ETSI TS 102 602 V1.1.1 (2009-01)

Technical Specification

Satellite Earth Stations and Systems (SES);
Broadband Satellite Multimedia;
Connection Control Protocol (C2P) for DVB-RCS;
Specifications

Reference

DTS/SES-00275

Keywords

broadband, control, DVB, multimedia, protocol, satellite

ETSI

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

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

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

Important notice

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

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Foreword

This Technical Specification (TS) has been produced by ETSI Technical Committee Satellite Earth Stations and Systems (SES).

Introduction

The present document is the first published version of a converged C2P specification for DVB-RCS systems, bringing together requirements for a range of different system scenarios. The present document is based on existing C2P implementation but with several important extensions and changes.

Readers of the present document are therefore advised that the present document may be subject to changes in order to incorporate any possible corrections that may result from the testing and verification of the first implementations of the present document.

1 Scope

The present BSM document defines and specifies a Connection Control Protocol (C2P) for DVB-RCS satellite networks, relying on specific DVB-RCS layer 2 signalling (DULM on the return link, unicast TIM on the forward link).

C2P is part of the control plane layer 2 functionality and is generally used for the dynamic establishment and management of connections between the ground elements of DVB-RCS networks (RCSTs, Gateways, NCC), regardless of their architectures and topologies (single-beam/multiple-beam architectures, transparent/regenerative satellite payloads, star/mesh topologies). In this context C2P can be seen as a complement to the functionality of the interfaces already defined in the DVB-RCS and DVB-S/S2 standards [1] and [i.1].

The present document is organized as follows:

- Clause 4 contains the description of the network reference scenarios (clause 4.2), of the C2P core elements (clause 4.3), of the connection types (clause 4.4) and of various data structures for dynamic connectivity support (clause 4.6).
- Clause 5 describes the state machines, including the timer and counter definitions and examples of state machines diagrams.
- Clause 6 describes the normal procedures and a few examples of exception procedures.
- Clause 7 describes the C2P Information Elements (IEs).
- Clause 8 defines the C2P messages.
- Annex A includes the description of C2P State Machines in UML format.
- Annex B includes examples of C2P scenarios.
- Annex C provides additional examples of exception procedures.
- Annex D includes examples of the formatting of C2P messages.

Further and more detailed background information for C2P definition and specification is provided in the C2P TR Background Information document [i.5].

The specifications in the present document apply to DVB-RCS network elements that are part of the same Interactive Network (IN). The RCSTs in the IN are Type A RCSTs (clause 8.1.1 in [1]). All RCSTs are capable of transmitting in ATM or MPEG2-TS formats and of receiving in MPEG2-TS format.

The present document defines Version 01 of the C2P Specifications. This is the first release of the C2P Specifications.

2 References

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

- For a specific reference, subsequent revisions do not apply.
- Non-specific reference may be made only to a complete document or a part thereof and only in the following cases:
  - if it is accepted that it will be possible to use all future changes of the referenced document for the purposes of the referring document;
  - for informative references.

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

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

2.1 Normative references

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

[1] ETSI EN 301 790 (v1.5.1): "Digital Video Broadcasting (DVB); Interaction channel for satellite distribution systems".

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

[3] ITU-T Recommendation H.222.0: "Information technology - Generic coding of moving pictures and associated audio information: Systems".

[4] ITU-T Recommendation I.361: "B-ISDN ATM layer specification".

2.2 Informative references

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

[i.1] ETSI EN 302 307: "Digital Video Broadcasting (DVB); Second generation framing structure, channel coding and modulation systems for Broadcasting, Interactive Services, News Gathering and other broadband satellite applications".

[i.2] ETSI TS 102 462: "Satellite Earth Stations and Systems (SES); Broadband Satellite Multimedia (BSM); QoS Functional Architecture".

[i.3] SatLabs System Recommendations (v1):"Quality of Service Specifications".

NOTE: Available at http://satlabs.org.

[i.4] SatLabs System Recommendations (v1): "Management and Control Planes Specifications".

NOTE: Available at http://satlabs.org.

[i.5] ETSI TR 102 603: "Satellite Earth Stations and Systems (SES); Broadband Satellite Multimedia (BSM); Connection Control Protocol (C2P) for DVB-RCS; Background Information".

[i.6] ETSI TR 101 790: "Digital Video Broadcasting (DVB); Interaction channel for Satellite Distribution Systems; Guidelines for the use of EN 301 790".

[i.7] ETSI TS 102 429-2: "Satellite Earth Stations and Systems (SES); Broadband Satellite Multimedia (BSM); Regenerative Satellite Mesh - B (RSM-B); DVB-S/DVB-RCS family for regenerative satellites; Part 2: Satellite Link Control layer".

[i.8] UML Specification v. 1.1 (OMG document ad/97-08-11).

NOTE: Available at http://www.omg.org.

[i.9] IETF RFC 1112: "Host Extensions for IP Multicasting".

[i.10] SatLabs System Recommendations (v2.1).

NOTE: Available at http://satlabs.org.

[i.11] AmerHis System, Interactive Broadband DVB-RCS/S OBP Communication System.

NOTE: Available at http://telecom.esa.int.

[i.12] IETF draft-combes-ipdvb-mib-rcs-04.doc: "The DVB-RCS MIB".

[i.13] IETF RFC 1901: "Introduction to Community-based SNMPv2".

[i.14] IETF RFC 3416: "Version 2 of the Protocol Operations for the Simple Network Management Protocol (SNMP)".

[i.15] IETF RFC 1518: "An Architecture for IP Address Allocation with CIDR".

[i.16] IETF RFC 4632: "Classless Inter-domain Routing (CIDR): The Internet Address Assignment and Aggregation Plan".

3 Definitions and abbreviations

3.1 Definitions

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

Class of Service (CoS): defines a specific behaviour regarding traffic handling/forwarding; can be used to categorize the traffic into different classes

connection: Layer 2 logical association between two or more network entities characterized by a C2P Class of Service (C2P CoS)

Connection Control Protocol (C2P): protocol that provides the interaction between RCSTs and NCC to support the set-up, modification and release of connections

control plane: part of a layered protocol architecture that, among other functions, is concerned with connection control functions, including the signalling necessary to set up, supervise and release connections

Digital Video Broadcasting Return Channel by Satellite (DVB-RCS): protocol for an interaction (or return) channel in satellite links

Digital Video Broadcasting via Satellite (DVB-S): protocol for broadcasting TV signals and, by extension, data over satellite

gateway: general term to identify both the TSGW and the RSGW

Interactive Network (IN): group of terminals serviced by an NCC

IP flow: sequence of IP packets from an IP source to an IP destination

NOTE: An IP flow may be identified based on the following attributes: IP source and destination address, protocol type, source and destination ports, class of service.

management plane: part of a layered protocol architecture that provides two types of functions, namely layer management and plane management functions

Management Station (MS): network element that manages all the elements of the system of one satellite interactive network (IN); it also controls the sessions, resources and connections of the ground terminals; it is composed of the NMC and the NCC

mesh connection: direct connection established between two RCSTs

multicast: communication capability, which denotes unidirectional distribution from a single source access point to a number of specified destination access points

Network Control Centre (NCC): network element that provide real time control of the IN (e.g. session control, connection control, routing, terminals' access control to satellite resources, etc.)

Network Management Centre (NMC): network element in charge of the management of all the system elements in the IN

Network Operation Centre (NOC): responsible for the centralized management and control functions in systems supporting multiple interactive networks, each controlled by its own NCC; NOC provides service and network (bandwidth) provisioning to the interactive network, co-ordination between NCCs, etc.

NOTE: In the case of a single interactive network the NOC and NCC functionality are merged.

On-Board Processor (OBP): router or switch or multiplexer in the sky; it can decouple the uplink and downlink air interface formats (modulation, coding, framing, etc.)

Quality of Service (QoS): network ability to provide service differentiation/guarantees and thus influence the perceived quality of communications with regard to a number of parameters (including delay, jitter, packet loss) that packets sent by the application experience when being transferred by the network

Return Channel Satellite Terminal (RCST): network element that provides the interface between the satellite system and external users

Regenerative Satellite Gateway (RSGW): network element in a regenerative satellite system that provides interconnection with terrestrial networks (Internet, ISDN/POTS and Intranet)

star connections: connections involving a gateway (TSGW in a transparent system or RSGW in a regenerative system)

NOTE: Star connections can involve one hop or double hop.

stream: logical flow of layer 2 data from one network reference point into the satellite network, resulting from the encapsulation of IP datagrams into MAC packets

Transparent Satellite Gateway (GW/TSGW): network element in a transparent satellite system that provides interfaces with terrestrial networks (Internet, ISDN/POTS and Intranet)

NOTE: The GW is typically integrated with the NCC in a single network element denoted as Hub.

user plane: user plane in a layered protocol architecture that provides the transfer of user data, along with associated controls (e.g. flow control, recovery from errors, etc.)

3.2 Abbreviations

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

AF	Assured Forwarding (DiffServ PHB)
ATM	Asynchronous Transfer Mode
AVBDC	Absolute Volume Based Dynamic Capacity
AVBDCRepTime	AVBC Repetition Time
ADR	Average Data Rate
BE	Best Effort (DiffServ PHB)
BoD	Bandwidth on Demand
BSM	Broadband Satellite Multimedia
C2P	Connection Control Protocol

C2P CoS	C2P Class of Service
C2P PDR	C2P Peak Data Rate (used in C2P request messages)
C2P SDR	C2P Sustainable Data Rate (used in C2P request messages)
Channel_ID	Channel Identifier
Channel_ID_NCC	Channel Identifier at NCC
Channel_IDxy	Channel identifier for MAC CoS y in the connectivity channel x
CL	Controlled Load (IntServ Class of Service)
Cnx	Connection
Cnx PDR'	(Admitted) Connection Peak Data Rate (used in C2P response messages)
Cnx SDR'	(Admitted) Connection Sustainable Data Rate (used in C2P response messages)
CnxProfile Entry	Connection Profile (Mapping Table) Entry
CnxProfile Index	Connection Profile (Mapping Table) Index
CnxRef ID	Connection Reference Identifier (used in the Active Connection Table)
CnxEstReq	Connection Establishment Request
CnxEstResp	Connection Establishment Response
CnxRelReq	Connection Release Request
CnxRelResp	Connection Release Response
CnxModReq	Connection Modify Request
CnxModResp	Connection Modify Response
CoS	Class of Service
CR	Capacity Request
CRA	Constant Rate Assignment
CSC	Common Signalling Channel
DAMA	Demand Assignment Multiple Access
DiffServ	Differentiated Services
DL	Down Link
DSCP	Differentiated Service Code Point
DSM-CC	Digital Storage Medium - Command and Control
DULM	Data Unit Label Method
DVB	Digital Video Broadcasting
DVB-RCS	Digital Video Broadcasting Return Channel Satellite
DVB-S	Digital Video Broadcasting by Satellite
DVB-S2	Digital Video Broadcasting by Satellite Transmission 2nd Generation
EF	Expedited Forwarding (DiffServ PHB)
ETSI	European Telecommunications Standards Institute
FCA	Free Capacity Assignment
FL	Forward Link
GW	GateWay
GRD	Guaranteed Rate & Delay
Group_ID	Group Identifier
GS	Guaranteed Service (IntServ Class of Service)
ID	Identifier
IE	Information Element
IETF	Internet Engineering Task Force
IN	Interactive Network
IntServ	Integrated Services
IP	Internet Protocol
IP@	IP address
IP CoS	IP Class of Service
IP PDR	IP Peak Data Rate (of an IP flow/flow aggregate)
IP SDR	IP Sustainable Data Rate (of an IP flow/flow aggregate)
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6
ISDN	Integrated Services Digital Network
ISP	Internet Service Provider
ITU	International Telecommunications Union
Kbps	Kilo bits per second (thousands of bits per second)
LAN	Local Area Network
LNM	Local Network Management
Logon_ID	Logon Identifier
MAC	Medium Access Control
MAC@	MAC address

MAC@rsc	MAC address of the source
MAC@dst	MAC address of the destination
MAC CoS	MAC Class of Service
Mbps	Mega bits per second (millions of bits per second)
MCD	Multi-Carrier Demodulator
MDR	Minimum Data Rate
MF-TDMA	Multiple-Frequency Time-Division Multiple Access
MIB	Management Information Base
MLD	Multicast Listener Discovery
MMT	Multicast Map Table
MPE	Multi-Protocol Encapsulation
MPEG	Moving Picture Experts Group
MS	Management Station
NCC	Network Control Centre
NMC	Network Management Centre
NOC	Network Operation Centre
OAM	Operation, Administration and Maintenance
OBP	On Board Processor
OSI	Open System Interconnection
PDR	Peak Data Rate
PHB	Per Hop Behaviour
Phb Entry	PHB (Mapping Table) Entry
Phb Index	PHB (Mapping Table) Index
Phb PDR'	PHB Peak Data Rate (admitted rate; used for the configuration of IP mechanisms)
Phb SDR'	PHB Sustainable Data Rate (admitted rate; used for the configuration of IP mechanisms)
PID	Program Identifier
PktClass Entry	Packet Classification (Table) Entry
PktClassIndex	Packet Classification (Table) Index
PSTN	Public Switched Telephone Network
PTM	Point-To-Multipoint
PTP	Point-To-Point
PVC	Permanent Virtual Circuit
QoS	Quality of Service
R1	First Rate (of a token bucket)
R2	Second rate (of a token bucket)
RBDC	Rate Based Dynamic Capacity
RBDCMax	RBDC Maximum (parameter)
RBDCTimeout	RBDC Timeout (parameter)
RC	Request Class
RC Entry	Request Class (Table) Entry
RC Index	Request Class (Table) Index
RCModReq	Request Class Modify Request
RCModResp	Request Class Modify Response
RCST	Return Channel Satellite Terminal
RF	Radio Frequency
RFC	(IETF) Request For Comments
RL	Return Link
RSAT	Regenerative SATellite
RSGW	Regenerative Satellite GateWay
RSVP	Resource ReSerVation Protocol
Route_ID	Route IDentifier
Rx	Receive
SCD	Single Carrier Demodulator
SCPC	Single Channel Per Carrier
SDP	Session Description Protocol
SDR	Sustainable Data Rate
SIP	Session Initiation Protocol
SLA	Service Level Agreements
SNMP	Simple Network Management Protocol
SSR	SatLabs Systems Recommendations
ST	Satellite Terminal
Stream_ID	Stream IDentifier

SYNC	SYNChronization (burst type)
TBTP	Terminal Burst Time Plan
TCP	Transmission Control Protocol
TDM	Time Division Multiplex
TIM	Terminal Information Message
TIMu	TIM unicast
TOS	Type Of Service
TRF	TRaFic (burst)
TS	Transport Stream
TSPEC	Traffic SPECification
TSGW	Transparent Satellite GateWay
TTL	Time To Live
Tx	Transmit
UDP	User Datagram Protocol
uimsbf	unsigned integer most significant bit first
UL	UpLink
VBDC	Volume Based Dynamic Capacity
VBDCMax	VBDC Maximum (parameter)
VBDCMaxBacklog	VBDC Maximum Backlog (parameter)
VBDCTimeout	VBDC Timeout (parameter)
VCC	Virtual Channel Connection
VCI	Virtual Connection Identifier
VLAN	Virtual LAN
VPI	Virtual Path Identifier

4 Connection Control Overview

4.1 Protocol High Level Functionality

Connection Control Protocol (C2P) is primarily used to enhance the control plane of DVB-RCS systems, by providing support for dynamic connectivity. More specifically, it is used to establish connections with pre-defined resources, and to modify or release these resources. To this end C2P relies on DVB-RCS specific signalling mechanisms, in particular the Data Unit Label Method (DULM) for return link (RL) signalling, and various descriptors in a unicast Terminal Information Table (TIMu) for forward link (FL) signalling. Within this broad scope the C2P also performs a number of additional C2P-related functions. The new features added by C2P to the control plane of DVB-RCS systems can be summarized as follows:

- Establishment/modification/release of connections between sets of communicating parties (network elements) in a DVB-RCS system (RCSTs, Gateway, NCC).
- QoS-driven dynamic allocation of bandwidth resources to connections, following the execution of a Connection Admission Control (CAC) function.
- Dynamic control of the communicating parties in the DVB-RCS system, via configuration parameters and policies.
- Dynamic allocation of logical resources to connections (PID or VPI/VCI).
- Configuration of the Route_ID.
- Assignment of the Channel_ID.
- Address resolution for the purpose of MPE encapsulation (i.e. the association between MAC addresses and IPv4 addresses of both parties involved in the connection).
- Definition of isolated and independent satellite sub-networks within the global interactive network (i.e. each subnetwork is characterized by its own terminal population, bandwidth resources, addressing space/plan).
- Dynamic multicast group management.

The above features apply to all DVB-RCS network reference scenarios, as identified in clause 4.2, and to various types of connections, as described in clause 4.3, which also defines other C2P relevant entities (channels, flows, etc.).

Since connections are established with adequate resources in order to satisfy the service requirements of various IP multimedia services and applications, C2P can be seen as part of the application (service) stratum of the overall Quality of Service (QoS) model defined in the BSM QoS Functional Architecture document (figure 5.1 in [i.2]). As such, it can be used as a vehicle for conveying QoS-related parameters for setting up the DVB-RCS network for each media component. The QoS functionality in the transport stratum at both network (IP) layer and link (MAC) layer is consistent with that defined in the SatLabs recommendations for QoS Specifications for DVB-RCS networks [i.3]. Moreover, the SatLabs Group has also provided recommendations for the Harmonized Management and Control Plane Specifications in DVB-RCS networks [i.4]. Every effort has been made to provide consistency of the C2P with these M&C specifications. However, some enhancements of the SatLabs-defined data structures [i.10] are required, in order to support the C2P functionality; they are identified where appropriate.

4.2 DVB-RCS Network Reference Scenarios

Network reference scenarios are in general defined based on the following criteria:

- Network architecture (number of spot beams in the system):
  - Single-beam.
  - Multi-beam.
- Network topology:
  - Star.
  - Mesh.
- Satellite payload architecture:
  - Transparent (or transponded).
  - Regenerative.

The following distinct network reference scenarios have been identified for the purpose of C2P specification:

- Single-beam/multi-beam, star transparent.
- Single-beam/multi-beam, mesh transparent.
- Single-beam/multi-beam, star/mesh regenerative.

C2P is particularly relevant to mesh scenarios, either transparent or regenerative, since it allows setting up mesh connections for one-hop communication between terminals. The term "transparent" refers both to a pure transparent satellite payload (bent-pipe) and also to a satellite payload which does not perform any demodulation but is able to support some form of physical layer switching (e.g. analogue/digital channel switching, with channels associated with carriers or sets of carriers). The former only allows mesh communications between terminals in the same beam, while with the latter the terminals can be located in different beams. A transparent system that supports mesh communications (part of a mesh network) also supports star communications (part of a star network), under the control of a unique Network Control Centre (NCC). The transparent mesh network can be seen as an overlay to the transparent star network. The NCC will allow setting up not only mesh connections for user traffic between terminals, but also star connections for signalling between terminals and NCC. The star connections and the corresponding physical channels will typically be operational before a particular mesh connection is required. The mesh terminals shall be capable of receiving and demodulating not only the DVB-S/S2 forward channel from NCC, but also the DVB-RCS return channels from other terminals. To this end the mesh terminals shall be equipped with MF-TDMA burst demodulators. The number of return channels (and consequently of mesh connections) that can be supported by a terminal depends on whether the demodulator is a Multi-Carrier Demodulator (MCD) or a single carrier demodulator (SCD). In the latter case, at a given time a terminal can only receive mesh traffic from the terminals transmitting on the frequency on which its SCD is tuned. This may limit the mesh connectivity between the terminals in the system, leading to the concept of partially meshed networks. A fully meshed network, by contrast, would allow mesh connectivity between all terminals in the system, subject to various other constraints (e.g. terminal transmit rate).

With regard to the mesh regenerative scenarios, special attention is be given to systems from the Regenerative Satellite Mesh - B (RSM-B) family [i.7], since the original C2P has been defined for a system from this family, namely the AmerHis system [i.11]. The current C2P specification is much broader in scope, since it applies to all network reference scenarios. Nevertheless, it takes into consideration the specifics of RSM-B systems, e.g. the fact that the NCC (and also the regenerative gateway) are members of the mesh network, i.e. they behave like terminals as far as the physical air interface is concerned.

A rather detailed description of each network reference scenario, as applicable to DVB-RCS networks, is provided in the C2P TR High Level Design and Guidelines document [i.6]. The description is in terms of architectures and the corresponding network elements. The key features of each network element are summarized below. With the exception of the NOC, all network elements are considered part of the same interactive network.

- *Return Channel Satellite Terminal (RCST)*: The RCST (or simply the terminal) is the interface between the Satellite System and external users. A terminal can in general be configured (by simple software change) to work in either transparent or regenerative systems. However, for some network scenarios (e.g. transparent mesh networks) hardware changes are also required (compared to a conventional RCST used in star transparent networks).

- It is expected that the RCST is able to provide service differentiation/guarantees based on standard IP QoS architectures/models (e.g. DiffServ, IntServ). This implies user plane functions (i.e. traffic conditioning/forwarding) and M&C plane functions (e.g. configuration/control of the DiffServ mechanisms and of other QoS-aware components).

- *Transparent Satellite Gateway (GW or TSGW)*: It operates in transparent satellite systems and provides interfaces with external networks, primarily for internetworking with Internet/Internet Service Provider networks (via a backbone network), but also with the telephony oriented networks (such as PSTN or ISDN) and with Intranets.

- *Regenerative Satellite Gateway (RSGW)*: It operates in regenerative satellite systems and provides interconnection with terrestrial networks (Internet, ISDN/POTS and Intranet).

- In some systems (e.g. RSM-B systems) the RSGWs are implemented as terminals as far as the air interface/RF equipment is concerned, i.e. they support modulation and demodulation functions similar to those of an RCST. The RSGW satellite terminals are referred to as GW-RCSTs; they are conventional RCSTs (see above) but with extra functionalities.

- Other systems may adopt different implementations for the RSGW. The present document only takes into account the RSGW implementation specific to RSM-B systems.

- Both GW and RSGW are expected to provide service differentiation/guarantees to subscribers based on the same QoS models as the RCST, i.e. DiffServ model, IntServ model.

- *Management Station (MS)*: The MS manages all the elements of the system. It also controls the sessions, connections and resources of the ground terminals. The MS can be logically (and even physically) decomposed in two modules.

- *Network Control Centre (NCC)*: The NCC controls the IN, provides session control, connection control, routing and resource access to the subscribers' RCSTs. It also manages the OBP configuration.

- In the transparent network scenarios the NCC is typically collocated with the GW and implemented as a single network element, referred to as NCC/GW or Hub.

- In some regenerative systems (e.g. RSM-B systems) the NCC is associated with a terminal, referred to as NCC-RCST. The NCC-RCST is the satellite terminal for the entire MS, supporting modulation and demodulation functions similar to those of an RCST.

- *Network Management Centre (NMC)*: The NMC is in charge of the management of all the system elements.

- *Network Operation Centre (NOC)*: The NOC is responsible for centralized management and control functions such as service and network (bandwidth) provisioning to the interactive network. It is used when the satellite system supports multiple interactive networks, each under the control of its own NCC. The coordination between NCCs is the responsibility of the NOC. In the case of a single interactive network the NOC and NCC functionalities are merged.

- *On-Board Processor (OBP)*: The OBP behaves as a router, switch or multiplex in the sky. It provides regeneration functions that can decouple the uplink (UL) and downlink (DL) air interface formats (modulation, coding, framing, etc.). The air interface formats can be either DVB-RCS or DVB-S/S2, depending on the network reference scenario and on the links involved.

- *Gateway*: It is a general term used to refer to either a GW/TSGW or an RSGW.

4.3 Definition of C2P Elements

A detailed definition of all C2P core elements has been provided in the C2P TR High Level Design and Guidelines document [i.5]. The document also introduced a general C2P model, applicable to all network reference scenarios, capturing the inter-relationships/interdependencies between various C2P elements and QoS specific elements. The model is an essential tool for defining the connection parameters and the structure/content of the C2P messages.

The definitions of the C2P elements are summarized below, for convenience.

4.3.1 Connection

A connection is understood as a layer 2 logical association between two entities or reference points required for traffic transmission between two communicating parties. The reference points constitute the end-points of the connection. In the case of multicast connections, one entity is a virtual entity representing a multitude of end-points.

For the present document all reference points of a connection are associated with network elements located on ground (RCST, RSGW, NCC/GW). Connections with one reference point at the satellite (OBP) are thus excluded. Also excluded are the connections involving more than one satellite hop.

The transmission of traffic associated with a connection is consistent with the connection's Class of Service (CoS), referred to as C2P CoS and equivalent to a MAC Class of Service (MAC CoS). Additional information can be found in [i.5]. The traffic is generated by a source and is intended for a destination. The source and destination, as communicating parties, are associated with the two reference points of the connection, therefore one can refer to source/destination reference points or parties. The primary User plane connotation of the source and destination is thus extended to the Control plane and used from the C2P point of view. The conventions regarding the use of source and destination in the context of different connection types are defined in clause 4.4.2.2.

A connection is defined by the combination of the following attributes:

- A pair of RCST MAC addresses (source and destination, for the two reference points of the connection).

- C2P Class of Service (C2P CoS).

- VPI/VCI pair (ATM format) or a PID (MPEG2-TS format), used for encapsulation/reassembly at the connection's reference points.

NOTE 1: In the case of MPEG2-TS format the destination MAC address is included in the DSM-CC header during the encapsulation process.

NOTE 2: For multicast connections, the destination MAC address for each end-point of the connection is derived from the IP multicast address.

NOTE 3: For connections triggered by an incoming IP packet (see clause 7.1 in [i.5]), the destination IP subnet of that packet may also be considered as an attribute of the connection.

NOTE 4: The VPI/VCI pair is alternatively referred to in the present document as VCC.

At the level of the entire network and for the purpose of connection management by the NCC (e.g. for exchanging messages relative to a connection), a connection can be uniquely identified by a pair of Connection Reference IDs.

A Connection Reference ID is used to "lump" under a unique logical identifier all connection attributes defined above. It will be used in all C2P messages relative to a particular connection. The Connection Reference ID can be set by the RCST/RSGW or by the NCC (as connection initiators), to values taken from two disjoint pools. When set by the NCC, the Connection Reference ID is uniquely defined in the entire network controlled by the NCC. When set by the RCST/RSGW, the Connection Reference ID has local meaning, allowing the RCST/RSGW to locally identify all its active connections in the network. For un-ambiguous identification at the network level, the Connection Reference ID set by an RCST/RSGW shall be extended (complemented) by the identity of the RCST/RSGW (e.g. the MAC address).

In the case of connections between two RCSTs (or RSGWs), e.g. RCST A and RCST B, the C2P signalling involves two legs: one signalling leg between the RCST A and NCC, and another signalling leg between the NCC and RCST B.

If the connection is initiated by NCC, the NCC will assign two Connection Reference IDs, one for each leg, which together uniquely identify the connection.

If the connection is initiated by one RCST (e.g. RCST A), the Connection Reference ID set by the RCST A, complemented by the identity of RCST A, will be used for all C2P messages exchanged between RCST A and NCC, while an NCC-set Connection Reference ID will be used for all C2P messages between NCC and RCST B.

In summary and more accurately, a connection can be uniquely identified in the entire network by:

- a pair of Connection Reference IDs, both set by the NCC, in the case of NCC-initiated connections; or
- in the case of RCST initiated connections, by a pair of Connection Reference IDs, one set by the RCST, plus the RCST MAC address, and another one set by the NCC.

The above attributes of a connection are defined from the point of view of the source reference point of the connection, where the transmission originates. They are therefore transmission attributes/parameters, and the definition in this clause applies to a unidirectional connection. Reception attributes/parameters need also be specified in the case of a bidirectional connection, seen as two unidirectional connections (clause 4.4).

A connection carries layer 2 packets in the form of streams (clause 4.3.2). The packets in a stream are from one or several IP flows (clause 4.3.3).

A connection of given type (clause 4.4) is established with a defined C2P CoS within a connectivity channel (clause 4.3.4).

4.3.2 Stream

In the context of C2P a stream is understood as a logical flow of layer 2 (MAC) data packets from one network reference point (e.g. RCST, NCC/GW or RSGW) into the satellite/DVB-RCS network. The notion of stream is therefore linked to the transmission satellite interface of the (source) reference point. The MAC data packets result from the segmentation and encapsulation of IP datagrams.

A stream is uniquely identified, for the purpose of segmentation/encapsulation and reassembly at the connection reference points, by a Stream_ID (as logical identifier), and all data packets in the stream share the same Stream_ID. The Stream_ID can be either:

- a {PID, Destination MAC address} pair, for the MPEG2-TS format; or
- a {VPI/VCI} pair in the case of ATM format.

The PID/Destination MAC address pair or the VPI/VCI pair used for stream identification is the same as that included in the set of attributes of the underlying connection.

4.3.3 IP Flow

An IP flow is a sequence of IP packets from an IP source to an IP destination. The packets can be associated with an elementary flow generated by an application, or with an aggregation of elementary flows sharing some common attributes (e.g. same service class/PHB, same source/destination addresses, same source/destination port number, same protocol type, same DSCP).

IP flows are carried as part of the streams associated with different connections, therefore a flow can be associated (by inheritance) with the connection's parameters. A connection can carry one or multiple IP flows.

4.3.4 Channel and Channel_ID

The channels are used in the context of DVB-RCS networks for the purpose of dynamic resource control (DAMA scheduling). As such, they refer to pools of return link (uplink) resources in a user beam (timeslots in the MF-TDMA frame), identified in RCSTs and NCC by different Channel_ID values. A Channel_ID value is used by the RCST to tag the capacity requests relative to a given channel, and by the NCC (DAMA Scheduler) to tag the capacity allocations (in TBTP) relative to the same channel.

The DVB-RCS standard [1] allows the use of channels (and Channel_IDs) for the purpose of either connectivity or QoS differentiation, or both. The present document will therefore distinguish, where appropriate, between "connectivity channels" and "QoS channels". The latter will preferably be designated as QoS classes, in the sense of MAC Classes of Service (MAC CoSs), *which are a reflection at the MAC layer of the classes of service implemented at IP layer (i.e. the DiffServ's PHBs/IntServ's CoSs)*. For the QoS model adopted for C2P Specifications please refer to clause 7.2.1 in [i.5].

Connectivity channels correspond to RL/UL physical partitions associated with different destination downlinks. In general, a partition may include slots anywhere in the MF-TDMA frame, but constraints are typically imposed regarding their location in order to simplify the resource configuration and control.

The "QoS channels" used for QoS differentiation correspond to logical partitions of the RL/UL resources, associated with different QoS classes (i.e. MAC CoSs), seen from the RCST point of view as Request Classes (RCs), according to SatLabs QoS model/nomenclature [i.3]. The partitions are defined in number of slots (i.e. they are scalars). Multiple QoS partitions are typically defined within a connectivity channel.

Each channel, whether defined for connectivity or QoS differentiation, is uniquely identified in an uplink user beam. The Channel_ID coding was extended from 4 bits to 8 bits in version 1.5.1 of the DVB-RCS standard [1]; as a result of this extension a maximum of 256 channels can be configured in a beam, for either connectivity or QoS differentiation. In system designs that allow the use of Channel_ID for both connectivity handling and QoS differentiation at the same time, the NCC and RCSTs should be aware of the usage of each configured channel and its Channel_ID (for connectivity or for QoS). A connectivity channel can be uniquely identified by the list of Channel_IDs assigned to the MAC CoSs supported within that connectivity channel, provided that they are unique in the beam (i.e. no Channel_ID is assigned to more than one MAC CoS defined for the beam).

4.3.5 Route and Route_ID

The Route_ID has been originally introduced in the DVB-RCS standard to facilitate "label switching" on board of regenerative satellites of uplink packets to destination downlinks; as such it identifies a destination (forward) downlink. In version 1.5.1 of the DVB-RCS standard its definition has been extended to allow for other usages, e.g. to indicate a connectivity channel, which is associated with a destination downlink as well, as defined in the present document (clause 4.3.4). In both cases the Route_IDs are used in association with routes. A route is a path for packet transmission from an uplink beam to a destination downlink.

The Route_ID is defined as a 16-bit subfield in the SAC field and in the Information Element (IE) used with the DULM method.

There is no conflict between the two usages of the Route_IDs, since the Route_IDs are not needed for both label switching and connectivity channels in the same system. On one hand, the Route_IDs are not needed for on-board label switching in the case of regenerative satellites based on routing/switching tables and in the case of transparent satellites. In such cases the corresponding bits in the Route_ID subfield (or a subset thereof) could be used for other purposes, e.g. for connectivity channels. In the context of C2P for example the Route_ID can be used to identify the list of Channel_IDs used for the MAC CoSs defined within a connectivity channel (clause 4.3.4). This information can be exploited by the RCST to optimize the filling of the slots assigned for a Request Class/MAC CoS (and tagged with the corresponding Channel_ID) with packets from other Request Classes within the same connectivity channel, according to SatLabs QoS specifications [i.3] (i.e. the unused slots assigned to a Request Class can be used for other Request Classes tagged with Channel_IDs grouped under the same Route_ID). Such usage of the Route_ID is implementation specific and is completely transparent to the satellite payload (i.e. the Route_ID is not extracted/used in the OBP). On the other hand, in systems supporting label switching (based on Route_IDs) there is no need for connectivity channels, therefore the usage of Route_IDs in such systems is completely transparent to the ground equipment (e.g. RCSTs).

4.4 Connection Types

In general the connections are defined for star and mesh topologies, for traffic or signalling. Furthermore, the traffic connections can be set-up by the NCC (NCC-initiated connections), by the RCST (RCST-initiated connections) or by the RSGW (RSGW-initiated connections), and can be of various types: point-to-point unidirectional or bidirectional connections, point-to-multipoint connections.

The various categories of connections are reviewed in the following clauses.

4.4.1 Star and mesh connections

Connections are differentiated in star and mesh, depending on whether they involve a Gateway or not:

- A Star connection is established between an RCST and a Gateway (NCC/GW or RSGW).
- A Mesh connection is established between two RCSTs, without the involvement of the Gateway (NCC/GW or RSGW).

NOTE: A connection between two RSGWs could be considered as star or mesh connection, depending on the context. For the purpose of definition of C2P messages the connection between two RSGWs will be assimilated with a mesh connection.

This above differentiation applies to both transparent and regenerative satellites that can be configured to support mesh and star topologies. In general, a network supporting mesh connections also supports star connections simultaneously (e.g. mesh overlay), primarily for signalling (e.g. to/from NCC) but also for access traffic (to/from Gateway).

In the case of star networks all traffic connections are of star type.

In the case of mesh networks the traffic connections are primarily of mesh type (between RCSTs), but can also be of star type (between RCSTs and NCC/GW or RSGW).

4.4.2 Signalling and traffic connections

The differentiation between:

- Signalling connections; and
- Traffic connections.

is based on the type of data conveyed by each connection.

Signalling connections convey various signalling messages (e.g. protocol-related messages, management and control messages, etc.).

Traffic connections convey user traffic.

Signalling connections could be mixed with traffic connections in the same connectivity channel and/or QoS class.

4.4.2.1 Signalling connections

Signalling connections are established to convey signalling messages between RCSTs and the Management Station (NMC or NCC), such as:

- Management messages to/from NMC (e.g. SNMP messages).
- C2P and other control messages to/from NCC in DULM/TIMu formats.
- Other messages associated with end-to-end signalling (e.g. for application session establishment, QoS management etc) to/from various servers in the system.

Separate connections could be setup for various signalling messages (e.g. for control messages, for management messages, for end-to-end messages). At least one default signalling connection should be set-up as part of the terminal logon process, based on information included in the logon messages received by the RCST (clause 4.6.1). In a system supporting C2P, this connection can then be used to exchange C2P messages in order to establish new connections, including signalling connections.

4.4.2.2 Traffic connections

Traffic connections are established to convey user data. They are established with pre-defined bandwidth parameters consistent with the C2P profile (clause 4.4.3), in order to ensure the expected service level for user data; their establishment therefore involves a Connection Admission Control (CAC) function. The C2P profile can be altered/renegotiated during the life of the connection (e.g. by relying on the C2P Connection Modify command), in response to changing traffic profile or as a result of new applications becoming active.

Traffic connections can be classified according to different criteria:

- Depending on the network entity that requested the connection, one can distinguish between two types of connections:
  - NCC-initiated;
  - RCST/RSGW-initiated.

The NCC-initiated connections are typically set-up during RCST/RSGW's logon to the network, but can also be dynamically established after the RCST/RSGW has been logged-on to the network. They are fully controlled by the NCC, i.e. they are set-up and released by the NCC and cannot be released at an RCST's request.

The RCST/RSGW-initiated connections are established and released dynamically after the RCST/RSGW has been logged-on to the network, upon explicit requests from RCST/RSGW, triggered by various events (e.g. arrival of an application packet, interception of an application session initiation message, etc. - see clause 7.1 in [i.5]). They may also be released upon explicit requests from the NCC, e.g. to solve an exception case (i.e. an unexpected event, see note 1).

NOTE 1: An exception case is defined as an unresolved or unstable situation of the C2P protocol due to an unexpected exchange/behaviour of messages, such as messages loss, cross request of connections (the same connection being requested at the same time from both sides of the connection), etc.

NOTE 2: The RCST-initiated and RSGW-initiated connections have been considered together, since in the RSM-B regenerative systems considered in the present document (see clause 4.2) the RSGW is implemented as a conventional RCST with additional functionalities. The RSGW-initiated connections are therefore equivalent to RCST-initiated connections from C2P point of view; the exchange of C2P messages is the same in both cases.

NOTE 3: The above definitions and interpretations apply to unicast connections. In the case of multicast, the RSGW-initiated multicast connections and RCST-initiated multicast connections are separately identified (as different connection types) in the C2P messages (clause 7.2.2.6).

With regard to the type of casting, the connections can be classified as unicast or multicast:

- A unicast connection is a point-to-point connection.
- A multicast connection is a point-to-multipoint connection.

A **unicast connection** can be established:

- between an RCST and an NCC/GW (transparent scenario); or
- between an RCST and an RSGW (regenerative scenario); or
- between two RCSTs (both transparent and regenerative scenarios) or two RSGWs (regenerative scenario).

In the case of NCC-initiated unicast connections between two RCSTs, two RSGWs or between an RCST and an RSGW, the two parties involved in the connection will be designated as A and B in the C2P procedures (i.e. RCST/RSGW A and RCST/RSGW B). They can both be regarded as peer parties, therefore the "peer" qualifier does not provide any differentiation between the two parties. In the description of C2P messages the two parties will be simply referred to as "RCST/RSGW" (i.e. RCST/RSGW A or B) and "the other end RCST/RSGW" (i.e. RCST/RSGW B or A).

In the case of RCST/RSGW-initiated unicast connections, the two parties involved in the connection will be designated as "initiating RCST/RSGW" and "peer party" (NCC/GW or RCST/RSGW), i.e. the party at the other end of the connection.

Regardless of the connection initiator, the unicast source and unicast destination are in general defined from the point of view of each party (NCC/GW or RCST/RSGW): the source (of the transmission) is associated with one party's (NCC/GW or RCST/RSGW) address and mask on its Ethernet interface, while the destination is associated with a subnet of the party (RCST/RSGW or NCC/GW) at the other end of the connection.

NOTE 4: Unicast connections can be unidirectional or bidirectional (see below). In the case of a unidirectional connection the source and destination are only defined with regard to the party (RCST/RSGW or NCC/GW) that transmits the unicast traffic.

A unicast connection carries IP elementary flows/flow aggregates over a stream, with an IP destination address which is part of an IP Unicast Group addresses (subnet).

A **multicast connection** can be established:

- from the NCC/GW towards many RCSTs (star transparent scenario); or
- from an RSGW towards many RCSTs and possibly other RSGW(s) (regenerative scenario); or
- from an RCST towards other RCSTs (mesh regenerative or transparent scenarios) and possibly the NCC/GW (mesh transparent scenario) or RSGW(s) (mesh regenerative scenario).

In all multicast cases and from a Control plane perspective, the network element/party which transmits the multicast traffic is referred to as multicast source, while the other parties are referred to as multicast destination parties.

- In the case of regenerative scenarios or transparent scenarios with some form of on-board physical layer switching (see clause 4.2), the multicast destination RCSTs and RSGWs (regenerative scenario) or NCC/GW (transparent scenario) can be at different destination downlinks, consistent with the configured connectivity.
- In the case of pure transparent mesh scenarios, all multicast destination RCSTs and the NCC/GW should be at the same destination downlink, reachable from the multicast source RCST. Two single-hop connections need to be established in the case where the RCSTs are at different destination downlinks: a unicast connection from the multicast source RCST to the NCC/GW, and a multicast connection from the NCC/GW to the multicast destination RCSTs. This case is referred to as "two-hop mesh transparent multicast". It is noted that the first connection could also be a multicast connection, if some of the multicast destination RCSTs were located in the same beam as the multicast source RCST.

A multicast connection carries multicast sessions, consisting of IP elementary flow(s) over a stream, with an IP destination address which is part of an IP Multicast Group addresses.

- With regard to the direction, a unicast connection can be further classified in one of the following types:
  - Unidirectional connection.
  - Bidirectional (or duplex) connection.

NOTE 5: The multicast connections are always unidirectional.

A bidirectional connection corresponds to two unidirectional connections (in opposite directions). An RCST, as an end-point of a bidirectional connection, must know the transmission and the reception parameters of the two unidirectional connections (the reception parameters at one end of the connection correspond to the transmission parameters at the other end-point and vice-versa). The two unidirectional connections could be identified by a unique pair of Connection Reference IDs (see clause 4.3.1), provided that the QoS architectures at the two ends of the connection and the C2P CoS of the two connections were the same (which is most likely, at least in the case of mesh connections), and the transmission/reception parameters could be given in a single set of C2P message exchange. If this were not the case, each unidirectional connection should be identified as a separate connection (with its own pair of Connection Reference IDs) and two sets of C2P message exchanges will be needed.

In the case of bidirectional connections originating at an RCST/RSGW, the two directions of the connection will be designated as "return" and "forward" from the point of view of the RCST/RSGW, understood as transmit direction and receive direction, respectively.

4.4.3 C2P connection profile

The C2P connection profile includes a set of parameters/attributes used in the C2P request messages for the establishment or modification of connections of given types. The parameters/attributes are as follows:

- Connection type:
  - Point-to-point (unicast) or point-to-multipoint (multicast).
  - Unidirectional or bidirectional.
  - RCST/RSGW-initiated or NCC-initiated.
- Class of Service of the connection (C2P CoS).

  The class of service of a connection shall be mapped to one of the classes of service supported at MAC layer (i.e. MAC CoSs).

  In the case of connections originating at an RCST or RSGW (for which the transmission is based on the MF-TDMA access schemes), the MAC CoSs are referred to as Request Classes (RCs), according to the SatLabs' QoS model/nomenclature [i.3] (clause 4.3.4). All active connections with a specified C2P CoS will be mapped to the same Request Class.

  In the case of connections originating at the NCC/GW (for which the transmission is based on the TDM access scheme used on the forward link), the MAC CoSs correspond to the QoS model adopted for the forward link (implementation specific).
- QoS Profile, consistent with the C2P CoS, defined in terms of connection transmission and reception bandwidth parameters (rates), namely:
  - Forward Sustainable Data Rate (SDR)/Peak Data Rate (PDR).
  - Return Sustainable Data rate (SDR)/Peak Data Rate (PDR).
- Connection inactivity timeout, used to trigger the connection release if there is no traffic activity.

At each RCST, the above information is stored in the Connection Profile Mapping Table (clause 4.6.3.2), which includes one entry (one set of parameters) for each defined connection profile. The C2P profile parameters will be configured by management for all defined connection profiles.

4.5 Overview of C2P messages

This clause gives an overview of the C2P messages. The complete, detailed definition of C2P messages is provided in clause 8, in terms of Information Elements. The Information Elements, containing various C2P parameters, are defined in clause 7.2.2.

Table 4.1 describes the messages supported by the Connection Control Protocol. It also provides a short description (overview) of each message and a reference to the clause where the message is described in detail.

Table 4.1: C2P messages

Formal C2P message name	Description	Reference clause
Connection establishment request	Request for the establishment of a new connection	See clause 8
Connection establishment response	Response to a new connection establishment request.	See clause 8
Connection release request	Request for the release of an existing connection	See clause 8
Connection release response	Response to a connection release request.	See clause 8
Connection modify request	Request for the modification of an existing connection	See clause 8
Connection modify response	Response to a connection modification request.	See clause 8
Connection modify join request (see note)	Request to join a multicast connection	See clause 8
Connection modify join response (see note)	Response to a connection join request.	See clause 8
Connection modify leave request (see note)	Request to leave a multicast connection	See clause 8
Connection modify leave response (see note)	Response to a multicast connection leave request.	See clause 8
RC modify request	Request for RC parameters modification	See clause 8
RC modify response	Response to RC parameters modification request	See clause 8
RCST capability request	RCST notification to NCC of RCST capability, or NCC request for RCST capability	See clause 8
RCST capability response	NCC acknowledgment of RCST capability notification, or RCST response to an RCST capability request	See clause 8
Connection status stats request	Enquiry for the status/statistics of a connection	See clause 8
Connection status stats response	Response to a connection status/statistics request	See clause 8
NOTE: The C2P join/leave multicast messages are related to dynamic multicasting and are included for the sake of completeness. They will not be defined in detail in clause 8 of the present document, which only applies to static multicasting. They may be included in future versions of the document.		

Depending on the direction, the Connection Control Protocol is supported by the following types of signalling messages:

- **From RCST/RSGW to NCC:** DULM messages (including the RCST identification through its Group_ID and Logon_ID assigned at logon, and a number of Information Elements).

- **From NCC to RCST/RSGW:** TIM unicast messages embedded in DSM-CC private sections (including the RCST identification through its MAC address contained in the DSM-CC header, and the Connection Control Descriptor).

4.6 Data structures and dynamic connectivity support

Various data structures and tables are used to store the information necessary to support dynamic connectivity, i.e. to dynamically setup, modify and release connections. Some of the information can be handled via structures already defined as part of DVB-RCS forward and return link signalling, in the forms of fields, flags and tables. They are used during the RCST logon process to establish default signalling connection(s) to the RCST (clause 4.6.1).

Other data structures are specifically defined for C2P-based dynamic connectivity support, in the form of a number of tables configured in RCST (clause 4.6.3). These tables can be used in conjunction with a number of flags associated with C2P functionality (clause 4.6.2).

All data structures defined in this clause for the RCST are also applicable to the RSGW, even if not specifically indicated for each data structure. The RSGW, as an RCST with extended functionality, will also be subject to the logon process, similar to a conventional RCST.

4.6.1 C2P logon information

The logon is typically initiated by the RCST, which sends a logon request to NCC as a CSC burst. The RCST signals to the NCC its capability to support dynamic connectivity by setting to "1" the "dynamic connectivity" flag in the CSC burst, as per DVB-RCS standard [1]. This indicates that the RCST supports not only the C2P, but also the extended Channel_ID format (1 byte) and the corresponding Capacity_Request_Format/Capacity_Request_Type (as per version 1.5.1 of [1]), and also the C2P return signalling by the DULM method. Additional RCST capabilities related to C2P operation (clause 7.2.2.16) are signalled to the NCC via a dedicated C2P message (clause 8.2.6), immediately after RCST logon (see below).

Upon receiving the CSC burst identifying the RCST (by its MAC address) and defining its standard capabilities, the NCC responds with a unicast Terminal Information Message (TIMu) (using the DSM-CC private section mechanism), after performing an admission control function, i.e. after checking the subscriber identification, the access rights and system resources availability.

In the case of successful admission, the TIMu message contains all the necessary information for the RCST logon into the system and for the establishment of default signalling connection(s) for control and management messages. The successful acceptance of the RCST into the system is signalled by setting to "0" the following flags in the RCST (network) status in the TIMu:

- Logon_fail_(busy).
- Logon_denied.
- Transmit_Disable.

In the case of unsuccessful acceptance, the RCST will retransmit the CSC bursts. CSC burst retransmission is governed by the following fields in the Contention Control Descriptor transmitted in a broadcast TIM (TIMb):

- CSC-response timeout.
- CSC-max-losses.
- Max-time-before-entry.

The minimum information needed for the establishment of default signalling connection(s) will be provided via the following descriptors, defined according to EN 301 790 [1] and carried in the TIMu:

- **Logon Initialize Descriptor**, that contains the logon information, in particular:
  - Group ID and Logon ID;
  - data_unit_labeling_flag set to "1";
  - return and forward signalling VPI/VCI (ATM profile) or Return CTRL_MNGM_PID (MPEG2-TS profile), to be used for RCST to NCC C2P message signalling.

NOTE 1: In the case of MPEG2-TS profile, the Return_TRF_PID included in the logon information is not used by the RCST to transmit traffic corresponding to connections established by using the C2P.

The Logon Initialize Descriptor allows the establishment of one default return signalling connection for both control (e.g. C2P messages) and management (e.g. SNMP messages). If separate signalling connections need to be established for management and control (e.g. to different logical network entities), the following approach can be used:

- In the case of MPEG profile, the Return CTRL_MNGM_PID could be used for the control signalling (including C2P messages) connection and another PID could be defined for the management signalling connection. An additional PID could be defined to separate C2P signalling from the general control signalling, if needed.

  In order to avoid any confusion, two new PIDs are defined: the Return_MNGM_PID, to be used for the management signalling; and the Return_CTRL_PID, to be used for the control signalling (including C2P messages).

- In the case of ATM profile, the return control connection and management connection can be differentiated by using different codes in the 3-bit PT field in the data unit header.

NOTE 2: In ATM networks the MSB in the PT field is set to zero for all user data and to 1 for control data. This allows four types of control data, of which only three are currently assigned (two for OAM and one for resource control). The fourth is reserved and could be used for any other local signalling. In the context of DVB-RCS networks based on IP connectivity (Type A terminal, according to [1]), all signalling based on ATM format is local (i.e. is limited to the satellite network), therefore the control codes can be freely used.

NOTE 3: An alternative solution is to define an additional VPI/VCI pair for signalling (e.g. for management, while reserving the default VPI/VCI for control), but this is not necessary, since the discrimination between management and control messages based on PT codes is simpler and more economic. This alternative solutions is therefore specifically excluded from the version of the C2P specified in the present document.

Network Layer Information Descriptor (NLID), used to provide the terminal with the additional information needed for the establishment of two separate signalling connections, one for management (e.g. SNMP messages) and another one for control (e.g. C2P messages). The information is included in the message body of the NLID (table 4.2).

Table 4.2: NLID format

Syntax	No. of bits	
	Reserved	Information
Network_layer_info_descriptor(){		
Descriptor_tag		8
Descriptor_length		8
Message body (see note)		255x8 max
}		
NOTE: The message body of the NLID, according to EN 301 790 [1], shall be formatted as an SNMP message intended for updating the RCST MIB (e.g. the SatLabs defined DVB-RCS MIB for terminals [i.12]). The message size should not exceed 255 bytes and it should preferably fit within a single TS packet. The message body is formatted according to RFC 1901 [i.13] and RFC 3416 [i.14], and the PDU type shall be a SetRequestPDU. Several Variable Bindings with objects defined in the RCST MIB may be included in the Variable Bindings list of NLID, depending on the capabilities of the RCST and the services requested by the RCST.		

The minimum information required for the management signalling connection and control signalling connection is identified in table 4.3 and is defined in MIB format. The prefixes of various parameters are as defined in the tables in which they will be used (e.g. Active Connection Table, Request Class Table). For generality, separate logical and physical (bandwidth) resources have been included in the Network Layer Information Descriptor for the management and control signalling connections. The resources, and the corresponding parameters, are grouped (and indexed) per signalling connection: index (1) is used for the management connection, while index (2) is used for the control connection. If only one signalling connection is used for both management and control/C2P (in which case the resources are shared), the NLID will include only one set of parameters, indexed as (1).

Table 4.3: NLID contents

NLID Parameters	No. of bits		Value/Comment
	Reserved	Information	
ActiveCnxRefId (1)		16	Reference ID of the MNGM active connection (referenced from NCC, since it is NCC-initiated) Index 1 is reserved for the MNGM active connection. Index 1 can be used for both MNGM and CTRL if only one active connection for MNGM and CTRL signalling is required
ActiveCnxChannelId (1)		8	Channel_ID of the MNGM active connection
ActiveCnxIPv4SrcAddrMask1 (1) or ActiveCnxIPv6SrcAddrMask1 (1)		32+8/128+8 (see note 4)	Management IP address of the RCST (air interface IP address)
ActiveCnxIPv4DestAddrMask1 (1) or ActiveCnxIPv6DestAddrMask1 (1)		32+8/128+8 (see note 4)	IP address of the NMC for SNMP traps sent to the NMC
ActiveCnxMACDestAddr (1)		48	(Destination) NMC MAC address used by the RCST to transmit management messages (see note 5)
ActiveCnxForwardStreamId (1)		24	PID value (Forward_MNGM_PID) used by the RCST to receive IP/MPE-based SNMP messages from NMC (Signalling Connection Forward Stream for management messages)
ActiveCnxReturnStreamId (1)		24	PID (Return_MNGM_PID) (or VCC) value used by the RCST to transmit IP/MPE-based (or ATM-based) SNMP messages towards NMC (Signalling Connection Return Stream for management messages) (see note 2)
ActiveCnxType (1)		8	Connection type of the MNGM active connection (unicast, bi-directional, NCC-initiated)
ActiveCnxInactivityTimeout (1)		8	0 = never times out
ActiveCnxC2PCoS (1)		8	C2P Class of Service set to 4 for Network Management (NM) (see note 6)
ActiveCnxAdmReturnSDR (1)		8	Admitted transmit Sustainable Data Rate for MNGM traffic
ActiveCnxAdmReturnPDR (1)		8	Admitted transmit Peak Data Rate for MNGM traffic
requestClassCRA (1)		32	CRA level for the Request Class
requestClassRBDCMax (1)		32	Max RBDC that can be request for the Request Class
requestClassRBDCTimeout (1)		32	Persistence of the RBDC Request for the Request Class
requestClassVBDCMax (1)		32	Max VBDC that can be allocated for the Request Class
requestClassVBDCTimeout (1)		32	Time after which the RCST considers that a VBDC pending request is lost
ActiveCnxRefId (2) (note 1)		16	Reference ID of the CTRL active connection (referenced from NCC, since it is NCC-initiated) Index 2 is reserved for CTRL active connection. If only one connection is required for MNGM and CTRL, Index 1 is used for this connection, for both MNGM and CTRL signalling
ActiveCnxChannelId (2)		8	Channel_ID of the CTRL active connection
ActiveCnxIPv4SrcAddrmask1 (2) or ActiveCnxIPv6SrcAddrMask1 (2)		32+8/128+8 (see note 4)	Control IP address of the RCST (air interface IP address)
ActiveCnxIPv4DestAddrMask1 (2) or ActiveCnxIPv6DestAddrMask1 (2)		32+8/128+8 (see note 4)	IP address of the NCC for C2P messages sent to the NCC
ActiveCnxMACDestAddr (2)		48	(Destination) NCC MAC address used by RCST to transmit control messages (see note 5)
ActiveCnxReturnStreamId (2)		24	PID (Return_CTRL_PID) (or VCC) value used by the RCST to transmit control messages (including C2P messages) to NCC (Signalling Connection Return Stream for C2P/control messages) (see note 3)
ActiveCnxType (2)		8	Connection type of the CTRL active connection (unicast, bi-directional, NCC-initiated)
ActiveCnxInactivityTimeout (2)		8	0 = never times out
ActiveCnxC2PCoS (2)		8	C2P Class of Service set to 4 for Network Management (NM) (see note 6)
ActiveCnxAdmReturnSDR (2)		8	Admitted transmit Sustainable Data Rate for CTRL traffic
ActiveCnxAdmReturnPDR (2)		8	Admitted transmit Peak Data Rate for CTRL traffic

NLID Parameters	No. of bits		Value/Comment
	Reserved	Information	
requestClassCRA (2)		32	CRA level for the Request Class
requestClassRBDCMax (2)		32	Max RBDC that can be request for the Request Class
requestClassRBDCTimeout (2)		32	Persistence of the RBDC request for the Request Class
requestClassVBDCMax (2)		32	Max VBDC that can be allocated for the Request Class
requestClassVBDCTimeout (2)		32	Time after which the RCST considers that a pending VBDC request is lost
PIDmmt	3	13	PID value dedicated to the reception of the Multicast Mapping Table (MMT) (see note 7)

NOTE 1: There is no need for a dedicated PID for forward control signalling, since this relies on the PID for the TIMu (carrying the Connection Control Descriptor) assigned during forward link acquisition.

NOTE 2: The Return_MNGM_PID (ReturnStreamID (1)) has been introduced in the table to separate the return SNMP signalling from the general return management and control signalling. In systems that do not require such separation, the Return_CTRL_MNGM_PID provided by the Logon Initialize Descriptor could be used for both management and control/C2P signalling messages, in which case the Return_MNGM_PID would not be needed. When Return_MNGM_PID is used and set to 0, the RCST shall not transmit any SNMP management messages to NMC.

NOTE 3: The Return_CTRL_PID (ReturnStreamID (2)) has been introduced in the table to separate the return C2P/control signalling from the general return management and control signalling. In systems that do not require such separation, the Return_CTRL_MNGM_PID provided by the Logon Initialize Descriptor could be used for both management and C2P/control signalling messages, in which case the Return_CTRL_PID would not be needed. When Return_CTRL_PID is used and set to 0, the RCST shall not transmit any C2P/control messages to NCC.

NOTE 4: The present specification supports both IPv4 and IPv6 addressing. IPv4 allocates 32-bit addresses (+8 bits for mask) and IPv6 allocates 128-bit addresses (+8 bits for mask).

NOTE 5: Only required for MPEG format.

NOTE 6: According to the C2P CoS coding defined in the Active Connection Table (clause 4.6.3.3).

NOTE 7: The PIDmmt allows identifying one Multicast Map Table per independent satellite sub-network (clause 4.1). A corresponding MIB object will be identified in the RCST MIB rcstConfig.Network subgroup.

- **Mesh Logon Initialize Descriptor:** This descriptor format is similar to the format of the Logon Initialize Descriptor defined in clause 8.5.5.10.4 in [1], but provides additional parameters for the initial logon of a mesh RCST, namely:
    - the gross maximum CRA assignment that will be allowed for mesh connections;
    - the gross maximum number of timeslots per superframe that can be assigned upon VBDC requests for mesh traffic resources;
    - the gross maximum rate that can be assigned upon RBDC requests for mesh traffic resources.

When used, it is provided **instead** of the existing Logon Initialize Descriptor, as a part of the logon response. Since Mesh Logon Initialize Descriptor is a superset of the existing Logon Initialize Descriptor, it allows the RCST to operate in both star and mesh modes simultaneously.

Second logon phase (RCST Capability Request message)

After a successful admission into the network and successful establishment of signalling connection(s), an RCST supporting dynamic connectivity may initiate a second logon phase, which consists in sending additional information regarding its capability to fully support the operation of the connection control protocol. This information, which includes the C2P version, the IP protocol version and the multicast option, is transmitted in a special Information Element in a C2P message (RCST Capability request), by relying on the signalling connection established for C2P messages. The NCC responds with a C2P acknowledgment message (RCST Capability response). After the completion of this phase of the logon process, the RCST (or the NCC/GW or RSGW) can initiate the establishment of traffic connections based on C2P.

The Information Element for the additional RCST capabilities is described in clause 7.2.2.17.

4.6.2 C2P functional flags

According to [1], the CSC burst includes two flags which determine the C2P-related functionality supported by the RCST and allow the NCC to configure the required functionality at logon time. The two flags are:

- Dynamic Connectivity flag: 1 bit coded as follows:
    - 0: for RCST supporting dynamic connectivity, meaning:
        - support of C2P;
        - support of the extended format (8 bits) of the Channel_ID;
        - support of 2/3-bit Capacity_Request_Format/Capacity_Request_Type (as per version 1.5.1 of [1]);
        - support of DULM method for return signalling;
    - 1: otherwise:
        - As stated in clause 4.6.1, declaring support for dynamic connectivity may trigger a second logon phase, allowing the RCST to advertise the supported C2P version, IP protocol version and multicast option.
- Transparent mesh reception support: Indicates support for the reception of transparent mesh signals (bursts). The field (2 bits) is coded as:
    - 11: no burst mode reception supported;
    - 10: single-carrier burst mode receiver;
    - 01: multi-carrier burst mode receiver;
    - 00: reserved.

4.6.3 RCST data structures

The control plane functionality added by C2P has been identified in clause 4.1. In order to support this functionality the RCST relies on a number of internal data structures (tables), namely:

- Packet Classification Table.
- PHB Mapping Table.
- Request Class Table.
- Connection Profile Mapping Table.
- Active Connection Table.

The first three tables have already been defined in the SatLabs SSR - Part 3 [i.4] and also in an IETF DVB-RCS MIB draft [i.12]. They only need to be updated to include new C2P-relevant parameters. The other tables are C2P-specific and their exact content is defined in this clause.

The primary usage of the above tables is to setup connections with defined C2P connection profiles and to modify/release them. Setting-up/modifying connections involves dynamic bandwidth allocation and configuration of parameters in various network components. In this context the information stored in the RCST data structures is used not only for the construction of C2P messages but also for local (internal) processing in RCST.

NOTE 1: The tables configured in an RCST contain primarily information pertaining to that RCST. It may also contain information pertaining to the other RCST, in the case of some particular transparent mesh connections; the corresponding parameters will be prefixed by "Other" (e.g. Other Channel_ID, Other Route_ID, Other Group_ID and Logon_ID).

NOTE 2: For consistency with the format used in the existing tables, all new parameters will be defined in the same format, i.e. as lists of MIB objects. The parameters will thus be ready for inclusion in the DVB-RCS MIB, once the C2P TS has been approved. They can be used e.g. for configuration management or for NCC internal signalling based on SNMP messages carrying C2P related parameters as MIB objects.

NOTE 3: For the parameters already included in the DVB-RCS MIB, the size of the fields (number of bits) in various tables and the assumed units are as defined in the DVB-RCS MIB. For the new parameters the size is the estimated size (number of information bits and possibly reserved bits), and the units are as defined in the corresponding tables (consistent with those defined for the Information Elements in clause 7.2.2). After inclusion in the MIB, the size of the fields may be rounded-up to some standard formats/sizes (e.g. Integer, Integer32 etc). The MIB will also include the units for each new parameter.

4.6.3.1 Packet Classification Table

The Packet Classification Table defines the packet classification used in RCST. It will be configured in each RCST by management or a local interface (e.g. CLI or web interface).

Packet classification is based on filter criteria/masks, including primarily layer 3 (IP) parameters, but also some layer 2 Ethernet parameters (e.g. VLAN Priority). As a result of classification, each packet matching a defined filter mask is associated with a packet type or flow type, uniquely identified by a Packet Classification Index (PktClassIndex). If no filter mask is matched the packet shall be discarded.

The minimum set of parameters for each packet/flow type is given in table 4.4, which includes SatLabs legacy parameters and a set of new parameters required for C2P support (in fair font).

Table 4.4: Packet Classification Table

IP Packet Class Table	No. of bits	Description
PktClassEntry		An entry in the Packet Classification Table.
PktClassIndex	16	Index of the Packet Classification Table. Used to identify a packet type or flow type.
PktClassDscpLow	32	Specifies the low value of a range of DSCP values to which a packet is compared. A value of 0 is used to inactivate.
PktClassDscpHigh	32	Specifies the high value of a range of DSCP values to which a packet is compared. A value of 63 is used to inactivate.
PktClassDscpMarkValue	32	Specifies the DSCP value used to mark (remark) a packet. Possible DSCP mark values are (0..63). A value of 64 indicates no DSCP marking.
PktClassIPProtocol	32	Specifies the IP protocol to which a packet is compared (e.g. TCP, UDP etc.).
PktClassIPSrcAddr	32+8/128+8 (see note 1)	Specifies the IP source address to which a packet is compared.
PktClassIPSrcAddrMask	32+8/128+8 (see note 1)	Specifies which bits of the IP source address will be matched.
PktClassIPDstAddr	32+8/128+8 (see note 1)	Specifies the IP destination address to which a packet is compared (see note 2).
PktClassIPDstAddrMask	32/+8/128+8 (see note 1)	Specifies which bits of the IP destination address will be matched.
PktClassSrcPortLow	32	Specifies the low range of the source port number to which a packet is compared.
PktClassSrcPortHigh	32	Specifies the high range of the source port number to which a packet is compared.
PktClassDstPortLow	32	Specifies the low range of the destination port number to which a packet is compared.
PktClassDstPortHigh	32	Specifies the high range of the destination port number to which a packet is compared.
PktClassVlanPri	32	Specifies the VLAN User Priority to which a packet is compared.
PktClassPhbAssociation	8	Associates the filter entry to a specific PHB/IntServ CoS (by reference to a PhbIndex in the PHB/CoS Mapping Table).
PktClassIntServCoSId	1	Specifies the identifier of the IntServ Class of Service of the packet, as obtained from a control interface. Possible values: "0":GS; "1": CL (see note 4).

IP Packet Class Table	No. of bits	Description
PktClassIPPriority	8	Applicable to IP elementary flows (e.g. IntServ), for priority scheduling (0: No priority; 1: Low priority; 255: High priority) (see note 5).
PktClassTraficSpecPolicyData	256 (see note 3)	Relevant traffic information/policies (typically derived from a control plane interface.
PktClassCnxProfileAssociation	16	Associates this entry to a specific connection profile (by reference to a CnxProfileIndex in the Connection Profile Mapping Table).
PktClassAction	1	Specifies if the packets mapped to this entry (flow type) can be transmitted to the satellite interface or should be discarded ("0": Permit; "1": Deny). The parameter can be related to a firewall function, used to avoid undesired incoming traffic.
PktClassOutOctets	32	Number of octets sent out over the satellite.
PktClassOutPkts	32	Number of packets sent out over the satellite (can be unicast or multicast, depending on the packet classification).
PktClassRowStatus	8	Standard SNMP row status.
NOTE 1: The present specification supports both IPv4 and IPv6 addressing. IPv4 allocates 32-bit (+8-bit mask) addresses and IPv6 allocates 128-bit addresses (+8-bit mask). NOTE 2: The PktClassIPDstAddr can contain an IP subnet that corresponds to several peer parties (e.g. several peer RCSTs). NOTE 3: The size of this field is a rough estimate. It will depend on the specific form of traffic information/policies (implementation specific). NOTE 4: Two classes of service are defined in IntServ - Guaranteed Service (GS) and Control Load (CL). The class of service of an IP flow is obtained from an RSVP message. NOTE 5: Used for internal processing in RCST (e.g. IP layer priority scheduling).		

There is one entry in the Packet Classification Table for each packet/flow type. Each entry in the table is associated with a PHB Entry in the PHB/CoS Mapping Table (clause 4.6.3.4) and a Connection Profile Entry in the Connection Profile Mapping Table (clause 4.6.3.2). A packet/flow type with a given connection profile may be defined to apply to different destination parties (e.g. different RCTSs), all part of a big subnet.

Once a packet has been classified into a pre-configured flow type/PktClassIndex, the RCST checks if an Active Connection Entry exists in the Active Connection Table (clause 4.6.3.3) for this PktClassIndex and for the defined destination IP address. If no entry is found, an Active Connection Entry will be automatically created with the profile taken from the Connection Profile Mapping Table. If an entry is found and the packet is already part of an existing flow or flow aggregate carried by the connection, it will be forwarded according to the parameters stored in the Active Connection Table. If an entry is found but the packet is not part of an existing flow/flow aggregate, the requested bandwidth of the corresponding connection will be updated in the Active Connection Table and used in a subsequent connection modify request, issued in order to accommodate the new flow.

NOTE 1: A packet/flow type, identified by a PktClassIndex, is defined for a destination IP subnet and may correspond to several peer parties (e.g. several peer RCSTs).

NOTE 2: An active connection is created for a specific destination IP address. Several connections can be created for a given PktClass Index, for different destination IP addresses, part of the same destination IP subnet.

NOTE 3: An active connection carries one or multiple flow(s) or flow aggregate(s).

The association of a PktClassIndex with a PHB/CoS on one side, and with an active connection with given profile on the other side, allows to determine the PHB/list of PHBs (for DiffServ) or IP CoS/list of IP CoSs (for IntServ) mapped to the connection.

4.6.3.2 Connection Profile Mapping Table

Connection Profile Mapping Table includes the set of parameters defining the connection (or C2P) profile (as per clause 4.4.3). They are used to specify the profile-related parameters in the C2P connection establishment/modify messages.

Connection Profile Mapping Table will be configured in each RCST by management or a local interface (e.g. CLI or web interface).

The minimum information required per connection profile entry, uniquely identified by a Connection Profile Index (CnxProfileIndex), is included in table 4.5.

