International Telecommunication Union

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

G.7713/Y.1704

(05/2006)

SERIES G: TRANSMISSION SYSTEMS AND MEDIA, DIGITAL SYSTEMS AND NETWORKS

Data over Transport – Generic aspects – Transport network control aspects

SERIES Y: GLOBAL INFORMATION INFRASTRUCTURE, INTERNET PROTOCOL ASPECTS AND NEXT-GENERATION NETWORKS

Internet protocol aspects – Operation, administration and maintenance

Distributed Call and Connection Management (DCM)

ITU-T Recommendation G.7713/Y.1704

ITU-T

ITU-T G-SERIES RECOMMENDATIONS

TRANSMISSION SYSTEMS AND MEDIA, DIGITAL SYSTEMS AND NETWORKS

INTERNATIONAL TELEPHONE CONNECTIONS AND CIRCUITS	G.100–G.199
GENERAL CHARACTERISTICS COMMON TO ALL ANALOGUE CARRIER-TRANSMISSION SYSTEMS	G.200–G.299
INDIVIDUAL CHARACTERISTICS OF INTERNATIONAL CARRIER TELEPHONE SYSTEMS ON METALLIC LINES	G.300–G.399
GENERAL CHARACTERISTICS OF INTERNATIONAL CARRIER TELEPHONE SYSTEMS ON RADIO-RELAY OR SATELLITE LINKS AND INTERCONNECTION WITH METALLIC LINES	G.400–G.449
COORDINATION OF RADIOTELEPHONY AND LINE TELEPHONY	G.450–G.499
TRANSMISSION MEDIA CHARACTERISTICS	G.600–G.699
DIGITAL TERMINAL EQUIPMENTS	G.700–G.799
DIGITAL NETWORKS	G.800–G.899
DIGITAL SECTIONS AND DIGITAL LINE SYSTEM	G.900–G.999
QUALITY OF SERVICE AND PERFORMANCE – GENERIC AND USER-RELATED ASPECTS	G.1000–G.1999
TRANSMISSION MEDIA CHARACTERISTICS	G.6000–G.6999
DATA OVER TRANSPORT – GENERIC ASPECTS	G.7000–G.7999
General	G.7000–G.7099
Transport network control aspects	**G.7700–G.7799**
ETHERNET OVER TRANSPORT ASPECTS	G.8000–G.8999
ACCESS NETWORKS	G.9000–G.9999

For further details, please refer to the list of ITU-T Recommendations.

ITU-T Recommendation G.7713/Y.1704

Distributed Call and Connection Management (DCM)

Summary

This Recommendation provides the requirements for the distributed call and connection management for both the User Network Interface (UNI) and the Network Node Interface (NNI). The requirements in this Recommendation specify the communications across interfaces to effect automated call operations and connection operations. Items covered in this Recommendation include:

– attribute specifications;
– message specifications;
– signal flows;
– DCM state diagrams; and
– management of DCM.

This Recommendation does not cover any aspects related to routing or automatic discovery.

Revisions to this Recommendation include the following changes:

– Aligning with revised ITU-T Rec. G.8080/Y.1304 (2006), *Architecture of the automatically switched optical network*;
– Merging Amendment 1 of ITU-T Rec. G.7713/Y.1704 into the main text;
– Adopting requirements for resilience and restoration (rerouting).

Source

ITU-T Recommendation G.7713/Y.1704 was approved on 7 May 2006 by ITU-T Study Group 15 (2005-2008) under the ITU-T Recommendation A.8 procedure.

Keywords

ASON, automatically switched optical network, automatic switched transport network, call control, connection management, connection operations, distributed call and connection management, external NNI, internal NNI, signalling, user network interface.

CONTENTS

Page

1 Scope ... 1

2 References ... 1

3 Terms and definitions ... 3

4 Abbreviations ... 4

5 Conventions ... 6

6 DCM requirements ... 7

6.1 Distributed call and connection management operations procedures ... 8

6.2 Signalling controller resilience ... 16

6.3 DCM signal flow – Exception handling ... 17

6.4 Restoration ... 28

7 DCM attributes list ... 31

7.1 UNI attributes list ... 33

7.2 I-NNI attributes list ... 36

7.3 E-NNI attributes list ... 38

8 DCM message sets ... 39

8.1 UNI messages ... 41

8.2 I-NNI messages ... 44

8.3 E-NNI messages ... 48

9 DCM state diagrams ... 51

9.1 Call state ... 53

9.2 Connection state ... 61

10 Management of the call and connection controller function ... 68

10.1 Setting up a connection ... 72

10.2 Releasing a Connection ... 73

Introduction

This Recommendation forms part of a suite of Recommendations covering the full functionality of the automatically switched optical network (ASON).

