the local RCST administrator – Super User.
5	**sysName**	DisplayString Size(0..255)	RW$_{IS}$	An administratively assigned name (usually the TCP/IP domain name). Defined at installation.
6	**sysLocation**	DisplayString Size(0..255)	RW$_{IS}$	The physical location of the IDU, including street address and GPS co-ordinates expressed as Cartesian co-ordinates 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 metre level.) The Installation Team shall define it – floating-point notation. The RCSTSysInstallLocation object of the RCST MIB provides physical location of the ODU antenna. Format: x <space> y <space> z. Values in metre.
7	**sysServices**	Integer(0..127)	R	A coded number that indicates the layer(s) for which this node performs services. Since the RCST supports the physical, data link, network, transport and application layer of the OSI model, this coded number is 79.
8	**sysORLastChange**	TimeTicks	R	The value of sysUpTime at the time of the most recent change in state or value of any instance of sysORID.
9	**sysORTable**	Sequence of sysOREntry	N	Table of dynamically configurable object resources in a SNMPv2 entity acting in an agent role.
9.1	**sysOREntry**	Sequence	N	Information on a particular dynamically configurable object resource. The RCST will report a list of entries describing the MIB branches it supports.
9.1.1	**sysORIndex**	Integer	R	Integer used as index into sysORTable.
9.1.2	**sysORID**	Object Identifier	R	An authoritative identification of a capability statement with respect to various MIB modules supported by the local SNMPv2 entity acting in an agent role.
9.1.3	**sysORDescr**	DisplayString Size(0..255)	R	A textual description of the capabilities identified by the corresponding instance of sysORID.
9.1.4	**sysORUpTime**	TimeTicks	R	The value of sysUpTime at the time this conceptual row was last instantiated.

F.5.3.2 interfaces Group

All objects in the MIB-2 interfaces group shall be supported by the RCST. These objects contain configuration, status and performance data for interfaces and are described in Table F.21.

The interface types supported are Ethernet interface, Return Link interface and Forward Link interface.

Table F.21: MIB-2 *interfaces* Group Objects

OID	Name	Syntax	Access	Description / Definition
1	**ifNumber**	Integer	R	Total number of network interfaces in the system. The supported value is 3.
2	**ifTable**	Sequence of ifEntry	N	List of interface entries. The number of entries is specified by ifNumber.
2.1	**ifEntry**	Sequence	N	An interface entry containing objects at the sub-network layer and below for a particular interface.
2.1.1	**ifIndex**	Integer	R	A unique value for each interface. The supported values are 1 for Ethernet; 2 for Return Link and 3 for Forward Link.
2.1.2	**ifDescr**	DisplayString Size(0..255)	R	A unique name for each interface. The supported values are "Ethernet"; "Return Link", or "Forward Link".
2.1.3	**ifType**	Integer	R	The type of interface. Supported values: 7 for Ethernet, TBD for Return Link and TBD for Forward Link.
2.1.4	**ifMtu**	Integer	R	The size (in octets) of the largest protocol data unit that can be sent or received on the interface. Supported values: 1500 for Ethernet. Others TBD.
2.1.5	**ifSpeed**	Gauge	R	Estimate of current bandwidth in bits per second.
2.1.6	**ifPhysAddress**	PhysAddress	R	The interface address at the protocol layer below the internetwork. Supported values: the RCST Ethernet MAC address for Ethernet, and the RCST MAC address for the Forward and Return Link.

2.1.7	**ifAdminStatus**	Integer:
		Up (1);
		Down (2);
		Testing (3)
			RW_{IS}	The desired state of the interface.
				Supported value: always up.

2.1.8	**ifOperStatus**	Integer:
		Up (1);
		Down (2);
		Testing (3)
			R	The current operational state of the interface.

2.1.9	**ifLastChange**	TimeTicks (1/100th of seconds)	R	The value of sysUpTime when the interface entered its current operational state. If the operational state of the interface has not changed since power up, the value will be 0.
2.1.10	**ifInOctets**	Counter	R	The total number of octets received on the interface, including framing octets. This counter is not valid for the Return Link and a value of 0 will be returned for the return link.
2.1.11	**ifInUcastPkts**	Counter	R	The number of subnetwork unicast packets delivered to a higher layer protocol. This counter is not valid for the Return Link and a value of 0 will be returned for the return link.
2.1.12	**ifInNUcastPkts**	Counter	R	The number of non-unicast (i.e. broadcast or multicast) packets delivered to a higher layer protocol. For the Ethernet I/F, this counter represents the number of ARP requests received from the host. This counter is not valid for the Return Link and a value of 0 will be returned for the return link.
2.1.13	**ifInDiscards**	Counter	R	The number of inbound packets discarded although no errors were found. This is due to a lack of buffer memory. This counter is not valid for the Return Link and a value of 0 will be returned for the return link.

OID	Name	Syntax	Access	Description / Definition
2.1.14	**ifInErrors**	Counter	R	The number of inbound packets discarded because they contain errors. This counter is not valid for the Return Link and a value of 0 will be returned for the return link.
2.1.15	**ifUnknownProtos**	Counter	R	The number of inbound packets discarded because of an unknown or unsupported protocol. This counter is not valid for the Return Link and a value of 0 will be returned for the return link.
2.1.16	**ifOutOctets**	Counter	R	The total number of octets transmitted out of the interface including framing octets. This counter is not valid for the Forward Link and a value of 0 will be returned for the forward link.
2.1.17	**ifOutUcastPkts**	Counter	R	The total number of unicast packet whose transmission to a single address was requested. This counter is not valid for the Forward Link and a value of 0 will be returned for the forward link.
2.1.18	**ifOutNUcastPkts**	Counter	R	The total number of packets whose transmission to a multicast or broadcast address was requested. For the Ethernet I/F this counter represents the number of multicast packet sent to Hosts. This counter is not valid for the Forward Link and a value of 0 will be returned for the forward link.
2.1.19	**ifOutDiscards**	Counter	R	The number of outbound packets that were free of errors but discarded. (i.e. packets that were filtered out, e.g. to free up memory) This counter is not valid for the Forward Link and a value of 0 will be returned for the forward link.
2.1.20	**ifOutErrors**	Counter	R	The number of outbound packets discarded because of errors. This counter is not valid for the Forward Link and a value of 0 will be returned for the forward link.
2.1.21	**ifOutQLen**	Gauge	R	Number of packets in the outbound queue. This counter is not valid for the Forward Link and a value of 0 will be returned for the forward link.
2.1.22	**ifSpecific**	Object Identifier	R	The identifier for a MIB that contains additional definitions relating to this interface type. Supported value: 0.0.0.0.

F.5.3.3 *ip* Group

Objects in the MIB-2 ip group that pertain to configuration and statistics of IP datagrams shall be supported by the RCST and are described in Table F.22.

Objects in this table not applicable to the RCST will have a value of 0.

Table F.22: MIB-2 *ip* Group Objects

OID	Name	Syntax	Access	Description / Definition

1	**ipForwarding**	Integer:
		Forwarding (1);
		Not forwarding (2)
		Default = (1)
			R	Indicates whether the system will route datagrams.
				Always set to Forwarding.

2	**ipDefaultTTL**	Integer Default = (0)	RW_{IS}	This variable is not applicable to the RCST. The only value supported is 0.
3	**ipInReceives**	Counter	R	Total number of incoming datagrams.
4	**ipInHdrErrors**	Counter	R	Input datagrams discarded due to header errors.
5	**ipInAddrErrors**	Counter	R	Input datagrams discarded because the destination IP address was not valid.
6	**ipForwDatagrams**	Counter	R	Number of incoming datagrams for which forwarding was attempted.
7	**ipInUnknownProtos**	Counter	R	Datagrams addressed to this system whose protocol was unknown or unsupported.
8	**ipInDiscards**	Counter	R	Correct datagrams that were discarded anyway, possibly because of lack of buffer memory.
9	**ipInDelivers**	Counter	R	The number of IP datagrams delivered to local protocols.
10	**ipOutRequests**	Counter	R	Total number of datagrams originating locally.
11	**ipOutDiscards**	Counter	R	Output datagrams discarded although there was no error.
12	**ipOutNoRoutes**	Counter	R	Output datagrams discarded because no route can be found, i.e. NCC is down, link has failed, etc.
13	**ipReasmTimeout**	Integer	R	The maximum number of seconds that received fragments are held for re-assembly. This variable is not applicable to the RCST. The only value supported is 0.
14	**IpReasmReqds**	Counter	R	The number of IP fragments received which needed to be reassembled. This variable is not applicable to the RCST. The only value supported is 0.
15	**IpReasmOKs**	Counter	R	The number of IP datagrams successfully reassembled. This variable is not applicable to the RCST. The only value supported is 0.
16	**IpReasmFails**	Counter	R	The number of times that re-assembly failed. This variable is not applicable to the RCST. The only value supported is 0.
17	**ipFragOKs**	Counter	R	Number of successfully fragmented datagrams. This variable is not applicable to the RCST. The only value supported is 0.
18	**ipFragFails**	Counter	R	Number of IP datagrams discarded because they needed to be fragmented but could not be. This variable is not applicable to the RCST. The only value supported is 0.
19	**ipFragCreates**	Counter	R	Number of IP datagram fragments created. This variable is not applicable to the RCST. The only value supported is 0.
	Notice:			RCST does not have any routing table, only one ATM and one Ethernet interface, therefore, all the following objects in this group are not applicable. A value of 0 will be returned if these variables are queried. They are coloured grey to indicate that they are not applicable.

OID	Name	Syntax	Access	Description / Definition
20	**IpAddrTable**	Sequence of ipAddrEntry	Not accessible	A list of IP entries. The number of entries is given by the ifNumber.
	IpAddrEntry	ipAddrEntry	Not accessible	Each IP address is used as an entry.
	IpAdEntAddr (Index)	IpAddress	R	The 32-bit IP address for this entry.
	IpAdEntIfIndex	Integer	R	Same as ifIndex, 0 for Ethernet, 1 for ATM.
	IpAdEntNetMask	IpAddress	R	The subnet mask for this IP address.
	IpAdEntBcastAddr	Integer	R	Indicate the broadcast format for the interface. 1 for the all 1's standard, 0 for the all 0's format.
	IpAdEntReasmMaxSize	Integer (0..65535)	R	The biggest datagram that can be reassembled from fragments received at this interface.
21	**IpRouteTable**		NA	This table is not supported by the RCST. Any of its parameters shall send a 'noSuchName' response to any query.
	IpRouteEntry		NA	
	IpRouteDest (Index)	IpAddress	RW$_{IS}$	A destination IP address. 0.0.0.0 is the default value.NOT SUPPORTED – 'noSuchName' when queried.
	IpRouteIfIndex	Integer	RW$_{IS}$	Same as ifIndex. 0 for Ethernet, 1 for ATM. NOT SUPPORTED – 'noSuchName' when queried.
	IpRouteMetric1	Integer	RW$_{IS}$	The primary routing metric for this route. Its meaning depends on the routing protocol. NOT SUPPORTED – 'noSuchName' when queried.
	IpRouteMetric2	Integer	RW$_{IS}$	Another routing metric. NOT SUPPORTED – 'noSuchName' when queried.
	IpRouteMetric3	Integer	RW$_{IS}$	Another routing metric. NOT SUPPORTED – 'noSuchName' when queried.
	IpRouteMetric4	Integer	RW$_{IS}$	Another routing metric. NOT SUPPORTED – 'noSuchName' when queried.
	ipRouteNextHop	IpAddress	RW$_{IS}$	The IP address of the next hop of this route. NOT SUPPORTED – 'noSuchName' when queried.
	ipRouteType	Integer Other(1), invalid(2), direct(3), indirect(4)	RW$_{IS}$	The status or type of the route. NOT SUPPORTED – 'noSuchName' when queried.
	ipRouteProto	Integer	R	The protocol by which the route was learned. NOT SUPPORTED – 'noSuchName' when queried.
	ipRouteAge	Integer		Number of seconds since last update or validation of this route. NOT SUPPORTED – 'noSuchName' when queried.
	IpRouteMask	Integer		Routing mask for the entry. NOT SUPPORTED – 'noSuchName' when queried.
	IpRouteMetric5	Integer		Yet another route metric. NOT SUPPORTED – 'noSuchName' when queried.
	ipRouteInfo	Object Identifier		A pointer to more MIB variables for the protocol. If no specified, the value should be {0 0}. NOT SUPPORTED – 'noSuchName' when queried.

OID	Name	Syntax	Access	Description / Definition
22	**IpNetToMediaTable**			
	IpNetToMediaEntry			
	IpNetToMediaIFIndex (Index)	Integer	RW_{IS}	Same as the ifIndex.
	IpNetToMediaPhyAddress	PhysAddress	RW_{IS}	Physical Address of the interface.
	IpNetToMediaNetAddress (Index)	IpAddress	RW_{IS}	The IP address.
	IpNetToMediatype	Integer other(1), invalid(2), dynamic(3), static(4)	RW_{IS}	How the entry was learned.
23	**ipRoutingDiscards**	Counter	R	The number of valid routing entries that were discarded.

F.5.3.4 *icmp* Group

All objects in the MIB-2 *icmp* group shall be supported by the RCST. These objects contain ICMP statistics data and are described in Table F.23.

Table F.23: MIB-2 *icmp* Group Objects

OID	Name	Syntax	Access	Definition
1	**icmpInMsgs**	Counter	R	The total number of ICMP messages which the entity received.
2	**icmpInErros**	Counter	R	The number of ICMP messages which the entity received but determined as having ICMP-specific errors (bad ICMP checksums, bad length, etc.).
3	**icmpInDestUnreachs**	Counter	R	The number of ICMP Destination Unreachable messages received.
4	**icmpInTimeExcds**	Counter	R	The number of ICMP Time Exceeded messages received.
5	**icmpInParmProbs**	Counter	R	The number of ICMP Parameter Problem messages received.
6	**icmpInSrcQuenchs**	Counter	R	The number of ICMP Source Quench messages received.
7	**icmpInRedirects**	Counter	R	The number of ICMP Redirect messages received.
8	**icmpInEchos**	Counter	R	The number of ICMP Echo (request) messages received.
9	**icmpInEchoReps**	Counter	R	The number of ICMP Echo Reply messages received.
10	**icmpInTimestamps**	Counter	R	The number of ICMP Timestamp (request) messages received.
11	**icmpInTimestampReps**	Counter	R	The number of ICMP Timestamp Reply messages received.
12	**icmpInAddrMasks**	Counter	R	The number of ICMP Address Mask Request messages received.
13	**icmpInAddrMaskReps**	Counter	R	The number of ICMP Address Mask Reply messages received.
14	**icmpOutMsgs**	Counter	R	The total number of ICMP messages which this entity attempted to send.
15	**icmpOutErrors**	Counter	R	This value should not include errors discovered outside the ICMP layer such as the inability of IP to route the resultant datagram.
16	**icmpOutDestUnreachs**	Counter	R	The number of ICMP Destination Unreachable messages sent.
17	**icmpOutTimeExcds**	Counter	R	The number of ICMP Time Exceeded messages sent.
18	**icmpOutParmProbs**	Counter	R	The number of ICMP Parameter Problem messages sent.
19	**icmpOutSrcQuenchs**	Counter	R	The number of ICMP Source Quench messages sent.
20	**icmpOutRedirects**	Counter	R	The number of ICMP Redirect messages sent.

OID	Name	Syntax	Access	Definition
21	**icmpOutEchos**	Counter	R	The number of ICMP Echo (request) messages sent.
22	**icmpOutEchoReps**	Counter	R	The number of ICMP Echo Reply messages sent.
23	**icmpOutTimestamps**	Counter	R	The number of ICMP Timestamp (request) messages sent.
24	**icmpOutTimestampReps**	Counter	R	The number of ICMP Timestamp Reply messages sent.
25	**icmpOutAddrMasks**	Counter	R	The number of ICMP Address Mask Request messages sent.
26	**icmpOutAddrMaskReps**	Counter	R	The number of ICMP Address Mask Reply messages sent.

F.5.3.5 *tcp* Group

All objects in the MIB-2 *tcp* group, except for the TCP Connection Table objects, shall be supported by the RCST. These objects contain configuration and statistics data of TCP connections and are described in Table F.24.

Table F.24: MIB-2 *tcp* Group Objects

OID	Name	Syntax	Access	Description / Definition

1	**tcpRtoAlgorithm**	Integer:
		Other (1);
		Constant (2);
		Rsre (3);
		Vanj (4)
			R	The algorithm used to compute the retransmission timeout.

2	**tcpRtoMin**	Integer	R	Minimum lower bound in milliseconds on the retransmission timeout.
3	**tcpRtoMax**	Integer	R	Maximum upper bound in milliseconds allowed for a retransmission timeout.
4	**tcpMaxConn**	Integer	R	A limit on the maximum number of concurrent TCP connections. -1 means that it is dynamically determined.
5	**tcpActiveOpens**	Counter	R	The number of outgoing connections requests from this system. Default is 0.
6	**tcpPassiveOpens**	Counter	R	The number of incoming connection requests to this system.
7	**tcpAttemptFails**	Counter	R	The number of failed connection requests – incoming and outgoing.
8	**tcpEstabResets**	Counter	R	The number of established or gracefully closing connections that have been terminated abruptly.
9	**tcpCurrEstab**	Gauge	R	The number of TCP connections that are in either ESTABLISHED or CLOSE-WAIT state.
10	**tcpInSegs**	Counter	R	Total number of segments received, including those received in error.
11	**tcpOutSegs**	Counter	R	Total number of segments sent, excluding those containing only retransmitted octets.
12	**tcpRetransSegs**	Counter	R	Total number of segments containing retransmitted data.
14	**tcpInErrs**	Counter	R	Total number of segments received with errors.
15	**tcpOutRsts**	Counter	R	Total number of TCP segments sent with the reset (RST) flag set to 1.
	Notice:			RCST does not have TCP connection table as such. The following objects in this group are not applicable to the RCST.
16	**tcpConnTable**			This table is not supported by the RCST. Any of its parameters shall send a 'noSuchName' response to any query.
	tcpConnEntry			
	tcpConnState	Integer	RW_{IS}	The current state for the connection. NOT SUPPORTED – 'noSuchName' when queried.
	TcpConnLocalAddress (Index)	IpAddress	R	The local IP address for this TCP connection. NOT SUPPORTED – 'noSuchName' when queried.

OID	Name	Syntax	Access	Description / Definition
	TcpConnLocalPort (Index)	Integer (0..65535)	R	The local port number for this TCP connection. NOT SUPPORTED – 'noSuchName' when queried.
	TcpConnRemAddress (Index)	IpAddress	R	The remote IP address for this TCP connection. NOT SUPPORTED – 'noSuchName' when queried.
	TcpConnRemPort (Index)	Integer (0..65535)	R	The remote port number for this TCP connection. NOT SUPPORTED – 'noSuchName' when queried.

F.5.3.6 *udp* Group

All objects in the MIB-2 *udp* group shall be supported by the RCST. These objects contain statistics of active UDP services and are described in Table F.25.

Table F.25: MIB-2 *udp* Group Objects

Id	Name	Syntax	Access	Definition
1	**udpInDatagrams**	Counter	R	Total number of UDP datagrams delivered to UDP applications.
2	**udpNoPorts**	Counter	R	Total number of received UDP datagrams for which there was no application at the destination port.
3	**udpInErrors**	Counter	R	Number of UDP datagrams that could not be delivered for some other reason.
4	**udpOutDatagrams**	Counter	R	Total number of outbound UDP datagrams.
5	**udpTable**	Sequence of udpEntry	N	UDP service table.
5.1	**udpEntry**	Sequence	N	Contains the list of variables that make up the table.
5.1.1	**udpLocalAddress**	IpAddress	R	The local IP address for this UDP listener.
5.1.2	**udpLocalPort**	Integer (0..65535)	R	The local port number for this UDP listener. The only UDP service running on the RCST is SNMP, which uses port 161.

F.5.3.7 transmission Group

This group contains objects that provide details about the underlying transmission medium for each interface. In fact, the *transmission* group is itself simply a node in the MIB-2 hierarchy under which various interface-specific groups are located. The RCST shall support the Ethernet interface *dot3* group.

F.5.3.8 *dot3* group

The Statistics Table objects for the Ethernet interface *dot3* group shall be supported by the RCST. These objects contain statistics for the Ethernet interface and are described in Table F.26.

Table F.26: MIB-2 *dot3* Group Objects

OID	Name	Syntax	Access	Description / Definition
2	**dot3StatsTable**	Sequence of dot3Stats Entry	N	Statistics for a collection of Ethernet-like interfaces attached to a particular system.
2.1	**dot3StatsEntry**	Sequence	N	Statistics for a particular interface to an Ethernet-like medium.
2.1.1	**dot3StatsIndex**	Integer	R	An index value that uniquely identifies an interface to an Ethernet-like medium. Same as ifIndex.
2.1.2	**dot3StatsAlignment Errors**	Counter	R	Number of frames with an alignment error, i.e. the length is not an integral number of octets and the frame cannot pass the FCS test.
2.1.3	**dot3StatsFCSErrors**	Counter	R	Number of frames with frame-check errors. i.e. there is an integral number of octets, but an incorrect FCS.
2.1.4	**dot3StatsSingle CollisionFrames**	Counter	R	Number of successfully transmitted frames for which there was exactly one collision.
2.1.5	**dot3StatsMultiple CollisionFrames**	Counter	R	Number of successfully transmitted frames for which there were multiple collisions.
2.1.6	**dot3StatsSQETest Errors**	Counter	R	Number of times that the Signal Quality Error Test Error message was generated by the interface.
2.1.7	**dot3StatsDeferred Transmissions**	Counter	R	Number of times the first transmission attempt was delayed because the medium was busy.
2.1.8	**dot3StatsLate Collisions**	Counter	R	Number of times that a collision was detected later than 64 octets into the transmission.
2.1.9	**dot3StatsExcessive Collisions**	Counter	R	Number of frames for which transmission failed because of excessive collisions.
2.1.10	**dot3StatsInternal MacTransmitErrors**	Counter	R	Number of frames for which transmission failed because of an internal MAC layer transmit error.
2.1.11	**dot3StatsCarrier SenseErrors**	Counter	R	Number of transmission attempts that failed because the carrier sense condition was lost or never asserted.
2.1.13	**dot3StatsFrameToo Longs**	Counter	R	Number of received frames that were bigger than the maximum permitted size.
2.1.16	**dot3StatsInternal MacReceiveErrors**	Counter	R	Number of frames for which reception failed because of an internal MAC layer receive error.

F.5.3.9 *snmp* Group

All objects of the MIB-2 *snmp* group shall be supported by the RCST. These objects contain configuration and statistics data of SNMP and are described in Table F.27.

Note that RFC 1907 [34] obsoletes SNMP group objects 7 to 27. But because we are not supporting the new objects, which replace the obsolete objects, we will continue to support all the objects in this group as specified in RFC 1213 [35].

Table F.27: MIB-2 *snmp* Group Objects

OID	Name	Syntax	Access	Description / Definition
1	***snmpInPkts***	Counter	R	Total number of incoming SNMP messages delivered by the transport service.
2	snmpOutPkts	Counter	R	Total number of outgoing SNMP messages passed to the transport service.
3	snmpInBadVersions	Counter	R	Number of incoming messages with an unsupported version.
4	snmpInBadCommunityNames	Counter	R	Number of incoming messages with an unknown community name.
5	snmpInBadCommunityUses	Counter	R	Number of incoming messages requesting an operation not supported for the community name.
6	snmpInASNParseErrs	Counter	R	Number of times message decoding failed.
8	snmpInTooBigs	Counter	R	Number of incoming messages with an error-status field of "too big".
9	snmpInNoSuchNames	Counter	R	Number of incoming messages with an error-status field of "noSuchName".
10	snmpInBadValues	Counter	R	Number of incoming messages with an error-status field of "badValue".
11	snmpInReadOnlys	Counter	R	Number of incoming messages with an error-status field of "readOnly".
12	snmpInGenErrs	Counter	R	Number of incoming messages with an error-status field of "genErr".
13	snmpInTotalReqVars	Counter	R	Total number of local MIB objects that have been retrieved successfully as a result of incoming get-requests and get-next-requests.
14	snmpInTotalSetVars	Counter	R	Total number of local MIB objects that have been updated successfully as a result of incoming set-requests.
15	snmpInGetRequests	Counter	R	Number of incoming get-requests accepted and processed.
16	snmpInGetNexts	Counter	R	Number of incoming get-next requests accepted and processed
17	snmpInSetRequests	Counter	R	Number of incoming set-requests accepted and processed.
18	snmpInGetResponses	Counter	R	Number of incoming get-responses accepted and processed.
19	snmpInTraps	Counter	R	Number of incoming traps accepted and processed.
20	snmpOutTooBigs	Counter	R	Number of outgoing messages sent with the error-status field set to "tooBig".
21	snmpOutNoSuchNames	Counter	R	Number of outgoing messages sent with the error-status field set to "noSuchName".
22	snmpOutBadValues	Counter	R	Number of outgoing messages sent with the error-status field set to "badValue".
24	snmpOutGenErrs	Counter	R	Number of outgoing messages sent with the error-status field set to "genErr".
25	snmpOutGetRequests	Counter	R	Number of outgoing get-requests generated.
26	snmpOutGetNexts	Counter	R	Number of outgoing get-next-requests generated.
27	snmpOutSetRequests	Counter	R	Number of outgoing set-requests generated.
28	snmpOutGetResponses	Counter	R	Number of outgoing get-responses generated.
29	snmpOutTraps	Counter	R	Number of outgoing traps generated.
30	snmpEnableAuthen Traps	Integer: enabled (1); disabled (2)	$\mathrm{RW_{IS}}$	Indicates whether the agent is allowed to generate authentication-failure traps.

OID	Name	Syntax	Access	Description / Definition
31	snmpSilentDrops	Counter	R	The total number of GetRequest-PDUs, GetNextRequest-PDUs, GetBulkRequest-PDUs, GetRequest-PDUs, and InformRequest-PDUs delivered to the SNMP entity, which were silently dropped because the size of a reply containing an alternate Response-PDU with an empty variable-bindings field was greater than either a local constraint or the maximum message size associated with the originator of the request.
32	snmpProxyDrops	Counter	R	The total number of GetRequest-PDUs, GetNextRequest-PDUs, GetBulkRequest-PDUs, SetRequest-PDUs, and InformRequest-PDUs delivered to the SNMP entity which were silently dropped because the transmission of the (possibly translated) message to a proxy target failed in a manner (other than a time-out) such that no Response-PDU could be returned.

F.6 SNMP Response Code

The SNMPv2 response codes listed in Table F.28 shall be supported. This table gives the use of Error Codes in Response-PDU.