Table 4.5: Connection Profile Mapping Table

Connection Profile Table	No. of bits	Description
CnxProfileEntry		An entry in the Connection Profile Mapping Table.
CnxProfileIndex	32	Index (identifier) of the Connection Profile.
CnxProfileConnectionType	8	Defines the type of the connection for the IP flow/aggregate (unicast or multicast, unidirectional or bidirectional, RCST/RSGW-initiated or NCC-initiated) mapped to this connection profile.
CnxProfileC2PCoS	8	MAC class of service for the connection profile (assumed the same in both directions in the case of a bidirectional connection) (see note).
CnxProfileReturnSDR	8	Transmit sustainable data rate for the connection. It refers to the aggregate rates for all transmit flows carried by the connection.
CnxProfileReturnPDR	8	Transmit peak data rate. It refers to the aggregate rates for all transmit flows carried by the connection.
CnxProfileForwardSDR	8	Receive sustainable data rate for the bidirectional connection. It refers to the aggregate rates of all receive flows carried by the connection.
CnxProfileForwardPDR	8	Receive peak data rate for the bidirectional connection. It refers to the aggregate rates of all receive flows carried by the connection.
CnxProfileInactivityTimeout	16	This parameter is used to trigger the release of the connection if there is no forward or return traffic activity (in seconds; 0 = never times out).
CnxProfileRowStatus	8	Standard SNMP row status.
NOTE: In the case of bidirectional connection the same C2P CoS assumed for both directions implies that identical or similar QoS architectures are implemented at both ends of the connection.		

The SDR and PDR parameters in the Connection Profile Mapping Table (Return SDR/PDR and also Forward SDR/PDR in the case of bidirectional connections) offers a convenient mechanism to specify the bandwidth resources that need to be requested for a flow type mapped to a specific connection profile entry, when the connection messages for connection establishment/modify are triggered by events in the user plane, such as the arrival of an IP packet. In the case where the connection establishment/modify messages were triggered by events in the control plane, e.g. the interception of an application session signalling message, the bandwidth parameters could be provided by the corresponding session signalling protocols (e.g. RSVP, SIP/SDP). This case is excluded from the version of the C2P specified in the present document.

A number of connection profiles shall be predefined for the anticipated applications. The connection profiles defined in a system are system-specific; an example is provided in [i.5].

4.6.3.3 Active Connections Table

The Active Connections Table contains all data necessary to fully characterize all active (open) connections of a given RCST. The table includes an entry for each active connection; the entry is dynamically created and updated every time a C2P message is received. The parameters are derived from the Information Elements of the C2P messages (clause 7.2.2).

The minimum information required per active connection, uniquely identified by a Connection Reference ID (ActiveCnxRefId), is included in table 4.6. Each entry in the table is associated to a Request Class Entry in the Request Class Table and to a Packet Class Entry in the Packet Classification Table.

The majority of the parameters in the table (e.g. addresses, VPI/VCI, PID, Channel_ID, Route_ID) are configured when the active connection is created. The rate parameters of the connection (Return SDR/PDR and also Forward SDR/PDR in the case of bidirectional connections) can be dynamically updated by connection modify request/response messages. The Active Connection Table includes both requested rate parameters and admitted rate parameters.

The requested rate parameters (ReqReturnSDR/PDR, ReqForwardSDR/PDR) represent the requested aggregate rates for all IP flows/flow aggregates mapped to a specific active connection at a given time. They are included in all C2P request messages related to the connection. The first request is for connection establishment, triggered e.g. by the first flow that is mapped to this connection, and the requested rates are obtained from the Connection Profile Mapping Table. The rates is then updated with each flow/flow aggregate that is activated/deactivated within the connection, and a C2P request for a connection modification is sent. The request includes the total (cumulative) rates, not just the incremental changes.

The admitted rate parameters (AdmReturnSDR/PDR, AdmForwardSDR/PDR) are dynamically updated by the NCC via C2P messages, in response to each C2P connection establishment/modify request, after the execution of a connection admission control function (CAC).

In the case of unidirectional connections the rate parameters apply to one direction and the table includes only one set of requested/admitted rate parameters - either the ReturnSDR/PDR (if the connection was initiated by this RCST/RSGW) or the ForwardSDR/PDR (if the connection was initiated by the other RCST/RSGW or NCC).

In the case of bidirectional connections, the requested/admitted rate parameters will be updated for both directions, regardless of the initiator of the connection establishment/modify request.

In the case of bidirectional connections between two RCSTs, identical or similar parameters (with the same terminology) will be maintained in the Active Connection Tables at the two ends of the connection (for a given ActiveCnxEntry), but the meaning is changed: the Source, Return and Forward parameters at one RCST become Destination, Forward and Return parameters, respectively, at the other RCST. For example, the requested/admitted Return and Forward SDR/PDR at one RCST will be reversed at the other RCST, and vice-versa.

With regard to the C2P CoS, Connection Type and Connection Inactivity Timeout, they will be identical at the two ends of the connections, which are assumed configured to support the same QoS architecture/same PHBs or IP CoSs. The Connection Inactivity Timeout is set by the NCC and may be different from that in the Connection Profile Mapping Table, used in the connection establishment request message.

Table 4.6: Active Connection Table

Active Connection Table	No. of bits	Description (see note 4)
ActiveCnxEntry		An entry in the Active Connection Table
ActiveCnxIndex	32	Index of the active connection
ActiveCnxRefId	16	Identifier of the active connection
ActiveCnxMACSrcAddr	48	MAC address of the RCST (see note 2)
ActiveCnxCause	8	Last reported cause of the active connection
ActiveCnxChannelId	8	Channel_ID of the connection; same value as in RC table
ActiveCnxMACDestAddr	48	MAC address of the peer RCST
ActiveCnxReturnStreamId	24	Transmit VCC for ATM or transmit PID for MPEG
ActiveCnxForwardStreamId	24	Receive VCC for ATM or receive PID for MPEG
ActiveCnxType	8	Defines the type of the connection (unicast or multicast, unidirectional or bidirectional, RCST/RSGW-initiated or NCC-initiated)
ActiveCnxC2PCoS	8	MAC class of service for the connection (assumed the same in both directions in the case of a bidirectional connection). Example of values (SatLabs): "1": Real Time (RT), "2": Critical Data (CD), "3": Best Effort (BE) and "4": Network Management (NM)
ActiveCnxReqReturnSDR	8	Requested transmit sustainable data rate of all flows mapped to the connection
ActiveCnxReqReturnPDR	8	Requested transmit peak data rate of all flows mapped to the connection
ActiveCnxReqForwardSDR	8	Requested receive sustainable data rate of all flows mapped to the connection (informational)
ActiveCnxReqForwardPDR	8	Requested receive peak data rate of all flows mapped to the connection (informational)
ActiveCnxAdmReturnSDR	8	Admitted transmit sustainable data rate of all transmitted flows carried by the connection
ActiveCnxAdmReturnPDR	8	Admitted transmit peak data rate of all transmitted flows carried by the connection
ActiveCnxAdmForwardSDR	8	Admitted receive sustainable data rate of all received flows carried by the connection (informational)
ActiveCnxAdmForwardPDR	8	Admitted receive peak data rate of all received flows carried by the connection (informational)

Active Connection Table	No. of bits	Description (see note 4)
ActiveCnxRouteId	16	Route_ID associated to the connection; equivalent to a Channel_ID list
ActiveCnxGroupLogonId	24	Group_ID and Logon_ID of the RCST
ActiveCnxOtherChannelId	8	Channel_ID of the other RCST (see notes 1 and 2)
ActiveCnxOtherRouteId	16	Route_ID of the other RCST (see notes 1 and 2)
ActiveCnxOtherGroupLogonId	24	Group_ID and Logon_ID of the other RCST (see notes 1 and 2)
ActiveCnxInactivityTimeout	16	Used to trigger the release of the connection if there is no forward or return traffic activity (in seconds; 0 = never times out)
ActiveCnxMaxPacketSize	16	Maximum packet size
ActiveCnxStatus	8	Status of the connection: 1: SetupInProgress, 2: ConnectionOpen, 3: ConnectionReleaseInProgress, 4: ConnectionModifyInProgress
ActiveCnxPktClassAssociation	32	Associates this entry to a specific Packet Class (by reference to a PktClassIndex in the Packet Classification Table)
ActiveCnxRequestClassAssociation	32	Associates this entry to a specific request class (by reference to a requestClassIndex in the Request Class Table)
ActiveCnxOutOctets	32	Number of octets sent out over the satellite
ActiveCnxOutPkts	32	Number of packets sent out over the satellite (can be unicast or multicast, depending on the packet classification)
ActiveCnxInOctets	32	Number of octets received from the satellite
ActiveCnxInPkts	32	Number of packets received from the satellite (can be unicast or multicast)
ActiveCnxIPv4SrcAddrMask1	32+8	Defines the IPv4 source address and mask
ActiveCnxIPv4SrcAddrMask2	32+8	Defines the IPv4 source address and mask
ActiveCnxIPv4SrcAddrMask3	32+8	Defines the IPv4 source address and mask
ActiveCnxIPv4SrcAddrMask4	32+8	Defines the IPv4 source address and mask
ActiveCnxIPv4SrcAddrMask5	32+8	Defines the IPv4 source address and mask
ActiveCnxIPv4SrcAddrMask6	32+8	Defines the IPv4 source address and mask
ActiveCnxIPv4SrcAddrMask7	32+8	Defines the IPv4 source address and mask
ActiveCnxIPv4SrcAddrMask8	32+8	Defines the IPv4 source address and mask (see note 3)
ActiveCnxIPv4DestAddrMask1	32+8	Defines the IPv4 destination address and mask
ActiveCnxIPv4DestAddrMask2	32+8	Defines the IPv4 destination address and mask
ActiveCnxIPv4DestAddrMask3	32+8	Defines the IPv4 destination address and mask
ActiveCnxIPv4DestAddrMask4	32+8	Defines the IPv4 destination address and mask
ActiveCnxIPv4DestAddrMask5	32+8	Defines the IPv4 destination address and mask
ActiveCnxIPv4DestAddrMask6	32+8	Defines the IPv4 destination address and mask
ActiveCnxIPv4DestAddrMask7	32+8	Defines the IPv4 destination address and mask
ActiveCnxIPv4DestAddrMask8	32+8	Defines the IPv4 destination address and mask (see note 3)
ActiveCnxIPv6SrcAddrMask1	128+8	Defines the IPv6 source address and mask
ActiveCnxIPv6SrcAddrMask2	128+8	Defines the IPv6 source address and mask
ActiveCnxIPv6SrcAddrMask3	128+8	Defines the IPv6 source address and mask
ActiveCnxIPv6SrcAddrMask4	128+8	Defines the IPv6 source address and mask
ActiveCnxIPv6SrcAddrMask5	128+8	Defines the IPv6 source address and mask
ActiveCnxIPv6SrcAddrMask6	128+8	Defines the IPv6 source address and mask
ActiveCnxIPv6SrcAddrMask7	128+8	Defines the IPv6 source address and mask
ActiveCnxIPv6SrcAddrMask8	128+8	Defines the IPv6 source address and mask (see note 3)
ActiveCnxIPv6DestAddrMask1	128+8	Defines the IPv6 destination address and mask
ActiveCnxIPv6DestAddrMask2	128+8	Defines the IPv6 destination address and mask
ActiveCnxIPv6DestAddrMask3	128+8	Defines the IPv6 destination address and mask
ActiveCnxIPv6DestAddrMask4	128+8	Defines the IPv6 destination address and mask
ActiveCnxIPv6DestAddrMask5	128+8	Defines the IPv6 destination address and mask
ActiveCnxIPv6DestAddrMask6	128+8	Defines the IPv6 destination address and mask
ActiveCnxIPv6DestAddrMask7	128+8	Defines the IPv6 destination address and mask
ActiveCnxIPv6DestAddrMask8	128+8	Defines the IPv6 destination address and mask (see note 3)
ActiveCnxRowStatus	8	Standard SNMP row status

NOTE 1: This parameter may be needed for bidirectional connections defined for some particular transparent mesh scenarios. The prefix "Other" is used for the RCST at the other end of the connection, referenced with regard to the RCST to which this table applies.

NOTE 2: Not used in the C2P messages defined in the present document.

NOTE 3: The maximum number of IP addresses and masks per active connection is 8. A connection can have from 0 to 8 IP addresses and masks. If the number of IP addresses and masks is less than 8, the unused fields are set to zero.

NOTE 4: The detailed description of the parameters in this table is provided in clause 7.

4.6.3.4 PHB/CoS Mapping Table

The PHB/CoS Mapping Table is a SatLabs legacy table, extended to satisfy additional C2P needs.

The original SatLabs table, labelled PHB Mapping Table, was changed to PHB/CoS Mapping Table in order to reflect the inclusion of the following parameters:

- IP classes of service (IP CoS) specific to IntServ;
- parameters specific to each PHB/CoS (e.g. rates, delay, jitter, buffer size);
- parameters associated with a defined mechanism, e.g. a token bucket mechanism (optional);
- statistics.

The parameters to be included in the PHB/CoS Table are shown in table 4.7. The new parameters and the changes to the original parameters are shown in fair font. There is one entry in the table for each PHB/CoS. Each entry is associated with a Request Class Entry (requestClassEntry) in the Request Class Table. For simplicity all parameters include the PHB prefix, even if they may refer to an IntServ CoS (i.e. GS or CL).

Table 4.7: PHB/CoS Mapping Table

PHB/CoS Table	No. of bits	Description
PhbMappingEntry		An entry in the PHB mapping table
PhbIdentifier	32	Identification of the PHB in the range (0..63)
PhbName	String	Name of the PHB
PhbIPCoSIdentifier	8	Identification of the IntServ CoS in the range (0, 2) (0: not valid; 1: GS; 2: CL) (see note 3)
PhbIPCoSName	String	Name of the IntServ CoS
PhbAdmReturnSDR	8	Admitted (true) transmit sustainable data rate (in bits/s) (see note 1)
PhbAdmReturnPDR	8	Admitted (true) transmit peak data rate (in bits/s) (see note 1)
PhbBufferSize	16	Traffic buffer size (in bytes)
PhbDelay	16	Delay specification (in milliseconds)
PhbJitter	16	Jitter specification (in milliseconds)
PhbPolicy	128 (see note 2)	As per DiffServ standard or system specific
PhbTokenBucketR1	16	First rate of a two-rate token bucket mechanism (in bits/s) (optional)
PhbTokenBbucketR2	16	Second rate of a two-rate token bucket mechanism (in bits/s) (optional)
PhbTokenBucketSize	16	Token buffer size (in bytes) (optional)
PhbRequestOutOctets	32	Number of octets sent out over the satellite
PhbRequestOutPkts	32	Number of packets sent out over the satellite (can be unicast or multicast, it depends of the packet classification)
PhbRequestOutDiscards	32	Number of outbound packets that were free of errors but discarded. (i.e. packets that were filtered out, e.g. to free up memory)
PhbRequestOutErrors	32	Number of outbound packets discarded because of errors
PhbRequestOutQLen	32	Number of packets in the outbound queue
PhbRequestClassAssociation		This object provides an association of this PHB/IP CoS to a Request Class (by referencing to a RequestClassIndex in the Request Class Table)
PhbMappingRowStatus	8	Standard SNMP row status
NOTE 1: This parameter may be used for internal processing in RCST (traffic conditioning), e.g. to derive the optional token bucket rate parameters. NOTE 2: The size of this field is a rough estimate; it will depend on the specific form in which the PHB policies are specified (implementation specific). The PHB policy in this table should be consistent with the IP traffic specification and policy data in the Packet Classification Table. NOTE 3: Two classes of service are defined in IntServ - Guaranteed Service (GS) and Control Load (CL). The class of service of an IP flow is obtained from an RSVP message.		

The PHB/CoS table is configured by management. The rate parameters (including the token bucket rates) are dynamically updated as connections are established/modified, by relying on the Active Connection association with a Packet Class and the Packet Class association with a PHB/CoS. For a given PHB/CoS the admitted rate parameters AdmReturnSDR/AdmReturnPDR shall reflect the components (that can be associated with this PHB/CoS) of the AdmReturnSDR/AdmReturnPDR parameters of all connections to which this PHB/CoS is mapped. The calculation of the PHB/CoS rate parameters from the connections' rate parameters is performed by each RCST based on rules that are system-specific. The derivation of the token bucket rate parameters from the AdmReturnSDR and AdmReturnPDR is based on rules that are both PHB/CoS-specific and implementation specific.

NOTE 1: A PHB/CoS can be mapped to multiple connections.

NOTE 2: Several PHBs/CoSs can be mapped to one connection (i.e. flows with different PHB/CoS can share one connection).

4.6.3.5 Request Class (RC) Table

The Request Class Table is a SatLabs legacy table, slightly changed to include some C2P-specific parameters (e.g., Route_ID and statistics).

The Request Class Table defines all layer 2 parameters for each supported RC, i.e. the MAC layer logical parameters (Channel_ID, Route_ID, PIDs or VPI/VCI), and the bandwidth parameters expressed as capacity categories limit values (CRA, RBDCmax, VBDCmax). These parameters are controlled by the NCC and shall be configured in both NCC and RCST. The minimum list of parameters in the Request Class Table is defined in table 4.8.

Table 4.8: Request Class Table

Request Class Table	No. of bits	Description
requestClassEntry		An entry in the Request Class (RC) table
requestClassIndex	32	Index of the Request Class table. A maximum of 256 entries can be created
requestClassName	String	Name of the RC
requestClassChanID	32	Channel_ID of this RC
requestClassVccVpi	32	VPI used for the RC (ATM profile)
requestClassVccVci	32	VCI used for the RC (ATM profile)
requestClassPidPoolReference	32	Reference to the preferred PID in the PID pool applicable to the RC (typically 1 PID per RC is used). PID pool table is as defined in [i.12]
requestClassCRA	32	Defines the CRA level for the Request Class
requestClassRBDCMax	32	Max RBDC that can be request for the Request Class
requestClassRBDCTimeout	32	Persistence of the RBDC request
requestClassVBDCMax	32	Max VBDC that can be allocated for the Request Class
requestClassVBDCTimeout	32	Time after which the RCST considers that the pending request are lost
requestClassVBDCMaxBackLog	32	VBDC Backlog per Request Class
requestClassRouteId	16	Route_ID associated to the request class; equivalent to a Channel_ID list
requestClassOutOctets	32	Number of octets sent out over the satellite (including the ATM or MPEG header)
requestClassOutCells	32	Number of cells (ATM or MPEG) sent over the satellite
requestClassRowStatus	8	Standard SNMP row status

The Request Class Table is configured by management with initial values and then the capacity limit values (CRA, RBDCMax, VBDCMax) can be dynamically updated (based on C2P messages) in response to a specific C2P RC modify request or when a connection is created, modified or released. In the latter case (which corresponds to an implicit RC modify request), the per-RC capacity limit values shall reflect the connection admitted transmit parameters (i.e. AdmReturnSDR/PDR, as captured in the Active Connection Table) of all connections mapped to this RC. The conversion of the aggregate (per-RC) rate parameters to capacity limit values shall be performed by the NCC, based on rules that are both RC-specific and system-specific.

The VCC/VPI and VCC/VCI in the Request Class Table correspond to the Transmit VCC (as Return Stream ID) in the Active Connection Table. In case of conflict, the VCC in the Active Connection Table takes precedence over the VCC in the Request Class Table.

The PID (as Return Stream ID) in the Active Connection Table corresponds to the preferred PID for the Request Class in the PID Pool referenced in the Request Class Table. If only one PID per Request Class is defined, it is the same as the PID specified in the Active Connection Table. In case of conflict the PID in the Active Connection Table takes precedence over the PID in the Request Class Table.

5 State Machines, counters and timers

5.1 Introduction

The state machine diagrams are a convenient way for capturing in graphical form the life of a connection. They are described in terms of states of a state machine, traversed by the NCC and RCSTs during the life of the connection, beginning and ending in the IDLE state.

This clause provides an overview of the C2P state machines, based on state diagram representations, various timers and counters, and lists of the possible transitions between states. The formal and complete definition of the state machines, including all the state transitions, is defined in UML [i.14] in annex A.

Specific state machines are defined for the RCST/RSGW and the NCC. Furthermore, different RCST/RSGW state machines may be defined, depending on the network scenario and on whether the RCST/RSGW is the connection's initiator, the "peer RCST/RSGW" or "the other RCST/RSGW" (see clause 4.4.2.2). Similarly, the NCC state machines may also be different, depending on whether the NCC is the connection initiator or not.

As part of this overview of the C2P state machines, figures 5.1 and 5.2 provide two "conceptual" C2P state machines/behavioural diagrams, PER-ST and PER-CONNECTION, respectively, as well as the relationship between them.
PER-ST state machine describes the behaviour of DVB-RCS Satellite Terminal (ST) seen from the point of view of the air interface; therefore PER-ST state machine applies to RCSTs and RSGWs (see clause 4.2). The NCC controls all STs in the system, therefore the NCC will keep an instance of the per-ST state machine for each provisioned ST in the network, in order to control if the ST is ready or not ready to establish connections, depending on whether it is synchronized or not.

- The **PER-ST state machine** comprises the two states in which a Satellite Terminal can be (see figure 5.1):
  - NOT READY: This state represents a lack of synchronization of the ST. It applies to RCSTs/RSGWs that may be switched off, or that may have been neither provisioned nor synchronized by/to the NCC yet.
  - READY: In this state the satellite terminal has been provisioned by the NCC and it is finely synchronized to it.

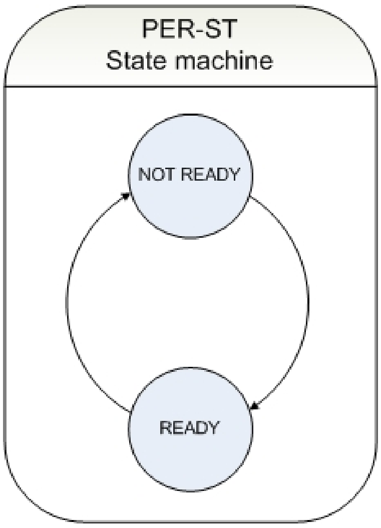


Figure 5.1: PER-ST state machine

- **PER-CONNECTION state machine** comprises the five main states a given connection may go through the life of the connections, as a result of executing setup, release or modification procedures. In all five states it is assumed that the terminal has been provisioned by the NCC and that it has achieved fine synchronization, i.e. it is in the READY state of the PER-ST state machine. It is therefore in the READY state of the PER-ST state machine that the PER-CONNECTION state machine's states are defined, as shown in figure 5.2 (and also in the example included in annex A). The transition from one state to another corresponds to the execution of various C2P commands.

  1) IDLE: No connection procedure is ongoing, but the terminal is synchronized, therefore the signalling connections are active and the terminal is ready to handle C2P messages. This is a "stable/static" state associated with the final outcome of an exchange of C2P messages. An ST can also transit to the IDLE state from the "NOT READY" state of the "PER-ST" state machine, after the ST has been provisioned and finely synchronized.

  2) SETUP IN PROGRESS: The connection is being established at either one or both terminals (as applicable), as well as at the NCC. It is a "transient" state from one stable state (IDLE) to another (CONNECTION OPEN).

  3) CONNECTION OPEN: The connection has been successfully established. This is a "stable/static" state associated with the final outcome of an exchange of C2P messages.

  4) CONNECTION MODIFY IN PROGRESS: The connection is being modified. This state includes the modification of the connection profile, in response to specific C2P commands, as well as the addition or dropping of IP flows to/from an established connection, without modifying its profile. It is a "transient" state from/to the same stable state (CONNECTION OPEN).

  5) RELEASE IN PROGRESS: The connection is being released at either one or both terminals (as applicable), as well as at the NCC, It is a "transient" state from one stable state (CONNECTION OPEN) to another (IDLE).

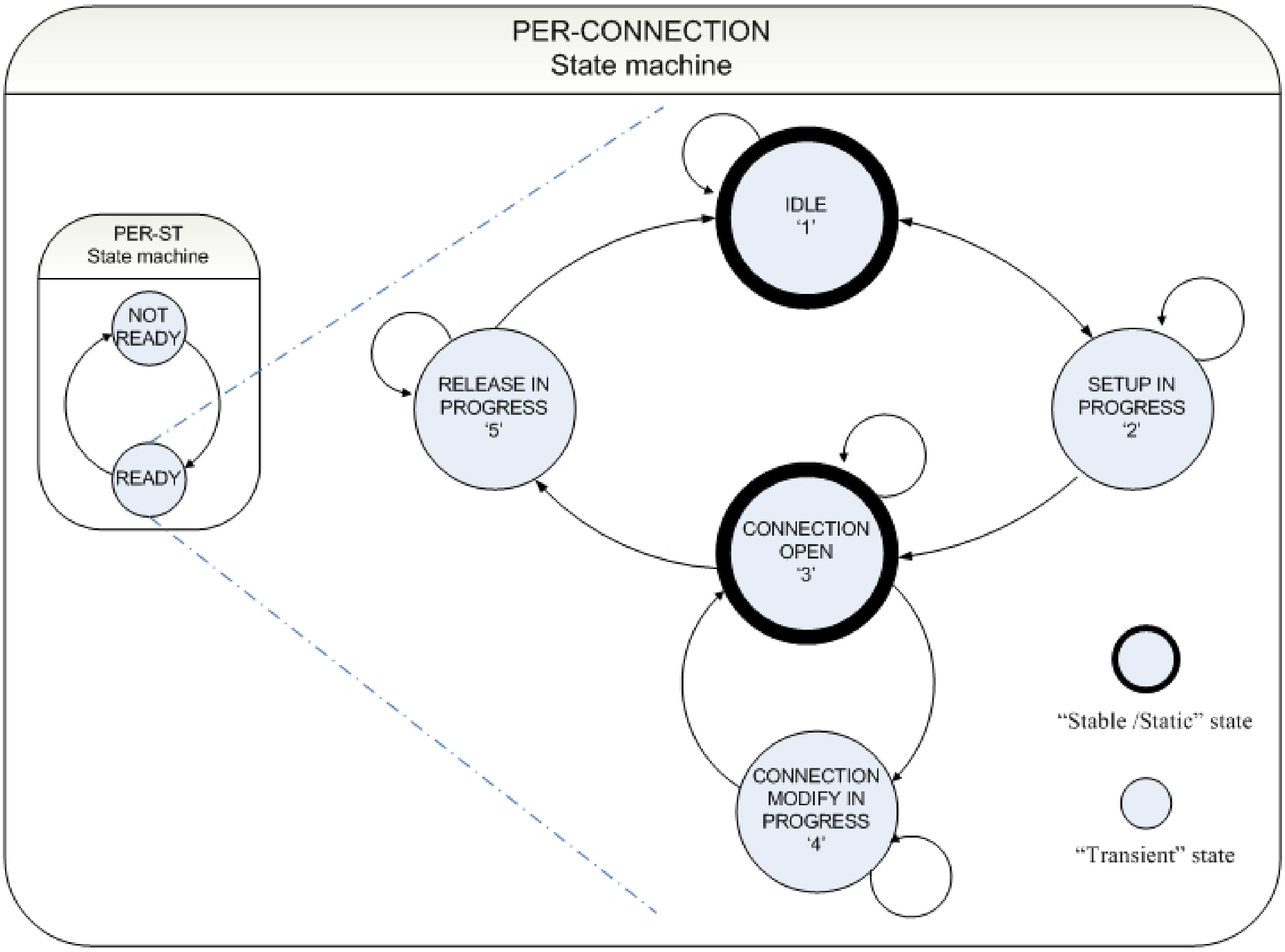


Figure 5.2: PER-CONNECTION state machine and its relationship with PER-ST state machine

- **GENERAL NOTES FOR THE ABOVE STATE MACHINE**

NOTE 1: For the three "transient" or "in progress" states ('2', '4', '5') the transient time is controlled by timers (associated with timeouts) and counters, to prevent indefinitely long waiting times and loops (repeats of C2P commands).

NOTE 2: From the point of view of C2P, the relevant outputs of each state in the state machines are the actual C2P messages, providing the transitions to other states.

NOTE 3: A number of state machines (determined by system and software parameters) will be running in parallel, especially at the NCC, for all activated (ongoing) connections.

NOTE 4: The above state machine diagrams and those included in annex A do not show the actions associated with connection status/stats requests and the responses to those requests, with the exception of the special case corresponding to "NCC busy" (see annex C).

NOTE 5: For a better understanding of how a connection control procedure is derived from the C2P UML state machines included in annex A, the "conceptual" state machines provided in figures 5.1 and 5.2 and their relationships with the C2P procedure described in clause 6.1.1 are included in annex A, immediately after the formal and complete definition of the UML state machines.

The transitions between states (including loops that return to the same state) are activated by:

a) Incoming C2P messages.

b) Trigger events, e.g. arrival of a user packets, interception of a signalling message.

c) Timer events, i.e. the expiration of a timeout.

d) Counter events, i.e. the exhaustion of the maximum number of repeats set for a specific C2P command.

e) Commands sent from a console (via Interfaces or Function Calls); these are requests or commands to take actions, coming from the interface either on the user data side (IP layer) or on the satellite network side.

5.2 Timers

The timers (and associated timeouts) defined in the following clauses shall be used to ensure that the C2P state machines never remain stuck in one of the unstable states.

The description of the timers and their values are provided in table 5.1 for RCST/RSGW and in table 5.2 for the NCC, where "M" refers to a Mandatory timer and "O" refers to an Optional Timer. Mandatory timers shall be implemented in all systems supporting C2P, while the implementation of the optional timers is system specific. The values of various timers are suggested, not required, and are system dependent. To ensure cross-vendor compatibility, any implementation of the RCST/RSGW's or NCC's C2P software should be confirmed for interoperability across the range of permissible timer values of the corresponding network element (RCST/RSGW against NCC, and vice versa).

Timers (and associated timeouts) are in general used in relationship with various C2P requests (e.g. connection establishment request, connection modify request, connection release request). They limit the time the requesting party is waiting for a response.

Some specific timers can be defined for releasing of connections due to traffic inactivity. Transmission Traffic Inactivity and Reception Traffic Inactivity timers shall be used to monitor traffic transmission and reception. These timers are reset with any transmitted/received packet. If there is no packet transmitted/received for a duration equal or exceeding a predefined timeout value (e.g. the Connection Inactivity Timeout in the Active Connection Table - see clause 4.6.3.3), a connection release request will be triggered.

For a better understanding of how timers work at the NCC and at the RCST/RSGWs please refer to clause 6 and annex C.

5.2.1 RCST/RSGW timers

Table 5.1: RCST/RSGW timers

Timer	Description	Default Value (notes 5 and 6)	Min Value	Max Value	M/O
T-RCST_CnxEstReq (note 1)	RCST/RSGW Connection Establishment Request Timer defines the time the RCST/RSGW waits for the reception of the NCC response to a Connection Establishment Request sent by the RCST/RSGW.	5 s	1 s	15 s	M
T-RCST_ CnxModReq (note 1)	RCST/RSGW Connection Modify Request Timer defines the time the RCST/RSGW waits for the reception of the NCC response to a Connection Modify Request sent by the RCST/RSGW.	5 s	1 s	15 s	M
T-RCST_CnxRelReq (note 1)	RCST/RSGW Connection Release RequestTimer defines the time the RCST/RSGW waits for the reception of the NCC response to a Connection Release Request sent by the RCST/RSGW.	5 s	1 s	15 s	M
T-RCST_RCModReq (note 1)	RCST/RSGW RC Modify Request Timer defines the time the RCST/RSGW waits for the reception of the NCC response to an Request Class Modify Request sent by the RCST/RSGW.	5 s	1 s	15 s	M
T-RCST_RCSTCapReq (note 1)	RCST/RSGW Capabilities Request Timer defines the time the RCST/RSGW waits for the NCC's acknowledgement to the information provided by the RCST Capability Request (Notification).	1 s	1 s	10 s	M
T-RCST_Wait_CnxEstReq (note 2)	RCST/RSGW Wait for Connection Establishment Request Timer defines the time the RCST/RSGW shall wait before sending another CnxEstReq to the NCC, after receiving a connection establishment reject from the NCC.	5 s	5 s	15 s	O
T-RCST_Wait_CnxModReq (note 2)	RCST/RSGW Wait for Connection Modify Request Timer defines the time the RCST/RSGW shall wait before sending another CnxModReq to the NCC, after receiving a connection modify reject from the NCC.	5 s	5 s	15 s	O
T-RCST_Wait_CnxRelReq (note 2)	RCST/RSGW Wait for Connection Release Request Timer defines the time the RCST/RSGW shall wait before sending another CnxRelReq to the NCC, after receiving a connection release reject from the NCC.	5 s	5 s	15 s	O
T-RCST_Wait_RCModReq (note 2)	RCST/RSGW Wait for RC Modify Request Timer defines the time the RCST/RSGW shall wait before sending another RCModReq to the NCC, after receiving an RC modify request reject from the NCC.	15 s	5 s	300 s	O

Timer	Description	Default Value (notes 5 and 6)	Min Value	Max Value	M/O
T-TrfInactivity	User transmission/reception traffic inactivity timeout controls the transmission/reception traffic activity for a certain connection. If during this time no traffic is transmitted or received by the RCST/RSGW for this connection, a connection release request will be issued (notes 3 and 4).	60 s	2 s	1 800 s	M
NOTE 1: An implementation may optionally combine all request timers (to which note 1 applies) into a single timer, if the number of pending requests (of any type) at a given type is limited to one. NOTE 2: An implementation may optionally combine all timers to which note 2 applies into a single timer, if the number of rejects (of any type) received at a given type is limited to one. NOTE 3: This is the default timeout value for the inactivity timer, to be used when no inactivity timeout value is provided within the C2P message. Whenever the inactivity timeout is present in the C2P message, it shall take precedence over the default timeout value. For bidirectional connections both timers shall expire before triggering a connection release. For unidirectional connections only the transmission or the reception timer, as applicable, should be considered. NOTE 4: In the case of bidirectional connections this timer is implemented at both ends of the connection, while in the case of unidirectional connections it is only implemented at one end. NOTE 5: Default values are provided as recommendation. The implemented value will be system dependant. NOTE 6: The granularity for all timers is 1 second.					

5.2.2 NCC timers

Table 5.2: NCC timers

Timer	Description	Default Value (note 1)	Min Value	Max Value	M/O
T-NCC_CnxEstReq	NCC's Connection Establishment Request timer defines the time the NCC waits for the reception of the response to a Connection Establishment Request sent by the NCC to the RCST/RSGW.	5 s	1 s	15 s	M
T-NCC_CnxModReq	NCC's Connection Modify Request timer defines the time the NCC waits for the reception of the response to a Connection Modify Request sent by the NCC to the RCST/RSGW.	5 s	1 s	15 s	M
T-NCC_CnxRelReq	NCC's Connection Release Request timer defines the time the NCC waits for the reception of the response to a Connection Release Request sent by the NCC to the RCST/RSGW.	5 s	1 s	15 s	M
T-NCC_RCModReq	NCC's RC Modify Request timer defines the time the NCC waits for the reception of the response to a Request Class Modify Request sent by the NCC to the RCST/RSGW.	5 s	1 s	15 s	M
T-NCC_RCSTCapReq	NCC's RCST Capabilities timer defines the time the NCC waits for the reception of the response to an RCST Capability Request sent by NCC to RCST/RSGW.	1 s	1 s	10 s	M

Timer	Description	Default Value (note 1)	Min Value	Max Value	M/O
T-NCC_CnxStatusStatsReq	NCC's Connection Status Statistics timer defines the time the NCC waits for the reception of the response to a Connection Status Statistics Request (Inquiry) sent by the NCC to the RCST/RSGW.	1 s	1 s	10 s	M
NOTE 1: Actual values used in the implementation of these timers are system-dependant. NOTE 2: The granularity for all timers is 1 second.					

5.3 Counters

The counters defined in the following clauses shall be used to determine how many times various messages may be repeated before a failure is declared and the state machine returns to a stable state (e.g. IDLE).

The description of the counters and their values are provided in table 5.3 for the RCST/RSGW and in table 5.4 for the NCC, where "M" refers to a Mandatory Counter and "O" refers to an Optional Counter. Mandatory counters shall be implemented in all systems supporting C2P, while the implementation of the optional counters is system specific. The values of various counters are suggested, not required, and are system dependent. To ensure cross-vendor compatibility, any implementation of the RCST/RSGW's or NCC's C2P software should be confirmed for interoperability across the range of permissible counter values of the corresponding network element (RCST/RSGW against NCC, and vice versa).

For a better understanding of how counters work at the NCC and at the RCST/RSGWs please refer to clause 6 and annex C.

5.3.1 RCST/RSGW counters

Table 5.3: RCST/RSGW counters

Counter	Description	Min Value (default) (note)	Max Value	M/O
N-RCST_CnxEstReq	Maximum number of Connection Establishment Requests that can be sent by the RCST/RSGW without receiving a response from the NCC.	2 s	7 s	M
N-RCST_CnxModReq	Maximum number of Connection Modify Requests that can be sent by the RCST/RSGW without receiving a response from the NCC.	2 s	7 s	M
N-RCST_CnxRelReq	Maximum number of Connection Release requests that can be sent by the RCST/RSGW without receiving a response from the NCC.	2 s	7 s	M
N-RCST_RCModReq	Maximum number of RC Modify Requests that can be sent by the RCST/RSGW without receiving a response from the NCC.	2 s	7 s	M
N-RCST_RCSTCapReq	Maximum number of RCST Capability Requests that can be sent by the RCST/RSGW without receiving a response (acknowledgement) from the NCC.	2 s	7 s	O
NOTE: Actual values used in the implementation of these counters are system-dependant.				

5.3.2 NCC counters

Table 5.4: NCC counters

Counter	Description	Min Value (default) (note)	Max Value	M/O
N-NCC_CnxEstReq	Maximum number of Connection Establishment Requests that can be sent by the NCC without receiving a response from the RCST/RSGW.	2 s	7 s	M
N-NCC_CnxModReq	Maximum number of Connection Modify Requests that can be sent by the NCC without receiving a response from the RCST/RSGW.	2 s	7 s	M
N-NCC_CnxRelReq	Maximum number Connection Release Requests that can be sent by the NCC without receiving a response from the RCST/RSGW.	2 s	7 s	M
N-NCC_RCModReq	Maximum number RC Modify Requests that can be sent by the NCC without receiving a response from the RCST/RSGW.	2 s	7 s	M
N-NCC_RCSTCapReq	Maximum number RCST Capability Requests (Inquiries) that can be sent by the NCC without receiving a response from the RCST/RSGW.	2 s	7 s	O
N-NCC_CnxStatusStatsReq	Maximum number of Connection Status Statistics Requests that can be sent by the NCC without receiving a response from the RCST/RSGW.	2 s	7 s	O
NOTE: Actual values used in the implementation of the counters are system-dependant.				

5.4 PER-Connection State machines

The specific state machines in this clause correspond to the "PER-CONNECTION" generic state machine in clause 5.1.

The state machine diagram and transition table describe the possible transitions between the states corresponding to the connection establishment/modify/release procedures, consistent with the description in clause 6.

Different state machines can be envisaged for the RCST/RSGW and the NCC, depending on the DVB-RCS network reference scenario and on the role of the RCST/RSGW and NCC in the connection.

The following clauses include the definition of the state machines applicable to mesh unicast connections, whether the connection is initiated by the RCST/RSGW or by the NCC.

For the complete state machine description refer to annex A.

5.4.1 RCST state machine

For the RCST state machine (figure 5.3) the following states have been defined.

Table 5.5: RCST states definition

State	Label	Definition
Idle	I#	Ready to start a connection
Setup in progress	S#	Waiting for connection establishment response
Connection open	C#	Connection open
Connection modify in progress	Cx#	Waiting for connection modify response
Release in progress	R#	Waiting for connection release response
Not Ready	N#	Waiting to be synchronized

The transitions between states are labelled. Table 5.6 provides a description of each transition, which is typically a sequence of a trigger or input event, sometimes with parameters, followed by an output, usually a C2P message.

Table 5.6 applies to unicast star/mesh regenerative connections and unicast mesh transparent connections. The same table covers the three possible roles of the RCST/RSGW:

- Initiating RCST/RSGW (the RCST/RSGW is the initiator of the connection).
- Peer RCST/RSGW (when the connections was initiated by another RCST/RSGW).
- The Other RCST/RSGW (for NCC initiated connections).

Some transitions between states are specific to only one role of the RCST/RSGW, therefore different colour shadowing of rows in table 5.6 have been used to identify the transition specific to each role:

- Pale green shadowing for transitions associated only to connections for the initiating RCST/RSGW.
- Pale yellow shadowing for transitions associated only to connections initiated by the NCC or another RCST/RSGW, when the RCST/RSGW is either the "peer RCST/RSGW" or "the other RCST/RSGW".
- No shadowing for transitions common to all three possible roles of the RCST/RSGW.

NOTE 1: Any possible transitions not covered within the state diagram tables, imply that no actions and no changes are to be taken.

NOTE 2: Some transitions in the state diagram tables are associated with terminal's logoff, which is system specific. The logoff decisions can be made by an RCST or by the NCC, in order to resolve a critical C2P message communication failure.. From the RCST/RSGW's point of view, the logoff decision will be taken after sending a maximum number of retries for connection release request and having no answer from the NCC. The RCST will logoff and try to logon again and thus to re-establish the signalling connection with the NCC. The terminal will transit to the NOT READY state and try to reach the IDLE state after achieving synchronization.

The decision to logoff the terminal is conditional and not forced, as it is not part of the connection protocol itself; it is just one of the possible outcomes of a critical C2P message communication failure. Therefore, there might be other considerations for maintaining the RCST logged into the system, despite the C2P mal-functioning.

NOTE 3: An NCC logoff indication may be produced either by a TIMu messages or by not sending CMT acknowledgments to the terminal.

NOTE 4: Two categories of error causes have been identified in the State Machines: OK (for accepted requests) or KO (for rejected requests). A future version of the C2P protocol may include the specific causes for all response messages (e.g. "success" or the cause of the rejection of a C2P message). If no value is provided it is assumed that both values are possible.

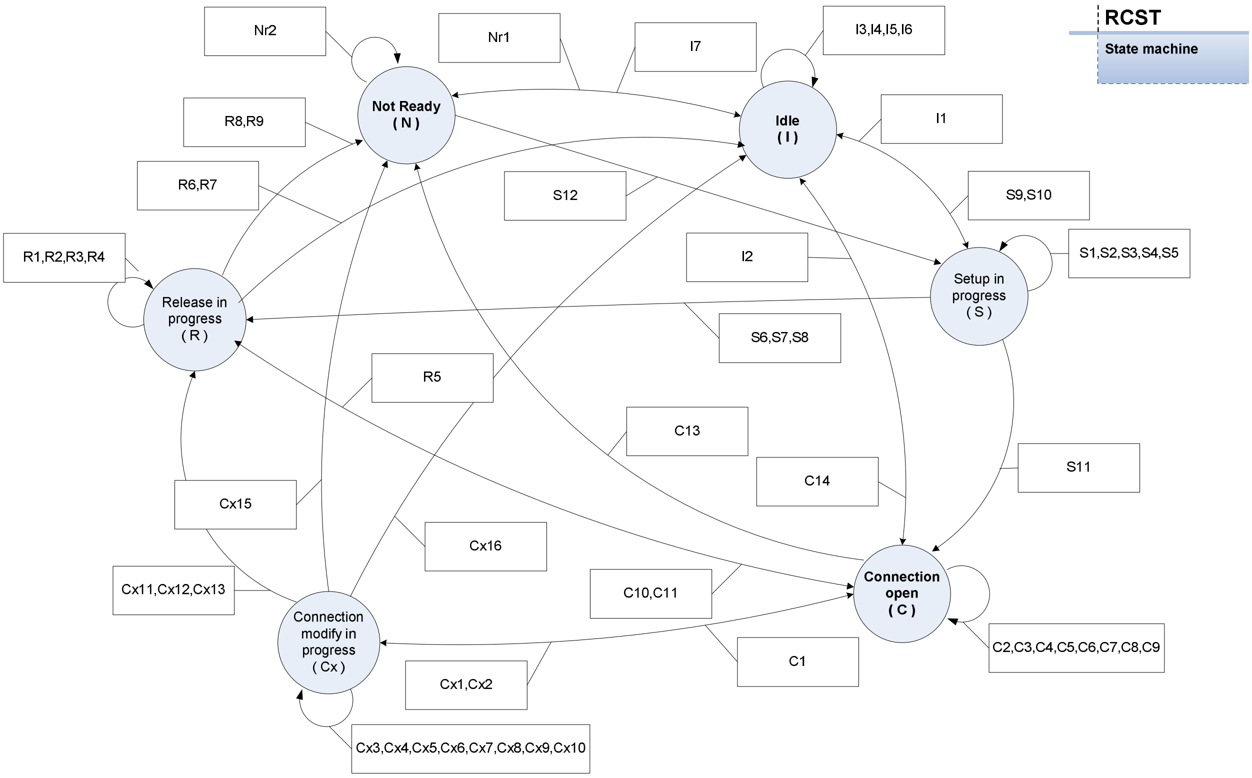


Figure 5.3: RCST/RSGW state machine diagram

Table 5.6: RCST/RSGW state machine transitions

State Diagram Label	From - To	Trigger at the Initiating RCST/RSGW or Message Received from the NCC	Other Parameters/Comments	Message Sent to the NCC/Actions

I1	Idle - Setup in progress	Connection Setup Trigger		CnxEstReq
				Start timer T-RCST_CnxEstReq;
				retry = 0

I2	Idle - Connection open	CnxEstReq	RCST CAC OK (note 2)	CnxEstResp OK
I3	Idle - Idle	CnxEstReq	RCST CAC KO	CnxEstResp KO
I4	Idle - Idle	CnxRelResp		None
I5	Idle - Idle	CnxRelReq		CnxRelResp OK (note 1)

I6	Idle - Idle	CnxModReq or CnxEstResp or CnxModResp or CnxStatusStatsReq (request status of a cnx that is not even being established)	Unexpected event: release connection to avoid inconsistencies	CnxRelReq to sender
				Start timer T-RCST_CnxRelReq;
				retry = 0; OK

I7	Idle - Not Ready	NCC Logoff Indication or RCST/RSGW Logoff trigger	RCST not synchronized	None
S1	Setup in progress - Setup in progress	CnxEstReq	Avoid cross connection set up from peer RCST/RSGW or by the NCC	CnxEstResp KO
S2	Setup in progress - Setup in progress	CnxEstResp KO	Connection establishment reject;	Stop timer T-RCST_CnxEstReq; start timer T-RCST-Wait_CnxEstReq
S3	Setup in progress - Setup in progress	Timer T-RCST_ CnxEstReq expires	retry < Counter N-RCST_ CnxEstReq;	CnxEstReq retry++;
S4	Setup in progress - Setup in progress	CnxStatusStatsReq	Send the actual connection status and all stats set to 0	CnxStatusStatsResp
S5	Setup in progress - Setup in progress	CnxModReq		CnxModResp KO

State Diagram Label	From - To	Trigger at the Initiating RCST/RSGW or Message Received from the NCC	Other Parameters/Comments	Message Sent to the NCC/Actions

S6	Setup in progress - Release in progress	Timer T-RCST_ CnxEstReq expires	Max nbr retries N-RCST_CnxEstReq reached	CnxRelReq
				Start timer T-RCST_CnxRelReq;
				retry = 0;

S7	Setup in progress - Release in progress	Release connection trigger		CnxRelReq
				Start timer T-RCST_CnxRelReq;
				retry = 0;

S8	Setup in progress - Release in progress	CnxRelResp or CnxModResp	Unexpected event: release connection to avoid inconsistencies	CnxRelReq
				Start timer T-RCST_CnxRelReq;
				retry = 0;

S9	Setup in progress - Idle	Timer T-RCST_Wait_CnxEstReq expires		None
S10	Setup in progress - Idle	CnxRelReq	NCC releases the connection	CnxRelResp OK
S11	Setup in progress - Connection open	CnxEstResp OK	Connection establishment successful;	Stop timer T-RCST_CnxEstReq;
S12	Setup in progress - Not ready	NCC Logoff Indication or Initiating RCST/RSGW Logoff trigger	RCST not synchronized	None

C1	Connection open - Connection modify in progress	Modify connection trigger		CnxModReq
				Start timer T-RCST_CnxModReq;
				retry = 0;

C2	Connection open - Connection open	CnxModReq	RCST CAC OK	CnxModResp OK
C3	Connection open - Connection open	CnxModReq	RCST CAC KO	CnxModResp KO
C4	Connection open - Connection open	CnxEstResp OK	RCST initiated connection. Connection initiated by this terminal, Retransmission from the NCC	Stop timer T-RCST_CnxEstReq
C5	Connection open - Connection open	CnxEstReq	RCST initiated connection. Avoid cross connection set up from peer RCST/RSGW or by the NCC	CnxEstResp KO
C6	Connection open - Connection open	CnxEstReq	NCC initiated connection, NCC retransmission	CnxEstResp OK
C7	Connection open - Connection open	CnxEstResp, CnxRelResp	NCC initiated connection, unexpected event	None
C8	Connection open - Connection open	CnxModResp	Unexpected event	None
C9	Connection open - Connection open	CnxStatusStatsReq	The RCST/RSGW signals the status and statistics of the requested active (open) connection;	CnxStatusStatsResp

C10	Connection open - Release in progress	Release connection trigger	Release connection initiated by this terminal	CnxRelReq
				Start timer T-RCST_CnxRelReq;
				retry = 0

C11	Connection open - Release in progress	CnxRelResp, CnxEstResp KO	RCST initiated connection Unexpected event: release resources to avoid inconsistencies;	CnxRelReq
				Start timer T-RCST_CnxRelReq;
				retry= 0

C12	Connection open - Not ready	NCC Logoff Indication or RCST Logoff trigger	RCST not synchronized	None
C13	Connection open - Idle	CnxRelReq	Connection released by the NCC or peer RCST/RSGW	CnxRelResp OK
Cx1	Connection modify in progress - connection open	Timer T-RCST-Wait_CnxModReq expires	Cnx remains open with the resources it had prior to the connection modify	None

State Diagram Label	From - To	Trigger at the Initiating RCST/RSGW or Message Received from the NCC	Other Parameters/Comments	Message Sent to the NCC/Actions
Cx2	Connection modify in progress - Connection open	CnxModResp OK	Connection modify accepted;	Stop timer T-RCST_CnxModReq;
Cx3	Connection modify in progress - Connection modify in progress	CnxModResp KO	Connection modify rejected;	Stop timer T-RCST_CnxModReq; start timer T-RCST-Wait_CnxModReq
Cx4	Connection modify in progress - Connection modify in progress	CnxModReq	Cross connection modify, requested from the peer RCST/RSGW while still pending the request from the initiating RCST/RSGW	CnxModResp KO
Cx5	Connection modify in progress - Connection modify in progress	Timer T-RCST_CnxModReq expires	retry < Counter N-RCST_ CnxModReq;	CnxModReq Start timer T-RCST_CnxModReq retry++
Cx6	Connection modify in progress - Connection modify in progress	CnxEstResp OK	RCST initiated connection Connection initiated by this RCST/RSGW, possible answer to retransmission	None
Cx7	Connection modify in progress - Connection modify in progress	CnxEstReq	RCST initiated connection Avoid cross connection set up from peer RCST/RSGW	CnxEstResp KO
Cx8	Connection modify in progress - Connection modify in progress	CnxEstReq	NCC initiated connection Retransmission from the NCC	CnxEstResp OK
Cx9	Connection modify in progress - Connection modify in progress	CnxEstResp or CnxRelResp	NCC initiated connection, Unexpected event, the connection can only be released by the NCC	None
Cx10	Connection modify in progress - Connection modify in progress	CnxStatusStatsReq	The RCST/RSGW signals the status and statistics of the requested active connection	CnxStatusStatsResp

Cx11	Connection modify in progress - Release in progress	Timer T-RCST_CnxModReq expires	Max nbr retries N-RCST_CnxModReq reached/	CnxRelReq
				Start timer T-RCST_CnxRelReq;
				retry = 0

Cx12	Connection modify in progress - Release in progress	Release connection trigger	RCST initiated the connection release	CnxRelReq
				Start timer T-RCST_CnxRelReq;
				retry = 0

Cx13	Connection modify in progress - Release in progress	CnxEstResp KO, CnxRelResp	RCST initiated connection, Unexpected event: release connection to avoid inconsistencies	CnxRelReq
				Start timer T-RCST_CnxRelReq;
				retry = 0;

Cx14	Connection modify in progress - Not ready	NCC Logoff Indication or RCST Logoff trigger	RCST not synchronized	None
Cx15	Connection modify in progress - Idle	CnxRelReq		CnxRelResp OK

R1	Release in progress - Release in progress	CnxModReq or CnxModResp or CnxEstReq or CnxEstResp	Unexpected event: release connection to avoid inconsistencies	CnxRelReq
				Start timer T-RCST_CnxRelReq;
				retry = 0

R2	Release in progress - Release in progress	Timer T-RCST_CnxRelReq expires	retry < Counter N-RCST_CnxRelReq;	CnxRelReq Start timer T-RCST_CnxRelReq; retry++
R3	Release in progress - Release in progress	CnxRelResp KO		Start timer T-RCST-Wait_CnxRelReq
R4	Release in progress - Release in progress	CnxStatusStatsReq		CnxStatusStatsResp

State Diagram Label	From - To	Trigger at the Initiating RCST/RSGW or Message Received from the NCC	Other Parameters/Comments	Message Sent to the NCC/Actions
R5	Release in progress - Connection open	Timer T-RCST-Wait_CnxRelReq expires	Try again to release the connection if release trigger is still active	None
R6	Release in progress - Idle	CnxRelResp OK		Stop timer T-RCST_CnxRelReq
R7	Release in progress - Idle	CnxRelReq		CnxRelResp OK
R8	Release in progress - Not ready	Timer T-RCST_CnxRelReq expires	Max nbr retries N-RCST_CnxRelReq reached	RCST logoffs (see note 3 from clause 5.4.1)
R9	Release in progress - Not ready	NCC Logoff Indication or RCST Logoff trigger		None
Nr1	Not ready - Idle (Ready)	Fine sync achieved	RCST synchronized	None
Nr2	Not ready - Not ready	NCC Logoff Indication or RCST Logoff trigger, CnxEstReq, Connection Setup trigger	RCST not synchronized, not ready to establish or accept connections	None
NOTE 1: This message corresponds to a CnxRelReq retransmission from the NCC, when the CnxRelResp sent by the RCST did not arrive correctly the first time. NOTE 2: As one extra check in the CAC process, the RCST should verify if there are no other pending connections with the same characteristics.				

5.4.2 NCC state machine

For the NCC state machine (figures 5.4 and 5.5) the following states have been defined.

Table 5.7: NCC states definition

State	Label	Definition
Not ready from both sides	Nrb#	Not ready to start a connection, both RCST involved in the connection are not synchronized
Not ready from one side	Nr#	Not ready to start a connection, at least one involved RCST is not synchronized
Idle	I#	Ready to start a connection
Setup in progress from one side	S#	Waiting for connection establishment response from one side
Setup in progress from both sides	Sb#	Waiting for connection establishment response from both sides
Connection open	C#	Connection open
Connection modify in progress from one side	Cx#	Waiting for connection modify in progress in progress response from one side
Connection modify in progress from both sides	Cxb#	Waiting for connection modify in progress in progress response from both sides
Release in progress from one side	R#	Waiting for connection release response from one side
Release in progress from both sides	Rb#	Waiting for connection release response from both sides

The NCC state machine in figures 5.4 and 5.5 and the description of transitions in tables 5.10 and 5.11 apply to unicast connections in the mesh transparent/regenerative scenarios. For a complete and exhaustive definition of the NCC state machine for all scenarios in SDL/UML formatting, refer to annex A.

Clauses 5.4.2.1 and 5.4.2.2 cover the two different roles of the NCC, corresponding to:

- NCC as an intermediate stage in the establishment of mesh unicast connections initiated by an RCST/RSGW (figure 5.4 and table 5.8).

- NCC as the initiator of the mesh unicast connection (figure 5.5 and table 5.9).

The following abbreviations and rules apply to both tables 5.10 and 5.11:

- C2P messages beginning with an "N" are sent by the NCC.

- C2P messages beginning with "R" refer to an RCST/RSGW (as in initiatingR, peerR, modifyingR, modifiedR, logoffR).

- Two categories of error causes have been identified in the State Machines: OK (for accepted requests) or KO (for rejected requests). A future version of the C2P protocol may include the specific causes for all response messages (e.g. "success" or the cause of the rejection of a C2P message). If no value is provided it shall be assumed that both values are possible.

For RCST initiated connections the two sides of the connection are being identified as InitiatingR and peerR. When performing a connection modify, initiated by any of the two sides, each one of them will be identified by modifyingR or modifiedR. When the sender is not specified, it can be any, i.e. the peerR or InitiatingR.

For NCC initiated connection, the two sides of the connection are being identified as "RCSTA" and "RCSTB", representing the RCST/RSGW A and RCST/RSGW B, respectively.

While establishing a new connection, the NCC should be capable to detect any possible cross connection situation. In other words, a request to establish a connection between two RCSTs with a defined C2P CoS should be rejected if there was a previously initiated on going connection establishment process between the same pair of RCSTs and with the same C2P CoS. This check should be part of the CAC done in the NCC before final acceptance of a new connection.

The following variables are used within the State Machine:

- bothSync = TRUE or FALSE to indicate if both RCSTs involved in the connection are synchronized or not.
- OtherR = contains the RCST/RSGW pending to respond a certain C2P message.
- logoffR = contains the RCST/RSGW that has been logoff.
- Sender = identifies the RCST/RSGW that sent the message.

NOTE 1: Any possible transitions not covered within the state diagram tables, imply that no actions and no changes are to be taken.

NOTE 2: Some transitions in the state diagram tables are associated with terminal's logoff, which is system specific. The logoff decisions can be made by an RCST or by the NCC, in order to resolve a critical C2P message communication failure. From the NCC's point of view, the logoff decision will be taken after sending a maximum number of retries for a C2P request and having no answer from the RCST/RSGW. The NCC will logoff the RCST/RSGW, waiting for the terminal to logon again and re-establish the C2P signalling connection with the NCC.

The decision to logoff the terminal is conditional and not forced, as it is not part of the connection protocol itself; it is just one of the possible outcomes of a critical C2P message communication failure. Therefore, there might be other considerations for maintaining the RCST logged into the system, despite the C2P mal-functioning.

NOTE 3: An NCC logoff indication may be produced either by a TIMu messages or by not sending CMT acknowledgments to the terminal.

5.4.2.1 RCST/RSGW initiated unicast connections

The NCC state machine for an RCST/RSGW initiated connections is represented in figure 5.4, while the corresponding transitions are described in table 5.8.

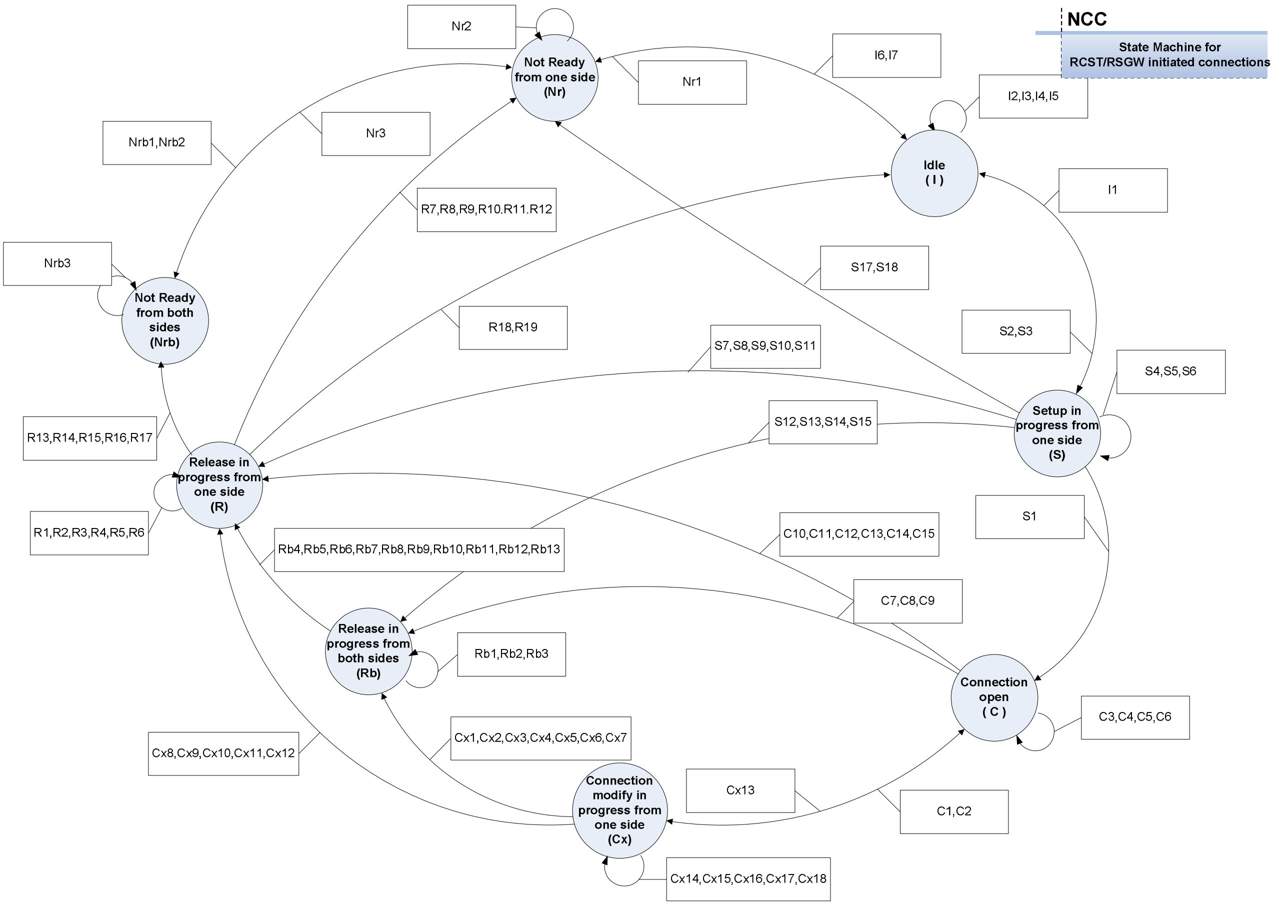

NCC
State Machine for RCST/RSGW initiated connections
Nr2
Not Ready from one side (Nr)
Nr1
I6,I7
I2,I3,I4,I5
Nrb1,Nrb2
Nr3
Idle (I)
I1
R7,R8,R9,R10.R11.R12
Nrb3
Not Ready from both sides (Nrb)
S17,S18
R18,R19
S2,S3
S7,S8,S9,S10,S11
S4,S5,S6
Setup in progress from one side (S)
R13,R14,R15,R16,R17
S12,S13,S14,S15
Release in progress from one side (R)
R1,R2,R3,R4,R5,R6
S1
Rb4,Rb5,Rb6,Rb7,Rb8,Rb9,Rb10,Rb11,Rb12,Rb13
C10,C11,C12,C13,C14,C15
C7,C8,C9
Release in progress from both sides (Rb)
Rb1,Rb2,Rb3
Connection open (C)
C3,C4,C5,C6
Cx1,Cx2,Cx3,Cx4,Cx5,Cx6,Cx7
Cx8,Cx9,Cx10,Cx11,Cx12
Connection modify in progress from one side (Cx)
Cx13
C1,C2
Cx14,Cx15,Cx16,Cx17,Cx18

Figure 5.4: NCC state machine for an RCST initiated connection

Table 5.8: NCC state machine transitions for RCST/RSGW-initiated connections

State Diagram Label	From - To	Trigger(s) or Message(s) Received	Other Parameters/Comments	Message Sent/Actions
I1	Idle - Setup in progress from one side	CnxEstReq	CAC OK Sender == InitiatingR	NCnxEstReq to peerR Retry = 0; Start timer T-NCC_CnxEstReq for peerR
I2	Idle - Idle	CnxModReq or CnxEstResp or CnxModResp	Unexpected event	NCnxRelReq to sender
I3	Idle - Idle	CnxEstReq	CAC KO Sender == InitiatingR	NCnxEstResp KO to initiatingR
I4	Idle - Idle	CnxRelResp		None
I5	Idle - Idle	CnxRelReq		NCnxRelResp (note 3)

I6	Idle - Not Ready from one side	Synchronization lost from RCST/RSGW		bothSync = False
				logoffR=initiatingR

I7	Idle - Not Ready from one side	NCC logoff trigger for an RCST/RSGW		Logoff Indication to the RCST/RSGW
				bothSync = False
				logoffR=initiatingR

S1	Setup in progress from one side - Connection open	CnxEstResp OK	Sender == peerR	Stop timer T-NCC_CnxEstReq for peerR NCnxEstResp OK to initiatingR
S2	Setup in progress from one side - Idle	CnxEstResp KO	Sender == peerR Connection rejected by peerR	Stop timer T-NCC_CnxEstReq for peerR NCnxEstResp KO to initiatingR
S3	Setup in progress from one side - Idle	CnxRelReq	Sender == peerR	Stop timer T-NCC_CnxEstReq for peerR NCnxRelResp OK to peer RCST/RSGW, NCnxEstResp KO to initiatingR

State Diagram Label	From - To	Trigger(s) or Message(s) Received	Other Parameters/Comments	Message Sent/Actions
S4	Setup in progress from one side - Setup in progress from one side	Timer T-NCC_ CnxEstReq timeout	retries < Counter N_NCC_CnxEstReq	NCnxEstReq retry to peerR; retry++; start timer T-NCC_CnxEstReq for peerR
S5	Setup in progress from one side - Setup in progress from one side	CnxEstReq	Sender == initiatingR Retry from the InitiatingR, still waiting for response from the peerR	None
S6	Setup in progress from one side - Setup in progress from one side	CnxEstReq	Sender == peerR Avoid a cross connection setup	NCnxEstResp KO to peerR
S7	Setup in progress from one side - Setup in progress from one side	CnxModReq		NCnxModResp KO to sender
S8	Setup in progress from one side - Release from one side	Timer T-NCC_ CnxEstReq timeout	Max nbr retries N-NCC_CnxEstReq exceeded	Stop timer T-NCC_CnxEstReq for peerR NCnxEstResp KO to initiatingR and NCnxRelReq to peerR Retry = 0 for peerR Start timer T-NCC_CnxRelReq for peerR

S9	Setup in progress from one side - Release from one side	Synchronization lost from the initiating RCST/RSGW		Stop timer T-NCC_CnxEstReq for peerR
				NCnxRelReq to peerR
				bothSync = False
				otherR = peerR
				logoffR = initiatingR
				Retry = 0 for peerR
				Start timer T-NCC_CnxRelReq for peerR

S10	Setup in progress from one side - Release from one side	CnxRelReq	Sender == initiatingR	Stop timer T-NCC_CnxEstReq for peerR NCnxRelResp OK to initiatingR, NCnxRelReq to peerR otherR = peerR Retry = 0 for peerR Start timer T-NCC_CnxRelReq for peerR
S11	Set up in progress from one side - Release from one side	CnxRelReq	Sender == peerR	Stop timer T-NCC_CnxEstReq for peerR NCnxRelResp OK to peerR, NCnxRelReq to initiatingR otherR = initiatingR Retry = 0 for InitiatingR Start timer T-NCC_CnxRelReq for InitiatingR

S12	Setup in progress from one side - Release from one side	NCC logoff trigger for initiating RCST/RSGW		Stop timer T-NCC_CnxEstReq for peerR
				Logoff Indication to InitiatingR,
				NCnxRelReq to peerR
				otherR = peerR
				bothSync = False
				logoffR = initiatingR
				Retry = 0 for peerR
				Start timer T-NCC_CnxRelReq for peerR

S13	Setup in progress from one side - Release from both sides	NCC Connection release trigger		Stop timer T-NCC_CnxEstReq for peerR NCnxRelReq to both sides involved in the connection Retry = 0 for InitiatingR Retry = 0 for peerR Start timers T-NCC_CnxRelReq for InitiatingR and peerR
S14	Setup in progress from one side - Release from both sides	CnxEstResp KO	Unexpected event	Stop timer T-NCC_CnxEstReq for peerR NCnxRelReq to initiatingR, NCnxRelReq to peerR Retry = 0 for InitiatingR Retry = 0 for peerR Start timers T-NCC_CnxRelReq for InitiatingR and peerR

State Diagram Label	From - To	Trigger(s) or Message(s) Received	Other Parameters/Comments	Message Sent/Actions
S15	Setup in progress from one side - Release from both sides	CnxRelResp or CnxModResp	Unexpected event: connection to be released	Stop timer T-NCC_CnxEstReq for peerR NCnxRelReq to both sides involved in the connection Retry = 0 for InitiatingR Retry = 0 for peerR Start timers T-NCC_CnxRelReq for InitiatingR and peerR
S16	Setup in progress from one side - Release from both side	CnxRelResp or CnxModResp	Sender == peerR Unexpected event: connection must be released towards one side;	Stop timer T-NCC_CnxEstReq for peerR NCnxRelReq to both sides involved in the connection Retry = 0 for InitiatingR Retry = 0 for peerR Start timers T-NCC_CnxRelReq for InitiatingR and peerR