ITU-T Recommendation G.7713/Y.1704

Distributed Call and Connection Management (DCM)

1 Scope

This Recommendation covers the areas associated with the signalling aspects of automatically switched optical network (ASON). Specifically, it provides the signalling requirements for the communications of call controller, connection controller and link resource manager. This Recommendation currently specifies operations for call setup and release based on a call having a single connection per call. Calls supporting multiple connections are for further study. Capability to modify calls is also for further study. Areas covered include:

– attribute specifications;

– message specifications;

– signal flows;

– DCM state diagrams; and

– management of DCM.

Other areas of ASON such as routing mechanisms, parameters associated with routing mechanisms, discovery, and naming and addressing are outside the scope of this Recommendation. This Recommendation provides the attribute and message specification, and signalling exchange that allows support for hierarchical, source and step-by-step routing.

This Recommendation uses the architecture and functional requirements as outlined in ITU-T Rec. G.8080/Y.1304 as the basis for the specification. This Recommendation aims to provide a protocol-neutral approach to describe the capability sets of the DCM. Capabilities specified in this Recommendation include support for soft permanent connections and switched connections.

In order to allow for interworking between multiple specific protocol implementations, an interworking function may need to be specified. This is currently for further study.

Transport of the DCM message sets is via a data communication network (DCN). One possible option for a DCN is described in ITU-T Rec. G.7712/Y.1703.

In order to provide an automated DCM mechanism, *a priori* knowledge of the network resources is needed. These resources may be manually provisioned or automatically discovered. Automatic discovery of the topology and the resources may be performed as per ITU-T Rec. G.7714/Y.1705.

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

– ITU-T Recommendation G.707/Y.1322 (2003), *Network node interface for the synchronous digital hierarchy (SDH)*.

– ITU-T Recommendation G.709/Y.1331 (2003), *Interfaces for the Optical Transport Network*.

– ITU-T Recommendation G.783 (2006), *Characteristics of synchronous digital hierarchy (SDH) equipment functional blocks*.

– ITU-T Recommendation G.784 (1999), *Synchronous digital hierarchy (SDH) management*.

– ITU-T Recommendation G.798 (2004), *Characteristics of optical transport network hierarchy equipment functional blocks*.

– ITU-T Recommendation G.803 (2000), *Architecture of transport networks based on the synchronous digital hierarchy (SDH)*.

– ITU-T Recommendation G.805 (2000), *Generic functional architecture of transport networks*.

– ITU-T Recommendation G.806 (2006), *Characteristics of transport equipment – Description methodology and generic functionality*.

– ITU-T Recommendation G.851.1 (1996), *Management of the transport network – Application of the RM-ODP framework*.

– ITU-T Recommendation G.852.2 (1999), *Enterprise viewpoint description of transport network resource model*.

– ITU-T Recommendation G.853.1 (1999), *Common elements of the information viewpoint for the management of a transport network*.

– ITU-T Recommendation G.872 (2001), *Architecture of optical transport networks*.

– ITU-T Recommendation G.874 (2001), *Management aspects of the optical transport network element*.

– ITU-T Recommendation G.875 (Draft), *Optical transport network (OTN) management information model for the network element view*.

– ITU-T Recommendation G.7712/Y.1703 (2003), *Architecture and specification of data communication network*.

– ITU-T Recommendation G.7714/Y.1705 (2005), *Generalized automatic discovery for transport entitles*.

– ITU-T Recommendation G.8080/Y.1304 (2006), *Architecture of the automatically switched optical network (ASON)*.

– ITU-T Recommendation G.8081/Y.1353 (2004), *Terms and definitions for Automatically Switched Optical Networks (ASON)*.

– ITU-T Recommendation M.3100 (2005), *Generic network information model*.

– ITU-T Recommendation Q.1901 (2000), *Bearer Independent Call Control protocol*.

– ITU-T Recommendation Q.2931 (1995), *Digital Subscriber Signalling System No. 2 – User-Network Interface (UNI) layer 3 specification for basic call/connection control* plus amendments.

– ITU-T Recommendation Q.2982 (1999), *Broadband integrated services digital network (B-ISDN) – Digital Subscriber Signalling System No. 2 (DSS2) – Recommendation Q.2931-based separated call control protocol*.

3 Terms and definitions

This Recommendation defines the following term:

3.1 signalling controller: A signalling controller contains the functions of connection control and/or call control.

The following terms are defined in ITU-T Rec. G.805:

– Administrative domain
– Layer network
– Link connection
– Management domain
– Subnetwork
– Subnetwork connection

The following term is defined in ITU-T Rec. G.806:

– Management Information (MI) signal

The following terms are defined in ITU-T Rec. G.8080/Y.1304:

– Access Group Container
– Agent
– Component
– Control Domain
– Call controller
– Call Segment
– Connection controller
– Connection admission control
– Hard Rerouting
– Routing controller
– Link resource manager
– Policy
– Protocol controller
– Rerouting Domain
– Restoration
– Routing Domain
– Soft Rerouting
– Soft permanent connection
– Switched connection
– Subnetwork point
– Subnetwork point pool
– Transport Domain
– UNI Transport Resource Identifier

The following term is defined in ITU-T Rec. G.7712/Y.1703:

– Data communication network

The following terms are defined in ITU-T Recs G.852.2 and G.853.1:

– Connection termination point. (Examples of connection termination points for technology specific instances may be found in ITU-T Recs G.784 (for SDH) and G.874 (for OTN).)
– Trail termination point. (Examples of trail termination points for technology specific instances may be found in ITU-T Rec. G.784 (for SDH) and ITU-T Rec. G.874 (for OTN).)
– Resource

The following term is defined in ITU-T Rec. G.783:

– TPmode/PortMode

4 Abbreviations

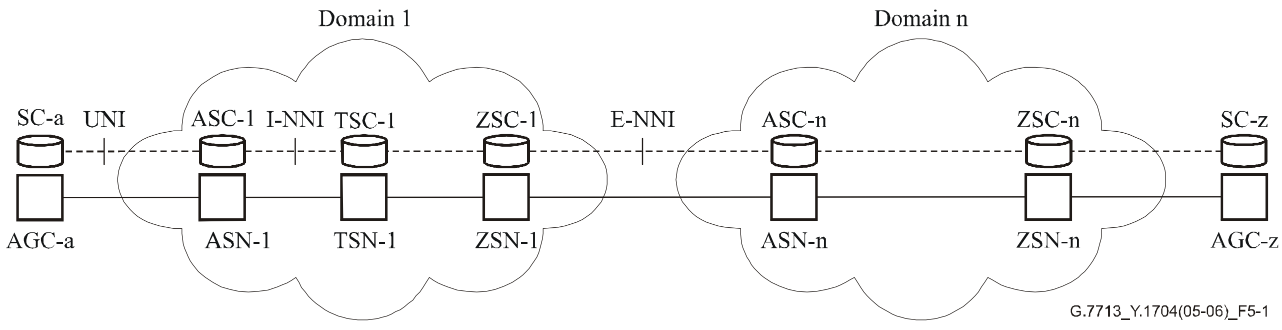
This Recommendation uses the following abbreviations:

ACC-n	A-end CC at domain n
AD	Administrative Domain
AGC	Access Group Container
AGC-a	A-end AGC
AGC-z	Z-end AGC
ARC	Alarm Reporting Control
ASC-n	A-end Signalling Controller in domain n
ASN-n	A-end SN in domain n
ASON	Automatically Switched Optical Network
CAC	Call Admission Control
CallC	Call Controller
CC	Connection Controller
CC-a	A-end Connection Controller
CC-z	Z-end Connection Controller
CCC	Calling/Called Party Call Controller
CCC-a	A-end CCC
CCC-z	Z-end CCC
CI	Characteristic Information
CoS	Class of Service
CP	Connection Point
CPS	Connection Point Status
CR-LDP	Constraint-based Routing Label Distribution Protocol
CTP	Connection Termination Point
DCM	Distributed Call and Connection Management
DCN	Data Communication Network
E-NNI	External NNI
GoS	Grade of Service

I-NNI	Internal NNI
LC	Link Connection
LRM	Link Resource Manager
MI	Management Information
MP	Management Plane
NCC	Network Call Controller
NCC-n	NCC in domain n
NNI	Network Node Interface
PC	Protocol Controller
PNNI	Private NNI
RC	Routing Controller
RSVP-TE	Resource Reservation Protocol-Traffic Engineering
SC	Switched Connection
SC-a	A-end user Signalling Controller
SC-z	Z-end user Signalling Controller
SN	SubNetwork
SNC	SubNetwork Connection
SNCr	SubNetwork Controller
SNP	SubNetwork Point
SNPP	SubNetwork Point Pool
SPC	Soft Permanent Connection
TCC-n	Transit CC in domain n
TCP	Termination Connection Point
TSC-n	Transit Signalling Controller in domain n
TSN-n	Transit SN in domain n
TTP	Trail Termination Point
UNI	User Network Interface
ZCC-n	Z-end CC at domain n
ZSN-n	Z-end SN in domain n
ZSC-n	Z-end Signalling Controller at domain n

5 Conventions

Within the distributed connection management environment, certain roles are assigned to different agents based on their location with respect to the signalling flow. Figure 5-1 identifies these reference points.



AGC-a	A-end location	SC-a	A-end user signalling controller (CCC and CC)
AGC-z	Z-end location	SC-z	Z-end user signalling controller (CCC and CC)
ASN-1	A-end subnetwork in domain 1	ASC-1	A-end signalling controller in domain 1 (NCC and CC)
TSN-1	Transit subnetwork in domain 1	TSC-1	Transit signalling controller in domain 1 (CC)
ZSN-1	Z-end subnetwork in domain 1	ZSC-1	Z-end signalling controller in domain 1 (NCC and CC)
ASN-n	A-end subnetwork in domain n	ASC-n	A-end signalling controller in domain n (NCC and CC)
ZSN-n	Z-end subnetwork in domain n	ZSC-n	Z-end signalling controller in domain n (NCC and CC)

Figure 5-1/G.7713/Y.1704 – Reference diagram for distributed connection management

Transport plane components in Figure 5-1 are the various subnetworks and access group containers (AGCs). These define the locality to which the control plane functions are associated. They are labelled in Figure 5-1 as AGC-a, ASN-1, TSN-1, ZSN-1, ASN-n, ZSN-n, and AGC-z.