Table F.28: Response code support

	GetRequest, GetNextRequest, GetBulkRequest	SetRequest	InformRequest
noError (0)	X	X	X
tooBig (1)	X	X	X
noSuchName (2)			
badValue (3)			
readOnly (4)			
genError (5)	X	X	X
noAccess (6)		X	
wrongType (7)		X	
wrongLength (8)		X	
wrongEncoding (9)		X	
wrongValue (10)		X	
noCreation (11)		X	
inconsistentValue (12)		X	
resourceUnavailable (13)		X	
commitFailed (14)		X	
undoFailed (15)		X	
authorizationError (16)	X	X	X
notWritable (17)		X	
inconstentName (18)		X	

F.7 ASN.1 MIB definition

The ASN.1 MIB definition is contained in archive tr_101790v010101p0.zip which accompanies the present document.

```
DVB-RCST-MIB DEFINITIONS ::= BEGIN
  IMPORTS
    DisplayString, PhysAddress
      FROM RFC1213-MIB
    OBJECT-GROUP, MODULE-COMPLIANCE, NOTIFICATION-GROUP
      FROM SNMPv2-CONF
    enterprises, IpAddress, Integer32, OBJECT-TYPE,
    MODULE-IDENTITY, NOTIFICATION-TYPE
      FROM SNMPv2-SMI
    RowStatus, MacAddress, TimeStamp
      FROM SNMPv2-TC;


  RCSTMIBMODULE-IDENTITY
    LAST-UPDATED  "200102141600Z"
    ORGANIZATION
      "DVB-RCS group of the DVB Project"
    CONTACT-INFO
      " --- "
    DESCRIPTION
      "Description."
    ::= { RCS 1 }


--
-- Node definitions
--

  dvb OBJECT IDENTIFIER ::= { enterprises 2696 }

  RCS OBJECT IDENTIFIER ::= { dvb 2 }

  RCSTMIBObjects OBJECT IDENTIFIER ::= { RCSTMIB 1 }

  RCSTSystem OBJECT IDENTIFIER ::= { RCSTMIBObjects 1 }

  installation OBJECT IDENTIFIER ::= { RCSTSystem 1 }

RCSTSysInstallOwner OBJECT-TYPE
  SYNTAX DisplayString (SIZE (0..255))
  MAX-ACCESS read-write
  STATUS current
  DESCRIPTION
    "Identifies the technical staff that performed the RCST installation and
commission of both ODU and IDU. It also provides contact information and is
defined locally by RCST Administrator or Installation Team. It shall only be
modifiable by the installer."
  ::= { installation 1 }


RCSTSysInstallDate OBJECT-TYPE
  SYNTAX DisplayString (SIZE (0..255))
  MAX-ACCESS read-write
  STATUS current
  DESCRIPTION
```

```
     "Identifies the installation date, Work Order number, and other relevant
installation information. It shall be defined locally by RCST Administrator or
Installation Team and shall only be modifiable by the Maintenance
(Update)/Installation Team."
   ::= { installation 2 }


RCSTSysInstallLocation OBJECT-TYPE
   SYNTAX DisplayString (SIZE (0..255))
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
     "Physical location of the ODU antenna expressed as Cartesian co-ordinates
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 metre level.)
     The Installation Team shall assign the location value to this variable.
Floating-point notation shall be used.
     Format: x <space> y <space> z. Values in metre.
     The system.sysLocation object of MIB-II provides physical location of the
IDU unit."
   ::= { installation 3 }


RCSTSysInstallSSPId OBJECT-TYPE
   SYNTAX DisplayString (SIZE(0..255))
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
     "Satellite Service Provider (SSP) Identifier."
   ::= { installation 4 }


RCSTSysInstallRCSTMacAddr OBJECT-TYPE
   SYNTAX MacAddress
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
     "RCST MAC Address of the Air Interface."
   ::= { installation 5 }


RCSTSysInstallOduAntennaSize OBJECT-TYPE
   SYNTAX Integer32
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
     "This object gives the diameter of the antenna. This value shall be given
in centimetre. Defined at installation."
   ::= { installation 6 }


RCSTSysInstallOduSSPA OBJECT-TYPE
   SYNTAX Integer32
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
     "This field describes the SSPA installed in the ODU and shall be defined by
the installer. The power shall be given in tenth of Watt for more flexibility,
i.e. 0.5W will be represented by 5, 1W by 10 and 2W by 20. Defined at
installation."
   ::= { installation 7 }
```

```
RCSTSysInstallAllowedDomainNextIndex OBJECT-TYPE
  SYNTAX Integer32
  MAX-ACCESS read-only
  STATUS current
  DESCRIPTION
    "Provides the next available index in the dynamic table
RCSTSysInstallAllowedDomainTable. The row-creation algorithm shall use this
value."
  ::= { installation 8 }


RCSTSysInstallAllowedDomainTable OBJECT-TYPE
  SYNTAX SEQUENCE OF RCSTSysInstallAllowedDomainEntry
  MAX-ACCESS not-accessible
  STATUS current
  DESCRIPTION
    "This feature gives the possibility for the RCST to reject logins to any
domain not specified as allowable in the RCST without sending a user login
request to the NCC.
    The installer is the only entity that can enable/disable this functionality
and enter/modify the allowed @domain names. The allowed @domain names will be
permanently stored in the RCST."
  ::= { installation 9 }


RCSTSysInstallAllowedDomainEntry OBJECT-TYPE
  SYNTAX RCSTSysInstallAllowedDomainEntry
  MAX-ACCESS not-accessible
  STATUS current
  DESCRIPTION
    "This represents one entry in the RCSTSysInstallAllowedDomainTable table.
Each row defines an allowed domain."
  INDEX { RCSTSysInstallAllowedDomainIndex }
  ::= { RCSTSysInstallAllowedDomainTable 1 }


RCSTSysInstallAllowedDomainEntry ::=
  SEQUENCE {
    RCSTSysInstallAllowedDomainIndex
      Integer32,
    RCSTSysInstallAllowedDomainName
      DisplayString,
    RCSTSysInstallAllowedDomainStatus
      RowStatus
    }


RCSTSysInstallAllowedDomainIndex OBJECT-TYPE
  SYNTAX Integer32
  MAX-ACCESS not-accessible
  STATUS current
  DESCRIPTION
    "Uniquely identifies each authorized domain name."
  ::= { RCSTSysInstallAllowedDomainEntry 1 }


RCSTSysInstallAllowedDomainName OBJECT-TYPE
  SYNTAX DisplayString (SIZE (0..255))
  MAX-ACCESS read-create
  STATUS current
  DESCRIPTION
    "Allowed domain name, permanently stored in the RCST and
enabled/disabled/defined by the Installer."
```

```
  ::= { RCSTSysInstallAllowedDomainEntry 2 }


RCSTSysInstallAllowedDomainStatus OBJECT-TYPE
  SYNTAX RowStatus
  MAX-ACCESS read-create
  STATUS current
  DESCRIPTION
    "This object enables the table to be dynamic. Moreover, it provides some
information about the validity of each row. It also enables creation/deletion of
rows."
  ::= { RCSTSysInstallAllowedDomainEntry 3 }


RCSTSysInstallCompNextIndex OBJECT-TYPE
  SYNTAX Integer32
  MAX-ACCESS read-only
  STATUS current
  DESCRIPTION
    "Next index available in the table RCSTSysInstallCompDescrTable. The row
creation algorithm shall use it."
  ::= { installation 10 }


RCSTSysInstallCompDescrTable OBJECT-TYPE
  SYNTAX SEQUENCE OF RCSTSysInstallCompDescrEntry
  MAX-ACCESS not-accessible
  STATUS current
  DESCRIPTION
    "Description of the different component of the RCST. This table shall be
defined at installation and, though it is defined as a dynamic table, once
defined by the installer, it shall never increase nor decrease.
    It is the installer that will definitively fix its size. This latter may
vary from one manufacturer to the other."
  ::= { installation 11 }


RCSTSysInstallCompDescrEntry OBJECT-TYPE
  SYNTAX RCSTSysInstallCompDescrEntry
  MAX-ACCESS not-accessible
  STATUS current
  DESCRIPTION
    "This represents one entry in the RCSTSysInstallCompDescrTable table. Each
row describes a particular object."
  INDEX { RCSTSysInstallCompDescrIndex }
  ::= { RCSTSysInstallCompDescrTable 1 }


RCSTSysInstallCompDescrEntry ::=
  SEQUENCE {
    RCSTSysInstallCompDescrIndex
      Integer32,
    RCSTSysInstallCompDescrString
      DisplayString,
    RCSTSysInstallCompDescrStatus
      RowStatus
    }


RCSTSysInstallCompDescrIndex OBJECT-TYPE
  SYNTAX Integer32
  MAX-ACCESS not-accessible
  STATUS current
  DESCRIPTION
    "Uniquely identifies each row of the table."
  ::= { RCSTSysInstallCompDescrEntry 1 }


RCSTSysInstallCompDescrString OBJECT-TYPE
  SYNTAX DisplayString (SIZE (0..255))
  MAX-ACCESS read-create
  STATUS current
  DESCRIPTION
    "256-character string that gives a thorough description of each component.
It shall contain a name reference, a version (HW, SW) and the like."
  ::= { RCSTSysInstallCompDescrEntry 2 }


RCSTSysInstallCompDescrStatus OBJECT-TYPE
  SYNTAX RowStatus
  MAX-ACCESS read-create
  STATUS current
  DESCRIPTION
    "This object enables the table to be dynamic. Moreover, it provides some
information about the validity of each row. It also enables creation/deletion of
rows."
  ::= { RCSTSysInstallCompDescrEntry 3 }


idu OBJECT IDENTIFIER ::= { RCSTSystem 2 }


RCSTSysIduShutTxAGCRateThresh OBJECT-TYPE
  SYNTAX Integer32
  MAX-ACCESS read-write
  STATUS current
  DESCRIPTION
    "AGC variation rate threshold, specified in tenth of dB/sec, above which
the IDU must disable its modulator output."
  ::= { idu 1 }


RCSTSysIduShutTxAGCWindow OBJECT-TYPE
  SYNTAX Integer32
  MAX-ACCESS read-write
  STATUS current
  DESCRIPTION
    "Integration window of the AGC variation rate measurement, specified in
seconds."
  ::= { idu 2 }


RCSTSysTargetEbN0 OBJECT-TYPE
  SYNTAX Integer32
  MAX-ACCESS read-write
  STATUS current
  DESCRIPTION
    "This value describes the wanted Eb/N0 value that enables operation of the
return link with the required error performance. The values shall be given in
tenth of dB and the initial value shall be equal to 7 dB. The range shall be
from 0 to 31.5 dBm with a precision of 0.1 dB."
  DEFVAL { 70 }
  ::= { idu 3 }

RCSTSysLoopGainD OBJECT-TYPE
  SYNTAX Integer32
  MAX-ACCESS read-write
  STATUS current
  DESCRIPTION
    "This ULPC parameter will affect the dynamic behaviour of the system and
determine its properties regarding acquisition and error smoothing performance.
The value shall be given in decimal (i.e. 15 stands for 1.5).
    The initial value shall be equal to 0.5. The range shall be from 0 to 2
with a precision of 0.1. "
  DEFVAL { 5 }
  ::= { idu 4 }


RCSTSysEbN0Range OBJECT-TYPE
  SYNTAX Integer32
  MAX-ACCESS read-only
  STATUS current
  DESCRIPTION
    "This value describes the possible range of Eb/N0 variation that can be
compensated by the system. It shall be identical to the possible rain-fade in dB
and corresponds also to the uplink power dynamic range the RCST can cover.
    The values shall be given in tenth of dB and the initial value shall be
equal to 8 dB. The range shall be from 0 to 20 dB with a precision of 0.1 dB. "
  DEFVAL { 80 }
  ::= { idu 5 }


capability OBJECT IDENTIFIER ::= { RCSTSystem 3 }


RCSTSysCapabilitySecurity OBJECT-TYPE
  SYNTAX Integer32
  MAX-ACCESS read-write
  STATUS current
  DESCRIPTION
    "'1' for RCST implementing the security mechanism as described in the DVB-
RCS normative document ETSI EN 301 790, '0' otherwise."
  ::= { capability 1 }


RCSTSysCapabilitySNMP OBJECT-TYPE
  SYNTAX Integer32
  MAX-ACCESS read-write
  STATUS current
  DESCRIPTION
    "'1' for RCST supporting SNMP, '0' otherwise."
  ::= { capability 2 }


RCSTSysCapabilityATMconnectivity OBJECT-TYPE
  SYNTAX Integer32
  MAX-ACCESS read-write
  STATUS current
  DESCRIPTION
    "'1' for RCST capable of ATM connectivity, '0' for not capable."
  ::= { capability 3 }


RCSTSysCapabilityMPEG2TSTRF OBJECT-TYPE
  SYNTAX Integer32
  MAX-ACCESS read-write
  STATUS current
  DESCRIPTION
    "'1' for RCST capable of MPEG2-TS TRF, '0' for not capable."
  ::= { capability 4 }


RCSTSysCapabilityRCSTboards OBJECT-TYPE
  SYNTAX Integer32
  MAX-ACCESS read-write
  STATUS current
  DESCRIPTION
    "Number of RCST forward link receivers: '00' for 1 receiver, '01' for 2,
'10' for more than 2, '11' reserved"
  ::= { capability 5 }


RCSTSysCapabilityRCSTACQ OBJECT-TYPE
  SYNTAX Integer32
  MAX-ACCESS read-write
  STATUS current
  DESCRIPTION
    "'0' for RCST not requiring ACQ bursts, '1' for ACQ required."
  ::= { capability 6 }


RCSTSysCapabilityMultiIdu OBJECT-TYPE
  SYNTAX Integer32
  MAX-ACCESS read-write
  STATUS current
  DESCRIPTION
    "'0' for single IDU/single ODU configuration, '1' when 2 or more IDUs are
connected to a single ODU."
  ::= { capability 7 }


RCSTSysCapabilitySWVersion OBJECT-TYPE
  SYNTAX Integer32
  MAX-ACCESS read-write
  STATUS current
  DESCRIPTION
    "System dependent. Can be used to define the RCST software version. (bit 7
to bit 0)"
  ::= { capability 8 }


RCSTSysCapabilityFreqHoppRange OBJECT-TYPE
  SYNTAX Integer32
  MAX-ACCESS read-only
  STATUS current
  DESCRIPTION
    "Defines the RCST burst to burst frequency hopping range capability: '00'
for 20 MHz, other patterns system dependent. "
  ::= { capability 9 }


RCSTSysCapabilityMFTDMA OBJECT-TYPE
  SYNTAX Integer32
  MAX-ACCESS read-write
  STATUS current
  DESCRIPTION
    "'1' for RCST supporting dynamic MF-TDMA, '0' for supporting fixed MF-
TDMA."
  ::= { capability 10 }


RCSTSysCapabilityRCSTClass OBJECT-TYPE
```

```
  SYNTAX Integer32
  MAX-ACCESS read-write
  STATUS current
  DESCRIPTION
    "System dependent. (2 lsb)"
  ::= { capability 11 }


RCSTSysCapabilityRCSTMode OBJECT-TYPE
  SYNTAX Integer32
  MAX-ACCESS read-write
  STATUS current
  DESCRIPTION
    "'000' for Installation mode, '001' for Operational mode, '010' for
Reference RCST mode. Other patterns reserved."
  ::= { capability 12 }


RCSTSysCapabilityCapacityReqCRA OBJECT-TYPE
  SYNTAX Integer32
  MAX-ACCESS read-write
  STATUS current
  DESCRIPTION
    "'1' for CRA supported, '0' for CRA not supported."
  ::= { capability 13 }


RCSTSysCapabilityCapacityReqRBDC OBJECT-TYPE
  SYNTAX Integer32
  MAX-ACCESS read-write
  STATUS current
  DESCRIPTION
    "'1' for RBDC supported, '0' for RBDC not supported."
  ::= { capability 14 }


RCSTSysCapabilityCapacityReqVBDC OBJECT-TYPE
  SYNTAX Integer32
  MAX-ACCESS read-write
  STATUS current
  DESCRIPTION
    "'1' for VBDC supported, '0' for VBDC not supported."
  ::= { capability 15 }


RCSTSysCapabilityCapacityReqAVBDC OBJECT-TYPE
  SYNTAX Integer32
  MAX-ACCESS read-write
  STATUS current
  DESCRIPTION
    "'1' for AVBDC supported, '0' for AVBDC not supported."
  ::= { capability 16 }


RCSTSysCapabilityCapacityReqFCA OBJECT-TYPE
  SYNTAX Integer32
  MAX-ACCESS read-write
  STATUS current
  DESCRIPTION
    "'1' for FCA supported, '0' for FCA not supported."
  ::= { capability 17 }
```

```
RCSTSysCapabilityCapacityReqDCRA OBJECT-TYPE
  SYNTAX Integer32
  MAX-ACCESS read-write
  STATUS current
  DESCRIPTION
    "'1' for DCRA supported, '0' for DCRA not supported."
  ::= { capability 18 }


RCSTConfig OBJECT IDENTIFIER ::= { RCSTMIBObjects 2 }

network OBJECT IDENTIFIER ::= { RCSTConfig 1 }


RCSTConfigNetworkOamIpAddr OBJECT-TYPE
  SYNTAX IpAddress
  MAX-ACCESS read-write
  STATUS current
  DESCRIPTION
    "OAM IP Address of the RCST.
    This object used with both ip and interfaces MIB-II subgroups determines
uniquely the interface through which OAM traffic is passing through.
    Note that the OAM IP address may be statically or dynamically assigned.
    It is system dependent whether the OAM IP address and the Traffic IP
address are the same address."
  ::= { network 1 }


RCSTConfigNetworkOamIpNetworkMask OBJECT-TYPE
  SYNTAX IpAddress
  MAX-ACCESS read-write
  STATUS current
  DESCRIPTION
    "Network Mask for the OAM IP Address."
  ::= { network 2 }


RCSTConfigNetworkTrfIpAddr OBJECT-TYPE
  SYNTAX IpAddress
  MAX-ACCESS read-write
  STATUS current
  DESCRIPTION
    "Current TRAFFIC IP Address of the RCST.
    This object used with both ip and interfaces MIB-II subgroups determines
uniquely the interface through which user traffic is passing through.
    By default, it is equal to the same address as in OID
RCSTConfigNetworkDefaultTrfIpAddr.
    This OID, RCSTConfigNetworkDefaultTrfIpAddr OID and
RCSTConfigNetworkTrfIpAddrAssign OID are linked together."
  ::= { network 3 }


RCSTConfigNetworkDefaultTrfIpAddr OBJECT-TYPE
  SYNTAX IpAddress
  MAX-ACCESS read-write
  STATUS current
  DESCRIPTION
    "Default TRAFFIC IP Address of the RCST. This object shall be the initial
address used in case the initial TRAFFIC IP address given by OID
RCSTConfigNetworkTrfIpAddr would change while in session in order to keep track
of the 'default' address. Its value shall never change while in session.
    RCSTConfigNetworkTrfIpAddr OID, this OID and
RCSTConfigNetworkTrfIpAddrAssign OID are linked together."
```

```
  ::= { network 4 }


RCSTConfigNetworkTrfIpAddrAssign OBJECT-TYPE
  SYNTAX INTEGER
    {
    static(1),
    dynamic(2)
    }
  MAX-ACCESS read-write
  STATUS current
  DESCRIPTION
    "Identifies whether the TRAFFIC IP address is statically ('static' - 1) or
dynamically ('dynamic' - 2) assigned.
    RCSTConfigNetworkTrfIpAddr OID and RCSTConfigNetworkDefaultTrfIpAddr OID
are linked to this object: when statically assigned, both OIDs represent the
same IP address. In case the IP address is dynamically assigned,
RCSTConfigNetworkTrfIpAddr OID will be the dynamic IP address while
RCSTConfigNetworkDefaultTrfIpAddr OID remains the static IP address."
  ::= { network 5 }


RCSTConfigNetworkTrfIpNetworkMask OBJECT-TYPE
  SYNTAX IpAddress
  MAX-ACCESS read-write
  STATUS current
  DESCRIPTION
    "Network Mask for the Traffic IP Address of the RCST."
  ::= { network 6 }


RCSTConfigNetworkDefaultTrfIpNetworkMask OBJECT-TYPE
  SYNTAX IpAddress
  MAX-ACCESS read-write
  STATUS current
  DESCRIPTION
    "Network Mask for the default Traffic IP Address of the RCST."
  ::= { network 7 }


RCSTConfigNetworkOamReleaseTimeout OBJECT-TYPE
  SYNTAX Integer32
  MAX-ACCESS read-write
  STATUS current
  DESCRIPTION
    "Time, specified in milliseconds, after which the RCST shall released the
Return Link and OAM Resources due to no OAM traffic (from Hosts or RCST) bound
for the Return Link, while one or more entries remain in the Host Authentication
table."
  ::= { network 8 }


RCSTConfigNetworkOamIpAddrAssign OBJECT-TYPE
  SYNTAX INTEGER
    {
    static(1),
    dynamic(2)
    }
  MAX-ACCESS read-write
  STATUS current
  DESCRIPTION
    "Identifies whether the OAM IP address is statically ('static'-1) or
dynamically ('dynamic'-2) assigned."
```

```
  ::= { network 9 }


RCSTConfigNetworkAuthPriIpAddr OBJECT-TYPE
  SYNTAX IpAddress
  MAX-ACCESS read-write
  STATUS current
  DESCRIPTION
    "IP address within the NCC, to which the RCST shall send authentication
messages."
  ::= { network 10 }


RCSTConfigNetworkAuthSecIpAddr OBJECT-TYPE
  SYNTAX IpAddress
  MAX-ACCESS read-write
  STATUS current
  DESCRIPTION
    "IP address within the NCC, to which the RCST shall send authentication
messages."
  ::= { network 11 }


accessPolicy OBJECT IDENTIFIER ::= { RCSTConfig 2 }


RCSTConfigAccessPolicyNextIndex OBJECT-TYPE
  SYNTAX Integer32
  MAX-ACCESS read-only
  STATUS current
  DESCRIPTION
    "Provides the next available index in the dynamic table
RCSTConfigAccessPolicyTable. The row-creation algorithm shall use this value."
  ::= { accessPolicy 1 }


RCSTConfigAccessPolicyTable OBJECT-TYPE
  SYNTAX SEQUENCE OF RCSTConfigAccessPolicyEntry
  MAX-ACCESS not-accessible
  STATUS current
  DESCRIPTION
    "The entries in this table map domain IP subnets to community names and
management entity names to IP addresses. The community names are used by the
RCST to determine the kind of access right the originator of an SNMP request
has."
  ::= { accessPolicy 2 }


RCSTConfigAccessPolicyEntry OBJECT-TYPE
  SYNTAX RCSTConfigAccessPolicyEntry
  MAX-ACCESS not-accessible
  STATUS current
  DESCRIPTION
    "This represents one entry in the RCSTConfigAccessPolicyTable table. Each
row assigns a particular community name to a subnet or an IP address (depending
on the definition of the network mask)."
  INDEX { RCSTConfigAccessPolicyIndex }
  ::= { RCSTConfigAccessPolicyTable 1 }


RCSTConfigAccessPolicyEntry ::=
  SEQUENCE {
    RCSTConfigAccessPolicyIndex
```

```
          Integer32,
        RCSTConfigAccessPolicyIpAddr
          IpAddress,
        RCSTConfigAccessPolicyNetMask
        IpAddress,RCSTConfigAccessPolicyCommunityName
          DisplayString,
        RCSTConfigAccessPolicyEntity
          DisplayString,
        RCSTConfigAccessPolicyStatus
          RowStatus
        }


RCSTConfigAccessPolicyIndex OBJECT-TYPE
    SYNTAX Integer32
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Its value is used to index this table."
    ::= { RCSTConfigAccessPolicyEntry 1 }


RCSTConfigAccessPolicyIpAddr OBJECT-TYPE
    SYNTAX IpAddress
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "IP address of a network management entity.
        Note that this IP address can be any address belonging to the subnet
defined using the OID 2.1.3 RCSTConfigAccessPolicyNetMask."
    ::= { RCSTConfigAccessPolicyEntry 2 }


RCSTConfigAccessPolicyNetMask OBJECT-TYPE
    SYNTAX IpAddress
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "Network mask for the management entity referred to in OID 2.1.5 of this
table.
        Note that this network mask can be set to 255.255.255.255 so that it maps
to a unique IP address."
    ::= { RCSTConfigAccessPolicyEntry 3 }


RCSTConfigAccessPolicyCommunityName OBJECT-TYPE
    SYNTAX DisplayString (SIZE (0..255))
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "Corresponding community name."
    ::= { RCSTConfigAccessPolicyEntry 4 }


RCSTConfigAccessPolicyEntity OBJECT-TYPE
    SYNTAX DisplayString (SIZE (0..255))
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "Management entity name."
    ::= { RCSTConfigAccessPolicyEntry 5 }
```

```
RCSTConfigAccessPolicyStatus OBJECT-TYPE
  SYNTAX RowStatus
  MAX-ACCESS read-create
  STATUS current
  DESCRIPTION
    "This object enables the table to be dynamic. Moreover, it provides some
information about the validity of each row. It also enables creation/deletion of
rows."
  ::= { RCSTConfigAccessPolicyEntry 6 }


RCSTConfigAccessPolMibViewNextIndex OBJECT-TYPE
  SYNTAX Integer32
  MAX-ACCESS read-only
  STATUS current
  DESCRIPTION
    "Provides the next available index in the dynamic table
RCSTConfigAccessPolMibViewTable. The row-creation algorithm shall use this
value."
  ::= { accessPolicy 3 }


RCSTConfigAccessPolMibViewTable OBJECT-TYPE
  SYNTAX SEQUENCE OF RCSTConfigAccessPolMibViewEntry
  MAX-ACCESS not-accessible
  STATUS current
  DESCRIPTION
    "The entries in this table describe the mapping of community names,
accessible object groups and the access privileges.
    An entry contains the community name, the OID branch accessible by members
of this community and the level of access privilege."
  ::= { accessPolicy 4 }


RCSTConfigAccessPolMibViewEntry OBJECT-TYPE
  SYNTAX RCSTConfigAccessPolMibViewEntry
  MAX-ACCESS not-accessible
  STATUS current
  DESCRIPTION
    "Description."
  INDEX { RCSTConfigAccessPolMibViewIndex }
  ::= { RCSTConfigAccessPolMibViewTable 1 }


RCSTConfigAccessPolMibViewEntry ::=
  SEQUENCE {
    RCSTConfigAccessPolMibViewIndex
      Integer32,
    RCSTConfigAccessPolMibViewCommunityName
      DisplayString,
    RCSTConfigAccessPolMibViewPrefix
      OBJECT IDENTIFIER,
    RCSTConfigAccessPolMibViewAccessRight
      INTEGER,
    RCSTConfigAccessPolMibViewStatus
      RowStatus
    }


RCSTConfigAccessPolMibViewIndex OBJECT-TYPE
  SYNTAX Integer32
  MAX-ACCESS not-accessible
  STATUS current
  DESCRIPTION
    "Its value is used to index this table."
  ::= { RCSTConfigAccessPolMibViewEntry 1 }


RCSTConfigAccessPolMibViewCommunityName OBJECT-TYPE
  SYNTAX DisplayString (SIZE (0..255))
  MAX-ACCESS read-create
  STATUS current
  DESCRIPTION
    "Community name of a network management entity."
  ::= { RCSTConfigAccessPolMibViewEntry 2 }


RCSTConfigAccessPolMibViewPrefix OBJECT-TYPE
  SYNTAX OBJECT IDENTIFIER
  MAX-ACCESS read-create
  STATUS current
  DESCRIPTION
    "Branch OID of the object groups which members of this community can
access."
  ::= { RCSTConfigAccessPolMibViewEntry 3 }


RCSTConfigAccessPolMibViewAccessRight OBJECT-TYPE
  SYNTAX INTEGER
    {
    read-only(1),
    read-write(2),
    not-accessible(3)
    }
  MAX-ACCESS read-create
  STATUS current
  DESCRIPTION
    "Level of access right the community has. The supported values shall be
'read-only' ('1'), 'read-write' ('2'), and 'not-accessible' ('3')."
  ::= { RCSTConfigAccessPolMibViewEntry 4 }


RCSTConfigAccessPolMibViewStatus OBJECT-TYPE
  SYNTAX RowStatus
  MAX-ACCESS read-create
  STATUS current
  DESCRIPTION
    "This object enables the table to be dynamic. Moreover, it provides some
information about the validity of each row. It also enables creation/deletion of
rows."
  ::= { RCSTConfigAccessPolMibViewEntry 5 }


lines OBJECT IDENTIFIER ::= { RCSTConfig 3 }

airIf OBJECT IDENTIFIER ::= { lines 1 }

rtnLk OBJECT IDENTIFIER ::= { airIf 1 }


RCSTConfigLinesAirIfRtnLkMaxEirp OBJECT-TYPE
  SYNTAX Integer32
  MAX-ACCESS read-write
  STATUS current
  DESCRIPTION
    "Maximum allowed EIRP in tenth of dBm on the return link."
  ::= { rtnLk 1 }


RCSTConfigLinesAirIfRtnLkFreqHopRange OBJECT-TYPE
  SYNTAX Integer32
  MAX-ACCESS read-write
  STATUS current
  DESCRIPTION
    "Gives the frequency hopping range of the RCST on the return channel. The
value shall be given in kHz. Up to now, 20000khz and 40000khz are the only
supported values.
    Refer to the DVB-RCS normative document EN 301 790 for more details - RCST
Capability 24-bit field."
  ::= { rtnLk 2 }


RCSTConfigLinesAirIfRtnLkDefIfLevel OBJECT-TYPE
  SYNTAX Integer32
  MAX-ACCESS read-write
  STATUS current
  DESCRIPTION
    "Default transmitted IF power level, specified in tenth of dBm, out of the
IDU for sending a CSC burst at RCST reboot or power on."
  ::= { rtnLk 3 }


RCSTConfigLinesAirIfRtnLkFirstIfLevel OBJECT-TYPE
  SYNTAX Integer32
  MAX-ACCESS read-write
  STATUS current
  DESCRIPTION
    "IF level at which the RCST received a first response during installation
when sending CSC bursts repeatedly with increasing level. (By adding a specific
offset to this level the RCST determines during installation the default value
of RCSTSysInstallRtnLkDefIFLevel.) Its value shall be given in tenth of dBm."
  ::= { rtnLk 4 }


access OBJECT IDENTIFIER ::= { airIf 2 }


RCSTConfigLinesAirIfAccessMode OBJECT-TYPE
  SYNTAX INTEGER
    {
    normal(1),
    abnormal(2)
    }
  MAX-ACCESS read-only
  STATUS current
  DESCRIPTION
    "Indicates whether the RCST is performing a 'normal' ('1') logon procedure
or an 'abnormal' ('2') access procedure (i.e. a disaster recovery process)."
  ::= { access 1 }


RCSTConfigLinesAirIfAccessTable OBJECT-TYPE
  SYNTAX SEQUENCE OF RCSTConfigLinesAirIfAccessEntry
  MAX-ACCESS not-accessible
  STATUS current
  DESCRIPTION
    "This table gathers all the different times/delays that are relevant to an
access procedure whatever the mode: normal or abnormal."
  ::= { access 2 }
```

```
RCSTConfigLinesAirIfAccessEntry OBJECT-TYPE
  SYNTAX RCSTConfigLinesAirIfAccessEntry
  MAX-ACCESS not-accessible
  STATUS current
  DESCRIPTION
    "Description."
  INDEX { RCSTConfigLinesAirIfAccessIndex }
  ::= { RCSTConfigLinesAirIfAccessTable 1 }


RCSTConfigLinesAirIfAccessEntry ::=