S17	Setup in progress from one side - Not ready from one side	NCC logoff trigger for peer RCST/RSGW		Stop timer T-NCC_CnxEstReq for peerR
				Logoff Indication to peerR,
				NCnxEstReq KO to initiatingR
				otherR=InitiatingR
				bothSync = False
				logoffR = peerR
				Retry = 0 for InitiatingR
				Start timer T-NCC_CnxRelReq for InitiatingR

S18	Setup in progress from one side - Not Ready from one side	Synchronization lost from the peer RCST/RSGW		Stop timer T-NCC_CnxEstReq for peerR NCnxEstReq KO to initiatingR bothSync = False logoffR = peerR Start timer T-NCC_CnxRelReq for InitiatingR

C1	Connection open - Connection modify in progress from one side	CnxModReq	(CAC OK && Sender == InitiatingR)	NcnxModReq to modifiedR
				modifyingR = initiatingR
				modifiedR = peerR
				Retry = 0 for modifiedR
				Start timer T-NCC_CnxModReq for modifiedR

C2	Connection open - Connection modify in progress from one side	CnxModReq	(CAC OK && Sender == peerR)	NcnxModReq to modifiedR
				modifyingR = peerR
				modifiedR = initiatingR
				Retry = 0 for modifiedR
				Start timer T-NCC_CnxModReq for modifiedR

C3	Connection open - Connection open	CnxModResp	Unexpected event	None
C4	Connection open - Connection open	CnxEstResp OK	Sender == peerR	None
C5	Connection open - Connection open	CnxModReq	CAC KO	NCnxModResp KO to sender of the message received
C6	Connection open - Connection open	CnxEstReq	Sender == InitiatingR	NCnxEstResp OK

C7	Connection open - Release from both sides	CnxEstResp	Unexpected event
			Sender == InitiatingR
				NCnxRelReq to both sides involved in the connection
				Retry = 0 for initiatingR
				Retry = 0 for peerR
				Start timers T-NCC_CnxRelReq for InitiatingR and peerR

C8	Connection open - Release from both sides	NCC connection ReleaseTrigger		NCnxRelReq to both sides involved in the connection
				Retry = 0 for initiatingR
				Retry = 0 for peerR
				Start timers T-NCC_CnxRelReq for InitiatingR and peerR

State Diagram Label	From - To	Trigger(s) or Message(s) Received	Other Parameters/Comments	Message Sent/Actions

C9	Connection open - Release from both sides	CnxRelResp or CnxEstResp KO	Unexpected event: connection must be released	NCnxRelReq to both sides involved in the connection
				Retry = 0 for initiatingR
				Retry = 0 for peerR
				Start timers T-NCC_CnxRelReq for InitiatingR and peerR

C10	Connection open - Release from one side	CnxRelReq	Sender == InitiatingR	NCnxRelResp OK to initiatingR, NCnxRelReq to peerR otherR == peerR Retry = 0 for peerR Start timer T-NCC_CnxRelReq for peerR
C11	Connection open - Release from one side	CnxRelReq	Sender == peerR	NCnxRelResp OK to sender, NCnxRelReq to initiatingR otherR = initiatingR Retry = 0 for initiatingR Start timer T-NCC_CnxRelReq for InitiatingR

C12	Connection open - Release from one side	Synchronization lost from initiating RCST/RSGW		NCnxRelReq to peerR
				otherR = peerR
				bothSync = False
				Retry = 0 for peerR
				Start timer T-NCC_CnxRelReq for peerR

C13	Connection open - Release from one side	Synchronization lost from peer RCST/RSGW		NCnxRelReq to initiatingR
				otherR = initiatingR
				bothSync = False
				Retry = 0 for initiatingR
				Start timer T-NCC_CnxRelReq for InitiatingR

C14	Connection open - Release from one side	NCC logoff trigger for initiating RCST/RSGW		Logoff indication to InitiatingR,
				NCnxRelReq to peerR
				otherR = peerR
				bothSync = False
				Retry = 0 for peerR
				Start timer T-NCC_CnxRelReq for peerR

C15	Connection open - Release from one side	NCC logoff trigger for peer RCST/RSGW		Logoff indication to peerR,
				NCnxRelReq to initiatingR
				otherR = initiatingR
				Retry = 0 for initiatingR
				bothSync = False
				Start timer T-NCC_CnxRelReq for InitiatingR

Cx1	Connection modify in progress from one side - Release from both sides	CnxRelResp	Unexpected event	NCnxRelReq to both RCST/RSGWs involved in the connection Retry = 0 for initiatingR Retry = 0 for peerR Stop timer T-NCC_CnxModReq for modifiedR Start timers T-NCC_CnxRelReq for InitiatingR and peerR
Cx2	Connection modify in progress from one side - Release from both sides	CnxEstResp KO	Sender == peerR Unexpected event	NCnxRelReq to both RCST/RSGWs involved in the connection Retry = 0 for initiatingR Retry = 0 for peerR Stop timer T-NCC_CnxModReq for modifiedR Start timers T-NCC_CnxRelReq for InitiatingR and peerR
Cx3	Connection modify in progress from one side - Release from both sides	CnxEstResp	Sender == InitiatingR Unexpected event	NCnxRelReq to both RCST/RSGWs involved in the connection Retry = 0 for initiatingR Retry = 0 for peerR Stop timer T-NCC_CnxModReq for modifiedR Start timers T-NCC_CnxRelReq for InitiatingR and peerR

State Diagram Label	From - To	Trigger(s) or Message(s) Received	Other Parameters/Comments	Message Sent/Actions

Cx4	Connection modify in progress from one side -Release from both sides	CnxEstReq	(Sender == InitiatingR && Sender == ModifyingR)
			Unexpected event
				NCnxRelReq to both RCST/RSGWs involved in the connection
				Retry = 0 for initiatingR
				Retry = 0 for peerR
				Start timers T-NCC_CnxRelReq for InitiatingR and peerR

Cx5	Connection modify in progress from one side - Release from both sides	CnxModResp	Sender == ModifyingR Unexpected event	NCnxRelReq to both RCST/RSGWs involved in the connection Retry = 0 for initiatingR Retry = 0 for peerR Stop timer T-NCC_CnxModReq for modifiedR Start timers T-NCC_CnxRelReq for InitiatingR and peerR
Cx6	Connection modify in progress from one side - Release from both sides	Timer T-NCC_ CnxModReq timeout	Max nbr retries N-NCC_CnxModReq exceeded	NCnxRelReq to both RCST/RSGWs involved in the connection Retry = 0 for initiatingR Retry = 0 for peerR Stop timer T-NCC_CnxModReq for modifiedR Start timers T-NCC_CnxRelReq for InitiatingR and peerR
Cx7	Connection modify in progress from one side - Release from both sides	NCC connection ReleaseTrigger		NCnxRelReq to both sides involved in the connection Retry = 0 for initiatingR Retry = 0 for peerR Stop timer T-NCC_CnxModReq for modifiedR Start timers T-NCC_CnxRelReq for InitiatingR and peerR
Cx8	Connection modify in progress from one side - Release from one side	CnxRelReq	Sender == InitiatingR	NCnxRelResp OK to initiatingR, NCnxRelReq to peerR otherR = peerR Retry = 0 for peerR Stop timer T-NCC_CnxModReq for modifiedR Start timer T-NCC_CnxRelReq for peerR
Cx9	Connection modify in progress from one side - Release from one side	Synchronization lost from initiating RCST/RSGW	otherR == peerR	NCnxRelReq to peerR bothSync = False Retry = 0 for peerR Stop timer T-NCC_CnxModReq for modifiedR Start timer T-NCC_CnxRelReq for peerR

Cx10	Connection modify in progress in progress from one side - Release from one side	Synchronization lost from peer RCST/RSGW		NCnxRelReq to initiatingR
				otherR = initiatingR
				bothSync = False
				Retry = 0 for initiatingR
				Stop timer T-NCC_CnxModReq for modifiedR
				Start timer T-NCC_CnxRelReq for InitiatingR

Cx11	Connection modify in progress from one side - Release from one side	NCC logoff trigger for initiating RCST/RSGW		Logoff indication to initiatingR, NCnxRelReq to peerR otherR = peerR bothSync = False Retry = 0 for peerR Stop timer T-NCC_CnxModReq for modifiedR Start timer T-NCC_CnxRelReq for peerR

State Diagram Label	From - To	Trigger(s) or Message(s) Received	Other Parameters/Comments	Message Sent/Actions
Cx12	Connection modify in progress from one side - Release from one side	NCC logoff trigger for peer RCST/RSGW		Logoff indication to peerR, NCnxRelReq to initiatingR otherR = initiatingR bothSync = False Retry = 0 for initiatingR Stop timer T-NCC_CnxModReq for modifiedR Start timer T-NCC_CnxRelReq for InitiatingR
Cx13	Connection modify in progress from one side - Connection open	CnxModResp	Sender == ModifiedR	Stop timer T-NCC_CnxModReq for ModifiedR
Cx14	Connection modify in progress from one side - Connection modify in progress from one side	CnxEstResp OK	Sender == PeerR	None
Cx15	Connection modify in progress from one side - Connection modify in progress from one side	CnxEstReq	Sender == InitiatingR && Sender != ModifyingR	NCnxEstResp OK to initiating RCST/RSGW
Cx16	**Connection modify in progress from one side - Connection modify in progress from one side**	**CnxModReq**	**Sender == Modified RCST/RSGW** **NCC avoids cross connection modify in progress in progress**	**NCnxModResp KO to sender of the message received**
Cx17	**Connection modify in progress from one side - Connection modify in progress from one side**	**CnxModReq**	**Sender == Modifying RCST/RSGW** **Retry from Modifying RCST/RSGW**	**None**

Cx18	**Connection modify in progress from one side - Connection modify in progress from one side**	**Timer T-NCC_ CnxModReq timeout**	**retry < Counter N-NCC_ CnxModReq**	**NCnxModReq to Modified RCST/RSGW;**
				retry++;
				Start timer N-CnxModReq for ModifiedR

R1	**Release from one side - Release from one side**	**CnxModReq or CnxModResp or CnxEstReq or CnxEstResp**		**NCnxRelReq to sender**

R2	**Release from one side - Release from one side**	**Timer T-NCC_ CnxRelReq timeout**	**retry < Counter N-NCC_ CnxRelReq**	**NCnxRelReq to otherR;**
				retry++;
				Start timer T-NCC-CnxRelReq for otherR

R3	Release from one side - Release from one side	NCC Logoff trigger for peer RCST/RSGWs	otherR!=peerR	None
R4	Release from one side - Release from one side	NCC Logoff trigger for initiating RCST/RSGWs	otherR!=initiatingR	None

State Diagram Label	From - To	Trigger(s) or Message(s) Received	Other Parameters/Comments	Message Sent/Actions
R5	Release from one side - Release from one side	CnxRelResp from RCST/RSGW	(sender != otherR)	None
R6	Release from one side - Release from one side	CnxRelReq	(sender !=otherR)	NCnxRelResp
R7	**Release from one side - Not Ready from one side**	**Timer T-NCC_ CnxRelReq timeout**	**bothSync == True && Max nbr retries N-NCC_CnxRelReq exceeded**	**Logoff indication to otherR**
R8	Release from one side - Not Ready from one side	NCC Logoff trigger for peer RCST/RSGW	otherR==peerR && bothSync == True	Logoff Indication to peerR
R9	Release from one side - Not Ready from one side	NCC Logoff trigger for initiating RCST/RSGW	otherR==initiatingR && bothSync == True	Logoff Indication to initiatingR
R10	Release from one side - Not Ready from one side	Synchronization lost from peer RCST/RSGW	otherR==peerR && bothSync == True	Stop timer T-NCC_CnxRelReq for otherR
R11	Release from one side - Not Ready from one side	Synchronization lost from Initiating RCST/RSGW	otherR==initiatingR && bothSync==True	Stop timer T-NCC_CnxRelReq for otherR
R12	Release from one side - Not Ready from one side	CnxRelResp from RCST/RSGW	sender ==otherR && bothSync == False	Stop timer T-NCC_CnxRelReq for otherR
R13	**Release from one side - Not Ready from both sides**	**Timer T-NCC_ CnxRelReq timeout**	**bothSync == False && Max nbr retries N-NCC_CnxRelReq exceeded**	**Logoff indication to otherR**
R14	**Release from one side - Not Ready from both sides**	**NCC Logoff trigger for peer RCST/RSGWs**	**otherR==peerR && bothSync == False**	**Logoff Indication peerR**
R15	Release from one side - Not Ready from both sides	NCC Logoff trigger for initiating RCST/RSGWs	otherR==initiatingR && bothSync==False	Logoff Indication to initiatingR
R16	Release from one side - Not Ready from both sides	Synchronization lost from peer RCST/RSGW	otherR==peerR && bothSync == False	Stop timer T-NCC_CnxRelReq for otherR
R17	Release from one side - Not Ready from both sides	Synchronization lost from Initiating RCST/RSGW	otherR==initiatingR && bothSync==False	Stop timer T-NCC_CnxRelReq for otherR
R18	Release from one side - Idle	CnxRelResp from RCST/RSGW	Sender == otherR && bothSync == True	Stop timer T-NCC_CnxRelReq for otherR
R19	Release from one side - Idle	CnxRelReq	Sender == otherR	NCnxRelResp

Rb1	Release from both sides - Release from both sides	Timer T-NCC_ CnxRelReq timeout for Initiating RCST	retry < Counter N-NCC_ CnxRelReq	NCnxRelReq to Initiating RCST;
				retry++;
				Start timer T-NCC_CnxRelReq for InitiatingR

Rb2	Release from both sides - Release from both sides	Timer T-NCC_ CnxRelReq timeout for Peer RCST	retry < Counter N-NCC_ CnxRelReq	NCnxRelReq to Peer RCST;
				retry++;
				Start timer T-NCC_CnxRelReq for peerR

Rb3	Release from both sides - Release from both sides	CnxEstReq,CnxEst Resp or CnxModReq or CnxModResp		None
Rb4	Release from both sides - Release from one side	CnxRelResp from InitiatingR		otherR = peerR Stop timer T-NCC_CnxRelReq for InitiatingR
Rb5	Release from both sides - Release from one side	CnxRelResp from peerR		otherR = initiatingR Stop timer T-NCC_CnxRelReq for peerR

State Diagram Label	From - To	Trigger(s) or Message(s) Received	Other Parameters/Comments	Message Sent/Actions
Rb6	Release from both sides - Release from one side	CnxRelReq from initiating RCST/RSGW		NCnxRelResp to InitiatingR Stop timer T-NCC_CnxRelReq for InitiatingR otherR = peerR
Rb7	Release from both sides - Release from one side	CnxRelReq from peer RCST/RSGW		NCnxRelResp to peerR Stop timer T-NCC_CnxRelReq for peerR otherR= initiatingR

Rb8	Release from both sides - Release from one side	Timer T-NCC_ CnxRelReq timeout for Initiating RCST	Max nbr retries N-NCC_CnxRelReq exceeded	Logoff indication to initiatingR
				bothSync = False
				otherR = peerR
				Stop timer T-NCC_CnxRelReq for InitiatingR

Rb9	Release from both sides - Release from one side	Timer T-NCC_ CnxRelReq timeout for peer RCST	Max nbr retries N-NCC_CnxRelReq exceeded	Logoff indication peerR
				bothSync = False
				otherR = InitiatingR
				Stop timer T-NCC_CnxRelReq for peerR

Rb10	Release from both sides - Release from one side	Synchronization lost from peer RCST/RSGW		bothSync = False
				otherR = InitiatingR
				Stop timer T-NCC_CnxRelReq for peerR

Rb11	Release from both sides - Release from one side	Synchronization lost from Initiating RCST/RSGW		bothSync = False
				otherR = peerR
				Stop timer T-NCC_CnxRelReq for initiatingR

Rb12	Release from both sides - Release from one side	NCC Logoff trigger for peer RCST/RSGW		Logoff indication to peerR
				bothSync = False
				otherR = InitiatingR, logoffR=peerR
				Stop timer T-NCC_CnxRelReq for peerR

Rb13	Release from both sides - Release from one side	NCC Logoff trigger for Initiating RCST/RSGW		Logoff indication to initiatingR
				bothSync = False
				otherR = peerR, logoffR=initiatingR
				Stop timer T-NCC_CnxRelReq for initiatingR

Nrb1	Not Ready from both sides - Not Ready from one side	Fine synchronization achieved from initiating RCST/RSGW		bothSync = False
				logoffR = peerR

Nrb2	Not Ready from both sides - Not Ready from one side	Fine synchronization achieved from peer RCST/RSGW		bothSync = False
				logoffR = initiatingR

Nrb3	Not Ready from both sides - Not Ready from both sides	CnxModReq or CnxModResp or CnxEstResp or CnxRelReq or CnxRelResp or RCSTCapReq	Unexpected event	None
Nr1	Not Ready from one side - Idle	Fine synchronization achieved from RCST/RSGW	Sender == logoffR Both sides are synchronized	bothSync = True
Nr2	Not ready from one side - Not ready from one side	CnxEstReq	The other side is not yet synchronized	CnxEstResp KO to sender
Nr3	Not Ready from one side - Not Ready from both sides	NCC Triggers logoff to RCST/RSGW		Logoff Indication to the synchronized RCST/RSGW bothSync = False

NOTE 1: Whatever transition not represented in table 5.8, no action is to be performed nor change of state is needed.

NOTE 2: In general, the Connection Release Request is issued by the connection initiator, e.g. following a control/management decision or as a result of higher layer signalling. However, an RCST/RSGW-initiated connection can also be released at NCC request.

NOTE 3: This message corresponds to a CnxRelReq retransmission from one RCST, when the CnxRelResp sent by the NCC did not arrive correctly the first time.

5.4.2.2 NCC initiated unicast connections

The NCC state machine for an RCST/RSGW initiated connections is represented in figure 5.5, while the corresponding transitions are described in table 5.9.

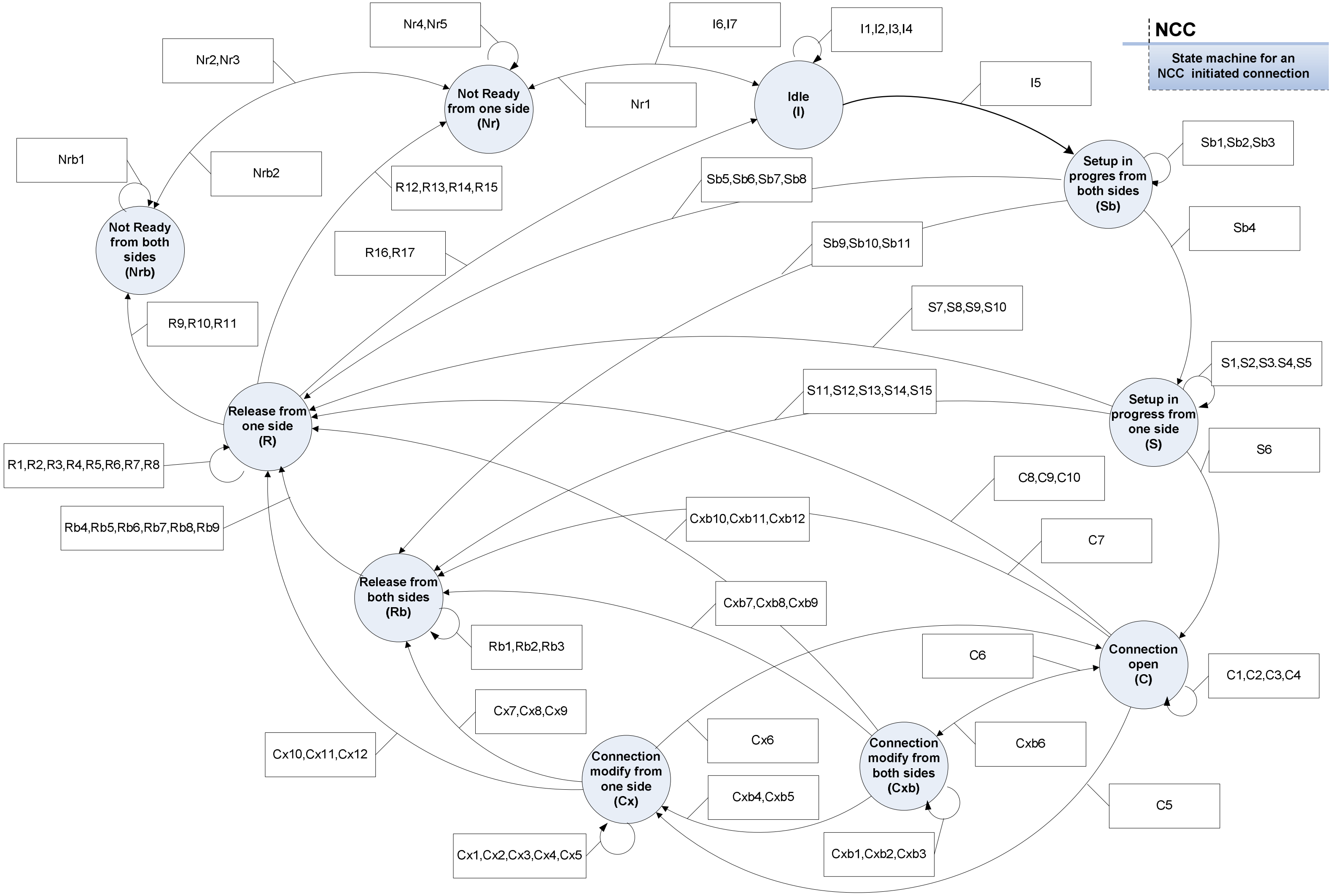


Figure 5.5: NCC State machine for an NCC initiated connection

Table 5.9: NCC state machine transitions for NCC-initiated connections

State Diagram Label	From - To	Trigger(s) or Message(s) Received	Other Parameters/Comments	Message Sent/Actions
I1	Idle - Idle	CnxEstReq or CnxModReq or CnxEstResp or CnxModResp		NCnxRelReq to Sender
I2	Idle - Idle	CnxRelReq	CAC OK	NCnxRelResp OK to Sender (note 3)
I3	Idle - Idle	CnxRelReq	CAC KO	NCnxRelResp KO to Sender
I4	Idle - Idle	CnxRelResp		None
I5	Idle - Setup in progress from both sides	NCC connection establishment trigger		NCnxEstReq to both RCSTA and RCSTB Retry for RCSTA = 0 Retry for RCSTB = 0 Start timers T-NCC_CnxEstReq for RCSTA and RCSTB

I6	Idle - Not Ready from one side	NCC logoff trigger for an RCST/RSGW		Logoff Indication to the RCST/RSGW indicated by the NCC's logoff trigger
				bothSync = False
				logoffR = RCSTA

I7	Idle - Not Ready from one side	Synchronization lost from any of the RCST/RSGWs		bothSync = False
				logoffR = RCSTA

Sb1	Setup in progress from both sides - Setup in progress from both sides	Timer T-NCC_ CnxEstReq expires for RCSTA (note 4)	retry < Counter N-NCC_CnxEstReq for RCSTA	NCnxEstReq retry to RCSTA retry++ for RCSTA Start timer T-NCC_CnxEstReq for RCSTA
Sb2	Setup in progress from both sides - Setup in progress from both sides	CnxEstReq	Unexpected event	NCnxEstResp KO to sender
Sb3	Setup in progress from both sides - Setup in progress from both sides	CnxRelReq	CAC KO	NCnxRelResp KO to sender
Sb4	Setup in progress from both sides - Setup in progress from one side	CnxEstResp OK from RCSTA	Waiting for the response from the other side (RCSTB)	Stop Timer T-NCC_CnxEstReq for RCSTA RCSTA = Sender OtherR = RCSTB
Sb5	Setup in progress from both sides - Release from one side	CnxEstResp KO from RCSTA		Stop timers T-NCC_CnxEstReq for RCSTA and RCSTB NCnxRelReq to RCSTB otherR = RCSTB Retry for RCSTB = 0 Start timer T-NCC_CnxRelReq for RCSTB
Sb6	Setup in progress from both sides - Release from one side	Synchronization lost from one RCSTA		Stop timers T-NCC_CnxEstReq for RCSTA and RCSTB otherR = RCSTB NCnxRelReq to RCSTB bothSync = False Retry for RCSTB = 0 Start timer T-NCC_CnxRelReq for RCSTB
Sb7	Setup in progress from both sides - Release from one side	NCC logoff trigger for RCSTA		Stop timers T-NCC_CnxEstReq for RCSTA and RCSTB Logoff indication RCSTA NCnxRelReq to RCSTB otherR = RCSTB bothSync = False Retry for RCSTB = 0 Start timer T-NCC_CnxRelReq for RCSTB

State Diagram Label	From - To	Trigger(s) or Message(s) Received	Other Parameters/Comments	Message Sent/Actions
Sb8	Setup in progress from both sides - Release from one side	CnxRelReq from RCSTA	CAC OK	Stop timers T-NCC_CnxEstReq for RCSTA and RCSTB NCnxRelResp to RCSTA NCnxRelReq to RCSTB otherR = RCSTB Retry for RCSTB = 0 Start timer T-NCC_CnxRelReq for RCSTB
Sb9	Setup in progress from both sides - Release from both sides	CnxModReq, or CnxModResp or CnxRelResp	Unexpected event: connection must be released	Stop timers T-NCC_CnxEstReq for RCSTA and RCSTB NCnxRelReq to both RCST A and RCSTB Retry for RCSTA = 0 Retry for RCSTB = 0 Start timers T-NCC_CnxRelReq for RCSTA and RCSTB
Sb10	Setup in progress from both sides - Release from both sides	Timer T-NCC_ CnxEstReq expires for RCSTA	Max nbr retries N-NCC_CnxEstReq exceeded for RCSTA	Stop timers T-NCC_CnxEstReq for RCSTA and RCSTB NCnxRelReq to both RCST A and RCSTB Retry for RCSTA = 0 Retry for RCSTB = 0 Start timers T-NCC_CnxRelReq for RCSTA and RCSTB
Sb11	Setup in progress from both sides - Release from both sides	NCC Connection release trigger		Stop timers T-NCC_CnxEstReq for RCSTA and RCSTB NCnxRelReq to both RCST A and RCSTB Retry for RCSTA = 0 Retry for RCSTB = 0 Start timers T-NCC_CnxRelReq for RCSTA and RCSTB
S1	Setup in progress from one side - Setup in progress from one side	Timer T-NCC_ CnxEstReq timeout for otherR	retry < N_NCC_CnxEstReq for otherR	NCnxEstReq retry to otherR retry++ for otherR Start timer T-CnxEstReq for otherR
S2	Setup in progress from one side - Setup in progress from one side	CnxEstReq	Unexpected event	NCnxEstResp KO to the sender
S3	Setup in progress from one side - Setup in progress from one side	CnxEstResp OK	Sender != otherR	None
S4	Setup in progress from one side - Setup in progress from one side	CnxRelReq	CAC KO	NCnxRelResp KO
S5	Setup in progress from one side - Setup in progress from one side	CnxModReq	Sender != otherR	NCnxModResp KO
S6	Setup in progress from one side - Connection open	CnxEstResp OK	Sender == otherR	None
S7	Setup in progress from one side - Release from one side	NCC trigger logoff for RCSTA		Stop timer T-NCC_CnxEstReq for otherR Logoff Indication to the RCSTA NCnxRelReq to RCSTB otherR = RCSTB bothSync = False Start timer T-NCC_CnxRelReq for otherR

State Diagram Label	From - To	Trigger(s) or Message(s) Received	Other Parameters/Comments	Message Sent/Actions
S8	Setup in progress from one side - Release from one side	Synchronization lost from RCSTA		Stop timer T-NCC_CnxEstReq for otherR NCnxRelReq to RCSTB otherR = RCSTB bothSync = False Start timer T-NCC_CnxRelReq for otherR
S9	Setup in progress from one side - Release from one side	CnxEstResp KO from RCSTA	Sender == otherR	Stop timer T-NCC_CnxEstReq for otherR NCnxRelReq to RCSTB otherR = RCSTB Retry for RCSTB = 0 Start timer T-NCC_CnxRelReq for otherR
S10	Setup in progress from one side - Release from one side	CnxRelReq from RCSTA	CAC OK	Stop timer T-NCC_CnxEstReq for otherR NCnxRelResp to RCSTA NCnxRelReq to RCSTB otherR = RCSTB Retry for RCSTB = 0 Start timer T-NCC_CnxRelReq for otherR
S11	Setup in progress from one side - Release from both sides	CnxEstRespKO	Sender != otherR Unexpected event	Stop timer T-NCC_CnxEstReq for otherR NCnxRelReq to both RCST A and RCSTB Retry for RCSTA = 0 Retry for RCSTB = 0 Start timer T-NCC_CnxRelReq for RCSTA and RCSTB
S12	Setup in progress from one side - Release from both sides	NCC's Connection release trigger		Stop timer T-NCC_CnxEstReq for otherR NCnxRelReq to both RCST A and RCSTB Retry for RCSTA = 0 Retry for RCSTB = 0 Start timer T-NCC_CnxRelReq for RCSTA and RCSTB
S13	Setup in progress from one side - Release from both sides	Timer T-NCC_ CnxEstReq timeout for otherR	Max nbr retries N-NCC_CnxEstReq exceeded for otherR	Stop timer T-NCC_CnxEstReq for otherR NCnxRelReq to both RCST A and RCSTB Retry for RCSTA = 0 Retry for RCSTB = 0 Start timer T-NCC_CnxRelReq for RCSTA and RCSTB
S14	Setup in progress from one side - Release from both sides	CnxModResp or CnxRelResp	Unexpected event: connection must be released	Stop timer T-NCC_CnxEstReq for otherR NCnxRelReq to both RCST A and RCSTB Retry for RCSTA = 0 Retry for RCSTB = 0 Start timer T-NCC_CnxRelReq for RCSTA and RCSTB
S15	Setup in progress from one side - Release from both sides	CnxModReq	Sender == otherR Unexpected event: connection must be released	Stop timer T-NCC_CnxEstReq for otherR NCnxRelReq to both RCST A and RCSTB Retry for RCSTA = 0 Retry for RCSTB = 0 Start timer T-NCC_CnxRelReq for RCSTA and RCSTB
C1	Connection open - Connection open	CnxModResp		None

State Diagram Label	From - To	Trigger(s) or Message(s) Received	Other Parameters/Comments	Message Sent/Actions
C2	Connection open - Connection open	CnxModReq	CAC KO	NCnxModResp KO to sender
C3	Connection open - Connection open	CnxEstResp OK		None
C4	Connection open - Connection open	CnxRelReq	CAC KO	NCnxRelResp KO to sender
C5	Connection open - Connection modify in progress from both sides	NCC connection modify in progress trigger	Both RCST/RSGW are ModifiedR	NCnxModReq to both RCST/RSGWs involved in the connection Retry for RCSTA = 0 Retry for RCSTB = 0 Start timers T-NCC_CnxModReq for RCSTA and RCSTB
C6	Connection open - Connection modify in progress from one side	CnxModReq from RCSTA	CAC OK	NCnxModReq to the RCSTB ModifyingR = RCSTA ModifiedR=RCSTB Retry for RCSTB = 0 Start timer T-NCC_CnxModReq for ModifiedR
C7	Connection open - Release from both sides	NCC Connection ReleaseTrigger or CnxRelResp or CnxEstResp KO or CnxEstReq		NCnxRelReq to both RCSTA and RCSTB Retry for RCSTA = 0 Retry for RCSTB = 0 Start timer T-NCC_CnxRelReq for RCSTA and RCSTB
C8	Connection open - Release from one side	NCC logoff trigger for RCSTA		NCC logoff indication RCSTA, NCnxRelReq to RCSTB logoffR=RCSTA bothSync=False Retry for RCSTB = 0 Start timer T-NCC_CnxRelReq for RCSTB
C9	Connection open - Release from one side	Synchronization lost from RCSTA		NCnxRelReq to RCSTB logoffR=RCSTA bothSync=False Retry for RCSTB = 0 Start timer T-NCC_CnxRelReq for RCSTB
C10	Connection open - Release from one side	CnxRelReq from RCST A	CAC OK	NCnxRelReq to RCSTB NcnxRelResp OK to RCSTA Retry for RCSTB = 0 Start timer T-NCC_CnxRelReq for RCSTB
Cxb1	Connection modify in progress from both sides - Connection modify in progress from both sides	CnxModReq	Avoid cross connection modify in progress	NCnxModResp KO to sender
Cxb2	Connection modify in progress from both sides - Connection modify in progress from both sides	Timer T-NCC_ CnxModReq timeout from RCSTA	retry < Counter N_NCC_CnxModReq for RCSTA	NCnxModReq retry to RCSTA; retry++ for RCSTA Start timer T-NCC_CnxModReq for RCSTA
Cxb3	Connection modify in progress from both sides - Connection modify in progress from both sides	CnxRelReq	CAC KO	NCnxRelResp KO to sender (note 3)

State Diagram Label	From - To	Trigger(s) or Message(s) Received	Other Parameters/Comments	Message Sent/Actions
Cxb4	Connection modify in progress from both sides - Connection modify in progress from one side	CnxEstResp OK		None
Cxb5	Connection modify in progress from both sides - Connection modify in progress from one side	CnxModResp from RCSTA	Waiting for the response from the other side	otherR = RCSTB
Cxb6	Connection modify in progress from both sides - Connection open	CnxModResp KO	Connection modify in progress has failed, connection remains with previous resources	None
Cxb7	Connection modify in progress from both sides - Release from both sides	NCC Connection release trigger		Stop timers T-NCC_CnxModReq for RCSTA and RCSTB NCnxRelReq to both RCST A and RCSTB Retry for RCSTA = 0 Retry for RCSTB = 0 Start timersT-NCC_CnxRelReq for RCSTA and RCSTB
Cxb8	Connection modify in progress from both sides - Release from both sides	CnxRelResp or CnxEstReq or CnxEstResp	Unexpected event	Stop timers T-NCC_CnxModReq for RCSTA and RCSTB NCnxRelReq to both RCST A and RCSTB Retry for RCSTA = 0 Retry for RCSTB = 0 Start timersT-NCC_CnxRelReq for RCSTA and RCSTB
Cxb9	Connection modify in progress from both sides - Release from both sides	Timer T-NCC_ CnxModReq timeout for RCSTA	Max nbr retries N-NCC_CnxModReq exceeded for RCSTA	Stop timers T-NCC_CnxModReq for RCSTA and RCSTB NCnxRelReq to both RCST A and RCSTB Retry for RCSTA = 0 Retry for RCSTB = 0 Start timersT-NCC_CnxRelReq for RCSTA and RCSTB
Cxb10	Connection modify in progress from both sides - Release from one side	NCC logoff trigger for RCSTA		Stop timers T-NCC_CnxModReq for RCSTA and RCSTB Logoff indication to RCSTA NCnxRelReq to RCSTB bothSync=False LogoffR=RCSTA Retry for RCSTB = 0 Start timer T-NCC_CnxRelReq for RCSTB
Cxb11	Connection modify in progress from both sides - Release from one side	Synchronization lost from RCSTA		Stop timers T-NCC_CnxModReq for RCSTA and RCSTB NCnxRelReq to RCSTB bothSync=False logoffR=RCSTA Retry for RCSTB = 0 Start timer T-NCC_CnxRelReq for RCSTB
Cxb12	Connection modify in progress from both sides - Release from one side	CnxRelReq from RCSTA	CAC OK	Stop timers T-NCC_CnxModReq for RCSTA and RCSTB NCnxRelReq to RCSTB NCnxRelResp OK to RCSTA Retry for RCSTB = 0 Start timer T-NCC_CnxRelReq for RCSTB

State Diagram Label	From - To	Trigger(s) or Message(s) Received	Other Parameters/Comments	Message Sent/Actions
Cx1	Connection modify in progress from one side - Connection modify in progress from one side	CnxEstResp OK		None
Cx2	Connection modify in progress from one side - Connection modify in progress from one side	CnxModResp OK, CnxModReq, CnxEstResp	Sender != otherR	None
Cx3	Connection modify in progress from one side - Connection modify in progress from one side	CnxModReq	Sender == otherR Avoid cross connection modify in progress	NCnxModResp KO to sender
Cx4	Connection modify in progress from one side - Connection modify in progress from one side	Timer T-NCC_ CnxModReq timeout for otherR	retry < Counter N_NCC_CnxModReq for otherR	NCnxModReq retry to otherR; retry++ for other Start timer T-NCC_CnxModReq for otherR
Cx5	Connection modify in progress from one side - Connection modify in progress from one side	CnxRelReq	CAC KO	NCnxRelResp KO to sender
Cx6	Connection modify in progress from one side - Connection open	CnxModResp	Sender == otherR	None
Cx7	Connection modify in progress from one side - Release from both sides	Timer T-NCC_ CnxModReq timeout for otherR	Max nbr retries N-NCC_CnxModReq exceeded for otherR	Stop timer T-NCC_CnxModReq for otherR NCnxRelReq to both RCST A and RCSTB Retry for RCSTA = 0 Retry for RCSTB = 0 Start timers T-NCC_CnxRelReq for RCSTA and RCSTB
Cx8	Connection modify in progress from one side - Release from both sides	NCC Connection release trigger		Stop timer T-NCC_CnxModReq for otherR NCnxRelReq to both RCST A and RCSTB Retry for RCSTA = 0 Retry for RCSTB = 0 Start timers T-NCC_CnxRelReq for RCSTA and RCSTB
Cx9	Connection modify in progress from one side - Release from both sides	CnxRelResp or CnxEstReq or CnxEstResp KO	Unexpected event: connection must be released	Stop timer T-NCC_CnxModReq for otherR NCnxRelReq to both RCST A and RCSTB Retry for RCSTA = 0 Retry for RCSTB = 0 Start timers T-NCC_CnxRelReq for RCSTA and RCSTB

State Diagram Label	From - To	Trigger(s) or Message(s) Received	Other Parameters/Comments	Message Sent/Actions
Cx10	Connection modify in progress from one side - Release from one side	NCC logoff trigger for RCSTA		Stop timer T-NCC_CnxModReq for otherR Logoff indication to the RCSTA NCnxRelReq to RCSTB otherR=RCSTB bothSync=False Retry for RCSTB = 0 Start timer T-NCC_CnxRelReq for RCSTB
Cx11	Connection modify in progress from one side - Release from one side	Synchronization lost from RCSTA		Stop timer T-NCC_CnxModReq for otherR NCnxRelReq to RCSTB otherR=RCSTB bothSync=False Retry for RCSTB = 0 Start timer T-NCC_CnxRelReq for RCSTB
Cx12	Connection modify in progress from one side - Release from one side	CnxRelReq from RCSTA	CAC OK	Stop timer T-NCC_CnxModReq for otherR NCnxRelReq to RCSTB NCnxRelResp OK to RCSTA Retry for RCSTB = 0 Start timer T-NCC_CnxRelReq for RCSTB
R1	Release from one side - Release from one side	CnxEstResp or CnxEstReq or CnxModReq or CnxModResp	Sender == otherR	NCnxRelReq to sender retry = 0 for Sender Start timer T-NCC_CnxRelReq for Sender
R2	Release from one side - Release from one side	CnxEstResp or CnxEstReq or CnxModReq or CnxModResp	Sender != otherR	None
R3	Release from one side - Release from one side	Timer T-NCC_ CnxRelReq timeout for otherR	retry < Counter N-NCC_CnxRelReq for otherR	NCnxRelReq to otherR retry++ for otherR Start timer T-NCC_CnxRelReq for otherR
R4	Release from one side - Release from one side	NCC Logoff trigger for RCSTA	otherR!=RCSTA	Stop timer T-NCC_CnxRelReq for otherR Logoff indication to RCSTA bothSync=False
R5	Release from one side - Release from one side	Synchronization lost from RCSTA	otherR!=RCSTA	Stop timer T-NCC_CnxRelReq for otherR bothSync=False
R6	Release from one side - Release from one side	CnxRelResp	Sender!=otherR	None
R7	Release from one side - Release from one side	CnxRelReq	CAC KO	NCnxRelResp KO to sender
R8	Release from one side - Release from one side	CnxRelReq from RCSTA	otherR != RCSTA	None
R9	Release from one side - Not Ready from both side	Timer T-NCC_ CnxRelReq timeout for otherR	Max nbr retries N-NCC_CnxRelReq exceeded for otherR bothSync==False	Stop timer T-NCC_CnxRelReq for otherR Logoff indication to the otherR
R10	Release from one side - Not Ready from both sides	NCC Logoff trigger for RCSTA	otherR==RCSTA && bothSync==False	Stop timer T-NCC_CnxRelReq for otherR Logoff indication to RCSTA
R11	Release from one side - Not Ready from both sides	Synchronization lost from RCSTA	otherR==RCSTA && bothSync==False	Stop timer T-NCC_CnxRelReq for otherR

R12	Release from one side - Not Ready from one side	Timer T-NCC_ CnxRelReq timeout for otherR	Max nbr retries N-NCC_CnxRelReq exceeded for otherR bothSync==True	Stop timer T-NCC_CnxRelReq for otherR Logoff indication to the other
				bothSync=False
				LogoffR=otherR

State Diagram Label	From - To	Trigger(s) or Message(s) Received	Other Parameters/Comments	Message Sent/Actions

R13	Release from one side - Not Ready from one side	NCC Logoff trigger for RCSTA	otherR==RCSTA && bothSync==True	Stop timer T-NCC_CnxRelReq for otherR Logoff indication to RCSTA
				bothSync=False
				logoffR=RCSTA

R14	Release from one side - Not Ready from one side	Synchronization lost from RCSTA	otherR==RCSTA && bothSync==True	Stop timer T-NCC_CnxRelReq for otherR bothSync=False logoffR=RCSTA
R15	Release from one side - Not ready from one side	CnxRelResp	Sender==otherR && bothSync==false	Stop timer T-NCC_CnxRelReq for otherR
R16	Release from one side - Idle	CnxRelResp	Sender==otherR && bothSync ==true	Stop timer T-NCC_CnxRelReq for otherR
R17	Release from one side - Idle	CnxRelReq from RCSTA	otherR==RCSTA && CAC OK	Stop timer T-NCC_CnxRelReq for RCSTA CnxRelResp OK to RCSTA
Rb1	Release from both sides - Release from both sides	CnxEstResp or CnxEstReq or CnxModReq or CnxModResp or other		None
Rb2	Release from both sides - Release from both sides	Timer T-NCC_ CnxRelReq timeout for RCSTA	retry < Counter N-NCC_ CnxRelReq for RCSTA	NCnxRelReq to RCSTA; retry++ for RCSTA Start timer T-NCC_CnxRelReq for RCSTA
Rb3	Release from both sides - Release from both sides	CnxRelReq	CAC KO	NCnxRelResp KO to sender

Rb4	Release from both sides - Release from one side	NCC logoff trigger for RCSTA		Stop timer T-NCC_CnxRelReq for RCSTA
				bothSync=False
				otherR=RCSTB

Rb5	Release from both sides - Release from one side	CnxRelResp from RCSTA		Stop timer T-NCC_CnxRelReq for RCSTA otherR= RCSTB

Rb6	Release from both sides - Release from one side	Timer T-NCC_ CnxRelReq timeout for RCSTA	Max nbr retries N-NCC_CnxRelReq exceeded for RCSTA	Stop timer T-NCC_CnxRelReq for RCSTA
				Logoff indication RCSTA (note 4)
				bothSync=False
				otherR=RCSTB
				logoffR=RCSTA

Rb7	Release from both sides - Release from one side	NCC logoff trigger for RCSTA		Stop timer T-NCC_CnxRelReq for RCSTA
				Logoff indication RCSTA
				bothSync=False
				otherR=RCSTB
				logoffR=RCSTA

Rb8	Release from both sides - Release from one side	Synchronization lost for RCSTA		Stop timer T-NCC_CnxRelReq for RCSTA
				bothSync=False
				otherR=RCSTB
				logoffR=RCSTA

Rb9	Release from both sides - Release from one side	CnxRelReq from RCSTA	CAC OK	Stop timers T-NCC_CnxRelReq for RCSTA and RCSTB NCnxRelReq to RCSTB NCnxRelResp OK to RCSTA Retry for RCSTB = 0 Start timer T-NCC_CnxRelReq for RCSTB
Nrb1	Not Ready from both sides - Not Ready from both sides	CnxEstReq or CnxModReq or CnxModResp or CnxEstResp or CnxRelReq or CnxRelResp	Unexpected event	None

State Diagram Label	From - To	Trigger(s) or Message(s) Received	Other Parameters/Comments	Message Sent/Actions
Nrb2	Not Ready from both sides - Not ready from one side	Fine synchronization from RCSTA		logoffR=RCSTB
Nr1	Not Ready from one side- Idle	Fine synchronization achieved from RCSTA	logoffR==RCSTA	bothSync=True
Nr2	Not Ready from one side - Not Ready from both sides	NCC logoff trigger for RCSTA	logoffR!=RCSTA	None
Nr3	Not Ready from one side - Not Ready from both sides	Synchronization lost from RCSTA	logoffR!=RCSTA	NCC logoff indication to RCSTB
Nr4	Not Ready from one side - Not Ready from one side	CnxModReq or CnxModResp or CnxEstResp or CnxRelReq or CnxRelResp or other	Unexpected event	None
Nr5	Not Ready from one side - Not Ready from one side	CnxEstReq	Connections cannot be processed before moving to idle state	CnxEstResp KO to sender

NOTE 1: Whatever transition not represented in table 5.9, no action is to be performed nor change of state is needed.
NOTE 2: In general, the Connection Release Request is issued by the connection initiator, e.g. following a control/management decision or as a result of higher layer signalling. However, an RCST/RSGW can also request to release an NCC-initiated connection.
NOTE 3: This message corresponds to a CnxRelReq retransmission from one RCST, the CnxRelResp sent by the NCC did not arrive correctly the first time.
NOTE 4: For a simpler representation, only the events and transitions related to RCSTA are shown. It is assumed that the same behaviour is expected for RCSTB.

6 Connection Control Procedures

This clause describes the normal procedures and a few examples of exception procedures, all consistent with the state machine diagrams in clause 5. Additional information is provided in annex A (UML state machines) and in annex C (other examples of C2P exception procedures).

The connection control procedures describe the C2P messages exchanged between the network elements in order to establish, modify or release traffic connections or to inquire into their status. The messages follow a well defined timing sequence. It is recalled that the signalling connections are established at RCST logon time based on standard DVB-RCS signalling, possibly with some adaptations.

In general, connection control procedures are differentiated depending on the type of connection (unicast/multicast, NCC-initiated, RCST-initiated, star/mesh) and the nature of the satellite payload (transparent or regenerative). The procedures identified in this clause apply to both transparent and regenerative cases, unless otherwise stated.

The procedures are described in terms of C2P messages, as identified in clause 4.5 and specified in detail in clause 8. C2P messages include parameters needed for the establishment, modification and release of a connection, or they include data associated with connection status monitoring.

The general template used to represent the connection control procedures correspond to star/mesh regenerative scenarios, for which the procedures are the most complex, involving more message exchanges. The procedures for star and mesh transparent scenarios can be derived from the procedures for regenerative scenarios, i.e. by changing the NCC to NCC/GW and by having the NCC/GW exchanging messages with only one RCST in the case of star connections. Additional clarifications or figures will be provided for the transparent reference scenarios where appropriate.

This section includes C2P procedures in which the exchange of C2P messages is successful. It also includes exception procedures considered of major relevance.

For the version of the C2P specified in the present document the procedures for the establishment/modification of RCST/RSGW initiated connections are typically triggered by an incoming IP packet. Other possible triggers have been identified in clause 7.1 in [i.5]. The procedures for the establishment/modification of NCC initiated connections can be triggered by an operator, by an application server (e.g. pre-scheduled connections) or by other events (clause 7.1 in [i.5]).

6.1 Point-to-point connection establishment procedures

Point-to-point connections are unicast connections, they are either unidirectional or bidirectional and they apply to both mesh and star topologies. For star connections, the connection establishment procedure depends on the type of satellite payload (transparent or regenerative).

Point-to-point connections can be initiated by either the RCST/RSGW or by the NCC and they can be established:

- Between two RCSTs (in mesh transparent or regenerative reference scenarios).
- Between two RSGWs (in mesh regenerative reference scenarios, where the RSGWs can be seen as terminals).
- Between an RCST and the NCC/GW (in star transparent reference scenarios).
- Between an RCST and an RSGW (in star regenerative reference scenarios).

For RCST/RSGW-initiated connections, the RCST/RSGW is referred to as initiating party. The peer party can be an RCST, an RSGW or the NCC/GW, depending on the reference scenario. If the connection is unidirectional, the initiating party is associated with the (traffic) source and the peer party is associated with the (traffic) destination. In the case of bidirectional connections, the initiating party and the peer party can be seen as both source and destination at the same time, depending on the direction.

For NCC-initiated connections, the two parties *involved in the connection*, *if different from the NCC/GW*, can both be regarded as peer parties and they *are designated as A and B (i.e. RCST/RSGW A and RCST/RSGW B) in the C2P procedures*. In the case of star transparent connections between an RCST and the NCC/GW, the terminology "initiating NCC/GW" and "RCST" is used. Unless otherwise indicated, for NCC-initiated star transparent unidirectional connections, the NCC/GW (the initiating party) is the source and the RCST is the destination. For star transparent bidirectional connections, the NCC/GW and the RCST are both source and destination at the same time, depending on the direction. See clause 4.4.2.2 for more information on terminology and traffic connections.

Once a unicast connection has been established, each RCST/RSGW involved in the connection starts monitoring the transmitted and/or received traffic activity, by setting up a T-TxTrfInactivity timer and/or a T-RxTrfInactivity timer. If no traffic is sent during the Connection Inactivity Timeout (as set in the Active Connection Table), a connection release procedure will be initiated.

6.1.1 RCST/RSGW initiated connections

6.1.1.1 Successful set up

Figure 6.1 shows the signalling (annotated and numbered) involved in a successful connection set up, valid for mesh transparent scenario, as well as for star and mesh regenerative scenarios. For the star transparent scenario, please refer to figure 6.2.

Once the connection is open and depending on whether the connection is unidirectional or bidirectional, the initiating party (an RCST or an RSGW) will set one or two timers, to monitor the traffic transmission and reception, as well as to control the lifetime (duration) of the connection. These are the Connection Inactivity Timers (T-RxTrfInactivity and T-TxTrfInactivity) listed in table 5.1.

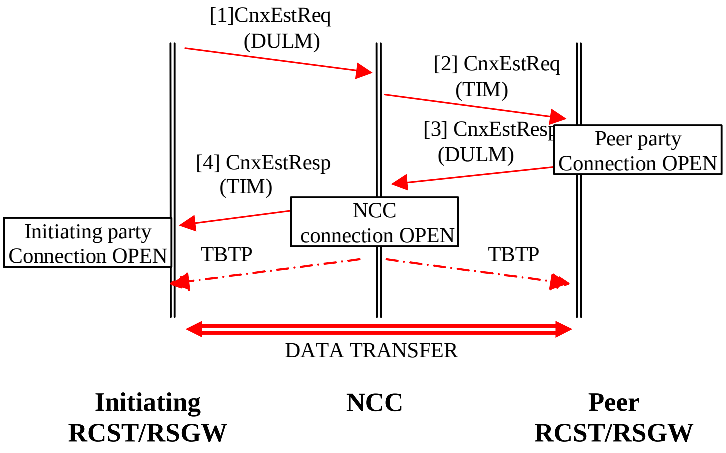


Figure 6.1: RCST/RSGW initiated Point-to-Point connection establishment: Successful set up (mesh transparent scenarios; star and mesh regenerative scenarios)

[1] The initiating RCST/RSGW requests the NCC for a connection setup and starts timer T-RCST_CnxEstReq.

[2] The NCC performs admission control checks, allocates the requested bandwidth, sends a Connection Establishment Request to the peer party and starts timer T-NCC_CnxEstReq.

[3] The peer party accepts the connection by sending the Connection Response message.

[4] Upon reception of the Connection Response message from the peer party, the NCC stops timer T-NCC_CnxEstReq and sends a Connection Response Message to the initiating RCST/RSGW. Upon reception of the Connection Response message from the NCC, the initiating party stops timer T-RCST_CnxEstReq.

Figure 6.1 also illustrates the distribution of TBTPs to both RCST/RSGWs. TBTPs' broadcast occurs independently and asynchronously relative to the C2P message. They are not part of the C2P message exchange, and neither is the transfer of IP data between the end-points of the connections (also illustrated as a "DATA TRANSFER" double arrow). The TBTPs and data transfer arrows will therefore not be represented in most of the remaining C2P procedures; they will usually be represented when the TBTP (and consequently the traffic transfer) is affected by the C2P procedure (e.g. the CRA of the RC pertaining to a connection is changed as a result of a connection establishment or modification procedure). The TBTP and DATA TRANSFER are shown in figure 6.1 merely to indicate that after a connection is established (open), the RCSTs/RSGWs can start transmitting connection's data only upon receiving suitable TBTPs (the TBTPs will be different if the RCSTs/RSGWs at the two ends of the connection are in different beams). As the RCSTs/RSGWs at the two ends of the connection begin data transmission/reception, they start transmission and/or reception inactivity timers (T-RxTrfInactivity and T-TxTrfInactivity).

In the case of a star transparent reference scenario, the connection set up takes place between an RCST and the NCC/GW, therefore steps [2] and [3] above will be omitted (see figure 6.2).

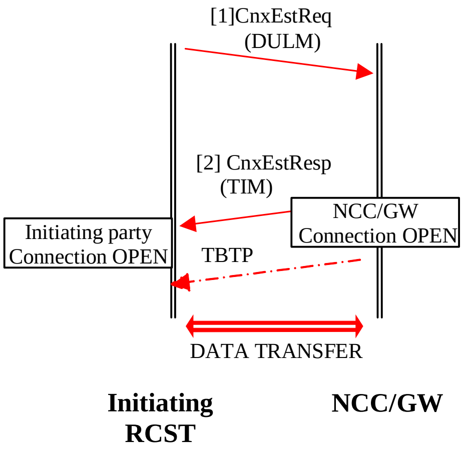


Figure 6.2: RCST initiated Point-to-Point connection establishment: Successful set up (star transparent scenario)

[1] The initiating party requests the NCC/GW for a connection setup and starts timer T-RCST_CnxEstReq.

[2] Upon reception of the Connection Request message from the initiating party, the NCC/GW stops timer T-NCC_CnxEstReq and sends a Connection Response message to the initiating party. Upon reception of the Connection Response message from the NCC/GW, the initiating party stops timer T-RCST_CnxEstReq.

6.1.1.2 Successful set up - Resolved cross connection

In this scenario, the NCC receives two connection establishment requests that involve the same pair of RCSTs/RSGWs and the same C2P CoS, i.e.: the requests (from each RCST/RSGW) are for the same connection. The NCC should continue the connection establishment procedure in response to the request received first (from the "initiating" RCST/RSGW) and reject the other request (from the "peer" RCST/RSGW) as shown in figure 6.3.

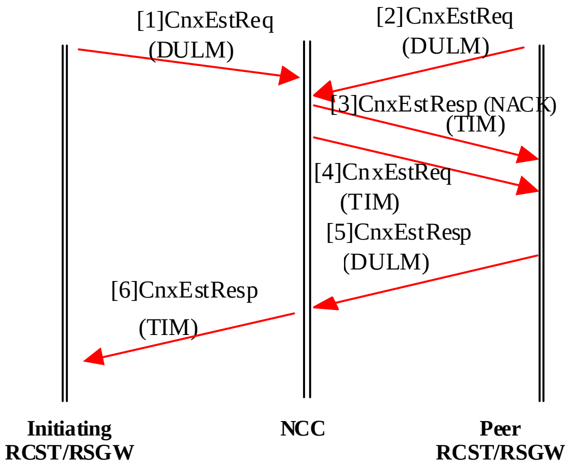


Figure 6.3: RCST/RSGW initiated Point-to-Point successful connection establishment: resolved collision (mesh transparent scenario; star and mesh regenerative scenarios)

[1] The Initiating RCST/RSGW requests the NCC for a connection setup and starts timer T-RCST_CnxEstReq.

[2] The Peer RCST/RSGW decides to establish the same type of connection, unaware of the request already being made. It sends a connection setup request and starts its timer T-RCST_CnxEstReq.

[3] The NCC performs admission control checks in response to both requests. If the second connection request corresponds to an on-going connection establishment process, the NCC rejects this second request (by sending a NACK) and continues with the establishment of the connection in response to the first request. It allocates the requested bandwidth for the connection, sends a Connection Establishment Request to the Peer RCST, and starts timer T-NCC_CnxEstReq.

[4] The Peer RCST/RSGW accepts the connection by sending a Connection Establishment Response message.

[5] Upon reception of the response from the Peer RCST/RSGW, the NCC stops T-NCC_CnxEstReq, sends a Connection Response Message to the Initiating RCST/RSGW, and sends appropriate TBTPs to both RCSTs/RSGWs.

[6] Upon reception of the Connection Response message, the Initiating RCST/RSGW stops timer T-RCST_CnxEstReq.

The NCC should handle the collision that may occur when both requests (for the same connection) are received at the same time from the two end-points of the connections (RCSTs/RSGWs). If the NCC does not succeed, the procedure becomes very inefficient (in terms of time, processing and signalling resources), as illustrated in figure 6.4 (mesh transparent scenario; star and mesh regenerative scenarios).

In figure 6.4, there are ongoing connection establishment procedures initiated by each RCST/RSGW (referred to as "Terminal A" and "Terminal B"): Terminal A sends the setup message [1a], whereas Terminal B sends the setup message [1b]. The NCC does not reject any of the request messages (for a duplicated connection), and replicates them as message [2a] and [2b] to the peer terminals. Since the terminals do not know how to handle the collision, both requests will be rejected at the terminals, as suggested by the NACK messages [3a] and [3b], from Terminal A and Terminal B respectively, replicated by NCC to the peer terminals as [4a] and [4b]. Depending on the timers expiry status, both terminals will then start connection release procedures (messages [5a] and [5b]) for the connection initiated by the other side (i.e. release message [5a] is for the setup message [2b], and release message [5b] is for the setup message [2a]). If after these connection release procedures the NCC is still unsure about the status of the connections at each terminal, it sends a release request message to each terminal (messages [9a] and [9b]), to ensure that all ongoing connections are released.

The NCC should avoid situations like that illustrated in figure 6.4, and resolve the collision according to the procedure in figure 6.3, even when the cross connection requests are simultaneous.

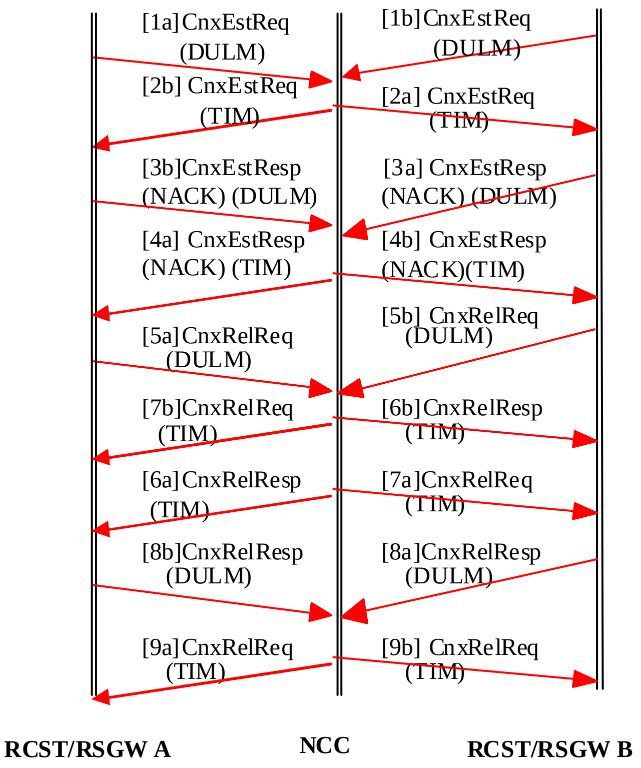


Figure 6.4: RCST/RSGW initiated Point-to-Point unsuccessful connection establishment: unresolved collision (mesh transparent scenario; star and mesh regenerative scenarios)

In star transparent reference scenarios, the collision may happen when the NCC/GW and the RCST try at the same time to establish the same type of connection. Similarly to the previous scenario, both the RCST and the NCC/GW should be aware of the collision: the RCST will always accept the connection established by the NCC/GW, the NCC/GW should reject the connection requested by the terminal. Following these rules, the connection can be successfully established, according to the procedure in figure 6.5. The unsuccessful resolution of the collision leads to the situation illustrated in figure 6.6.

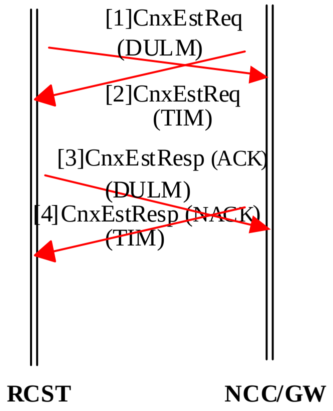


Figure 6.5: RCST initiated Point-to-Point successful connection establishment: resolved collision (star transparent scenario)

[1] The RCST requests the NCC/GW for a connection setup and starts timer T-RCST_CnxEstReq.

[2] The NCC/GW requests the same type of connection and starts timer T-NCC_CnxEstReq.

[3] The RCST accepts the connection request from the NCC/GW, even if it has already sent a request for the same type of connection.

[4] The NCC/GW verifies that the connection request received from the RCST is for a connection for which there is already an on-going connection establishment process. It therefore rejects the request from the RCST and continues with the establishment of the on-going connection.

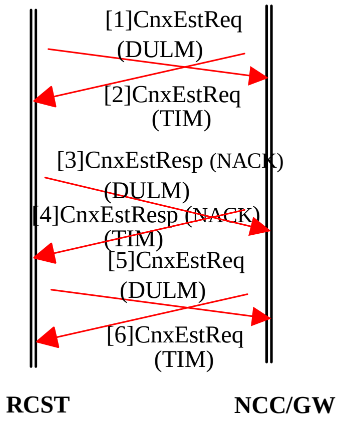


Figure 6.6: RCST initiated Point-to-Point unsuccessful connection establishment: unresolved collision (star transparent scenario)

The exchanged messages in figure 6.6 are similar to those for the case illustrated in figure 6.4.

6.1.2 NCC initiated connections

Figure 6.7 represents a successful NCC-initiated point-to-point connection establishment procedure for all but star transparent scenario, for which the applicable procedure is represented in figure 6.8.

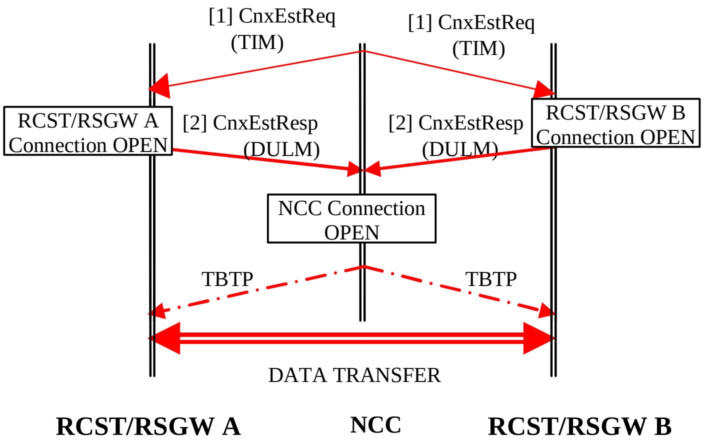


Figure 6.7: NCC-initiated Point-to-Point connection establishment: Successful set-up (mesh transparent scenario; star and mesh regenerative scenarios)

[1] NCC performs admission control checks, allocates requested bandwidth, sends Connection Establishment Request to both parties (RCST/RSGW A and RCST/RSGW B) and starts timer T-NCC_CnxEstReq for both parties.

[2] Both RCSTs/RSGWs accept the connection by sending Connection Response messages. From this moment the connection is considered open.

[3] Upon reception of the responses from both RCSTs/RSGWs, the NCC stops T-NCC_CnxEstReq and sends appropriate TBTPs to both parties.

[4] Upon reception of the TBTPs, both RCSTs/RSGWs begin data transmission.

In the case of star transparent scenario, the procedure followed for a successful NCC/GW-initiated connection set up is represented in figure 6.8.

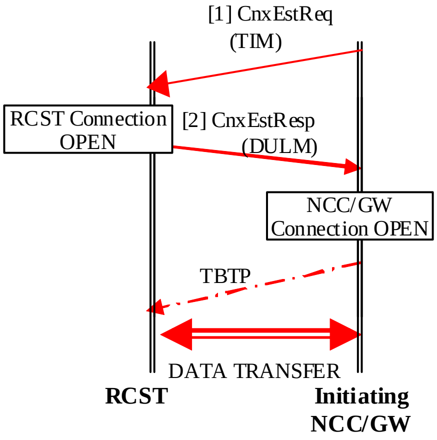


Figure 6.8: NCC-initiated Point-to-Point connection establishment: Successful set-up (star transparent scenario)

6.2 Point-to-multipoint connection establishment procedures

The point-to-multipoint (multicast) connections are unidirectional connections. The source of a multicast session can be an RCST (mesh multicast), or a Gateway (star multicast). The gateway can be either an RSGW, in the case of star regenerative multicast, or the NCC/GW, in the case of star transparent multicast. The destination of the multicast session is constituted, in general, by other RCSTs, but it can also include other RSGWs (in the case of regenerative star/mesh scenario) or the NCC/GW (in the case of transparent scenarios).

Depending on the connection initiator, three different types of multicast connections are distinguished:

- Point-to-multipoint RSGW-initiated (star/mesh regenerative reference scenario).
- Point-to-multipoint RCST-initiated (mesh regenerative/transparent reference scenario, star transparent reference scenario).
- Point-to-multipoint NCC-initiated (mesh regenerative/transparent reference scenario, star transparent reference scenario).

The terminology "multicast source" and "multicast destination" is used to discern the network element which is the source of the multicast session, from the network elements (typically terminals) which belong to the multicast session/group, a.k.a. the multicast destination parties (receivers, recipients or listeners of the multicast session).

For point-to-multipoint RCST-initiated or RSGW-initiated connection, the multicast source is the initiating RCST or RSGW.

For point-to-multipoint NCC-initiated connections, the multicast source can be either an RCST (mesh multicast scenarios) or an RSGW (star regenerative multicast scenario) or the NCC/GW (star transparent multicast scenario).

Once a multicast connection has been established, the source RCST/RSGW starts monitoring the transmitted traffic activity, by setting up a T-TxTrfInactivity timer. If no traffic is sent during the Connection Inactivity Timeout (as set in the Active Connection Table), a connection release procedure will be initiated.

The Multicast Map Table (MMT) [i.4] defines the association of PIDs with multicast IP addresses corresponding to multicast groups/sessions. It is periodically distributed by the NCC on each Transport Stream (TS), whether multicast connections are established or not. Its distribution is part of the standard DVB-RCS procedures and is not C2P specific. However, for the purpose of C2P, the MMT table should be updated to support the association of VCCs with multicast IP addresses corresponding to multicast groups/sessions. This is required in order to allow the use of the ATM format by the RCST/RSGW, for transmission and/or reception.

The MMT needs to be known by all multicast destination parties that are part of at least one multicast group. In general the destination parties can be located at different destination downlinks, reachable from the multicast source. In the case of pure transparent systems all destination parties should be at the same destination downlink. Destination parties at different destination downlinks can only be reached by two-hop multicasting - from multicast source RCST to NCC/GW and from NCC/GW to other RCSTs (clause 4.4.2.2). The MMT should therefore be periodically distributed on all destination downlinks.

In regenerative scenarios, the two-hop multicasting, while conceivable, is in general not needed due to on-board connectivity. Nevertheless an RSGW can be a destination point for a multicast connection initiated by an RCST or another RSGW.

6.2.1 RCST/RSGW initiated connections

This clause provides the RCST-initiated or RSGW-initiated procedure for successful point-to-multipoint connection establishment.

RCST-initiated point-to-multipoint connections can be from a multicast source RCST:

- Towards other RCSTs (mesh regenerative/transparent reference scenarios).
- Towards RSGWs (star regenerative reference scenario).
- Towards the NCC/GW (star transparent reference scenario).

RSGW-initiated point-to-multipoint connections can be from a multicast source RSGW:

- Towards RCSTs (star regenerative reference scenario).
- Towards other RSGWs (star/mesh regenerative reference scenario).

From the point of view of C2P procedures, the RCST-initiated or RSGW-initiated point-to-multipoint connection establishment procedures are equivalent and are representative for the regenerative reference scenarios. For the transparent reference scenario the procedures will be particularized and additional descriptions and/or figures will be provided, as needed.

A successful standard connection establishment procedure is shown in figure 6.9, where the multicast source is the initiating RCST/RSGW. Only one multicast destination party is illustrated.