Distributed call and connection management is also known as "signalling" and this Recommendation will also use this convention. Call-related functions at end users are known as Calling/Called Party Call Controllers, or CCCs. An originating CCC is a "CCC-a" and a destination CCC is a "CCC-z". Call controllers associated with a subnetwork are Network Call Controllers (NCC) and for a particular domain n, is an "NCC-n".

Connection controllers for end users are identified as CC-a and CC-z. Within a domain n, the A-end, transit, and Z-end connection controllers are known as ACC-n, TCC-n, and ZCC-n.

A signalling controller contains the functions of connection control and/or call control. For end users, this is denoted as SC-a and SC-z. Within domain n, the A-end, transit, and Z-end signalling controllers are known as ASC-n, TSC-n, and ZSC-n. Note that TSCs usually do not have call control as shown in Figure 5-1.

An address for signalling control is assigned to the signalling controller and is used by the protocol controller to exchange information between call controllers or between connection controllers. The signalling controller address is a control address, and the signalling channel will be identified by two adjacent signalling controller names. The signalling channel is provided by DCN communication.

This Recommendation also uses the following terms:

– LC Establish: establishing a link connection refers to obtaining (or choosing) an existing link connection for usage to satisfy a connection request, i.e., removing it from a list of available link connections.

– LC Free: freeing a link connection refers to returning the link connection to the list of available link connections.

- SNC Created: creating a subnetwork connection refers to creating a connection in a connection function between two SNPs within a subnetwork.
- SNC Released: releasing a subnetwork connection refers to releasing a connection in a connection function between two SNPs within a subnetwork.
- Allocate: allocate refers to either establishing an LC or creating an SNC.
- De-allocate: de-allocate refers to either freeing an LC or releasing an SNC.
- Downstream: depending on the routing model used, downstream may refer to either the component that is the subordinate (hierarchical model) or the component that is the next hop (source-based or step-by-step model).
- Upstream: depending on the routing model used, upstream may refer to either the component that is the parent (hierarchical model) or the component that is the previous hop (source-based or step-by-step model).

6 DCM requirements

Prior to any calls being established, contracts between the requester and the provider need to be set up. This contract may specify things such as:

- Contract ID;
- Service level agreement and service level specification;
- Information required to allow policy control of a request. For example, this may include information that may be used to provide authentication and integrity.

Characteristics that impact the signalling performance may include:

- Capacity of the data communication network used for transporting the signalling messages;
- Size of the switched transport network (in terms of nodes and links);
- Total number of call requests per time period, which may include new calls, protection events, and restoration events;
- Average size of messages;
- Mix of connection types;
- Time to complete a call request;
- Percentage of requests received by the network that will be retried if the requested operation is unsuccessful;
- Additional bandwidth requirements to realize a robust messaging mechanism (e.g., time-out and retransmit);
- Synchronous versus asynchronous message transport.

In order to meet the requirements specified in ITU-T Rec. G.8080/Y.1304, and to allow for future extensions to the capabilities, the DCM mechanism must provide the flexibility and extensibility to support multiple application requirements. These include support of basic DCM capabilities, as well as an extended set of capabilities (such as supplementary services). These basic capabilities provide the necessary mechanisms for setup and release of connections.

Call and Connection management operations (i.e., signalling) occur within a control domain whose scope enables complete call and connection actions. The requirements in the subsequent clauses assume that call/connection operations are within one domain that may contain smaller domains within it. This reflects routing hierarchy which results in hierarchical routing domains. Signalling components at the edge of a routing domain use the routing components within that domain to provide routes through that domain.

6.1 Distributed call and connection management operations procedures

The call controller (CallC), connection controller (CC) and link resource manager (LRM) functions provide the supervision and management of both call and connection requests, including the primitive operations for setting up a connection, modifying a connection, and releasing a connection. To complete an operation, the CallC, CC and LRM need to interact with other components as well as interact with each other. The CallC, CC and LRM interact with the following components to setup or release a connection:

– routing controller (RC): routing controller provides route information as queried by the CC;
– call admission control (CAC) function;
– call controller (CallC);
– connection controller (CC);
– link resource manager (LRM).

As described in ITU-T Rec. G.8080/Y.1304, the calling party call controller interacts with a called party call controller by means of one or more intermediate network call controllers (NCC). The NCC function is provided at the network edge (i.e., UNI reference point) and may also be provided at gateways between domains (i.e., E-NNI reference points). The functions performed by NCCs at the network edge are defined by the policies associated by interactions between users and network, and the functions performed by NCCs at domain boundaries are defined by the policies associated by the interactions between the domains. As such, an end-to-end call is considered to consist of multiple **call segments** when the call traverses multiple domains. Each call segment could have one or more connections (LC or SNC) associated with it. This allows for flexibility in the choices of signalling, protection and recovery paradigms in different domains.