  SEQUENCE {
    RCSTConfigLinesAirIfAccessIndex
      Integer32,
    RCSTConfigLinesAirIfAccessTinitial
      Integer32,
    RCSTConfigLinesAirIfAccessTto
      Integer32,
    RCSTConfigLinesAirIfAccessTresendMin
      Integer32,
    RCSTConfigLinesAirIfAccessTresendMax
      Integer32,
    RCSTConfigLinesAirIfAccessCscMax
      Integer32,
    RCSTConfigLinesAirIfAccessNMax
      Integer32,
    RCSTConfigLinesAirIfAccessTBackOffN
      Integer32
    }


RCSTConfigLinesAirIfAccessIndex OBJECT-TYPE
  SYNTAX Integer32
  MAX-ACCESS not-accessible
  STATUS current
  DESCRIPTION
    "A unique value characterizes each access mode. This matches the
RCSTConfigLinesAirIfAccessMode value, in fact."
  ::= { RCSTConfigLinesAirIfAccessEntry 1 }


RCSTConfigLinesAirIfAccessTinitial OBJECT-TYPE
  SYNTAX Integer32
  MAX-ACCESS read-write
  STATUS current
  DESCRIPTION
    "This object defines the maximum delay the RCST will wait before it issues
the first CSC slot to get connected. This value shall be set to 0 when the RCST
is in the normal connection mode. It is part of the abnormal login process.
    Value shall be defined in multiples of superframe.
    Refer to the DVB-RCS normative document EN 301 790 for more details -
abnormal login procedure."
  ::= { RCSTConfigLinesAirIfAccessEntry 2 }


RCSTConfigLinesAirIfAccessTto OBJECT-TYPE
  SYNTAX Integer32
  MAX-ACCESS read-write
  STATUS current
  DESCRIPTION
    "This variable represents a fixed time-out delay the RCST has to wait in
the random access scheme after having sent a CSC slot. This time shall be
defined in multiples of superframe.
```

```
    Refer to the DVB-RCS normative document EN 301 790 for more details."
  ::= { RCSTConfigLinesAirIfAccessEntry 3 }


RCSTConfigLinesAirIfAccessTresendMin OBJECT-TYPE
  SYNTAX Integer32
  MAX-ACCESS read-write
  STATUS current
  DESCRIPTION
    "This object defines the minimum delay the RCST shall wait before retrying
to send a CSC slot when no reply was received after
RCSTConfigLinesAirIfAccessTto. In the DVB-RCS normative document EN 301 790,
this value is equal to 0.
    Refer to the DVB-RCS normative document EN 301 790 for more details -
Tresend parameter."
  DEFVAL { 0 }
  ::= { RCSTConfigLinesAirIfAccessEntry 4 }


RCSTConfigLinesAirIfAccessTresendMax OBJECT-TYPE
  SYNTAX Integer32
  MAX-ACCESS read-write
  STATUS current
  DESCRIPTION
    "This object defines the maximum delay the RCST shall wait before retrying
to send a CSC slot when no reply was received after
RCSTConfigLinesAirIfAccessTto. In the DVB-RCS normative document EN 301 790,
this parameter is referred to as 'Tresend'.
    Refer to the DVB-RCS normative document EN 301 790 for more details -
Tresend parameter."
  ::= { RCSTConfigLinesAirIfAccessEntry 5 }


RCSTConfigLinesAirIfAccessCscMax OBJECT-TYPE
  SYNTAX Integer32
  MAX-ACCESS read-write
  STATUS current
  DESCRIPTION
    "This object gives the maximum number of CSC slots / connection attempts a
RCST is allowed to issue. Once this threshold has been reached, the RCST shall
wait for RCSTConfigLinesAirIfAccessTBackOffn time before retrying the complete
login process. Moreover, every time this amount is exceeded, an internal
“AccessN" counter in the terminal shall be incremented by 1.
    Refer to the DVB-RCS normative document EN 301 790 for more details -
normal login procedure."
  ::= { RCSTConfigLinesAirIfAccessEntry 6 }


RCSTConfigLinesAirIfAccessNMax OBJECT-TYPE
  SYNTAX Integer32
  MAX-ACCESS read-write
  STATUS current
  DESCRIPTION
    "This threshold shall define the maximum value for an internal “AccessN"
counter in the terminal. Once this threshold has been reached, the RCST shall
wait for an operator intervention before retrying a complete login procedure.
    In other words, this parameter represents the maximum number of time
acquisition is retried before the IDU must disable its modulator output.
    Refer to the DVB-RCS normative document EN 301 790 for more details -
normal login procedure."
  ::= { RCSTConfigLinesAirIfAccessEntry 7 }
```

```
RCSTConfigLinesAirIfAccessTBackOffN OBJECT-TYPE
  SYNTAX Integer32
  MAX-ACCESS read-write
  STATUS current
  DESCRIPTION
    "This defines the time a RCST has to wait before retrying the complete
login process once it has reached the maximum amount of CSC slots it is allowed
to use for this purpose. Note that this delay is defined according to an
exponential back-off process, which is function of the internal “AccessN"
counter in the terminal.
    Refer to the DVB-RCS normative document EN 301 790 for more details -
normal login procedure."
  ::= { RCSTConfigLinesAirIfAccessEntry 8 }


RCSTConfigLinesAirIfAccessNetId OBJECT-TYPE
  SYNTAX Integer32
  MAX-ACCESS read-write
  STATUS current
  DESCRIPTION
    "Interactive network identifier."
  ::= { access 3 }


RCSTConfigLinesAirIfAccessPopId OBJECT-TYPE
  SYNTAX Integer32
  MAX-ACCESS read-write
  STATUS current
  DESCRIPTION
    "Population identifier."
  ::= { access 4 }


RCSTConfigLinesAirIfAccessStartTranspFreq OBJECT-TYPE
  SYNTAX Integer32
  MAX-ACCESS read-write
  STATUS current
  DESCRIPTION
    "Frequency of the start transponder carrying a DVB Network Information
Table (NIT) to which any RCST shall trigger to acquire forward link.
    Its value shall be given in multiple of 100 kHz."
  ::= { access 5 }


RCSTConfigLinesAirIfAccessStartTranspOrbPos OBJECT-TYPE
  SYNTAX Integer32
  MAX-ACCESS read-write
  STATUS current
  DESCRIPTION
    "Orbital position of the satellite casting the start transponder that
carries a DVB Network Information Table (NIT) to which any RCST shall trigger to
acquire forward link.
    Its value shall be given in tenth of degree, i.e. '192' would represent the
19.2 degrees position. The installer shall define this parameter."
  ::= { access 6 }


RCSTConfigLinesAirIfAccessStartTranspPolar OBJECT-TYPE
  SYNTAX Integer32
  MAX-ACCESS read-write
  STATUS current
  DESCRIPTION
```

```
    "2-bit field giving the polarization of the start transponder carrying a
DVB Network Information Table (NIT) to which any RCST shall trigger to acquire
forward link:
linear and horizontal('00'),
linear and vertical  ('01'),
circular left    ('10'),
circular right   ('11')."
  ::= { access 7 }


RCSTConfigLinesAirIfAccessStartTranspFec OBJECT-TYPE
  SYNTAX Integer32
  MAX-ACCESS read-write
  STATUS current
  DESCRIPTION
    "Specifies the Forward Error Correction used on the start transponder
carrying a DVB Network Information Table (NIT) to which any RCST shall trigger
to acquire forward link."
  ::= { access 8 }


RCSTConfigLinesAirIfAccessStartTranspSymbRate OBJECT-TYPE
  SYNTAX Integer32
  MAX-ACCESS read-write
  STATUS current
  DESCRIPTION
    "Specifies the symbol rate on the start transponder carrying a DVB Network
Information Table (NIT) to which any RCST shall trigger to acquire forward link.
    Its value shall be given in multiple of 100 symbol/s."
  ::= { access 9 }


RCSTLife OBJECT IDENTIFIER ::= { RCSTMIBObjects 3 }

RCSTStatus OBJECT IDENTIFIER ::= { RCSTLife 1 }


RCSTLifeRCSTStatusMode OBJECT-TYPE
  SYNTAX INTEGER
    {
    commissioning_mode(1),
    normal_operational_mode(2),
    reference_RCST_mode(3)
    }
  MAX-ACCESS read-only
  STATUS current
  DESCRIPTION
    "Indicates the current operational mode of the RCST. Supported values are:
    - 'commissioning_mode' ('1'),
    - 'normal_operational_mode' ('2'),
    - 'reference_RCST_mode' ('3').
    Other values are reserved for future use.
    Refer to the DVB-RCS normative document EN 301 790 for more details - RCST
Capability 24-bit field."
  ::= { RCSTStatus 1 }


RCSTLifeRCSTStatusCurrent OBJECT-TYPE
  SYNTAX INTEGER
    {
    idle(1),
    initialized(2),
    hold(3),
    oam_active(4),
    active(5),
    fault(6)
    }
  MAX-ACCESS read-only
  STATUS current
  DESCRIPTION
    "Defines the current RCST operational status:
    - 'idle' ('1'),
    - 'initialized' ('2'),
    - 'hold' ('3'),
    - 'oam_active' ('4'),
    - 'active' ('5'),
    - 'fault(6)
    Other values shall be reserved for future use."
  DEFVAL { 1 }
  ::= { RCSTStatus 2 }


RCSTLifeRCSTStatusCurrentTimestamp OBJECT-TYPE
  SYNTAX TimeStamp
  MAX-ACCESS read-only
  STATUS current
  DESCRIPTION
    "Timestamp indicating at what time the current operational status has been
reached by the RCST. Every time the RCST status changes, the value of this
object shall be assigned to RCSTLifeRCSTStatusPreviousTimeStamp and then updated
to be equal to the sysUpTime MIB-II object value."
  ::= { RCSTStatus 3 }


RCSTLifeRCSTStatusPrevious OBJECT-TYPE
  SYNTAX INTEGER
    {
    idle(1),
    initialized(2),
    hold(3),
    oam_active(4),
    active(5),
    fault(6)
    }
  MAX-ACCESS read-only
  STATUS current
  DESCRIPTION
    "Defines the previous RCST operational status:
    - 'idle' ('1'),
    - 'initialized' ('2'),
    - 'hold' ('3'),
    - 'oam_active' ('4'),
    - 'active' ('5'),
    - fault(6)
    Other values shall be reserved for future use."
  DEFVAL { 1 }
  ::= { RCSTStatus 4 }


RCSTLifeRCSTStatusPreviousTimestamp OBJECT-TYPE
  SYNTAX TimeStamp
  MAX-ACCESS read-only
  STATUS current
  DESCRIPTION
```

```
    "Timestamp indicating at what time the previous operational status has been
reached by the RCST. This object receives the old value of
RCSTLifeRCSTStatusCurrentTimestamp every time the RCST status changes."
  ::= { RCSTStatus 5 }


RCSTLifeRCSTStatusRebootStatus OBJECT-TYPE
  SYNTAX INTEGER
    {
    success(1),
    failure(2)
    }
  MAX-ACCESS read-only
  STATUS current
  DESCRIPTION
    "Provide some information about the result of the reboot process. The
following values shall be supported:
    - success ('1')
    - failure ('2')"
  ::= { RCSTStatus 6 }


RCSTLifeRCSTStatusDownloadStatus OBJECT-TYPE
  SYNTAX INTEGER
    {
    success(1),
    failure(2)
    }
  MAX-ACCESS read-only
  STATUS current
  DESCRIPTION
    "Provide some information about the result of the download process. The
following values shall be supported:
      - success ('1')
      - failure ('2')"
  ::= { RCSTStatus 7 }


RCSTLifeRCSTStatusCurrentImageId OBJECT-TYPE
  SYNTAX Integer32
  MAX-ACCESS read-only
  STATUS current
  DESCRIPTION
    "Current image of the Software Download material.
    RCST Images will be distributed to the RCST in one binary file.
    In the Software download the NCC is the FTP server, the RCST is the FTP
client and the transfer shall be made in binary mode."
  ::= { RCSTStatus 8 }


RCSTLifeRCSTStatusCurrentImageValidFlag OBJECT-TYPE
  SYNTAX INTEGER
    {
    valid(1),
    not_valid(2)
    }
  MAX-ACCESS read-only
  STATUS current
  DESCRIPTION
    "Status of the image ID given by RCSTLifeRCSTStatusCurrentImageId. The
following values are supported:
    - 'valid' (1)
    - 'not_valid' (2)."
  ::= { RCSTStatus 9 }


RCSTLifeRCSTStatusAlternateImageId OBJECT-TYPE
  SYNTAX Integer32
  MAX-ACCESS read-only
  STATUS current
  DESCRIPTION
    "Alternate image of the Software Download material.
    RCST Images will be distributed to the RCST in one binary file.
    In the Software download the NCC is the FTP server, the RCST is the FTP
client and the transfer shall be made in binary mode."
  ::= { RCSTStatus 10 }


RCSTLifeRCSTStatusAlternateImageValidFlag OBJECT-TYPE
  SYNTAX INTEGER
    {
    valid(1),
    not_valid(2)
    }
  MAX-ACCESS read-only
  STATUS current
  DESCRIPTION
    "Status of the image ID given by RCSTLifeRCSTStatusAlternateImageId. The
following values are supported:
    - 'valid' (1)
    - 'not_valid' (2)."
  ::= { RCSTStatus 11 }


trapLog OBJECT IDENTIFIER ::= { RCSTLife 2 }


RCSTLifeTrapLogTable OBJECT-TYPE
  SYNTAX SEQUENCE OF RCSTLifeTrapLogEntry
  MAX-ACCESS not-accessible
  STATUS current
  DESCRIPTION
    "This table contains a history of the past 255 traps this RCST has sent to
the remote manager. It provides a mechanism for the remote manager to detect any
lost trap and to recover information, which may otherwise be lost.
    A trap is logged in this table before it is sent. In the event that a trap
gets lost, the remote manager can detect the loss by a gap in the index value of
the variables included in the trap. It can then choose to retrieve the
information, which were sent in the trap. Note that this table does not log the
traps sent to the TM manager."
  ::= { trapLog 1 }


RCSTLifeTrapLogEntry OBJECT-TYPE
  SYNTAX RCSTLifeTrapLogEntry
  MAX-ACCESS not-accessible
  STATUS current
  DESCRIPTION
    "Description."
  INDEX { RCSTLifeTrapLogIndex }
  ::= { RCSTLifeTrapLogTable 1 }


RCSTLifeTrapLogEntry ::=
  SEQUENCE {
    RCSTLifeTrapLogIndex
```

```
        Integer32,
     RCSTLifeTrapLogNumber
        OBJECT IDENTIFIER,
     RCSTLifeTrapLogEventTime
        TimeStamp
     }


RCSTLifeTrapLogIndex OBJECT-TYPE
  SYNTAX Integer32
  MAX-ACCESS not-accessible
  STATUS current
  DESCRIPTION
     "Table index, which uniquely identifies an entry in this table. Note that
even though this table will have a maximum number of entries, say 255, it should
be implemented as a wrap-around FIFO queue with a monotonously increasing index
number up to the maximum representable by the processor to facilitate the
detection of lost traps."
  ::= { RCSTLifeTrapLogEntry 1 }


RCSTLifeTrapLogNumber OBJECT-TYPE
  SYNTAX OBJECT IDENTIFIER
  MAX-ACCESS read-only
  STATUS current
  DESCRIPTION
     "The trap OID."
  ::= { RCSTLifeTrapLogEntry 2 }


RCSTLifeTrapLogEventTime OBJECT-TYPE
  SYNTAX TimeStamp
  MAX-ACCESS read-only
  STATUS current
  DESCRIPTION
     "Identifies the time of occurrence of the event that generated the current
Trap. The time shall be given according to the sysUpTime SNMP MIB-II value."
  ::= { RCSTLifeTrapLogEntry 3 }


trapDestination OBJECT IDENTIFIER ::= { RCSTLife 3 }


RCSTLifeTrapDestTableNextIndex OBJECT-TYPE
  SYNTAX Integer32
  MAX-ACCESS read-only
  STATUS current
  DESCRIPTION
     "The next available index to be used for a new row creation in the trap
destination table."
  ::= { trapDestination 1 }


RCSTLifeTrapDestTable OBJECT-TYPE
  SYNTAX SEQUENCE OF RCSTLifeTrapDestEntry
  MAX-ACCESS not-accessible
  STATUS current
  DESCRIPTION
     "This table defines the destination entities for traps generated by the
RCST SNMP agent.
     Typical configuration for the RCST is to send network management traps to
the SMS, and call control traps to the TM in the NCC."
  ::= { trapDestination 2 }
```

```
RCSTLifeTrapDestEntry OBJECT-TYPE
  SYNTAX RCSTLifeTrapDestEntry
  MAX-ACCESS not-accessible
  STATUS current
  DESCRIPTION
    "Description."
  INDEX { RCSTLifeTrapDestIndex }
  ::= { RCSTLifeTrapDestTable 1 }


RCSTLifeTrapDestEntry ::=
  SEQUENCE {
    RCSTLifeTrapDestIndex
      Integer32,
    RCSTLifeTrapDestEntity
      DisplayString,
    RCSTLifeTrapDestOid
      OBJECT IDENTIFIER,
    RCSTLifeTrapDestStatus
      RowStatus
    }


RCSTLifeTrapDestIndex OBJECT-TYPE
  SYNTAX Integer32
  MAX-ACCESS not-accessible
  STATUS current
  DESCRIPTION
    "A value that uniquely identifies this entry."
  ::= { RCSTLifeTrapDestEntry 1 }


RCSTLifeTrapDestEntity OBJECT-TYPE
  SYNTAX DisplayString (SIZE (0..255))
  MAX-ACCESS read-create
  STATUS current
  DESCRIPTION
    "Trap destination management entity."
  ::= { RCSTLifeTrapDestEntry 2 }


RCSTLifeTrapDestOid OBJECT-TYPE
  SYNTAX OBJECT IDENTIFIER
  MAX-ACCESS read-create
  STATUS current
  DESCRIPTION
    "The OID of the trap to be sent."
  ::= { RCSTLifeTrapDestEntry 3 }


RCSTLifeTrapDestStatus OBJECT-TYPE
  SYNTAX RowStatus
  MAX-ACCESS read-create
  STATUS current
  DESCRIPTION
    "The entity configuring this destination entry should set this variable to
Active once it has provided all the necessary information in this table and is
ready to receive traps. If the destination community address and subnet have not
been configured, this set will fail. Traps will only be sent to those
destinations whose status is Active. To suspend receiving traps, the owner of
the destination can set this variable to notInService. To remove an entry from
this table, set the status to Destroy."
  ::= { RCSTLifeTrapDestEntry 4 }
```

```
trap OBJECT IDENTIFIER ::= { RCSTLife 4 }


RCSTLifeTrapRCSTDownloadStatus NOTIFICATION-TYPE
   OBJECTS {
      RCSTSysInstallRCSTMacAddr, RCSTSysInstallSSPId,
RCSTLifeRCSTStatusDownloadStatus, RCSTLifeTrapLogEventTime
   }
   STATUS current
   DESCRIPTION
      "Sent by RCST to SMS upon software download completion in ord0er to inform
the SMS about the success or failure of the process.
      Contains the following varbinds:
         - RCSTSysInstallRCSTMacAddr,
         - RCSTSysInstallSSPId,
         - RCSTLifeRCSTStatusDownloadStatus (failure or
success)RCSTLifeTrapLogEventTime."
   ::= { trap 1 }


RCSTLifeTrapRCSTRebootStatus NOTIFICATION-TYPE
   OBJECTS {
      RCSTSysInstallRCSTMacAddr, RCSTSysInstallSSPId,
RCSTLifeRCSTStatusCurrentImageId, RCSTLifeRCSTStatusRebootStatus,
RCSTLifeTrapLogEventTime
   }
   STATUS current
   DESCRIPTION
      "Sent by RCST to SMS in the scope of the RCST Reboot action (part of
Software Upgrade/Download process). This event is sent in order to provide the
NCC with information about the success or the failure of the action.
      Contains the following varbinds:
         - RCSTSysInstallRCSTMacAddr
         - RCSTSysInstallSSPId
         - RCSTLifeRCSTStatusCurrentImageId
         - RCSTLifeRCSTStatusRebootStatus (failure or success)
         - RCSTLifeTrapLogEventTime."
   ::= { trap 2 }


RCSTLifeTrapPingStatus NOTIFICATION-TYPE
   OBJECTS {
      RCSTSysInstallRCSTMacAddr, RCSTSysInstallSSPId,  RCSTLifeTrapLogEventTime,
RCSTActPingResult, RCSTActPingTableIndex
}
   STATUS current
   DESCRIPTION
      "Always sent after completion of the Ping Device command requested by the
SMS. Contains the following varbinds:
      - RCSTSysInstallRCSTMacAddr,
      - RCSTSysInstallSSPId,
      - RCSTLifeTrapLogEventTime,
      - RCSTActPingResult Value,
      - RCSTActPingTableIndex."
   ::= { trap 3 }


RCSTCallCntl OBJECT IDENTIFIER ::= { RCSTMIBObjects 4 }

callCntl OBJECT IDENTIFIER ::= { RCSTCallCntl 1 }


RCSTCallCntlTmRCSTId OBJECT-TYPE
```

```
  SYNTAX PhysAddress
  MAX-ACCESS read-only
  STATUS current
  DESCRIPTION
    "This object is used by the Traffic Manager to identify the RCST sessions
it controls. Each SNMP trap exchange between a RCST and the TM must contain this
identifier.
The default value for this variable is the RCST's MAC address, which is the one
sent in the CSC burst and will be initialized to such by the RCST."
  ::= { callCntl 1 }


RCSTCallCntlAtmConnTableNextIndex OBJECT-TYPE
  SYNTAX Integer32
  MAX-ACCESS read-only
  STATUS current
  DESCRIPTION
    "The next available index to be used for a new row creation in the trap
destination table."
  ::= { callCntl 2 }


RCSTCallCntlAtmConnTable OBJECT-TYPE
  SYNTAX SEQUENCE OF RCSTCallCntlAtmConnEntry
  MAX-ACCESS not-accessible
  STATUS current
  DESCRIPTION
    "This table contains information about the ATM connections between the NCC
and the RCST."
  ::= { callCntl 3 }


RCSTCallCntlAtmConnEntry OBJECT-TYPE
  SYNTAX RCSTCallCntlAtmConnEntry
  MAX-ACCESS not-accessible
  STATUS current
  DESCRIPTION
    "Description."
  INDEX { RCSTCallCntlAtmConnIndex }
  ::= { RCSTCallCntlAtmConnTable 1 }


RCSTCallCntlAtmConnEntry ::=
  SEQUENCE {
  RCSTCallCntlAtmConnIndex
    Integer32,
  RCSTCallCntlAtmConnVpi
    Integer32,
  RCSTCallCntlAtmConnVci
    Integer32,
    RCSTCallCntlAtmConnType
      INTEGER,
    RCSTCallCntlAtmConnCraMax
      Integer32,
    RCSTCallCntlAtmConnVbdcMax
      Integer32,
  RCSTCallCntlAtmConnStatus
    RowStatus
  }


RCSTCallCntlAtmConnIndex OBJECT-TYPE
  SYNTAX Integer32
  MAX-ACCESS not-accessible
  STATUS current
  DESCRIPTION
    "This is the table index, which uniquely identifies an entry in this table.
To avoid lengthy negotiation between the TM and the RCST on which index to use,
the following implementation should be followed: index = 1 is reserved for the
OAM VCC; index = 2 is reserved for the traffic VCC which the TM autonomously
creates for the RCST at start up."
  ::= { RCSTCallCntlAtmConnEntry 1 }


RCSTCallCntlAtmConnVpi OBJECT-TYPE
  SYNTAX Integer32
  MAX-ACCESS read-create
  STATUS current
  DESCRIPTION
    "The VPI of an ATM connection between the NCC and the RCST."
  ::= { RCSTCallCntlAtmConnEntry 2 }


RCSTCallCntlAtmConnVci OBJECT-TYPE
  SYNTAX Integer32
  MAX-ACCESS read-create
  STATUS current
  DESCRIPTION
    "The VCI of an ATM connection between the NCC and the RCST."
  ::= { RCSTCallCntlAtmConnEntry 3 }


RCSTCallCntlAtmConnType OBJECT-TYPE
  SYNTAX INTEGER
    {
    oam_traffic(1),
    user_traffic(2)
    }
  MAX-ACCESS read-create
  STATUS current
  DESCRIPTION
    "Indicates whether the connection is used to carry OAM traffic or user
traffic. Supported values shall be oam_traffic (1) and user_traffic (2)."
  ::= { RCSTCallCntlAtmConnEntry 4 }


RCSTCallCntlAtmConnCraMax OBJECT-TYPE
  SYNTAX Integer32
  MAX-ACCESS read-create
  STATUS current
  DESCRIPTION
    "Maximum number of CRA bytes per second allocated to the corresponding VCC.
    This object is useful for the queuing mechanism.
    This object shall be defined through the TIM message for the OAM traffic.
While for the user traffic, it shall be set via SNMP Set message issued by the
TM."
  ::= { RCSTCallCntlAtmConnEntry 5 }


RCSTCallCntlAtmConnVbdcMax OBJECT-TYPE
  SYNTAX Integer32
  MAX-ACCESS read-create
  STATUS current
  DESCRIPTION
    "Maximum number of VBDC time slots per frame allocated to the corresponding
VCC.
```

```
     This object is useful for the queuing mechanism.
     This object shall be defined through the TIM message for the OAM traffic.
While for the user traffic, it shall be set via SNMP Set message issued by the
TM."
  ::= { RCSTCallCntlAtmConnEntry 6 }


RCSTCallCntlAtmConnStatus OBJECT-TYPE
  SYNTAX RowStatus
  MAX-ACCESS read-create
  STATUS current
  DESCRIPTION
     "The entries in the connection table are typically created or deleted
either by the RCST installer or by the TM. When the connection has been
successfully created, this variable should be set to active (1). To delete a
connection, set this variable to destroy (6). This variable is initialized to
notInService (2) when a row is created.
     Protocol:
     When the RCST wishes to obtain traffic transport capacity from the NCC
(i.e. to transition from the OAM_active to the active state), it sends a trap to
the TM to make a VCC request. The TM should utilize this table to communicate
the allocated resource to the RCST. In the event that allocation has failed, set
the VPI and VCI to 0 and this variable to notInService (4).
As a response to the RCST's traffic VCC release request trap, the TM should set
the corresponding RCSTCallCntlAtmConnStatus to destroy so that the RCST SNMP
agent can clean up the table entry.
In the case of the OAM VCC, the RCST will delete the entry after it has sent out
the RCSTCallCntlTrapOamReleaseRequest trap."
  ::= { RCSTCallCntlAtmConnEntry 7 }


RCSTCallCntlRCSTHoldCommand OBJECT-TYPE
  SYNTAX INTEGER
     {
     clear(1),
     hold(2)
     }
  MAX-ACCESS read-write
  STATUS current
  DESCRIPTION
     "This variable is used by the Traffic Manager to issue a Hold command to
the RCST. The Hold command implies a logoff and hold if the RCST is not already
in the Initialized state. It will disable the Return Link and the RCST will
transition from the Initialized to the Hold state.
     This variable will be reset when RCST Enable Command comes back.
     If RCST is already in Hold state then no response will be sent to any new
Hold Command."
  DEFVAL { 1 }
  ::= { callCntl 4 }


RCSTCallCntlRCSTLogoffConfirm OBJECT-TYPE
  SYNTAX INTEGER
     {
     not_confirmed(1),
     confirmed(2)
     }
  MAX-ACCESS read-write
  STATUS current
  DESCRIPTION
     "This variable is used by the Traffic Manager as a response to the RCST's
RCSTCallCntlTrapRCSTLogoffRequest trap (OID 5 Table 21) (TM confirms RCST Log-
off to the RCST). Setting this variable to Confirmed (2) will trigger the RCST
to do its own cleanup."
  ::= { callCntl 5 }


RCSTCallCntlRCSTLogoffCommand OBJECT-TYPE
  SYNTAX INTEGER
    {
    clear(1),
    logoff(2)
    }
  MAX-ACCESS read-write
  STATUS current
  DESCRIPTION
    "This variable is used by the Traffic Manager to issue a RCST Logoff
command to the RCST. (Sent by TM to RCST through an SNMP Set Request)
This variable will be set to Clear (1) when the RCST has successfully re-logged
in, i.e. successfully getting OAM VCC assigned by the TM."
  DEFVAL { 1 }
  ::= { callCntl 6 }


RCSTCallCntlUserLogoffTableNextIndex OBJECT-TYPE
  SYNTAX Integer32
  MAX-ACCESS read-only
  STATUS current
  DESCRIPTION
    "The next available index to be used for a new row creation in the trap
destination table."
  ::= { callCntl 7 }


RCSTCallCntlUserLogoffTable OBJECT-TYPE
  SYNTAX SEQUENCE OF RCSTCallCntlUserLogoffEntry
  MAX-ACCESS not-accessible
  STATUS current
  DESCRIPTION
    "This table is used to track the status of the User logoff request to the
TM. The RCST issues the user logoff request to the TM using an SNMP trap. When
the TM has finished cleaning the user resources it will set the Logoff status
variable to indicate that logoff is complete. Entries in this table are
therefore temporary. When the logoff process has been completed, its
corresponding entry will be removed.
This table will only contain Username / Host IP address entries that are also
present in the RCST's User Authentication table.
    The RCST User table could have easily supported this functionality. The
reason to have a separate table is to cleanly separate this control interface
from the network management interface at the cost of some small duplication."
  ::= { callCntl 8 }


RCSTCallCntlUserLogoffEntry OBJECT-TYPE
  SYNTAX RCSTCallCntlUserLogoffEntry
  MAX-ACCESS not-accessible
  STATUS current
  DESCRIPTION
    "Description."
  INDEX { RCSTCallCntlUserLogoffIndex }
  ::= { RCSTCallCntlUserLogoffTable 1 }


RCSTCallCntlUserLogoffEntry ::=
  SEQUENCE {
    RCSTCallCntlUserLogoffIndex
        Integer32,
    RCSTCallCntlUserName
        DisplayString,
    RCSTCallCntlUserHostIpAddr
        IpAddress,
        RCSTCallCntlUserLogoffStatus
            INTEGER,
        RCSTCallCntlUserStatus
        RowStatus
    }


RCSTCallCntlUserLogoffIndex OBJECT-TYPE
    SYNTAX Integer32
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Index of entry in the User Logoff Request Table."
    ::= { RCSTCallCntlUserLogoffEntry 1 }


RCSTCallCntlUserName OBJECT-TYPE
    SYNTAX DisplayString (SIZE (0..255))
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "Name of the user who has requested the logoff."
    ::= { RCSTCallCntlUserLogoffEntry 2 }


RCSTCallCntlUserHostIpAddr OBJECT-TYPE
    SYNTAX IpAddress
    MAX-ACCESS read-only