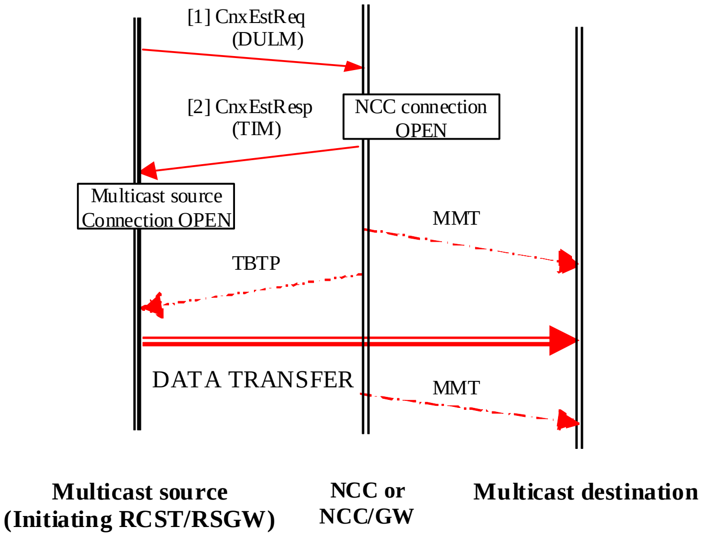


Figure 6.9: RCST/RSGW initiated Point-to-Multipoint connection establishment: Successful set-up

[1] The Initiating RCST/RSGW requests the NCC for a multicast connection setup and starts timer T-RCST_CnxEstReq.

[2] The NCC performs admission control checks, allocates the requested bandwidth, opens the connection and sends a Connection Establishment Response to the multicast source. Upon reception of the connection establishment response, the multicast source considers that the connection is open and stops the timer T-RCST_CnxEstReq.

Upon opening the connection, the NCC forwards the corresponding TBTP to the multicast source RCST/RSGW, and the updated MMT to the destination RCSTs/RSGW, to inform them about the new multicast IP source address and PID/VCC values. Once the TBTP is received by the source RCST/RSGW, it will be able to transmit multicast traffic. The NCC will continue to periodically update the MMT.

The above procedures also apply in the case of transparent reference scenarios. If two-hop multicasting is supported (see clause 4.4.2.2), the MMT should be distributed in all beams where possible multicast destination RCSTs are located.

6.2.2 NCC initiated connections

The NCC-initiated point-to-multipoint connections can be:

- from the initiating NCC/GW towards RCSTs (transparent star connections), with the NCC/GW as the star multicast source;
- from a multicast source RCST towards other RCSTs (mesh transparent/regenerative reference scenario) and possibly to RSGWs (only for regenerative scenario) or towards the NCC/GW (star transparent reference scenario, for two-hop multicast);

- from a multicast source RSGW towards RCSTs (regenerative star scenario) or towards other RSGWs (regenerative mesh scenario).

NOTE: For star transparent reference scenarios, no C2P messages will be required for the NCC-initiated connection with the NCC/GW as multicast source; there is therefore no procedure defined for this case.

Figure 6.10 shows the successful standard procedure in which the NCC requests the multicast source to open a multicast connection to multiple multicast destinations. Only one multicast destination party is illustrated.

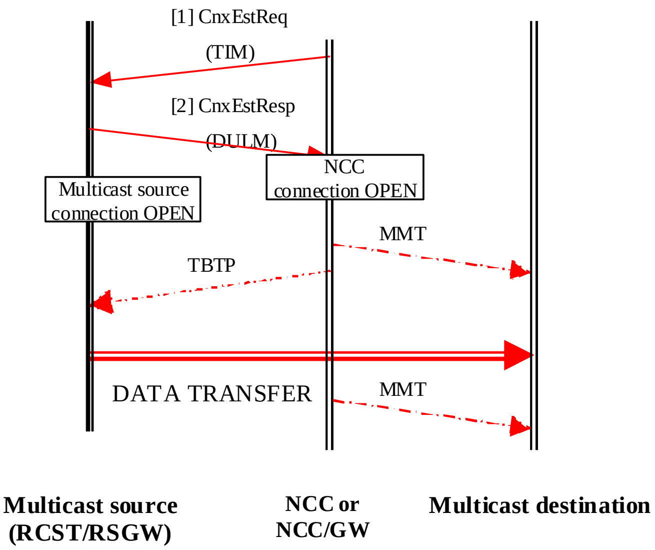


Figure 6.10: NCC initiated Point-to-Multipoint connection establishment: Successful set-up

[1] The NCC requests the multicast source RCST/RSGW for a multicast connection setup and starts timer T-NCC_CnxEstReq.

[2] The multicast source RCST/RSGW answers with a positive connection establishment response, and from this moment the multicast source considers that the connection is in open state.

Upon reception of connection establishment response, the NCC stops the timer T-NCC_CnxEstReq, considers the connection is open and updates the MMT. The NCC sends the TBTP to the multicast source so that multicast traffic can be initiated.

The above procedure also applies, in the case of the transparent reference scenario, to mesh multicasting from a source RCST to multicast destination RCSTs. If two-hop multicasting is supported (to reach RCSTs in different beams - see clause 4.4.2.2), the MMT should be distributed in all beams where possible multicast destination RCSTs are located.

6.3 Point-to-point Connection release procedures

Connection release procedures apply to all types of connections (point-to-point, point-to-multipoint, RCST-initiated, RSGW-initiated or NCC-initiated, unidirectional, bidirectional).

In general, the Connection Release Request is issued by the connection initiator, e.g. following a control/management decision or as a result of higher layer signalling. However, an RCST/RSGW-initiated connection can also be released at NCC request.

In particular and regardless of the connection initiator, a Connection Release Request can be triggered by traffic transmission and/or reception inactivity timers configured in RCSTs/RSGWs (as end-points of the connection), upon expiration of predefined timeouts (clause 5.2). In the case of NCC-initiated connection, the NCC can accept or reject a Connection Release Request from an RCST/RSGW; in the latter case it will respond with a "reject" message.

The outcome of a connection release procedure depends on whether the connection is point-to-point or point-to-multipoint.

Point-to-point connections

Connection release implies the complete closure of a connection. The connection with all its parameters will be removed from all internal data structures (Active Connection Table in RCSTs/RSGWs, NCC tables/database), and all relevant logical and bandwidth resources attached to the connection will be released at both ends of the connection and at the NCC. In the case of bidirectional connections the resources for both directions should be released.

In the case of mesh transparent scenario and mesh/star regenerative scenario, connection release messages should be used on both legs of the signalling path (between an RCST/RSGW and NCC, and between NCC and the other RCST/RSGW), regardless of the connection initiator (RCST, RSGW or NCC). In star transparent reference scenarios the signalling path only has one leg (between RCST and NCC/GW); therefore only one connection release request/response message is needed.

Point-to-multipoint connections

For point-to-multipoint (or multicast) connections, connection release may imply:

- Removal of a session, if there is no active listener (destination) of the session.

- Closure of the multicast connection, as a result of a specific request or if there are no active listeners of any session carried by the connection (see clause 4.4.2.2). The multicast connection with all its parameters will be removed from all internal data structures (Active Connection Table in RCSTs/RSGWs, NCC tables/database), and all relevant logical and bandwidth resources attached to the multicast connection will be released at the network element identified with the multicast source and at the NCC. Since point-to-multipoint connections are unidirectional, no resources need to be released at the multicast destinations. The MMT distributed to the destination parties will be updated accordingly (i.e. to reflect the release of the multicast connection).

The second case above will be considered for the multicast connection release procedures define in this clause.

6.3.1 RCST/RSGW initiated connections

In the case of RCST/RSGW-initiated connections to other RCST/RSGW (mesh transparent/regenerative scenarios, star regenerative scenario) the first connection release request (for one leg of the signalling path) could be generated either by the initiating party or by the peer party. The second connection release request (for the second leg of the signalling path) will always be generated by NCC. In the case of star transparent scenarios there will be only one connection release request, generated by either the RCST or the NCC/GW.

6.3.1.1 Successful release by the RCST/RSGW

Figure 6.11 represents the baseline scenario for a successful release of an RCST/RSGW initiated connection, in which the connection release request is issued by the initiating RCST/RSGW, e.g. after the transmission traffic inactivity timer has timed out.

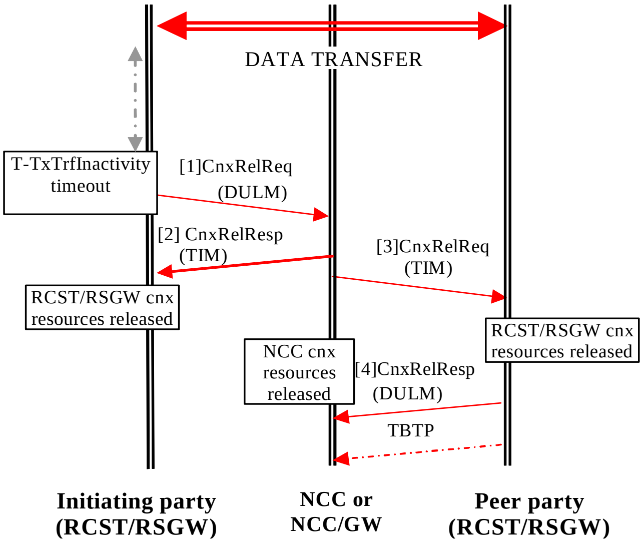

DATA TRANSFER
T-TxTrfInactivity timeout
[1]CnxRelReq (DULM)
[2] CnxRelResp (TIM)
[3]CnxRelReq (TIM)
RCST/RSGW cnx resources released
RCST/RSGW cnx resources released
NCC cnx resources released
[4]CnxRelResp (DULM)
TBTP
Initiating party (RCST/RSGW)
NCC or NCC/GW
Peer party (RCST/RSGW)

Figure 6.11: RCST/RSGW initiated Point-to-Point connection successful release: RCST/RSGW release

[1] One side of the connection requests a connection release and starts timer T-RCST_CnxRelReq.

[2] The NCC answers immediately with a connection release response to the Initiating party. Upon reception of the connection release response from the NCC, the Initiating RCST/RSGW stops the timer T-RCST_CnxRelReq, and releases the connection (removing it from the Active Connection Table) and all its resources.

[3] The NCC sends a connection release request to the other side of the connection, and start timer T-NCC_CnxRelReq.

[4] The other side (Peer RCST/RSGW) answers with a connection release response, removes the connection from the Active Connection Table and releases connection's resources. Upon reception of the connection release response, the NCC stops timer T-NCC_CnxRelReq and releases all connection's resources. The NCC updates the TBTP accordingly (i.e. it removes the CRA assignment associated with the released connection, if any).

In this procedure the initiating RCST/RSGW (i.e. the issuer of the connection release request) has to handle collision scenarios, where it could receive a connection release request (for the same connection reference id) from the NCC, instead of a connection release response. Such scenarios are covered in annex C.

In a star transparent reference scenario only steps [1] and [2] are required.

6.3.1.2 Successful release by the NCC

Figure 6.12 represents the baseline scenario, where the NCC decides to release an RCST/RSGW-initiated connection and sends connection release requests to both parties - the initiating RCST/RSGW and the peer RCST/RSGW.

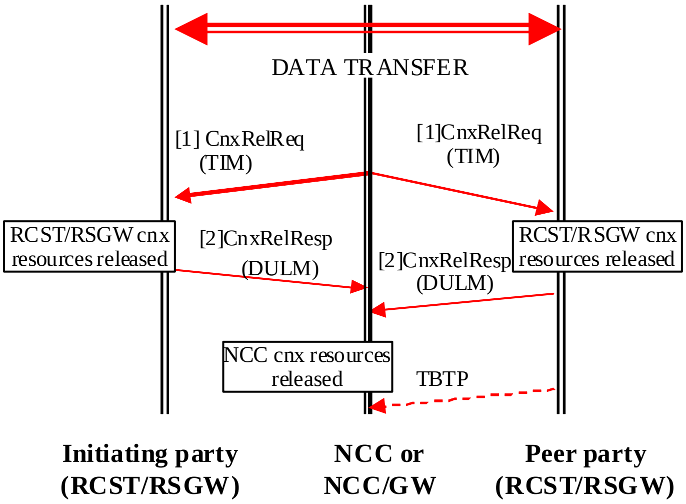


Figure 6.12: RCST/RSGW initiated Point-to-Point connection successful release: NCC release

[1] The NCC sends connection release requests to both sides of the connection and starts timer T-NCC_CnxRelReq.

[2] Each RCST/RSGW answers with a connection release response, removes the connection from the Active Connection Table and releases connection's resources. Upon reception of the connection release responses from both parties, the NCC stops the timer T-NCC_CnxRelReq and releases all connection resources. The NCC updates the TBTP accordingly (i.e. it removes the CRA assignment associated with the released connection, if any).

In this scenario, the NCC has to handle collision scenarios, in which it could receive a release request from an RCST/RSGW instead of a release response, for the same connection reference id.

In a star transparent reference scenario only release request is sent, to the initiating RCST/RSGW.

6.3.2 NCC initiated connections

For NCC-initiated connections, the two ends of the connection (parties), if different from the NCC/GW, are denoted as RCST/RSGW A and RCST/RSGW B (see clause 4.4.2.2).

Since the NCC-initiated connections are fully controlled and managed by the NCC, only the NCC can release such connections. Requests by other parties for the release of NCC-initiated connection will be rejected.

In all scenarios but the star transparent scenario, the NCC sends connection release requests to both RCST/RSGW A and RCST/RSGW B.

In star transparent scenarios, the NCC/GW only sends one connection release request, to the RCST involved in the connection.

Figure 6.13 shows a successful connection release procedure initiated by the NCC.

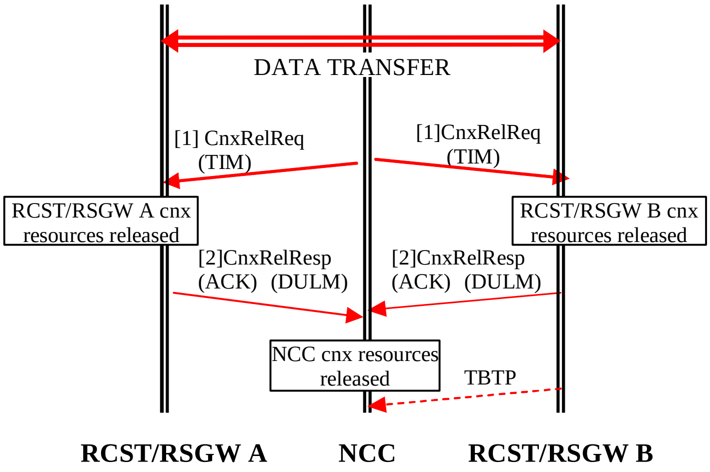


Figure 6.13: NCC initiated Point-to-Point connection successful release

[1] The NCC sends connection release requests to both sides of the connection and starts timer T-NCC_CnxRelReq.

[2] Each party answers with a connection release response, removes the connection from the Active Connection Table and releases connection's resources. Upon reception of the connection release responses from both parties, the NCC stops the timer T-NCC_CnxRelReq and releases all connection resources. The NCC updates the TBTP accordingly (i.e. it removes the CRA assignment associated with the released connection, if any)

NOTE: The above procedure is similar to the procedure for the release by NCC of an RCST/RSGW-initiated connection (clause 6.3.1.2); with minor differences regarding the terminology (e.g. RCST/RSGW A and RCST/RSGW B should be used instead of Initiating RCST/RSGW and Peer RCST/RESGW, respectively).

6.4 Point-to-multipoint connection release procedures

6.4.1 RCST/RSGW initiated connections

For RCST/RSGW-initiated connections, either the RCST/RSGW or the NCC may issue a connection release request.

After the connection is released, the corresponding MMT will be updated in all the destination beams involved.

Figure 6.14 represents a baseline successful release procedure, where the multicast source (i.e. the initiating RCST/RSGW) decides to release a multicast connection after the transmission traffic inactivity timer has timed out or the RCST/RSGW has received another connection release trigger.

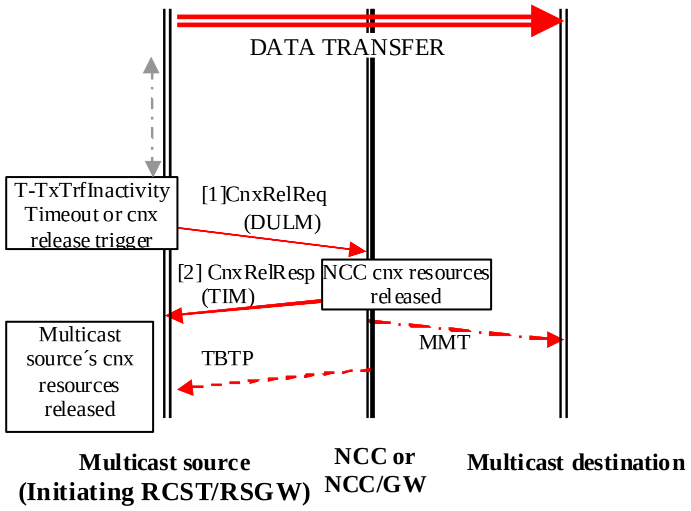

DATA TRANSFER
T-TxTrfInactivity Timeout or cnx release trigger
[1]CnxRelReq (DULM)
[2] CnxRelResp (TIM)
NCC cnx resources released
Multicast source´s cnx resources released
MMT
TBTP
Multicast source (Initiating RCST/RSGW)
NCC or NCC/GW
Multicast destination

Figure 6.14: RCST/RSGW initiated Point-to-Multipoint connection successful release: RCST/RSGW release

[1] The multicast source requests a connection release and starts timer T-RCST_CnxRelReq.

[2] The NCC answers to the multicast source with a connection release response and releases the connection resources. Upon reception of the NCC response, the multicast source stops the timer T-RCST_CnxRelReq, removes the connection from its tables/database, releases all resources linked to the connection and sends an updated MMT. The NCC also updates the TBTP accordingly (i.e. it removes the CRA assignment associated with the released connection, if any).

In this procedure, the multicast source has to handle possible collision scenarios, where it could receive a connection release request from the NCC, instead of a release response for the same connection reference id.

In the case of transparent reference scenarios, the NCC/GW can be among the multicast destinations, if two-hop multicasting is supported (clause 4.4.2.2).

The procedure for the release by the NCC of an RCST/RSGW-initiated multicast connection is similar to that described for NCC initiated connection in clause 6.4.2, with some minor differences regarding the terminology.

6.4.2 NCC initiated connections

Since the NCC-initiated connections are fully controlled and managed by the NCC, a multicast connection release request can only be generated by the NCC.

The multicast source can be and RCST, an RSGW or the NCC/GW.

Figure 6.15 represents the successful standard procedure for the release of a point-to-multipoint NCC-initiated connection, where the multicast source is an RCST/RSGW.

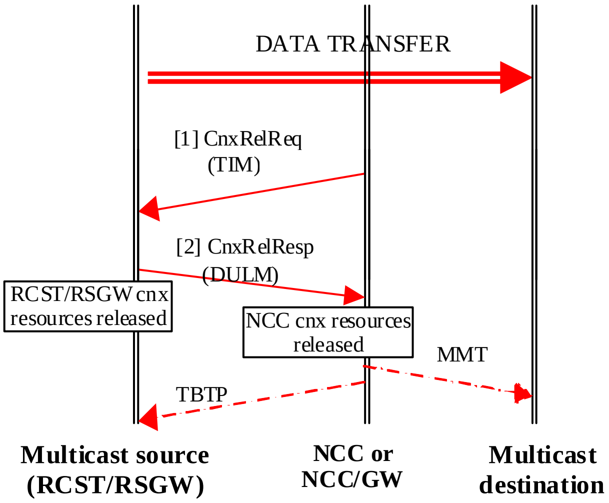


Figure 6.15: NCC initiated Point-to-Multipoint connection successful release

[1] The NCC sends a connection release request to the multicast source, and starts timer T-NCC_CnxRelReq.

[2] The RCST/RSGW multicast source answers with a connection release response, removes the connection from the Active Connection Table and releases the connection's resources. Upon reception of the RCST/RSGW's response, the NCC stops the timer T-NCC_CnxRelReq, removes the connection from its table/database , releases all the resources linked to the connection and sends an updated MMT. The NCC also updates the TBTP accordingly (i.e. it removes the CRA assignment associated with the released connection, if any)

In a star transparent reference scenario, the same procedure applies between the RCST (multicast source) and the NCC/GW. If the NCC/GW is the multicast source, no connection release request is needed; the NCC only has to distribute an updated version of the MMT.

6.5 Point-to-point connection modify procedures (rate change)

Point-to-point connection modify procedures are intended for the modification of the connection return profile and forward profile, defined in terms of rates (Return SDR/PDR, Forward SDR/PDR). A connection modify request can also result in an implicit request for the modification of the capacity allocated to the Request Class pertinent to the connection (see clause 6.7).

Point-to-point connection modify procedures can be triggered by NCC or by any party involved in the transmission of the traffic carried by the connection. They apply to all types of connections, whether initiated by an RCST, an RSGW or by the NCC.

For mesh regenerative/transparent scenarios and star regenerative scenarios, the connection modify procedures involve both legs of the signalling path: from one RCST/RSGW to NCC and from NCC to the other RCST/RSGW.

In the case of star transparent scenarios, only one leg of the signalling path is involved, between the RCST and the NCC/GW (one end of the connection is the NCC/GW).

6.5.1 RCST/RSGW initiated connections

Figure 6.16 represents a successful connection modify procedure applicable to an RCST/RSGW-initiated connection and triggered by the initiating RCST/RSGW.

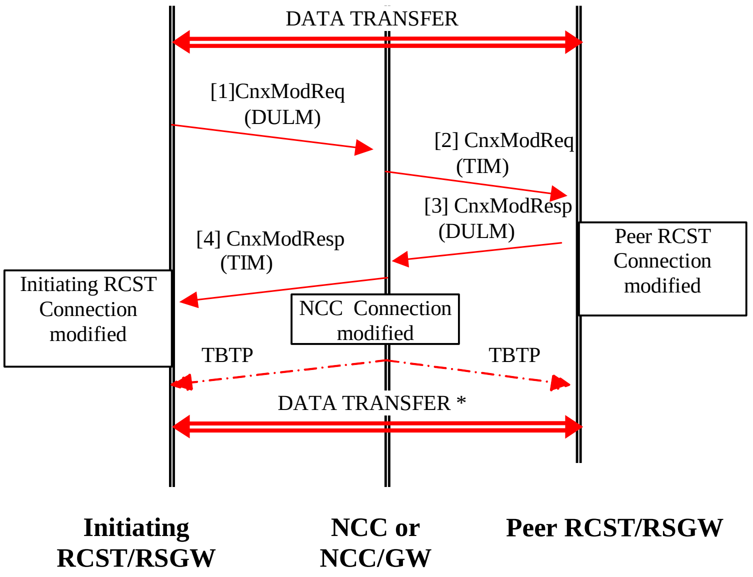


Figure 6.16: RCST/RSGW initiated Point-to-Point connection successful modification: RCST/RSGW modify

[1] The initiating RCST/RSGW requests the NCC for a connection modify and starts timer T-RCST_CnxModReq.

[2] The NCC performs admission control checks, changes the affected connection profile(s), sends a Connection Modify Request to the peer party (RCST or RSGW) and starts timer T-NCC_CnxModReq.

[3] The peer party accepts the Connection Modify Request, updates accordingly the connection profile in the Active Connection Table and sends the Connection Modify Response message.

[4] Upon reception of the Connection Modify Response message from the peer party, the NCC stops timer T-NCC_CnxModReq and sends a Connection Modify Response Message to the RCST/RSGW. Upon reception of the Connection Modify Response message from NCC, the initiating RCST/RSGW stops timer T-RCST_CnxModReq and updates the connection profile in the Active Connection Table.

The NCC also updates the TBTP accordingly (i.e. it changes the CRA assignment, if affected by the connection modification).

The procedure is the same for the case where the connection modification was triggered by the peer RCST/RSGW.

With the exception of steps [2] and [3], the procedure also applies to star transparent scenarios.

The procedure for a connection modify requested by the NCC is similar to that in clause 6.5.2 for NCC-initiated connections, with minor differences in terminology (i.e. Initiating RCST/RSGW and Peer RCST/RSGW should be used instead of RCST/RSGW A and RCST/RSGW B, respectively).

6.5.2 NCC initiated connections

Figure 6.17 represents a successful connection modify procedure applicable to an NCC-initiated connection and triggered by the initiating NCC.

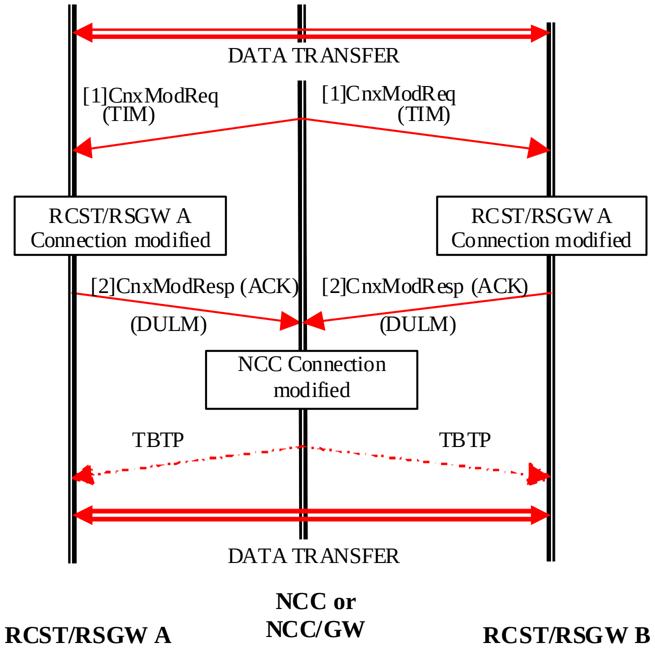


Figure 6.17: NCC initiated Point-to-Point connection successful modification: NCC modify

[1] The NCC sends connection modify requests to both parties of the connection and starts timer T-NCC_CnxRelReq.

[2] Each party changes accordingly the connection profile in its Active Connection Table and answers with a connection modify response. Upon reception of the connection modify responses from both parties, the NCC stops the timer T-NCC_CnxRelReq, modifies the parameters of the connection in its table/database and updates the TBTP accordingly (e.g. to reflect any change to the CRA assignment, if affected by the connection modification).

In the case of star transparent reference scenario the above procedure only applies to the exchange of messages between NCC and one RCST/RSGW.

The procedure for a connection modify requested by an RCST/RSGW is similar to that in clause 6.5.1 for RCST/RSGW-initiated connections, with minor differences in terminology (e.g. RCST/RSGW A and RCST/RSGW B should be used instead of Initiating RCST/RSGW and Peer RCST/RESGW, respectively).

6.6 Point-to-multipoint connection modify procedures (rate change)

Since point-to-multipoint connections are unidirectional, point-to-multipoint connection modify procedures are used for the modification of the return profile of a connection (Return SDR/PDR) and possibly of the capacity allocated to the Request Class pertinent to the connection (see clause 6.7).

Similar to point-to-point connection modify procedures (clause 6.5), point-to-multipoint connection modify procedures can be triggered by the NCC or by any party involved in the transmission of the traffic carried by the connection, and they apply to all types of connections, whether initiated by an RCST, an RSGW or by the NCC.

The multicast destinations consist in general of other RCSTs. In the case of regenerative scenarios, a destination may also include an RSGW. In the case of transparent reference scenarios, the NCC/GW may also be among the multicast destinations, if two-hop multicasting is supported (clause 4.4.2.2).

6.6.1 RCST/RSGW initiated connections

Figure 6.18 represents a baseline successful multicast connection modify procedure, where the multicast source (the initiating RCST/RSGW) issues the connection modify request.

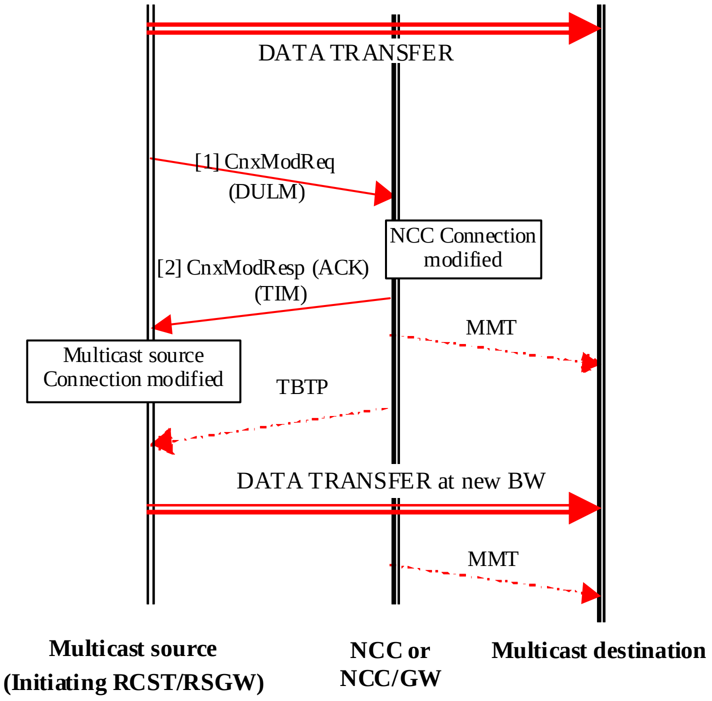


Figure 6.18: RCST/RSGW initiated Point-to-Multipoint successful connection modify: Initiating RCST/RSGW modify

[1] The multicast source requests the NCC for a multicast connection modify and starts timer T-RCST_CnxModReq.

[2] The NCC performs admission control checks, changes the connection Return Profile in its tables/database and sends a Connection Modify Response to the multicast source. Upon reception of the Connection Modify Response, the multicast source updates the Return Profile in the Active Connection Table and stops timer T-RCST_CnxModReq.

The NCC continues to distribute the TBTP and the MMT. The TBTP may need to be updated as a result of connection modification, e.g. to reflect a change to CRA assignment.

The procedure for the modification by the NCC of an RCST/RSGW-initiated multicast connection is similar to that described for NCC-initiated connection in clause 6.6.2, with some minor differences regarding the terminology.

6.6.2 NCC initiated connections

Figure 6.19 shows a procedure for the successful modification of an NCC-initiated multicast connection between a multicast source RCST/RSGW and a number of destinations (one illustrated), triggered by the initiating NCC.

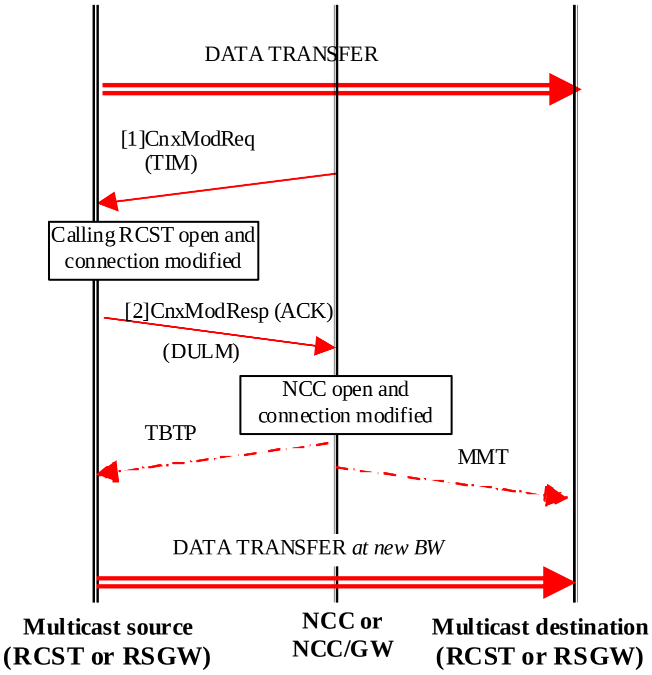

DATA TRANSFER
[1]CnxModReq
(TIM)
Calling RCST open and connection modified
[2]CnxModResp (ACK)
(DULM)
NCC open and connection modified
TBTP
MMT
DATA TRANSFER *at new BW*
Multicast source (RCST or RSGW)
NCC or NCC/GW
Multicast destination (RCST or RSGW)

Figure 6.19: NCC initiated Point-to-Multipoint successful connection modify: NCC modify

[1] The NCC requests the multicast source for a connection modify and starts timer T-NCC_CnxModReq.

[2] The multicast source accepts the connection modification request, updates the Return Profile in the Active Connection Table and sends a Connection Modify Response to the NCC. Upon reception of the Connection Modify Response, the NCC stops timer T-NCC_CnxModReq. It continues sending the TBTP and MMT. The TBTP may need to be updated as a result of connection modification, e.g. to reflect any change to CRA assignment.

The procedure for the modification by an RCST/RSGW of an NCC-initiated multicast connection is similar to that described for RCST/RSGW-initiated connections in clause 6.6.1, with some minor differences regarding the terminology.

6.7 RC modify procedures

RC modify messages are used to modify the RC_Capacity_Parameters (CRA, RBDCMax and VBDCMax) of a Request Class associated with a Channel_ID. As such, they are not connection control messages in strict sense, and they are not intended to modify the parameters of individual connections. They are rather used to modify the transmit profile of the aggregate of connections mapped to the Request Class, expressed as RC_Capacity_Parameters, but are agnostic of the individual connections in the aggregate (which can be of any type). The RC modify messages can be associated with an individual connection (and a Connection Reference ID) only in the particular case of one connection mapped to the Request Class.

For all reference scenarios, an RC Modify Request can be explicitly initiated by the Initiating RCST/RSGW or by the NCC. For mesh reference scenarios, if the Initiating RCST/RSGW is engaged in a bidirectional connection with other RCST/RSGW, the Peer RCST/RSGW may also initiate an RC Modify Request of its own.

When initiated by NCC, separate RC Modify Requests may be sent to each RCST/RSGWs.

An RC Modify Request can also be implicitly initiated/triggered by an RCST/RSGW or the NCC, as a result of the setting-up, modification or release of connections (see clause 4.6.3.4), e.g. when the RC resources are insufficient to accommodate a new connection or the modification of an existing connection.

The Request Classes are RCST/RSGW-specific entities, corresponding to the MAC Classes of Service (MAC CoSs) supported in the NCC. They are characterized by capacity categories limit values (CRA, RBDCMax and VBDCMax). An RC and the corresponding MAC CoS are identified by the same Channel_ID. The RCs in different RCSTs/RSGWs with the same Channel_ID are mapped to the same MAC CoS. The resources allocated at NCC level to a given MAC CoS partition should match the summation of all resources allocated to all RCSTs for the RC mapped to that MAC CoS.

The RC Modify Requests are intended to modify the resources allocated to individual RCs, and not those of the corresponding MAC CoS configured in NCC. The MAC CoS resources at NCC level are quasi-static and they are in general allocated/modified by a network management function. Such function could be triggered by an RC Modify Request, e.g. when MAC CoS resources are insufficient to accommodate the additional resources requested for an RC. Dynamic modification of the MAC CoS resources is considered outside the scope of the C2P Specification defined in the present document.

6.7.1 RCST/RSGW initiated RC modify

Figure 6.20 shows a successful exchange of messages in the case of an RCST/RGSW-initiated RC modify procedure.

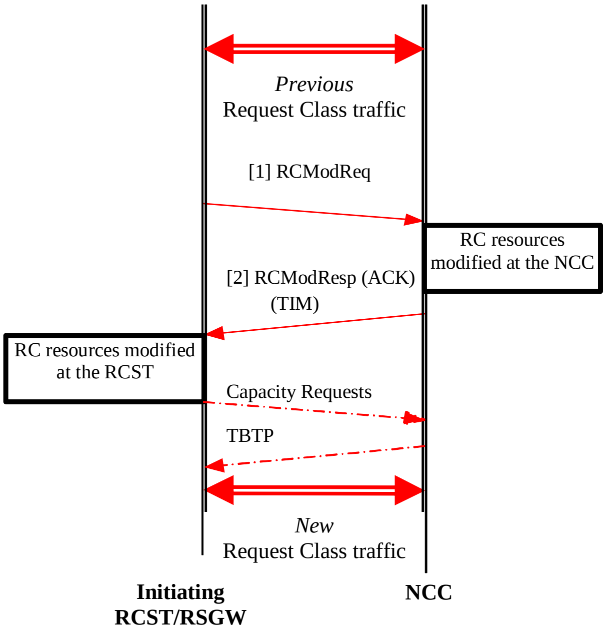


Figure 6.20: RCST/RSGW initiated successful RC modify

[1] The terminal requests the NCC for the modification of an RC (mapped to a MAC CoS at NCC level) and starts timer T-RCST_RCModReq.

[2] The NCC checks the availability of resources within the corresponding MAC CoS partition, for the RC modification. If resources are available (even partially), the NCC sends an RC Modify Response message to the terminal, with the new RC resources. If additional resources are not available, the RC Modify Response will include the currently allocated RC resources.

Upon reception of the NCC's response, the RCST/RSGW stops timer T-RCST_RCModReq and starts issuing capacity requests for the RC within the newly updated capacity limit values.

6.7.2 NCC initiated RC modify

The NCC may also ask the terminal (RCST or RSGW) to modify the resources allocated to a Request Class supported by the terminal, irrespective of its role in various connections that might be active at a given time.

Figure 6.21 shows a successful exchange of messages between the NCC and a terminal in the case of an NCC-initiated RC modify procedure.

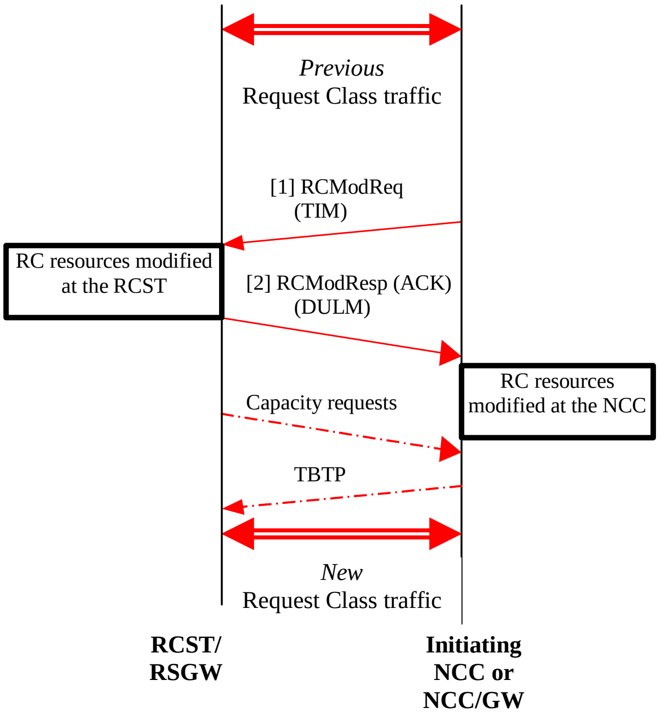


Figure 6.21: NCC initiated successful RC modify

[1] The NCC requests a terminal (either an RCST or an RSGW) for the modification of the Request Class resources and starts timer T-NCC_RCModReq.

[2] The terminal checks if the requested modifications can be accommodated within its current limitations (e.g. maximum transmit rate, resource allocated to other RCs), and it sends an RC Modify Response message, including the RC resources it can accept. The terminal can then start issuing capacity requests for the RC within the newly updated capacity limit values. Upon reception of the terminal's response, the NCC stops timer T-RCST_RCModReq, updates the resources for the modified RC, accepts capacity requests within the newly updated capacity limit values and assign capacity (in TBTP) accordingly.

This procedure applies to all reference scenarios.

6.8 RCST Capability request/response procedures

According to clause 4.6.2, after a successful admission into the network and successful establishment of signalling connection(s), an RCST (or RSGW) supporting dynamic connectivity may initiate a second logon phase. The information exchanged in this phase includes the C2P version, the IP protocol version and the multicast option, and it is transmitted within a special Information Element in the "RCST Capability Request" C2P message. The NCC may answer to this request by an "RCST Capability Response" C2P message.

This clause includes the most relevant procedures related to the exchange of messages in the "second logon phase" (clause 6.8.1). It also includes the rational and the message exchanged in the case of NCC-initiated RCST Capability Request (clause 6.8.2).

6.8.1 RCST/RSGW initiated RCST Capability request

Figure 6.22 represents a successful RCST/RGSW-initiated Capacity Request procedure.

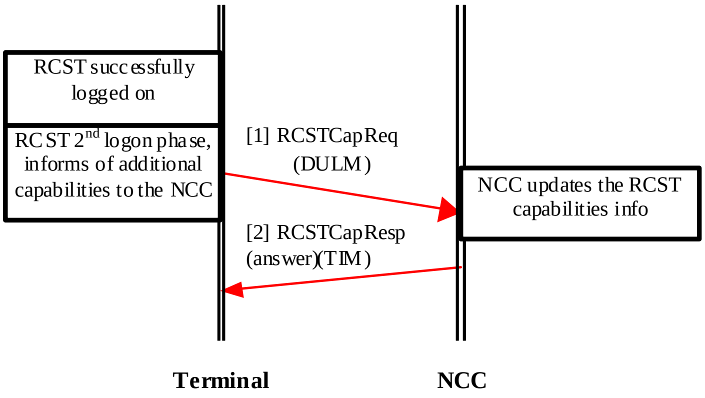

RCST successfully logged on
RCST 2nd logon phase, informs of additional capabilities to the NCC
[1] RCSTCapReq (DULM)
NCC updates the RCST capabilities info
[2] RCSTCapResp (answer)(TIM)
Terminal
NCC

Figure 6.22: RCST-initiated successful RCST Capability request

[1] After a successful logon, the RCST may optionally start a second logon phase, to notify additional parameters towards the NCC.

[2] The NCC acknowledges the information provided by the RCST by means of a RCST Capability Response.

6.8.2 NCC initiated RCST Capability request

The RCST/RSGW Capability Request may also be initiated by the NCC. The NCC-initiated Capability Request procedure allows the NCC to check the additional capabilities of the RCST any moment after a successful logon. This situation could be required, for example, after a graceful handover of NCC, in which the session logon information is not lost but is worth checking the additional capabilities information related to C2P, before any RCST (or the NCC/GW or RSGW) can initiate the establishment traffic connections based on C2P.

Figure 6.23 represents a successful NCC-initiated RCST Capability Request procedure.

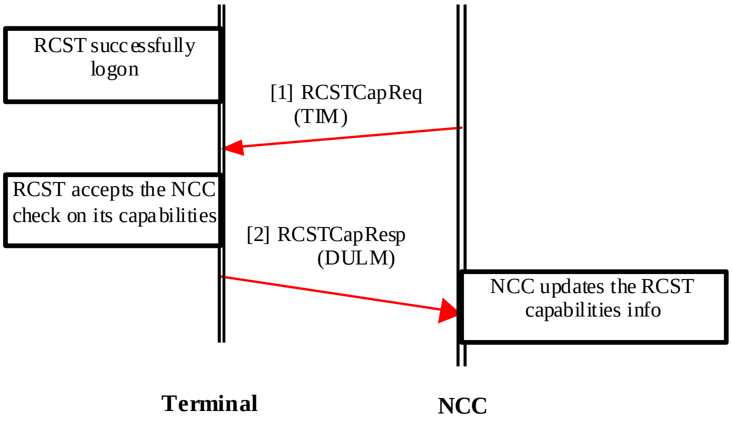

RCST successfully logon
[1] RCSTCapReq (TIM)
RCST accepts the NCC check on its capabilities
[2] RCSTCapResp (DULM)
NCC updates the RCST capabilities info
Terminal
NCC

Figure 6.23: NCC-initiated successful RCST Capability Request

[1] Due to an NCC graceful handover, or just for maintenance purposes, the NCC may request the RCST for additional capabilities after verifying that the RCST is correctly synchronized.

[2] The RCST should always reply to the RCST Capability Request message with the RCST Capability Response message.

NOTE 1: An NCC-initiated Capability Request message is intended for query purposes, since the NCC cannot modify the queried terminal's capabilities.

NOTE 2: This procedure applies to all reference scenarios.

7 Connection Control Messages Formatting

7.1 Overview

The message formatting described in this clause shall be used for all C2P messages.

The messages supported by the Connection Control Protocol have been identified in clause 4.5. They are carried in either DULM format - from RCST/RSGW to NCC, or in TIM unicast (TIMu) format - from NCC to RCST/RSGW. The message formats are described in terms of Information Elements.

NOTE: Clause 8 provides a detailed description of each C2P message.

Clause 7.2 defines all the Information Elements and their usage with the DULM format, for both ATM and MPEG2-TS encapsulation methods.

Clause 7.3 defines the Connection Control Descriptor to be used with the TIMu format.

7.2 RCST to NCC messages (DULM)

7.2.1 DULM format

Data Unit Labelling Method (DULM) is a message based method that allows an RCST/RSGW to transmit control and/or management information to NCC, in the payload of Data Units within TRF bursts already assigned to the RCST/RSGW. The data units are either ATM cells or MPEG2-TS packets, depending on the encapsulation mode used in the RCST/RSGW. The information carried in the data units is in the form of Information Elements (IEs).

7.2.1.1 DULM with ATM formatting

A DULM message in ATM format (table 7.1) is composed of an integer number (between 1 and 64) of Information Elements. Each IE shall have the format described in [1], and is made of 2 bytes of header (IE type and IE segment length) plus a body of n bytes, with n between 1 and 512. A given DULM message may be composed of IEs of different lengths (e.g. m bytes for IE Body (1) and p bytes for IE Body (2) in table 7.1).

Table 7.1: DULM Message in ATM Format

Message field	Description	Length
IE Type (1)	Type of information carried by IE	7 bits
IE Segment Length (1)	Length of the part of the IE included in this ATM packet	9 bits
IE Body (1)	IE (1) content	m*8 bits
IE Type (2)	Type of information carried by IE	7 bits
IE Segment Length (2)	Length of the part of the IE included in this ATM packet	9 bits
IE Body (2)	IE (2) content	p*8bits

The DULM message is transmitted using standard AAL5 mechanisms, as specified in ITU-T Recommendation I.363-5 [2].

The Information Elements for ATM TRF case shall be as defined in table 7.2, which is a version of table 18 in [1] with minor changes (in fair font) regarding the references in the IE body column and the reservation of an IE range for Connection Control. The IEs reserved for Connection Control shall be as defined in table 7.3.

Table 7.2: IEs for the ATM TRF Case

IE type (ATM)		IE length (note 1)	IE body
0x00	Capacity Request	2 bytes	As per clause 6.6.1.1 in [1]
0x01	M&C	2 bytes	As per clause 6.6.1.1 in [1]
0x02	Group_and_Logon_ID	3 bytes	As per clause 6.6.1.1 in [1] (see note 2)
0x03	Message Header	4 bytes	As described in clause 7.2.2
0x04	Cause	2 bytes	As described in clause 7.2.2
0x05	Channel_ ID	1 byte	As described in clause 7.2.2
0x06	Source Address	6 bytes	As described in clause 7.2.2
0x07	Destination Address	6 bytes	As described in clause 7.2.2
0x08	Forward Stream Identifier	3 bytes	As described in clause 7.2.2
0x09	Return Stream Identifier	3 bytes	As described in clause 7.2.2
0x0A	(Connection) Type	1 byte	As described in clause 7.2.2
0x0B	Forward Profile	3 bytes	As described in clause 7.2.2
0x0C	Return Profile	3 bytes	As described in clause 7.2.2
0x0D	Security Sign-on Response	8 bytes	As per clause 9.4.9.2 in [1]
0x0E	Route_ID	2 bytes	As described in clause 7.2.2
0x0F - 0x1E	Reserved for Connection Control	As described in clause 7.2.2	
0x1F	Wait	As per clause 9.4.9.9 in [1]	
0x20 - 0x30	Reserved		
0x31	Main Key Exchange Response	As per clause 9.4.9.4 in [1]	
0x32	Reserved		
0x33	Quick Key Exchange Response	As per clause 9.4.9.6 in [1]	
0x34	Reserved		
0x35	Explicit Key Exchange Response	As per clause 9.4.9.8 in [1]	
0x36	ACM	2 bytes	as per clause 6.6.1.1 in [1]
0x37	Mobility_Control_Message	4 bytes	as per clause 6.6.1.1 in [1]
0x38	Continuous Carrier Control	As per clause 10.4.4 in [1]	
0x39 - 0x7F	Reserved		

NOTE 1: "IE length": length (in bytes) of the IE body.
NOTE 2: Group_and_Logon_ID: concatenation of the 1-byte Group_ID and the 2-byte Logon_ID fields, in this order.

Table 7.3: IEs for the ATM TRF Case Reserved for Connection Control

(C2P-related) IE type (ATM)		IE length (note 1)	IE body
0x0F	Connection_Inactivity_Timeout	2 bytes	As described in clause 7.2.2
0x10	Other_Group_and_Logon_ID (note 2)	3 bytes	As described in clause 7.2.2
0x11	Other_Channel_ID	1 byte	As described in clause 7.2.2
0x12	Other_Route_ID	2 bytes	As described in clause 7.2.2
0x13	IPv6_Source_Address	17 bytes	As described in clause 7.2.2
0x14	IPv6_Destination_Address	17 bytes	As described in clause 7.2.2
0x15	Max_Packet_Size	2 bytes	As described in clause 7.2.2
0x16	Traffic_Spec_and_Policy_Data	32 bytes	As described in clause 7.2.2
0x17	RC_Capacity_Parameters	12 bytes	As described in clause 7.2.2
0x18	RCST_Capability	6 bytes	As described in clause 7.2.2
0x19	MAC_Destination_Address	6 bytes	As described in clause 7.2.2
0x1A	Connection_Status_Stats	17 bytes	As described in clause 7.2.2
0x1B - 0x1E	Reserved		

NOTE 1: "IE length": length (in bytes) of the IE body.
NOTE 2: "Other_Group_and_Logon_ID" is the concatenation of the 1-byte Other Group_ID and the 2-byte Other Logon_ID fields, in this order.
NOTE 3: The IPv6 address shall be 17 bytes long: 16 byte for IPv6 and include a byte for the mask.

7.2.1.2 DULM with MPEG formatting

For RCSTs using the optional MPEG2-TS TRF bursts format, the PID defined for C2P/control messages shall be used in the header of C2P signalling bursts. This PID (either Return_CTRL_PID or Return_CTRL_MGNM_PID) is obtained by the RCST during the logon procedure, as detailed in clause 4.6.1.

The DULM message in MPEG format and the semantics of the related Information Elements are defined in table 7.4.

Table 7.4: DULM Message in MPEG Format

Message field	Description	Length
MPEG-2 Header		32 bits
Group_ID		8 bits
Logon_ID		16 bits
IE Type (1)	Type of information carried by IE	5 bits (7-3 MSB)
N/C	0= New IE 1= Continuation of IE	1 bit (bit 2)
F/C	0= IE finishes in this MPEG packet 1= IE continues in next MPEG packet	1 bit (bit 1)
L/C	0= IE is the last of this MPEG packet 1= Another IE follows in this MPEG packet	1 bit (bit 0)
IE Segment Length (1)	Length of the part of the IE included in this MPEG packet	8 bits
IE (1)	IE content	8 bits
IE (1)	IE content	
IE (1)	IE content	8 bits
IE Type (2)	Type of information carried by IE	5 bits (7-3 MSB)
N/C	0= New IE 1= Continuation of IE	1 bit (bit 2)
F/C	0= IE finishes in this MPEG packet 1= IE continues in next MPEG packet	1 bit (bit 1)
L/C	0= IE is the last of this MPEG packet 1= Another IE follows in this MPEG packet	1 bit (bit 0)
IE Segment Length (2)	Length of the part of the IE included in this MPEG packet	8 bits
IE (2)	IE content	

NOTE 1: When an IE spans over several MPEG packets, the IE header is duplicated on these MPEG packets with N/C=0, F/C=1, L/C=0 for first one, N/C=1, F/C=1, L/C=0 for the following ones, and N/C=1, F/C=0, L/C=x for the last one.

NOTE 2: Padding bytes set to all "0" are appended to the last IE (L/C=0) of a MPEG packet.

NOTE 3: The PUSI bit embedded in MPEG header of DULM C2P messages is ignored by the NCC.

NOTE 4: IE segment length: It indicates the length of the part of the IE included in this MPEG packet, in number of bytes, from the byte immediately following the "segment length" field.

The Information Elements shall be as defined in table 7.5, which is a version of table 19 in [1] with minor changes (in fair font) regarding the references in the IE body column and the addition of a new IE type (Extended IE Type for Connection Control), allowing the extension of the number of IE types for Connection Control. The additional IEs for Connection Control shall be as defined in table 7.6.

Table 7.5: IEs for the MPEG TRF Case

IE type (MPEG)		IE length	IE body
0×00	Capacity Request	2 bytes	As per clause 6.6.1.1 in [1]
0x01	M&C	2 bytes	As per clause 6.6.1.1 in [1]
0x02	Reserved		
0x03	Message Header	4 bytes	As described in clause 7.2.2
0x04	Cause	2 bytes	As described in clause 7.2.2
0x05	Channel_ID	1 byte	As described in clause 7.2.2
0x06	Source Address	6 bytes	As described in clause 7.2.2
0x07	Destination Address	6 bytes	As described in clause 7.2.2
0x08	Forward Stream Identifier	3 bytes	As described in clause 7.2.2
0x09	Return Stream Identifier	3 bytes	As described in clause 7.2.2
0x0A	(Connection) Type	1 byte	As described in clause 7.2.2
0x0B	Forward Profile	3 bytes	As described in clause 7.2.2
0×0C	Return Profile	3 bytes	As described in clause 7.2.2
0x0D	Security Sign-on Response	8 bytes	As per clause 9.4.9.2 in [1]
0x0E	Route_ID	2 bytes	As described in clause 7.2.2
0x0F-0x10	Reserved		
0x11	Main Key Exchange Response	As per clause 9.4.9.4 in [1]	
0x12	Reserved		
0x13	Quick Key Exchange Response	As per clause 9.4.9.6 in [1]	
0x14	Reserved		
0x15	Explicit Key Exchange Response	As per clause 9.4.9.8 in [1]	
0x16	ACM	2 bytes	as per clause 6.6.1.1 in [1]
0x17	Mobility_Control_Message	4 bytes	as per clause 6.6.1.1 in [1]
0x18	Continuous Carrier Control	As per clause 10.4.4 in [1]	
0x19 - 0x1D	Reserved		
0x1E	Extended IE Type for Connection Control	As described in clause 7.2.2	
0x1F	Wait	As per clause 9.4.9.9 in [1]	

The coding of the Extended IE Type for Connection Control IE (i.e. 0x1E) is denoted as IE type (1). The first byte of this IE determines the extension of the IE type of a C2P-related IE, denoted as IE type ext. The complete coding of a C2P-related IE type shall be calculated as:

IE type = IE type (1) + IE type ext = 0x1E + IE type ext

An example is illustrated below for the first extended IE.

IE type (MPEG)	IE length	IE body
Extended IE Type (1)	Extended IE Type for C2P	5 bits (7-3 MSB) (0x1E)
N/C	0= New IE 1= Continuation of IE	1 bit (bit 2)
F/C	0= IE finishes in this MPEG packet 1= IE continues in next MPEG packet	1 bit (bit 1)
L/C	0= IE is the last of this MPEG packet 1= Another IE follows in this MPEG packet	1 bit (bit 0)
IE Segment Length (1)	Length of the part of the IE included in this MPEG packet (including the IE Type extension)	8 bits
IE Type extension (1)	C2P-related IE type as defined in table 7.6	8 bits
C2P IE	C2P IE content	

This approach allows future evolutions of ETSI C2P TS (e.g. addition of new IE types), while preserving the reserved IEs in table 7.5 for future evolutions of DVB-RCS.

Table 7.6: IEs for the MPEG TRF Case Reserved for Connection Control

(C2P-related) IE type (MPEG)		IE length	IE body
0x1E+ext 0x02	Connection_Inactivity_Timeout	2 bytes	As described in clause 7.2.2
0x1E+ext 0x03	Other_Group_and_Logon_ID	3 bytes	As described in clause 7.2.2
0x1E+ext 0x04	Other_Channel_ID	1 bytes	As described in clause 7.2.2
0x1E+ext 0x05	Other_Route_ID	2 bytes	As described in clause 7.2.2
0x1E+ext 0x06	IPv6_Source_Address	17 bytes	As described in clause 7.2.2
0x1E+ext 0x07	IPv6_Destination_Address	17 bytes	As described in clause 7.2.2
0x1E+ext 0x08	Max_Packet_Size	2 bytes	As described in clause 7.2.2
0x1E+ext 0x09	Traffic_Spec_and_Policy_Data	32 bytes	As described in clause 7.2.2
0x1E+ext 0x0A	RC_Capacity_Parameters	12 bytes	As described in clause 7.2.2
0x1E+ext 0x0B	RCST_Capability	6 bytes	As described in clause 7.2.2
0x1E+ext 0x0C	MAC_Destination_Address	6 bytes	As described in clause 7.2.2
0x1E+ext 0x0D	Connection_Status_Stats	17 bytes	As described in clause 7.2.2
0x1E+ext 0x0E - 0x1E+ext 01F	Future extensions of C2P		
NOTE 1: "IE length": length (in bytes) of the IE body. NOTE 2: "Other_Group_and_Logon_ID" is the concatenation of the 1-byte Other Group_ID and the 2-byte Other Logon_ID fields, in this order.			

NOTE 5: IE type coding for the extended IE Types used for Connection Control is different in the ATM case and MPEG case, since the methods used for coding are different.

The description of all IE types, including those reserved for connection control, is provided in clause 7.2.2.

7.2.2 Connection Control IE Types Description

- The connection control IE types for ATM and MPEG profiles are identified in table 7.7. The table also includes a brief description (overview) of each IE and associated parameters, consistent with the definitions provided in clause 4.3, and a reference to the applicable MIB objects defined in clause 4.6.1. A detailed description of each IE is provided in the following clauses of this clause 7.2.2. From the point of view of DULM signalling, an IE can be mandatory, optional or not applicable (blank), depending on the specific C2P message under consideration (see clause 8 for a complete description of all C2P messages).

- The parameters associated with the IEs in table 7.7 are defined from the point of view of one end-point (or party) of a connection (e.g. an RCST, as illustrated in the table). Parameters for the other end-point of the connection are prefixed by "Other".

Table 7.7: C2P IE Types

C2P parameter	Description	Applicable MIB objects
Message Header	Includes the C2P message type and length, the addressing type and the identifier of the connection affected by the C2P signalling.	ActiveCnxRefId
Cause	Conveys the reason (cause) for rejecting a C2P request.	ActiveCnxCause
Channel_ID	MAC Channel_ID, as defined in the DVB-RCS standard. In the context of C2P the Channel_ID is used to identify a given MAC CoS within a given connectivity channel.	ActiveCnxChannelId
Source Address	Defines the address of the source, consistent with the addressing type in clause 7.2.2.1	ActiveCnxIPv4SrcAddrMask1 ActiveCnxIPv4SrcAddrMask2 ActiveCnxIPv4SrcAddrMask3 ActiveCnxIPv4SrcAddrMask4 ActiveCnxIPv4SrcAddrMask5 ActiveCnxIPv4SrcAddrMask6 ActiveCnxIPv4SrcAddrMask7 ActiveCnxIPv4SrcAddrMask8 ActiveCnxMACSrcAddress
Destination Address	Defines the address of the destination, consistent with the addressing type in clause 7.2.2.1.	ActiveCnxMACDestAddr

C2P parameter	Description	Applicable MIB objects
Forward Stream Identifier	Identifies a single receive flow or an aggregation of flows of layer 2 packets pertaining to a connection, seen from the RCST point of view. It is either a {VPI, VCI} pair or a PID, depending on the ATM or MPEG2-TS format of the receive flow.	**ActiveCnxForwardStreamId**
Return Stream Identifier	Identifies a single transmit flow or aggregation of flows of layer 2 packets pertaining to a connection, seen from the RCST point of view. It is either a {VPI, VCI} pair or a PID, depending on the ATM or MPEG2-TS format of the transmit flow.	ActiveCnxReturnStreamId
(Connection) Type	Defines the connection type in terms of direction, casting and party that initiated the connection.	ActiveCnxType
Forward Profile	Defines the Class of Service of the connection (C2P CoS) and the connection's overall bandwidth resources for the receive stream, seen from the RCST point of view (note 1).	ActiveCnxReqForwardSDR ActiveCnxReqForwardPDR ActiveCnxAdmForwardSDR ActiveCnxAdmForwardPDR
Return Profile	Defines the Class of Service of the connection (C2P CoS) and the connection's overall bandwidth resources for the transmit stream, seen from the RCST point of view (note 1).	ActiveCnxC2PCoS ActiveCnxReqReturnSDR ActiveCnxReqReturnPDR ActiveCnxAdmReturnSDR ActiveCnxAdmReturnPDR
Route_ID	Defines a destination downlink physical path associated with a connectivity channel, equivalent to a Channel_ID list.	ActiveCnxRouteId
Connection_Inactivity_Timeout	Defines the connection inactivity timeout. If no transmission and/or reception traffic is detected during this period, the connection should be released.	ActiveCnxInactivityTimeout
Other_Group_and_Logon_ID	Defines the Group_ID and Logon_ID of the other party (notes 2 and 3).	ActiveCnxOtherGroupLogonId
Other_Channel_ID	Specifies the MAC Channel_ID of the other party, defined according to the DVB-RCS standard and used in the context of C2P by the other party to identify a given MAC CoS within a given connectivity channel (notes 2 and 3).	ActiveCnxOtherChannelId
Other_Route_ID	Defines a destination downlink physical path associated with a connectivity channel, used by the other party as an equivalent to a Channel_ID list (notes 2 and 3).	ActiveCnxOtherRouteId
IPv6_Source_Address	Defines the IPv6 address and mask of the source, if IPv6 is supported.	ActiveCnxIPv6SrcAddrMask1 ActiveCnxIPv6SrcAddrMask2 ActiveCnxIPv6SrcAddrMask3 ActiveCnxIPv6SrcAddrMask4 ActiveCnxIPv6SrcAddrMask5 ActiveCnxIPv6SrcAddrMask6 ActiveCnxIPv6SrcAddrMask7 ActiveCnxIPv6SrcAddrMask8
IPv6_Destination_Address	Defines the IPv6 address and mask of the destination, if IPv6 is supported.	ActiveCnxIPv6DestAddrMask1 ActiveCnxIPv6DestAddrMask2 ActiveCnxIPv6DestAddrMask3 ActiveCnxIPv6DestAddrMask4 ActiveCnxIPv6DestAddrMask5 ActiveCnxIPv6DestAddrMask6 ActiveCnxIPv6DestAddrMask7 ActiveCnxIPv6DestAddrMask8
Max_Packet_Size	Defines the traffic maximum packet size, as a specific traffic parameter used for traffic conditioning.	ActiveCnxMaxPacketSize
Traffic_Spec_and_Policy_Data	Defines traffic-related parameters and policies for traffic processing/conditioning.	PktClassTraficSpecPolicyData
RC_Capacity_Parameters	Describes the capacity categories parameters (CRA, RBDCMax and VBDCMax) of the Request Class pertaining to the connection.	requestClassCRA requestClassRBDCMax requestClassVBDCMax

C2P parameter	Description	Applicable MIB objects
RCST_Capability	Describes the RCST/RSGW capabilities related to C2P operation, in particular those associated with IP layer and above functionalities (e.g. IP protocol, type of multicast). It also allows specifying the C2P version.	rcstC2pVersion rcstC2pTransportOption rcstIpProtocolVersion rcstIpMulticastOption (note 4)
MAC_Destination_Address	Defines the MAC address of the destination.	ActiveCnxMACDestAddr
Connection_Status_Statistics	Defines the status and statistics of a connection.	ActiveCnxStatus ActiveCnxOutOctets ActiveCnxOutPkts ActiveCnxInOctets ActiveCnxInPkts
NOTE 1: For bidirectional connections the return/forward bandwidth parameters (rates) at one end-point of the connection become forward/return bandwidth parameters (rates) at the other end-point of the connection. NOTE 2: Only applies to connections for which the other party is an RCST/RSGW. NOTE 3: This parameter may be needed for bidirectional connections defined for some particular transparent mesh scenarios. Not used in the C2P messages defined in the present document. NOTE 4: These objects will be introduced in a future version of the DVB-RCS MIB.		

7.2.2.1 Message Header IE

Message Header IE is described in table 7.8.

Table 7.8: Message Header IE

Message Header IE (0x03) fields	Size (bits)	Description/Encoding
Message Description	8 bits	Message Description provides information about: - Message type: type of the signalling message being transferred, as specified in table 7.9. - Addressing type: addressing scheme used by the signalling message that determines the interpretation of the source and destination address fields, as specified in table 7.9.
Message Length	8 bits	C2P message length in bytes, from the 1st IE type byte to the end of last IE of the C2P message
Connection Reference ID	16 bits	Local connection identifier, encoded as 16 bits uimsbf (unsigned integer msb first), with the most significant bit set as follows: - "0" when Connection Reference ID is set by RCST/RSGW. - "1" when Connection Reference ID is set by NCC (note).
NOTE: For RC modify messages, the Connection Reference ID shall be ignored or a dummy (invalid) value could be used to ignore error cases. The invalid value of the connection reference ID is 0x0000.		

Table 7.9 specifies the Message type and the Addressing type in the Message Header and their encoding.

Table 7.9: Message type and Addressing type IE

Message Description	Size (bits)/Encoding	Description/Code	
Message type	5 bits uimsbf	CnxEstReq	0x01
		CnxEstResp	0x02
		CnxRelReq	0x03
		CnxRelResp	0x04
		CnxModifyReq	0x05
		CnxModifyResp	0x06
		CnxModifyJoinReq (note 1)	0x07
		CnxModifyJoinResp (note 1)	0x08
		CnxModifyLeaveReq (note 1)	0x09
		CnxModifyLeaveResp (note 1)	0x0A
		RCModifyReq	0x0B
		RCModifyResp	0x0C
		RCSTCapReq	0x0D
		RCSTCapResp	0x0E
		CnxStatusStatsReq	0x0F
		CnxStatusStatsResp	0x10
		Reserved	0x11-0x1F
Addressing type (note 2)	3 bits uimsbf	New addressing specified (note 3)	0x00
		Source MAC address Destination IP address	0x01
		Source MAC address List of source IP masks	0x02
		Destination MAC address List of destination IP masks	0x03
		Source IP address Destination IP address	0x04
		No addresses specified	0x05
		Source MAC address Source IP mask	0x06
		Destination MAC address Destination IP mask	0x07

NOTE 1: C2P command for future version of C2P.

NOTE 2: If the Addressing type includes an IP address or IP mask, these can be defined in terms of either IPv4 or IPv6.

NOTE 3: The 0x00 code "New Addressing specified" is a new addressing type defined for IPv4 with ATM and for IPv6 with ATM or MPEG.

The "New Addressing specified" type for IPv4 with ATM can use any combinations of the following addresses or no address at all:

- Source Address IE (0x06) = IPv4 Source Address IE (or list of IPv4 Source Address IEs).
- Destination Address IE (0x07) = IPv4 Destination Address IE (or list of IPv4 Destination Address IEs).

The "New Addressing specified" type for IPv6 with ATM can use any combinations of the following addresses or no address at all:

- IPv6 Source Address IE (0x13) (or list of IPv6 Source Address IEs).
- IPv6 Destination Address IE (0x14) (or list of IPv6 Destination Address IEs).

The "New Addressing specified" type for **IPv6 with MPEG** can use any combinations of the following addresses or no address at all:

- IPv6 Source Address IE (0x1E+ext 0x06) (or list of IPv6 Source Address IEs).
- IPv6 Destination Address IE (0x1E+ext 0x07) (or list of IPv6 Destination Address IEs).
- MAC Destination Address IE (0x1E+ext 0x0C).

For IPv4 with ATM format, there is a direct mapping of the source and destination address IEs to the ActiveCnxIPv4SrcAddrMask1 and ActiveCnxIPv4DestAddrMask1 in the Active Connection Table, respectively.

For IPv6 with ATM or MPEG format, there is a direct mapping of the IPv6 source and destination address IEs to the ActiveCnxIPv6SrcAddrMask1 and ActiveCnxIPv6DestAddrMask1 in the Active Connection Table, respectively.

For IPv4 with MPEG format, the mapping is not direct and depends on the addressing type; for the addressing type 07, for example, the following mapping applies:

- Source address IE (0x06) : Destination MAC address : MIB: ActiveCnxMACDestAddr.
- Destination address IE (0x07) : Destination IP address : MIB: ActiveCnxIPv4DestAddrMask1.

7.2.2.2 Cause IE

Cause IE is described in table 7.10. The usage of the Cause IE is system dependent, but if used it shall follow the description given in this table, in which "M" refers to a Mandatory cause and "O" refers to an Optional cause.