The number of connections associated with call segments may not be the same even in one end-to-end call. In Figure 6-1, the UNI call segment has one LC associated with it, the subnetwork call segment for domain 1 has two associated SNCs. This allows the network to have different policies in their domain. All transport resources in Figure 6-1 are in a single overall domain that contains domains 1 and **n**. Routing in this overall domain provides the knowledge that domains 1 and **n** need to be traversed in order to support a call between the two clients in the figure.

Note that both calls and connections could be across intra-carrier E-NNI reference points. The concept of call segments and call/connection separation enables the following applications:

– Domain-based protection. The number of SNCs could be different between domains.
– Domain-based restoration. SNC failure may not cause an LC to go down, and a rerouting procedure could be provided by network to restore the failed SNC (refer to ITU-T Rec. G.8080/Y.1304).

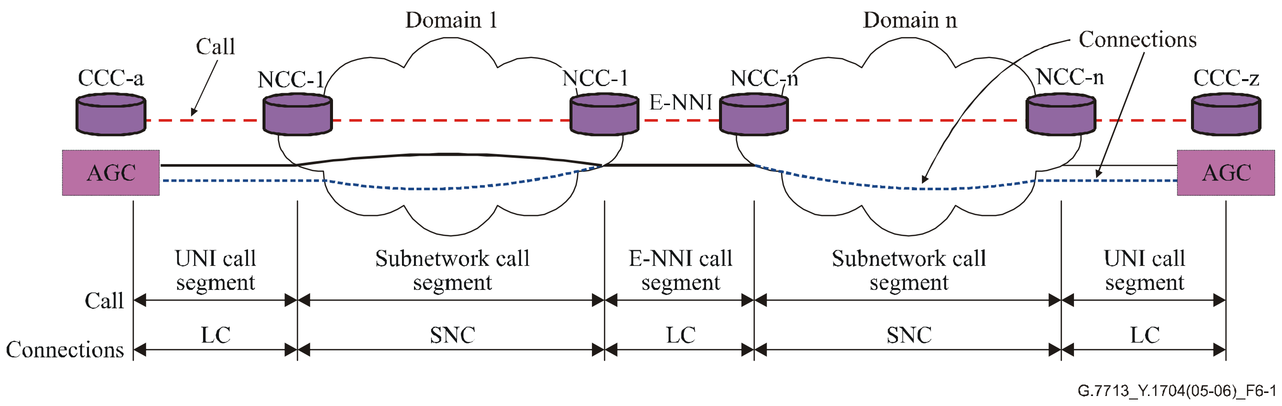


Figure 6-1/G.7713/Y.1704 – Call segments and connections

The NCC at domain boundaries will also allow each domain to have independent functions, e.g., one domain could have 1 + 1 protection capability whilst another domain does not.

The NCC and CC at the network edge and boundaries perform different functions.

The call controllers perform the following:

– The NCC correlates the SNCs to the call.
– The NCC works with the Calling/Called Party Call Controller at the network edge to correlate LC(s) to the call.
– The NCC works with its peer NCC at domain boundaries to correlate LC(s) to a call.
– The NCC correlates the LC and SNCs that are associated with the same call.
– The CC establishes the connections that are associated to each call segment.

As the communication between the controllers are defined as an external interface in ITU-T Rec. G.8080/Y.1304, messages are defined in this Recommendation to help the exchange of information.

The end-to-end establishment of a call involves requesting the call, requesting connections, and setup of different types of resources to create a connection. Figure 6-2 illustrates the connection that is set up to support a call.

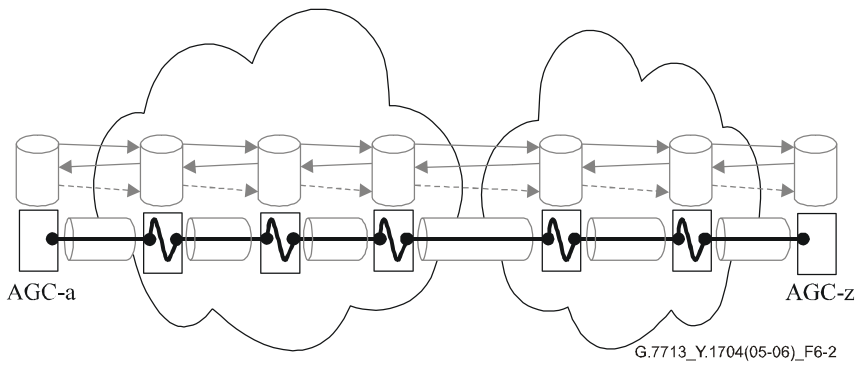


Figure 6-2/G.7713/Y.1704 – LC and SNC establishment for a call request

The following resources are used for call setup:

– Subnetwork point (SNP);
– SNP pool (SNPP);
– Link connection (LC).