    STATUS current
    DESCRIPTION
        "IP address of Host from which User has logged in."
    ::= { RCSTCallCntlUserLogoffEntry 3 }


RCSTCallCntlUserLogoffStatus OBJECT-TYPE
    SYNTAX INTEGER
        {
        pending(1),
        complete(2),
        failed(3)
        }
    MAX-ACCESS read-create
    STATUS current
    DESCRIPTION
        "Status of the User Logoff Request.
        Initially set to 'pending' (1) when a new entry is created by the RCST SNMP
agent in this table. This is to identify that a User Logoff Request has been
issued to the TM but is pending. When the Logoff process has been completed, the
TM will set this variable to 'complete' (2). This has the cascading effect of
deleting the corresponding entry of the same Username from the User table and
also deleting this particular entry from this table.
In case there would be no reply from the TM, a time-out (the same as the one
used at login - Login Timer) would be raised and the variable would be set to
'failed' (3). The user would then receive a notification about the logoff
failure and would have to issue a new trial manually. Note that the NCC shall
also be able to set this object to 'failed' in case something would go wrong at
the NCC side."
```

```
   ::= { RCSTCallCntlUserLogoffEntry 4 }


RCSTCallCntlUserStatus OBJECT-TYPE
   SYNTAX RowStatus
   MAX-ACCESS read-create
   STATUS current
   DESCRIPTION
      "This object enables the table to be dynamic. Moreover, it provides some
information about the validity of each row. It also enables creation/deletion of
rows."
   ::= { RCSTCallCntlUserLogoffEntry 5 }


callCntlTrap OBJECT IDENTIFIER ::= { RCSTCallCntl 2 }


RCSTCallCntlTrapTrafficVCCRequest NOTIFICATION-TYPE
   OBJECTS { RCSTCallCntlTmRCSTId, RCSTCallCntlAtmConnTableNextIndex }
   STATUS current
   DESCRIPTION
      "Sent by the RCST to the TM to request a Traffic VCC as part of a re-login
process, to transition from the OAM Active state to the Active state and acquire
a Traffic VCC to support Host traffic. Contains the following varbinds:
         - RCSTCallCntlTmRCSTIdk,
         - RCSTCallCntlAtmConnTableNextIndex.
      The TM should communicate the response to this request by creating an entry
in the RCSTCallCntlAtmConnTable using the provided index."
   ::= { callCntlTrap 1 }


RCSTCallCntlTrapOamReleaseRequest NOTIFICATION-TYPE
   OBJECTS { RCSTCallCntlTmRCSTId }
   STATUS current
   DESCRIPTION
      "Sent by the RCST to the TM to request the release of the current OAM RL
resources, to transition from the OAM Active state to the Initialized state.
Contains the following varbinds:
      - RCSTCallCntlTmRCSTId."
   ::= { callCntlTrap 2 }


RCSTCallCntlTrapTrafficReleaseRequest NOTIFICATION-TYPE
   OBJECTS { RCSTCallCntlTmRCSTId, RCSTCallCntlAtmConnIndex }
   STATUS current
   DESCRIPTION
      "Sent by the RCST to the TM to request the release of the current Traffic
RL resources, but keep the current OAM RL resources, to transition from the
Active state to the OAM Active state. Contains the following varbinds:
   - RCSTCallCntlTmRCSTId
   - RCSTCallCntlAtmConnIndex
      The TM should communicate the response to this request by destroying an
entry in the RCSTCallCntlAtmConnTable using the provided index."
   ::= { callCntlTrap 3 }


RCSTCallCntlTrapUserLogoffRequest NOTIFICATION-TYPE
   OBJECTS {
      RCSTCallCntlTmRCSTId, RCSTCallCntlUserLogoffIndex, RCSTCallCntlUserName,
RCSTCallCntlHostIpAddr }
   STATUS current
   DESCRIPTION
```

```
     "Sent by the RCST to the TM to request the release of resources for a
specific User, i.e. User Logoff. Contains the following varbinds:
     - RCSTCallCntlTmRCSTId,
     - RCSTCallCntlUserLogoffIndex,
     - RCSTCallCntlUserName,
     - RCSTCallCntlHostIpAddr.
     The TM should communicate the response to this request by setting the
corresponding RCSTCallCntlUserLogoffStatus variable to 'complete (2)'."
   ::= { callCntlTrap 4 }


RCSTCallCntlTrapRCSTLogoffRequest NOTIFICATION-TYPE
   OBJECTS { RCSTCallCntlTmRCSTId }
   STATUS current
   DESCRIPTION
     "Sent by the RCST to the TM (for RCST log-off) to request the release of
all current Return Link resources, to transition from the Active or OAM Active
state to the Initialized or Hold state. Contains the following varbinds:
     - RCSTCallCntlTmRCSTId.
     The protocol is such that when the TM has finished its processing of this
request, it will set the RCSTCallCntlRCSTLogoffConfirm variable, so that the
RCST can do its own cleanup."
   ::= { callCntlTrap 5 }


callCntlMpeg OBJECT IDENTIFIER ::= { RCSTCallCntl 3 }


RCSTCallCntlMpegConnTableNextIndex OBJECT-TYPE
   SYNTAX Integer32
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
     "The next available index to be used for a new row creation in the MPEG
Connection Table."
   ::= { callCntlMpeg 1 }


RCSTCallCntlMpegConnTable OBJECT-TYPE
   SYNTAX SEQUENCE OF RCSTCallCntlMpegConnEntry
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
     "This table contains information about the MPEG connections between the NCC
and the RCST."
   ::= { callCntlMpeg 2 }


RCSTCallCntlMpegConnEntry OBJECT-TYPE
   SYNTAX RCSTCallCntlMpegConnEntry
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
     "Description."
   INDEX { RCSTCallCntlMpegConnIndex }
   ::= { RCSTCallCntlMpegConnTable 1 }


RCSTCallCntlMpegConnEntry ::=
   SEQUENCE {
     RCSTCallCntlMpegConnIndex
       Integer32,
     RCSTCallCntlMpegConnType
```

```
        Integer32,
      RCSTCallCntlMpegConnPid
        Integer32,
      RCSTCallCntlMpegEntryStatus
        RowStatus
  }


RCSTCallCntlMpegConnIndex OBJECT-TYPE
  SYNTAX Integer32
  MAX-ACCESS not-accessible
  STATUS current
  DESCRIPTION
    "This is the table index, which uniquely identifies an entry in this table.
To avoid lengthy negotiation between the TM and the RCST on which index to use,
the following implementation should be followed: index = 1 is reserved for the
CTRL MNGM PID (DULM); index = 2 is reserved for the traffic PID which the TM
autonomously creates for the RCST at start up."
  ::= { RCSTCallCntlMpegConnEntry 1 }


RCSTCallCntlMpegConnType OBJECT-TYPE
  SYNTAX Integer32
    {
    CtrlMngm(1),
    Trf(2),
    NativeMpeg(3),
    AdditionalTrfPid(4)
    }
  MAX-ACCESS read-create
  STATUS current
  DESCRIPTION
    "Type of connection to which the PID is associated:
    1- CTRL_MNGM (DULM traffic)
    2- TRF (user traffic)
    3- Native MPEG (traffic)
    4- Additional TRF PID"
  ::= { RCSTCallCntlMpegConnEntry 2 }


RCSTCallCntlMpegConnPid OBJECT-TYPE
  SYNTAX Integer32
  MAX-ACCESS read-create
  STATUS current
  DESCRIPTION
    "The PID used for this type of connection between the RCST and the NCC.
Bits 13 to 31 unused."
  ::= { RCSTCallCntlMpegConnEntry 3 }


RCSTCallCntlMpegEntryStatus OBJECT-TYPE
  SYNTAX RowStatus
  MAX-ACCESS read-create
  STATUS current
  DESCRIPTION
    "The entries in the MPEG connection table are typically created or deleted
either by the RCST SNMP agent or by the NCC network manager."
  ::= { RCSTCallCntlMpegConnEntry 4 }


callCntlTrapMpeg OBJECT IDENTIFIER ::= { RCSTCallCntl 4 }
```

```
RCSTCallCntlTrapMpegPidRequest NOTIFICATION-TYPE
  OBJECTS { RCSTCallCntlTmRCSTId, RCSTCallCntlMpegConnTableNextIndex,
RCSTCallCntlMpegConnType }
  STATUS current
  DESCRIPTION
    "This trap is used by the RCST to request a PID cf RCSTCallCntlMpegConnType
3 or 4. Contains the following varbinds:
    - RCSTCallCntlTmRCSTId
    - RCSTCallCntlMpegConnTableNextIndex
    - RCSTCallCntlMpegConnType
    The TM shall respond to this request by creating an entry in the
RCSTCallCntlMpegConnTable, using the provided intex."
  ::= { callCntlTrapMpeg 1 }


RCSTCallCntlTrapMpegPidReleaseRequest NOTIFICATION-TYPE
  OBJECTS { RCSTCallCntlTmRCSTId, RCSTCallCntlMpegConnIndex }
  STATUS current
  DESCRIPTION
    "This trap is used to release a PID for MPEG. Contains the following
varbinds:
    - RCSTCallCntlTmRCSTId
    - RCSTCallCntlMpegConnIndex
    The TM shall respond to this request by removing an entry from the
RCSTCallCntlMpegConnTable, using the provided index."
  ::= { callCntlTrapMpeg 2 }


RCSTActions OBJECT IDENTIFIER ::= { RCSTMIBObjects 5 }


RCSTActDownloadUrl OBJECT-TYPE
  SYNTAX DisplayString (SIZE (0..255))
  MAX-ACCESS read-write
  STATUS current
  DESCRIPTION
    "This object provides the complete URL that is used by the download
process. This URL is made up of a identification (login), a password as well as
the complete file path."
  ::= { RCSTActions 1 }


RCSTActDownloadOwner OBJECT-TYPE
  SYNTAX DisplayString (SIZE (0..255))
  MAX-ACCESS read-write
  STATUS current
  DESCRIPTION
    "Identifies the entity that has initiated the download process."
  ::= { RCSTActions 2 }


RCSTActDownloadOwnerIpAddr OBJECT-TYPE
  SYNTAX IpAddress
  MAX-ACCESS read-write
  STATUS current
  DESCRIPTION
    "Provides the IP address of the entity that has initiated the download
process. Note that this entity is named in RCSTActDownloadOwner."
  ::= { RCSTActions 3 }


RCSTActPingTableNextIndex OBJECT-TYPE
  SYNTAX Integer32
```

```
   MAX-ACCESS read-only
   STATUS current
   DESCRIPTION
     "The next available table index for new row creation. (Suggestion: it may
be agreed on not to support this variable. The manager can just use some random
number for the table index)."
   ::= { RCSTActions 4 }


RCSTActPingTable OBJECT-TYPE
   SYNTAX SEQUENCE OF RCSTActPingEntry
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
     "This table supports the Ping command issued from a network management
entity to the RCST."
   ::= { RCSTActions 5 }


RCSTActPingEntry OBJECT-TYPE
   SYNTAX RCSTActPingEntry
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
     "Identifies the entity that has initiated the Ping. It can be used to avoid
collision in a multi-manager environment."
   INDEX { RCSTActPingTableIndex }
   ::= { RCSTActPingTable 1 }


RCSTActPingEntry ::=
   SEQUENCE {
     RCSTActPingTableIndex
       Integer32,
     RCSTActPingDestination
       IpAddress,
     RCSTActPingFrom
       INTEGER,
     RCSTActPingSize
       Integer32,
     RCSTActPingResultTimeStamp
       TimeStamp,
     RCSTActPingResult
       INTEGER,
     RCSTActPingTime
       Integer32,
     RCSTActPingOwnerIpAddr
       IpAddress,
     RCSTActPingRowStatus
       RowStatus
     }


RCSTActPingTableIndex OBJECT-TYPE
   SYNTAX Integer32
   MAX-ACCESS not-accessible
   STATUS current
   DESCRIPTION
     "An index uniquely identifying an entry in this table."
::= { RCSTActPingEntry 1 }


RCSTActPingDestination OBJECT-TYPE
```

```
  SYNTAX IpAddress
  MAX-ACCESS read-create
  STATUS current
  DESCRIPTION
    "Identifies the device to be pinged by the RCST using its IP address."
::= { RCSTActPingEntry 2 }


RCSTActPingFrom OBJECT-TYPE
  SYNTAX INTEGER
    {
    oam_ip_address(1),
    traffic_ip_address(2)
    }
  MAX-ACCESS read-create
  STATUS current
  DESCRIPTION
    "Identifies the RCST interface to issue the ping command from, either the
OAM IP address ('oam_ip_address' '1') or the Traffic IP address
(traffic_ip_address' '2')."
  ::= { RCSTActPingEntry 3 }


RCSTActPingSize OBJECT-TYPE
  SYNTAX Integer32
  MAX-ACCESS read-create
  STATUS current
  DESCRIPTION
    "Specifies the number of bytes to include in the ping command."
::= { RCSTActPingEntry 4 }


RCSTActPingResultTimeStamp OBJECT-TYPE
  SYNTAX TimeStamp
  MAX-ACCESS read-only
  STATUS current
  DESCRIPTION
    "The value of sysUpTime of the last Ping result."
::= { RCSTActPingEntry 5 }


RCSTActPingResult OBJECT-TYPE
  SYNTAX INTEGER
    {
    none(1),
    passed(2),
    failed(3),
inprogress(4)
    }
  MAX-ACCESS read-only
  STATUS current
  DESCRIPTION
    "Result of the last ping command. The supported values shall be: 'none'
(1), 'pass' (2), 'failed' (3), 'inprogress' (4)."
  ::= { RCSTActPingEntry 6 }


RCSTActPingTime OBJECT-TYPE
  SYNTAX Integer32
  MAX-ACCESS read-only
  STATUS current
  DESCRIPTION
```

```
     "Time response, in milliseconds, of the last ping command. Set to (0) if
ping command has failed."
   ::= { RCSTActPingEntry 7 }


RCSTActPingOwnerIpAddr OBJECT-TYPE
   SYNTAX IpAddress
   MAX-ACCESS read-create
   STATUS current
   DESCRIPTION
     "Identifies the entity that has initiated the Ping. It can be used to avoid
collision in a multi-manager environment."
   ::= { RCSTActPingEntry 8 }


RCSTActPingRowStatus OBJECT-TYPE
   SYNTAX RowStatus
   MAX-ACCESS read-create
   STATUS current
   DESCRIPTION
     "The supported RowStatus values are:
     - active (1),
     - notInService (2),
     - notReady (3),
     - createAndGo (4),
     - createAndWait (5),
     - destroy (6).
     When this variable is set to Active, it triggers the start of the Ping
test. If all the necessary information has not been provided, this set will
fail. When the Ping command has been executed, the agent will set this variable
to notInService. For more information about this textual convention, one shall
refer to IETF RFC 2579."
   ::= { RCSTActPingEntry 9 }


RCSTActRebootStatusTrapFlag OBJECT-TYPE
   SYNTAX INTEGER
     {
     send_trap(1),
     no_trap(2)
     }
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
     "Identifies whether the RCST is expected to send the SNMP Enterprise Trap
(RCST Reset/Reboot Status) after resetting/rebooting.
Set ('1' - 'send_ trap') by the SMS prior to issuing the Reset/Reboot command if
it wishes to receive a reset/reboot completion trap.  (Note to RCST designer:
this flag needs to be in non-volatile memory.)
     Cleared ('2' - 'no_trap') by the RCST after sending the Trap. This is done
so that if the RCST resets/reboots due to reasons other than the command from
the SMS, a trap will not be sent there."
   ::= { RCSTActions 6 }


RCSTActRebootCommand OBJECT-TYPE
   SYNTAX INTEGER
     {
     reboot(1),
     idle(2)
     }
   MAX-ACCESS read-write
   STATUS current
```

```
  DESCRIPTION
    "This variable shall force the RCST to reboot when set to 1 - 'reboot'.
Default value shall be 2 - 'idle'."
  DEFVAL { 2 }
  ::= { RCSTActions 7 }


RCSTActDownloadCommand OBJECT-TYPE
  SYNTAX INTEGER
    {
    download(1),
    idle(2)
    }
  MAX-ACCESS read-write
  STATUS current
  DESCRIPTION
    "This variable shall initiate a RCST download process when set to 1 -
'download'. Default value shall be 2 - 'idle'."
  DEFVAL { 2 }
  ::= { RCSTActions 8 }


RCSTActDropSACommand OBJECT-TYPE
  SYNTAX INTEGER
    {
    drop_sa(1),
    idle(2)
    }
  MAX-ACCESS read-write
  STATUS current
  DESCRIPTION
    "This variable shall force the RCST to drop its SA (in order to support
IPSec failure, for example) when set to 1 - 'drop_SA'. Default value shall be 2
- 'idle'."
  DEFVAL { 2 }
  ::= { RCSTActions 9 }


RCSTActBootImage OBJECT-TYPE
  SYNTAX Integer32
  MAX-ACCESS read-write
  STATUS current
  DESCRIPTION
    "This variable shall trigger a boot from the current image if set to 1 or
from the alternate image if set to 2. No action shall be taken in case another
value would be assigned to this SNMP object.
Note that when/before booting from the alternate image it is not required to
check if the alternate image is valid via an SNMP Set message."
  ::= { RCSTActions 10 }


RCSTActValidateCurrentImageCommand OBJECT-TYPE
  SYNTAX INTEGER
    {
    validate_current_image(1),
    idle(2)
    }
  MAX-ACCESS read-write
  STATUS current
  DESCRIPTION
    "This variable shall force the RCST to validate the current image when set
to 1 - 'validate_current_image', i.e. the one that it is currently running. When
not in used, it shall be equal to 2 - 'idle' - the default value.
```

```
     This command shall be available to the Installer, the Super-user as well as
the NCC."
   DEFVAL { 2 }
   ::= { RCSTActions 11 }


RCSTActTrafficReleaseTimeout OBJECT-TYPE
   SYNTAX Integer32
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
     "Time, specified in milliseconds, after which the RCST shall released its
Traffic VCC due to no traffic from Hosts bound for the Return Link, while one or
more entries remain in the Host Authentication table."
   ::= { RCSTActions 12 }


RCSTActDefaultDomainName OBJECT-TYPE
   SYNTAX DisplayString (SIZE (0..255))
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
     "Default domain name.
     The Super-user can enable/disable this mandatory feature and set the
default @domain name. The default @domain name will be permanently stored in the
RCST. The RCST will append the default @domain name to all unqualified logins.
Some explanations about this object can be found in the installation subgroup
section of this MIB specification."
   ::= { RCSTActions 13 }


RCSTActDefaultDomainStatus OBJECT-TYPE
   SYNTAX INTEGER
     {
     enabled(1),
     disabled(2)
     }
   MAX-ACCESS read-write
   STATUS current
   DESCRIPTION
     "Default domain name feature enabled (1) or disabled (2) by the Super-
user."
   ::= { RCSTActions 14 }

RCSTPrivate OBJECT IDENTIFIER ::= { RCSTMIBObjects 6 }


--
-- Conformance definitions
--


RCSTMIBConformance OBJECT IDENTIFIER ::= { RCSTMIB 2 }

RCSTMIBCompliances OBJECT IDENTIFIER ::= { RCSTMIBConformance 1 }

RCSTBasicCompliance MODULE-COMPLIANCE
   STATUS current
   DESCRIPTION
     "The compliance statement for the Satellite Interactive Terminal (RCST)."

MODULE -- RCSTMIB
```

```
-- Conditionally mandatory groups

GROUP RCSTSysInstallGroup
  DESCRIPTION
    "Mandatory in the RCST MIB."

GROUP RCSTSysIduGroup
  DESCRIPTION
    "Mandatory in the RCST MIB."

GROUP RCSTConfigNetworkGroup
  DESCRIPTION
    "Mandatory in the RCST MIB."

GROUP RCSTConfigAccessPolicyGroup
  DESCRIPTION
    "Mandatory in the RCST MIB."

GROUP RCSTConfigLinesAirIfRtnLkGroup
  DESCRIPTION
    "Mandatory in the RCST MIB."

GROUP RCSTConfigLinesAirIfAccessGroup
  DESCRIPTION
    "Mandatory in the RCST MIB."

GROUP RCSTLifeRCSTStatusGroup
  DESCRIPTION
    "Mandatory in the RCST MIB."

GROUP RCSTLifeTrapLogGroup
  DESCRIPTION
    "Mandatory in the RCST MIB."

GROUP RCSTLifeTrapDestGroup
  DESCRIPTION
    "Mandatory in the RCST MIB."

GROUP RCSTCallCntlGroup
  DESCRIPTION
    "Mandatory in the RCST MIB."

GROUP RCSTActionsGroup
  DESCRIPTION
    "Mandatory in the RCST MIB."

GROUP RCSTLifeTrapGroup
  DESCRIPTION
    "Mandatory in the RCST MIB."

GROUP RCSTCallCntlTrapGroup
  DESCRIPTION
    "Mandatory in the RCST MIB."

  ::= { RCSTMIBCompliances 1 }

RCSTMIBObjGroups OBJECT IDENTIFIER ::= { RCSTMIBConformance 2 }


RCSTSysInstallGroup OBJECT-GROUP
  OBJECTS {
    RCSTSysInstallOwner, RCSTSysInstallDate, RCSTSysInstallLocation,
RCSTSysInstallSSPId, RCSTSysInstallRCSTMacAddr, RCSTSysInstallOduAntennaSize,
RCSTSysInstallOduSSPA, RCSTSysInstallAllowedDomainNextIndex,
RCSTSysInstallAllowedDomainName, RCSTSysInstallAllowedDomainStatus,
RCSTSysInstallCompNextIndex, RCSTSysInstallCompDescrString,
RCSTSysInstallCompDescrStatus
  }
  STATUS current
  DESCRIPTION
    "A collection of objects providing installation information, i.e. the kind
of information defined at commissioning (or when the RCST gets upgraded) by the
installer and that should not change."
  ::= { RCSTMIBObjGroups 1 }


RCSTSysIduGroup OBJECT-GROUP
  OBJECTS {
    RCSTSysIduShutTxAGCRateThresh, RCSTSysIduShutTxAGCWindow,
RCSTSysTargetEbN0, RCSTSysLoopGainD, RCSTSysEbN0Range
  }
  STATUS current
  DESCRIPTION
    "A collection of objects related to the ULPC algorithm as well as to the
  IDU."
  ::= { RCSTMIBObjGroups 2 }


RCSTConfigNetworkGroup OBJECT-GROUP
  OBJECTS {
    RCSTConfigNetworkOamIpAddr, RCSTConfigNetworkOamIpNetworkMask,
RCSTConfigNetworkTrfIpAddr, RCSTConfigNetworkDefaultTrfIpAddr,
RCSTConfigNetworkTrfIpAddrAssign, RCSTConfigNetworkTrfIpNetworkMask,
RCSTConfigNetworkDefaultTrfIpNetworkMask, RCSTConfigNetworkOamReleaseTimeout,
RCSTConfigNetworkOamIpAddrAssign, RCSTConfigNetworkAuthPriIpAddr,
RCSTConfigNetworkAuthSecIpAddr
  }
  STATUS current
  DESCRIPTION
    "A collection of objects that defines somehow the network configuration of
the RCST."
  ::= { RCSTMIBObjGroups 3 }


RCSTConfigAccessPolicyGroup OBJECT-GROUP
  OBJECTS {
    RCSTConfigAccessPolicyNextIndex, RCSTConfigAccessPolicyIpAddr,
RCSTConfigAccessPolicyNetMask, RCSTConfigAccessPolicyCommunityName,
RCSTConfigAccessPolicyEntity, RCSTConfigAccessPolicyStatus,
RCSTConfigAccessPolMibViewNextIndex, RCSTConfigAccessPolMibViewCommunityName,
RCSTConfigAccessPolMibViewPrefix, RCSTConfigAccessPolMibViewAccessRight,
RCSTConfigAccessPolMibViewStatus
  }
  STATUS current
  DESCRIPTION
    "A collection of objects that defines the access rights to the RCST MIB as
well as some MIB views according to the different community names that have been
defined."
  ::= { RCSTMIBObjGroups 4 }


RCSTConfigLinesAirIfRtnLkGroup OBJECT-GROUP
  OBJECTS {
    RCSTConfigLinesAirIfRtnLkMaxEirp, RCSTConfigLinesAirIfRtnLkFreqHopRange,
RCSTConfigLinesAirIfRtnLkDefIfLevel, RCSTConfigLinesAirIfRtnLkFirstIfLevel
  }
  STATUS current
  DESCRIPTION
    "A collection of objects related to the Return Link interface of the RCST."
  ::= { RCSTMIBObjGroups 5 }


RCSTConfigLinesAirIfAccessGroup OBJECT-GROUP
  OBJECTS {
    RCSTConfigLinesAirIfAccessMode, RCSTConfigLinesAirIfAccessTinitial,
RCSTConfigLinesAirIfAccessTto, RCSTConfigLinesAirIfAccessTresendMin,
RCSTConfigLinesAirIfAccessTresendMax, RCSTConfigLinesAirIfAccessCscMax,
RCSTConfigLinesAirIfAccessNMax, RCSTConfigLinesAirIfAccessTBackOffN,
RCSTConfigLinesAirIfAccessNetId, RCSTConfigLinesAirIfAccessPopId,
RCSTConfigLinesAirIfAccessStartTranspFreq,
RCSTConfigLinesAirIfAccessStartTranspOrbPos,
RCSTConfigLinesAirIfAccessStartTranspPolar,
RCSTConfigLinesAirIfAccessStartTranspFec,
RCSTConfigLinesAirIfAccessStartTranspSymbRate
  }
  STATUS current
  DESCRIPTION
    "A collection of objects defining some timing about the access procedure of
a RCST to the network. Moreover, the characteristics of the transponder the RCST
should lock on to when granted access are provided in this group."
  ::= { RCSTMIBObjGroups 6 }


RCSTLifeRCSTStatusGroup OBJECT-GROUP
  OBJECTS {
    RCSTLifeRCSTStatusMode, RCSTLifeRCSTStatusCurrent,
RCSTLifeRCSTStatusCurrentTimestamp, RCSTLifeRCSTStatusPrevious,
RCSTLifeRCSTStatusPreviousTimestamp, RCSTLifeRCSTStatusRebootStatus,
RCSTLifeRCSTStatusDownloadStatus, RCSTLifeRCSTStatusCurrentImageId,
RCSTLifeRCSTStatusCurrentImageValidFlag, RCSTLifeRCSTStatusAlternateImageId,
RCSTLifeRCSTStatusAlternateImageValidFlag
  }
  STATUS current
  DESCRIPTION
    "A collection of objects giving some information about the status of the
Satellite Interactive Terminal. The status of the terminal is given with respect
to some major procedure - among other things: download and reboot processes"
  ::= { RCSTMIBObjGroups 7 }


RCSTLifeTrapLogGroup OBJECT-GROUP
  OBJECTS {
    RCSTLifeTrapLogNumber, RCSTLifeTrapLogEventTime
  }
  STATUS current
  DESCRIPTION
    "A collection of objects providing some log of the RCST traps."
  ::= { RCSTMIBObjGroups 8 }


RCSTLifeTrapDestGroup OBJECT-GROUP
  OBJECTS {
    RCSTLifeTrapDestTableNextIndex, RCSTLifeTrapDestEntity,
RCSTLifeTrapDestOid, RCSTLifeTrapDestStatus
  }
  STATUS current
  DESCRIPTION
```

```
    "A collection of objects that defines the destination of each trap sent by
the RCST. Each destination is represented by an entity. Those entities are
related via the access rights definition to the different communities."
  ::= { RCSTMIBObjGroups 9 }


RCSTCallCntlGroup OBJECT-GROUP
  OBJECTS {
    RCSTCallCntlTmRCSTId, RCSTCallCntlAtmConnTableNextIndex,
RCSTCallCntlAtmConnVPI, RCSTCallCntlAtmConnVCI, RCSTCallCntlAtmConnType,
RCSTCallCntlAtmConnCRAMax, RCSTCallCntlAtmConnVBDCMax,
RCSTCallCntlAtmConnStatus, RCSTCallCntlRCSTHoldCommand,
RCSTCallCntlRCSTLogoffConfirm, RCSTCallCntlRCSTLogoffCommand,
RCSTCallCntlUserLogoffTableNextIndex, RCSTCallCntlUserName,
RCSTCallCntlUserHostIpAddr, RCSTCallCntlUserLogoffStatus, RCSTCallCntlUserStatus
  }
  STATUS current
  DESCRIPTION
    "A collection of objects defining some Call Control actions."
  ::= { RCSTMIBObjGroups 10 }


RCSTActionsGroup OBJECT-GROUP
  OBJECTS {
    RCSTActDownloadUrl, RCSTActDownloadOwner, RCSTActDownloadOwnerIpAddr,
RCSTActPingTableNextIndex, RCSTActPingDestination, RCSTActPingFrom,
RCSTActPingSize, RCSTActPingResultTimeStamp, RCSTActPingResult, RCSTActPingTime,
RCSTActPingOwnerIpAddr, RCSTActPingRowStatus, RCSTActRebootStatusTrapFlag,
RCSTActRebootCommand, RCSTActDownloadCommand, RCSTActDropSACommand,
RCSTActBootImage, RCSTActValidateCurrentImageCommand,
RCSTActTrafficReleaseTimeout, RCSTActDefaultDomainName,
RCSTActDefaultDomainStatus
  }
  STATUS current
  DESCRIPTION
    "A collection of objects that defines some actions that could be issued."
  ::= { RCSTMIBObjGroups 11 }


RCSTSysCapabilityGroup OBJECT-GROUP
  OBJECTS {
    RCSTSysCapabilitySecurity, RCSTSysCapabilitySNMP,
RCSTSysCapabilityATMconnectivity, RCSTSysCapabilityMPEGTSTRF,
RCSTSysCapabilityRCSTboards, RCSTSysCapabilityRCSTACQ,
RCSTSysCapabilityMultiIdu, RCSTSysCapabilitySWVersion,
RCSTSysCapabilityFreqHoppRange, RCSTSysCapabilityMFTDMA,
RCSTSysCapabilityRCSTClass, RCSTSysCapabilityRCSTMode, RCSTSysCapabilityReqCRA,
RCSTSysCapabilityReqRBDC, RCSTSysCapabilityReqVBDC, RCSTSysCapabilityReqAVBDC,
RCSTSysCapabilityReqFCA, RCSTSysCapabilityReqDCRA
  }
  STATUS current
  DESCRIPTION
    "A collection of objects related to the capabilities of the RCST, which
shall also be given in the RCST capability field of the CSC burst."
  ::= { RCSTMIBObjGroups 12 }


RCSTCallCntlMpegGroup OBJECT-GROUP
  OBJECTS {
    RCSTCallCntlMpegConnTableNextIndex, RCSTCallCntlMpegConnType,
RCSTCallCntlMpegConnPid, RCSTCallCntlMpegEntryStatus
  }
  STATUS current
  DESCRIPTION
    "A collection of objects defining some Call Control actions for the MPEG
burst option."
  ::= { RCSTMIBObjGroups 13 }


RCSTMIBNotGroups OBJECT IDENTIFIER ::= { RCSTMIBConformance 3 }


RCSTLifeTrapGroup NOTIFICATION-GROUP
  NOTIFICATIONS {
    RCSTLifeTrapRCSTDownloadStatus, RCSTLifeTrapRCSTRebootStatus,
RCSTLifeTrapPingStatus
  }
  STATUS current
  DESCRIPTION
    "A collection of objects defining notifications about major processes
result."
  ::= { RCSTMIBNotGroups 1 }


RCSTCallCntlTrapGroup NOTIFICATION-GROUP
  NOTIFICATIONS {
    RCSTCallCntlTrapTrafficVCCRequest, RCSTCallCntlTrapOamReleaseRequest,
RCSTCallCntlTrapTrafficReleaseRequest, RCSTCallCntlTrapUserLogoffRequest,
    RCSTCallCntlTrapRCSTLogoffRequest
  }
  STATUS current
  DESCRIPTION
    "A collection of objects defining notifications about Call Control actions
result."
  ::= { RCSTMIBNotGroups 2 }


RCSTCallCntlTrapMpegGroup NOTIFICATION-GROUP
  NOTIFICATIONS {
    RCSTCallCntlTrapMpegPidRequest, RCSTCallCntlTrapMpegPidReleaseRequest
  }
  STATUS current
  DESCRIPTION
    "A collection of objects defining notifications about Call Control actions
result for the MPEG burst option."
  ::= { RCSTMIBNotGroups 3 }


END
```