Table 7.10: Cause IE

Cause IE (0x04)	Size (bits)	Cause/Code/Applicability			Description
Cause type	16 bits	Success	0x0000	M	Request completed without incidents.
		NCC refuses connection	0x0001	M	The NCC refuses to create the connection due to network provisioning policies.
		Other RCST/RSGW refuses connection	0x0002	M	The other RCST/RSGW did not accept the requested connection. This cause is sent to the initiated RCST/RSGW during the establishment of a connection.
		Unknown destination	0x0003	M	The specified destination is not registered in the NCC tables.
		No more PIDs available in the system	0x0004	M	The NCC has temporarily run out of PIDs, making impossible the establishment of new connections.
		QoS cannot be guaranteed	0x0005	M	The requested connection can not be treated with the desired CoS (note 1).
		Other RCST/RSGW capacity exceeded	0x0006	M	The available capacity of the other RCST/RSGW would be exceeded with the requested capacity. This cause is sent to the initiated RCST/RSGW during the establishment of a connection (note 2).
		No more Channel_IDs available	0X0007	M	The system has no more Channel_IDs available to support new connections with a C2P CoS different from that of existing connections.
		Other RCST/RSGW not synchronized	0x0008	M	The other RCST/RSGW is not yet synchronized with the NCC. This cause is sent to the initiated RCST/RSGW during the establishment of a connection.
		NCC closes connection	0x0009	M	Connection closed by the NCC.
		No answer	0x000A	M	An RCST/RSGW or the NCC did not answer to a C2P request message or the message was lost.
		Unexpected event	0x000B	M	The received message was not expected, e.g. an RC modify request is received when the NCC is in the "Setup In Progress" state.
		Not enough BW	0x000C	M	There is not enough BW at the RC/MAC CoS level to satisfy the request (note 3).
		BW excess	0x000D	M	The requested BW exceeds the maximum bandwidth the RCST/RSGW is allowed to request (note 4).
		RCST/RSGW capacity exceeded	0x000E	M	The capacity assigned to the RCST/RSGW would exceed the requested capacity and/or the capacity that can be supported by the RCST/RSGW.
		Wrong Parameter	0x000F	M	A parameter in a C2P message did not have a valid format or was not within a valid range.
		NCC refuses connection release	0x0010	M	NCC refuses the request for the release of an NCC-initiated connection.
		NCC unavailability	0x0011	O	The NCC is unable to process the messages received.
		Pending connection from the other side	0x0012	O	This cause is used whenever an RCST/RSGW is trying to establish a connection with another RCST/RSGW that has already been requested by the latter.
		Client has not enough rights for such a request	0x0013	O	The terminal has made a request that bypasses its security policy or its SLA. For instance, in a multicast scenario a terminal could have not enough rights to begin a multicast session, or certain terminals might not be allowed to begin a connection with a gateway.
		Connection already established	0x0014	M	The requested connection is already active.
		NCC busy	0x0015	O	NCC is busy, i.e. it is not able to respond to the C2P request within the expected time.
		Connection does not exist	0x0016	M	Invalid Connection Reference ID.
		NCC does not support the RCST/RSGW C2P related capabilities	0x0017	M	The NCC does not support the C2P version, the IP protocol version or the IP multicast option supported by the RCST/RSGW (note 5).

Cause IE (0x04)	Size (bits)	Cause/Code/Applicability			Description
		RCST closes connection	0x0018	M	The RCST decides to initiate the release of the connection.
		Other	0x0019	O	The cause for not successfully completing a request is not among those identified in this clause.
		User defined causes	0x001A-0x001F	O	
		Reserved	0x0020-0xFFFF	O	

NOTE 1: When a particular QoS parameter cannot be committed by the NCC (e.g. the CRA corresponding to the requested SDR/PDR), the Cause "QoS cannot be guaranteed" is used in the response message, but it should not be interpreted as a reject, provided that the response message includes the QoS parameters that can be committed by the NCC (e.g. the admitted rate parameters PDR/SDR and the capacity categories parameters of the RC pertaining to the connection). The values of the admitted parameters can be zero. It is up to the initiating RCST/RSGW to decide if the admitted parameters are acceptable or not. If not, the initiating RCST/RSGW shall issue a CnxRelReq message.

NOTE 2: The "RCST/RSGW capacity exceeded" cause should not be interpreted as a reject, provided that the response message includes the capacity accepted by the RCST/RSGW and this is considered acceptable to the initiating RCST/RSGW or NCC (unless they apply "all-or-nothing" policies). If no valid values of capacity are provided, this cause should be interpreted as a reject.

NOTE 3: "Not enough BW" cause can be used with a connection establishment/modify request, when there is not enough bandwidth for the RC to accommodate a new connection, or with an RC modify request, e.g. when there is not enough bandwidth in the corresponding MAC CoS partition of return link resources at NCC to accommodate the increase of the guaranteed capacity requested for the RC.

NOTE 4: "BW excess" cause is given when an RCST/RSGW is requesting capacity that exceeds the maximum service capacity values (e.g. the maximum CRA assignment to the terminal for all STAR communications, or the maximum CRA assignment to the terminal for all MESH communications).

NOTE 5: If the C2P version is not the same and if the NCC has a more recent C2P version, the RCST/RSGW continues with the C2P state machine, otherwise the RCST/RSGW stops the C2P state machine. If the IP protocol is not the same, the IPv4 protocol shall be used as the default. If the IP multicast options in the RCST/RSGW are not the same as those in the NCC, the RCST/RSGW can not support Multicast, but the C2P state machine can be used for unicast.

7.2.2.3 Channel_ID IE

Channel_ID IE is described in table 7.11.

Table 7.11: Channel_ID IE

Channel IE (0x05)	Size (bits)	Description
Channel_ID	1 byte	Transmission channel identification, as defined in DVB-RCS [1]; 8 bits if "dynamic connectivity" is supported, otherwise 4 bits.

Channel_ID 0x00 shall be used as default value for the default signalling connection(s) established during terminal logon.

7.2.2.4 Source/Destination Address IE

Source Address IE and Destination Address IE are described in tables 7.12 and 7.13, respectively. In the case of Addressing types including IP addresses or IP subnet masks, IPv4 is assumed. For IPv6-based addressing please refer to IPv6_Source_Address IE and IPv6_Destination_Address IE in clause 7.2.2.13.

Table 7.12: Source Address IE

Source Address IE (0x06)	Size (bits)	Description/Encoding
Source Address Field	48 bits	For IPv4 with MPEG, the Source Address Field refers to the Source IPv4 address(es) and mask(s), or MAC addresses, used by RCSTs and NCC according to the addressing types defined in table 7.9. For IPv4 with ATM, the Source Address Field refers to the IPv4 source address(es) and mask(s). The field is uimsbf (unsigned integer msb first) encoded

Table 7.13: Destination Address IE

Destination Address IE (0x07)	Size (bits)	Description/Encoding
Destination Address Field	48 bits	For IPv4 and MPEG, the Destination Address Field refers to the Destination IPv4 address(es) and mask(s), or MAC addresses, used by RCSTs and NCC according to the addressing types defined in table 7.9. For IPv4 with ATM, the Destination Address Field refers to the IPv4 destination address(es) and mask(s). The field is uimsbf (unsigned integer msb first) encoded.

The IP masks are represented in Classless Inter Domain Routing (CIDR) notation (RFC 1518 [i.15] and RFC 4632 [i.16]). The address coding uses the 5 byte CIDR format aa.bb.cc.dd/ee (4 bytes for IPv4 address + 1 byte for the shortened mask value). The mask value (or length) defines the number of left-most contiguous mask bits that are set to one. In the case of IP addresses (4 bytes) or subnets (5 bytes) the most significant bits are unused (set to zero).

7.2.2.5 Forward/Return Stream Identifier IE

7.2.2.5.1 Applicability

IMPORTANT NOTE:

The return stream is understood as transmit stream, and the forward stream is understood as receive stream.

A stream is qualified as return/transmit stream or forward/receive stream with regard to one of the end-points (reference points) of the connection. An end-point point of a connection can be an RCST, an NCC/GW or an RSGW. A return stream at one end-point of a bidirectional connection becomes forward stream at the other end-point of the connection, and vice-versa.

The identifier of a return stream/forward stream can be either a {VPI, VCI} pair or a single PID value, depending on the ATM or MPEG2-TS profile of the transmit/receive flow.

If the end-point is an RCST:

- the return stream (from RCST) can be based on either the ATM profile or the MPEG2-TS profile, regardless of the other end-point of the connection (RCST for mesh connections, NCC/GW or RSGW for star connections);
- the forward stream (to RCST) can also be based on either the ATM profile or the MPEG2-TS profile, but only for mesh connections; for star connections the forward stream is always in MPEG2-TS format.

If the end-point is an NCC/GW:

- the return stream (from NCC/GW) is based on the MPEG2-TS format;
- the forward stream (to NCC/GW) can be based on either the ATM profile or the MPEG2-TS profile.

If the end-point is an RSGW:

- the return stream (from RSGW) can be based on either the ATM profile or the MPEG2-TS profile;
- the forward stream (to RSGW) can be based on either the ATM profile or the MPEG2-TS profile.

7.2.2.5.2 MPEG2-TS stream

Forward Stream Identifier IE and Return Stream Identifier IE for the MPEG2-TS profile are described in table 7.14 and table 7.15, respectively. The applicability is as per clause 7.2.2.5.1.

Table 7.14: Forward Stream Identifier IE for the MPEG2-TS Profile

Forward Stream Identifier IE (0x08)	Size (bits)	Description
Reserved	11 bits	PID extension. Default value all '0'
Forward PID	13 bits	Field is uimsbf (unsigned integer msb first) encoded. PID format complies with ITU-T Recommendation H.222.0 [3].

Table 7.15: Return Stream Identifier IE for the MPEG2-TS Profile

Return Stream Identifier IE (0x09)	Size (bits)	Description
Reserved	11 bits	PID extension. Default value all '0'
Return PID	13 bits	Field is uimsbf (unsigned integer msb first) encoded. PID format complies with ITU-T Recommendation H.222.0 [3].

The 11 reserved bits are added in order to make the size of the stream identifier the same for both MPEG2-TS profile and ATM profile (clause 7.2.2.5.3).

7.2.2.5.3 ATM stream

Forward Stream Identifier IE and Return Stream Identifier IE for the ATM profile are described in table 7.16 and table 7.17, respectively. The applicability is as per clause 7.2.2.5.1.

Table 7.16: Forward Stream Identifier IE for the ATM Profile

Forward Stream Identifier IE (0x08)	Size (bits)	Description
VPI/VCI	24 bits	VPI and VCI formats are in accordance with ITU-T Recommendation I.361 [4]. VPI uses 8 bits and VCI uses 16 bits.

Table 7.17: Return Stream Identifier IE for the ATM Profile

Return Stream Identifier IE (0x09)	Size (bits)	Description
Return VPI/VCI	24 bits	VPI and VCI formats are in accordance with ITU-T Recommendation I.361 [4]. VPI uses 8 bits and VCI uses 16 bits.

7.2.2.6 Connection Type IE

Connection Type IE is described in table 7.18.

Table 7.18: Connection Type IE

Connection Type IE (0x0A)	Size (bits)	Description/Code	
Connection Type	8 bits	point-to-point bidirectional RCST/RSGW-Initiated	0x01
		point-to-point bidirectional NCC-Initiated	0x02
		point-to-point unidirectional RCST/RSGW-Initiated	0x03
		point-to-point unidirectional NCC-Initiated	0x04
		point-to-multipoint unidirectional RSGW-Initiated (regenerative "star/mesh" multicast)	0x05
		point-to-multipoint unidirectional RCST-Initiated ("mesh/star" multicast)	0x06
		point-to-multipoint unidirectional NCC-Initiated (transparent "star/mesh" multicast, regenerative star/mesh multicast)	0x07
		multipoint-to-point (for future use)	0x08
		multipoint-to-multipoint (for future use)	0x09
		Reserved	0x0A - 0xFF

The connection types in table 7.18 apply to traffic connections. Signalling connections are typically established by default at logon time (clause 4.6.1). Additional signalling connections can be established after the RCST logon, based on C2P messages; such signalling connections are not included in version of the C2P Specifications defined in the present document.

The point-to-point connection types apply to both star connections and mesh connections, regardless of the type of payload (transparent or regenerative).

7.2.2.7 Forward/Return Profile IE

IMPORTANT NOTE:
The return profile is understood as transmit profile, and the forward profile is understood as receive profile.

Forward Profile IE and Return Profile IE are described in table 7.19 and table 7.20, respectively. They include service related parameters for the connection, i.e. the Class of Service (C2P CoS) and the return/transmit and forward/receive rate parameters.

The transmission and reception are based on either MF-TDMA or TDM primary access schemes, depending on the network reference scenario and connection's end-point under consideration.

Table 7.19: Forward Profile IE

Forward Profile IE (0x0B)	Size (bits)	Description/Encoding
C2P CoS	8 bits	This field defines the class of service of the connection (C2P CoS)
Forward Peak Data Rate	8 bits	Receive Peak Data Rate (Forward PDR) of the connection, coded in one byte as follows: - 1 bit (MSB) defines the Scaling Factor: value '1' represents a Scaling Factor of 16, values '0' represents a Scaling Factor of 1 - 7 bits representing a multiple M of 4 kbps (uimsbf encoded) - the resulting rate is given by: PDR = (Scaling Factor) x (M) x 4 kbps
Forward Sustainable Data Rate	8 bits	Receive Sustainable Data Rate (Forward SDR) of the connection, coded in one byte as follows: - 1 bit (MSB) defines the Scaling Factor: value '1' represents a Scaling Factor of 16, values '0' represents a Scaling Factor of 1 - 7 bits representing a multiple M of 4 kbps (uimsbf encoded) - the resulting rate is given by: SDR = (Scaling Factor) x (M) x 4 kbps

Table 7.20: Return Profile IE

Return Profile IE (0x0C)	Size (bits)	Description/Encoding
C2P CoS	8 bits	This field defines the class of service of the connection (C2P CoS)
Return Peak Data Rate	8 bits	Transmit Peak Data Rate (Return PDR) of the connection, coded in one byte as follows: - 1 bit (MSB) defines the Scaling Factor: value '1' represents a Scaling Factor of 16, values '0' represents a Scaling Factor of 1 - 7 bits representing a multiple M of 4 kbps: uimsbf (unsigned integer msb first) encoded - the resulting rate is given by: PDR = (Scaling Factor) x (M) x 4 kbps
Return Sustainable Data Rate	8 bits	Transmit Sustainable Data Rate (Return SDR) of the connection, coded in one byte as follows: - 1 bit (MSB) defines the Scaling Factor: value '1' represents a Scaling Factor of 16, values '0' represents a Scaling Factor of 1 - 7 bits representing a multiple M of 4 kbps: uimsbf (unsigned integer msb first) encoded - the resulting rate is given by: SDR = (Scaling Factor) x (M) x 4 kbps

In the case of connections originating at an RCST or RSGW (i.e. connections for which the transmission is based on the MF-TDMA access schemes), the class of service of a connection, as captured by the C2P CoS parameter, is mapped to one of the classes of service supported at MAC layer (i.e. MAC CoSs), also referred to as Request Classes (RCs). As an example, SatLabs has defined three RCs: Real Time (RT), Critical Data (CD) and Best Effort (BE).

Each RC/MAC CoS is characterized by a set of RC_Capacity_Parameters (clause 7.2.2.9), that shall reflect the rate parameters (Return PDR/SDR) in the Return Profile of all connections mapped to that RC/MAC CoS. The conversion of rate parameters to capacity parameters is class-specific and is performed at NCC. Guideline for conversion for each class can be found in [i.5].

7.2.2.8 Route_ID IE

Route_ID IE is described in table 7.21.

Table 7.21: Route_ID IE

Route_ID IE (0x0E)	Size (bits)	Description/Comments
Route_ID	16 bits	Route_ID format and usage are defined in DVB-RCS standard [1]. In the context of C2P, the Route_ID is used in RCST on the transmit side, for QoS optimization across a list of Channel_IDs associated with a connectivity channel (destination downlink).

7.2.2.9 RC_Capacity_Parameters IE

RC_Capacity_Parameters IE is described in table 7.22. They include limit (maximum) values per RC for the Constant Rate Assignment (CRA), Rate Based Dynamic Capacity (RBDC) assignment and Volume Based Dynamic Capacity (VBDC) assignment.

All RC capacity parameters in table 7.22 are transmission parameters with regard to an RCST/RSGW, as end-point of a connection.

NOTE: The transmission from the NCC/GW, as end-point of a connection, is not based on the MF-TDMA access scheme; therefore the Request Classes are not applicable to NCC/GW.

Table 7.22: RC_Capacity_Parameters IE

RC_Capacity_Parameters IE (0x17) ATM (0x1E+ext0x0A) MPEG	Size (bits)	Value/Comments
Return RC_CRA	32	Define the Continuous Rate Assignment (CRA) level for the request class, in bit per second (bps)
Return RC_RBDCmax	32	Maximum Rate-Based Dynamic Capacity (RBDC) that can be requested for the request class, in number of 2 Kbits/s
Return RC_VBDCmax	32	Maximum Volume-Based Dynamic Capacity (VBDC) that can be allocated to the request class, in payload units (one ATM cell or one MPEG cell) per superframe

The RC_Capacity_Parameters IE is used in the CnxEstReq, CnxEstResp, CnxModifyReq, CnxModifyResp, RCModifyReq and RCModifyResp messages.

7.2.2.10 Other_Group_and_Logon_ID IE

Other_Group_Logon_ID IE is described in table 7.23.

Table 7.23: Other_Group_and_Logon_ID IE

Other_Group_and_Logon_ID IE (0x10) ATM (0x1E+ext0x03) MPEG	Size (bits)	Description/Comments
Other_Group_ID	8 bits	Group_ID, as defined in DVB-RCS [1]. In the context of C2P, the Other_Group_ID is applicable to the other end RCST (or RSGW).
Other_Logon_ID	16 bits	Logon_ID, as defined in DVB-RCS [1]. In the context of C2P, the Other_Logon_ID is applicable to the other end RCST (or RSGW).

7.2.2.11 Other_Channel_ID IE

Other_Channel_ID IE is described in table 7.24.

Table 7.24: Other_Channel_ID IE

Other_Channel_ID IE (0x11) ATM (0x1E+ext0x04) MPEG	Size (bits)	Value/Comments
Other_Channel_ID	8 bits	Channel_ID format and usage as defined in DVB-RCS [1]. In the context of C2P, the Other_Channel_ID is used by the other end RCST (or RSGW) to identify a MAC Class of Service.

7.2.2.12 Other_Route_ID IE

Other_Route_ID IE is described in table 7.25.

Table 7.25: Other_Route_ID IE

Other_Route_ID IE (0x12) ATM (0x1E+ext0x05) MPEG	Size (bits)	Value/Comments
Other_Route_ID	16 bits	Route_ID format and usage as defined in DVB-RCS [1]. In the context of C2P the Other_Route_ID is used by the other end RCST (or RSGW) on the transmit side, for QoS optimization across a list of Other_Channel_IDs associated with a connectivity channel (destination downlink).

7.2.2.13 IPv6_Source_Address/IPv6_Destination_Address IE

IPv6_Source_Address_IE and IPv6_Destination-Address IE are described in table 7.26 and table 7.27, respectively.

Table 7.26: IPv6_Source_Address IE

IPv6_Source_Address IE (0x13) ATM (0x1E+ext0x06) MPEG	Size (bits)	Description/Encoding
IPv6_Source_Address	136 bits	Source IPv6 address(es) and mask(s). The field is uimsbf encoded (unsigned integer msb first).

Table 7.27: IPv6_Destination_Address IE

IPv6_Destination_Address IE (0x14) ATM (0x1E+ext0x07) MPEG	Size (bits)	Description/Encoding
IPv6_Destination_Address	136 bits	Destination IPv6 address(es) mask(s). The field is uimsbf encoded (unsigned integer msb first).

The IP masks are represented in Classless Inter Domain Routing (CIDR) notation (RFC 1518 [i.15] and RFC 4632 [i.16]). The address coding uses the 17-byte CIDR format (16 bytes for IPv6 address + 1 byte for the shortened mask value). The mask value (or length) defines the number of left-most contiguous mask bits that are set to one.

7.2.2.14 Connection_Inactivity_Timeout IE

Connection_Inactivity_Timeout IE is described in table 7.28.

Table 7.28: Connection_Inactivity_Timeout IE

Connection_Inactivity_Timeout IE (0x0F) ATM (0x1E+ext0x02) MPEG	Size (bits)	Description
Connection_Inactivity_Timeout	16 bits	Connection_Inactivity_Timeout (in seconds) is used to release a connection, if no transmission and/or reception of traffic is detected during the timeout period. If set to 0, there is no activity timeout, the connection always stays active. The unit is in number of seconds.

7.2.2.15 Maximum_Packet_Size IE

Maximum_Packet_Size IE is described in table 7.29.

Table 7.29: Maximum_Packet_Size IE

Maximum packet size IE (0x15) ATM (0x1E+ext0x08) MPEG	Size (bits)	Description/Comments
Maximum_Packet_Size	16 bits	Defines the maximum size (in bytes) of an incoming packet; it may be used as traffic parameter for traffic conditioning. (Example:1 500 bytes for IP/Ethernet)

7.2.2.16 Traffic_Spec_and_Policy_Data IE

Traffic_Spec_and_Policy_Data IE is described in table 7.30.

Table 7.30: Traffic_Specification_and_Policy_Data IE

Traffic_Spec_and_Policy_ Data IE (0x16) ATM (0x1E+ext0x09) MPEG	Size (bits)	Description/Comments
Traffic data	128 bits (note)	Defines traffic/service related parameters for traffic filtering/processing/conditioning
Policy data	128 bits (note)	Defines policies/rules for traffic filtering/processing/conditioning
NOTE: The size is a rough estimate. The exact size will depend on the concrete traffic specification and policy formulation (implementation specific).		

7.2.2.17 RCST_Capability IE

The RCST_Capability IE allows an RCST to signal its capability related to C2P operation, in particular those associated with IP layer and above functionalities. It also allows specifying the C2P version and includes ample reserve bits (up to one ATM payload worth of data) for future needs.

Table 7.31 defines the parameters included in the RCST_Capability IE.

Table 7.31: RCST_Capability IE

RCST_Capability (0x18) ATM (0x1E+ext0x0B) MPEG	Size (bits)/Encoding	Description/Code	
C2P version	3 bits uimsbf	C2P not supported	000
		Version 1 of C2P	001
		Reserved for future versions	010-111
C2P message transport	2 bits uimsbf	DULM/TIMu	00
		Reserved	01-11
Reserved	3 bits uimsbf		
IP protocol version	3 bits uimsbf	CnxEstReq IPv6 traffic transport and IPv4 M&C/internal addressing	000
		Any mix of IPv6 and IPv4 (note 1)	001
		Either IPv6 or IPv4	010
		Only IPv6	011
		Only IPv4	100
		Reserved	101-111
Reserved	5 bits uimsbf		
IP multicast option (note 2)	3 bits uimsbf	No multicast support	000
		Static multicast configuration (MMT)	001
		IGMPv2 dynamic multicast	010
		IGMPv3 dynamic multicast	011
		MLDv1 for IPv6 (derived from IGMPv2)	100
		MLDv2 for IPv6 (derived from IGMPv3)	101
		Reserved	110-111
Reserved	5 bits uimsbf		
Reserved for future features	24 bits uimsbf		

NOTE 1: In the case of IPv6 and IPv4 mix, the IPv6 IEs will be used for both protocol versions. Only the last five bytes will be used for IPv4 source/destination addresses, and all other bytes will be set to zero.

NOTE 2: IP Multicast option is system specific. The version of C2P specified in the present document only requires support for static multicast configuration, based on the Multicast Mapping Table (MMT). The other multicast options are for future version of the C2P.

The RCST_Capability IE is transmitted as a C2P message, even if its applicability goes beyond the C2P basic operational requirements. It can be seen as a general control plane tool for the configuration of features that are not necessarily DVB-RCS specific but are useful for the RCST operation in a DVB-RCS based network.

The message carrying the RCST_Capability IE is transmitted over a default signalling connection established at logon (clause 4.6.1).

7.2.2.18 MAC Destination Address IE

Table 7.32 defines the parameters included in the MAC_Destination_Address IE.

Table 7.32: MAC_Destination_Address IE

MAC_Destination_Address (0x19 for ATM, 0x1E+ext 0x0C for MPEG)	Size (bits)/Encoding	Description/Code
MAC destination address	48 bits uimsbf	Destination MAC address, used for MPEG format

7.2.2.19 Connection_Status_Stats IE

The Connection_Status_Stats IE allows an RCST to signal the status and statistics per active connection. This Connection_Status_Stats IE is included in the Connection_Status_Stats Request/Response C2P messages.

Table 7.33 defines the parameters included in the Connection_Status_Stats IE.

Table 7.33: Connection_Status_Stats IE

Connection_Status_Stats IE (0x1A for ATM, 0x1E+ext 0x0D for MPEG)	Size (bits)/Encoding	Description/Code	
Connection Reference	16 bits uimsbf	Connection reference	
Status	8 bits uimsbf	Status of the connection as defined in clause 4.6.3.3	
OutOctets	32 bits uimsbf	Number of octets sent into this active connection	(note)
OutPkts	32 bits uimsbf	Number of packets sent into this active connection	(note)
InOctets	32 bits uimsbf	Number of octets received from the active connection	(note)
InPkts	32 bits uimsbf	Number of packets received from the active connection	(note)
NOTE: If the terminal does not support statistics this parameter is set to '0'.			

The Connection_Status_Stats Request/Response message can contain a list of Connection_Status_Stats IEs.

7.3 NCC to RCST messages (TIMu)

7.3.1 TIMu format

The C2P messages sent from NCC to RCSTs/RSGWs use the Connection Control Descriptor (table 7.34), consistent with the definition in [1]. The descriptor includes in the message body the same IEs as those specified in clause 7.2.2 for DULM. Some of the IEs in the descriptor have already been defined in the DVB-RCS Guidelines document [i.5], while others are new, added for the purpose of C2P support. For backward compatibility, an extension bit is included (Extended_connection_control_descriptor_flag), to indicate the extension headers for the new IEs.

Table 7.34: Connection Control Descriptor

Syntax	No. of bits	
	Reserved	Information
Connection_control_descriptor (){		
Descriptor_tag		8
Descriptor_length		8
Message_header_IE_flag		1
Cause_IE_flag		1
Channel_ID_IE_flag		1
Source_address_IE_flag		1
Destination_address_IE_flag		1
Forward_stream_identifier_IE_flag		1
Return_stream_identifier_IE_flag		1
Connection_type_IE_flag		1
Forward_profile_IE_flag		1
Return_profile_IE_flag		1
Route_ID_IE_flag		1
RC_capacity_parameters_IE_flag		1
Other_Group_and_Logon_ID_IE_flag		1
Other_Channel_ID_flag		1
Other_Route_ID_flag		1
Extended_connection_control_descriptor_flag		1
If (Extended_connection_control_descriptor_flag == 1) {		
Connection_inactivity_timeout_IE_flag		1
IPv6_source_address_IE_flag		1
IPv6_destination_address_IE_flag		1
Maximum_packet_size_IE_flag		1
Traffic_Spec_and_Policy_data_IE_flag		1
RCST_capability_IE_flag		1
MAC_Destination_Address_flag		1

Syntax	No. of bits	
	Reserved	Information
Connection_Status_Stats_flag		1
Main_Key_Exchange_IE_Flag		1
Quick_Key_Exchange_IE_Flag		1
Explicit_Key_Exchange_IE_Flag		1
Reserved		5
}		
If (Message_header_IE_flag == 1) {		
Message Header IE		32
}		
If (Cause_IE_flag == 1) {		
Cause IE		16
}		
If (Route_IE_flag == 1) {		
Route_ID IE		16
}		
If (Channel_ID_IE_flag == 1) {		
Channel_ID IE		8
}		
If (Source_address_IE_flag == 1) {		
Source_address_loop_count		8
For (i=0;i<=Source_address_loop_count;i++) {		
Source Address IE		48
}		
If (Destination_address_IE_flag == 1) {		
Destination_address_loop_count		8
For (i=0;i<=Destination_address_loop_count;i++) {		
Destination Address IE		48
}		
If (Forward_stream_identifier_IE_flag == 1) {		
Forward Stream Identifier IE		24
}		
If (Return_stream_identifier_IE_flag == 1) {		
Return Stream Identifier IE		24
}		
If (Connection_Type_IE_flag == 1) {		
Connection Type IE		8
}		
If (Forward_profile_IE_flag == 1) {		
Forward Profile IE		24
}		
If (Return_profile_IE_flag == 1) {		
Return Profile IE		24
}		
If (Other_Group_and_Logon_ID_IE_flag == 1) {		
Other_Group_and_Logon_ID IE		24
}		
If (Other_Channel_ID_IE_flag == 1) {		
Other_Channel_ID IE		8
}		
If (Other_Route_ID_IE_flag == 1) {		
Other_Route_ID IE		16
}		
If (Extended_connection_control_descriptor_flag == 1) {		
If (Connection_inactivity_timeout _IE_Flag== 1) {		
Connection_Inactivity_Timeout IE		16
}		
If (IPv6_source_address_IE _flag ==1){		
Source_address_loop_count		
For (i=0;i<=Source_address_loop_count;i++) {		
IPv6_Source_Address IE		136
}		
If (IPv6_destination_address_IE _flag ==1){		

Syntax	No. of bits	
	Reserved	Information
Destination_address_loop_count		
For (i=0;i<=Destination_address_loop_count;i++) {		
IPv6_Destination_Address IE		136
}		
If (Maximum_packet_size_IE_flag ==1){		
Maximum_Packet_Size IE		16
}		
If (Traffic_spec_and_policy_data_IE_flag ==1){		
Traffic_Spec_Data IE		256
}		
If (RCST_capability_IE_flag = = 1) {		
RCST_Capability IE		96
}		
If (MAC_Destination_Address_IE_flag = = 1) {		
MAC_Destination_Address IE		48
}		
If (Connection_Status_Stats_IE_flag = = 1) {		
Connection_Status_Stats IE		136
}		
If (Main_Key_Exchange_IE_flag == 1){		
Main_Key_Exchange IE		48+Pns+3*Ppka
}		
If (Quick_Key_Exchange_IE_flag = = 1) {		
Quick_Key_Exchange IE		48+Pns

}
If (Explicit_Key_Exchange_IE_flag = = 1) {		

Explicit_Key_Exchange IE		56+Pns+Pea
}		
}		

- descriptor_tag: The descriptor tag is an 8 bit field which identifies the descriptor. Its value is given in the Tag value column of table 64 in [1];

- descriptor_length: The descriptor length is an 8 bit field specifying the number of bytes of the descriptor immediately following the descriptor_length field;

- message_body: This variable length field shall contain a C2P signalling message for passing to the target connection control entity. Its length shall not exceed 255 bytes and it is likely to limit it so that related clause fits into a single TS packet. As defined in previous clause, message_length corresponds to the full message body (starting from message_description).

NOTE: The values Pns, Ppka and Pea will depend on the security implementation following last version of DVB-RCS standard [1].

8 Connection Control Messages

The connection control procedures shall use the C2P messages defined in this clause.

The C2P messages shall be built using the Information Elements (IEs) defined in clause 7. The codes used for the IE types shall be as defined in the applicable tables in clause 7.2.2.

The C2P messages should be consistent with the C2P procedures defined in clause 6.

The template in table 8.1 should be used for the construction of all C2P messages. It includes all IE types relevant to the version of the C2P Specifications defined in the present document. The Message Header sub-fields are mandatory in all C2P commands. The applicability of other IEs will depend on the message type and the reference scenario.

The following conventions are used to indicate the applicability of an IE to a C2P message:

- Mandatory (M): required in all messages and for all scenarios or only under certain circumstances ("if" flag).
- Optional (O): optional, for additional information.
- Not Applicable (Blank): not required in that message.

The following rules shall apply to the construction of all C2P messages:

TRANSMITTING SIDE

When formatting C2P messages for transmission, the transmitting party **should** assemble the necessary IEs in the order and with the applicability conventions defined in this clause.

RECEIVING SIDE

When receiving a C2P message, the receiving party **shall** process the message as follows:

- The receiving party **shall** process all Mandatory and Optional IEs regardless of the order in which they appear in the received message.
- If the C2P message is missing one or more Mandatory IEs, the receiving party **shall** ignore the message.
- If the C2P message contains one or more Not-Applicable IEs, the receiving party **shall** process the rest of the message as if these IEs were not present.

Table 8.1 also includes some generic notes, applicable to all C2P messages. They will not be repeated for each C2P message, but will be considered when defining the specifics of each C2P message. Additional and/or more specific notes will be provided for each message, as needed.

Table 8.1: C2P message fields template

IE Fields		Source to Destination message (note 1)	Value/Comments	
Message Header	Message type	M	HEX value (1 Byte) (note 2)	5 most significant bits BINARY format
	Addressing type (note 3)			3 least significant bits BINARY format
	Length	M	C2P message length	
	Connection ref	M	Connection reference id selected by RCST/RSGW or NCC	
Cause		O	Reason to send the C2P message	
Channel_ID		O	Channel ID of RCST	
Source Address		O (if IPv4)	IPv4 source address(es) and mask(s), or MAC address, depending on the addressing type	
Destination Address		O (if IPv4)	IPv4 destination address(es) and mask(s)	
Forward Stream Identifier		O	Identifies the receive flow (VCC or PID)	
Return Stream Identifier		O	Identifies the transmit flow (VCC or PID)	
Connection Type		O	Type of the connection	
Forward Profile		O	Forward C2P CoS and receive bandwidth parameters (Forward SDR/PDR) (note 4)	
Return Profile		O	Return C2P CoS and transmit bandwidth parameters (Return SDR/PDR) (note 4)	
Route_ID		O	Define the destination downlink of the connection	
Connection_Inactivity_Timeout		O	Define connection inactivity timeout	
IPv6_Source_Address		O (if IPv6)	IPv6 source IPv6 address(es) and mask(s)	
IPv6_Destination_Address		O (if IPv6)	IPv6 destination IPv6 address(es) and mask(s)	
Maximum_Packet_Size		O	Maximum IP packet size	
Traffic_Spec_and_Policy_Data		O	Traffic-related parameters and policies	
RC_Capacity_Parameters		O	Define CRA, VBDCmax and RBDCmax	
RCST_Capability		O	RCST capabilities related to C2P	
MAC Destination Address		O	Destination MAC address of the peer RCST/RSGW or NCC	
Connection_Status_Stats		O	Status and statistics of the connection	

IE Fields	Source to Destination message (note 1)	Value/Comments
NOTE 1: A parameter is mandatory (M), optional (O) or missing (blank), depending on the type of message and on the reference scenario. NOTE 2: The combined message type and addressing type require at least two values: one for the IPv4 with MPEG format, corresponding to the original C2P design [i.11], and one for the format used for other supported combination of addresses (IPv4 with ATM and IPv6 with MPEG or ATM). More values may be specified for the IPv4 with MPEG format. NOTE 3: For an addressing type code different from 0x00, the specified Source Address and Destination Address fields shall be consistent with the addressing type, while the fields IPv6_Source_Address, IPv6_Destination_Address and MAC Destination Address shall be blank. For the addressing type 0x00, the source and destination address fields shall be consistent with the type of C2P message and the reference scenario in which they are used (see clause 7.2.2.1). NOTE 4: SDR/PDR can be requested parameters or admitted parameters, depending on the specific C2P procedure/C2P message. They can be set by the initiating RCST/RSGW or NCC. Forward and Return C2P CoS can be the same.		

The comments below are applicable to all C2P messages and will not be repeated for each message:

- Message type and Addressing type are expressed in binary representation (coding).
- The concatenation of Message type and Addressing type (the left part in the Value/Comments column header), which represents the Message Description, is expressed in hexadecimal representation (coding).At least two values will be specified for each C2P messages: one for IPv4 with MPEG format and another one for the other format used for other supported combination of addresses (IPv4 with ATM and IPv6 with MPEG or ATM).
- The values/codes associated with different IEs are also in hexadecimal representation.
- The source/destination and forward/return Information Elements are defined from the point of view of the network element/party (RCST/RSGW or NCC) sending the C2P message:
    - According to clause 7.2.2, the return shall be understood as transmit and the forward shall be understood as receive.
    - According to clause 4.4.2.2, *the* IP source address(es) and mask(s) are defined on the *Ethernet interface of the party sending the C2P message (e.g. an RCST), while the IP destination address(es) and mask(s) are defined on the Ethernet interface of the other party (e.g. the other RCST). In the case of bidirectional connections this comment applies to both parties involved in the connection; regardless of the connection's initiator; the* IP source address(es) and mask(s) at one party become *IP destination address(es) and mask(s) at the other party and vice-versa.*

The following clauses specify the C2P messages (requests and responses) for the establishment, modification and release of the following types of connections:

- Point-to-point (unicast) connections:
    - RCST/RGSW-initiated.
    - NCC-initiated.
- Point-to-multipoint (multicast) connections:
    - RCST/RGSW-initiated.
    - NCC-initiated.

Other clauses specify the RC modify commands, the RCST Capability commands and the Connection Status and Statistics commands.

All C2P messages specified in this clause shall be consistent with the connection control procedures defined in clause 6, which only apply to traffic connections. The establishment of signalling connections by using C2P messages, while possible, is specifically excluded from the present document, according to which the signalling connections are established by default at logon time (see clause 4.6.1).

8.1 Point-to-point connection establishment message

Point-to-point connections are unicast connections, initiated by either the RCST/RSGW or the NCC. They can be unidirectional or bidirectional and can be established:

- Between two RCSTs (mesh transparent or regenerative reference scenarios).
- Between two RSGWs (assimilated to mesh connections in mesh regenerative reference scenario, since the RSGWs can be seen as conventional terminals with extended capabilities).
- Between an RCST and the NCC/GW or RSGW (star transparent or regenerative reference scenarios).

For the terminology used for the parties involved in point-to-point connections under different network reference scenarios please refer to clause 4.4.2.2.

8.1.1 Connection Establishment Request (CnxEstReq)

8.1.1.1 RCST/RSGW initiated unicast connection

Depending on the network reference scenario:

- If the connection is initiated by an RCST (initiating party), the peer party will be another RCST (in mesh transparent or regenerative reference scenarios), the RSGW (in star regenerative scenario) or the NCC/GW (in star transparent reference scenario).
- If the connection is initiated by an RSGW (initiating party), the peer party will be an RCST (star regenerative connections) or another RSGW (particular case of mesh regenerative connections).

If the peer party is different from the NCC/GW, the connection establishment request involves two messages: one from the initiating RCST/RSGW to NCC, and another one from the NCC to the peer party.

In the case of unidirectional connections, the initiating RCST/RSGW is considered the unicast source.

8.1.1.1.1 From initiating RCST/RSGW to NCC (DULM)

Table 8.2: Unicast CnxEstReq Fields - RCST/RSGW initiated: initiating RCST/RSGW to NCC

IE Fields		Initiating RCST/RSGW to NCC	Value/Comments	
Message Header	Message type	M	'0x0C'(IPv4 with MPEG) 0x01 (otherwise)	"00001"
	Addressing type			"100" (IPv4 with MPEG) "000" (otherwise)
	Length	M	C2P message length	
	Connection ref	M	Connection reference id selected by the initiating RCST/RSGW	
Cause				
Channel_ID				
Source Address		M (if IPv4)	IPv4 source address(es) and mask(s)	
Destination Address		M (if IPv4)	IPv4 destination IPv4 address(es) and mask(s) of the peer party	
Forward Stream Identifier				
Return Stream Identifier				
Connection Type		M	'0x01' point-to-point bidirectional RCST/RSGW-initiated, or '0x03' point-to-point unidirectional RCST/RSGW-initiated	
Forward Profile		O (note)	Forward C2P CoS and requested receive bandwidth parameters (Forward SDR/PDR)	
Return Profile		M	Return C2P CoS and requested transmit bandwidth parameters (Return SDR/PDR)	
Route_ID				
Connection_Inactivity_Timeout		O	Connection inactivity timeout	
IPv6_Source_Address		M (if IPv6)	IPv6 source address(es) and mask(s)	

IE Fields	Initiating RCST/RSGW to NCC	Value/Comments
IPv6_Destination_Address	M (if IPv6)	IPv6 destination address(es) and mask(s) of the peer party
Maximum_Packet_Size	O	
Traffic_Spec_and_Policy_Data	O	
RC_Capacity_Parameters		
RCST_Capability		
MAC Destination Address		
Connection_Status_Stats		
NOTE: Not needed in the case of unidirectional connections.		

8.1.1.1.2 From NCC to peer RCST/RSGW (TIMu)

Table 8.3: Unicast CnxEstReq Fields - RCST/RSGW initiated: NCC to peer RCST/RSGW

IE Fields		NCC to peer RCST/RSGW	Value/Comments	
Message Header	Message type	M	'0x0E' or '0x0A' or '0x0D' (IPv4 with MPEG) 0x01 (otherwise)	"00001"
	Addressing type			"110" or "010" or "101" (IPv4 with MPEG) "000" (otherwise)
	Length	M	C2P message length	
	Connection ref	M	Connection reference id selected by the NCC	
Cause				
Channel_ID		O (note 2)	New or an already existing Channel_ID, to be used by the peer RCST/RSGW.	
Source Address		O (if IPv4) (note 2)	In the case of bidirectional connections: MAC address of the initiating RCST/RSGW for the IPv4 with MPEG format; or IPv4 source address(es) and mask(s) for the IPv4 with ATM format	
Destination Address		O (if IPv4) (note 2)	In the case of bidirectional connections: IPv4 destination address(es) and mask(s) of the initiating RCST/RSGW.	
Forward Stream Identifier		M	Identifies the receive flow (VCC or PID) (note 1)	
Return Stream Identifier		O (note 2)	Identifies the transmit flow (VCC or PID) (note 1)	
Connection Type		M	'0x01' point-to-point bidirectional RCST/RSGW-initiated; or '0x03' point-to-point unidirectional RCST/RSGW-initiated	
Forward Profile		M	Forward C2P CoS and admitted receive bandwidth parameters (Forward SDR/PDR), filled by the NCC (notes 1 and 3)	
Return Profile		O (note 2)	Return C2P CoS and admitted transmit bandwidth parameters (Return SDR/PDR), filled by the NCC (notes 1 and 3)	
Route_ID		O (note 2)		
Connection_Inactivity_Timeout		O	Connection inactivity timeout	
IPv6_Source_Address		O (if IPv6)	IPv6 source address(es) and mask(s)	
IPv6_Destination_Address		O (if IPv6) (note 2)	IPv6 destination address(es) and mask(s) of the initiating RCST/RSGW	
Maximum_Packet_Size		O		
Traffic_Spec_and_Policy_Data		O		
RC_Capacity_Parameters		O (note 2)	Define CRA, VBDCmax and RBDCmax	
RCST_Capability				
MAC Destination Address		O (If IPV6 with MPEG) (note 2)	MAC address of the initiating RCST/RSGW	
Connection_Status_Stats				
NOTE 1: In the case of bidirectional connections the Forward and Return Profile values are swapped by NCC to match the peer party point of view. NOTE 2: Not needed in the case of unidirectional connections. NOTE 3: Forward and Return profiles can be different from the ones sent by the initiating RCST/RSGW (as reflected by the "admitted" bandwidth parameters), if they cannot be supported by the NCC (for bandwidth or policy reasons).				

The message described in this clause is not needed in the case of star transparent connections from RCST to NCC/GW, since the peer party is the NCC/GW.

8.1.1.2 NCC initiated unicast connection (TIMu)

Depending on the network reference scenario, the NCC-initiated unicast connection can be:

- Between the NCC/GW and an RCST (in star transparent reference scenario)
  - The connection establishment request message will be sent by the NCC/GW, as initiating party, to the RCST.
  - In the case of unidirectional connections, the unicast source can be either the initiating NCC/GW or the RCST.
- Between two RCSTs (in mesh transparent or regenerative reference scenarios), between an RCST and an RSGW (in star regenerative reference scenarios) or between two RSGWs (a particular case of mesh regenerative reference scenario).
  - Separate connection establishment request messages will be sent by the initiating NCC to the two parties, referred to as "RCST/RSGW" and "the other end RCST/RSGW" (clause 4.4.2.2). The messages will have the same structure but the relevant fields will be filled with data pertinent to the party receiving the message.
  - In the case of unidirectional connections, an RCST/RSGW can be either unicast source or unicast destination.

Table 8.4 includes the parameters in the C2P message sent by NCC to one party of the connection (RCST/RSGW), pertinent to that party. In the case of connections between two RCSTs, between two RSGWs or between an RCST and an RSGW, some parameters in the messages (e.g. the destination addresses) refer to the other party, i.e. the RCST/RSGW at the other end of the connection.

Table 8.4: Unicast CnxEstReq Fields - NCC initiated: NCC to RCST/RSGW

IE Fields		Initiating NCC to RCST/RSGW	Comments	
Message Header	Message type	M	'0x0F' or '0x0B' or '0x0D' (IPv4 with MPEG) 0x01 (otherwise)	"00001"
	Addressing type			"111" or "011" or "101" (IPv4 with MPEG) "000" (otherwise)
	Length	M	C2P message length	
	Connection ref	M	NCC selected connection reference id	
Cause				
Channel_ID		M	New or existing Channel_ID to be used by the peer RCST/RSGW	
Source Address		O (if IPv4) (note 2)	In the case of bidirectional connections between NCC/GW and RCST MAC address of the NCC for the IPv4 with MPEG format; or IPv4 source address(es) and mask(s) for the IPv4 with ATM format. In the case of bidirectional connections between two RCSTs, two RSGWs or an RCST and an RSGW: MAC address of the other end RCST/RSGW for the IPv4 with MPEG format; or IPv4 source address(es) and mask(s) for the IPv4 with ATM format.	

IE Fields	Initiating NCC to RCST/RSGW	Comments
Destination Address	O (if IPv4) (note 2)	In the case of bidirectional connections between NCC/GW and RCST: IPv4 destination address(es) and mask(s) of the NCC. In the case of bidirectional connections between two RCSTs, two RSGWs or an RCST and an RSGW: IPv4 destination address(es) and mask(s) of the other end RCST/RSGW
Forward Stream Identifier	O (note 1)	VPI/VCI or PID assigned to connection for the Forward path
Return Stream Identifier	O (note 1)	VPI/VCI or PID assigned to connection for the Return path
Connection Type	M	'0x02' point-to-point bidirectional NCC-Initiated or '0x04' point-to-point unidirectional NCC-Initiated
Forward Profile	O (note 1)	Forward C2P CoS and admitted receive bandwidth parameters (Forward SDR/PDR), filled by the NCC.
Return Profile	O (note 1)	Return C2P CoS and admitted transmit bandwidth parameters (Return SDR/PDR), filled by the NCC.
Route_ID	O	
Connection_Inactivity_Timeout	O	Value = 0 (no timeout)
IPv6_Source_Address	O (if IPv6) (note 2)	IPv6 source address(es) and mask(s)
IPv6_Destination_Address	O (if IPv6) (note 2)	If bidirectional: IPv6 destination address(es) and mask(s) of the other end RCST/RSGW
Maximum_Packet_Size	O	
Traffic_Spec_and_Policy_Data	O	
RC_Capacity_Parameters	M	Define CRA, VBDCmax and RBDCmax
RCST_Capability		
MAC Destination Address	O (if IPV6 with MPEG) (note 2)	In the case of bidirectional connections between NCC/GW and RCST MAC address of the NCC In the case of bidirectional connections between two RCSTs, two RSGWs or an RCST and an RSGW: MAC address of the other end RCST/RSGW
Connection_Status_Stats		
NOTE 1: Only the Forward or Return Stream ID and Profile, as appropriate, will be present in the case of unidirectional connections. NOTE 2: Not needed in the case of unidirectional connections.		

8.1.2 Connection Establishment Response (CnxEstResp)

8.1.2.1 RCST/RSGW initiated unicast connection

According to clause 8.1.1.1, the peer party can be an RCST, an RSGW or the NCC/GW.

If the peer party is different from the NCC/GW, the connection establishment response involves two messages: one from the peer party (RCST or RSGW) to NCC, and another one from the NCC to the initiating party (RCST or RSGW).

8.1.2.1.1 From peer RCST/RSGW to NCC (DULM)

Table 8.5: Unicast CnxEstResp Fields - RCST/RSGW initiated: peer RCST/RSGW to NCC

IE Fields		Peer RCST/RSGW to NCC	Comments	
Message Header	Message type	M	'0x17' or '0x13' or '0x05' (IPv4 with MPEG) 0x10 (otherwise)	"00010"
	Addressing type			"111" or "011" or "101" (IPv4 with MPEG) "000" (otherwise) "000" (note 1)
	Length	M	C2P message length	
	Connection ref	M	As defined in the corresponding CnxEstReq message	
Cause		M	It indicates whether the command has been successfully executed or not and provides the reason (cause) in case of un-success	
Channel_ID		O (note 4)	New or an already existing Channel_ID, to be used by the peer RCST/RSGW.	
Source Address		O (if IPv4) (note 4)	In the case of bidirectional connections: MAC address of the initiating RCST/RSGW for the IPv4 with MPEG format; or IPv4 source address(es) and mask(s) for the IPv4 with ATM format.	
Destination Address		O (If IPv4) (note 4)	In the case of bidirectional connections: IPv4 destination address(es) and mask(s) of the initiating RCST/RSGW.	
Forward Stream Identifier		O	Identifies the receive flow (VCC or PID) (note 1)	
Return Stream Identifier		O (note 4)	Identifies the transmit flow (VCC or PID) (note 1)	
Connection Type		O	'0x01' point-to-point bidirectional RCST/RSGW-Initiated; or '0x03' point-to-point unidirectional RCST/RSGW-Initiated	
Forward Profile		O	Forward C2P CoS and RCST/RSGW-accepted receive bandwidth parameters (Forward SDR/PDR) (note 2)	
Return Profile		O	Return C2P CoS and RCST/RSGW-accepted transmit bandwidth parameters (Return SDR/PDR) (note 2)	
Route_ID		O (note 4)		
Connection_Inactivity_Timeout		O		
IPv6_Source_Address		O (if IPv6)	IPv6 source address(es) and mask(s)	
IPv6_Destination_Address		O (if IPv6) (note 4)	IPv6 destination address(es) and mask(s) of the initiating RCST/RSGW	
Maximum_Packet_Size		O		
Traffic_Spec_and_Policy_Data		O		
RC_Capacity_Parameters		O (note 4)	Accepted CRA, VBDCmax and RBDCmax	
RCST_Capability				
MAC Destination Address		O (if IPv6 with MPEG) (note 4)	MAC address of the initiating RCST/RSGW	
Connection_Status_Stats				

NOTE 1: For IPv4 with MPEG format. In the case of a connection response reject message (cause different from success), Addressing type should be 0x05 (No address specified). No address fields will be present in the message.

NOTE 2: Forward and Return profiles can be different from the ones sent by the NCC (as reflected by the "accepted" bandwidth parameters), if they cannot be supported by the peer RCST/RSGW (for bandwidth or policy reasons).

NOTE 3: Only the Receive or Transmit Profiles, as appropriate, will be present in the case of unidirectional connections.

NOTE 4: Not needed in the case of unidirectional connections.

The message described in this clause is not needed in the case of star transparent connections from RCST/RSGW to NCC/GW.

8.1.2.1.2 From NCC to initiating RCST/RSGW (TIM)

Table 8.6: Unicast CnxEstResp Fields - RCST/RSGW initiated: NCC to initiating RCST/RSGW

IE Fields		RCST/RSGW to initiating NCC	Comments	
Message Header	Message type	M	'0x17' or '0x13' or '0x15' (IPv4 with MPEG) '0x10' (otherwise)	"00010"
	Addressing type			"111" or "011" or "101" (IPv4 with MPEG) "000" (otherwise) (note 1)
	Length	M	C2P message length	
	Connection ref	M	As defined in the corresponding CnxEstReq message	
Cause		M	It indicates whether the command has been successfully executed or not and provides the reason (cause) in case of un-success	
Channel_ID		M (note 3)	New or existing Channel_ID, to be used by the initiating RCST/RSGW	
Source Address		M (If IPv4 with MPEG) or O (if IPv4 with ATM)	MAC address of the peer RCST/RSGW for the IPv4 with MPEG format; or IPv4 source address(es) and mask(s) for the IPv4 with ATM format.	
Destination Address		M (If IPv4)	IPv4 destination address(es) and mask(s) of the peer RCST/RSGW	
Forward Stream Identifier		O (note 3)	VPI/VCI or PID assigned to the forward connection path	
Return Stream Identifier		M	VPI/VCI or PID assigned to the return connection path	
Connection Type		M	'0x01' point-to-point bidirectional RCST/RSGW-Initiated or '0x03' point-to-point unidirectional RCST/RSGW-Initiated	
Forward Profile		O (note 3)	Forward C2P CoS and admitted receive bandwidth parameters (Forward SDR/PDR) (note 2)	
Return Profile		M	Return C2P CoS and admitted transmit bandwidth parameters (Return SDR/PDR) (note 2)	
Route_ID		O (note 3)		
Connection_Inactivity_Timeout		O		
IPv6_Source_Address		O (if IPv6)	IPv6 source address(es) and mask(s)	
IPv6_Destination_Address		M (if IPv6)	IPv6 destination address(es) and mask(s) of the peer RCST/RSGW	
Maximum_Packet_Size		O		
Traffic_Spec_and_Policy_Data		O		
RC_Capacity_Parameters		M	Define CRA, VBDCmax and RBDCmax	
RCST_Capability				
MAC Destination Address		M (IPV6 with MPEG)	MAC address of the peer party	
Connection_Status_Stats				
NOTE 1: For IPv4 with MPEG format, In the case of a connection response reject message (cause different from success), Addressing type should be 0x05 (No address specified). No address fields will be present in the message. NOTE 2: Forward and Return profiles can be different from the one requested by the initiating RCST/RSGW (as reflected by the "admitted" bandwidth parameters) if they cannot be supported by the NCC or the peer RCST/RSGW (for bandwidth or policy reasons). NOTE 3: Not needed in the case of unidirectional connections.				

8.1.2.2 NCC initiated unicast connection

According to clause 8.1.2.1, the RCST(s) or RSGW(s), as parties or end-points of a unicast connection, are referred to as "RCST/RSGW" and "the other end RCST/RSGW". Table 8.7 only includes the parameters pertinent to one party (RCST or RSGW). In the case of a star transparent connection there is only one RCST involved, since the other end of the connection is the NCC/GW.

Table 8.7: Unicast CnxEstResp Fields - NCC-initiated: RCST/RSGW to initiating NCC

IE Fields		RCST/RSGW to initiating NCC	Comments	
Message Header	Message type	M	'0x17' or '0x13' or '0x15' (IPv4 with MPEG) 0x10 (otherwise)	"00010"
	Addressing type			"111" or "011" or "101" (IPv4 with MPEG) "000" (otherwise) (note 1)
	Length	M	C2P message length	
	Connection ref	M	As defined in the corresponding CnxEstReq message	
Cause		M	It indicates whether the command has been successfully executed or not and provides the reason (cause) in case of un-success	
Channel_ID		O	New or existing Channel_ID, to be used by the peer RCST/RSGW	
Source Address		O (If IPv4) (note 4)	In the case of bidirectional connections between NCC/GW and RCST MAC address of the NCC for the IPv4 with MPEG format; or IPv4 source address(es) and mask(s) for the IPv4 with ATM format. In the case of bidirectional connections between two RCSTs, two RSGWs or an RCST and an RSGW: MAC address of the other end RCST/RSGW for the IPv4 with MPEG format; or IPv4 source address(es) and mask(s) for the IPv4 with ATM format.	
Destination Address		O (If IPv4) (note 4)	In the case of bidirectional connections between NCC/GW and RCST: IPv4 destination address(es) and mask(s) of the NCC. In the case of bidirectional connections between two RCSTs/RSGWs or an RCST and an RSGW: IPv4 destination address(es) and mask(s) of the other end RCST/RSGW	
Forward Stream Identifier		O (note 3)	VPI/VCI or PID assigned to connection for the Forward path	
Return Stream Identifier		O (note 3)	VPI/VCI or PID assigned to connection for the Return path	
Connection Type		M	'0x02' point-to-point bidirectional NCC-Initiated or '0x04' point-to-point unidirectional NCC-Initiated	
Forward Profile		O (note 3)	Forward C2P CoS and accepted receive bandwidth parameters (Forward SDR/PDR (note 2)	
Return Profile		O (note 3)	Return C2P CoS and accepted transmit bandwidth parameters (Return SDR/PDR (note 2)	
Route_ID		O		
Connection_Inactivity_Timeout		O	Value = 0 (no timeout)	
IPv6_Source_Address		O (If IPv6)	IPv6 source address(es) and mask(s) (note 1)	
IPv6_Destination_Address		O (If IPv6) (note 4)	IPv6 destination address(es) and mask(s) of the other end RCST/RSGW	
Maximum_Packet_Size		O		
Traffic_Spec_and_Policy_Data		O		
RC_Capacity_Parameters		O	Define CRA, VBDCmax and RBDCmax	
RCST_Capability				

IE Fields	RCST/RSGW to initiating NCC	Comments
MAC Destination Address	O (If IPv6 with MPEG) (note 4)	In the case of bidirectional connections between NCC/GW and RCST: MAC address of the NCC In the case of bidirectional connections between two RCSTs/RSGWs or an RCST and an RSGW: MAC address of the other end RCST/RSGW
Connection_Status_Stats		
NOTE 1: For IPv4 with MPEG format, In the case of a connection response reject message (cause different from success), addressing type should be 0x05 (No address specified). No address fields will be present in the message. NOTE 2: Forward and Return Profiles can be different from the one sent by the NCC (as reflected by the "accepted" bandwidth parameters), if they cannot be supported by the NCC or the other end RCST/RSGW (for bandwidth or policy reasons). NOTE 3: Only the Forward or Return Stream ID and Profile, as appropriate, will be present In the case of unidirectional connections. NOTE 4: Not needed in the case of unidirectional connections.		

8.2 Point-to-multipoint connection establishment messages

A point-to-multipoint connections is a multicast unidirectional connection between a multicast source and a multicast destination, as end points (or parties) of the connection.

According to clause 4.4.2.2, a multicast connection carries multicast sessions, consisting of IP streams with an IP destination address which is part of a Multicast Group addresses.

The source of a multicast session can be an RCST (mesh multicast) or a Gateway (RSGW or NCC/GW) (star multicast) (see clause 4.4.2.2 for the nomenclature used for multicast connections).

The destination (party) of a multicast session is constituted, in general, by other RCSTs, but it can also include other RSGWs (in the case of regenerative star/mesh scenario) or the NCC/GW (in the case of transparent scenario, if two-hop multicasting is supported).

The multicast connections can be initiated by the RCST, RSGW or NCC; therefore three different types of multicast connections are distinguished:

- Point-to-multipoint RGSW-initiated (star/mesh regenerative reference scenario).
- Point-to-multipoint RCST-initiated (mesh regenerative/transparent multicast, star transparent reference scenario).
- Point-to-multipoint NCC-initiated (mesh regenerative/transparent reference scenario, star transparent reference scenario).

For point-to-multipoint RCST-initiated or RGSW-initiated connection, the multicast source is the corresponding RCST or RSGW.

For point-to-multipoint NCC-initiated connections the multicast source can be either an RCST, or an RSGW or the NCC/GW.

The tables in the following clauses only include the parameters pertinent to one destination downlink/one destination party. Since the multicast connections are unidirectional, the messages used for their establishment do not include information about the receive side, but only about the transmit side (i.e. the Return Stream Identifier and Return Profile).

8.2.1 Connection Establishment Request (CnxEstReq)

8.2.1.1 RSGW initiated regenerative multicast connection

The RGSW-initiated multicast connections can be:

- towards RCSTs (star regenerative reference scenario); or
- towards other RSGWs (star/mesh regenerative reference scenario).

Table 8.8: Multicast CnxEstReq Fields - RSGW initiated: Initiating RSGW (Multicast source) to NCC

Fields		Initiating RSGW (Multicast source) to NCC	Comments	
Message Header	Message type	M	'0x0C' (IPv4 with MPEG) '0x08' (otherwise)	"00001"
	Addressing type			"100" (IPv4 with MPEG) "000" (otherwise)
	Length	M	C2P message length	
	Connection ref	M	Connection reference ID selected by the multicast source RSGW	
Cause				
Channel_ID				
Source Address		O (If IPv4)	IPv4 source address and mask of the IP multicast session	
Destination Address		M (If IPv4)	IPv4 destination address and mask of the IP multicast session	
Forward Stream Identifier				
Return Stream Identifier				
Connection Type		M	'0x05' point-to-multipoint RSGW-Initiated (regenerative star/mesh multicast) (note)	
Forward Profile				
Return Profile		M	Return C2P CoS and requested transmit bandwidth parameters (Return SDR/PDR) filled by the RSGW	
Route_ID				
Connection_Inactivity_Timeout		O		
IPv6_Source_address		O (if IPv6)	IPv6 source address and mask of the IP multicast session	
IPv6_Destination_Address		M (if IPv6)	IPv6 destination address and mask of the IP multicast session	
Maximum_Packet_Size		O		
Traffic_Spec_and_Policy_data		O		
RC_Capacity_Parameters				
RCST_Capability				
MAC_Destination_Address				
Connection_Status_Stats				
NOTE: Star multicast is towards RCSTs, mesh multicast is towards other RSGWs.				

8.2.1.2 RCST initiated multicast connection

The RCST-initiated multicast connections can be:

- towards other RCSTs (mesh regenerative/transparent reference scenario); or
- towards RSGWs (star regenerative reference scenario); or
- towards the NCC/GW (star transparent reference scenario, if two-hop multicasting is supported).

Table 8.9: Multicast CnxEstReq Fields - RCST initiated: Initiating RCST (Multicast source) to NCC

Fields		Initiating RCST (Multicast source) to NCC	Comments	
Message Header	Message type	M	'0x0C' (IPv4 with MPEG) '0x08' (otherwise)	"00001"
	Addressing type			"100" (IPv4 with MPEG) "000" (Otherwise)
	Length	M	C2P message length	
	Connection ref	M	Connection reference ID selected by the multicast source RCST	
Cause				
Channel_ID				
Source Address		O (If IPv4)	IPv4 source address and mask of the multicast IP session (note)	
Destination Address		M (If IPv4)	IPv4 destination address and mask of the IP multicast session	
Forward Stream Identifier				
Return Stream Identifier				
Connection Type		M	'0x06' point-to-multipoint RCST-Initiated (mesh/star multicast) (note)	
Forward Profile				
Return Profile		M	Return C2P CoS and requested transmit bandwidth parameters (Return SDR/PDR) filled by the RCST	
Route_ID				
Connection_Inactivity_Timeout		O		
IPv6_Source_address		O (if IPv6)	IPv6 source address and mask of the IP multicast session	
IPv6_Destination_Address		M (if IPv6)	IPv6 destination address and mask of the IP multicast session	
Maximum_Packet_Size		O		
Traffic_Spec_and_Policy_data		O		
RC_Capacity_Parameters				
RCST_Capability				
MAC_Destination_Address				
Connection_Status_Stats				
NOTE: Mesh multicast is towards other RCSTs and possibly RSGWs (regenerative only), star multicast is towards the NCC/GW (two-hop transparent multicast).				

8.2.1.3 NCC initiated multicast connection

The NCC-initiated multicast connections can be:

- from the initiating NCC/GW towards RCSTs (star transparent reference scenario);
- from a multicast source RCST towards other RCSTs (mesh transparent/regenerative reference scenarios) and possibly RSGWs (only for regenerative scenario) or towards the NCC/GW (star transparent reference scenario, if two-hop multicasting is supported);
- from a multicast source RSGW to RCSTs (star regenerative scenario) or to other RSGWs (mesh regenerative scenario).

The NCC-initiated request message for the establishment of a multicast connection from an RCST/RSGW (as multicast source) is described in table 8.10. No C2P message is needed when the multicast source is the NCC/GW.

Table 8.10: Multicast CnxEstReq Fields - NCC initiated: Initiating NCC to multicast source RCST/RSGW

Fields		Initiating NCC to RCST/RSGW	Comments	
Message Header	Message type	M	'0x0F' (IPv4 with MPEG) '0x08' (otherwise)	"00001"
	Addressing type			"111" (IPv4 with MPEG) "000" (otherwise)
	Length	M	C2P message length	
	Connection ref	M	Connection reference id selected by NCC	
Cause				
Channel_ID		M	A new or already existing Channel_ID	
Source Address		M (If IPv4 with MPEG) O (If IPv4 with ATM)	MAC address corresponding to the IP multicast address for IPv4 with MPEG format (note 1); or IPv4 source address and mask of the IP multicast session for the IPv4 with ATM format.	
Destination Address		M (If IPv4)	IPv4 destination address and mask of the multicast IP session	
Forward Stream Identifier				
Return Stream Identifier		M	Multicast VPI/VCI or PID	
Connection Type		M	'0x07' Point-to-multipoint NCC-Initiated (note 2)	
Forward Profile				
Return Profile		M	Return C2P CoS and admitted transmit bandwidth parameters (Return SDR/PDR), filled by the NCC	
Route_ID		O		
Connection_Inactivity_Timeout		O	Value = 0 (no timeout)	
IPv6_Source_address		O (if IPv6)	IPv6 source address and mask of the multicast IP session	
IPv6_Destination_Address		M (if IPv6)	IPv6 destination address and mask of the multicast IP session	
Maximum_Packet_Size		O		
Traffic_Spec_and_Policy_data		O		
RC_Capacity_Parameters		M	Define CRA, VBDCMax, and RBDCMax	
RCST_Capability				
MAC_Destination_Address		M (If IPv6 with MPEG)	MAC address corresponding to the IP multicast address (note 1)	
Connection_Status_Stats				

NOTE 1: The destination multicast MAC addresses are derived from the multicast group IP address, as defined in RFC 1112 [i.9].

NOTE 2: Transparent star connections are from the initiating NCC/GW towards RCSTs or from the source RCST towards NCC/GW (for two-hop mesh multicast); transparent mesh connections are from the source RCST towards other RCSTs; regenerative star connections are from source RSGW towards RCSTs; regenerative mesh connections are from the source RCST towards other RCSTs and possibly towards RSGWs.

8.2.2 Connection Establishment Response (CnxEstResp)

8.2.2.1 RSGW initiated regenerative multicast connection

See clause 8.2.1.1 for the flavours of the RGSW-initiated regenerative multicast connections.

Table 8.11: Multicast CnxEstResp Fields - RSGW initiated: NCC to Initiating RSGW (Multicast source)

Fields		NCC to Initiating RSGW (Multicast source)	Comments	
Message Header	Message type	M	'0x11' or '0x15' (IPv4 with MPEG) '0x10' (otherwise)	"00010"
	Addressing type			"001"or "101" (IPv4 with MPEG) "000" (otherwise)
	Length	M	C2P message length	
	Connection ref	M	As defined in the corresponding CnxEstReq message	
Cause		M	It indicates whether the command has been successfully executed or not (error reason) (note 1)	
Channel_ID		M	Can be a new or an already existing Channel_ID	
Source Address		M (If IPv4 with MPEG) O (If IPv4 with ATM)	MAC address corresponding to the IP multicast address for IPv4 with MPEG format (note 2); or IPv4 source address and mask of the IP multicast session for the IPv4 with ATM format.	
Destination Address		M (If IPv4)	IPv4 destination address of the IP multicast packet (IPv4 multicast group address)	
Forward Stream Identifier				
Return Stream Identifier		M	Multicast VPI/VCI or PID	
Connection Type		M	'0x05' Point-to-multipoint RSGW-Initiated (regenerative star/mesh multicast) (note 4)	
Forward Profile				
Return Profile		O	Return C2P CoS and admitted rate parameters for RSGW multicast transmission (Return SDR/PDR) (note 3)	
Route_ID		O		
Connection_Inactivity_Timeout		O		
IPv6_Source_address		O (if IPv6)	IPv6 source address and mask of the IP multicast session	
IPv6_Destination_Address		M (if IPv6)	IPv6 destination address and mask of the multicast IP session	
Maximum_Packet_Size		O		
Traffic_Spec_and_Policy_data		O		
RC_Capacity_Parameters		M	Define CRA, VBDCmax and RBDCmax	
RCST_Capability				
MAC_Destination_Address		M (If IPV6 with MPEG)	MAC address corresponding to the IP multicast address (note 2)	
Connection_Status_Stats				
NOTE 1: In the case of a connection response reject message (cause different from success), Addressing type should be 0x05 (No address specified). No address fields will be present in the message. NOTE 2: The destination multicast MAC addresses are derived from the group IP address, as defined in RFC 1112 [i.9]. NOTE 3: Return Profile can be different from the one sent by the RSGW (as reflected by the "admitted return rate" parameters), if it cannot be supported due to capacity or policy reasons. NOTE 4: Star multicast is towards RCSTs, mesh multicast is towards other RSGWs.				

8.2.2.2 RCST initiated multicast connection

See clause 8.2.1.2 for the flavours of the RCST-initiated multicast connections.