The LC is established by the allocation of SNPs, which may be negotiated between LRMs. This then allows the CC to create an SNC. Allocation of SNPs may be represented as a change of state of the SNP (e.g., from AVAILABLE to PROVISIONED; note that SNPs with state of POTENTIAL or BUSY cannot be used for connection setup). SNP states are described in ITU-T Rec. G.8080/Y.1304.

Figure 6-3 illustrates the establishment of LC by the LRMs.

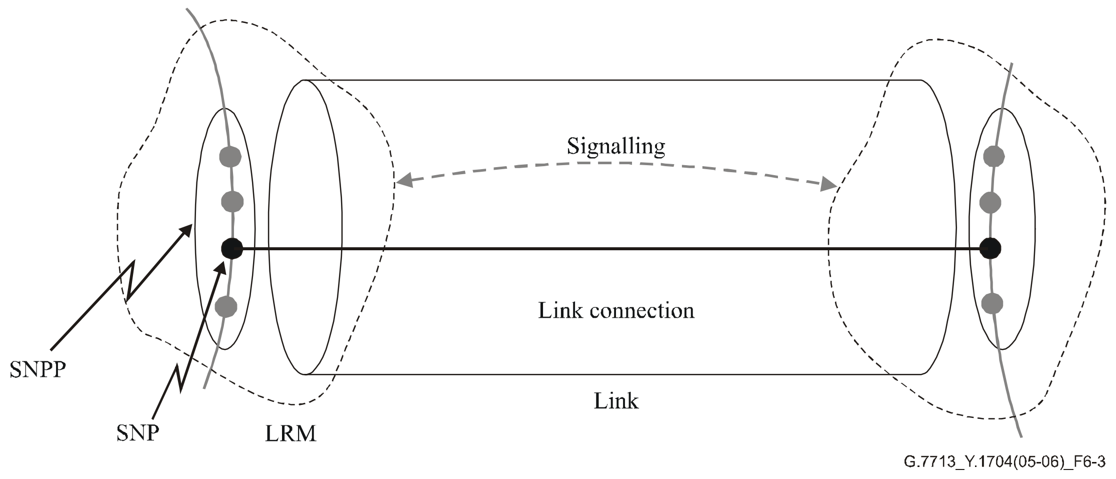


Figure 6-3/G.7713/Y.1704 – Establishment of link connection by allocating the SNPs

NOTE 1 – In the case of establishing an LC, alternate behaviours are possible, e.g., for user → network signalling, the user may also specify the LC to use (i.e., the user specifies the SNP); however, the network may choose an alternate LC to use, i.e., specify a different SNP.

In order to set up an SNC, SNPs must already exist and be identified by the LRM in order to bind these SNPs to create an SNC. This involves the LRM negotiating with the upstream LRM for an SNP (which may represent an LC) and negotiating with the downstream LRM for an SNP (which may represent an LC). These incoming and outgoing LCs (and their associated ingress and egress SNPs) identify the SNPs that are used to create the SNC.

NOTE 2 – Choosing resources for a connection operation does not imply allocation of these resources. Resource allocation may occur at any phase of the signalling, e.g., allocation may occur during the initial request or during response to the request. In addition, these resources may first be "reserved" prior to allocation. Reserved in the context of call setup refers to identifying the resources that are available for use, but not committing these resources until the allocation phase occurs. Using reservation prevents another request from identifying the same resource for use, and it avoids the resources from undergoing state changes if the call was denied. All this may be handled as part of setting the status of SNPs and interactions with LRM components. ITU-T Rec. G.8080/Y.1304 provides the list of possible states of the SNPs.

Setting up of an SNC is a process that occurs internal to a subnetwork and is controlled by the CC. An SNC is created after determining the SNPs for both ingress and egress connection points. The ingress and egress SNPs are identified as part of the establishment of the LCs (via the LRM). Figure 6-4 depicts an SNC that has been set up along with the associated connection controller and the SNP pair that are involved with setting up an SNC.