Annex G: Example for a security and authentication concept

G.1 User authentication using RADIUS

This clause describes how users of an RCST can be authenticated by service providers in the satellite interactive network by applying RADIUS, which is specified in [10]. Users of an RCST are persons using an IP host that is connected to the RCST. Typically, a LAN connects one or more IP hosts to the RCST or a host is integrated with the RCST into one unit.

With authentication implemented in a satellite interactive network it is possible to restrict MAC layer capacity assignments or connection to other networks to RCSTs with authenticated users. In particular, users can authenticate to a service provider that provides them with connection to the Internet and in addition carries out the billing for the use of satellite interactive network resources. It is possible that different users behind an RCST authenticate to different service providers.

The RCST runs a web server that allows users to initiate authentication by means of a web browser running on their host. In cases where the RCST shall authenticate automatically without involvement of a user and a host, a static user can be configured on it. The user authentication as described here requires SNMP and the RCST MIB as defined in Clause 8.5.

G.1.1 User Authentication process

The RCST receives a login request from the user on a host behind the RCST, or the RCST generates a login request for the Static Users, which are defined below. The User is identified by user id and password and optionally by a domain name. The domain name indicates the entity that does the authentication. In particular this can be a service provider. If a domain name is not provided, the RCST will append one. After the RCST gets the user information, it passes the user information to the NCC. NCC authenticates the user with the appropriate service provider. If its service provider accepts the user and the NCC is able to allocate resources to the RCST, the NCC allocates the RCST traffic resources of the CRA category and sends an access-accept message to the RCST. If the user is rejected by the service provider or the NCC is unable to allocate resources to the RCST, the NCC informs the RCST and sends an access-reject message to the RCST containing the reason for access rejection. The user authentication sub-system shall provide the following:

> The RCST shall implement a lightweight NAS module, which is called RADIUS User Client in the following, as a part of the RCST software offering. Modifications to standard RADIUS apply only to the link segment between the RADIUS User Client of the RCST and a specific device at the NCC, which is called RADIUS Client in the following. Standard RADIUS is used between the RADIUS Client and a RADIUS server or RADIUS proxy. Therefore, the solution interworks with standard radius servers of ISPs.
>
> The NCC shall generate a "Random Number". This Random Number shall be given to the RCST through the field "Random Number" of the Network Layer Info Descriptor. The Network Layer Info Descriptor is contained in the TIM message that the NCC sends as a reply to a CSC burst. The RCST shall retain the Random Number and use it for all user authentications for the duration of a return link acquisition.
>
> The random number is used in a modified version of CHAP. The reason of circumventing standard CHAP is to require one round-trip less. Therefore, the user access to the network is shortened by about 520 ms in the case of geo-stationary satellites.
>
> The same Random Number shall be used for all Access_Request messages for the duration of a return link acquisition.
>
> If the NCC sends an Access_Accept message to the RCST, then the RCST sends an "authenticated" message to the user. The user can then start sending traffic. At this stage the RCST will pass all traffic from the host this User authenticated from or from all hosts if there is a static user entry.
>
> RADIUS Challenge is not supported. If an ISP sends an Access_Challenge message, then the RADIUS client transforms it into an Access_Reject.

The user remains authenticated until it explicitly logs off the network. When RCST receives the log off message from the user, it informs the NCC Traffic Manager through a user log-off SNMP Trap message. The NCC cleans-up its resources and responds back to the RCST using SNMP Set Message.

The NCC RADIUS Client shall ignore subsequent Access_Request message from the same user while the authentication is in progress.

G.1.2 User Authentication Message Flow and Steps

G.1.2.1 User Authentication Accept Message Flow and Steps

Figure G-1 illustrates the user authentication process for the case that the user is accepted.

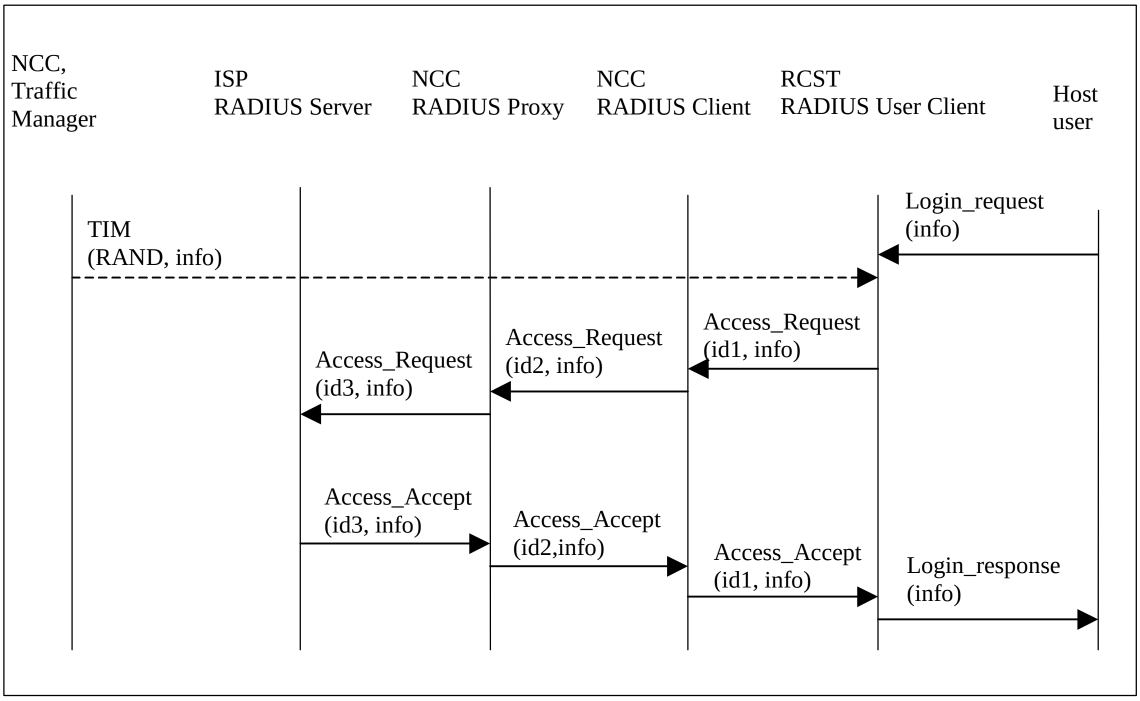


Figure G-1: User Authentication Accept

The following steps describe the user authentication:

The user provides his login credentials i.e. User id, password to the RCST. RCST starts its Return Link acquisition process.

During the process of the return link acquisition, the RCST receives a TIM message that contains a random number in the Network Layer Info Descriptor and some other parameters. The random number is used to generate the user CHAP password.

On receiving the login credentials, the RCST examines the request. The RCST uses the random number that it received in the TIM to encrypt the user password in its first Access_Request to the RADIUS Client at the NCC.

The NCC RADIUS Client forms a RADIUS Access_Request [10] message to the RADIUS Proxy.

RADIUS Proxy checks the user name / ID and determines if the user should be authenticated locally or user authentication should be extended to a service provider. The User Name / ID format defines the service provider identity.

If required, the RADIUS Access_Request message is extended to the user's service provider.

The service provider's RADIUS Server responses with the Access_ Accept Message.

The RADIUS Proxy extends the response to the RADIUS Client.

The NCC RADIUS Client performs its IP resourcing, and sends an Access_Accept to the RCST.

The RCST informs the user about the successful authentication.

G.1.2.2 User Authentication Reject Message Flow and Steps

Figure G-2 illustrates the user authentication process for the case that the user is rejected.

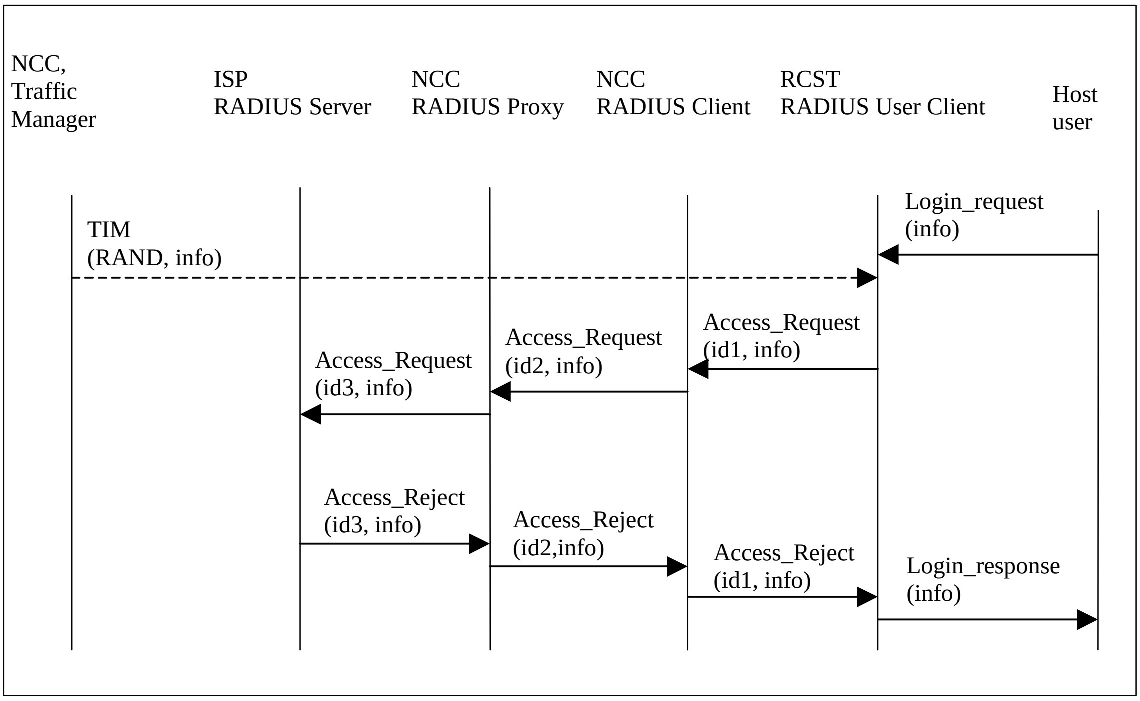


Figure G-2: User Authentication Reject

The following steps describe the subsequent user authentication reject:

The user provides his login credentials i.e. User id, password to the RCST. RCST starts its Return Link acquisition process.

During the process of the return link acquisition, the RCST receives a TIM message that contains a random number and some other parameters. The random number is used to generate the user CHAP password.

On receiving the login credentials, the RCST examines the request. RCST uses the session random number to encrypt the user CHAP password in its Access_Request to the NCC.

The NCC RADIUS Client forms a RADIUS Access_Request [10] message to the RADIUS Proxy.

The RADIUS Proxy checks the user name / ID and determines if the user should be authenticated locally, or user authentication should be extended to a service provider. The user Name / ID format defines the service provider identity.

If required, the RADIUS Access_Request message is extended to the user service provider.

The service provider's RADIUS Server responses with the Access_Reject message.

RADIUS Proxy extends the response to the RADIUS Client.

The NCC RADIUS Client extends an Access_Reject to the RCST.

RCST informs user of failure.

G.1.2.3 User Authentication Service provider Challenge Message Flow and Steps

Figure G-3 illustrates the authentication process for the case that the service provider replies with a challenge.

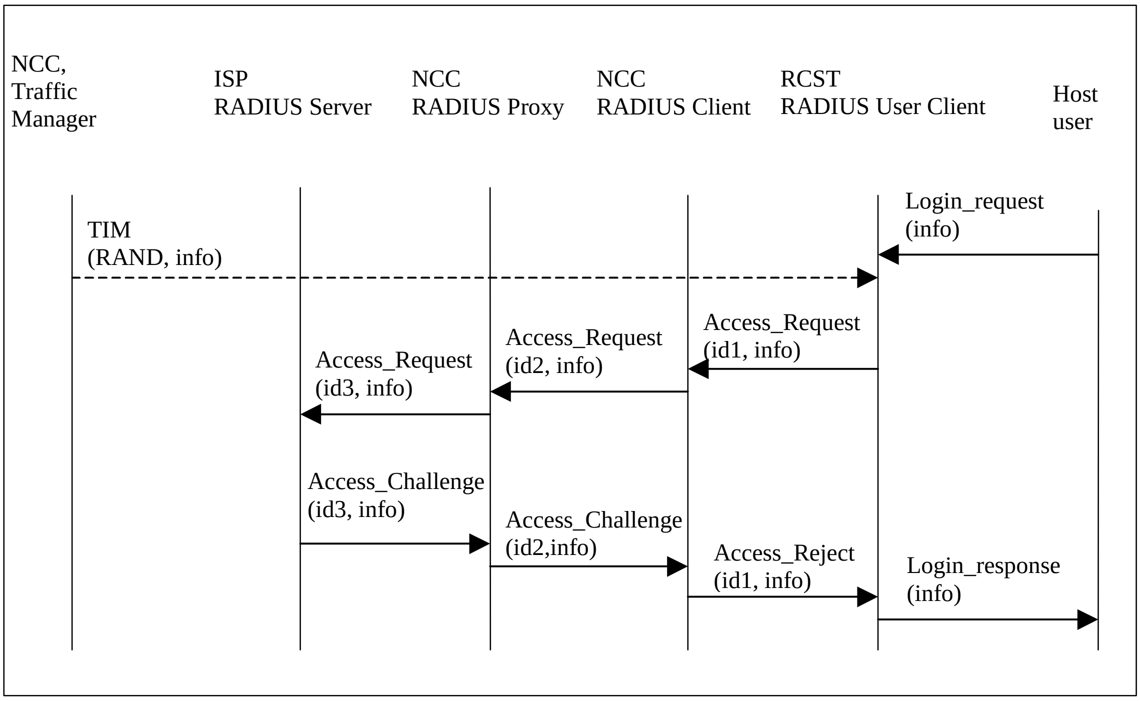


Figure G-3: User Authentication Challenge

The following steps describe the subsequent user authentication reject when it is challenged by service provider:

The user provides his login credentials i.e. User id, password to the RCST. RCST starts its Return Link acquisition process.

During the process of the return link acquisition a TIM message that contains a random number and some other parameters. The random number is used to generate the user CHAP password.

On receiving the login credentials, the RCST examines the request. RCST uses the session random number to encrypt the user CHAP password in its Access_Request to the NCC.

The NCC RADIUS Client forms a RADIUS Access_Request [10] message to the RADIUS Proxy.

The RADIUS Proxy checks the user name / ID and determines if the user should be authenticated locally or user authentication should be extended to a service provider. The User Name / ID format defines the service provider identity.

If required, the RADIUS Access_Request message is extended to the user service provider.

The service provider RADIUS Server responses with the Access_Challenge message.

The NCC RADIUS Client forms an Access_Reject and embeds the rejection reason in the messages to the RCST.

RCST informs user of failure.

G.1.3 User Authentication Message Format

The message format complies with [10]. Password hiding is not supported. There is no "secret" between RCST and NCC.

Table G.1 shows the attributes that can be used with the different types of RADIUS messages, which are identified by the code field. The relevant messages are described in the following.

Table G.1: The attributes that can be used with the different types of RADIUS messages

Type	Name	Value	CODES		Comments
			Mandatory	Optional	
1	User-Name	User ID	1	2,3	
3	CHAP_Password	CHAP ID (1 Octet) + CHAP response (16 Octets)	1		CHAP ID is an RCST generated random number less than 256. The CHAP response is created by MD5(CHAP_ID + user_password + CHAP_Challenge)
4	NAS IP	Host IP address	1	2,3	This deviates from [RFC 2865]
5	NAS Port	Host port id		1,2,3	This deviates from [RFC 2865]
18	Reply_Message	String	3	2	Information to be display to the user
32	NAS Id	RCST MAC Address	1	2,3	
60	CHAP-Challenge	CHAP (16 bits) Random number from Radius Client	1		NCC Radius Client generated random number, transmitted in a TIM during logon to the RCST

G.1.3.1 Access_Request for User

On receiving the login request from the user, the RCST retrieves the Radius Client generated random number (received with the TIM message) and forms an Access_Request Message of the following fields:

Code = 1

Identifier: RCST selected unique transaction identifier. It remains the same through out the protocol negotiation until the user authentication is complete.

Length: as defined in [10].

Request Authenticator: RCST generated random number, called the Request Authenticator.

Attribute: As described in Table G.1. Attribute types 1, 3, 4, 32 and 60 are mandatory. Attribute type 5 is optional.

As a reply the RCST will receive either an Access_Accept or an Accept_Reject message from Radius Client from the NCC.

G.1.3.2 Access_ Reject

When the RADIUS Client receives a service provider challenge or reject message, the Radius Client forms an Access_Reject and sends it to RCST. The Access_Reject Message of the following fields:

Code = 3

Identifier: RCST selected unique transaction identifier. It remains the same through out the protocol negotiation until the user authentication is complete.

Length: as defined in [10].

Response Authenticator: MD5 of (Code + Identifier + Length + Request Authenticator (from access request) +Attributes) where + denotes concatenation.

Attribute: As described in Table G.1 Attribute type 18 is mandatory. Attribute types 1, 4, 5 and 32 are optional.

The Reply Message string of attribute 18 consists of two fields. First field indicates the reject code in the form of two-character code. The code is followed by space (one character) and by string holding the textual message. The Reply_Message string is terminated by null character.

Reject Codes (First two bytes):

Code 00 – reason unknown

Code 01 – Service provider RADIUS Server Reject (or challenge)

Code 02 – NCC generated a reject

Code 08 – reason unknown and Information to be displayed to the user starting at fourth byte of string

Code 09 – Service provider RADIUS Server Reject (or challenge) and Information to be displayed to the user starting at fourth byte of string stating "Authentication Failed RADIUS"

Code 10 – NCC generated a reject and Information to be displayed to the user starting at fourth byte of string stating "AUTHORIZATION Failed NCC"

G.1.3.3 Access_ Accept

On receiving Access-Accept the from the RADIUS Proxy, The RADIUS Client forms an Access_Accept Message and sends it to RCST. The Access_Accept message contains the following fields:

Code = 2

Identifier: RCST selected unique transaction identifier. It remains the same through out the protocol negotiation until the user authentication is complete.

Length: as defined in [10].

Response Authenticator: MD5 of (Code + Identifier + Length + Request Authenticator (from access request) +Attributes) where + denotes concatenation.

Attribute: As described in Table G.1 none of the attribute types is mandatory. Attribute types 1, 4, 5, 18 and 32 are optional.

G.1.4 User Authentication Table

The user authentication table is a table maintained at the RCST to keep track of the authentication related information of all users (normal and static) that are logged into the RCST. A User's authentication status is dependent on the authentication process. The following applies to users and the user authentication status.

This User Authentication Table is a representation of the parameters and should not be interpreted a design constraint. Implementations of the table may vary across RCSTs.

Table G.2: User Authentication Table

Username	Password	IP Address	User Type	State	Attributes
Any valid username	Any valid locally CHAP password	Any valid Host IP address	Normal, Static	NOT AUTHENTICATED, AUTHENTICATION REQUESTED, AUTHENTICATED	Automatic Authentication

Username:

A character string used to uniquely identify a User within a domain. A user name can be qualified (such as <user>@<domain>).

The RCST shall provide the following functionality:

Default @domain

The Superuser can enable/disable this mandatory feature and set the default @domain name. The default @domain name will be permanently stored in the RCST. The RCST will append the default @domain name to all unqualified logins.

Enforced @domain

This optional feature gives the possibility for the RCST to reject logins to any domain not specified as allowable in the RCST without sending a user login request to the NCC.

The installer is the only entity that can enable/disable this functionality and enter/modify the enforced @domain name. The enforced @domain name will be permanently stored in the RCST.

This concept can be extended to a list of authorized @domains (instead of a unique enforced @domain).