Table 8.12: Multicast CnxEstResp - RCST initiated: NCC to Initiating RCST (Multicast source)

Fields		NCC to initiating RCST (Multicast Source)	Comments	
Message Header	Message type	M	'0x17' or '0x15' (IPv4 with MPEG) '0x10' (otherwise)	"00010"
	Addressing type			"111" or "101" (IPv4 with MPEG) "000" (otherwise)
	Length	M	C2P message length	
	Connection ref	M	As defined in the corresponding CnxEstReq message	
Cause		M	It indicates whether the command has been successfully executed or not (error reason) (note 1)	
Channel_ID		M	Can be a new or an already existing Channel_ID	
Source Address		M (If IPv4 with MPEG) O (If IPv4 with ATM)	MAC address corresponding to the IP multicast address for IPv4 with MPEG format (note 2); or IPv4 source address(es) and mask(s) of the IP multicast session for the IPv4 with ATM format.	
Destination Address		O (if IPv4)	Destination IPv4 address of the IP multicast session	
Forward Stream Identifier				
Return Stream Identifier		M	Multicast VPI/VCI or PID	
Connection Type		M	'0x06' Point-to-multipoint RSGW-Initiated (mesh/star multicast) (note 4)	
Forward Profile				
Return Profile		M	Return C2P CoS and admitted rate parameters for RCST multicast transmission (return SDR/PDR) (note 3)	
Route_ID		O		
Connection_Inactivity_Timeout		O		
IPv6_Source_address		O (If IPv6)	IPv6 source address and mask of the IP multicast session	
IPv6_Destination_Address		M (if IPv6)	IPv6 destination address and mask of the IP multicast session	
Maximum_Packet_Size		O		
Traffic_Spec_and_Policy_data		O		
RC_Capacity_Parameters		M	Define CRA, VBDCmax and RBDCmax	
RCST_Capability				
MAC_Destination_Address		M (If IPv6 with MPEG)	MAC address corresponding to the IP multicast address (note 2)	
Connection_Status_Stats				

NOTE 1: In the case of a connection response reject message (cause different from success), Addressing type should be 0x05 (No address specified). No address fields will be present in the message.

NOTE 2: Multicast MAC addresses are derived from the group IP address, as defined in RFC 1112 [i.9].

NOTE 3: Transmit Profile can be different from the one sent by the RCST (as reflected by the "admitted return rate parameters), if it cannot be supported due to capacity or policy reasons.

NOTE 4: Mesh multicast is towards other RCSTs and possibly RSGWs (regenerative only); star multicast is towards the NCC/GW (two-hop transparent multicast).

8.2.2.3 NCC initiated multicast connection

See clause 8.2.1.3 for the flavours of the NCC-initiated multicast connections.

Table 8.13: Multicast CnxEstResp - NCC initiated: Multicast source RCST or RSGW to NCC

Fields		Multicast source (RCST or RSGW) to Initiating NCC	Comments	
Message Header	Message type	M	'0x17' or '0x15' (IPv4 with MPEG) '0x10' (otherwise)	"00010"
	Addressing type			"111" or "101" (IPv4 with MPEG) "000" (Otherwise)
	Length	M	C2P message length	
	Connection ref	M	As defined in the corresponding CnxEstReq message	
Cause		M	It indicates whether the command has been successfully executed or not (error reason)	
Channel_ID		Ó	Can be a new or an already existing Channel_ID	
Source Address		O	MAC address corresponding to the IP multicast address for IPv4 with MPEG format (note 2); or IPv4 source address and mask of the IP multicast session for the IPv4 with ATM format.	
Destination Address		O (If IPv4)	Destination IPv4 address of the IP multicast session	
Forward Stream Identifier				
Return Stream Identifier		O	Multicast VPI/VCI or PID	
Connection Type		O	"0x07" Point-to-multipoint NCC-Initiated (note 4)	
Forward Profile				
Return Profile		O	Return C2P CoS and accepted rate parameters for RCST/RSGW multicast transmission (return SDR/PDR) (note 3)	
Route_ID		O		
Connection_Inactivity_Timeout		O	Value = 0 (no timeout)	
IPv6_Source_address		O (If IPv6)	IPv6 source address and mask of the IP multicast session	
IPv6_Destination_Address		O (if IPv6)	IPv6 destination address of the IP multicast session	
Maximum_Packet_Size		O		
Traffic_Spec_and_Policy_data		O		
RC_Capacity_Parameters		O		
RCST_Capability				
MAC_Destination_Address		O (If IPv6 with MPEG)	MAC address corresponding to the IP multicast address (note 2)	
Connection_Status_Stats				

NOTE 1: In the case of a connection response reject message (cause different than success), the addressing type should be 0x05 (No address specified). No address fields will be present in the message.

NOTE 2: Multicast MAC addresses are derived from the group IP address, as defined in RFC 1112 [i.9].

NOTE 3: Return Profile can be different from the one sent by the NCC (as reflected by the "admitted return rate parameters), if it cannot be supported due to capacity or policy reasons.

NOTE 4: Transparent star connections are from the initiating NCC/GW towards RCSTs or from the source RCST towards NCC/GW (for two-hop mesh multicast), transparent mesh connections are from the source RCST towards other RCSTs, regenerative star connections are from source RSGW towards RCSTs, regenerative mesh connections are from the source RCST to other RCSTs and possibly towards RSGWs.

8.3 Connection release messages

Connection release messages apply to all types of connections, initiated by either an RCST/RSGW or NCC.

According to clause 6.3, in the case of mesh transparent scenario and mesh/star regenerative scenario, connection release messages should be used on both legs of the signalling path (between an RCST/RSGW and NCC, and between NCC and the other RCST/RSGW), regardless of the connection initiator (RCST, RSGW or NCC). In star transparent scenarios the signalling path only has one leg (between RCST and NCC/GW); therefore only one connection release request/response message is needed.

The structure of connection release messages, as shown in tables 8.14 and 8.15, is very simple and is the same for all connections, whether initiated by RCST/RSGW or NCC. The terminology used for the parties involved in a connection, which is as defined in clause 4.4.2.2, applies to the case where the connection release request is issued by the connection initiator. If the connection release request were triggered by an inactivity timer at RCST/RSGW, the message structure would remain the same but the terminology would need to be changed to reflect a specific reference scenario (e.g. for NCC-initiated connections between two RCST/RSGW, the initiating RCST/RSGW and peer RCST/RSGW should be replaced by RCST/RSGW A and RCST/RSGW B, respectively).

8.3.1 Connection Release Request (CnxRelReq)

Table 8.14: Connection Release Request Fields

Fields		Initiating RCST/RSGW to NCC	NCC to peer RCST/RSGW	Initiating NCC to RCST/RSGW	Comments	
Message Header	Message type	M	M	M	'0x1D' (IPv4 with MPEG) '0x18' (otherwise)	"00011"
	Addressing type					"101" (IPv4 with MPEG) "000" (otherwise)
	length	M	M	M	C2P message length	
	connection ref	M	M	M	Connection reference id	
Cause		M	M	M	It indicates whether the command has been successfully executed or not (error reason) See clause 7.2.2.2	
Channel_ID		O	M	M	Existing Channel_ID	
Source Address						
Destination Address						
Forward Stream Identifier						
Return Stream Identifier						
Connection Type						
Forward Profile						
Return Profile						
Route_ID						
Connection_Inactivity_Timeout						
IPv6_Source_address						
IPv6_Destination_Address						
Maximum_Packet_Size						
Traffic_Spec_and_Policy_data						
RC_Capacity_Parameters			O	O	Updated CRA, VBDCmax and RBDCmax	
RCST_Capability						
MAC_Destination_Address						
Connection_Status_Stats						

8.3.2 Connection Release Response (CnxRelResp)

Table 8.15: Connection Release Response Fields

Fields		Peer RCST/RSGW to NCC	NCC to initiating RCST/RSGW	RCST/RSGW to Initiating NCC	Comments	
Message Header	Message type	M	M	M	'0x25' (IPv4 with MPEG) '0x20' (otherwise)	"00100"
	Addressing type					"101" (IPv4 with MPEG) "000" (otherwise)
	length	M	M	M	C2P message length	
	connection ref	M	M	M	Connection reference id	
Cause		m	M	M	It indicates whether the command has been successfully executed or not (error reason) See clause 7.2.2.2	
Channel_ID		O	O	O	Existing Channel ID	
Source Address						
Destination Address						
Forward Stream Identifier						
Return Stream Identifier						
Connection Type						
Forward Profile						
Return Profile						
Route_ID						
Connection_Inactivity_Timeout						
IPv6_Source_address						
IPv6_Destination_Address						
Maximum_Packet_Size						
Traffic_Spec_and_Policy_data						
RC_Capacity_Parameters			O		Updated CRA, VBDCmax and RBDCmax	
RCST_Capability						
MAC_Destination_Address						
Connection_Status_Stats						

8.4 Connection modify messages

Connection modify messages apply to point-to-point and point-to-multipoint connections, whether initiated by an RCST, an RSGW or by the NCC.

Connection modify requests can be triggered by the NCC or by any party involved in the transmission of the traffic carried by the connection.

In the case of **point-to-point connections**:

- For mesh regenerative/transparent scenarios and star regenerative scenarios, connection modify messages shall be used for both legs of the signalling path: from one RCST/RSGW to NCC and from NCC to the other RCST/RSGW.

- For star transparent scenario, connection modify messages shall be exchanged between NCC/GW and RCST.

In the case of **point-to-multipoint connections**, connection modify messages shall only be exchanged between NCC and multicast source (if different from the NCC/GW).

Connection modify messages in the present document are only used for the modification of the Return and Forward Profiles of a connection and optionally for the modification of the RC capacity parameters of the RC pertinent to the connection.

The terminology used for the parties involved in the Connection Modify Request (table 8.16) and Connection Modify Response (table 8.17) is as defined in clause 4.4.2.2.

8.4.1 Connection Modify Request (CnxModReq)

Table 8.16: Connection Modify Request Fields

Fields		Initiating RCST/RSGW to NCC (note 1)	NCC to peer RCST/RSGW (note 2)	Initiating NCC to RCST/RSGW	Comments	
Message Header	Message type	M	M	M	For IPv4 with MPEG: '0x2D' Otherwise: '0x28'	"00101"
	Addressing type					For IPv4 with MPEG: "101" Otherwise: "000"
	length	M	M	M	C2P message length	
	connection ref	M	M	M	Existing connection reference id	
Cause						
Channel_ID		O	O	O	Existing Channel ID	
Source Address						
Destination Address						
Forward Stream Identifier		O	O	O		
Return Stream Identifier		O	O	O		
Connection Type		M	M	M	Any connection type	
Forward Profile		O	M	O	Forward C2P CoS and requested bandwidth parameters (Forward SDR/PDR) filled by the initiating RCST/RSGW or NCC (note 3)	
Return Profile		M	O	O	Return C2P CoS and requested bandwidth parameters (Return SDR/PDR) filled by the initiating RCST/RSGW or NCC (note 3)	
Route_ID		O				
Connection_Inactivity_Timeout		O	O	O		
IPv6_Source_address						
IPv6_Destination_Address						
Maximum_Packet_Size		O	O	O		
Traffic_Spec_and_Policy_data		O	O	O		
RC_Capacity_Parameters		O	O	O	CRA, VBDCmax, and RBDCmax (note 4)	
RCST_Capability						
MAC_Destination_Address						
Connection_Status_Stats						

NOTE 1: This message is sent for all RCST/RGSW-initiated connections, regardless of the peer party (other RCST/RSGW or NCC/GW).

NOTE 2: This message is not needed for RCST-initiated connections for which the NCC/GW is the peer party (star transparent).

NOTE 3: Connection Modify Request can be used to modify the Forward Profile and/or the Return Profile of the connection in the case of bidirectional connection, or only of the Return Profile in the case of unidirectional connections. Point-to-multipoint connections are unidirectional.

NOTE 4: In the case of a request message from Initiating RCST/RSGW to NCC, RC Capacity parameters contain the requested new values for CRA, VBDCMax and RBDCMax. In the case of a request message from the NCC to a peer RCST/RSGW or from the Initiating NCC to an RCST/RSGW, the RC capacity parameters reflect the updated new values of CRA, VBDCMax and RBDCMax.

8.4.2 Connection Modify Response (CnxModResp)

Table 8.17: Connection Modify Response Fields

Fields		NCC to initiating RCST/RSGW (note 1)	Peer RCST/RSGW to NCC (note 2)	RCST/RSGW to initiating NCC	Comments	
Message Header	Message type	M	M	M	For IPv4 with MPEG: '0x25' Otherwise: '0x20'	"00100"
	Addressing type					For IPv4 with MPEG: "101" Otherwise: "000"
	length	M	M	M	C2P message length	
	connection ref	M	M	M	Existing Connection reference id	
Cause		M	M	M	It indicates whether the command has been successfully executed or not (error reason).	
Channel_ID		O	O	O	Existing Channel ID	
Source Address						
Destination Address						
Forward Stream Identifier		O	O	O		
Return Stream Identifier		O	O	O		
Connection Type		M	M	M	Any connection type	
Forward Profile		O	M	O	Forward C2P CoS and admitted/accepted bandwidth parameters (Forward SDR/PDR) from NCC or RCST/RSGW (notes 3 and 4)	
Return Profile		M	O	O	Return C2P CoS and admitted/accepted bandwidth parameters (Return SDR/PDR) from the NCC or RCST/RSGW (notes 3 and 4)	
Route_ID		O				
Connection_Inactivity_Timeout		O	O	O		
IPv6_Source_address						
IPv6_Destination_Address						
Maximum_Packet_Size		O	O	O		
Traffic_Spec_and_Policy_data		O				
RC_Capacity_Parameters		O	O	O	Updated CRA, VBDCmax, and RBDCmax	
RCST_Capability						
MAC_Destination_Address						
Connection_Status_Stats						

NOTE 1: This message is sent for all RCST/RGSW-initiated connections, regardless of the peer party (other RCST/RSGW or NCC/GW).

NOTE 2: This message is not needed for RCST/RGSW-initiated connections for which the NCC/GW is the peer party (star transparent).

NOTE 3: Connection Modify Response can be used to modify the Forward Profile and/or the Return Profile of the connection, depending on the type of connection (unidirectional or bidirectional). Point-to-multipoint connections are unidirectional.

NOTE 4: Forward and Return Profiles can be different from those sent by the initiating party (as reflected by the "admitted/accepted" rate parameters), if the peer/other party cannot support them or if there are not enough resources.

8.5 RC Modify messages

Please refer to clause 6.7 for the usage of RC modify messages, complemented by the following observations:

- For connections having RCSTs or RSGWs as end points, separate explicit RC Modify Requests may be sent by/to each RCST/RSGW. Tables 8.18 and 8.19 only include the parameters (fields) for one RCST/RSGW.
- For RC modify messages implicitly triggered by NCC as a result of receiving a connection establishment/modify request, the fields in the RC modify messages are the same as in the case of explicit RC modify messages, but the requested values are derived from the Transmit Profile in the corresponding connection establishment/modify request and the current RC_Capacity_Parameters of the RC pertaining to the connection.

NOTE: The implicit RC modify requests triggered by an RCST/RSGW are part of connection establishment/modify requests, provided that the RC_Capacity_Parameters field is present in the request messages.

8.5.1 RC Modify Request (RCModReq)

Table 8.18: RC Modify Request Fields

Fields		RCST/RSGW to NCC	NCC to RCST/RSGW	Comments	
Message Header	Message type	M	M	'0x5D' (IPv4 with MPEG) or '0x58' (otherwise)	"01011"
	Addressing type				"101" (IPv4 with MPEG) or "000" (otherwise)
	length	M	M	C2P message length	
	connection ref	M	M	No connection reference id (0x0000)	
Cause					
Channel_ID		M	M	Existing Channel_ID	
Source Address					
Destination Address					
Receive Stream Identifier					
Transmit Stream Identifier					
Connection Type		O	O	Any connection type	
Receive Profile					
Transmit Profile					
Connection_Inactivity_Timeout					
IPv6_Source_address					
IPv6_Destination_Address					
Maximum_Packet_Size					
Traffic_Spec_and_Policy_data					
RC_Capacity_Parameters		M	M	Requested CRA, RBDCmax, VBDCmax values	
RCST_Capability					
MAC_Destination_Address					
Connection_Status_Stats					

8.5.2 RC Modify Response (RCModResp)

Table 8.19: RC Modify Response Fields

Fields		NCC to RCST/RSGW	RCST/RSGW to NCC	Comments	
Message Header	Message type	M	M	'0x65' (IPv4 with MPEG) or '0x60' (otherwise)	"01100"
	Addressing type				"101" (IPv4 with MPEG) or "000" (otherwise)
	length	M	M	C2P message length	
	connection ref	M	M	No connection reference id (note)	
Cause		M	M	It indicates whether the command has been successfully executed or not (error reason).	
Channel_ID		M	M	Existing RC Channel_ID	
Source Address					
Destination Address					
Receive Stream Identifier					
Transmit Stream Identifier					
Connection Type		O	O	Any connection type	
Receive Profile					
Transmit Profile					
Connection_Inactivity_Timeout					
IPv6_Source_address					
IPv6_Destination_Address					
Maximum_Packet_Size					
Traffic_Spec_and_Policy_data					
RC_Capacity_Parameters		M	M	Admitted/Accepted CRA, RBDCmax, VBDCmax values (note)	
RCST_Capability					
MAC_Destination_Address					
Connection_Status_Stats					
NOTE: RC capacity parameters can be different from the requested ones, as reflected by the "admitted" parameters or "accepted" parameters, if they cannot be supported by the NCC or RCST/RSGW, respectively.					

8.6 RCST Capability messages

The rational for RCST capability messages is provided in clauses 4.6.1 and 6.8.

According to clause 4.6.1, an RCST can notify its C2P-related capabilities to the NCC during the second logon phase; the NCC should respond with an acknowledgment message to the RCST.

According to clause 6.8, the NCC can request information from an RCST about its C2P-related capability; the RCST should respond by sending the requested information to the NCC.

8.6.1 RCST Capability Request (RCSTCapReq)

This C2P command shall include:

- the notification by a logging-on RCST/RSGW of its C2P-related capabilities to the NCC during the second logon phase; or
- the RCST Capability Request sent by the NCC to an RCST.

Table 8.20: RCST Capability Request Fields

Fields		logging-on RCST/RSGW to NCC	NCC to RCST/RSGW	Comments	
Message Header	Message type	M	M	'0x8D' (IPv4 with MPEG) '0x88' (otherwise)	"10001"
	Addressing type				"101" (IPv4 with MPEG) "000" (otherwise)
	length	M	M	C2P message length	
	connection ref	M	M	No connection reference id (0x0000)	
Cause					
Channel_ID					
Source Address					
Destination Address					
Receive Stream Identifier					
Transmit Stream Identifier					
Connection Type					
Receive Profile					
Transmit Profile					
Connection_Inactivity_Timeout					
IPv6_Source_address					
IPv6_Destination_Address					
Maximum_Packet_Size					
Traffic_Spec_and_Policy_data					
RC_Capacity_Parameters					
RCST_Capability		M	O	C2P version, IP protocol version, IP multicast option supported by the calling RCST	
MAC_Destination_Address					
Connection_Status_Stats					

8.6.2 RCST Capability Response (RCSTCapResp)

This C2P command shall include:

- the acknowledgment by the NCC, in response to an RCST Capability Request of a logging-on RCST/RSGW during the second logon phase; or
- the RCST/RSGW's response to the RCST Capability Request sent by the NCC.

Table 8.21: RCST Capability Response Fields

Fields		NCC to logging-on RCST/RSGW	RCST/RSGW to NCC	Comments	
Message Header	Message type	M	M	'0x95' (IPv4 with MPEG) '0x90' (otherwise)	"10010"
	Addressing type				"101" (IPv4 with MPEG) "000" (otherwise)
	length	M	M	C2P message length	
	connection ref	M	M	No connection reference id (0x0000)	
Cause		M	M	It indicates whether the command has been successfully executed or not (error reason) (note 1)	
Channel_ID					
Source Address					
Destination Address					
Receive Stream Identifier					
Transmit Stream Identifier					
Connection Type					
Receive Profile					
Transmit Profile					
Connection_Inactivity_Timeout					
IPv6_Source_address					
IPv6_Destination_Address					
Maximum_Packet_Size					
Traffic_Spec_and_Policy_data					
RC_Capacity_Parameters					
RCST_Capability		O	M	NCC to RCST/RSGW: C2P version, IP protocol version, IP multicast option supported by the NCC (note 2) RCST/RSGW to NCC: C2P version, IP protocol version, IP multicast option supported by the RCST/RSGW	
MAC_Destination_Address					
Connection_Status_Stats					

NOTE 1: Cause code '0x0000' (success) represents an acknowledgment that the NCC supports the RCST/RSGW capabilities or an acknowledgment that the RCST/RSGW is returning its capabilities to the NCC.

NOTE 2: These parameters are returned for information purposes in the case of Cause code '0x0017'.

8.7 Connection Status Stats Messages

8.7.1 Connection Status Stats Request (CnxStatusStatsReq)

Table 8.22: Connection Status Stats Request Fields

Fields		NCC to RCST/RSGW	Comments	
Message Header	Message type	M	'0x7D' (IPv4 with MPEG) '0x78' (otherwise)	"01111" = 0x0F
	Addressing type			"101" (IPv4 with MPEG) "000" (otherwise)
	length	M	C2P message length	
	connection ref	M	No connection reference id (0x0000)	
Cause				
Channel_ID				
Source Address				
Destination Address				
Forward Stream Identifier				
Return Stream Identifier				
Connection Type				
Forward Profile				
Return Profile				
Connection_Inactivity_Timeout				
IPv6_Source_address				
IPv6_Destination_Address				
Maximum_Packet_Size				
Traffic_Spec_and_Policy_data				
RC_Capacity_Parameters				
RCST_Capability				
MAC_Destination_Address				
Connection_Status_Stats		O	This IE contains only the connection reference ID for which the NCC wants to get the status and statistics. This IE can be repeated in this message. If this IE is not present in this message, the Connection_Status_Stats response will contain all the active connection reference IDs and the status and the statistics of all active connections of the RCST/RSGW	

8.7.2 Connection Status Stats Response (CnxStatusStatsResp)

Table 8.23: Connection Status Stats Response Fields

Fields		RCST/RSGW to NCC	Comments	
Message Header	Message type	M	'0x85' (IPv4 with MPEG) '0x80' (otherwise)	"10000" = 0x10
	Addressing type			"101" (IPv4 with MPEG) "000" (otherwise)
	length	M	C2P message length	
	connection ref	M	No connection reference id (0x0000)	
Cause				
Channel_ID				
Source Address				
Destination Address				
Forward Stream Identifier				
Return Stream Identifier				
Connection Type				
Forward Profile				
Return Profile				
Connection_Inactivity_Timeout				
IPv6_Source_address				
IPv6_Destination_Address				
Maximum_Packet_Size				
Traffic_Spec_and_Policy_data				
RC_Capacity_Parameters				
RCST_Capability				
MAC_Destination_Address				
Connection_Status_Stats		M	One or multiple Connection_Status_Stats IE(s) can be included in this message, depending of the number of requested connection status and statistics in the Connection_Status_Stats Request message. Each IE contains the connection reference ID, the status and the statistics of the connection.	

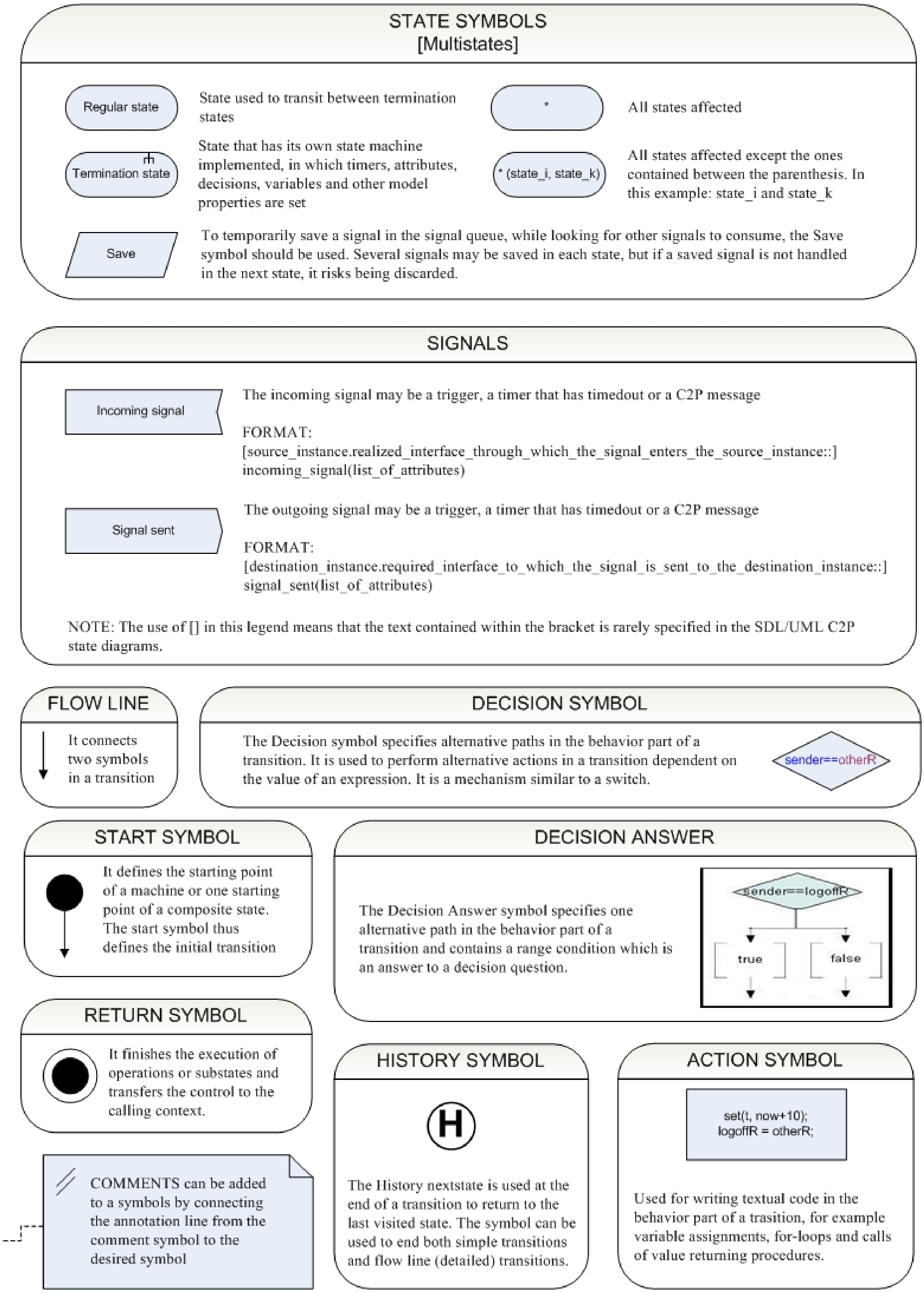
Annex A (informative): C2P state machines

The C2P state machines have been specified thanks to SDL/UML notation. Please refer to the legend depicted in figure A.1 in order to interpret the following C2P SDL/UML state machines. See [i.8] for a more exhaustive explanation of UML.

The states, counters and timers used in the C2P state machines have been introduced in clause 5.

No expected behaviour is specified for any possible transition not covered within these diagrams.

Table A.1: Brief explanation of SDL/UML symbols used in the C2P state machines



STATE SYMBOLS
[Multistates]

Regular state: State used to transit between termination states

Termination state: State that has its own state machine implemented, in which timers, attributes, decisions, variables and other model properties are set

*: All states affected

* (state_i, state_k): All states affected except the ones contained between the parenthesis. In this example: state_i and state_k

Save: To temporarily save a signal in the signal queue, while looking for other signals to consume, the Save symbol should be used. Several signals may be saved in each state, but if a saved signal is not handled in the next state, it risks being discarded.

SIGNALS

Incoming signal: The incoming signal may be a trigger, a timer that has timedout or a C2P message

FORMAT:
[source_instance.realized_interface_through_which_the_signal_enters_the_source_instance::]
incoming_signal(list_of_attributes)

Signal sent: The outgoing signal may be a trigger, a timer that has timedout or a C2P message

FORMAT:
[destination_instance.required_interface_to_which_the_signal_is_sent_to_the_destination_instance::]
signal_sent(list_of_attributes)

NOTE: The use of [] in this legend means that the text contained within the bracket is rarely specified in the SDL/UML C2P state diagrams.

FLOW LINE

It connects two symbols in a transition

DECISION SYMBOL

The Decision symbol specifies alternative paths in the behavior part of a transition. It is used to perform alternative actions in a transition dependent on the value of an expression. It is a mechanism similar to a switch.

START SYMBOL

It defines the starting point of a machine or one starting point of a composite state. The start symbol thus defines the initial transition

DECISION ANSWER

The Decision Answer symbol specifies one alternative path in the behavior part of a transition and contains a range condition which is an answer to a decision question.

RETURN SYMBOL

It finishes the execution of operations or substates and transfers the control to the calling context.

COMMENTS can be added to a symbols by connecting the annotation line from the comment symbol to the desired symbol

HISTORY SYMBOL

The History nextstate is used at the end of a transition to return to the last visited state. The symbol can be used to end both simple transitions and flow line (detailed) transitions.

ACTION SYMBOL

Used for writing textual code in the behavior part of a trasition, for example variable assignments, for-loops and calls of value returning procedures.

A.1 RCST PER-CONNECTION state machine

The RCST PER-CONNECTION C2P behaviour should conform to the UML state machine specification contained in an Adobe Portable Document Format™ file (RCST_UMLstateMachine.pdf contained in archive ts_102602v010101p0.zip) which accompanies the present document.

As shown in the C2P procedures, the RCST behaviour is totally equivalent to the RSGW behaviour. Therefore the RCST state machine also represents the RSGW.

A.2 NCC PER-CONNECTION state machine

The NCC PER-CONNECTION C2P behaviour should conform to the UML state machine specification contained in an Adobe Portable Document Format™ file (NCC_UMLstateMachine.pdf contained in archive ts_102602v010101p0.zip) which accompanies the present document.

The NCC state machine also covers the Transparent Gateway behaviour (identified by the STAR flag).

A.3 RCST PER-RC state machine

The RCST PER-RC C2P behaviour should conform to the UML state machine specification contained in an Adobe Portable Document Format™ file (RCST - RC modify.pdf contained in archive ts_102602v010101p0.zip) which accompanies the present document.

As shown in the C2P procedures, the RSGW behaviour is totally equivalent to the RCST behaviour; therefore the RCST state machine is also representative for the RSGW.

A.4 NCC PER-RC state machine

The NCC PER-RC C2P behaviour PER-RC should conform to the UML state machine specification contained in an Adobe Portable Document Format™ file (NCC - RC Modify.pdf contained in archive ts_102602v010101p0.zip) which accompanies the present document.

The NCC state machine also covers the Transparent Gateway behaviour.

A.5 Example of the mapping between C2P UML, PER-ST and PER-CONNECTION state machines

This annex shows how the connection control procedure "RCST/RSGW initiated RC modify" (see clause 6.7.1.1) is derived from the C2P SDL/UML state machine to the "conceptual PER-ST and PER-CONNECTION state machines included in clause 5.

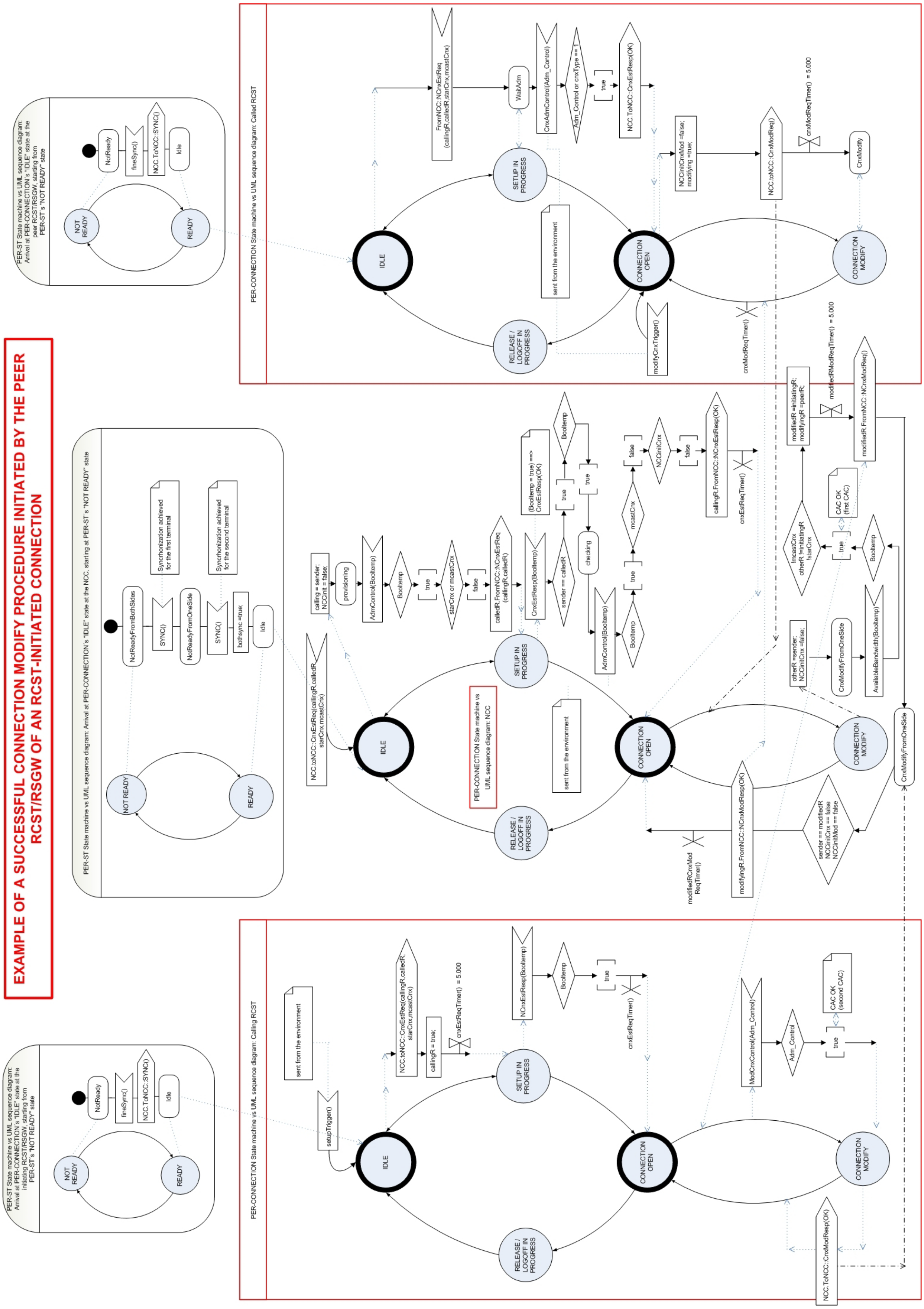

EXAMPLE OF A SUCCESSFUL CONNECTION MODIFY PROCEDURE INITIATED BY THE PEER RCST/RSGW OF AN RCST-INITIATED CONNECTION
PER-ST State machine vs UML sequence diagram: Arrival at PER-CONNECTION's "IDLE" state at the peer RCST/RSGW, starting from PER-ST's "NOT READY" state
NOT READY
READY
NotReady
fineSync()
NCC ToNCC::SYNC()
Idle
PER-CONNECTION State machine vs UML sequence diagram: Called RCST
IDLE
SETUP IN PROGRESS
CONNECTION OPEN
CONNECTION MODIFY
RELEASE / LOGOFF IN PROGRESS
FromNCC::NCnxEstReq (callingR,calledR,starCnx,mcastCnx)
WaitAdm
CnxAdmControl(Adm_Control)
Adm_Control or cnxType == 1
true
NCC ToNCC::CnxEstResp(OK)
sent from the environment
NCCinitCnxMod =false; modifying =true;
NCC.toNCC::CnxModReq()
cnxModReqTimer() = 5.000
CnxModify
modifyCnxTrigger()
cnxModReqTimer()
PER-ST State machine vs UML sequence diagram: Arrival at PER-CONNECTION's "IDLE" state at the NCC, starting at PER-ST's "NOT READY" state
NotReadyFromBothSides
SYNC()
NotReadyFromOneSide
Synchronization achieved for the first terminal
Synchronization achieved for the second terminal
bothsync =true;
Idle
NCC.toNCC::CnxEstReq(callingR,calledR, starCnx,mcastCnx)
calling = sender; NCCinit = false;
provisioning
AdmControl(Booltemp)
Booltemp
true
starCnx or mcastCnx
false
calledR FromNCC::NCnxEstReq (callingR,calledR)
CnxEstResp(Booltemp)
sender == calledR
checking
(Booltemp = true) ==> CnxEstResp(OK)
Booltemp
true
AdmControl(Booltemp)
Booltemp
mcastCnx
false
NCCinitCnx
callingR FromNCC::NCnxEstResp(OK)
cnxEstReqTimer()
PER-CONNECTION State machine vs UML sequence diagram: NCC
sent from the environment
modifiedR =initiatingR; modifyingR =peerR;
modifiedRModReqTimer() = 5.000
modifiedR FromNCC::NCnxModReq()
!mcastCnx otherR !=initiatingR !starCnx
CAC OK (first CAC)
otherR =sender; NCCinitCnx =false;
CnxModifyFromOneSide
AvailableBandwidth(Booltemp)
CnxModifyFromOneSide
modifyingR FromNCC::NCnxModResp(OK)
sender == modifiedR NCCinitCnx == false NCCinitMod == false
modifiedRCnxMod ReqTimer()
PER-ST State machine vs UML sequence diagram: Arrival at PER-CONNECTION's "IDLE" state at the initiating RCST/RSGW, starting from PER-ST's "NOT READY" state
PER-CONNECTION State machine vs UML sequence diagram: Calling RCST
sent from the environment
setupTrigger()
NCC.toNCC::CnxEstReq(callingR,calledR, starCnx,mcastCnx)
callingR = true;
cnxEstReqTimer() = 5.000
NCnxEstResp(Booltemp)
cnxEstReqTimer()
ModCnxControl(Adm_Control)
Adm_Control
CAC OK (second CAC)
NCC ToNCC::CnxModResp(OK)

Figure A.1: Example of C2P UML diagram

Annex B (informative): C2P scenarios

This annex presents examples of C2P messages exchanges for four basic C2P scenarios:

- Scenario 1, for point-to-point bi-directional connections with ATM format.
- Scenario 2, for point-to-point bi-directional connections with MPEG2-TS format.
- Scenario 3, for point-to-multipoint uni-directional connection with ATM format.
- Scenario 4, for point-to-multipoint uni-directional connection with MPEG2-TS format.

C2P Scenario 1 (figure B.1) and C2P Scenario 2 (figure B.2) are representative of mesh communication between two RCSTs in a regenerative network reference scenario. They both describe examples of connection establishment for video conferencing at 384 kbps video rate between two video conference equipments, installed behind the two RCSTs. With the addition of all layer 2 and 3 headers, the bandwidth required for this video conference could be, for example, as high as 464 kbps. The connection is of real time type (i.e. C2P CoS = RT) with SDR = PDR = 464 kbps, mapped to CRA.

The only difference between the two C2P scenarios is that Scenario 1 assumes the ATM format, while Scenario 2 assumes the MPEG2-TS format. With the assumed network reference scenario (mesh regenerative), the formats apply to both return direction and forward direction of the connection. The two formats require different Stream Identifiers (VCCs for Scenario 1, PIDs for Scenario 2).

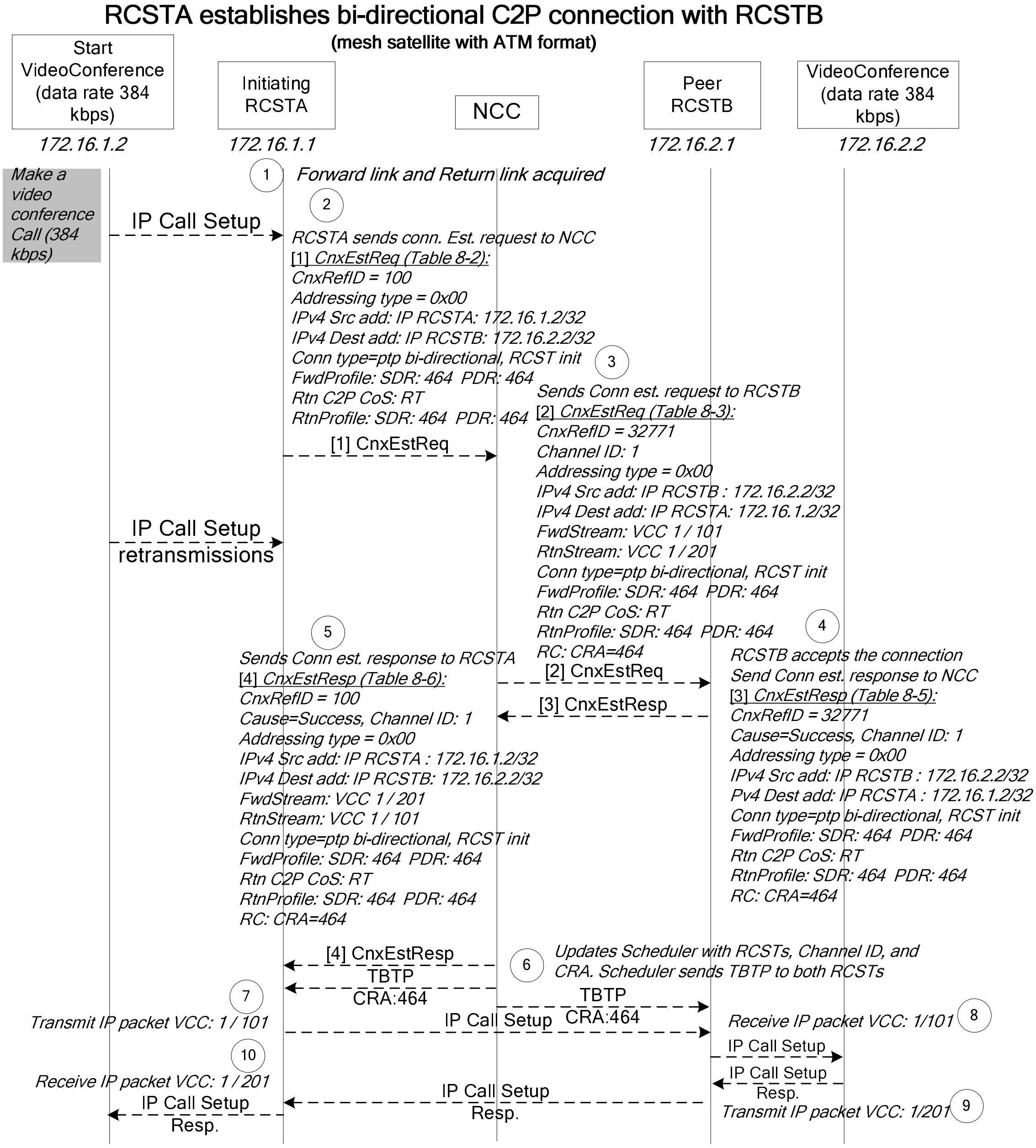

RCSTA establishes bi-directional C2P connection with RCSTB
(mesh satellite with ATM format)
Start VideoConference (data rate 384 kbps)
Initiating RCSTA
NCC
Peer RCSTB
VideoConference (data rate 384 kbps)
172.16.1.2
172.16.1.1
172.16.2.1
172.16.2.2
Make a video conference Call (384 kbps)
1
Forward link and Return link acquired
2
IP Call Setup
RCSTA sends conn. Est. request to NCC
[1] CnxEstReq (Table 8-2):
CnxRefID = 100
Addressing type = 0x00
IPv4 Src add: IP RCSTA: 172.16.1.2/32
IPv4 Dest add: IP RCSTB: 172.16.2.2/32
Conn type=ptp bi-directional, RCST init
FwdProfile: SDR: 464 PDR: 464
Rtn C2P CoS: RT
RtnProfile: SDR: 464 PDR: 464
3
Sends Conn est. request to RCSTB
[2] CnxEstReq (Table 8-3):
CnxRefID = 32771
Channel ID: 1
Addressing type = 0x00
IPv4 Src add: IP RCSTB : 172.16.2.2/32
IPv4 Dest add: IP RCSTA: 172.16.1.2/32
FwdStream: VCC 1 / 101
RtnStream: VCC 1 / 201
Conn type=ptp bi-directional, RCST init
FwdProfile: SDR: 464 PDR: 464
Rtn C2P CoS: RT
RtnProfile: SDR: 464 PDR: 464
RC: CRA=464
[1] CnxEstReq
IP Call Setup retransmissions
5
Sends Conn est. response to RCSTA
[4] CnxEstResp (Table 8-6):
CnxRefID = 100
Cause=Success, Channel ID: 1
Addressing type = 0x00
IPv4 Src add: IP RCSTA : 172.16.1.2/32
IPv4 Dest add: IP RCSTB: 172.16.2.2/32
FwdStream: VCC 1 / 201
RtnStream: VCC 1 / 101
Conn type=ptp bi-directional, RCST init
FwdProfile: SDR: 464 PDR: 464
Rtn C2P CoS: RT
RtnProfile: SDR: 464 PDR: 464
RC: CRA=464
4
RCSTB accepts the connection
Send Conn est. response to NCC
[3] CnxEstResp (Table 8-5):
CnxRefID = 32771
Cause=Success, Channel ID: 1
Addressing type = 0x00
IPv4 Src add: IP RCSTB : 172.16.2.2/32
Pv4 Dest add: IP RCSTA : 172.16.1.2/32
Conn type=ptp bi-directional, RCST init
FwdProfile: SDR: 464 PDR: 464
Rtn C2P CoS: RT
RtnProfile: SDR: 464 PDR: 464
RC: CRA=464
[2] CnxEstReq
[3] CnxEstResp
[4] CnxEstResp
6
Updates Scheduler with RCSTs, Channel ID, and CRA. Scheduler sends TBTP to both RCSTs
TBTP
CRA:464
TBTP
CRA:464
7
Transmit IP packet VCC: 1 / 101
IP Call Setup
8
Receive IP packet VCC: 1/101
IP Call Setup
10
Receive IP packet VCC: 1 / 201
IP Call Setup Resp.
IP Call Setup Resp.
IP Call Setup Resp.
9
Transmit IP packet VCC: 1/201

Figure B.1: Point-to-point bi-directional connections between two RCSTs (ATM format)

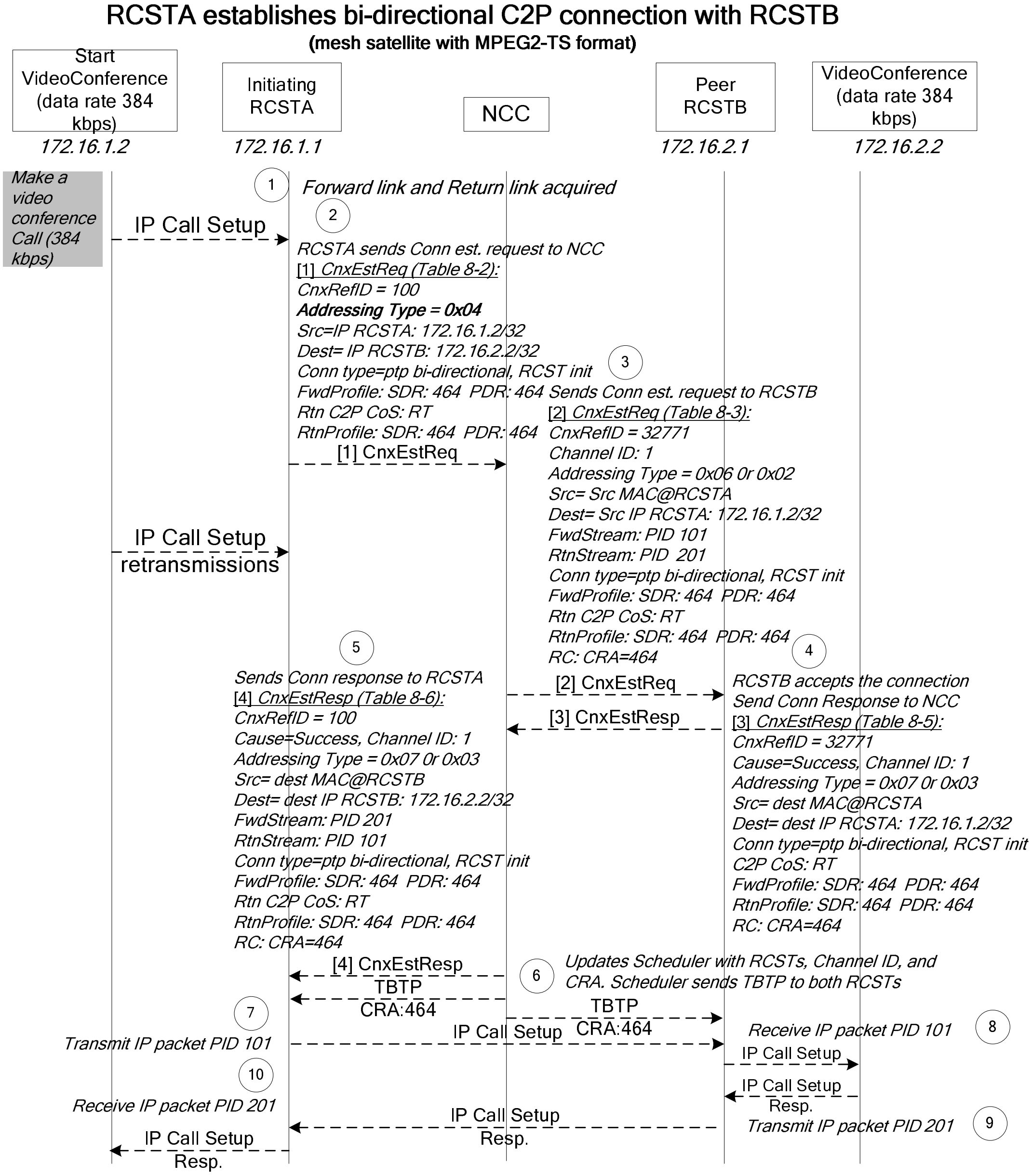

RCSTA establishes bi-directional C2P connection with RCSTB
(mesh satellite with MPEG2-TS format)
Start VideoConference (data rate 384 kbps)
172.16.1.2
Initiating RCSTA
172.16.1.1
NCC
Peer RCSTB
172.16.2.1
VideoConference (data rate 384 kbps)
172.16.2.2
Make a video conference Call (384 kbps)
1
Forward link and Return link acquired
2
IP Call Setup
RCSTA sends Conn est. request to NCC
[1] CnxEstReq (Table 8-2):
CnxRefID = 100
Addressing Type = 0x04
Src=IP RCSTA: 172.16.1.2/32
Dest= IP RCSTB: 172.16.2.2/32
Conn type=ptp bi-directional, RCST init
FwdProfile: SDR: 464 PDR: 464
Rtn C2P CoS: RT
RtnProfile: SDR: 464 PDR: 464
[1] CnxEstReq
3
Sends Conn est. request to RCSTB
[2] CnxEstReq (Table 8-3):
CnxRefID = 32771
Channel ID: 1
Addressing Type = 0x06 0r 0x02
Src= Src MAC@RCSTA
Dest= Src IP RCSTA: 172.16.1.2/32
FwdStream: PID 101
RtnStream: PID 201
Conn type=ptp bi-directional, RCST init
FwdProfile: SDR: 464 PDR: 464
Rtn C2P CoS: RT
RtnProfile: SDR: 464 PDR: 464
RC: CRA=464
IP Call Setup retransmissions
5
Sends Conn response to RCSTA
[4] CnxEstResp (Table 8-6):
CnxRefID = 100
Cause=Success, Channel ID: 1
Addressing Type = 0x07 0r 0x03
Src= dest MAC@RCSTB
Dest= dest IP RCSTB: 172.16.2.2/32
FwdStream: PID 201
RtnStream: PID 101
Conn type=ptp bi-directional, RCST init
FwdProfile: SDR: 464 PDR: 464
Rtn C2P CoS: RT
RtnProfile: SDR: 464 PDR: 464
RC: CRA=464
[2] CnxEstReq
[3] CnxEstResp
4
RCSTB accepts the connection
Send Conn Response to NCC
[3] CnxEstResp (Table 8-5):
CnxRefID = 32771
Cause=Success, Channel ID: 1
Addressing Type = 0x07 0r 0x03
Src= dest MAC@RCSTA
Dest= dest IP RCSTA: 172.16.1.2/32
Conn type=ptp bi-directional, RCST init
C2P CoS: RT
FwdProfile: SDR: 464 PDR: 464
RtnProfile: SDR: 464 PDR: 464
RC: CRA=464
[4] CnxEstResp
TBTP
CRA:464
6
Updates Scheduler with RCSTs, Channel ID, and CRA. Scheduler sends TBTP to both RCSTs
TBTP
CRA:464
7
Transmit IP packet PID 101
IP Call Setup
8
Receive IP packet PID 101
IP Call Setup
10
Receive IP packet PID 201
IP Call Setup Resp.
9
Transmit IP packet PID 201
IP Call Setup Resp.
IP Call Setup Resp.

Figure B.2: Point-to-point bi-directional connections between two RCSTs (MPEG2-TS format)

C2P Scenario 3 (figure B.3) and C2P Scenario 4 (figure B.4) are representative for the establishment of a mesh multicast connection in a regenerative/star network reference scenario, from one RCST (RCST A) to many RCSTs (only one illustrated - RCST B). The multicast source application is installed on a PC behind RCST A. It sends traffic at 512 kbps to multicast listener application installed on a PC behind the RCST B. The connection is of critical data type (i.e. C2P CoS = CD) with SDR = PDR = 512 kbps, mapped to RBDC.

The differences between C2P Scenario 3 and C2P Scenario 4 stem from the use of different formats (ATM for Scenario 3, MPEG2-TS for Scenario 4), which require different Return Stream Identifier (VCCs for Scenario 3, PIDs for Scenario 4).

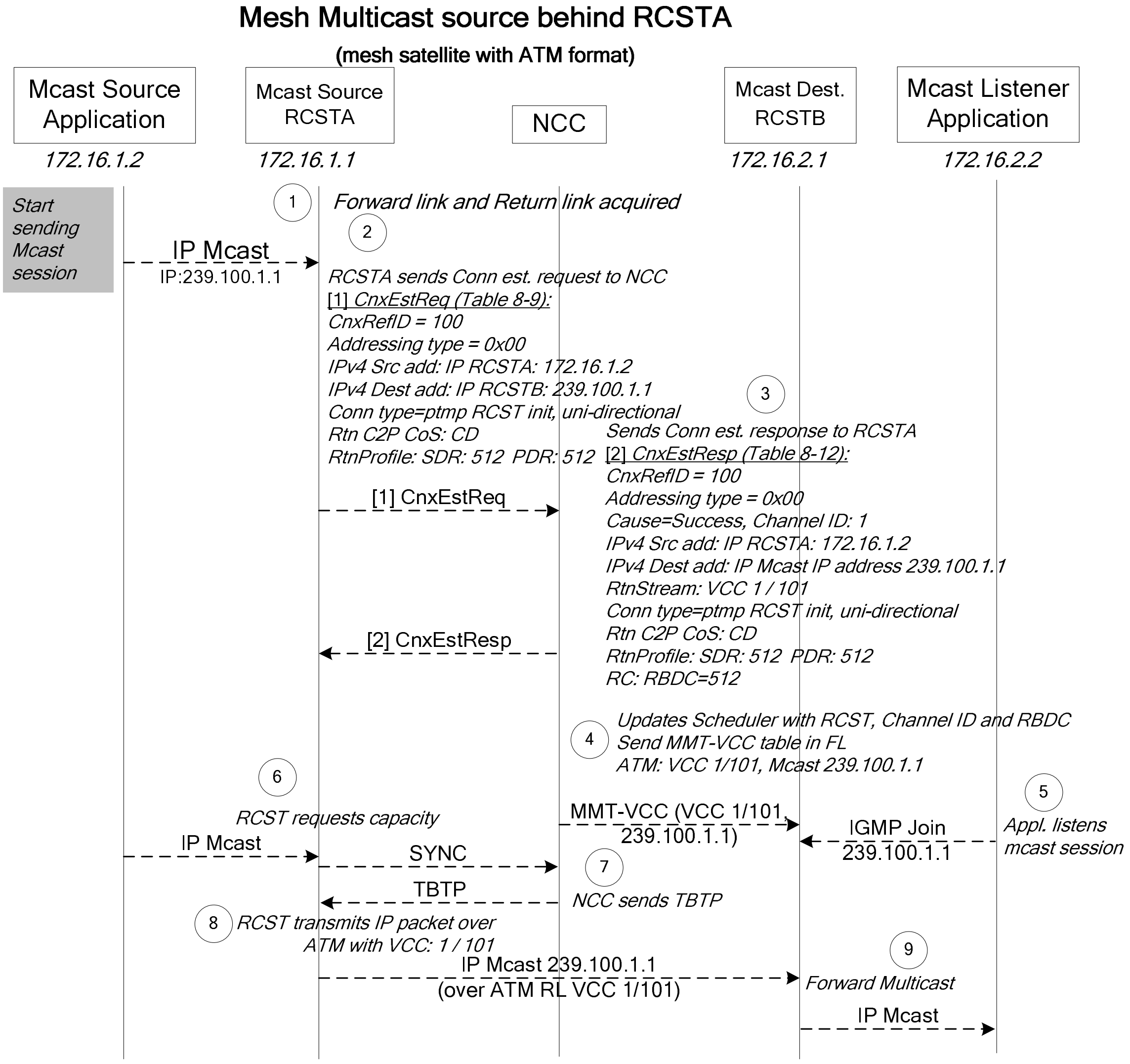

Mesh Multicast source behind RCSTA
(mesh satellite with ATM format)
Mcast Source Application
172.16.1.2
Mcast Source RCSTA
172.16.1.1
NCC
Mcast Dest. RCSTB
172.16.2.1
Mcast Listener Application
172.16.2.2
Start sending Mcast session
1
Forward link and Return link acquired
2
IP Mcast
IP:239.100.1.1
RCSTA sends Conn est. request to NCC
[1] CnxEstReq (Table 8-9):
CnxRefID = 100
Addressing type = 0x00
IPv4 Src add: IP RCSTA: 172.16.1.2
IPv4 Dest add: IP RCSTB: 239.100.1.1
Conn type=ptmp RCST init, uni-directional
Rtn C2P CoS: CD
RtnProfile: SDR: 512 PDR: 512
3
Sends Conn est. response to RCSTA
[2] CnxEstResp (Table 8-12):
CnxRefID = 100
Addressing type = 0x00
Cause=Success, Channel ID: 1
IPv4 Src add: IP RCSTA: 172.16.1.2
IPv4 Dest add: IP Mcast IP address 239.100.1.1
RtnStream: VCC 1 / 101
Conn type=ptmp RCST init, uni-directional
Rtn C2P CoS: CD
RtnProfile: SDR: 512 PDR: 512
RC: RBDC=512
[1] CnxEstReq
[2] CnxEstResp
4
Updates Scheduler with RCST, Channel ID and RBDC
Send MMT-VCC table in FL
ATM: VCC 1/101, Mcast 239.100.1.1
6
RCST requests capacity
5
MMT-VCC (VCC 1/101, 239.100.1.1)
IGMP Join
239.100.1.1
Appl. listens mcast session
IP Mcast
SYNC
7
TBTP
NCC sends TBTP
8
RCST transmits IP packet over ATM with VCC: 1 / 101
9
IP Mcast 239.100.1.1
(over ATM RL VCC 1/101)
Forward Multicast
IP Mcast

Figure B.3: Point-to-multipoint uni-directional mesh connection (ATM format)

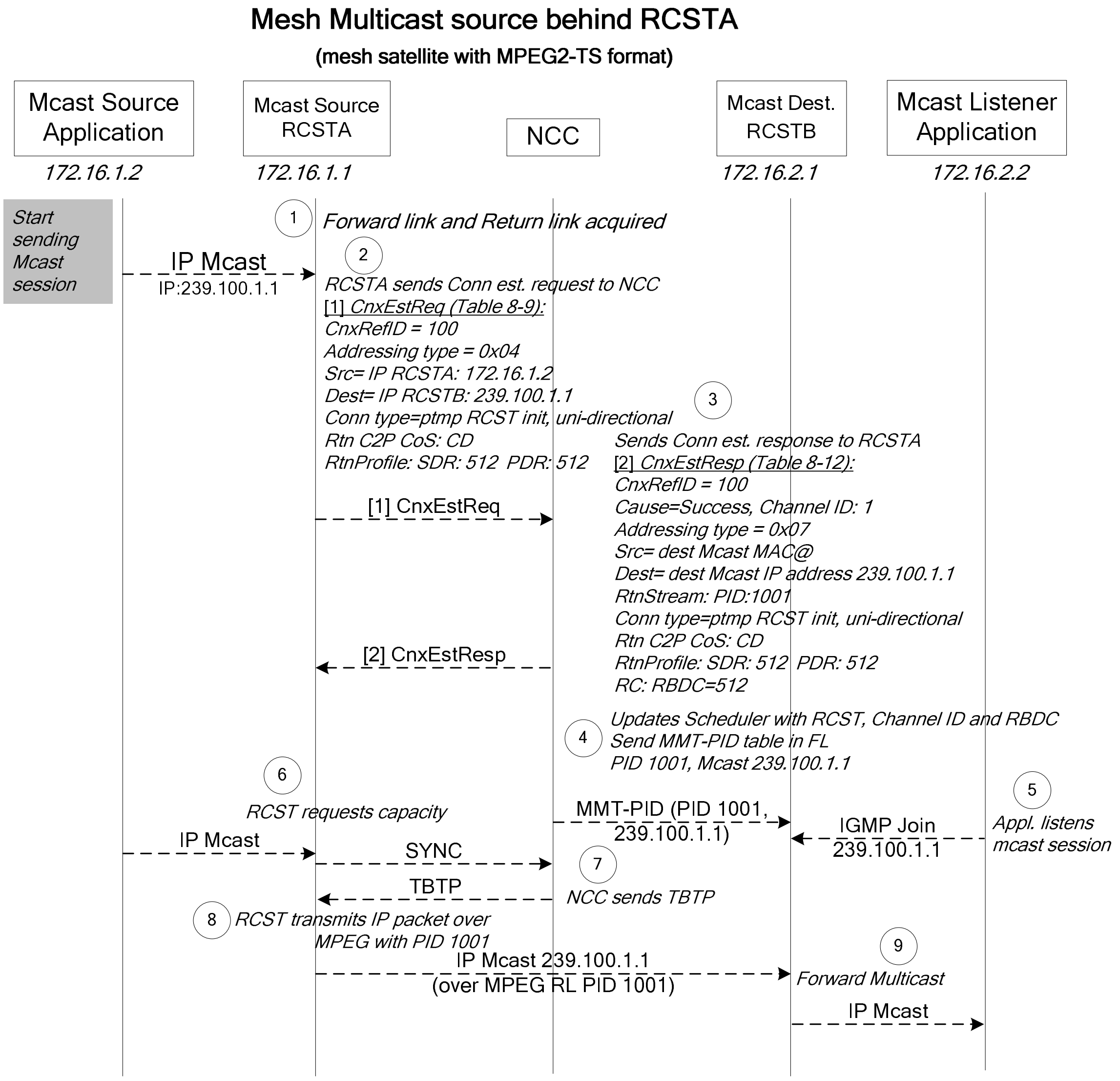

Mesh Multicast source behind RCSTA
(mesh satellite with MPEG2-TS format)
Mcast Source Application
172.16.1.2
Mcast Source RCSTA
172.16.1.1
NCC
Mcast Dest. RCSTB
172.16.2.1
Mcast Listener Application
172.16.2.2
Start sending Mcast session
1
Forward link and Return link acquired
IP Mcast
IP:239.100.1.1
2
RCSTA sends Conn est. request to NCC
[1] CnxEstReq (Table 8-9):
CnxRefID = 100
Addressing type = 0x04
Src= IP RCSTA: 172.16.1.2
Dest= IP RCSTB: 239.100.1.1
Conn type=ptmp RCST init, uni-directional
Rtn C2P CoS: CD
RtnProfile: SDR: 512 PDR: 512
3
Sends Conn est. response to RCSTA
[2] CnxEstResp (Table 8-12):
CnxRefID = 100
Cause=Success, Channel ID: 1
Addressing type = 0x07
Src= dest Mcast MAC@
Dest= dest Mcast IP address 239.100.1.1
RtnStream: PID:1001
Conn type=ptmp RCST init, uni-directional
Rtn C2P CoS: CD
RtnProfile: SDR: 512 PDR: 512
RC: RBDC=512
[1] CnxEstReq
[2] CnxEstResp
4
Updates Scheduler with RCST, Channel ID and RBDC
Send MMT-PID table in FL
PID 1001, Mcast 239.100.1.1
6
RCST requests capacity
5
Appl. listens mcast session
MMT-PID (PID 1001, 239.100.1.1)
IGMP Join 239.100.1.1
IP Mcast
SYNC
7
NCC sends TBTP
TBTP
8
RCST transmits IP packet over MPEG with PID 1001
IP Mcast 239.100.1.1
(over MPEG RL PID 1001)
9
Forward Multicast
IP Mcast

Figure B.4: Point-to-multipoint uni-directional mesh connection (MPEG2-TS format)

Annex C (informative): Examples of C2P exception procedures

Exception procedures involve C2P procedures (setup/release/modification) in which a C2P message gets lost and the C2P message is sent, up to a maximum number of retries, at the expiry of NCC or RCST/RSGW timers,. This way, exception procedures can lead to success or to failure according to the original intention of the procedure.

The exception procedures included in this annex complement clause 6 and the scope of C2P UML/SDL state machine provided in annex A.

NOTE: Even though a few of the procedures included in this annex depict logoff decisions made by an RCST or by the NCC, the logoff of terminals is system specific. In many scenarios, and as a result of a critical C2P message communication failure, there are logoff decisions taken at the NCC or logoff decisions initiated by the RCST. Such logoff decisions are conditional and not forced, as they are not part of the connection protocol itself. Therefore, there might be other considerations for maintaining the RCST logged into the system, despite the C2P mal-functioning.

C.1 Point-to-point connection establishment procedures

C.1.1 RCST/RSGW initiated connections

C.1.1.1 Successful set up - Initiating RCST/RSGW retry

Figure C.1 shows a successful setup procedure involving a retransmission of the setup request message from the initiating RCST/RSGW. Steps [1] to [4] are similar to those for the successful case (clause 6.1.1.1), except that in this case the connection establishment request sent by the NCC [4] does not reach the initiating party and two new steps are added, [5] and [6].

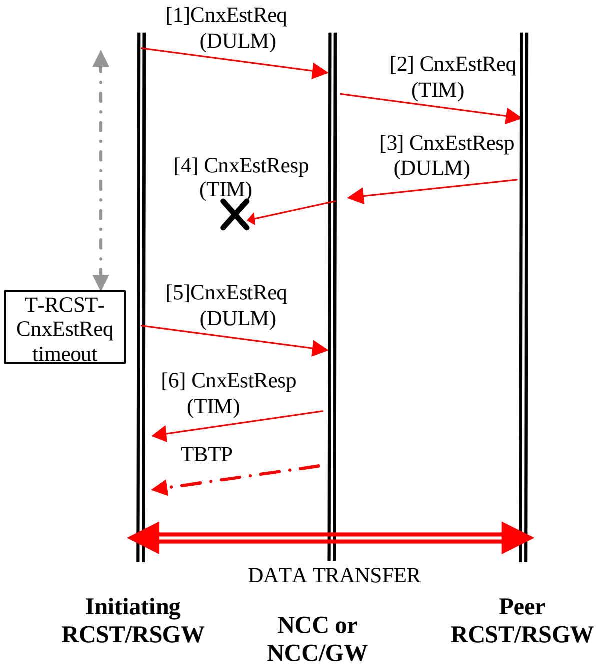


Figure C.1: RCST/RSGW initiated Point-to-Point successful connection establishment: Initiating RCST/RSGW retry

[5] If timer T-RCST_CnxEstReq expires and still no answer from the NCC is received at the Initiating RCST/RSGW, the initiating RCST/RSGW retransmits the setup request message to the NCC with appropriate retry count and the connection reference ID (from this point on referred simply as "Identifier") used for this connection.

[6] The NCC recognizes that the Connection Establishment Request is for a connection already registered, and resends the connection establishment response. The NCC's response manages to arrive at the Initiating RCST/RSGW. Upon reception of the Connection Establishment Response, the initiating RCST/RSGW stops timer T-RCST_CnxEstReq. Data transfer will start upon reception of an appropriate TBTP.

In the case of transparent star scenarios, steps [2] and [3] in figure C.1 will be omitted.

C.1.1.2 Successful set up - NCC retry

Figure C.2 shows a successful setup procedure involving the retransmission of the NCC's Connection Establishment Request to the peer RCST/RSGW. Steps [1] to [3] are similar to those for the successful case (clause 6.1.1.1), and three new steps are added, [4], [5] and [6].

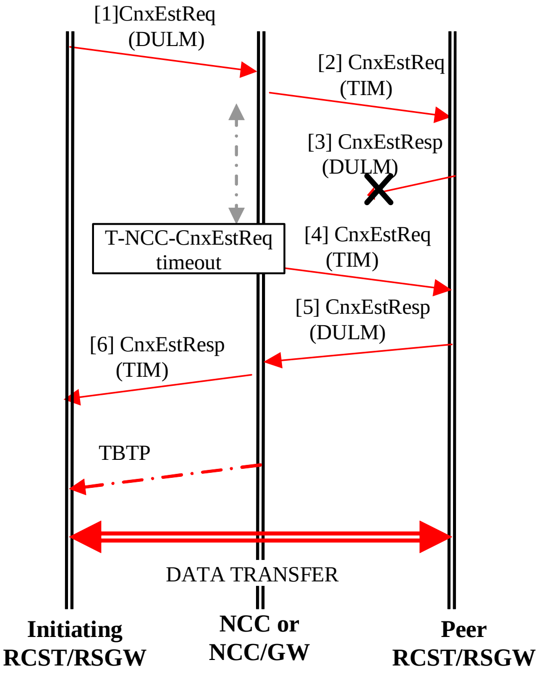


Figure C.2: RCST/RSGW initiated Point-to-Point successful connection establishment: NCC retry

[4] If timer T-NCC_CnxEstReq expires and still no answer from the Peer RCST/RSGW is received at the NCC, the NCC retransmits the setup request message to the NCC with appropriate retry count and Identifier used for this connection.