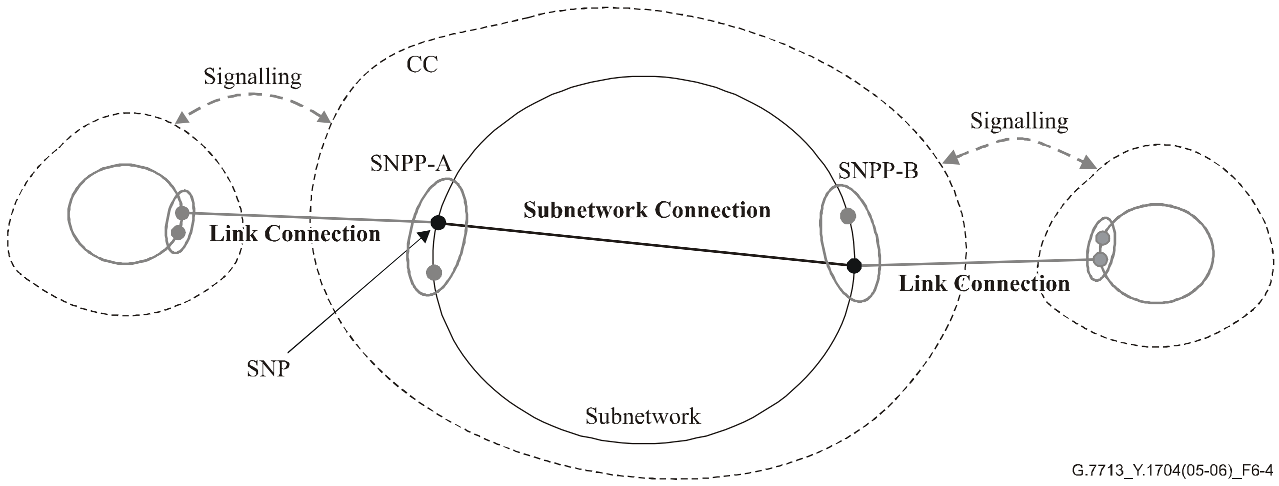


Figure 6-4/G.7713/Y.1704 – Creation of subnetwork connection after link connection establishments

6.1.1 Process for call request

To support switched connection (SC) service, a call request is initiated by the A-end user request agent (CCC-a) via a "call setup request" message sent by the CCC-a (user CallC) to a calling party call controller. This call request specifies the information associated with the call that the user is requesting. The information may include service-related information and policy-related information. This information is received by the CallC within the ASC. The CallC next processes the call request and interacts with other components within the ASC to support the call request.

To support soft permanent connection (SPC) service, the client call controller is handled by the management plane, with management requesting the NCC to establish calls. The transport plane location of this call endpoint is an SPC endpoint, and a UNI Transport Resource Identifier is associated with those resources. Where there is a common transport network address administration for SNPPs, a UNI Transport Resource Identifier is not required for the SPC endpoint.

NOTE – To provide error handling and prevent non-deterministic transitions, a timeout (timer expiry) mechanism is required. The timeout mechanism is initiated by the user upon call request (for both call setup request and call release request). Specific details of exception handling related to timeout is specified in 6.3.

6.1.1.1 SPC and SC interworking

An SPC endpoint is associated with an NCC that supports SPC service initiation and termination. Similarly an SC endpoint is associated with an NCC that supports SC service initiation and termination. Each type of endpoint has UNI Transport Resource Identifiers that are used to identify the transport resources between the client and the network at the UNI reference point.

An SPC endpoint can make calls to an SC endpoint using the UNI Transport Resource Identifier of that SC endpoint. Similarly, an SC endpoint can make calls to an SPC endpoint using the UNI Transport Resource Identifier of that SPC endpoint. Either SPC or SC endpoint may release the call. Procedures described in later clauses apply to this type of call for all ingress to egress NCC communication.

A particular SNP for the SPC endpoint can be specified in either calling or called cases. If left unspecified, the CC associated with the SPC endpoint is free to assign the SNP.

6.1.1.2 Setting up a call

Figure 6-5 illustrates the setup of a call, and the associated signal flows between the relevant components.

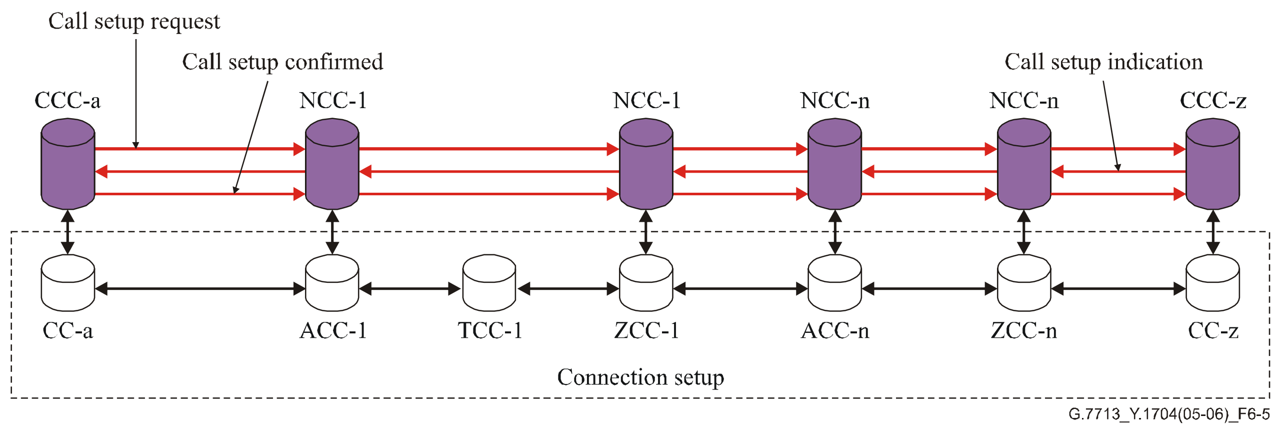


Figure 6-5/G.7713/Y.1704 – Call setup request processing: Logical request progression

For a call request to set up a call, the steps include:

– The Calling Party Call Controller (CCC-a) requests call setup. At the ingress NCC-1, processes are initiated to check the call request (this may include checking for authentication and integrity of the request as well as constraints placed by policy decisions). The request is also sent to the intermediate Network Call Controllers. Processes included in the egress NCC (NCC-n associated with ZCC-n in Figure 6-5) may include verifying that the call request is accepted end-to-end (e.g., request for CCC-z call verification).