These two features can be run simultaneously on the same RCST. The @domain enforcement and default domain features must then be enabled and the default @domain name must match the enforced @domain name.

Password:

Any character string, maximum 64 8-bit ASCII characters.

IP Address:

IP address of the host from which the user is logging in

User Type:

Normal User:

All users who login from host machines behind the RCST using the HTML web pages located in the RCST.

Static User:

A Static User Entry is an entry in the RCST User Authentication Table that persists across RCST power cycles.

Only one entry exists in the User Authentication Table as a Static user entry.

Only a Superuser may define, remove or modify a Static User Entry. It is never removed from the User Authentication Table as part of the RCST system processes.

Static User entry has the Automatic Authentication attribute set upon creation.

If the RCST supports static user entry, then it blocks normal user logins (not Superuser). That is, the Static user and Normal users cannot exist together.

Removing a Static User and Adding Normal Users:

The Superuser removes the Static User from the User Authentication Table.

Normal users are added to the User Authentication Table as they login.

NOTE 1: The Superuser shall force the user log off procedure when removing users of any type.

Removing Normal Users and Adding Static User:

The Superuser removes all Normal User from the User Authentication Table.

After all Normal Users are removed from the User Authentication Table, the Superuser adds the Static User to the User Authentication Table.

NOTE 2: The Superuser shall force the user log off procedure when removing users of any type.

States:

All table entries will be in one of the following states:

AUTHENTICATION REQUESTED

AUTHENTICATED

NOT AUTHENTICATED

AUTHENTICATION REQUESTED:

The user is AUTHENTICATION REQUESTED when the RCST starts the authentication process for the user until an authentication response from the NCC is received or authentication timeout expires.

AUTHENTICATED:

The user is AUTHENTICATED upon receiving the accept message from the NCC. At this point this user will also have the Automatic Authentication attribute set.

NOT AUTHENTICATED:

All users in the table for whom an authentication process has not been started.

NOTE 3: This includes static users for whom the authentication fails.

EXAMPLE 1: If the RCST is in the INITIALIZED state, a normal user entry may be in the NOT AUTHENTICATED state with the Automatic Authentication attribute set (as in the case when the normal user was authenticated at the time the RCST logs off the network).

EXAMPLE 2: If the RCST is in the INITIALIZED state, the static user will be in the NOT AUTHENTICATED state with the Automatic Authentication attribute set.

Attributes:

AUTOMATIC AUTHENTICATION ATTRIBUTE

Automatic Authentication Attribute for Normal Users

When set, this attribute will trigger automatic authentication of a user entry in the UAT.

The purpose of this attribute is to signal the RCST about an automatic user authentication procedure upon the RCST's return link acquisition.

Automatic Authentication attribute for all normal users are initially defaulted to False.

The Automatic Authentication attribute is set when the user reaches the AUTHENTICATED state. This attribute remains set until the user entry is removed from the User Authentication table.

Automatic Authentication Attribute for Static Users

Automatic Authentication attribute for a static user is set at the time when a static user is created and remain until the entry is removed.

Other User Authentication Table Properties:

The User Authentication Table in the RCST can contain either

a single static entry or

one or more normal (non-static) entries

A normal user is removed from the User Authentication Table when:

The RCST transitions to the HOLD state.

The User logs off (After the RCST receives the user log-off response from the NCC)

Superuser removes the user entry from the User Authentication Table.

The User fails authentication.

Power is removed from the RCST or the RCST is reset.

A Static User is removed from the User Authentication Table when:

The Superuser removes the static user entry from the User Authentication Table.

G.1.5 CHAP password Crypto Engine

To hide the password, this authentication process uses a method based on the RSA Message Digest Algorithm MD5. The algorithm takes as input a message of arbitrary length and produces as output a 128-bit "fingerprint" or "message digest" of the input. It is conjectured that it is computationally infeasible to produce two messages having the same message digest, or to produce any message having a given pre-specified target message digest. The MD5 algorithm is intended for digital signature applications.

The MD5 algorithm is designed to be quite fast on 32-bit machines. In addition, the MD5 algorithm does not require any large substitution tables; the algorithm can be coded quite compactly. Please refer to [24] for more detail on MD5.

RCST MD5 crypto Engine takes in three variables: user password, RCST generated CHAP ID, and NCC generated Random numbers and produces CHAP Password as output called "message digest" of the inputs. Details are specified in Clause G.1.3.

G.2 IPSec Solution and Definition

This clause describes an implementation of IPSec on DVB-RCS networks, together with parameter and option choices that are useful for this kind of networks. The objective is to provide to the end user default security on the satellite link and to prevent unauthorized eavesdropping of traffic. There are cases where the provision of such security is a legal requirement for network operators. It must be implemented in the RCST and the NCC, because they belong to the area that is controlled by the network operator. An implementation between two hosts or between a host and a service provider would not fulfil this requirement, but is possible in addition to the implementation described here. Security is also intended to protect the user's identity; including his exact location, signalling to and from the user, and the data traffic to and from the user. IPSec shall protect some user identity such as IP address. Other user identity protection such as user authentication is defined elsewhere in the present document. The NCC shall implement IPSec functionality though a security gateway, which is defined in RFC 2401 [25].

The RCSTs shall implement IPSec software/hardware and function as peers to the NCC security gateway. The NCC security gateway policies shall be statically configured to protect all networks behind the RCST with the RCST traffic IP addresses as its peer. Since a peer has one or more hosts behind it, the NCC security gateway is configured to be aware of the subnet behind each RCST. The peer configuration is not updateable on a real time basis.

The NCC secure tunnel, IKE SA, set-up is bi-directional. The RCST or the NCC security gateway can initiate the SA negotiation. The IPSec tunnel is set up between the NCC security gateway and RCSTs traffic interface. Once the secure tunnel is set up, only user traffic shall be encrypted while OAM data shall not be encrypted.

To summarize, IPSec sub-system shall provide the following:

- Security gateway with tunnel mode to establish secure-connections from an RCST to the NCC.
- RCSTs that have IPSec software/hardware are peers to NCC security gateway.
- Support for Encapsulated Secure Payload (ESP), and single security association (SA) between RCST and NCC.
- The IPSec tunnel is established between an RCST and the NCC when RCST has traffic resources (traffic VCC), and the NCC and the RCST exchange user traffic.
- RCST traffic data is encrypted.

- RCST OAM data, i.e. SNMP using the RCST MIB, is not encrypted. The reason for not encrypting it is that building a tunnel before the RCST can start authenticating users would slow down the login process significantly because of the long roundtrip delay in conjunction with geo-stationary satellites.
- Peer to the NCC security gateway secure tunnels are statically configured.
- The SA re-negotiation life is configurable at both ends (security gateway, RCST). Negotiation re-keying time is determined by the shorter of the two configuration values.
- Both RCST and NCC must be IPSec interoperable.
- Both RCST and NCC shall use same compression definition.

G.2.1 SA negotiation and Secure Tunnel setup

A security association (SA) is a set of policy and key(s) used to protect information. The IKE SA is the shared policy and key(s) used by the negotiating peers in this protocol to protect their communication.

The SA negotiation takes place between the RCST and the NCC security gateway. The IP packets must have source address (RCST traffic IP address) and the destination address (the NCC security gateway IP address). All packets that use the RCST traffic IP address, as their source must be tunnelled to the NCC IPSec gateway.

SA Negotiation is considered IP traffic on traffic VCC, because of the relation between IPSec and the separation of control and traffic data within the constraints of the IP Topology. Specifically, all control data are filtered to go to the Control LAN. The traffic data is filtered to go to the security gateway. Hence, tunnel establishments must result in this split, regardless of what VCC is used.

In order to do SA negotiation/re-negotiation, the RCST must be in the Active state. This means, the Traffic resources (namely Traffic VCC) have been allocated to RCST.

The RCST shall initiate the IPSec negotiation when the first user is authenticated and the RCST traffic resources are allocated. RCST shall undergo a state transition from OAM Active to Active state and initiate a secure association (SA) set-up. The IPSec negotiation shall take place over the traffic channel and shall be destined to the NCC security gateway using traffic IP address. Thereafter all hosts traffic data shall be tunnelled to the NCC security gateway. To summarize the steps:

1) 1st user is authenticated successfully (Using OAM VCC).
2) RCST acquires Traffic resources.
3) RCST goes into Active state.
4) RCST starts IPSec negotiation (Using Traffic VCC).

Similarly, the NCC security gateway can initiate the IPSec negotiation, when data arrives from the terrestrial network and is destined to the host behind the RCST. It is RCST responsibility to reject or accept the dialog and negotiate back to the completion of SA set-up.

Once the secure tunnel is set up, only user traffic shall be encrypted. Therefore, RCST shall separate the user traffic from the OAM data. User traffic shall be encrypted, and OAM data shall not be encrypted.

G.2.2 RCST SA re-negotiation

The SA re-negotiation life shall be configurable at both ends (security gateway, RCST). Negotiation re-keying time is determined by the shorter of the two configuration values. If the RCST SA time-to-live expires, the RCST shall re-negotiate the key. If the NCC security gateway SA time-to-live expires, the NCC starts re-negotiations. The re-negotiation to completion is determined by the RCST. RCSTs shall only re-negotiate SA when they have their traffic resource; therefore, RCSTs shall not acquire traffic VCC to re-negotiate their SA.

G.2.3 RCST Wakeup SA Negotiation

Traffic Wakeup is initiated in the following manner. RCST is in the Initialized state with one or more users in the previously authenticated state. A return link acquisition shall be triggered by the presence of unsent traffic in the RCSTs transmit buffer. Since the forward link is never surrendered, transmit buffers may get filled in response to a message from the NCC that was sent to an entity behind the RCST. Upon sensing unsent traffic the RCST goes through the OAM Acquisition and Traffic Acquisition procedure described in 8.5.3. Subsequent to the access request, the RCST would obtain the traffic VCC and then start the IP Sec tunnel establishment. An IPSec tunnel need be established only if a Secure Association does not exist or has expired. After the IPSec tunnel is established the traffic is sent as encrypted information directly to the security gateway.

G.2.3.1 RCST Interfaces

RCSTs shall statically be configured as peers to the NCC security gateway. The IP Address used for the NCC security gateway IP address will be the end-point at which the RCSTs terminate the IPSec tunnel. Encryption shall be enabled to all traffic ports (if more than one).

G.2.4 Redundancy

The peer Network (RCSTs) shall support automatic roll-over from primary to secondary when primary security gateway failure. The roll-over timer shall be configurable.

G.3 RCST Security Requirements

This clause describes basic security requirements for an RCST, which enable a secure overall network. The RCST security should provide mechanisms for:

- Protection against violation
- Detection of violation
- Containment of violation
- Recovery from violation

G.3.1 Architecture Overview

The RCST should implement realms of access. A realm is defined as an area of functionality. Inside a particular realm, only certain actions may be taken and certain data accessed. The RCST implements six realms:

- Air Interface
- NCC
- Installer
- Superuser
- User
- Service

G.3.2 Protection Against Violation

The RCST should provide protection against:

- Loss of Privacy – reading of information by unauthorized individuals.
- Loss of Data – the corruption or erasure of information.

Protection should be realized using a Login and Password mechanism.

Users provide login and password Identification to authenticate to the RCST. Factory defaults to no user accounts on the system.

The Superuser provides login and password identification to authenticate to the RCST. The Superuser account and password should be factory defaulted to predetermined values.

Installers have a pre-determined username and password combination to access the Installer Session Access screen.

If the RCST provides a web server for communication with the Users, Superuser and Installer , then Login sessions should take place using secure HTTP.

Installer Session Access - after installers have logged into the RCST with username and password they will be given a screen that presents the installer with the MAC address of the RCST and a Session Challenge. The same screen will ask for a session ID. The installer will take the MAC address, Session Challenge and their installer ID and then either call the network operator, use a special program or use special hardware to generate a session ID. The installer will then enter the session ID into the RCST. This will give access to the RCST for a particular session.

User Traffic Protection - no provision is made for user traffic protection, however RCST installation manuals should recommend the use of switched Ethernet to provide a measure of traffic protection.

G.3.3 Containment of Violation

Violations should be contained using the following mechanism:

Security Realms - RCST access is separated into five realms. A clear separation must be provided between security realms. Actions in each realm should be limited to those appropriate for those realms. Data access should be restricted for each realm as appropriate. Appropriate access for each realm to the SNMP MIB objects is defined as part of the MIB definition in Clause 8.5. Transition from one realm to another should not be permitted.

G.3.4 Recovery from Violation

The NCC may be able to assist recovery from certain violations, however, some violations may require installer interaction.

It is strongly recommended that the RCST have the capability to display all super-user configurable parameters on a single screen, so that the super-user can keep a back up of configurable parameters.

Annex H:
Example of Authentication of RCST users with RADIUS

H.1 Introduction

This annex gives an example on how to use the RADIUS protocol [10] to authenticate RCST users. For user authentication the RCST will act as a Network Access Server, with a RADIUS client forwarding user information to the RADIUS server. The RADIUS server is located at the NCC or at the ISP.

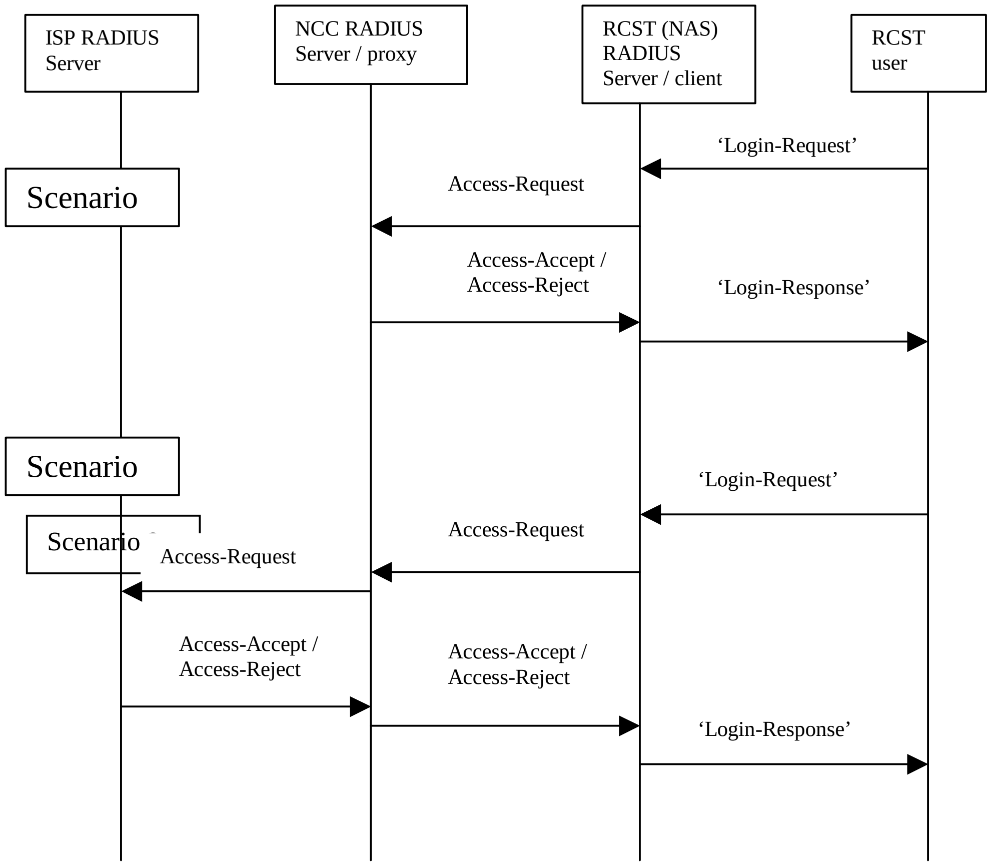
When authentication is performed at the ISP the RADIUS server at the NCC will act as a proxy RADIUS server. Whether the NCC or the ISP is responsible for authentication depends on the User-Name attribute in the Access Request packet. The User-Name attribute will be formatted as a Network Access Identifier according to [26]. The RADIUS client at the RCST may support various authentication protocols like PPP PAP, PPP CHAP, and customized login prompts.

The present document gives the minimum number of attributes that must be supported by the terminal for authentication with the NCC / ISP.

H.2 Authentication scenarios

The RADIUS server at the NCC supports the following authentication scenarios, found suitable for satellite communication.

Authentication supporting PPP PAP or customized login prompt:



Scenario 1:

The user is authenticated by the NCC, the RCST uses MD5 (shared secret and Request Authenticator) to encrypt the password in the User-Password attribute of the Access Request. The User-Name attribute of the Access Request message is a network access identifier (26) with the default domain name (NCC realm).

Scenario 2:

The user is authenticated by the ISP. The network access identifier in the User-Name attribute must contain the domain name of the ISP to which the Access Request shall be forwarded. (ISP realm).The domain name used in the User-Name attribute shall be authenticated by the NCC RADIUS server (identified by the NCC as a legal ISP for that user), the username shall be authenticated by the ISP. At the NCC the User Password must be decrypted (using the secret shared between the NCC and the RCST) and encrypted using the secret shared between the NCC and the ISP before the request is forwarded.

H.3 Authentication attributes for scenarios 1 and 2.

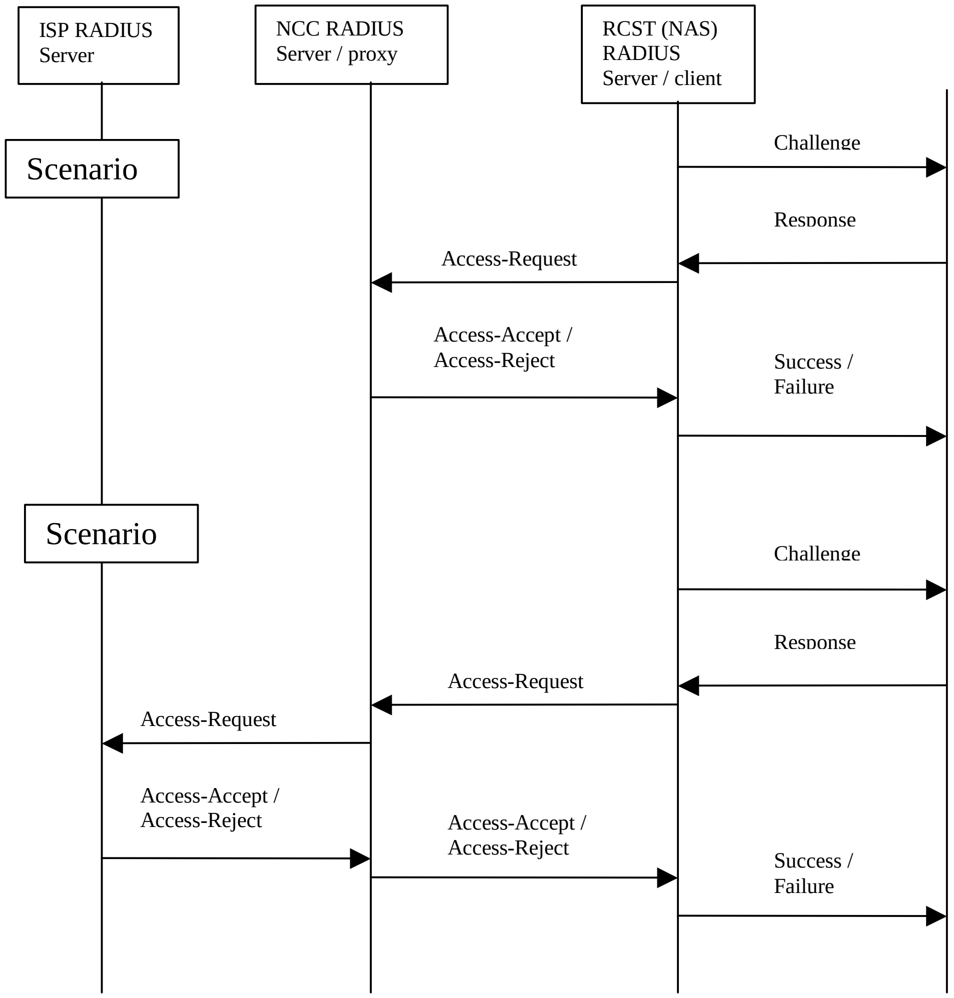
The following attributes must be provided as a minimum:

Type	Name	Value	Codes	Comments
1	User-Name	Username@realm (RFC 2486 [26])	1,2	The default realm (NCC realm) must be provided by the RCST, if the realm is not given from the user.
2	User-Password		1	Decrypted and encrypted by RADIUS server if proxy.
32	NAS-Identifier	RCST MAC address	1	
18	Reply-Message		3	
27	Session-Timeout		2	
28	Idle-Timeout		2	

Access_Request = code 1, Access_accept = code2. Access Reject = code 3

Type refer to type of attribute as defined in [10]

Authentication supporting PPP CHAP:


ISP RADIUS Server
NCC RADIUS Server / proxy
RCST (NAS) RADIUS Server / client
Scenario
Challenge
Response
Access-Request
Access-Accept / Access-Reject
Success / Failure
Scenario
Challenge
Response
Access-Request
Access-Request
Access-Accept / Access-Reject
Access-Accept / Access-Reject
Success / Failure

Scenario 3:

The CHAP password attribute used in the Access Request carries the CHAP ID + the 16 octet CHAP response returned by the PPP user. MD5 (CHAP Identifier + user password + challenge). The CHAP Challenge generated by the RCST is transmitted in the CHAP-Challenge attribute. The RADIUS server computes its own response value MD5(CHAP Identifier + user password (from local database) + CHAP Challenge) and compares the result with the CHAP Password.

Scenario 4:

This scenario is equal to scenario 3 but the authentication is performed at the ISP. The NCC performs authentication of the domain (realm) part of the User-Name attribute and the ISP authenticates the username.

Authentication attributes for scenarios 3 and 4.

Type	Name	Value	Codes	Comments
1	User-Name	Username@realm (RFC 2486 [26])	1,2	The default realm (NCC realm) must be provided by the RCST, if the realm is not given from the user.
3	CHAP-Password		1	
32	NAS-Identifier	RCST MAC address	1	
18	Reply-Message		3	
60	CHAP –Challenge		1	
27	Session-Timeout		2	
28	Idle-Timeout		2	

Access_Request = code 1, Access_accept = code2. Access Reject = code 3

Type refer to type of attribute as defined in RFC 2865 [10].

H.4 Re-Authentication

The RADIUS attributes Session-Timeout and Idle-Timeout will be used by the RADIUS server at the NCC/ISP to inform the RCST of the maximum duration of the authenticated session, and the idle period allowed before the RADIUS client at the RCST should either initiate re-authentication or terminate the session.

Annex I: Example for Procedures and Operations providing additional functionality

I.1 RCST software download

This clause describes a procedure for downloading operating software to RCSTs. The procedure is based on FTP and therefore data must be transmitted to each individual RCST. Therefore, it is appropriate for individual downloads and for download to a small number of terminals. A procedure based on data broadcast, which is the preferred method for a large number of terminals, is not defined here, because a standardized method for download to set-top boxes can be adopted when it is available.

The procedure makes use of SNMP and requires implementation of the RCST MIB as defined in Clause 8.5. An RCST image will be distributed to the RCST in one binary file by the NCC. The RCST does not have to maintain one binary file for the image, but does have to provide an imageID upon request by the NCC. Both the structure and imageID of each image is RCST vendor specific.

The RCST is required to maintain 2 images of the software. The images are referred to as the current and alternate image. The current image is the image the RCST is running. The alternate image is the other image and is not running. The RCST software download process can only update the alternate image.

A new image has to pass two validation stages. The download (checksum) validation is done after the FTP transfer of the new image. The software validation is accomplished via a three-message handshake between the NCC and RCSTs using a Trap and SNMP Set Request/SNMP Response after the new image is booted by the RCST. An RCST will always have at least one validated image. This is so that an RCST can recover from a failed software upgrade by rebooting back to an image that runs correctly and has been validated already.

The RCST Software Upgrade involves the use of three separate processes, "RCST Software Download From The NCC", "Boot Alternate Image", and "RCST Reboot". The "RCST Software Download From The NCC" process is the FTP transfer of a new alternate image to the RCST. The "Boot Alternate Image" command instructs the RCST to boot the alternate image upon the next reboot. The "RCST Reboot" is the process of having the RCST reboot and checking to see if the alternate image is to be booted or if the RCST should just reboot the current image.

Having three separate processes allows for a time delay between the download of the new image, the command to boot the alternate image, and the actual reboot. Also, having the Boot Alternate Image command separate from the RCST Reboot process, allows the alternate image to be in place for the next reboot, which does not have to be initiated by the NCC.

The RCST software download also provides the mechanism to "downgrade" to the previous version of software. The downgrade would involve the use of the "Boot Alternate Image", and "RCST Reboot" processes. Extra logic is placed in the "RCST Reboot" process to allow the RCST to reboot correctly to the alternative image when both are already validated.

Every software download should be co-ordinated between terminal manufacturer, network operator, service provider, terminal owner and user. In cases where new software improves the terminal by extending functionality and increasing performance, it is appropriate that the user or owner decides whether he wants a software download. Their influence must be limited in cases where the new software has an impact on network operation. In both cases the software download should be made at a point in time that is convenient for the user. This co-ordination is not part of the software download specification given here. It can be done by existing means, for example e-mail, before the NCC initiates the download procedure.

I.1.1 RCST Software Download From The NCC

For the software download, the FTP server is located at the NCC, the RCST is the FTP client and the transfer shall be made in binary mode.

Before the download begins the NCC verifies the checksum of the image to be sent.

1) The NCC sends an SNMP Set Request containing a URL for the target protocol/server/filename that contains the following information:

2) ftp://<ftp username>:<ftp password>@<IP address of FTP server>/<full path and filename>

3) The NCC sends an SNMP Set Request to tell the RCST to begin the download.

4) The RCST initiates the FTP session.

5) Upon completion of the download the RCST terminates the FTP session.

6) The RCST does a download validation on the image.

7) Upon successful download validation the RCST populates RCST MIB with the information from the image.

8) The RCST sends an SNMP trap, Download Status, to the NCC providing information on success or failure of the download.

9) If the NCC does not receive an SNMP Trap from the RCST for a specified period after the "download begin" command is issued, the NCC assumes the download failed.

10) If the download process fails, the NCC sends an SNMP trap to the NCC.

I.1.2 Boot Alternate Image

NCC sends an SNMP Set Request, Boot Alternate Image message to the RCST to instruct the RCST to boot from the alternate RCST image on the next boot.

I.1.3 RCST Reboot

NCC sends the RCST an SNMP Set Request commanding the RCST to reboot. The NCC expects a SNMP RESPONSE. The NCC then marks the boot status as UNKNOWN.

The RCST Reboot process is depicted in the flowchart in Figure I.1.