[5] The peer RCST/RSGW recognizes that the request is for a connection already open, and resends the Connection Establishment Response. Upon reception of the response from the peer RCST/RSGW, the NCC stops timer T-NCC_CnxEstReq.

[6] The NCC sends the Connection Establishment Response to the Initiating RCST/RSGW. Upon reception of the message, the initiating RCST/RSGW stops timer T-RCST_CnxEstReq. Data transfer will start upon reception of an appropriate TBTP.

In the case of transparent star scenarios, where the connection set up takes place between an RCST and the NCC/GW, steps [2] to [5] in figure C.2 should be omitted and step [6] should be renumbered as [2].

C.1.1.3 Unsuccessful set up - Peer RCST/RSGW responses lost

Figure C.3 shows an unsuccessful setup procedure due to a peer RCST/RSGW which is not responding, from the point of view of the NCC i.e. the peer party Connection Establishment Responses messages get lost on their way to the NCC. Steps [1] and [2] are similar to the successful establishment procedure described in clause 6.1.1.1.

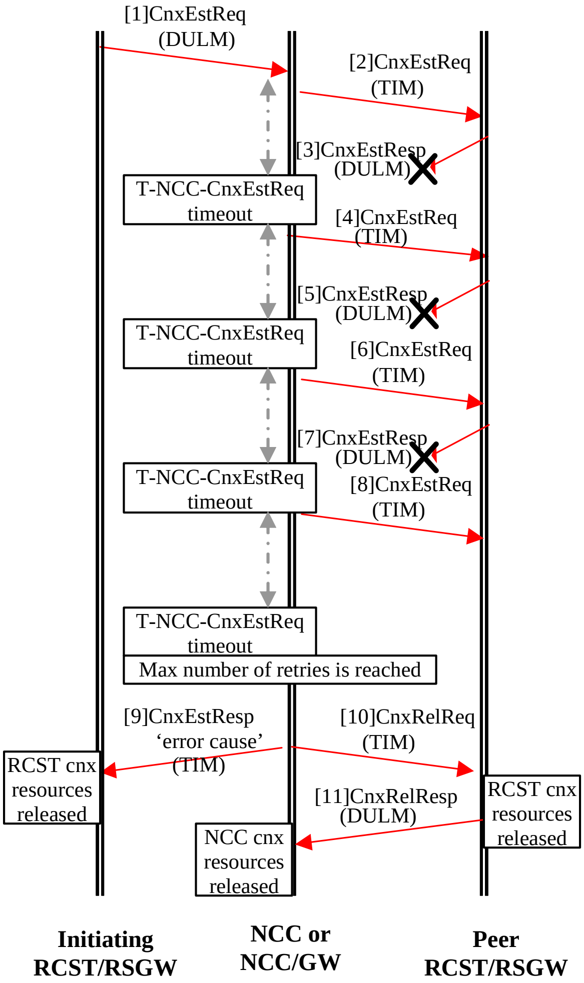


Figure C.3: RCST/RSGW initiated Point-to-Point unsuccessful connection establishment: Peer RCST/RSGW no answer

[3][5][7]

The Connection Establishment Response message from the peer RCST/RSGW does not reach the NCC. Timer T-NCC_CnxEstReq expires.

[4][6][8]

The NCC retransmits the setup request message to the peer RCST/RSGW with appropriate retry count and Identifier. NCC may retry up to a maximum of N-NCC_CnxEstReq retries.

[9][10]

Maximum number of retries is reached. A Connection Establishment Response reject is sent to the initiating RCST/RSGW, with the error Cause IE 'no answer'. Upon reception of the negative Connection Establishment Response, the Initiating RCST/RSGW releases its resources for the connection and it should wait up to the Inhibition timer expires to send another request (see procedure C.1.1.1). To avoid inconsistency problems and due to the lack of information about the peer side, the NCC sends a connection release request to the peer RCST/RSGW.

[11] Upon reception of the release request, the Peer RCST/RSGW sends a Connection Release Response to the NCC and releases its resources for the connection. Upon reception of the release response, the NCC releases its resources for the connection.

If peer RCST's connection establishment responses are lost, it could be expected that the connection release responses will also get lost; however in this procedure the connection release responses manage to arrive at the NCC.

See clause 6.3.1 for further information on release procedures.

This procedure does not apply to transparent star scenarios.

C.1.1.4 Unsuccessful set up - Peer RCST/RSGW does not get any NCC set up message

Figure C.4 shows an unsuccessful set-up procedure due to a peer RCST/RSGW not responding. Step [1] is similar to the successful establishment procedure described in clause 6.1.1.1.

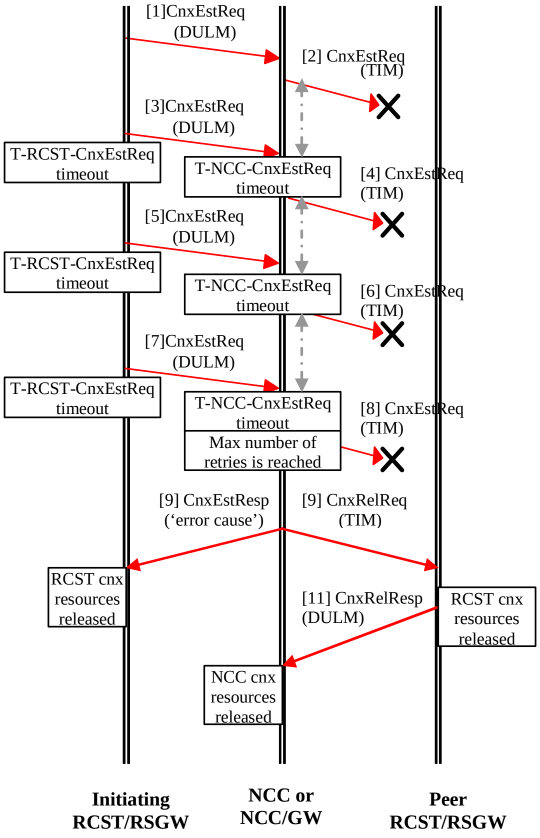


Figure C.4: RCST/RSGW initiated Point-to-Point unsuccessful connection establishment: Peer RCST/RSGW does not get any of the NCC's set-up messages

[2][8]

The Connection Establishment request message from the NCC does not reach the peer party. Timer T-NCC_CnxEstReq expires. The NCC retransmits the setup request message to the peer RCST/RSGW with appropriate retry count and Identifier. The NCC may retry up to a maximum N-NCC_CnxEstReq number of retries.

[9] Maximum number of retries is reached at the NCC. To avoid inconsistency problems and due to the lack of information about the peer side, the NCC will send connection release requests to the peer RCST and a connection establishment response reject to the Initiating RCST. The NCC starts T-NCC_CnxRelReq timer for the peer RCST.

[10] Upon reception of the release response, the NCC stops timer T-NCC_CnxRelReq for the peer RCST and releases the resources for the connection. See release procedures in clause 6.3 for more details.

In the case of a transparent star scenario, the connection set up takes place between an RCST and the NCC/GW, and this procedure does not apply.

C.1.1.5 Unsuccessful set up - Initiating RCST/RSGW retry

Figure C.5 shows an unsuccessful setup procedure involving a retransmission of the setup from the initiating RCST/RSGW. Steps [1] to [3] are similar to the successful establishment procedure described in clause 6.1.1.1.

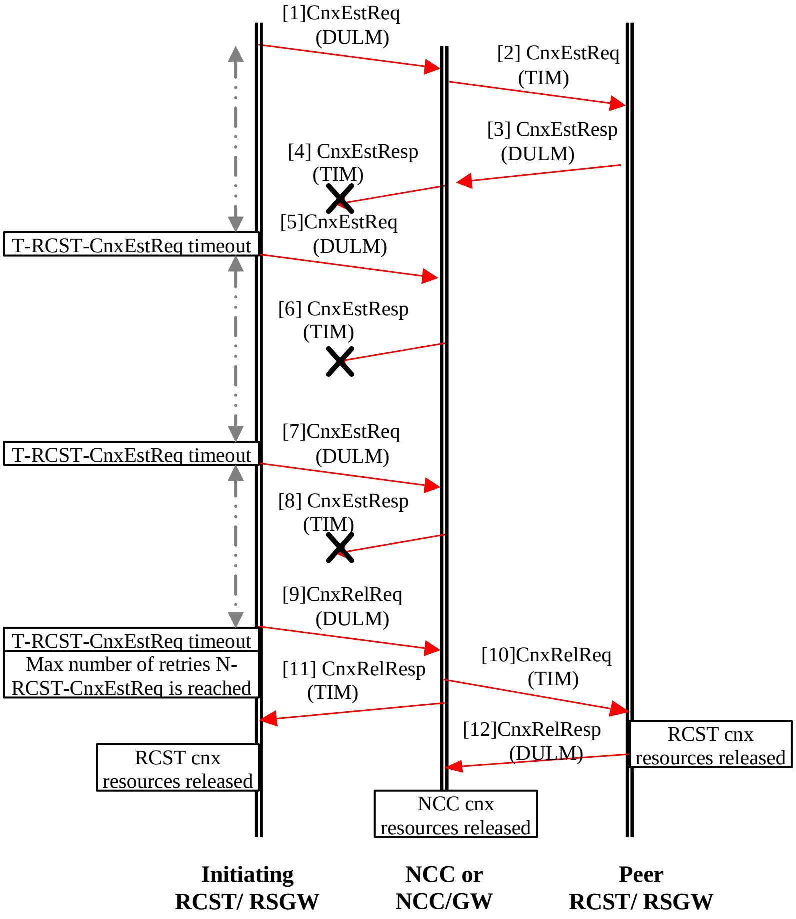


Figure C.5: RCST/RSGW initiated Point-to-Point unsuccessful connection establishment: Initiating RCST/RSGW retry

[4][6][8]

The NCC's connection establishment response does not reach the initiating RCST/RSGW. Timer T-RCST_CnxEstReq expires.

[5][7]

The RCST/RSGW retransmits the connection establishment request to the NCC with appropriate retry count and identifier. Initiating RCST/RSGW may retry up to a maximum of N-RCST_CnxEstReq retries.

[9][10][11][12]

Maximum number of retries is reached. A connection release request is sent to the NCC and the initiating RCST/RSGW starts timer T-RCST_CnxRelReq. See release procedures in clause 6.3 for further information on how resources are released.

In case of a transparent star scenario, where the connection set up takes place between an initiating RCST and the NCC/GW, steps [2] and [3] will be omitted.

C.1.1.6 Unsuccessful set up - NCC no answer

Figure C.6 shows an unsuccessful set up procedure due to an NCC not responding. From the point of view of the initiating RCST/RSGW, this case is equivalent to clause C.1.1.5. This procedure applies to all scenarios (regenerative and transparent).

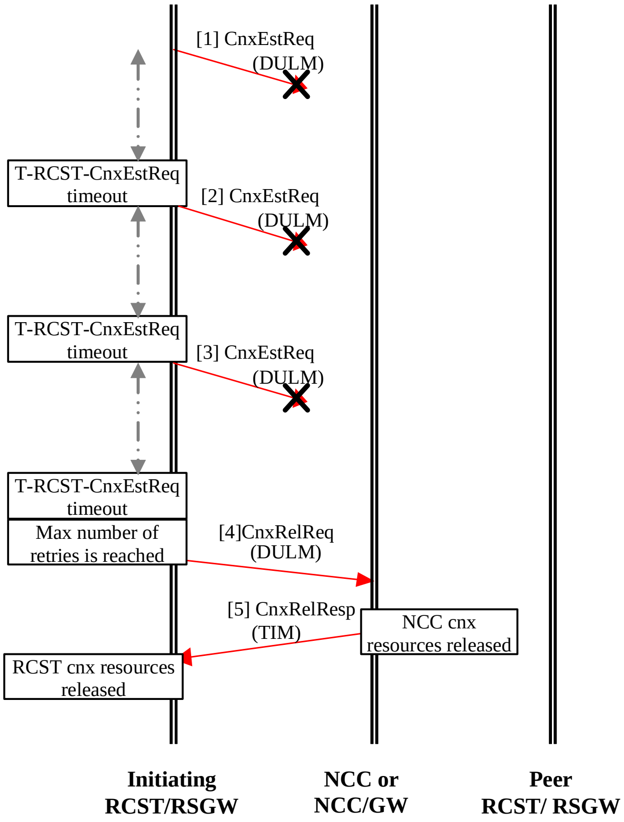


Figure C.6: NCC-initiated Point-to-Point unsuccessful connection establishment: NCC no answer

[1] The Connection Establishment response message from the NCC does not reach the initiating RCST/RSGW. Timer T-RCST_CnxEstReq expires.

[2][3]

The RCST/RSGW retransmits the setup request message to the NCC with appropriate retry count and Identifier. RCST/RSGW may retry up to a maximum of N-RCST_CnxEstReq retries.

[4] As no response is received from the NCC, the RCST/RSGW sends a connection release request. See release procedures in clause 6.3.

This procedure is applicable to all reference scenarios.

C.1.1.7 Unsuccessful set up - NCC reject

Figure C.7 shows an unsuccessful set-up procedure due to the rejection of the establishment's request at the NCC. This procedure applies to all reference scenarios (transparent and regenerative).

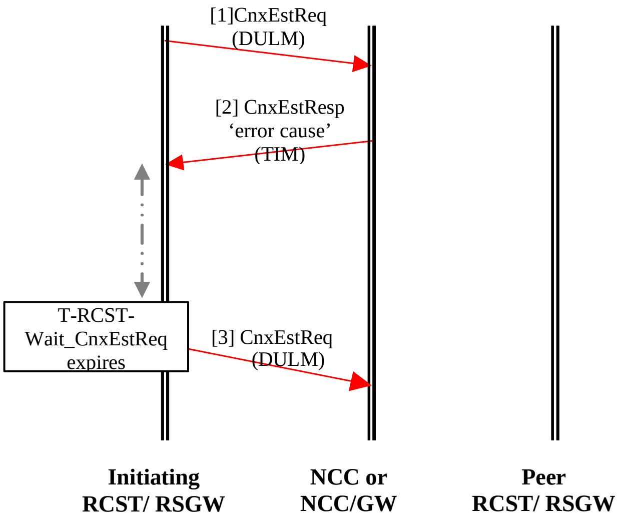


Figure C.7: RCST/RSGW initiated Point-to-Point unsuccessful connection establishment: NCC reject

[1] The initiating RCST/RSGW requests the NCC for a connection setup and starts timer T-RCST_CnxEstReq.

[2] The NCC rejects the connection and sends a connection establishment response with one of the Cause IEs from table 7.10. If the connection triggering set up is still active, the initiating RCST/RSGW should activate timer T-RCST-Wait_CnxEstReq.

[3] When timer T-RCST_Wait_CnxEstReq expires, the initiating RCST/RSGW can send another connection establishment request with the same characteristics as the one that had been rejected by the NCC in [2].

This procedure is applicable to all reference scenarios.

C.1.1.8 Unsuccessful set up - Peer RCST/RSGW reject

Figure C.8 shows an unsuccessful establishment procedure due to a reject at the peer RCST/RSGW. Steps [1] and [2] are similar to the successful case.

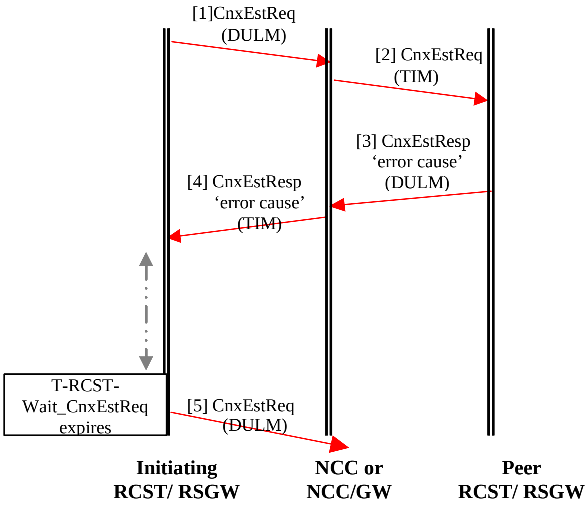


Figure C.8: RCST/RSGW initiated Point-to-Point unsuccessful connection establishment: Peer RCST/RSGW reject

[3] The peer RCST/RSGW rejects the connection by sending a connection establishment response with the appropriate error cause.

[4] The connection reject is send by the NCC to the initiating RCST/RSGW including the error causes. If the connection triggering set up is still active, the initiating RCST/RSGW should activate timer T-RCST-Wait_CnxEstReq.

[5] When timer T-RCST_Wait_CnxEstReq expires, the initiating RCST/RSGW can send another connection establishment request with the same characteristics as the one that had been rejected by the peer RCST/RSGW in [3].

This procedure does not apply to star transparent reference scenarios.

C.1.1.9 Unsuccessful set up - Initiating RCST/RSGW release

Figure C.9 shows an unsuccessful set-up procedure due to a connection release message initiated by the initiating RCST/RSGW in the middle of the connection setup process. Steps [1] to [3] are similar to the successful establishment procedure described in clause 6.1.1.1.

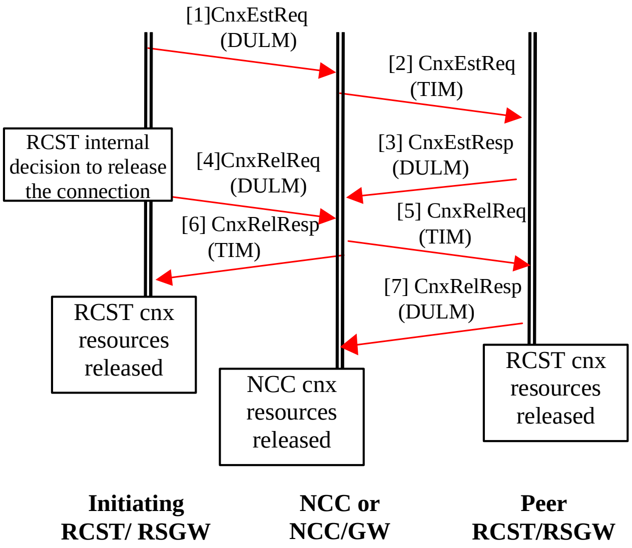


Figure C.9: RCST/RSGW initiated Point-to-Point unsuccessful connection establishment: Initiating RCST/RSGW retry

[4] The initiating RCST/RSGW decides to send a connection release request (e.g. due to an internal trigger).

[5][6]

The NCC, upon reception of the release request, releases the connection resources, and sends a connection release request to the peer RCST/RSGW and a connection release response to the initiating RCST/RSGW.

[7] The peer RCST/RSGW answers with a connection release response.

In case of a transparent star scenario, the connection establishment takes place between an RCST and the NCC/GW, and steps [2], [3], [5] and [7] will be omitted.

C.1.1.10 Unsuccessful set up - NCC release

Figure C.10 represents an unsuccessful connection set-up procedure, in which the NCC decides to release the connection in the middle of the connection establishment procedure. Steps [1] to [3] are similar to the connection establishment procedure described in clause 6.1.1.1.

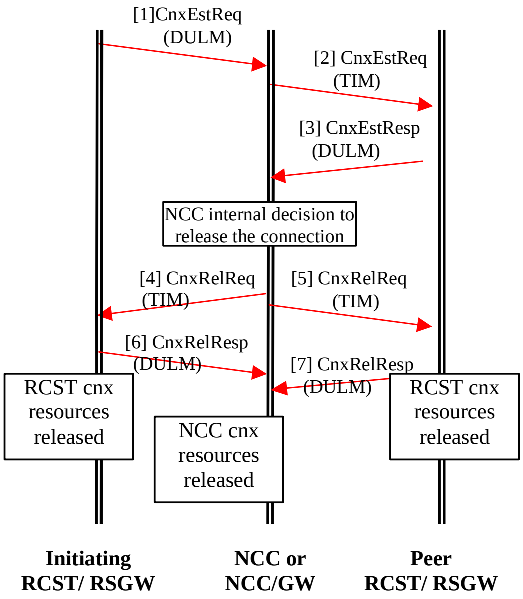


Figure C.10: RCST/RSGW initiated Point-to-Point unsuccessful connection establishment: NCC release

[4][5]

The NCC decides to release the connection in the middle of the establishment procedure (e.g. due to an internal trigger or via a console command) and sends connection release requests to the initiating RCST/RSGW and to the peer RCST/RSGW. The NCC starts timers T-NCC_CnxRelReq for the initiating RCST/RSGW and for the peer RCST/RSGW.

[6][7]

Initiating RCST/RSGW and peer RCST/RSGW reply the NCC by sending each a connection release response. Upon reception of both responses, the NCC stops timers T-NCC_CnxRelReq for the initiating RCST/RSGW and for the peer RCST/RSGW, and it releases its resources for the connection.

In case of a transparent star scenario, the connection establishment takes place between an RCST/RSGW and the NCC/GW, and steps [2], [3], [5] and [7] are omitted.

C.1.1.11 Unsuccessful set up - Peer RCST/RSGW release

Figure C.11 represents an unsuccessful connection set up procedure in which the peer RCST/RSGW decides to release the connection. One message might get lost due to several momentary factors (rain fade, bad antenna pointing, etc) which do not affect the following message, which manages to arrive at the NCC.

Steps [1] and [2] are similar to the connection successful setup.

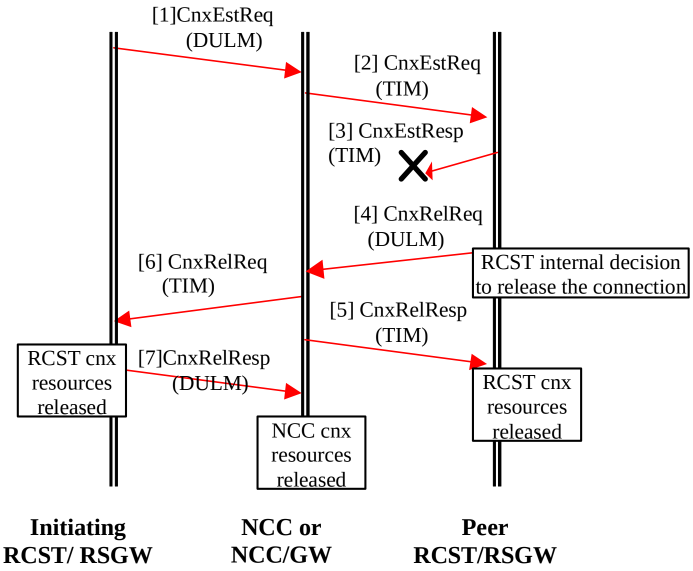


Figure C.11: RCST/RSGW initiated Point-to-Point unsuccessful connection establishment: Peer RCST/RSGW release

[3] Peer RCST/RSGW sends a Connection Establishment Response to the NCC, but the response does not reach the NCC.

[4] Peer RCST/RSGW decides to send a Connection Release Request to the NCC (e.g. via a console command). The request arrives at the NCC. The NCC answers the Peer RCST/RSGW by sending a connection release response to it and the NCC also sends a Connection Release Request to the initiating RCST/RSGW. See release procedures in clause 6.3 for further information on the release of the resources at the RCST/RSGWs and at the NCC.

C.1.2 NCC initiated connections

C.1.2.1 Successful set up - NCC retry

Figure C.12 represents a successful connection set up procedure initiated by the NCC. In this case, one peer RCST/RSGW does not answer and requires several retries from the NCC.

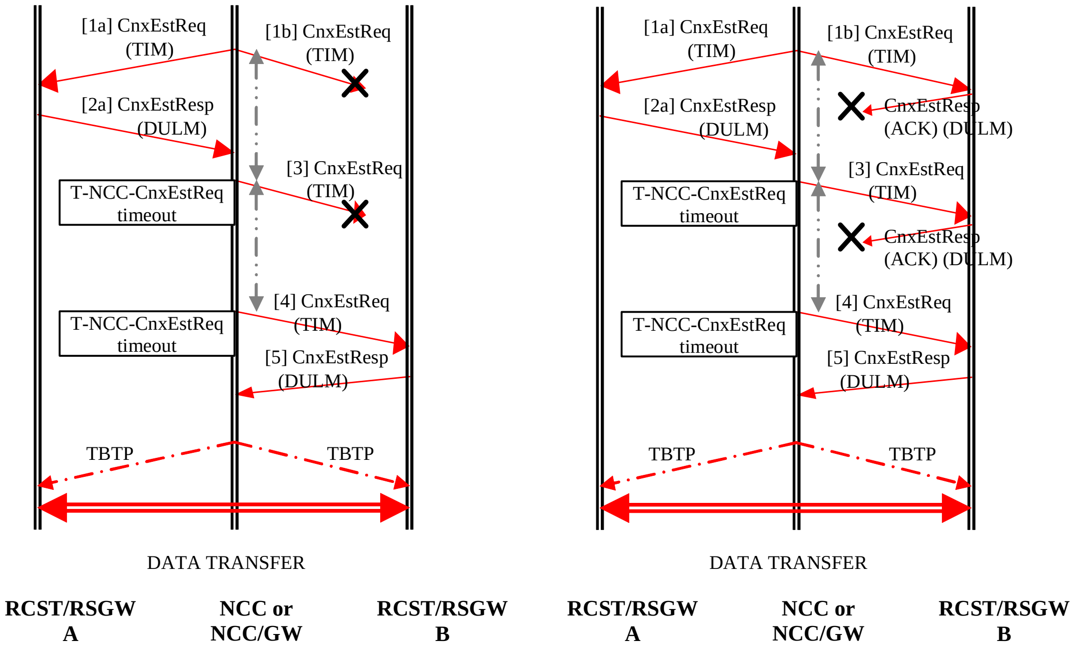


Figure C.12: NCC-initiated Point-to-Point successful connection establishment: NCC retry

[1a][1b]

The NCC sends a Connection Establishment Request to both parties (RCST/RSGW A and RCST/RSGW B) and starts timer T-NCC_CnxEstReq for both parties.

[2a] RCST/RSGW A accepts the connection by sending a positive Connection Establishment Response. Upon reception of this message, the NCC stops timer T-NCC_CnxEstReq for RCST A.

[3][4]

Either because the Connection Establishment Request sent to RCST/RSGW B gets lost (left side of figure C.12) or because RCST/RSGW B's positive Connection Establishment Response gets lost on its way to the NCC (right side of figure C.12), timer T-NCC_CnxEstReq for RCST/RSGW B timeouts and the NCC retransmits the establishment request message to RCST/RSGW B with appropriate retry count and Identifier. The NCC may retry up to a maximum N-NCC_CnxEstReq number of times.

[5] Before reaching the maximum number of retries, the NCC receives RCST/RSGW B's positive Connection Establishment Response message. From this moment the connection is considered open.

In case of a transparent star scenario, where the connection set up takes place between the NCC/GW and RCST/RSGW B, steps [1a], [2] should be omitted.

C.1.2.2 Unsuccessful set-up - One of the peer parties does not answer the NCC's setup requests

Figure C.13 represents an unsuccessful connection set up procedure in which one of the peer RCST/RSGWs does not answer the NCC's setup requests but it manages to answer the NCC's release request. Steps [1a], [1b] and [2a] are similar to the successful setup for one of the sides of the connection.

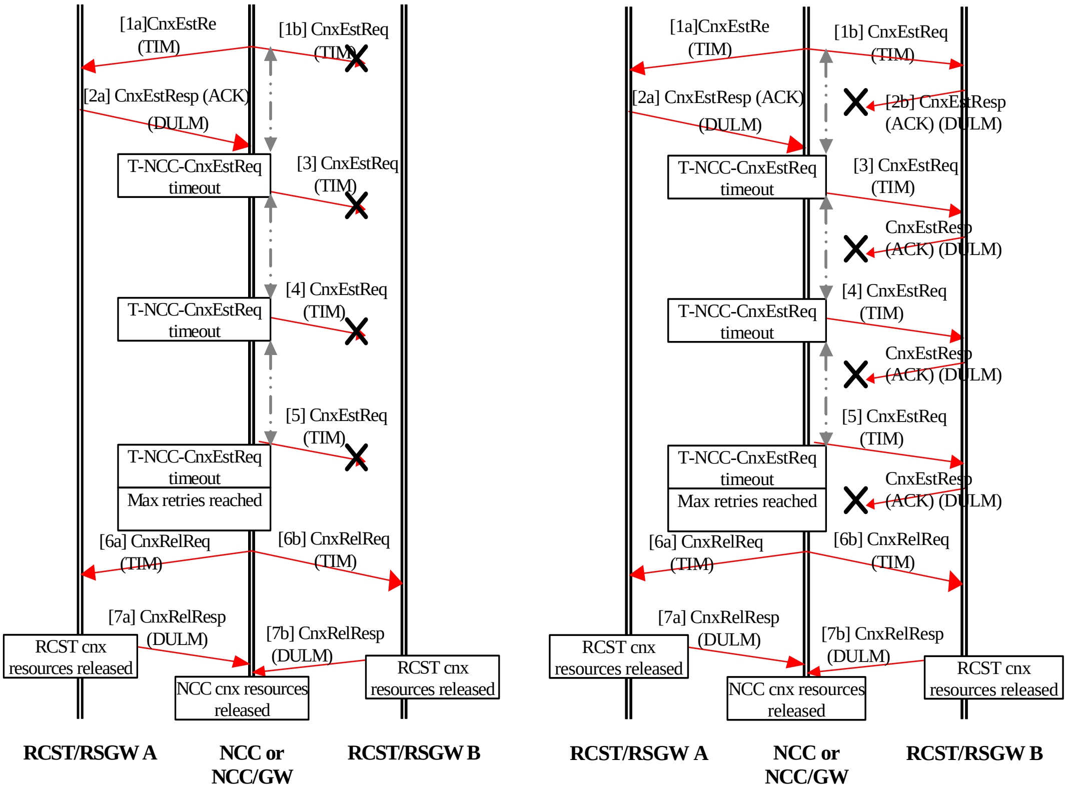


Figure C.13: NCC-initiated Point-to-Point unsuccessful connection establishment: One of the peer parties does not answer the NCC's setup requests

[1a][1b]

The NCC sends a Connection Establishment Request to both parties (RCST/RSGW A and RCST/RSGW B) and starts timer T-NCC_CnxEstReq for both parties.

[2a] RCST/RSGW A accepts the connection by sending a positive Connection Establishment Response. Upon reception of this message, the NCC stops timer T-NCC_CnxEstReq for RCST A.

[3][4][5]

Either because the Connection Establishment Request sent to RCST/RSGW B gets lost (left side of figure C.12) or because RCST/RSGW B's positive Connection Establishment Response gets lost on its way to the NCC (right side of figure C.12), timer T-NCC_CnxEstReq for RCST/RSGW B timeouts and the NCC retransmits the establishment request message to RCST/RSGW B with appropriate retry count and Identifier. The NCC may retry up to a maximum N-NCC_CnxEstReq number of times.

[6] The maximum number of N-NCC_CnxEstReq retries is reached. The NCC sends connection release requests to both sides, and it starts timers T-NCC_CnxRelReq for both parties.

[7] RCST/RSGW A and RCST/RSGW B answer the NCC with connection release responses which manage to arrive at the NCC. Upon reception of each response, the NCC stops timer T-NCC_CnxRelReq for the correspondent RCST/RSGW. See clause 6.3 for further information on the release of the resources at the RCST/RSGWs and at the NCC.

In case of a transparent star scenario, the exception procedure for connection set up takes place between the initiating NCC/GW and RCST/RSGW B, and steps [1a], [2a], [6a] and [7a] towards RCST/RSGW A are not considered.

C.1.2.3 Unsuccessful set-up - One of the peer parties does not answer the NCC's setup requests nor the NCC's release requests

Figure C.14 represents an unsuccessful connection set up procedure involving a peer RCST/RSGW no answer.
Steps [1a][1b] and [2a] are similar to C.1.2.1's steps.

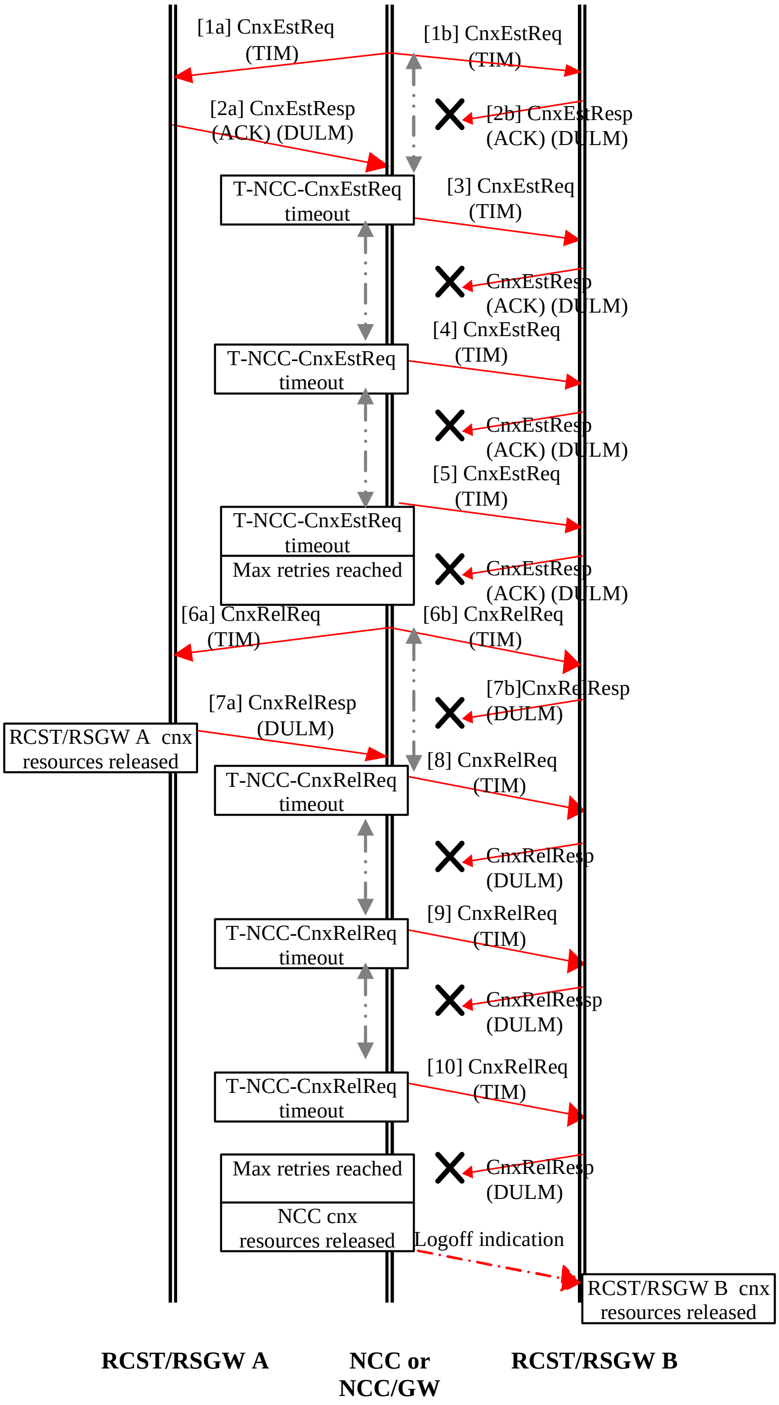


Figure C.14: NCC-initiated Point-to-Point unsuccessful connection establishment: One of the peer parties does not answer the NCC's setup requests nor the NCC's release requests

[2b][3][4][5]

RCST/RSGW B's connection establishment response get lost. Timer T-NCC_CnxEstReq expires for RCST/RSGW B and the NCC retransmits the connection establishment request to RCST/RSGW B with appropriate retry count and Identifier. The procedure is repeated until the NCC reaches the maximum N-NCC_CnxEstReq number of retries.

[6a][6b]

The NCC sends connection release requests to RCST/RSGW A and RCST/RSGW B and starts timers T-NCC_CnxRelReq for RCST/RSGW A and for RCST/RSGW B.

[7a][7b]

Upon reception of RCST/RSGW A's connection release response, the NCC stops timer Timer T-NCC_CnxRelReq for RCST/RSGW A. RCST/RSGW B's connection release responses never arrive at the NCC.

[8][9][10]

Timer T-NCC_CnxRelReq expires for RCST/RSGW B and the NCC retransmits the connection release request. The procedure is repeated until the NCC reaches the maximum N-NCC_CnxRelReq number of retries. After the maximum number of release retries is reached, the NCC may decide to logoff RCST/RSGW B.

In case of a transparent star scenario, where the exception procedure connection set up takes place between the initiating NCC/GW and RCST B, steps [1a], [2a], [6a] and [7a] are not considered.

C.1.2.4 Unsuccessful set-up - One of the peer parties rejects the connection

Figure C.15 represents the procedure in which the connection is rejected from one of the sides of the connection. Step [1a] and [1b] is similar to the successful connection set up (clause 6.1.1.1).

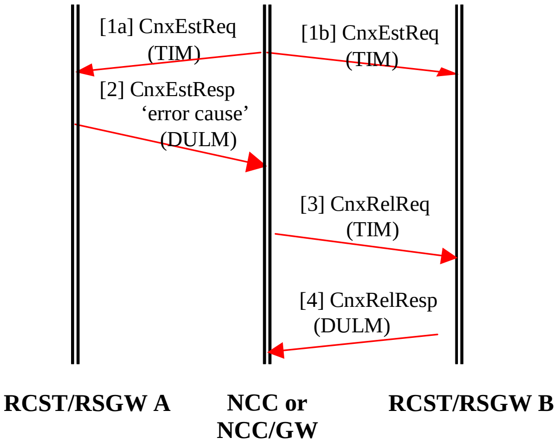


Figure C.15: NCC-initiated Point-to-Point unsuccessful connection establishment: One of the peer parties rejects the connection

[1] NCC performs admission control checks, allocates requested bandwidth, sends Connection Establishment Request to the peer RCST/RSGWs and starts timer T-NCC_CnxEstReq for both RCST/RSGWs.

[2] One of the peer RCST/RSGWs, in this case RCST/RSGW A, rejects the connection and sends a negative connection establishment response with the appropriate "error cause".

[3] NCC sends a connection release request to the other peer, RCST/RSGW B, and starts timer T-NCC_CnxRelReq for this peer.

[4] Upon reception of RCST/RGSW B's connection release response, the NCC stops timer T-NCC_CnxRelReq.

In case of a transparent star scenario, the exception procedure connection set up takes place between an the initiating NCC/GW and RCST A, steps [1b], [3] and [4] should be omitted.

C.1.2.5 Unsuccessful set-up - NCC release

Figure C.16 represents the procedure in which, in the middle of the connection establishment phase initiated by the NCC, the NCC decides to release the connection (e.g. due to internal management trigger).

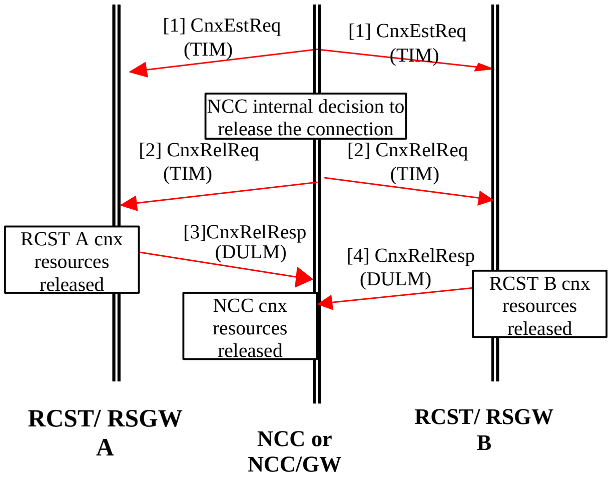


Figure C.16: NCC-initiated Point-to-Point unsuccessful connection establishment: NCC release

[1] NCC performs admission control checks, allocates requested bandwidth, sends Connection Establishment Requests to both RCST/RSGWs and starts timer T-NCC_CnxEstReq for both RCST/RSGWs.

[2][3][4]

Before receiving any connection release response from either of the peer RCST/RSGWs, the NCC decides to release the connection, sends Connection Release Requests to both sides of the connection and starts timers T-NCC_CnxRelReq for both RCST/RSGW A and RCST/RSGW B. Upon reception of RCST/RSGW A's and RSCT/RSGW B's connection release responses, the NCC stops the respective T-NCC_CnxRelReq timers and releases its resources for the connection.

In case of a transparent star scenario, the connection set up takes place between the NCC/GW and RCST.

C.2 Point-to-multipoint connection establishment procedures

C.2.1 RCST/RSGW initiated connections

C.2.1.1 Successful set-up: RCST/RSGW multicast source retry

Figure C.17 represents the procedure in which the multicast source successfully retries sending the setup request message to the NCC.

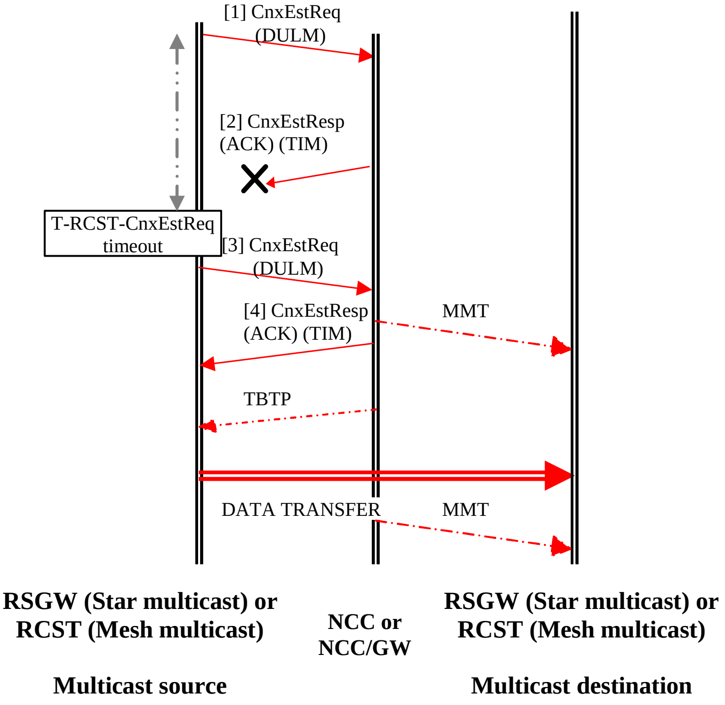


Figure C.17: RCST/RSGW initiated Point-to-Multipoint successful connection establishment: multicast source retry

[1] The multicast source requests the NCC for a multicast connection setup and starts timer T-RCST_CnxEstReq.

[2] The NCC performs admission control checks, allocates requested bandwidth and sends a Connection Establishment that does not reach the multicast source.

[3] Timer T-RCST_CnxEstReq expires. The multicast source retransmits the connection establishment request to the NCC with appropriate retry count and Identifier. The multicast source may retry up to a maximum N-RCST_CnxEstReq number of retries.

[4] The NCC recognizes the request belongs to an already established connection and resends the positive Connection Establishment Response towards the multicast source. Upon reception of the message, the multicast source stops timer T-RCST_CnxEstReq.

The NCC will periodically keep updating the MMT to the multicast destination and the NCC will send the TBTP to the multicast source so that multicast traffic can be initiated.

In case of a transparent reference scenario, the procedure previously described is also applicable, except that the multicast traffic transmitted from the multicast source will reach the multicast destination (in the same or in different destination downlink) thanks to a double hop provided by the NCC/GW.

C.2.1.2 Unsuccessful set-up: NCC reject

Figure C.18 represents the procedure in which the NCC rejects the connection establishment request sent by the multicast source.

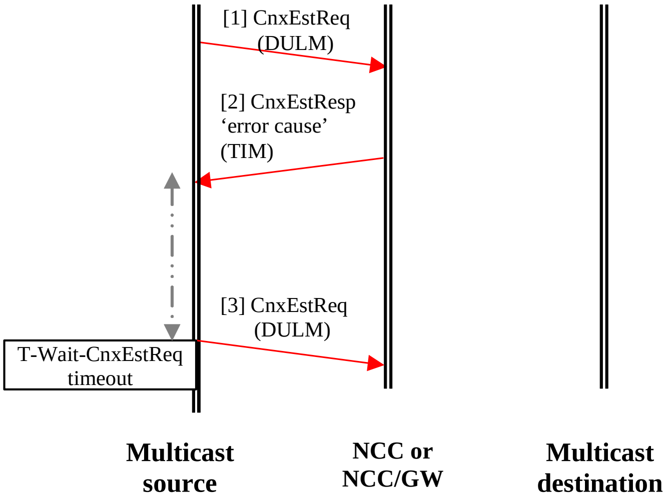


Figure C.18: RCST/RSGW initiated Point-to-Multipoint unsuccessful connection establishment: NCC reject

[1] The multicast source sends a connection establishment request and starts timer T-RCST_CnxEstReq.

[2] The NCC performs admission control checks, decides to reject the connection and sends a negative Connection Establishment Response to the multicast source with the appropriate "error cause". Upon reception of the response, and if the connection triggering is still active, the multicast source starts timer T-RCST_Wait-CnxEstReq.

[3] When timer T-RCST_Wait-CnxEstReq expires, the multicast source can retry sending another connection establishment request.

This procedure applies to all reference scenarios.

C.2.1.3 Unsuccessful set-up: RCST/RSGW multicast source release

Figure C.19 represents the procedure in which the multicast source releases the initiated connection before receiving a response by the NCC.

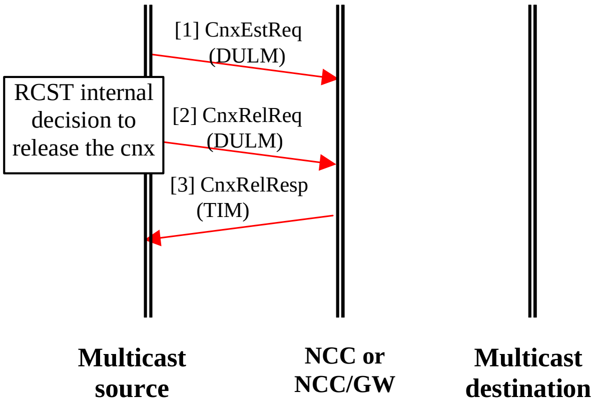


Figure C.19: RCST/RSGW initiated Point-to-Multipoint unsuccessful connection establishment: multicast source release

[1] The multicast source requests the NCC for a multicast connection setup and starts timer T-RCST_CnxEstReq.

[2] Before the connection is established, the multicast source decides to release it (e.g. due to an internal trigger or a console command), stops timer T-RCST_CnxEstReq, sends a connection release request to the NCC and starts timer T-RCST_CnxRelReq.

[3] The NCC accepts the connection release request and releases its resources for the connection. Upon reception of the NCC's Connection Release Response, the multicast source stops T-RCST_CnxRelReq timer (see connection release procedures).

This procedure applies to all reference scenarios.

C.2.1.4 Unsuccessful set-up: NCC release

Figure C.20 represents the procedure in which the multicast connection is released by the NCC, even if the multicast source did not complete the opening of the connection.

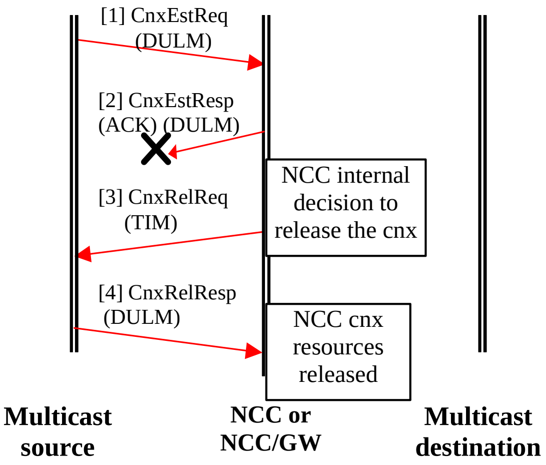


Figure C.20: RCST/RSGW initiated Point-to-Multipoint unsuccessful connection establishment: NCC release

[1] The multicast source requests the NCC for a multicast connection setup and starts timer T-RCST_CnxEstReq.

[2] The NCC replies with a positive connection establishment response, but the message does not reach the multicast source.

[3] Before timer T-RCST_CnxEstReq expires at the multicast source, the NCC decides to send a connection release request (e.g. due to a console command), stops timer T-RCST_CnxEstReq and starts timer T-RCST_CnxRelReq. The connection release request manages to arrive at the multicast source.

[4] Upon reception of the multicast source's response, the NCC stops timer T-RCST_CnxRelReq and releases its resources for the connection.

This procedure applies to all reference scenarios.

C.2.1.5 Unsuccessful set-up: Multicast source retry

The left hand side of figure C.21 represents the procedure in which the multicast source retries sending the multicast connection establishment message to the NCC and where the response from the NCC never reaches the RCST/RSGW. From the terminal's point of view, there is no way to distinguish between this scenario and the scenario in the right hand side of figure C.21, in which the multicast source is asking for a point-to-multipoint connection but the message is always lost. Either way, it is an unsuccessful setup because the multicast source will retry up to the maximum number of times and a release procedure will be initiated at the multicast source.

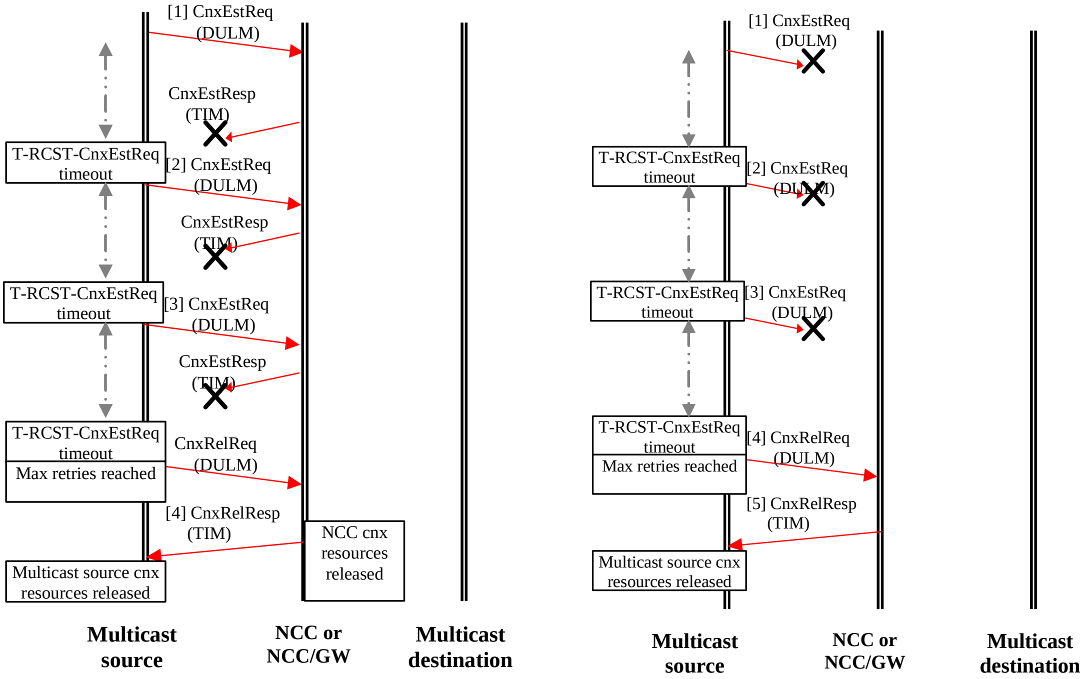


Figure C.21: RCST/RSGW initiated Point-to-Multipoint unsuccessful connection establishment: NCC no answer

[1] The multicast source requests the NCC for a multicast connection setup and starts timer T-RCST_CnxEstReq. Either in the left hand side of figure C.21, where the NCC's connection establishment response does not reach the multicast source, or in the right hand side of figure C.21, where the multicast source's connection establishment request gets lost on its way to the NCC, timer T-RCST_CnxEstReq expires at the multicast source.

[2] The multicast source retransmits the setup request message to the NCC with appropriate retry count and Identifier. The multicast source may retry up to a maximum N-RCST_CnxEstReq number of times.

[3] The maximum number of retries is reached and still no answer is received from the NCC. The multicast source sends a connection release request and starts timer T-RCST-CnxRelReq. Upon reception of the NCC's connection release response (which has managed to arrive at the multicast source), the multicast source stops timer T-RCST-CnxRelReq and releases the resources for the connection.

This procedure applies to all reference scenarios.

C.2.2 NCC initiated connections

C.2.2.1 Successful set-up: NCC retry

Figure C.22 shows another successful multicast setup procedure in which the NCC sends the connection establishment request twice.

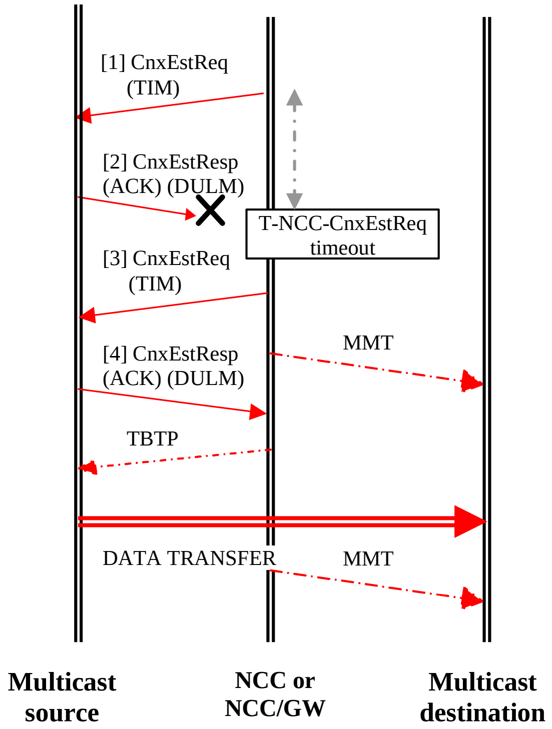


Figure C.22: NCC initiated Point-to-Multipoint successful connection establishment: NCC retry

[1] The NCC requests the multicast source for a multicast connection setup and starts timer T-NCC_CnxEstReq.

[2] The multicast source replies with a connection establishment response, but the message does not reach the NCC.

[3] Timer T-NCC_CnxEstReq expires. The NCC retransmits the setup request message to the multicast source with appropriate retry count and Identifier. The NCC may retry up to a maximum N-NCC_CnxEstReq number of times.

[4] On the first retry, the multicast source's reply manages to arrive at the NCC. Upon reception of the connection establishment response, the NCC stops timer T-NCC_CnxEstReq, considers the connection is open and updates the MMT.

This procedure is applicable to all reference scenarios.

C.2.2.2 Unsuccessful set-up: Multicast source reject

Figure C.23 shows an unsuccessful multicast setup procedure in which the multicast source rejects the connection establishment request.

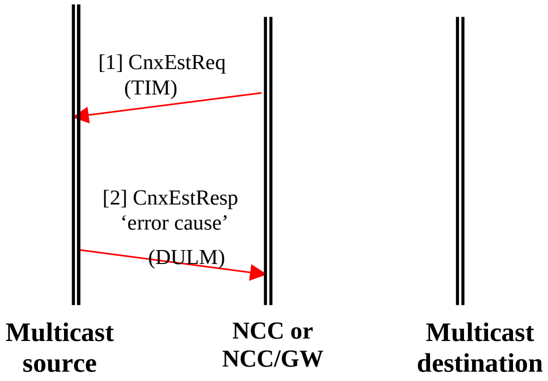


Figure C.23: NCC initiated Point-to-Multipoint unsuccessful connection establishment: Multicast source reject

[1] The NCC requests the multicast source for a multicast connection establishment and starts timer T-NCC_CnxEstReq.

[2] The multicast source rejects the connection establishment request and sends a connection establishment response with the appropriate "error cause". Upon reception of this message, the NCC stops timer T-NCC_CnxEstReq.

This procedure is applicable to all reference scenarios.

C.2.2.3 Unsuccessful set-up: NCC release

Figure C.24 represents the procedure in which the NCC decides to send a connection release request before the multicast source's connection establishment response has arrived at the NCC.

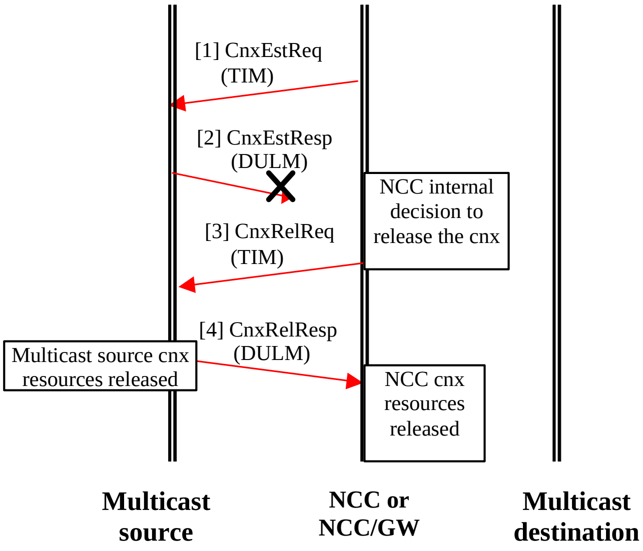


Figure C.24: NCC initiated Point-to-Multipoint connection establishment: NCC release

[1] The NCC requests the multicast source for a multicast connection setup and starts timer T-NCC_CnxEstReq.

[2] The multicast source answers the request with a connection establishment response that does not reach the NCC.

[3] The NCC decides to release the connection (e.g. due to an internal trigger). It stops timer T-NCC_CnxEstReq, sends a connection release request to the multicast source and starts timer T-NCC_CnxRelReq.

[4] The multicast source replies with a connection release response. See clause 6.4 for more information of release procedures for multicast connections.

This procedure is applicable to all reference scenarios.

C.2.2.4 Unsuccessful set-up: Multicast source no answer

The left hand side of figure C.25 represents the procedure in which the NCC tries to establish an NCC-initiated multicast connection, but the multicast source's responses don't arrive at the NCC. The right hand side of figure C.25 represents an equivalent procedure from the NCC's point of view, in which the NCC's connection establishment requests don't arrive at the multicast source. Either way, the maximum number of retries is reached at the NCC and the NCC releases the connection.

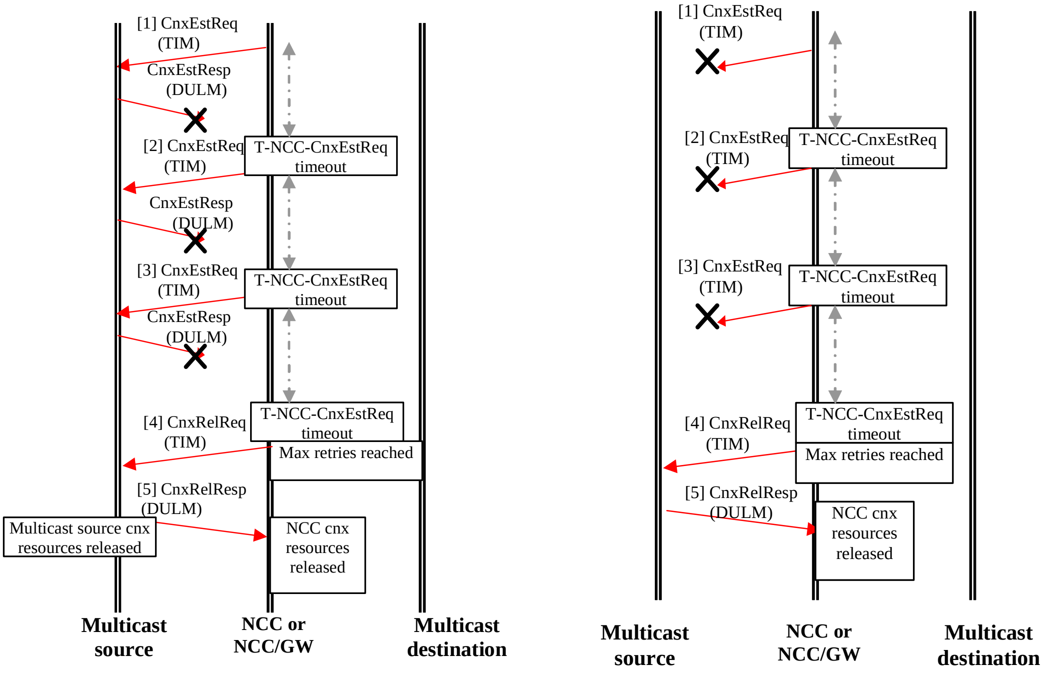


Figure C.25: NCC initiated Point-to-Multipoint unsuccessful connection establishment: Multicast source no answer

[1] The NCC requests the multicast source for a multicast connection setup and starts timer T-NCC_CnxEstReq. On the left hand side of figure C.25, the multicast source's responses don't arrive at the NCC. On the right hand side of figure C.25, the NCC's connection establishment requests don't arrive at the multicast source. Either way, timer T-NCC_CnxEstReq expires and still no answer is received at the NCC.

[2][3]

The NCC retransmits the connection establishment request to the multicast source with appropriate retry count and Identifier. The NCC may retry up to a maximum N-NCC_CnxEstReq number of times.

[4] Maximum number of retries is reached and still no answer is received from the multicast source. The NCC sends the multicast source a release request and starts timer T-NCC_CnxRelReq.

[5] The multicast source releases its multicast connection's resources and replies the NCC with a connection release response that manages to arrive at the NCC. Upon reception of the response, the NCC releases the connection and stops timer T-NCC_CnxRelReq.

This procedure is applicable to all reference scenarios.

C.2.2.5 Unsuccessful set-up: Logoff multicast source

Figure C.26 represents an unsuccessful connection set up involving a multicast source not answering. The procedure is similar to the one described in clause C.2.2.4, however in this case the multicast source's connection release responses do not arrive at the NCC, and the NCC may eventually decide to logoff the multicast source. Steps [1] to [4] are the same as in clause C.2.2.4.

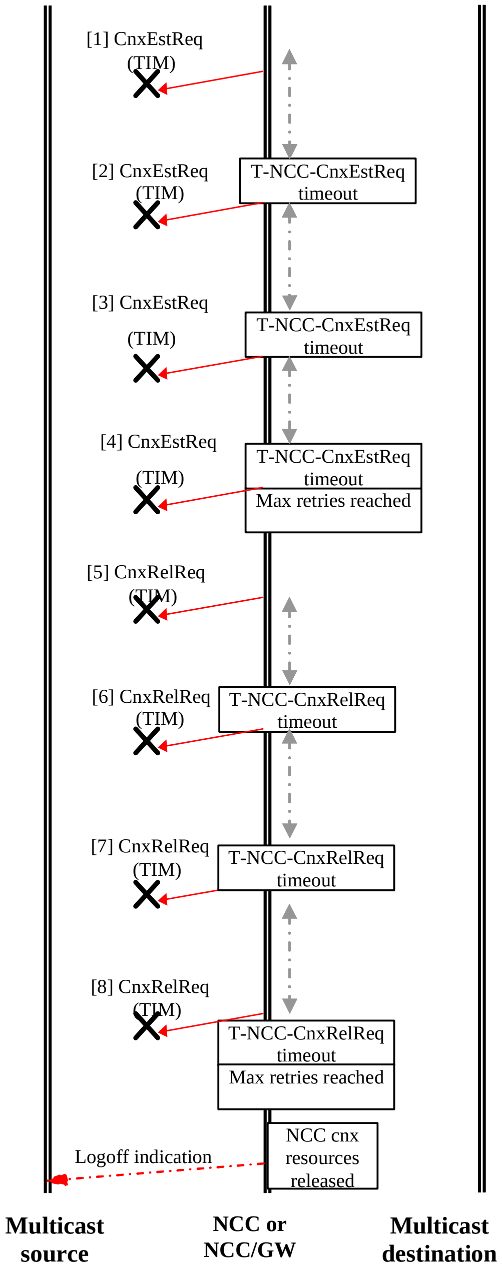


Figure C.26: NCC-initiated Point-to-Multipoint unsuccessful connection establishment: Logoff multicast source

[5] The NCC's connection release request either does not reach the multicast source or the multicast source's reply gets lost on its way to the NCC. Either way, the multicast source seems to be not answering from the point of view of the NCC. Upon expiry of timer T-NCC_CnxRelReq, the NCC sends another Connection Release Request to the multicast source with appropriate retry count and Identifier. The NCC may retry up to an N-NCC_CnxRelReq maximum number of times.

[6][7][8]

The maximum number of retries is reached and still no answer is received from the multicast source, and the NCC may decide to logoff the terminal, releasing its resources for the connection.

This procedure is also applicable to all reference scenarios.

C.3 Point-to-point Connection release procedures

C.3.1 RCST/RSGW initiated connections

C.3.1.1 Successful release by the RCST/RSGW: RCST/RSGW retry or NCC no answer

Figure C.27 shows an RCST/RSGW requesting a connection release request for an RCST/RSGW initiated connection. In this exception procedure, the answer from the NCC never reaches the initiating RCST/RSGW. The initiating RCST/RSGW will retry sending a connection release request up to a maximum number of times. If still no answer is received, the terminal may decide to logoff, as it understands that if no answer is received from the NCC, the NCC could be in an abnormal state.

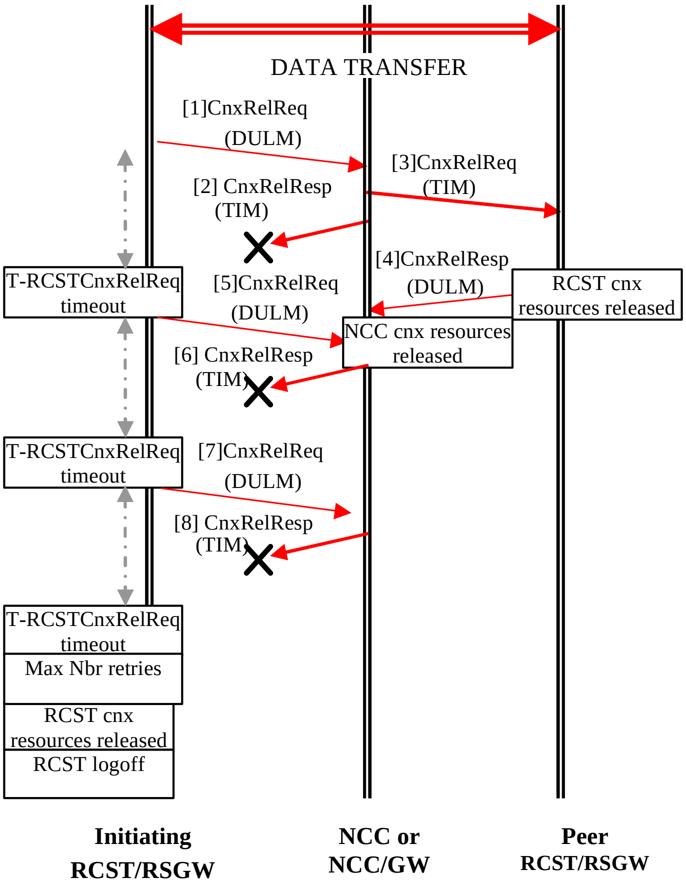


Figure C.27: RCST/RSGW initiated Point-to-Point successful connection release: RCST/RSGW retry or NCC no answer

[1] One side of the connection requests a connection release and starts timer T-RCST_CnxRelReq.

[2] The NCC replies by sending a connection release response to the initiating party.

[3] The NCC sends a connection release request to the other side of the connection and starts timer T-NCC_CnxRelReq for the peer RCST/RSGW.

[4] Upon reception of the peer RCST/RSGW's connection release response, the NCC stops timer T-NCC_CnxRelReq and releases all its resources for the connection.

[5][6][7][8]

Timer T-RCST_CnxRelReq expires. The initiating RCST/RSGW retransmits the release request message to the NCC with appropriate retry count and Identifier. The initiating RCST/RSGW may retry up to a maximum N-RCST_CnxRelReq number of times. Maximum number of retries is reached and, as the initiating RCST/RSGW cannot be sure about the NCC status, it may decide to logoff (whether this logoff is mandatory or optional is a system specific issue).

In a star transparent reference scenario, the same procedure applies, by skipping steps [3] and [4].

C.3.1.2 Successful release initiated by the RCST/RSGW: RCST/RSGW no answer

Figure C.28 shows a successful release of an RCST/RSGW initiated connection. In this procedure, the response from the peer RCST/RSGW gets lost on its way to the NCC. The maximum number of retries is reached at the NCC and the NCC may decide to send a logoff indication to the peer RCST/RSGW, after which the NCC considers the connection has been released.