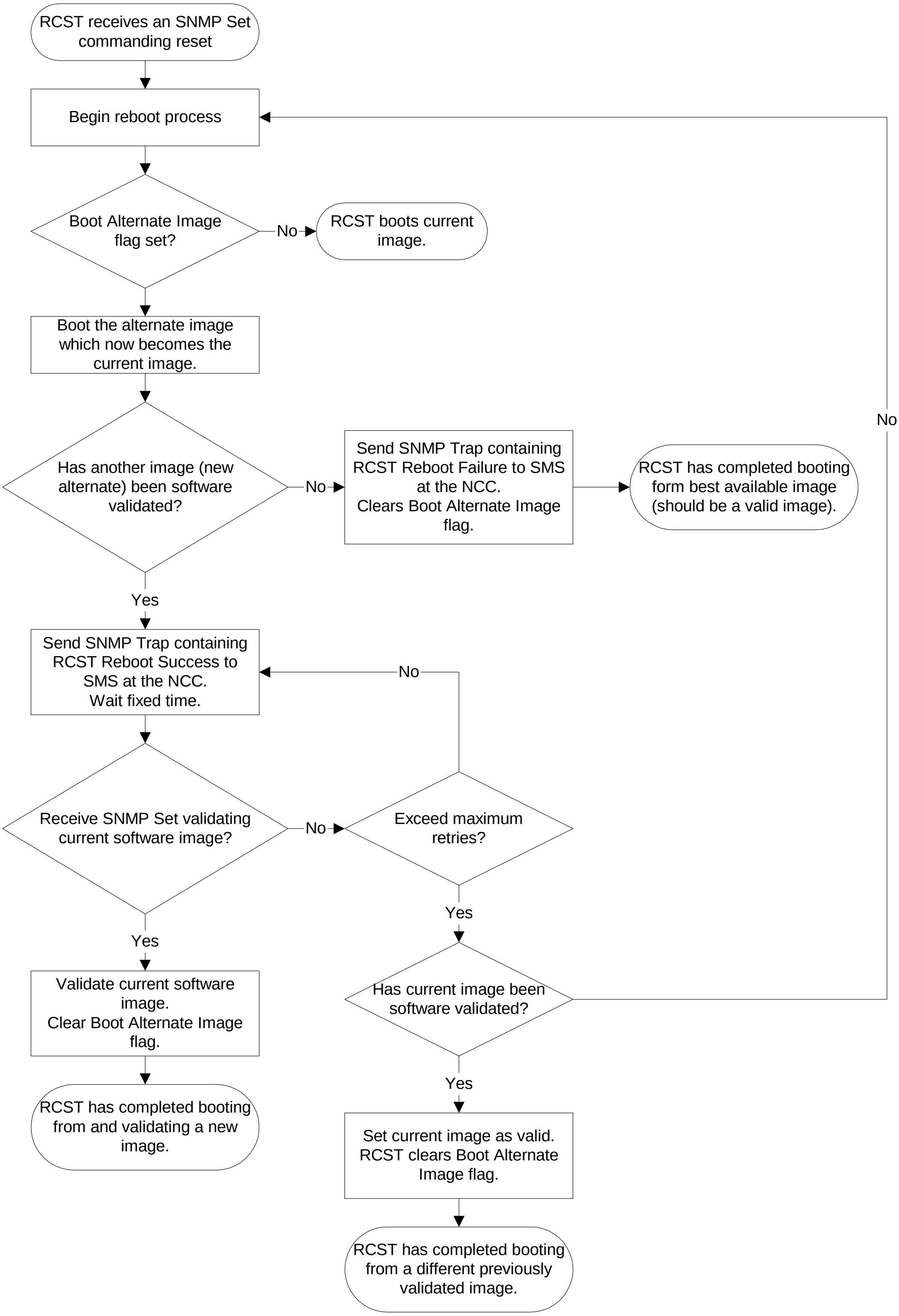

RCST receives an SNMP Set commanding reset
Begin reboot process
Boot Alternate Image flag set?
No
RCST boots current image.
Boot the alternate image which now becomes the current image.
Has another image (new alternate) been software validated?
No
Send SNMP Trap containing RCST Reboot Failure to SMS at the NCC.
Clears Boot Alternate Image flag.
RCST has completed booting form best available image (should be a valid image).
No
Yes
Send SNMP Trap containing RCST Reboot Success to SMS at the NCC.
Wait fixed time.
No
Receive SNMP Set validating current software image?
No
Exceed maximum retries?
Yes
Yes
Validate current software image.
Clear Boot Alternate Image flag.
Has current image been software validated?
RCST has completed booting from and validating a new image.
Yes
Set current image as valid.
RCST clears Boot Alternate Image flag.
RCST has completed booting from a different previously validated image.

Figure I.1: RCST Software Download

Whenever the NCC receives an SNMP Trap Reboot Status, Reboot Status=RCST reboot failure, the NCC sends an SNMP Set Request to perform software validation on the current image, updates the boot image ID in the local database with the current image information presented in the SNMP trap and changes the RCST reboot status to FAILED. The NCC logs the boot failure. The NCC repeats sending the SNMP Set Request for a specified number of times with a specified interval until the RCST acknowledges with a SNMP Response or the maximum number of attempts is reached.

Whenever the NCC receives an SNMP Trap Reboot Status, Reboot Status=RCST reboot success, the NCC sends a SNMP Set Request to perform software validation on the current image. The NCC repeats sending the SNMP Set Request for a specified number of times with a specified interval until the RCST acknowledges with a SNMP Response or the maximum number of attempts is reached. On receipt of the SNMP Response the NCC updates the boot image ID in the local database with the current image information presented in the SNMP trap and changes the RCST boot state to NORMAL. If the maximum number of attempts is exceeded, no changes are made in the NCC local database boot image ID and boot status.

I.1.4 Performance Parameters

NCC requests for OMs from the RCST periodically by sending an SNMP Get Request to the RCST.

RCST sends the requested parameters to the NCC by sending an SNMP Get Response to the NCC.

I.1.5 Fault Traps

RCST sends the fault traps to the NCC.

I.1.6 RCST Current Image ID

NCC sends a SNMP GET REQUEST, RCSTCurrent Image ID, message to get the current RCST image ID.

RCST responds with the currently running image ID.

I.1.7 RCST Alternate Image ID

NCC sends an SNMP GET REQUEST, RCST Alternate Image ID, message to get the alternate RCST image ID.

RCST responds with the alternate image ID.

I.2 Installation and Commissioning

The installation and commissioning of an RCST consists of the following steps:

1) Mounting and cabling of equipment components.

2) Antenna alignment in azimuth and elevation by maximizing the receive signal (like for DTH satellite reception).

3) Antenna polarization alignment by minimizing the receive signal on the other polarization (like for DTH satellite reception).

4) The installer enters all information that is specific for the RCST and its operation, in particular the geographical location.

5) The installer initiates that the RCST logs on to the satellite interactive network in "Installation Mode".

6) The RCST transmits CSC bursts with stepwise increasing power until the NCC replies with a TIM. In the CSC burst the RCST Mode field is set to Installation Mode.

7) The NCC replies with a TIM where the ACQ Assign Descriptor indicates unlimited ACQ slot assignment.

8) The RCST stores the transmit power that was successfully received by the NCC. For later normal logins the RCST uses this power increased by the expected rain attenuation.

9) The RCST and NCC perform coarse synchronization.

10) If requested by the network operator (depending on achievable accuracy of alignment by receive signal):

10.1) The RCST requests the installer to do fine antenna alignment.

10.2) The NCC provides the installer continuously with information on the level of the return link signal (ACQ bursts) that it receives.

10.3) The installer does fine antenna alignment by maximizing the receive level of the return link signal.

10.4) The installer indicates to the RCST that fine antenna alignment is finished.

11) The RCST and NCC perform fine synchronization.

12) If SNMP is implemented (see Clause 8.5), the NCC exchanges RCST specific information with the RCST by objects of the RCST MIB.

13) The installer initiates log off.

14) The installer initiates log on in "Operational Mode".

I.3 RCST System Processes

This clause provides a description of different processes represented in a hierarchical way. System processes are used to describe the system wide operations such as Login, Logoff, etc. The State Transition Processes are the processes that are used during transition from one state to another state (these are identified in the various state machines). The functional processes are used to describe a specific function.

These processes may affect several state machines and cover end-to-end system functions.

Note that a non-standard RADIUS is used between the RCST and the NCC. The standard RADIUS CHAP challenge mechanism is not used. Instead, the random number received from the TIM is used as the CHAP CHALLENGE in the encryption of the CHAP password. The reason for this deviation from standard RADIUS is the long transmission delay that is typical for geo-stationary satellites. Since each message exchange is delayed, the random number is transmitted once in a TIM during logon rather than by RADIUS CHAP challenge. This deviation from the standards applies only between the RCST and the NCC. Standard RADIUS and CHAP are used between the RADIUS Client at the NCC and the RADIUS server. Therefore, the existing RADIUS servers of Internet Service Providers can be used for authentication.

I.3.1 RCST Power On

RCST Power-On is a User-initiated process that consists of the RCST going from its unpowered state to the INITIALIZED state as follows:

1) the User powers the RCST

2) all RCST subsystems perform self-tests

3) the RCST achieves the IDLE state

4) the RCST acquires the Forward Link

5) the RCST achieves the INITIALIZED state

6) if the RCST was in the HOLD state before being powered off, it must return to the HOLD state

Depending on the outdoor environmental conditions, the RCST transmit subsystem should need no more than one minute of warm-up time before it is ready to transmit. During this warm-up time, the RCST may transition to the INITIALIZED state. The user will be informed if the transmit subsystem is not ready.

I.3.2 RCST Reset

RCST Reset is a process that can be initiated by the RCST, the NCC or the Superuser, and consists of the RCST going from its current state to the INITIALIZED state.

- the RCST may receive an RCST Reset command, either as an SNMP message from the NCC or over the web server from the Superuser or may determine to reset itself
- the RCST prepares for reset (i.e. close file system, etc.)
- RCST subsystems reset as specified by the Reset command, and perform self-tests
- the RCST achieves the IDLE state
- the RCST acquires the Forward Link
- the RCST achieves the INITIALIZED state

If the RCST was in the HOLD state before being reset, it must return to the HOLD state.

I.3.3 RCST Login

This is the process of the RCST going from INITIALIZED through OAM_ACTIVE to ACTIVE, one User going to AUTHENTICATED and the Secure Customer Device going from NO_ENCRYPT to ENCRYPT. If an accounting system is implemented in the satellite interactive network, then an RCST Login CDR is generated at the time the RCST transitions to OAM_ACTIVE.

I.3.4 RCST Re-login

This is the process of the RCST re-acquiring the return link and synchronizing all the user states to send user traffic. User state synchronization involves the RCST re-authenticating all users who have their Automatic Authentication attribute set. In order to accomplish this, the RCST should contain at least one user in the User Authentication Table. If an accounting system is implemented in the satellite interactive network, then an RCST Login CDR is generated at the time the RCST transitions to OAM_ACTIVE.

I.3.5 RCST Logoff

RCST Logoff is a process that can be initiated by the NCC or a Superuser, that consists of the RCST going from the ACTIVE state or the OAM ACTIVE state to the INITIALIZED state as follows:

1) The RCST is in the ACTIVE state or the OAM ACTIVE state.

2) When initiated by a the Superuser, the Superuser selects the RCST Logoff command and provides appropriate authentication (User, Password) information if not already done so. The RCST sends an RCST Logoff Request to the NCC and waits for a confirmation from the NCC.

3) When initiated by the NCC, the RCST receives an RCST Logoff command from the NCC. An SNMP Response is sent back to the NCC; the converse of the previous step.

4) If the RCST is in ACTIVE state when it receives confirmation from the NCC (regardless of whether RCST Logoff was initiated by the NCC or the Superuser), the RCST and the NCC release the Traffic and OAM VCCs and associated TBTP assignments. Finally, the RCST returns to the INITIALIZED state.

5) If the RCST is in OAM ACTIVE state when it receives confirmation from the NCC (regardless of whether RCST Logoff was initiated by the NCC or the Superuser), the RCST and the NCC release the OAM VCC and associated TBTP assignment. Finally, the RCST returns to the INITIALIZED state.

6) The RCST goes to TxD and loses the Return Link, the NCC releases RCST Validation and Host Authentication information.

7) If the logoff process has been initiated by the Superuser, the Superuser has to be informed of the action by the web server.

I.3.6 RCST Wake Up

RCST Wake up is the process of an RCST moving to either the OAM_ACTIVE or ACTIVE state to reply to forward link traffic. The exact RCST state depends on the type of traffic the RCST is replying to. The RCST transitions to the OAM_ACTIVE state to respond to OAM traffic. The RCST transitions to the ACTIVE state to forward unicast user traffic destined for a Host. For forwarding multicast traffic destined for hosts the RCST stays in the Initialized state.

I.3.6.1 Traffic Initiated RCST Wake Up

Traffic Initiated RCST Wake up is the process of the RCST acquiring Return Link Traffic capacity (going from the INITIALIZED or OAM_ACTIVE states to ACTIVE state) when it has data in its traffic buffer that comes from a host responding to Forward Link traffic.

For Normal Users not logged into RCST, unicast traffic on forward and return link will be dropped.

For Normal and Static Users logged to the RCST but not authenticated with the NCC, traffic on forward link will be forwarded to the host and traffic on return link will be buffered until successful authentication.

Multicast traffic should be forwarded to users. RCST should honour all JOIN requests irrespective of user authentication status.

Requirements:

- The RCST must have at least one User with Automatic Authentication attribute set for this host IP address in the User Authentication Table. Note this can be a static User Entry.

- The RCST is configured to be 'Wake-able' (in the NCC at service commissioning). The IP-DVB Gateway has a permanent entry for the RCST Secure Customer Device(s).

- Hosts behind the RCST have public IP addresses.

I.3.6.2 OAM RCST Wakeup

OAM RCST wakeup is the process of the RCST acquiring Return Link capacity (going from the INITIALIZED to OAM_ACTIVE state) when it has data in its OAM buffer that comes from responding to Forward Link traffic.

I.3.7 RCST Disable

The process of the RCST going from any state to INITIALIZED and then to the HOLD state. This process can be initiated by the NCC or Superuser as follows:

- When initiated by the Superuser, the Superuser selects the RCST Disable command from the RCSTs web page. The RCST then goes to the HOLD state.

- When initiated by the NCC, the RCST receives an RCST Logoff and Hold command as an SNMP message from the NCC and the RCST goes to the HOLD state.

NOTE: The RCST shall store its current state in non-volatile memory and shall recognize during power-on that it was last in the HOLD state and shall transition to the HOLD state after performing self-tests and acquiring the Forward Link. The RCST shall not pass forward link traffic to the user when in the HOLD state.

I.3.8 RCST Enable

The process of the RCST going from HOLD to INITIALIZED state Only the entity (NCC, Superuser) that placed the RCST in the HOLD state can take it out of the HOLD state.

I.3.9 User Login

The process of a user going from NOT AUTHENTICATED to AUTHENTICATED state. This may cause the RCST Login process to be executed.

I.3.10 User Logoff

The process of a user going from AUTHENTICATED to NOT AUTHENTICATED state. This may cause the RCST to initiate the Traffic Release process (if there are no more static or dynamic users in the user authentication table). For a normal user, the user logoff is generated when the user logs off on the web page or when the Superuser removes the entry. For static users, the user logoff is generated when the Superuser removes the entry.

I.4 State Transition Processes

The State transition processes describing the RCST Operations State Machine, Encryption, RCST Transmission, and Host Configuration State Machine shall comply with the following specifications.

The table in Clause I.4.1 is a legend for subsequent tables.

I.4.1 Name of Transition in State Machine

Initiating Events	The events that cause the transition to occur. 1. Event1 OR 2. Event2 OR 3. Event3 etc. Where Event can be a combination of multiple conditions (i.e. Condition1 AND Condition2).
Time (Min/Max/Avg) (Seconds)	The Min/Max/Avg time required to perform the state transition.
States: The initial state of the respective state machine from where the transition started. The final state of the respective state machine after the transition.	
Process: A series of operations performed during the state transition	

I.4.2 RCST Operations State Machine

I.4.2.1 Forward Link Acquisition

Initiating Events	1. The RCST has reached IDLE state.
States: Initial State: IDLE Final State: INITIALIZED	
Process: 1. Functional process: Forward Link Acquisition. 2. RCST locks local oscillator to PCR. 3. RCST enables normal user access to web pages.	

I.4.2.2 OAM Acquisition

The RCST must use the most recent set of PMT-SI tables before the start of the OAM Acquisition process.

Initiating Events	1. Data in the OAM buffer 2. New user login. 3. Data in the traffic buffer from the IP address of a user with the Automatic Authentication attribute set or Data in the traffic buffer and a static user entry in the User Authentication Table.
States: Initial State: INITIALIZED Final State: OAM_ACTIVE	
Process: **Data in the OAM buffer** 1. Functional Process: Return Link Acquisition. This results in the OAM VCC being built. 2. The NCC receives the bandwidth request and assigns timeslots to the request as available up to the requested amount. 3. The RCST waits for timeslots to be allocated in the TBTP and then services the OAM queue. **NOTE 1:** When the NCC receives the RCST synchronization message it generates an RCST login CDR. Process: **New user login** 1. Functional Process: Return Link Acquisition. This results in the OAM VCC being built. 2. The RCST creates a RADIUS request for the new user login. The RCST uses the Random Number received in Network Layer Information Message conveyed in the Network Layer Info Descriptor via the TIM message to encrypt the password. The RADIUS request is put in the OAM queue. 3. The NCC receives the bandwidth request and assigns timeslots to the request as available up to the requested amount. 4. The RCST waits for timeslots to be allocated in the TBTP and then services the OAM queue. **NOTE 2:** The RCST sends the RADIUS request with the username and CHAP password to the NCC. **NOTE 3:** When the NCC receives the RCST synchronization message it generates an RCST login CDR. Process: **Data in the traffic buffer from the IP address of a user with the Automatic Authentication attribute set or Data in the traffic buffer and a static user entry in the User Authentication Table**. 1. Functional Process: Return Link Acquisition. This results in the OAM VCC being built. 2. The RCST creates a RADIUS request for all users in the UAT in the NOT AUTHENTICATED state with the Automatic Authentication attribute set. The RCST uses the Random Number received in Network Layer Information Message conveyed in the Network Layer Info Descriptor via the TIM message to encrypt the password. The RADIUS request is put in the OAM queue. **NOTE 4:** The RCST may prioritize the authentication of users such that the user who generated traffic is authenticated first. 3. The NCC receives the bandwidth request and assigns timeslots to the request as available up to the requested amount. 4. The RCST waits for timeslots to be allocated in the TBTP and then services the OAM queue. **NOTE 5:** The RCST sends the RADIUS request(s) with the username and CHAP password to the NCC. **NOTE 6:** When the NCC receives the RCST synchronization message it generates an RCST login CDR.	

I.4.2.3 Traffic Acquisition

Initiating Events	1. **User authentication initiated**: Reception of Successful user authentication message from the NCC. The new user is the first user to be authenticated for that RCST in the current return link acquisition period. **EXAMPLE SCENARIO:** The RCST transitioned to the OAM_ACTIVE state from the INITIALIZED state due to a User login. 2. **Traffic initiated from a user in the User Authentication Table with the Automatic Authentication attribute set when at least one user is in the User Authenticated Table in the AUTHENTICATED state** or **Traffic initiated with a static user entry in the AUTHENTICATED state in the User Authentication Table** **EXAMPLE SCENARIO:** The RCST in the ACTIVE state released Traffic VCC due to Traffic Release timer expiry. The RCST transitioned to the OAM_ACTIVE state. One of the AUTHENTICATED users generated traffic triggering the RCST to acquire the Traffic VCC. 3. **Traffic initiated from a user in the User Authentication Table with the Automatic Authentication attribute set when NO users are in the User Authenticated Table in the AUTHENTICATED state** or **Traffic initiated with a static user entry in the NOT AUTHENTICATED state in the User Authentication Table** **EXAMPLE SCENARIO:** The RCST in the ACTIVE state released Traffic VCC due to Traffic Release timer expiry and released OAM VCC due to OAM Release timer expiry. Users are still logged in. A user generates traffic triggering the RCST to acquire the Return Link.
States: Initial State: OAM_ACTIVE Final State: ACTIVE	
Process: **User authentication initiated** 1. The NCC's RADIUS Client receives a Access-Accept message from the service provider's RADIUS 2. If a CBR RCST, the NCC sets CRA capacity for the RCST (configurable on a per RCST basis). 3. The NCC's RADIUS Client sends an Access-Accept message to the RCST 4. The NCC sends an SNMP Set message to the RCST setting the Traffic VCC assignment. NOTE 1: This message will also contain traffic contract parameters (maxCRA, maxVBDC, maxRBDC, FCAusable). 5. The NCC allocates the timeslots in the TBTP for the Traffic VCC and forwards TBTP to the RCST. 6. The RCST receives the Access-Accept message and executes Authentication Successful process (I.4.3.3.4) 7. The RCST receives the Traffic VCC. The RCST sends an SNMP Response message to the NCC. 8. If there are Automatic Authentication Users in the NOT AUTHENTICATED state, then the RCST sends RADIUS requests for each Automatic Authentication User. RCST receives TBTP and services Traffic queue. Process: **Traffic initiated from a user in the User Authentication Table with the Automatic Authentication attribute set when at least one user is in the User Authenticated Table in the AUTHENTICATED state** Or **Traffic initiated with a static user entry in the AUTHENTICATED state in the User Authentication Table** 1. RCST sends a SNMP Trap message to the NCC for Traffic VCC assignment. 2. If a CBR RCST, the NCC sets CRA capacity for the RCST (configurable on a per RCST basis). 3. The NCC sends an SNMP Set message to the RCST setting the Traffic VCC assignment. NOTE 2: This message will also contain traffic contract parameters (maxCRA, maxVBDC, maxRBDC, FCAusable). 4. The NCC allocates the timeslots in the TBTP for the Traffic VCC and forwards the TBTP to RCST. 5. The RCST receives the Traffic VCC. The RCST sends an SNMP Response message to the NCC. 6. RCST receives TBTP and services Traffic queue. **NOTE 3:** The RCST may prioritize the authentication of users such that the user who generated traffic is authenticated first.	

Process: **Traffic initiated from a user in the User Authentication Table with the Automatic Authentication attribute set when NO users are in the User Authenticated Table in the AUTHENTICATED state** Or **Traffic initiated with a static user entry in the NOT AUTHENTICATED state in the User Authentication Table** 1. The RCST creates a RADIUS request for all users in the NOT AUTHENTICATED state with the Automatic Authentication attribute set. The RCST uses the Random Number received in the TIM message to encrypt the password. The RCST sends the RADIUS request with the username and CHAP password to the NCC. **NOTE 4**: The RCST may prioritize the authentication of users such that the user who generated traffic is authenticated first. Upon reception of a successful user authentication at the NCC: 2. If a CBR RCST, the NCC sets CRA capacity for the RCST (configurable on a per RCST basis). 3. The NCC sends an SNMP Set message to the RCST setting the Traffic VCC assignment. NOTE 5: This message will also contain traffic contract parameters (maxCRA, maxVBDC, maxRBDC, FCAusable). 4. The NCC allocates the timeslots in the TBTP for the Traffic VCC and forwards TBTP to the RCST. 5. The RCST receives the Access-Accept message. 6. The RCST receives the Traffic VCC. The RCST sends an SNMP Response message to the NCC. 7. The RCST receives TBTP and services the Traffic queue.

I.4.2.4 Traffic Release

Initiating Events	1. Traffic Release Timer expires . 2. User Authentication Table on RCST is empty (i.e. The last user logs off and the RCST is in the ACTIVE state).
States: Initial State: ACTIVE Final State: OAM_ACTIVE	
Process: 1. The RCST sends an SNMP Trap message containing the Traffic Release Request to the NCC. 2. The NCC receives Traffic Release Request and releases Traffic Resources. 4. The NCC de-allocates the Traffic VCC. 5. The NCC sends an SNMP Set message containing the Traffic Release Reply to the RCST to inform of successful release. 6. The RCST clears Traffic VCC and sends a SNMP response to the NCC.	

I.4.2.5 OAM Release

Initiating Events	1. OAM Release Timer expires. 2. **Superuser initiated:** Superuser via the Web Interface issues an RCST Logoff command. 3. **NCC operator initiated:** Reception of an RCST Logoff Command message from the **NCC** via SNMP.
States: Initial State: OAM_ACTIVE Final State: INITIALIZED	
Process: **OAM Release Timer expires** 1. The RCST sends an SNMP Trap message containing the OAM Release Request to the **NCC**. 2. The RCST notifies all users connected via a Web page that the RCST is logging out . **NOTE 1:** Hosts not in the login process do not receive direct notification of RCST logoff. No suitable method has been identified at this time to notify users that do not have the browser open. 3. The RCST clears all users that do not have the Automatic Authentication attribute set, from the User Authentication Table. 4. The RCST changes all users that do have the Automatic Authentication attribute set, to NOT AUTHENTICATED. 5. RCST clears OAM VCC. 6. The RCST stops sending SYNC bursts. NOTE 2: The NCC detects loss of SYNC and initiates Functional Process: Loss of Sync at **NCC**. The RCST has no RF resources at this Point. Process: **Superuser initiated** 1. The RCST sends a SNMP Trap message containing the RCST Logoff Request message to the **NCC**. 2. The RCST notifies all users connected via a Web page that the RCST is logging out. **NOTE 3:** Hosts not in the login process do not receive direct notification of RCST logoff. No suitable method has been identified at this time to notify users that do not have the browser open. 3. The RCST clears all users that do not have the Automatic Authentication attribute set, from the User Authentication Table. 4. The RCST changes all users that do have the Automatic Authentication attribute set, to NOT AUTHENTICATED. 5. The RCST must flush all (OAM and Traffic) buffers. 6. RCST clears OAM VCC. 7. The RCST stops sending Sync. NOTE 4: The **NCC** detects Loss of Sync and initiates Functional Process: Loss of Sync at **NCC**.	
Process: **NCC operator initiated** 1. RCST receives the RCST Logoff command from the NCC via SNMP and sends response. 2. The RCST notifies all users connected via a Web page that the RCST is logging out. **NOTE 5:** Hosts not in the login process do not receive direct notification of RCST logoff. No suitable method has been identified at this time to notify users that do not have the browser open. 3. The RCST clears all users that do not have the Automatic Authentication attribute set, from the User Authentication Table. 4. The RCST changes all users that do have the Automatic Authentication attribute set, to NOT AUTHENTICATED. 5. The RCST must flush all (OAM and Traffic) buffers. 6. RCST clears OAM VCC. 7. The RCST stops sending SYNC bursts. NOTE 6: The NCC detects Loss of Sync and initiates Functional Process: Loss of Sync at NCC. **NOTE 7:** After receiving a Logoff command, the RCST will send a verification to the NCC via an SNMP Response. The RCST must ensure that this Response does not cause the RCST to re-acquire the return link	

I.4.2.6 Return Link Release

Note this is an OAM process (either the NCC or a Superuser issues RCST Logoff command). This process will log the RCST off from the NCC and forces the releases of all RF resources as well as making the NCC cleanup all resources.

Initiating Events	1. **Superuser initiated, RCST (ACTIVE):** Superuser via the Web Interface issues an RCST Logoff command. 2. **NCC operator initiated, RCST (ACTIVE):** Reception of an RCST Logoff Command message from the NCC via SNMP.
States: Initial State: ACTIVE Final State: INITIALIZED	
Process: **Superuser initiated, RCST (ACTIVE)** 1. The RCST sends a SNMP Trap message containing the RCST Logoff Request message to the NCC. 2. The RCST notifies all users connected via a Web page that the RCST is logging out. NOTE 1: Hosts not in the login process do not receive direct notification of RCST logoff. No suitable method has been identified at this time to notify users that do not have the browser open. 3. The RCST clears all users that do not have the Automatic Authentication attribute set, from the User Authentication Table. 4. The RCST changes all users that do have the Automatic Authentication attribute set, to NOT AUTHENTICATED. 5. The RCST must flush all (OAM and Traffic) buffers. 6. RCST clears Traffic VCC. 7. RCST clears OAM VCC. 8. RCST stops sending SYNC bursts. NOTE 2: The NCC detects Loss of Sync and initiates Functional Process: Loss of Sync at NCC. Process: NCC operator initiated, RCST (ACTIVE) 1. RCST receives the RCST Logoff command from the NCC via SNMP and acknowledges via an SNMP Response. 2. The RCST notifies all users connected via a Web page that the RCST is logging out. **NOTE 3:** Hosts not in the login process do not receive direct notification of RCST logoff. No suitable method has been identified at this time to notify users that do not have the browser open. 3. The RCST clears all users that do not have the Automatic Authentication attribute set, from the User Authentication Table. 4. The RCST changes all users that do have the Automatic Authentication attribute set, to NOT AUTHENTICATED. 5. The RCST must flush all (OAM and Traffic) buffers. 6. RCST clears Traffic VCC. 7. RCST clears OAM VCC. 8. RCST stops sending Sync. NOTE 4: The NCC detects Loss of Sync and initiates Functional Process: Loss of Sync at NCC. NOTE 5: After receiving a Logoff command, the RCST will send a verification to the NCC via an SNMP Response. The RCST must ensure that this Response does not cause the RCST to re-acquire the return link.	