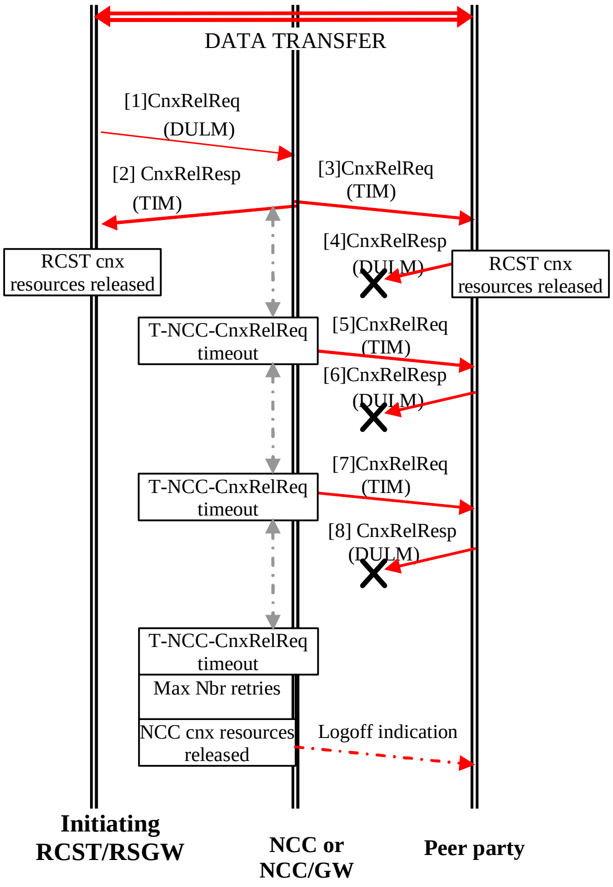


Figure C.28: RCST/RSGW initiated Point-to-Point connection - Successful release initiated by the RCST/RSGW: RCST/RSGW no answer

[1] The connection's initiator RCST/RSGW requests a connection release.

[2][3]

The NCC sends a connection release response to the initiating RCST/RSGW, a connection release request to the peer party and starts timer T-NCC_CnxRelReq for the peer RCST/RSGW.

[4][5][6][7][8]

Timer T-NCC_CnxRelReq expires for the peer RCST/RSGW. The NCC retransmits the release request message to the peer RCST/RSGW with appropriate retry count and Identifier. The NCC may retry up to a maximum N-NCC_CnxRelReq number of times. Maximum number of retries is reached, the NCC closes the connection as it cannot be sure about the status of the peer party and it may decide to send a logoff to the peer RCST/RSGW.

In a star transparent reference scenario, the same procedure applies by skipping steps [1] and [2].

C.3.1.3 Successful release initiated by the NCC: NCC retry

Figure C.29 shows a successful release procedure initiated by the NCC for an RCST/RSTW initiated connection. In this case, the peer RCST/RSGW release responses get lost. The maximum number of retries is reached at the NCC, so the NCC may decide to send a logoff to the peer RCST/RSGW, after which the NCC considers the connection has been released and closed.

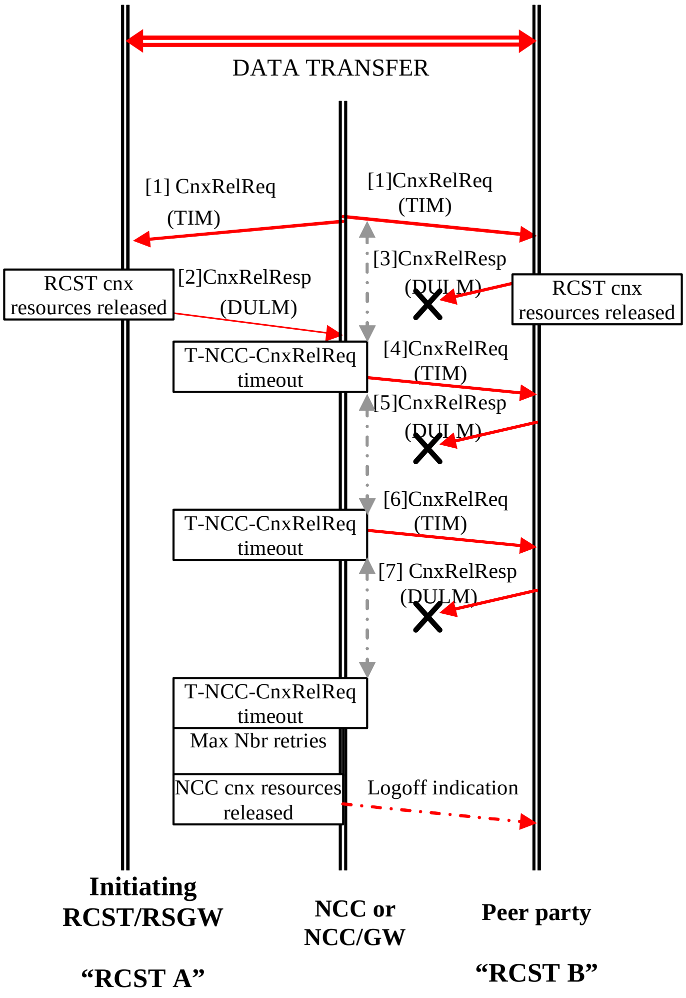


Figure C.29: RCST/RSGW initiated Point-to-Point connection - Successful release initiated by the NCC: NCC retry

[1] The NCC requests a connection release to both peers involved in the connection (RCST A, which had previously initiated the connection about to be released, and RCST B) and starts timer T-NCC_CnxRelReq.

[2] RCST A answers with a connection release response and it releases its connection resources.

[3] The NCC stops timer T-NCC_CnxRelReq for peer RCST A and waits for RCST B to answer.

[4][5][6][7][8]

Timer T-NCC_CnxRelReq expires for RCST B. The NCC retransmits the release request message to RCST B with appropriate retry count and Identifier. The NCC may retry up to a maximum N-NCC_CnxRelReq number of times. Maximum number of retries is reached, the NCC closes the connection as it cannot be sure about the status of RCST B and it may decide to send a logoff to this terminal.

In a star transparent reference scenario, the same procedure applies, by skipping steps [1] and [2].

C.3.1.4 Successful release by the NCC: Peer RCST/RSGW logoff

Figure C.30 represents a procedure in which one of the sides of the connection decides to logoff. The NCC would detect that one of the sides is "not synchronized" and will release all the connection where the RCST/RSGW was involved.

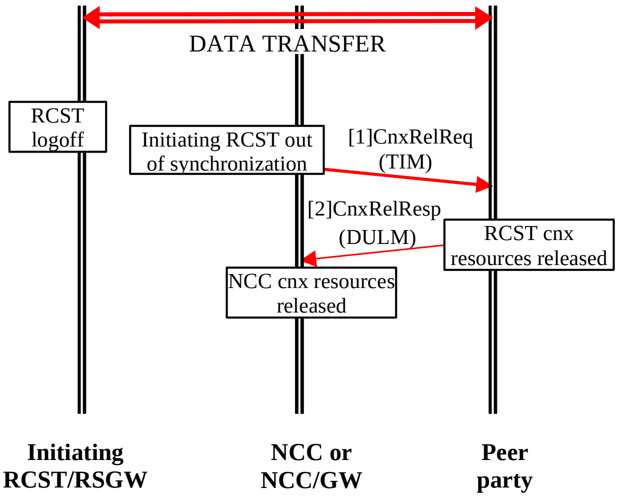


Figure C.30: RCST/RSGW initiated Point-to-Point connection - Successful release initiated by the NCC: Peer RCST/RSGW logoff

One side of the connection logoffs (the peer party).

[1] The NCC detects that the initiating RCST/RSGW is no longer synchronized. The NCC sends a connection release request to the peer RCST/RSGW and starts timer T-NCC_CnxRelReq.

[3] The peer RCST/RSGW answers with a connection release response. Upon reception of connection release response, the NCC stops timer T-NCC_CnxRelReq and releases all resources.

In a star transparent reference scenario, the NCC/GW should consider the connection is released after having detected the RCST/RSGW logoff status.

C.3.1.5 Successful release by the RCST/RSGW: NCC busy

Figure C.31 represents the procedure in which, the NCC due to internal reasons, cannot process the connection release request sent by the terminal, and it will send a connection release reject with an appropriate error cause (e.g. "NCC busy" or "NCC unavailability"). The RCST/RSGW should wait for timer T-RCST-Wait_CnxEstReq to expire before sending another connection release request. If after a maximum number of retries the NCC keeps rejecting the connection release request, the RCST/RSGW will release its resources for the connection. It is then the NCC's responsibility to request the status of all active connections at the RCSG/RSGW by sending a connection status statistics request.

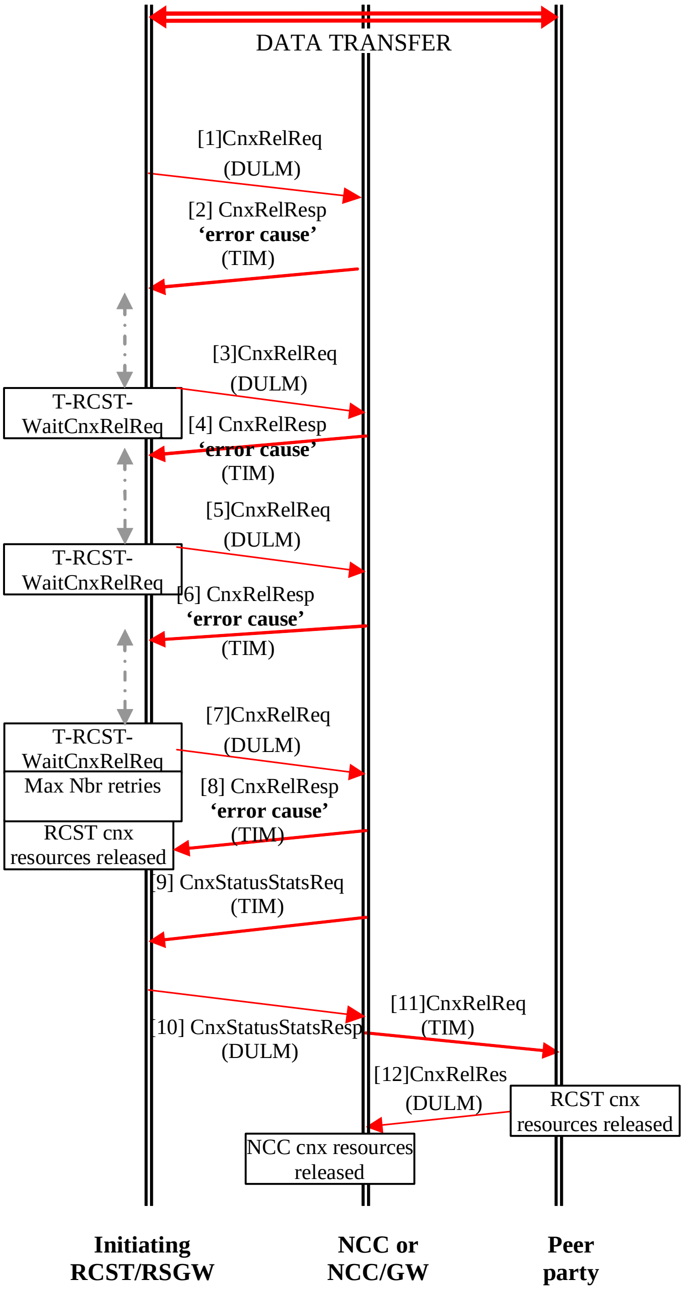


Figure C.31: RCST/RSGW initiated connection - Point-to-Point connection release initiated by the RCST/RSGW: NCC busy

[1] The initiating RCST/RSGW requests a connection release and starts timer T-RCST_CnxRelReq.

[2] The NCC rejects the request and sends a connection release response initiating RCST/RSGW that includes an appropriate "error cause" (e.g. "NCC busy" or "NCC unavailability").

[3][4][5][6][7]

After an NCC reject, it is recommended to wait a certain time before a sending another message to the NCC. This way, the initiating RCST/RSGW starts timer T-RCST-Wait_CnxRelReq. When this timer expires, initiating RCST/RSGW can send another connection release request. This procedure may be repeated up to N-RCST_CnxRelReq times.

[8] Maximum number of retries is reached; the initiating RCST/RSGW releases its connection resources and assumes the connection is no longer active.

[9] The NCC verifies the status of the initiating RCST/RSGW's active connections by sending a Connection Status Stats Request.

[10] The initiating RCST/RSGW answers back with a Connection Status Stats Response. Upon reception of this response, the NCC updates its internal information about initiating RCST/RSGW's active connections.

[11] The NCC sends a connection release request to peer RCST/RSGW and starts timer T-NCC_CnxRelReq.

[12] Peer RCST/RSGW answers with a connection release response. Upon reception of this response, the NCC stops timer T-NCC_CnxRelReq and releases the resources for this connection.

In a star transparent reference scenario, the same procedure applies, by skipping steps [11] and [12].

C.4 Point-to-multipoint connection release procedures

C.4.1 RCST/RSGW initiated connections

C.4.1.1 Successful release by the multicast source: NCC's responses get lost

Figure C.32 represents the procedure in which the multicast source (that had previously initiated the connection) requests a connection release to the NCC, but the NCC's answer never reaches the multicast source. The multicast source will retry up to a maximum number of retries. If still no answer is received, the multicast source may logoff (system specific issue).

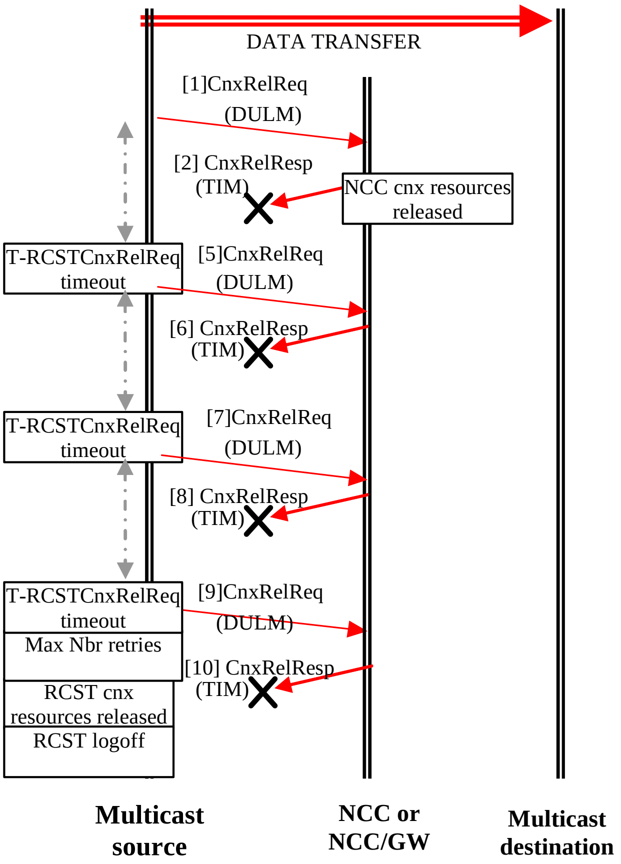


Figure C.32: RCST/RSGW initiated Point-to-Multipoint connection - Successful release by the multicast source: NCC's responses get lost

[1] The multicast source requests a connection release to the NCC and starts timer T-RCST_CnxRelReq.

[2] The NCC sends a release response and distributes a new MMT removing the IP multicast group.

[3][4][5][6][7][8][9][10]

Timer T-RCST_CnxRelReq expires. The multicast source retransmits the release request message to the NCC with appropriate retry count and Identifier. The multicast source may retry up to a maximum retries N-RCST_CnxRelReq. When the maximum number of retries is reached and still no answer is received from the NCC, the multicast source releases its resources for the connection and it may decide to logoff if it has no other active connections.

In a star transparent reference scenario, the same procedure applies between the RCST and the NCC/GW.

C.4.1.2 Successful release by the multicast source: Multicast source's requests get lost

Figure C.33 represents a procedure that, from the multicast source's point of view, is equivalent to clause C.4.1.1. The terminal cannot distinguish whether the message has gotten lost in the way from the terminal to the NCC or in the way from the NCC to the terminal. The multicast source cannot assume that the connection has been released because resources are kept in the NCC and the NCC has no criteria to release the connection later on (no control of traffic timers). It is necessary that the multicast source logoffs if there are no active connections at the multicast source.

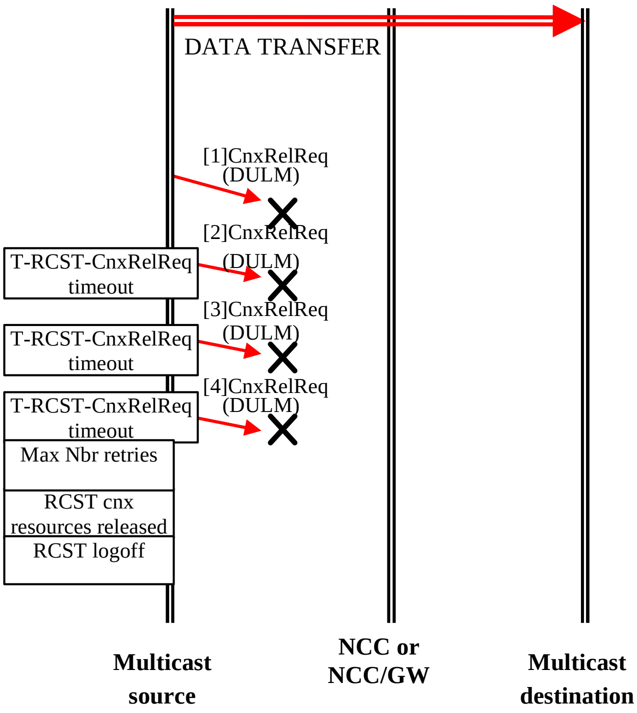


Figure C.33: RCST/RSGW Point-to-Multipoint initiated connection - Successful release initiated by the multicast source: Multicast source's requests get lost

[1] The multicast source requests a connection release and starts timer T-RCST_CnxRelReq. The request gets lost on its way to the NCC.

[2][3][4]

Timer T-RCST_CnxRelReq expires. The multicast source retransmits the release request message to the NCC with appropriate retry count and Identifier. The multicast source may retry up to a maximum retries N-RCST_CnxRelReq. When the maximum number of retries is reached, the multicast source closes the connection and eventually logoffs.

In a star transparent reference scenario, the same procedure applies between the RCST (multicast source) and the NCC/GW.

C.4.1.3 Successful release: NCC busy

In this scenario the NCC, due to internal reasons, cannot process the connection release request sent by the multicast source and it sends a connection release reject with an error cause (e.g. "NCC busy" or "NCC unavailable"). The multicast source waits an T-RCST-Wait_CnxEstReq time before retrying the connection release request. If after a maximum number of retries the NCC keeps sending the connection release reject, the multicast source releases the resources of the connection. It is under the NCC's responsibility to request the status of the multicast source's active connections by sending a connection status stats request.

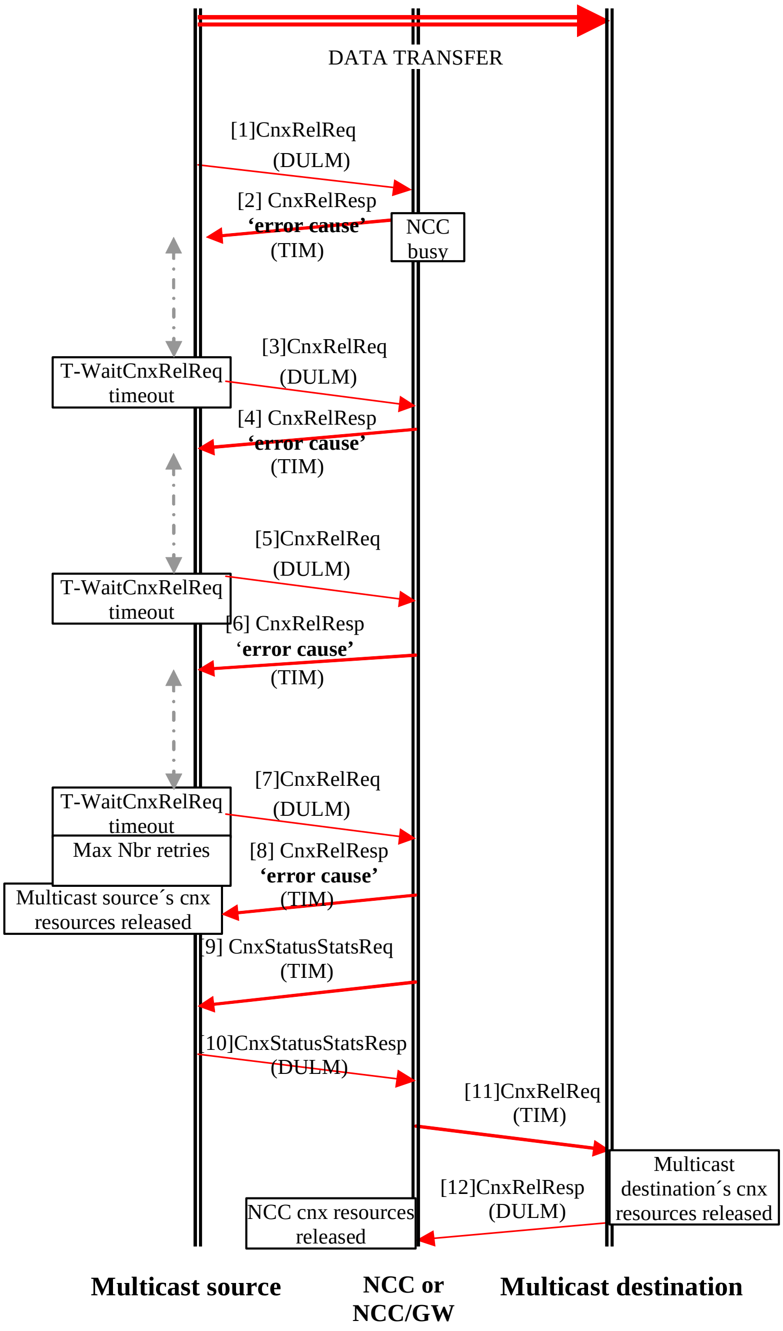


Figure C.34: RCST/RSGW Point-to-Multipoint initiated connection - Successful release: NCC busy

[1] The multicast source requests a connection release and starts timer T-RCST_CnxRelReq.

[2] The NCC rejects the connection release request and sends a connection release response to the multicast source with an appropriate "error cause" (e.g. "NCC busy" or "NCC unavailability"). Upon reception of the reject, the multicast source starts timer T-RCST-Wait_CnxRelReq.

[3][4][5][6][7]

When timer T-RCST-Wait_CnxRelReq expires, the multicast source can send another connection release request. This procedure may be repeated up to a maximum N-RCST_CnxRelReq number of times.

[8] Maximum number of retries is reached, the multicast source releases its resources for the connection and assumes the connection no longer active.

[9] The NCC verifies the status of the multicast source's active connections by sending a Connection Status Stats Request.

[10] The multicast source sends a Connection Status Stats Response that contains the list, status and statistics of all active connections at the multicast source. The NCC updates its internal information about the multicast source's active connections.

[11] The NCC sends a connection release request to the multicast destination and starts timer T-NCC_CnxRelReq.

[12] The multicast destination answers with a connection release response. Upon reception of the connection release response, the NCC stops timer T-NCC_CnxRelReq and releases its resources for the connection.

In a star transparent reference scenario, the same procedure applies between the multicast source and the NCC/GW.

C.4.1.4 Unsuccessful release by the NCC (of an RCST/RSGW initiated connection): Multicast source no answer

Figure C.35 represents the procedure in which the NCC sends a release request to the multicast source. The multicast source answers with a successful release response but the message gets lost on its way to the NCC. After a maximum number of retries, the NCC releases the resources of the connection and may decide to logoff the multicast source.

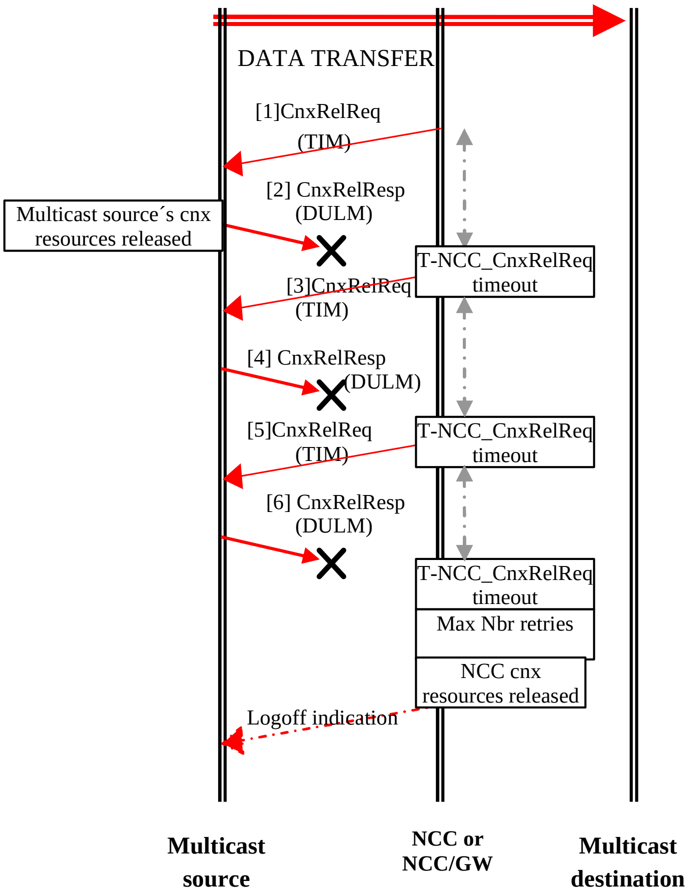


Figure C.35: RCST/RSGW initiated connection Point-to-Multipoint unsuccessful release (NCC initiated): Multicast source no answer

[1] The NCC sends a connection release request to the multicast source and starts timer T-NCC_CnxRelReq. The NCC distributes an updated MMT, in which the IP multicast group has been removed.

[2] The multicast source answers with a connection release response and releases its connection resources.

[3][4][5][6]

Timer T-NCC_CnxRelReq expires. The NCC retransmits the release request message to the multicast source with appropriate retry count and Identifier. The NCC may retry up to a maximum N-NCC_CnxRelReq number of times. When the maximum number of retries is reached, the NCC closes the connection as it cannot be sure about the status of the multicast source , releases the connection resources and may decide to send a logoff to this terminal

In a star transparent reference scenario, the same procedure applies between the RCST (multicast source) and the NCC/GW.

C.5 Point-to-point connection modify procedures (rate change)

C.5.1 RCST/RSGW initiated connections

C.5.1.1 Unsuccessful connection modify profile - NCC's responses get lost

Figure C.36 shows a connection modify request initiated by one terminal. The NCC processes if the modification can be accepted and generates a connection modify response which never reaches the terminal. After a maximum number of retries, the answer has not reached the terminal yet.

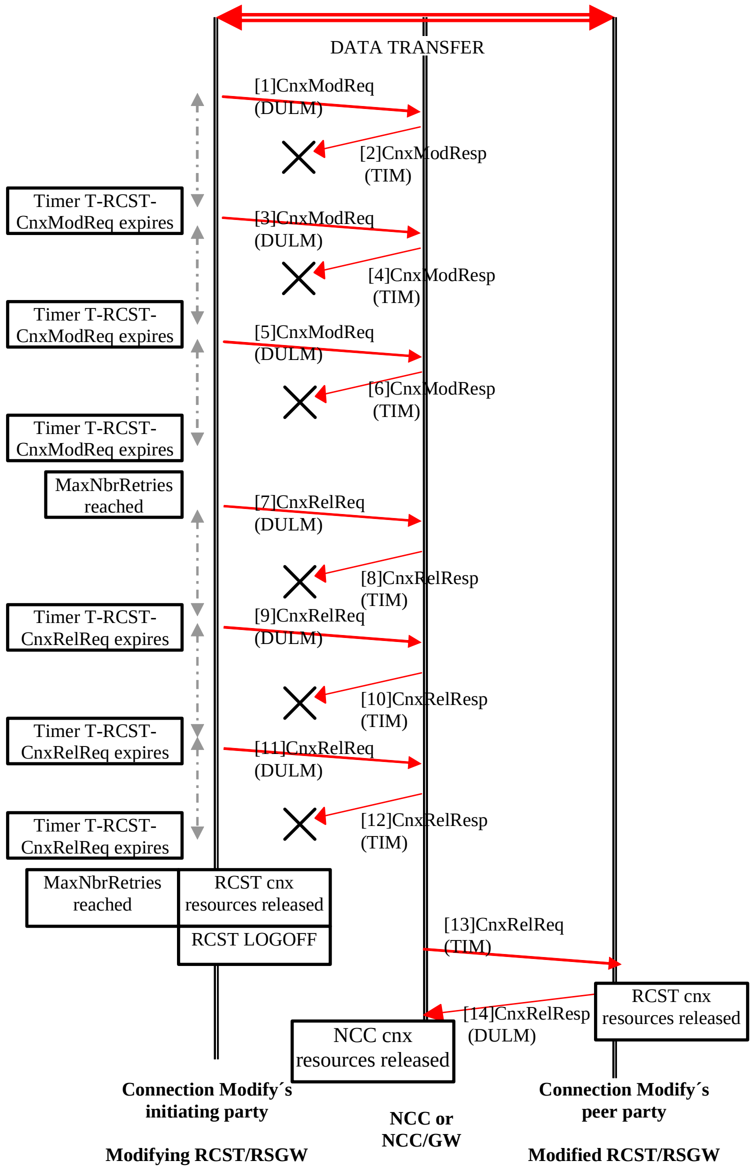


Figure C.36: RCST/RSGW initiated Point-to-Point unsuccessful connection modify: NCC's responses lost

[1] The Connection Modify Initiating RCST/RSGW (i.e. modifying RCST/RSGW) starts a connection modify procedure by sending a Connection Modify Request to the NCC. The modifying RCST/RSGW starts timer T-RCST_CnxModReq. When the request arrives at the NCC, the NCC modifies its resources for the connection, sends a connection modify response to the modifying RCST/RSGW and starts timer T-NCC_CnxModReq.

[2][3][4][5][6]

The connection modify request from the NCC does not reach the modifying RCST/RSGW. T-RCST_CnxModReq expires. The modifying RCST/RSGW retransmits the modify request with appropriate retry count and Identifier. The multicast source may retry up to a maximum N-RCST_CnxModReq number of times.

[7][8][9][10][11][12]

When the maximum number of retries is reached, the modifying RCST/RSGW sends a connection release request to the NCC and starts timer T-RCST_CnxRelReq. Upon reception of the release request, the NCC releases its resources for the modified connection and sends a connection release response to the modifying RCST/RSGW, which gets lost. When timer T-RCST_CnxRelReq expires, the modifying RCST/RSGW retransmits the release request message with appropriate retry count and Identifier. The modifying RCST/RSGW may retry up to a maximum N-RCST_CnxRelReq number of times. The maximum number of retries is reached for the modifying RCST/RSGW and still no answer is received; Modifying RCST/RSGW sends a logoff message to the NCC and releases its resources for the connection.

[13][14]

Upon detection of the initiating party's logoff, the NCC sends a release request to the peer RCST/RSGW. Upon reception of the request, the peer RCST/RSGW releases its resources for the connection and answers the NCC back with a positive connection release response. Upon reception of the response, the NCC releases the connection resources.

NOTE: From the point of view of Modifying RCST/RSGW, this procedure is similar to the case in which both the connection modify requests and the connection release requests get lost on their way to the NCC.

Similar procedure applies to a star transparent scenario except for steps [13] and [14].

C.5.2 NCC initiated connections

C.5.2.1 Unsuccessful connection modify: RCST A's responses get lost

Figure C.37 shows an connection modify procedure initiated by the NCC, in which the responses from one the RCST/RSGW always get lost.

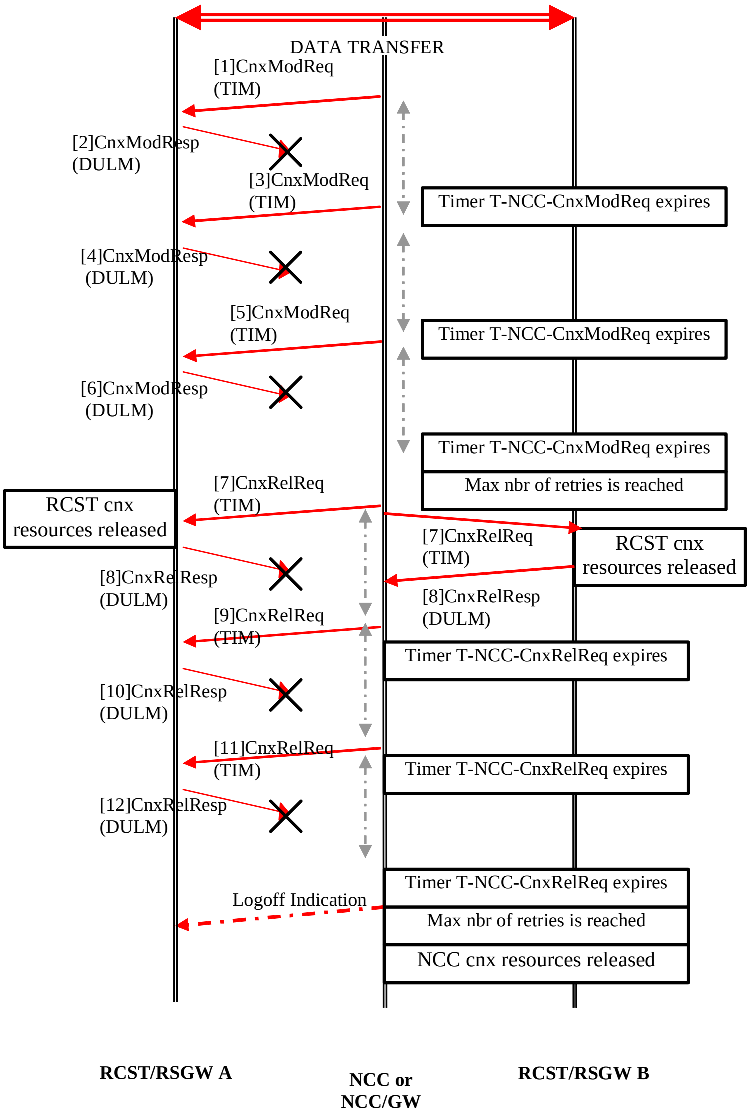


Figure C.37: RCST/RSGW initiated Point-to-Point unsuccessful connection modify: RCST A's connection modify responses lost

[1] The NCC requests a connection modify and starts timer T-NCC_CnxModReq, but no response is received from the RCST A.

[2][3][4][5][6]

Timer T-NCC_CnxModReq expires and the NCC resends the Connection Modify Request to RCST A. The NCC may retry up to a maximum N-NCC-CnxModReq number of times.

[7][8][9][10][11]

The maximum number of retries is reached and the NCC still does not know whether the requests have not arrived at the RCST A or whether RCST A's responses have got lost. The NCC starts a release procedure by sending connection release requests towards both peers involved in the connection (RCST A and RCST B) and by starting timers T-NCC_CnxRelReq for RCST A and RCST B. RCST B's release response arrives at the NCC, whereas RCST A's response gets lost. The NCC retries sending the message up to a maximum N-NCC_ CnxRelReq number of times. When the maximum number of retries is reached, and still no answer is received from RCST A at the NCC, the NCC may logoff RCST A.

For star transparent scenario similar procedure applies, except for steps [7] and [8] for the RCST B.

C.6 Point-to-multipoint connection modify procedures (rate change)

C.6.1 RCST/RSGW initiated connections

C.6.1.1 Unsuccessful connection modify profile - NCC's responses lost

Figure C.38 shows a connection modify request procedure initiated by the multicast source. The NCC processes if the modification can be accepted and generates a connection modify response which never reaches the terminal. After a maximum number of retries, the answer has not reached the terminal.

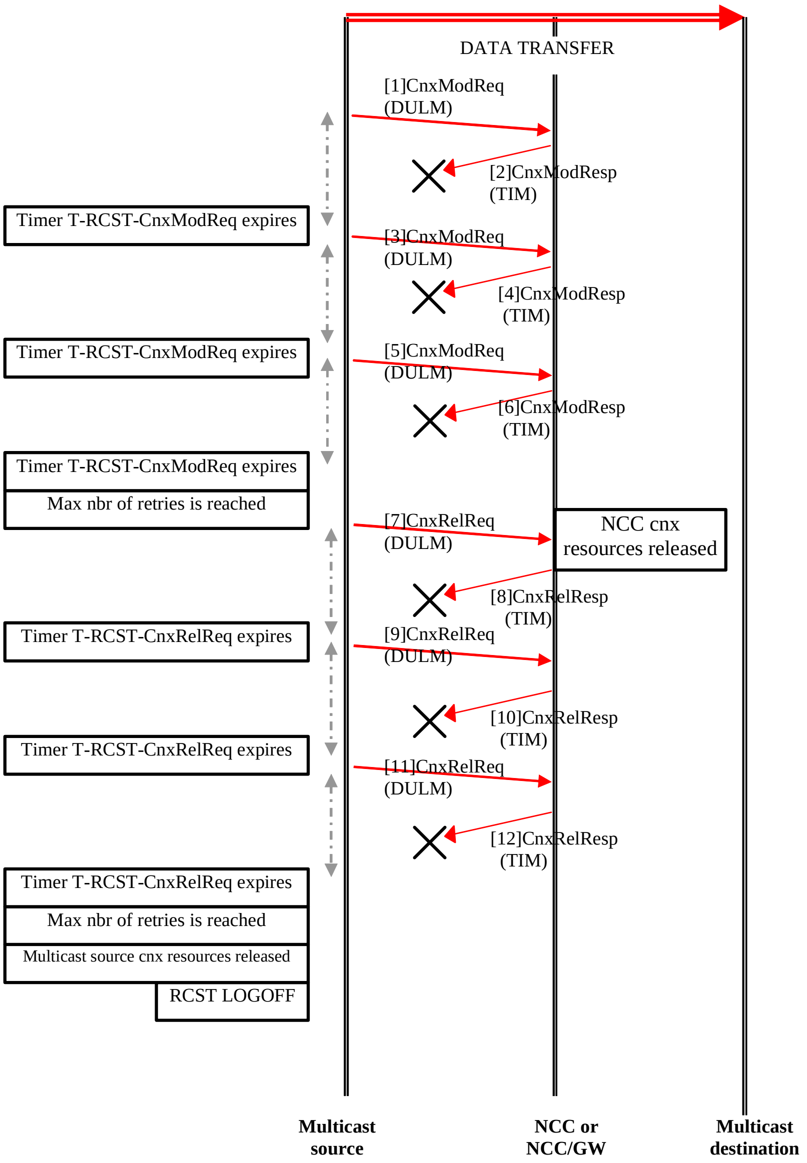

DATA TRANSFER
[1]CnxModReq
(DULM)
[2]CnxModResp
(TIM)
Timer T-RCST-CnxModReq expires
[3]CnxModReq
(DULM)
[4]CnxModResp
(TIM)
Timer T-RCST-CnxModReq expires
[5]CnxModReq
(DULM)
[6]CnxModResp
(TIM)
Timer T-RCST-CnxModReq expires
Max nbr of retries is reached
[7]CnxRelReq
(DULM)
NCC cnx resources released
[8]CnxRelResp
(TIM)
Timer T-RCST-CnxRelReq expires
[9]CnxRelReq
(DULM)
[10]CnxRelResp
(TIM)
Timer T-RCST-CnxRelReq expires
[11]CnxRelReq
(DULM)
[12]CnxRelResp
(TIM)
Timer T-RCST-CnxRelReq expires
Max nbr of retries is reached
Multicast source cnx resources released
RCST LOGOFF
Multicast source
NCC or NCC/GW
Multicast destination

Figure C.38: RCST/RSGW initiated Point-to-Multipoint unsuccessful connection modify: NCC's responses lost

[1] The multicast source starts a connection modify procedure by sending a request to the NCC. The multicast source starts timer T-RCST_CnxModReq. When the request arrives at the NCC, the NCC modifies its resources for the connection, sends a positive modify response to the multicast source and starts timer T-NCC_CnxModReq.

[2][3][4][5][6]

The connection modify request message from the NCC does not reach the multicast source. T-RCST_CnxModReq expires. The multicast source retransmits the modify request message with appropriate retry count and Identifier. The multicast source may retry up to a maximum N-RCST_CnxModReq number of times.

[7][8][9][10][11][12]

When the maximum number of retries is reached at the multicast source and still no answer is received from the NCC, the multicast source sends a connection release request to the NCC and starts timer T-RCST_CnxRelReq. Upon reception of the release request, the NCC releases its resources for the connection and sends a release response to the multicast source, which gets lost. When timer T-RCST_CnxRelReq expires, the multicast source retransmits the release request message with appropriate retry count and Identifier. The multicast source may retry up to a maximum N-RCST_CnxRelReq number of times. If the maximum number of retries is reached at the terminal and still no answer is received, the multicast source releases its resources for the connection and it may logoff.

NOTE: From the point of view of the multicast source, this procedure is similar to the case in which both the connection modify requests and the connection release requests get lost on their way to the NCC.

This procedure applies to all reference scenarios.

C.6.2 NCC initiated connections

C.6.2.1 Unsuccessful connection modify: RCST/RSGW connection modify responses lost

Figure C.39 represents the a connection modify procedure for a multicast connection initiated by the NCC, in which the responses from one of the RCST/RSGWs (i.e. the multicast source) always get lost.

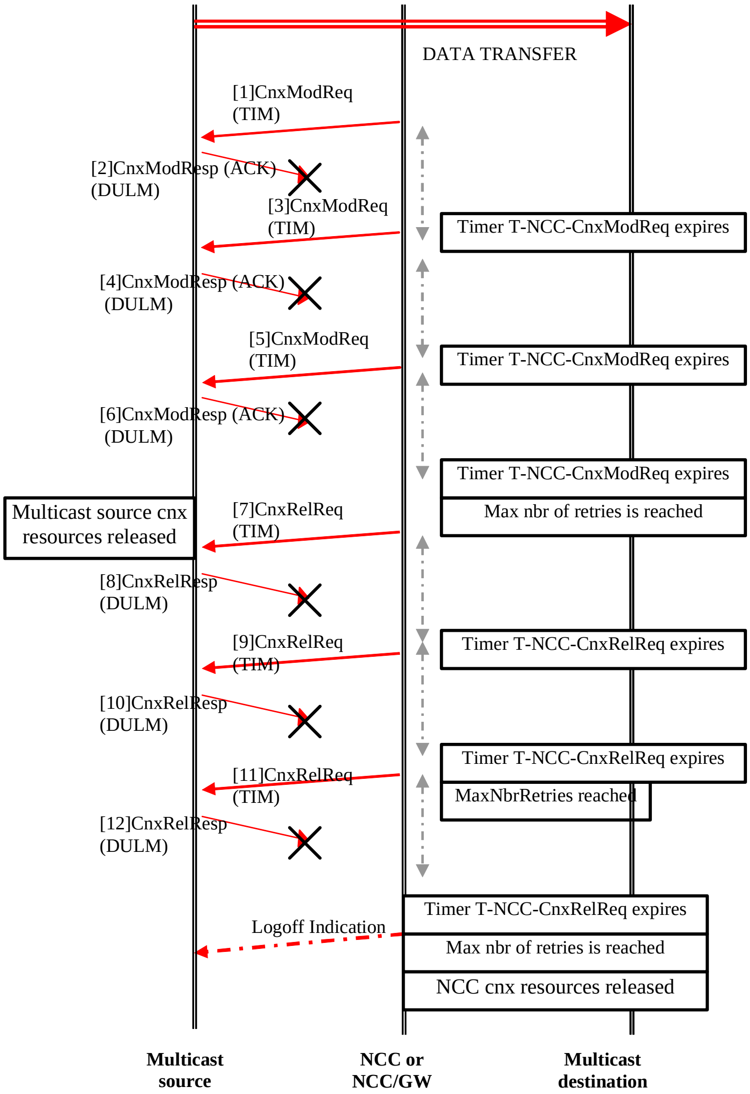

DATA TRANSFER
[1]CnxModReq (TIM)
[2]CnxModResp (ACK) (DULM)
[3]CnxModReq (TIM)
Timer T-NCC-CnxModReq expires
[4]CnxModResp (ACK) (DULM)
[5]CnxModReq (TIM)
Timer T-NCC-CnxModReq expires
[6]CnxModResp (ACK) (DULM)
Timer T-NCC-CnxModReq expires
Max nbr of retries is reached
Multicast source cnx resources released
[7]CnxRelReq (TIM)
[8]CnxRelResp (DULM)
[9]CnxRelReq (TIM)
Timer T-NCC-CnxRelReq expires
[10]CnxRelResp (DULM)
[11]CnxRelReq (TIM)
Timer T-NCC-CnxRelReq expires
MaxNbrRetries reached
[12]CnxRelResp (DULM)
Logoff Indication
Timer T-NCC-CnxRelReq expires
Max nbr of retries is reached
NCC cnx resources released
Multicast source
NCC or NCC/GW
Multicast destination

Figure C.39: RCST/RSGW initiated Point-to-Multipoint unsuccessful connection modify: RCST/RSGW's responses lost

[1] The NCC requests a connection modify and starts timer T-NCC_CnxModReq, but no response is received from the multicast source.

[2][3][4][5][6]

Timer T-NCC_CnxModReq expires and the NCC resends the Connection Modify Request to the multicast source. The multicast source may retry sending the message up to maximum N-NCC_CnxModReq times.

[7][8][9][10][11][12]

> After the maximum number of retries is reached, the NCC starts a release procedure by sending a CnxRelReq towards the multicast source. The NCC starts timer T-NCC_CnxRelReq. If the maximum number of retries N-NCC_CnxRelReq is reached and still no answer is received, the NCC releases the resources for the connection and it may logoff the multicast source.

This procedure applies to all reference scenarios.

C.7 RC modify procedures

C.7.1 RCST/RSGW initiated RC modify

C.7.1.1 Unsuccessful RC modify - NCC no answer

Figure C.40 shows an unsuccessful exchange of RC messages in which either the Initiating RCST/RSGW's RC Modify Requests or the NCC's RC Modify Responses get lost. Hence, from the point of view of the terminal that initiates the RC Modify procedure, the NCC seems to be "not answering".

In this RC Modify exception procedure, the Initiating RCST/RSGW sends an RC Modify request to the NCC that never arrives at the NCC. The Initiating RCST/RSGW resends the request when timer T-RCST_RCModReq expires, and the procedure may be repeated for as many times as counter N-RCST_RCModReq indicates. When the maximum number of retries is reached, the Initiating RCST/RSGW sets a specific timer of RC modify procedures called T-RCST-WaitInhibition_RCModReq. If this timer expires, still no response from the NCC is received and the RC Modify triggering conditions are still active, then the Initiating RCST/RSGW restarts the whole process again.

This schema may be repeated until the maximum N-RCST_WaitInhibition_RCModReq number of retries is reached. If this is the case and still no answer is received from the NCC, the Initiating RCST/RSGW logoffs (see note), releasing all the connections attached to the Initiating RCST/RSGW's mapping to a Channel_ID at the NCC. Also, the NCC sends release requests to the peer parties of all the connections mapped to the same Channel_ID which have the Initiating RCST/RSGW as the other peer party involved in such connections.

NOTE: It should be taken into consideration that, under the rare circumstances described above, a logoff is the most logical decision, even if it is eventually left to the terminal's manufacturer the final decision on whether to implement the logoff or not. The UML state machines included in annex A do perform the logoff if the above circumstances apply.

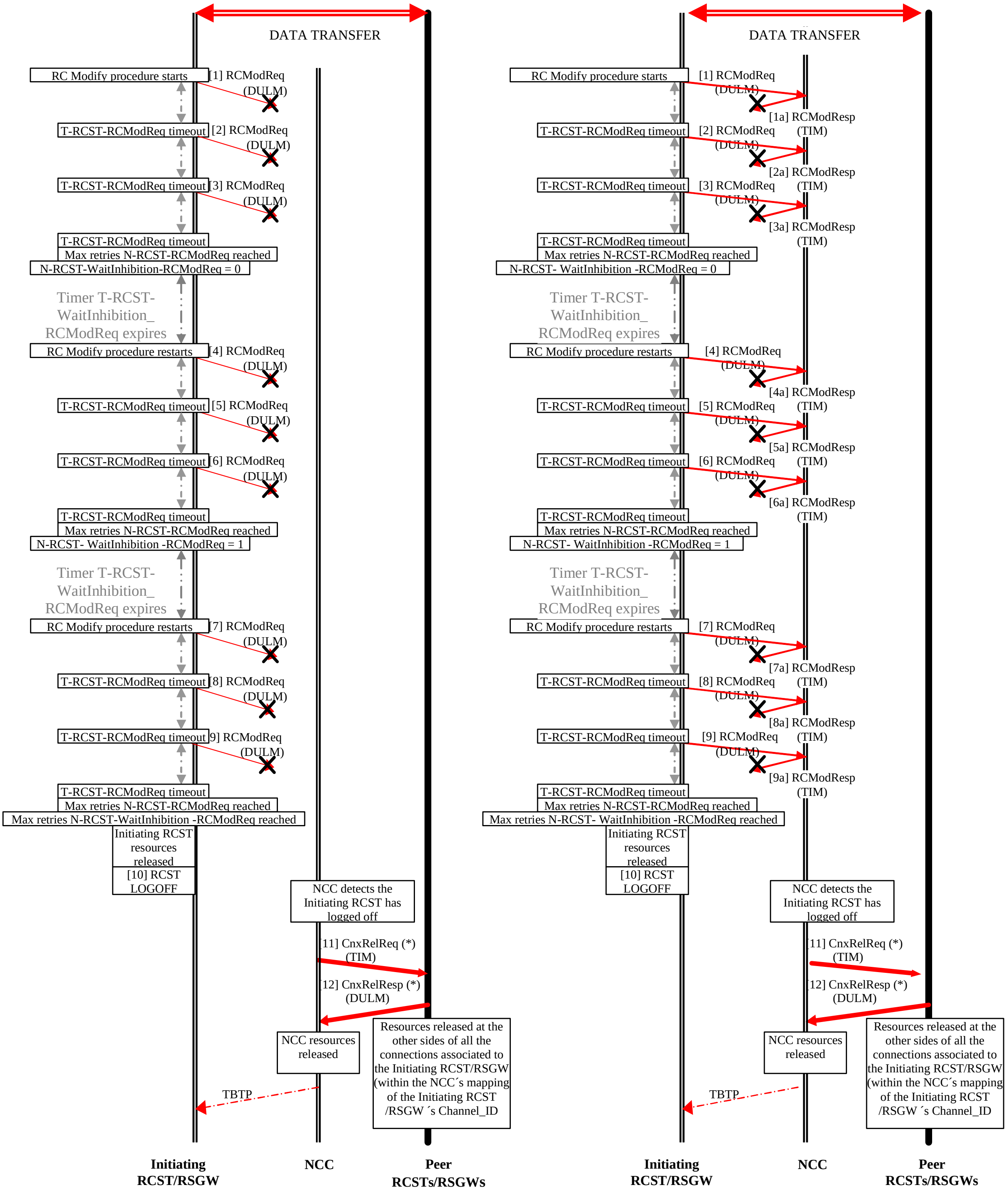

DATA TRANSFER
RC Modify procedure starts
[1] RCModReq (DULM)
T-RCST-RCModReq timeout
[2] RCModReq (DULM)
[3] RCModReq (DULM)
Max retries N-RCST-RCModReq reached
N-RCST-WaitInhibition-RCModReq = 0
Timer T-RCST-WaitInhibition_RCModReq expires
RC Modify procedure restarts
[4] RCModReq (DULM)
[5] RCModReq (DULM)
[6] RCModReq (DULM)
N-RCST- WaitInhibition -RCModReq = 1
[7] RCModReq (DULM)
[8] RCModReq (DULM)
[9] RCModReq (DULM)
Max retries N-RCST-WaitInhibition -RCModReq reached
Initiating RCST resources released
[10] RCST LOGOFF
NCC detects the Initiating RCST has logged off
[11] CnxRelReq (*) (TIM)
[12] CnxRelResp (*) (DULM)
NCC resources released
Resources released at the other sides of all the connections associated to the Initiating RCST/RSGW (within the NCC´s mapping of the Initiating RCST /RSGW ´s Channel_ID
TBTP
[1a] RCModResp (TIM)
[2a] RCModResp (TIM)
[3a] RCModResp (TIM)
[4a] RCModResp (TIM)
[5a] RCModResp (TIM)
[6a] RCModResp (TIM)
[7a] RCModResp (TIM)
[8a] RCModResp (TIM)
[9a] RCModResp (TIM)
Initiating RCST/RSGW
NCC
Peer RCSTs/RSGWs

Figure C.40: RCST/RSGW-initiated Point-to-Point unsuccessful RC modify: NCC no answer

[1][1a]

The initiating RCST/RSGW sends an RC Modify Requests to the NCC. Either the terminal's request [1] or the NCC's response [1a] gets lost. Timer T-RCST_RCModReq expires.

[2][2a][3][3a]

The initiating RCST/RSGW retransmits the RC Modify Request to the NCC with appropriate retry count and the unique Channel_ID used for this connection. Either the terminal's requests [2][3] or the NCC's responses [2a][3a] get lost. The initiating RCST/RSGW may retry up to a maximum of N-RCST_RCModReq retries. If the maximum number of retries is reached, the initiating RCST/RSGW sets an Inhibition timer.

[4][4a][5][5a][6][6a][7][7a][8][8a][9][9a]

Timer T-RCST-WaitInhibition_RCModReq expires and still no response from the NCC is received. If the RC Modify triggering conditions are still active, then the Initiating RCST/RSGW restarts the whole process again. Steps [1][1a][2][2a][3][3a] may be repeated up to a maximum of N- RCST_WaitInhibition_ RCModReq times.

[10*]

Maximum number of N-RCST_WaitInhibition_RCModReq retries is reached and still no answer is received from the NCC. The Initiating RCST/RSGW logoffs and the NCC may (*) or may not reach detect the lack of synchronization for the initiating RCST/RSGW.

[11*][12*]

The NCC sends connection release requests to the peer parties of all the connections mapped to the same Channel_ID which have the Initiating RCST/RSGW as the other peer party involved in such connections. In figure C.40 a thick line represents these peer RCSTs/RSGWs.

This procedure also applies to star transparent scenarios except for steps [11*] and [12*].

C.7.2 NCC initiated RC modify

C.7.2.1 Unsuccessful RC modify - RCST/RSGW no answer

Figure C.41 shows an unsuccessful exchange of RC messages in which either the NCC's RC Modify Requests or RCST/RSGW A's RC Modify Responses get lost. Hence, from the point of view of the NCC (or NCC/GW) that initiates the RC Modify procedure, RCST/RSGW A seems to be "not answering".

In this RC Modify exception procedure, the NCC sends an RC Modify request to RCST/RSGW A that never arrives at the terminal. The NCC resends the request when timer T-NCC_RCModReq expires and it may retry sending the request up to a maximum N-NCC_RCModReq number of times. When the maximum number of retries is reached, the NCC sets a specific timer of RC modify procedures called T-NCC-WaitInhibition_RCModReq. If this timer expires, still no response from RCST/RSGW A is received and the RC Modify triggering conditions are still active, then the NCC restarts the whole process again..

This schema may be repeated until the maximum N-NCC_WaitInhibition_RCModReq number of retries is reached. If this is the case and still no answer is received from RCST/RSGW A, the NCC logoffs RCST/RSGW A (see note) and sends release requests to "the other side" of all the connections mapped to the same Channel_ID which have RCST/RSGW A as one of the peer parties involved in such connections.

NOTE: It should be taken into consideration that, under the rare circumstances described above, a logoff is the most logical decision, even if it is eventually left to the terminal's manufacturer the final decision on whether to implement the logoff or not. The UML state machines included in annex A do perform the logoff if the above circumstances apply.

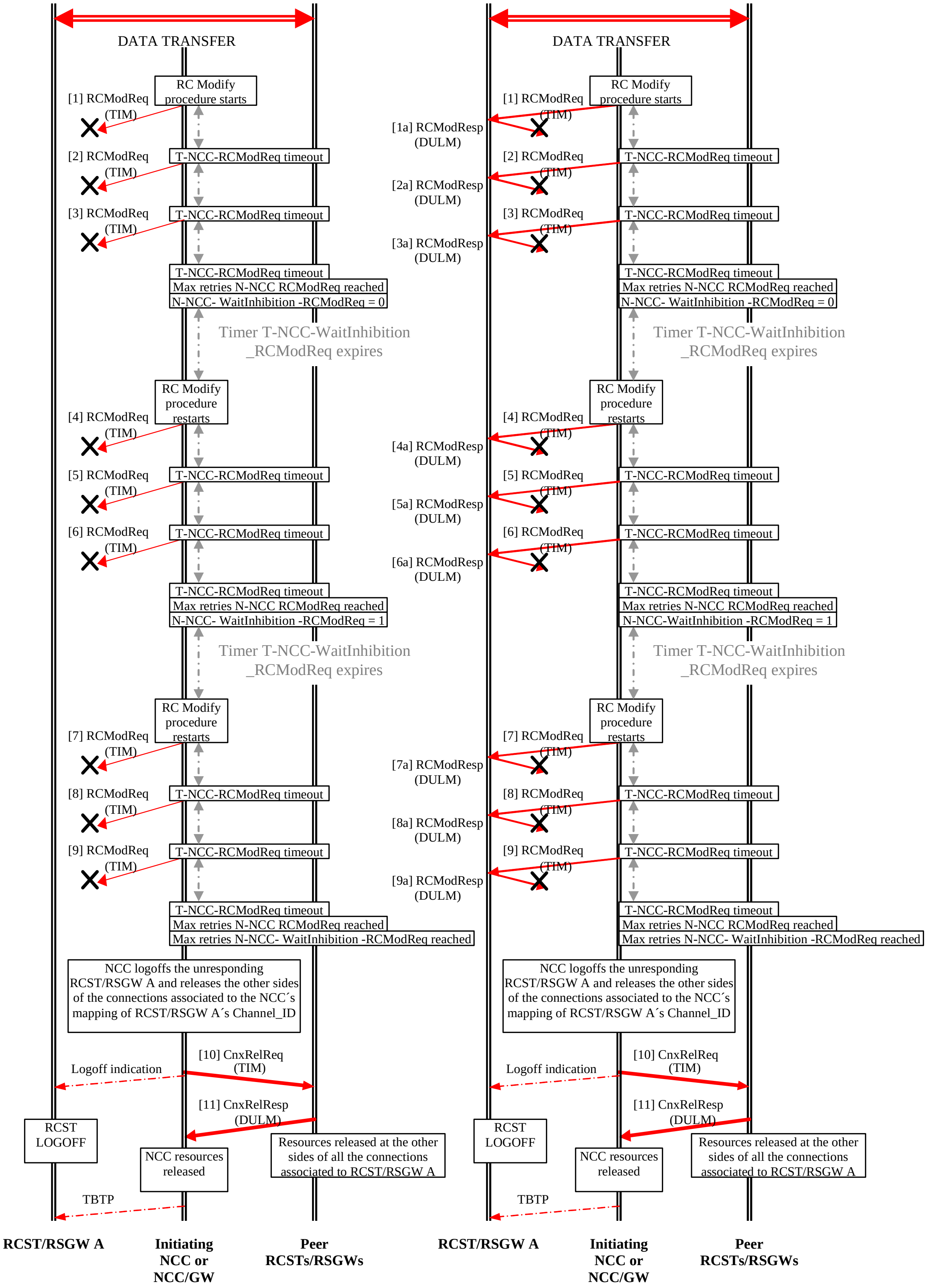

DATA TRANSFER
RC Modify procedure starts
[1] RCModReq (TIM)
T-NCC-RCModReq timeout
[2] RCModReq (TIM)
T-NCC-RCModReq timeout
[3] RCModReq (TIM)
T-NCC-RCModReq timeout
Max retries N-NCC RCModReq reached
N-NCC- WaitInhibition -RCModReq = 0
Timer T-NCC-WaitInhibition _RCModReq expires
RC Modify procedure restarts
[4] RCModReq (TIM)
T-NCC-RCModReq timeout
[5] RCModReq (TIM)
T-NCC-RCModReq timeout
[6] RCModReq (TIM)
T-NCC-RCModReq timeout
Max retries N-NCC RCModReq reached
N-NCC- WaitInhibition -RCModReq = 1
Timer T-NCC-WaitInhibition _RCModReq expires
RC Modify procedure restarts
[7] RCModReq (TIM)
T-NCC-RCModReq timeout
[8] RCModReq (TIM)
T-NCC-RCModReq timeout
[9] RCModReq (TIM)
T-NCC-RCModReq timeout
Max retries N-NCC RCModReq reached
Max retries N-NCC- WaitInhibition -RCModReq reached
NCC logoffs the unresponding RCST/RSGW A and releases the other sides of the connections associated to the NCC´s mapping of RCST/RSGW A´s Channel_ID
Logoff indication
[10] CnxRelReq (TIM)
[11] CnxRelResp (DULM)
RCST LOGOFF
NCC resources released
Resources released at the other sides of all the connections associated to RCST/RSGW A
TBTP
RCST/RSGW A
Initiating NCC or NCC/GW
Peer RCSTs/RSGWs
DATA TRANSFER
RC Modify procedure starts
[1] RCModReq (TIM)
[1a] RCModResp (DULM)
T-NCC-RCModReq timeout
[2] RCModReq (TIM)
[2a] RCModResp (DULM)
T-NCC-RCModReq timeout
[3] RCModReq (TIM)
[3a] RCModResp (DULM)
T-NCC-RCModReq timeout
Max retries N-NCC RCModReq reached
N-NCC- WaitInhibition -RCModReq = 0
Timer T-NCC-WaitInhibition _RCModReq expires
RC Modify procedure restarts
[4] RCModReq (TIM)
[4a] RCModResp (DULM)
T-NCC-RCModReq timeout
[5] RCModReq (TIM)
[5a] RCModResp (DULM)
T-NCC-RCModReq timeout
[6] RCModReq (TIM)
[6a] RCModResp (DULM)
T-NCC-RCModReq timeout
Max retries N-NCC RCModReq reached
N-NCC-WaitInhibition -RCModReq = 1
Timer T-NCC-WaitInhibition _RCModReq expires
RC Modify procedure restarts
[7] RCModReq (TIM)
[7a] RCModResp (DULM)
T-NCC-RCModReq timeout
[8] RCModReq (TIM)
[8a] RCModResp (DULM)
T-NCC-RCModReq timeout
[9] RCModReq (TIM)
[9a] RCModResp (DULM)
T-NCC-RCModReq timeout
Max retries N-NCC RCModReq reached
Max retries N-NCC- WaitInhibition -RCModReq reached
NCC logoffs the unresponding RCST/RSGW A and releases the other sides of the connections associated to the NCC´s mapping of RCST/RSGW A´s Channel_ID
Logoff indication
[10] CnxRelReq (TIM)
[11] CnxRelResp (DULM)
RCST LOGOFF
NCC resources released
Resources released at the other sides of all the connections associated to RCST/RSGW A
TBTP
RCST/RSGW A
Initiating NCC or NCC/GW
Peer RCSTs/RSGWs

Figure C.41: NCC-initiated Point-to-Point unsuccessful RC modify: RCST/RSGW no answer

[1][1a]

The NCC sends an RC Modify Requests to RCST/RSGW A. Either the NCC's request [1] or the terminal's response [1a] gets lost. Timer T-NCC_RCModReq expires.

[2][2a][3][3a]

The NCC retransmits the RC Modify Request to RCST/RSGW A with appropriate retry count and the unique Channel_ID used for this connection. Either the NCC's requests [2][3] or the terminal's responses [2a][3a] get lost. The NCC may retry up to a maximum of N-NCC_RCModReq retries. If the maximum number of retries is reached, the NCC sets an Inhibition timer.

[4][4a][5][5a][6][6a][7][7a][8][8a][9][9a]

Timer T-NCC_WaitInhibition_RCModReq expires, still no response from RCST/RSGW A is received and the RC Modify triggering conditions are still active, then the NCC restarts the whole process again.
Steps [1][1a][2][2a][3][3a] may be repeated up to a maximum of N-NCC_WaitInhibition_RCModReq times.
If the maximum number of N-NCC_WaitInhibition_RCModReq retries is reached and still no answer is received from RCST/RSGW A, the NCC logoffs RCST/RSGW A.

[10][11]

The NCC sends connection release requests to "the other side" of all the connections mapped to the same Channel_ID which have RCST/RSGW A as one of the peer parties involved in such connections. On figure C.41 a thick line represents these other peer RCSTs/RSGWs.

This procedure also applies to star transparent scenarios except for steps [10] and [11].

C.8 RCST Capability procedures

C.8.1 RCST/RSGW initiated RCST Capability notify

C.8.1.1 Unsuccessful RCST/RSGW initiated RCST Capability notify

In this exception procedure, after a maximum number of retries and no acknowledge received from the NCC, the initiating RCST/RSGW decides to logoff.

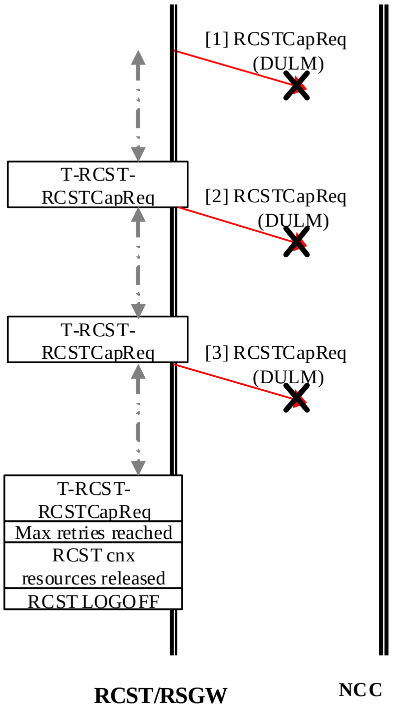


Figure C.42: RCST-initiated unsuccessful Capability Notify: NCC no answer

[1] After a successful logon, an RCST/RSGW may optionally start a second logon phase to notify additional parameters towards the NCC. If this is the case, the RCST/RGSW sends an RCST Capability request to the NCC and starts timer T-RCST-RCSTCapReq.

[2][3]

The request does not reach the NCC. When timer T-RCST-RCSTCapReq expires, the initiating RCST retries sending another RCST Capability request. The initiating RCST may retry up to a maximum N-RCST-RCSTCapReq number of times. Maximum number of retries is reached and still no response is received from the NCC. The initiating RCST should logoff even though this is a very drastic action to perform, since it releases all active connections.

This procedure applies to all reference scenarios.

C.8.2 NCC initiated Capability Notify

C.8.2.1 Unsuccessful NCC initiated RCST Capability Notify

Figure C.43 represents the NCC initiated RCST Capability Notify procedure towards a RCST/RSGW where the NCC's requests get lost.

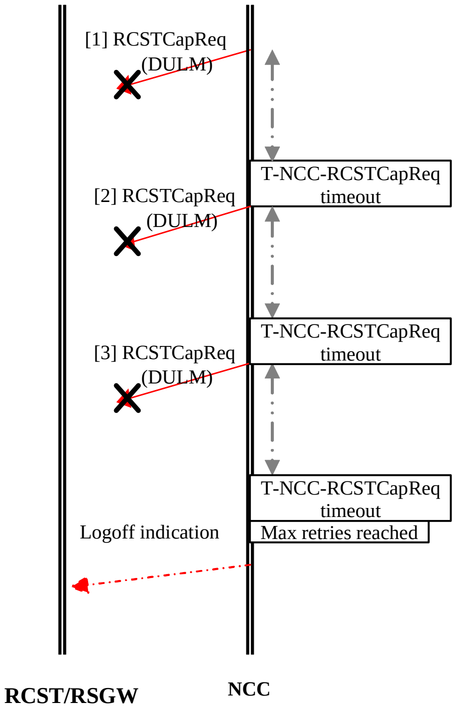


Figure C.43: NCC-initiated unsuccessful RCST Capability Notify: RCST/RSGW no answer

[1] The NCC may request an RCST/RSGW for additional capabilities (e.g. due to an NCC graceful handover or just for maintenance purposes) after verifying that RCST/RSGW is correctly synchronized. If this is the case, the NCC sends an RCST Capability request to an RCST/RSGW and starts timer T-NCC-RCSTCapReq.

[2][3]

The request does not reach the RCST/RSGW. When timer T-NCC-RCSTCapReq expires, the NCC retries sending another RCST Capability request. The NCC may retry up to a maximum N-NCC-RCSTCapReq number of times. Maximum number of retries is reached and still no response is received from the RCST/RSGW, and the NCC should perform a drastic action by logging off the RCST/RSGW.

NOTE 1: This is a very rare exception case which leads to a drastic logoff of the queried RCST/RSGW by the NCC that releases all the active connections.

NOTE 2: An NCC-initiated Capability Notify procedure is intended for query purposes, since the NCC cannot modify the queried terminal's capabilities.

NOTE 3: This procedure applies to all reference scenarios.

History

Document history		
V1.1.1	January 2009	Publication