I.4.2.7 Return Link Disable

Initiating Events	1. **Superuser initiated** 1.1 Superuser via the Web Interface issues an RCST Disable command. 1.2 Reaching the INITIALIZED state (from any state) with the Disable by Superuser parameter set. **NOTE 1:** The Disable by Superuser parameter is preserved over RCST reboots. 2. **NCC operator initiated** 2.1 Reception of an RCST Disable command from the NCC via SNMP. 2.2 Reaching the INITIALIZED state (from any state) with the Disable by **NCC** parameter set. **NOTE 2:** The Disable by **NCC** parameter is preserved over RCST reboots
States: Initial State: INITIALIZED Final State: HOLD	
Process: **Superuser initiated** 1. The RCST tracks that the Superuser has issued the Disable command. 2. RCST clears all non-static Users in the User Authentication Table on RCST. Process: **NCC operator initiated** 1. The RCST tracks that the NCC has issued the command that set the Disable parameter. 2. The RCST clears all non-static users in the User Authentication Table.	

I.4.2.8 Return Link Enable

Initiating Events	1. **Superuser initiated**: Superuser via the Web Interface issues an RCST Enable command and the Superuser had issued the Disable command. 2. **NCC operator initiated**: Reception of an RCST Enable Command message from the NCC via SNMP Set message and the NCC had issued the Disable command.
States: Initial State: HOLD Final State: INITIALIZED	
Process: **Superuser initiated** 1. The Superuser via the web interface issues the RCST Enable command. 2. The RCST transitions to the INITIALIZED state. Process: **NCC operator initiated** 1. The RCST receives the SNMP Set Message (RCST Enable). 2. The RCST transitions to the INITIALIZED state. 3. The response is placed in the OAM buffer. 4. The RCST executes the OAM acquisition – OAM data initiated process.	

I.4.3 RCST Configurations State Machine

The Initial and Final States are from the RCST perspective.

I.4.3.1 Encryption

I.4.3.1.1 Phase 1 SA Acquisition

Initiating Events	1. The RCST receives the Traffic VCC and has no SA established. 2. The NCC IPSec sends the first IKE negotiation message.
States: Initial State: NO ENCRYPT Final State: PHASE 1 ESTABLISHED	
Process: **The RCST receives the Traffic VCC and has no SA established** 1. The RCST notifies the IPSec software to start IPSec IKE negotiation (total 6 messages). NOTE 1: The IPSec negotiation must be queued BEFORE any data already in the Traffic Queue. 2. The IPSec software completes IKE Phase 1 with the NCC IPSec. Process: **The NCC IPSec sends the first IKE negotiation message.** 1. The RCST receives the first IKE negotiation message. 2. The RCST IPSec software responds to the IKE negotiation via the traffic queue. **NOTE 2**: The IPSec negotiation must be queued BEFORE any data already in the Traffic Queue. **NOTE 3**: If the RCST is not in the ACTIVE state, this may cause the RCST to transition to the ACTIVE state. 3. The RCST IPSec software completes IKE Phase 1 negotiation with the **NCC** IPSec.	

I.4.3.1.2 Phase 2 SA Acquisition

Initiating Events	1. Phase 1 SA Acquisition has finished successfully. 2. Received incoming traffic for IPSec Peer. 3. The Peer IPSec starts Phase 2 SA negotiation.
States: Initial State: PHASE 1 ESTABLISHED Final State: ENCRYPT	
NOTE 1: The IPSec negotiation must be queued BEFORE any data already in the Traffic Queue. **NOTE 2:** If the RCST is not in the ACTIVE state, this may cause the RCST to transition to the ACTIVE state. **NOTE 3:** If the RCST cannot re-establish the Return Link via the system processes, the IPSec software behaves as if the packet was lost. **NOTE 4:** Traffic starts flowing at the point when the Phase 2 SA Acquisition is complete. Process: **Phase 1 SA Acquisition has finished successfully.** 1. The local IPSec send a Phase 2 IKE negotiation message to the peer IPSec. 2. The peer IPSec responds to the IKE negotiation (in the case of the RCST via the traffic queue) 3. The local IPSec completes IKE Phase 2 negotiation with the peer IPSec Process: **Received incoming traffic for IPSec Peer.** 1. The local IPSec send a Phase 2 IKE negotiation message to the peer IPSec. 2. The peer IPSec responds to the IKE negotiation (in the case of the RCST via the traffic queue) 3. The local IPSec completes IKE Phase 2 negotiation with the peer IPSec Process: **The Peer IPSec starts Phase 2 SA negotiation** 1. The Peer IPSec sends a Phase 2 IKE negotiation message to the Local IPSec. 2. The Local IPSec responds to the IKE negotiation. 3. The Peer IPSec completes IKE Phase 2 negotiation with the Local IPSec.	

I.4.3.1.3 Phase 2 SA Release

Initiating Events	1. The local IPSec Phase 2 SA Time-To-Live timer (for The NCC minimum 2 minutes, maximum 23 Hours, default 8 Hours) expires. 2. The **NCC** Maximum amount of Data exceeded at the **NCC** IPSec (default No maximum). 3. The **NCC** Idle timer expires at **NCC** IPSec (minimum 15 minutes, maximum Indefinite).
States: Initial State: ENCRYPT & NEGOTIATE Final State: PHASE 1 ESTABLISHED	
Process: **The Local IPSec Phase 2 SA Time-To-Live timer expires** 1. Local Phase 2 SA is released. **NOTE 1:** Traffic stops flowing at this point. Process: **The NCC Maximum amount of Data exceeded at the NCC IPSec** **NOTE 2:** Traffic from the **NCC** to the RCST stops flowing at this point. 1. NCC Phase 2 SA is released. **NOTE 3:** The maximum amount of data is for input or output for IPSec server. Process: **The NCC Idle timer expires at NCC IPSec** 1. The **NCC** Phase 2 SA is released. **NOTE 4:** Traffic from the NCC to the RCST has stopped flowing at this point.	

I.4.3.1.4 Phase 1 SA Release

Initiating Events	1. Phase 1 SA timer expires at local IPSec. 2. IPSec receives a SA tear-down message.
States: Initial State: PHASE 1 ESTABLISHED Final State: NO ENCRYPT	
Process: **Phase 1 SA timer expires at local IPSec** 1. The Phase 1 SA timer expires at the local IPSec. 2. The local IPSec sends the SA tear-down message to its peer. 3. The local IPSec tears down SA. 4. The peer IPSec software receives the SA tear-down message. 5. The peer IPSec tears down SA. Process: **IPSec receives a SA tear-down message** 1. The local IPSec software receives the SA tear-down message. 2. The local IPSec software tears-down the IPSec tunnel.	

I.4.3.1.5 Full Encrypt Release

Initiating Events	1. Phase 1 SA timer expires at local IPSec. 2. Phase 1 SA tear-down message received from the peer IPSec.
States: Initial State: ENCRYPT Final State: NO ENCRYPT	
Process: **Phase 1 SA timer expires at local IPSec** 1. The local IPSec sends the SA tear-down message to its peer. 2. The local IPSec tears-down the SA. **NOTE 1:** Traffic from local to peer stops flowing at this point. 3. The peer IPSec software receives the SA tear-down message. 4. The peer IPSec tears down SA. **NOTE 2:** Traffic from peer to local stops flowing at this point. Process: **Phase 1 SA tear-down message received from the peer IPSec** 1. The local IPSec tears-down the SA. **NOTE 3:** Traffic from local to peer stops flowing at this point.	

I.4.3.1.6 Full Encrypt & Negotiate Release

Initiating Events	1. Phase 1 SA timer expires at local IPSec. 2. Phase 1 SA tear-down message received from the peer IPSec.
States: Initial State: ENCRYPT & NEGOTIATE Final State: NO ENCRYPT	
Process: **Phase 1 SA timer expires at local IPSec** 1. The local IPSec sends the SA tear-down message to its peer. 2. The local IPSec tears-down the SA. **NOTE 1:** Traffic from local to peer stops flowing at this point. 3. The peer IPSec software receives the SA tear-down message. 4. The peer IPSec tears down SA. **NOTE 2:** Traffic from peer to local stops flowing at this point. Process: **Phase 1 SA tear-down message received from the peer IPSec** 1. The local IPSec tears-down the SA. **NOTE 3:** Traffic from local to peer stops flowing at this point.	

I.4.3.1.7 Phase 2 SA Re-negotiation

Initiating Events	1. 15/16 of the maximum amount of Data exceeded at the NCC IPSec (default No maximum). 2. 15/16th of the Phase 2 SA Time-To-Live timer expires at the NCC. 3. The RCST IPSec reaches its grace period for the Time-To-Live timer.
States: Initial State: ENCRYPT Final State: ENCRYPT & NEGOTIATE	
NOTE 1: At no time during the below processes is traffic interrupted. **NOTE 2:** The IPSec server SA negotiation message sets a 16 s timer. If response is not received within the 16 s timeout, the renew message is send again (total of four times). If reply is not received after the forth time, the IPSec server Switch gives up. However, if any data is received from the involved partner, the IPSec server Switch starts (4*16 s) all over again. **NOTE 3:** when 15/16th of SA life-time (time-to-live, amount of data) passes, the IPSec server starts re-negotiating for a new pair of SA. When the new pair of SA is successfully negotiated, the new pair of SA is used. The flow of traffic is not effected during the negotiation. Process: **15/16 of the maximum amount of Data exceeded at the NCC IPSec** 1. The **NCC** IPSec sends the Phase 2 SA re-negotiation message to the RCST IPSec. 2. The **NCC** IPSec starts re-negotiation. 3. The RCST IPSec receives the Phase 2 SA re-negotiation message from the **NCC** IPSec. 4. The RCST IPSec starts re-negotiation. **NOTE 4:** The maximum amount of data is for input or output for IPSec server. Process: **15/16th of the Phase 2 SA Time-To-Live timer expires at the NCC.** 1. The **NCC** IPSec sends the Phase 2 SA re-negotiation message to the RCST IPSec. 2. The **NCC** IPSec starts re-negotiation. 3. The RCST IPSec receives the Phase 2 SA re-negotiation message from the **NCC** IPSec. 4. The RCST IPSec starts re-negotiation. Process: **The RCST IPSec reaches its grace period for the Time-To-Live timer.** 1. The RCST IPSec sends the Phase 2 SA re-negotiation message to the **NCC** IPSec. 2. The RCST IPSec starts re-negotiation. 3. The **NCC** IPSec receives the Phase 2 SA re-negotiation message from the RCST IPSec. 4. The **NCC** IPSec starts re-negotiation. **NOTE 5:** Grace Period maximum is 10 % of the timer.	

I.4.3.1.8 Phase 2 SA Renewed

Initiating Events	1. Successful negotiation
States: Initial State: ENCRYPT & NEGOTIATE Final State: ENCRYPT	
NOTE: At no time during the below process is traffic interrupted. Process: **Successful Negotiation** 1. The Local IPSec has successfully negotiated with the Peer IPSec. 2. The encrypted data continues to be exchanged.	

I.4.3.2 RCST Transmission

I.4.3.2.1 Transmission Enable

Initiating Events	1. RCST in the INITIALIZED state and initiates OAM Acquisition. 2. RCST in OAM ACTIVE or ACTIVE state has previously become TxD due to some fault condition, and the fault condition has been resolved before some fault timeout.
States: Initial State: TxD Final State: TxE	
Process: 1. The RCST checks if the SSPA is ready. If SSPA is not ready: 1. The RCST notifies the users that the RCST is not ready. 2. Waits for the SSPA to become ready. 2. RCST enables its transmission chain.	

I.4.3.2.2 Transmission Disable

Initiating Events	1. RCST in the OAM ACTIVE or ACTIVE state transitions to the INITIALIZED state. 2. RCST in the OAM ACTIVE or ACTIVE state detects some fault condition.
States: Initial State: TxE Final State: TxD	
Process: 1. RCST disables its transmission chain, effectively muting its SSPA.	

I.4.3.3 User Authentication State Machine

The Initial and Final States described in following clauses are from the User perspective.

I.4.3.3.1 Normal User login to RCST

Initiating Events	1. Normal User browses Web Page on RCST to Login .
States: Initial State: USER NOT LOGGED IN TO RCST End State: AUTHENTICATION REQUESTED	
Process: 1. The user enters User name and password through the RCST Web page. The RCST Web page provides the user with a status of the log-in request. 2. The RCST captures and stores the host IP address, username, password. 3. The user status is changed to AUTHENTICATION REQUESTED in the User Authentication Table. If the RCST is in the INITIALIZED state: 1. The OAM acquisition – New User Login is executed. If the RCST is in the OAM_ACTIVE or ACTIVE state: 1. The RCST retrieves the RAND (random number) from the TIM message received during the Return Link Acquisition. 2. The RCST uses the RAND to encrypt the user password and sends the Access-Request message to the **NCC**. 3. The RCST starts the timer on the RADIUS authentication process. 4. The RCST services the OAM queue, forwarding the message to the RADIUS Client in the NCC via the OAM VCC. 4. The NCC performs Functional Process: User Login at NCC. **NOTE:** The user login CDR is recorded at the time the response to the access-request is sent to the RCST.	

I.4.3.3.2 Static User Login to RCST

Initiating Events	1. Superuser enters Static User parameters into the RCST
States: Initial State: NOT LOGGED IN TO RCST End State: NOT AUTHENTICATED	
Process: 1. RCST Superuser access RCST configuration web page and provides username and password to be stored in the User Authentication Table for the static user. 2. The user status is changed to NOT AUTHENTICATED in the User Authentication Table.	

I.4.3.3.3 RCST Re-login to NCC

Initiating Events	1. RCST receives data in the Traffic Buffer
States: Initial State: NOT AUTHENTICATED End State: AUTHENTICATION REQUESTED	
Process: 1. The RCST retrieves the IP address, username and password for the user from the User Authentication Table for the static user. 2. The user status is changed to AUTHENTICATION REQUESTED in the User Authentication Table. If the RCST is in the INITIALIZED state: 1. The OAM ACQUISITION – New User Login is executed. If the RCST is in the OAM_ACTIVE, ACTIVE state: 1. The RCST retrieves the RAND (random number) from the TIM message received during the Return Link Acquisition. 2. The RCST uses the RAND to encrypt the user password and sends the Access-Request message to the NCC. 3. The RCST services the OAM queue, forwarding the message to the RADIUS Client in the NCC via the OAM VCC. 4. The NCC performs Functional Process: User Login at NCC. For more details on User Authentication refer to Appendix **17**. **NOTE:** The user login CDR is recorded at the time the response to the access-request is sent to the RCST.	

I.4.3.3.4 Authentication Successful

Initiating Events	Received an Access-Accept message at the RCST from RADIUS Client at the NCC
States: Initial State: AUTHENTICATION REQUESTED End State: AUTHENTICATED	
Process: 1. The RCST updates the user status to AUTHENTICATED and sets the Automatic Authentication attribute in the User Authentication Table. 2.Stop timer from RADIUS authentication (Note: This is not expiring the timer but stopping it from expiring). 2. If the user is a Normal User the RCST updates the Web page to show the successful authentication.	

I.4.3.3.5 Authentication Failure - Normal User

Initiating Events	1. Received an Access-Reject message at the RCST from the RADIUS Client at the NCC 2. Timer for the RADIUS authentication expires (no reply from RADIUS Client in the NCC for a TBD amount of time). 3. RCST loss of synchronization.
States: Initial State: AUTHENTICATION REQUESTED End State: NOT LOGGGED IN TO RCST	
Process: 1. The RCST updates Web page to show authentication failure. 2. RCST removes failed non-static user entry from the User Authentication Table and cleans-up all the user related transactions that have been maintained at the RCST. 3. RCST stops the RADIUS authentication timer (Note: This is not expiring the timer but stopping it from expiring).	

I.4.3.3.6 Static User Authentication Failure

Initiating Events	1. Received an Access-Reject message at the RCST from the RADIUS Client at the NCC 2. Received no reply from the RADIUS Client in the NCC for a TBD amount of time. 3. RCST loss of synchronization.
States: Initial State: AUTHENTICATION REQUESTED End State: NOT AUTHENTICATED	
Process: 1. The RCST logs the authentication failure. 2. RCST does not remove failed user entry from the User Authentication Table, but cleans-up all the user related transactions that have been maintained at the RCST. 3. RCST stops the RADIUS authentication timer (Note: This is not expiring the timer but stopping it from expiring).	

I.4.3.3.7 Logoff from RCST - Not Authenticated

Initiating Events	1. User browses Web Page on RCST to Logoff 2. Superuser clears the user from the User Authentication Table. 3. RCST executes: RCST Disable (see 0)
States: Initial State: NOT AUTHENTICATED End State: NOT LOGGED IN TO RCST	
Process: **User browses Web Page on RCST to Logoff** 1. The user enters Username and Password. The RCST Web page provides the user with a status of the log-off request. 2. The RCST verifies the Username and Password with the values in the User Authentication Table. If the username is not in the table. The RCST updates the web page to show failure of logout. This causes the user's Web browser to stops refreshing the web page. If the username is in the table 1. The RCST removes the non-static User entry from the User Authentication Table. Process: **Superuser clears the user from the User Authentication Table** 1. The RCST removes the entry from the User Authentication Table.	

I.4.3.3.8 Logoff from RCST - Authentication Requested

Initiating Events	1. User browses Web Page on RCST to Logoff 2. Superuser clears the user from the User Authentication Table. 3. OAM VCC release
States: Initial State: AUTHENTICATION REQUESTED End State: NOT LOGGED IN TO RCST	
Process: **User browses Web Page on RCST to Logoff** 1. The user enters Username and Password. The RCST Web page provides the user with a status of the log-off request. 2. The RCST verifies the Username and Password with the values in the User Authentication Table. If the username is not in the table1. The RCST updates the web page to show failure of logout. This causes the user's Web browser to stops refreshing the web page. If the username is in the table 1. The RCST sends an SNMP Trap message containing the User Logoff message to the NCC. Included in this message is the Username, Host IP address and RCST MAC Address. 2. Begin Functional Process: User Logoff at NCC. 3. The RCST removes the non-static User entry from the User Authentication Table. 4. The RCST updates the web page to show success of logout. This causes the web browser to stop refreshing the web page. NOTE 1: If the RCST receives an authentication success after a user has been logged off it will silently drop this. Process: **Superuser clears the user from the User Authentication Table** 1. The RCST sends an SNMP Trap message containing the User Logoff message to the NCC. Included in this message is the Username, Host IP address and RCST MAC Address. 2. Begin Functional Process: User Logoff at NCC. 3. The RCST removes the non-static User entry from the User Authentication Table. **NOTE 2**: If the RCST receives an authentication success after a user has been logged off it will silently drop this. **NOTE 3**: If this is the last User in the User Authentication Table AND the RCST is in the ACTIVE state, the RCST executes the Traffic Release process **NOTE 4**: When the NCC receives an SNMP Trap message (User Logoff), the NCC generates the user logoff CDR. Process: **OAM VCC release** The RCST clears all users that do not have the Automatic Authentication attribute set from the User Authentication Table.	

I.4.3.3.9 Logoff from RCST - Authenticated

Initiating Events	1. User browses Web Page on RCST to Logoff 2. Superuser clears the user from the User Authentication Table.
States: Initial State: AUTHENTICATED End State: NOT LOGGED IN TO RCST	
Process: **User browses Web Page on RCST to Logoff** 1. The user enters Username and Password. The RCST Web page provides the user with a status of the log-off request. 2. The RCST verifies the Username and Password with the values in the User Authentication Table. If the username is not in the table1. The RCST updates the web page to show failure of logout. This causes the user's Web browser to stops refreshing the web page. If the username is in the table 1. The RCST sends an SNMP Trap message containing the User Logoff message to the NCC. Included in this message is the Username, Host IP address and RCST MAC Address. 2. Begin Functional Process: User Logoff at NCC. 3. The RCST removes the non-static User entry from the User Authentication Table. 4. The RCST updates the web page to show success of logout. This causes the web browser to stop refreshing the web page. Process: **Superuser clears the user from the User Authentication Table** 1. The RCST sends an SNMP Trap message containing the User Logoff message to the NCC. Included in this message is the Username, Host IP address and RCST MAC Address. 2. Begin Functional Process: User Logoff at NCC. 3. The RCST removes the non-static User entry from the User Authentication Table. **NOTE 1:** If this is the last User in the User Authentication Table AND the RCST is in the ACTIVE state, the RCST executes the Traffic Release process **NOTE 2:** When the NCC receives an SNMP Trap message (User Logoff), the NCC generates the user logoff CDR.	

I.4.3.3.10 RCST Transition to INITIALIZED State

Initiating Events	1. RCST transitions to INITIALIZED State.
Time (Min/Max/Avg) (Seconds)	
States: Initial State: AUTHENTICATED End State: NOT AUTHENTICATED	
Process: **RCST transitions to INITIALIZED State** 1. Upon transition of the RCST to the INITIALIZED state all Authenticated Users are transitioned to the NOT AUTHENTICATED state.	

I.5 RCST Power Control

An RCST typically contains a Solid State Power Amplifier (SSPA) for amplifying the transmit signal. The control of the RCST SSPA operating point is required to obtain optimum transmission quality in the satellite interactive network. Operation in the non-linear region of the SSPA causes spectrum re-growth of the transmitted carrier, which interferes into adjacent carriers. At the nominal SSPA output power the spectrum re-growth shall not exceed -20 dBc. The RCST should provide means for avoiding that the SSPA operates at a point where the spectrum re-growth exceeds the required -20 dBc. This should be achieved over the specified range of temperature and carrier frequency.

In order to mitigate rain fade the NCC controls the RCST transmit power by entries in the CMT or by a Correction Message Descriptor. If the NCC requests the RCST to increase the power, then the RCST has to take care that the output power is limited as described above.

I.6 Multicast Handling

This clause describes an implementation of end-to-end multicasting. Multicast session operation has two components: an RCST component and a NCC component.

I.6.1 Invoking a Multicast Session from RCST-side

When a user wishes to join a multicast session (Multicast session means PID/IP pair), the user starts on his host an application utilizing IP multicast. The Host application uses the Class D address and sends an IGMP JOIN message over the local (Host's) network. The RCST picks up the JOIN message and uses the IP address to look up the corresponding Transport Stream ID, Original Network ID and PID in the Multicast PID Mapping Table. When a match is made, the PID is put into the RCST's active PID list so that the RCST can begin to decode the multicast data when it is broadcast from the NCC. If the number of RCST-supported multicast sessions is exceeded (minimum 14 multicast IP addresses, which can be spread over the maximum of 10 different PIDs per RCST assigned to multicast, no multiplex of IP multicast addresses over MAC multicast addresses is assumed on the same PID), the RCST ignores the JOIN message received from the host and no error message is given to the user by the RCST. All error processing for such a case is assumed to be handled by the application.

This is depicted below in Figure I-2.

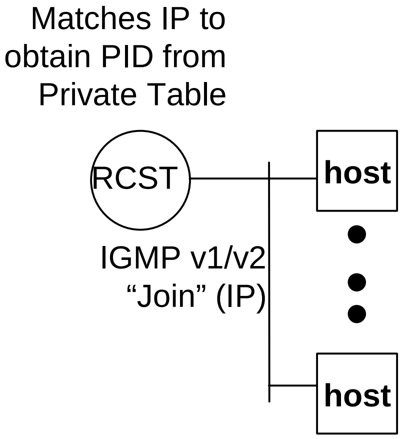


Figure I-2: Invoking a multicast receive session: RCST side

The RCST calculates the MAC address based on [27] the MAC address is not included in the Multicast PID Mapping Table sent from the Gateway.

To find the Multicast PID Mapping Table the RCST parses the RCS Map Table and selects the IP/DVB data broadcast services providing a match between the RCST's own population id and the population id contained in the linkage descriptors pointing to IP/DVB services (linkage_type 0x06). Only one IP/DVB service is assigned to a single RCST.

Once the IP/DVB service is identified, the RCST parses the PMT where it will get the PID value carrying the Multicast PID Mapping Table identified by the Elementary Loop with the stream type = 0x05 and table id = **0xA7** in the RCS content descriptor.

From the Multicast PID Mapping Table the RCST will know on which PID(s) it will get its multicast IP traffic.

The RCST maintains a counter of all active hosts on a particular group. The counter is incremented with each JOIN, and decrements with every LEAVE, with the PID being retired from the active list when the counter reaches zero.

I.6.2 Revoking a Multicast Session from RCST-side

When a user wishes to leave a multicast session, the user stops the application on his host that uses the IP multicast session. The host application uses the Class D address and sends an IGMP LEAVE message over the local (host's) network.

The RCST sends queries for hosts on every active group on a periodic basis. If there are no more listeners, the RCST can stop filtering on that multicast MAC address. If there is no more MAC filtering on this PID, the RCST removes the PID from the RCST's active cache.

I.6.3 Multicast Source Transmission

Multicast sessions may be sourced from the following entities using the following formats:

- From an RCST to the NCC, using unicast encapsulation. The RCST will block native multicast transmission from the hosts to the NCC (since the RCST will never transmit any IP traffic with a class D destination address to the NCC). This is necessary for avoiding that the RCST sends back to the Internet multicast traffic that has been received from the Internet over a different connection.
- Via the Internet to the NCC, using unicast encapsulation.
- Via a dedicated link into the Mrouter, using a native multicast or using unicast encapsulation.
- From a multicast server located at the NCC.

All encapsulated multicast transmissions go through a GRE Decapsulator which de-capsulates the packets and re-routes into the mrouter using native multicast, as shown in Figure I-3. GRE encapsulation is used for Multicast. GRE is described in [28] and [29].

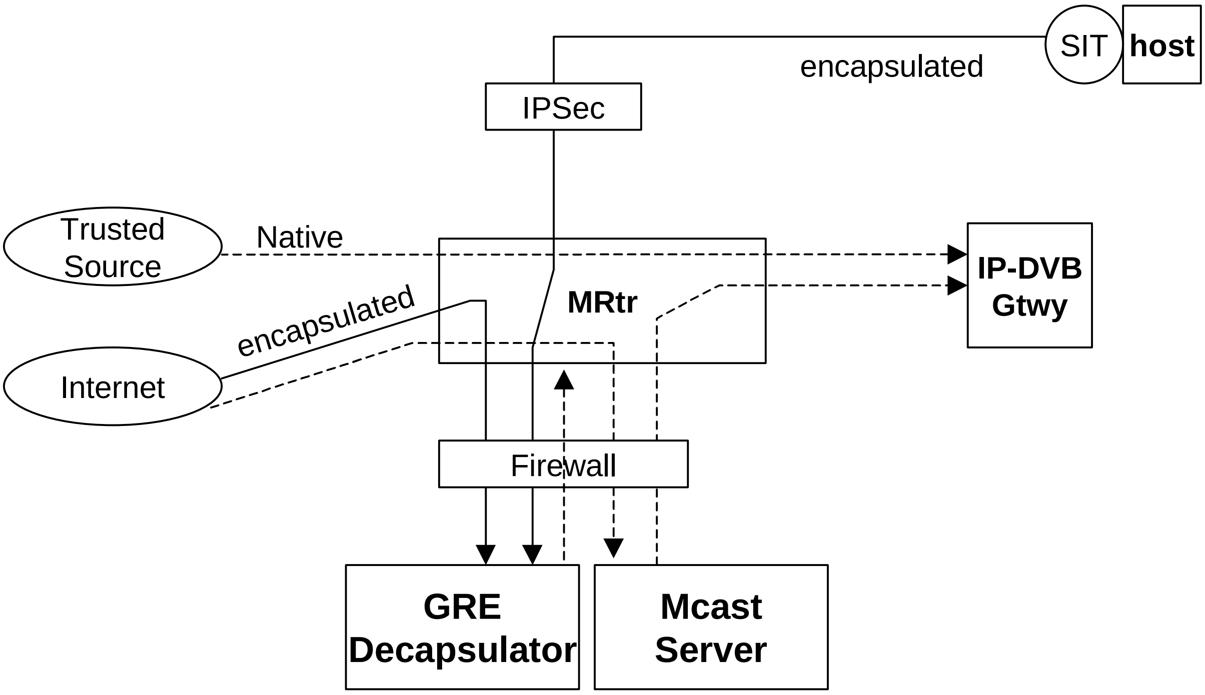


Figure I-3: Multicast source transmission model

The GRE Decapsulator receives encapsulated multicast transmissions. It strips the unicast encapsulation and forwards the multicast addressed packets through the firewall to the multicast router. The router then forwards the packets to the IP/DVB gateway. The GRE Decapsulator is mainly used for streaming applications. The multicast server is the source for native local multicast streams. It is used in particular for store and forward applications. Therefore, is must be possible to submit a file to the multicast server through the firewall from external networks.

History

Document history		
V1.2.2	December 2000	Publication as EN 301 790
V1.1.1	September 2001	Publication