– Upon successful checking, the Calling Party Call Controller (CCC-a) continues the call setup request by initiating a connection setup request to the CC. The process for connection setup request is described in 6.1.2. Note that, based on different protocol design decisions, initiation of connection setup request may occur in a different order as shown in Figure 6-5. The requirement is that a network connection is set up before the call is completed.

– Upon successful indication by the connection setup request process (across all call segments) the call setup request is successfully completed, and transfer of user characteristic information may begin.

If the connection setup request process was unsuccessful, a call denied notification is sent to the user.

An interface exists between a call controller and connection controller in the ITU-T Rec. G.8080/Y.1304 component description for initiating a connection request, but is not described in this Recommendation.

6.1.1.3 Releasing a call

Figure 6-6 illustrates the release of a call, and the associated signal flows between the relevant components.

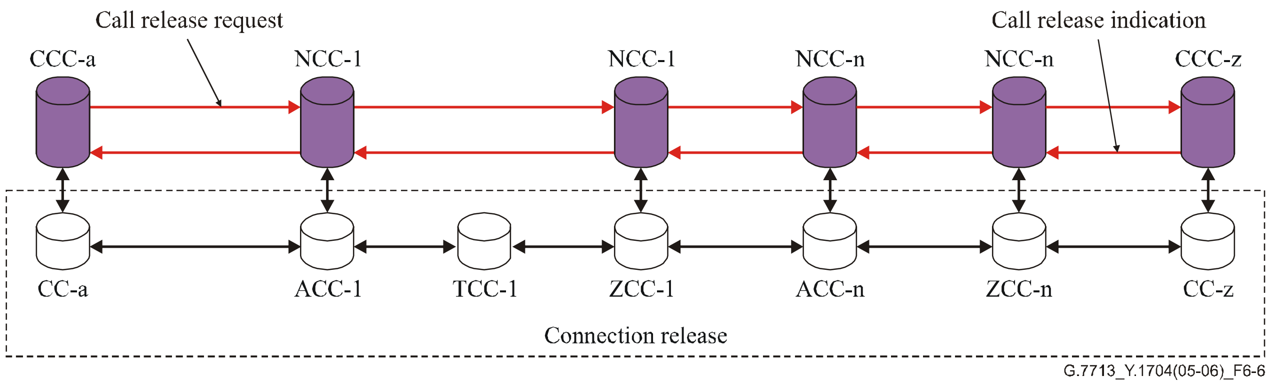


Figure 6-6/G.7713/Y.1704 – Call release request processing: Logical request progression

Any call controller may initiate a call release request. A call release request (upon verification) must always result in a successful call release. However, any defects associated with the release request may be subsequently reported to a management system (including specific information regarding any partial connections not released), and procedures may be in place to prevent access/use of the connection that was unsuccessfully released. For a call release request originating from the calling party call controller as in Figure 6-6:

– Check the call release request at the ingress Network Call Controller (ingress NCC-1). This may include checking for authentication and integrity of the request, as well as constraints placed by policy decisions.

– Upon successful checking, the call release request continues by initiating a connection release request. The process for connection release request is described in 6.1.2. Note that based on different protocol design decisions, initiation of connection release request may occur in a different order as shown in Figure 6-6. The requirement is that a connection is released before a call is released. If there are multiple connections associated with a call segment, all of them are released.

– Upon indication by the connection release request process(es), the call release request is successfully completed.

While connection setup may be denied and thus result in call denial, a connection release that is denied (e.g., due to inability to de-allocate resources, release an SNC or free an LC) results in notification to the MP; however, the call release request should indicate successful release of a call. This assumes that the call release request has been successfully checked prior to connection release being initiated.

NOTE – Depending on the "characteristics" of the transport network (e.g., whether monitoring and trace is enabled), race conditions may occur between the call release request message and the connection release request. Based on this race condition between the signalling progression from CCC-a to CCC-z, and the transport signal (e.g., unequipped or OCI) progression from AGC-a to AGC-z, certain alarms may be raised at downstream subnetworks. To support such an environment, a mechanism is needed to allow for disabling/enabling of the monitoring/trace capabilities associated with the call prior to de-allocation of connections. For example, this may include initiation of ARC or TPmode/PortMode process prior to any initiation of connection release request. Defect reporting suppression may be needed to prevent triggering the protection/restoration process.

6.1.2 Process for connection request

A connection request is initiated as a result of a call request process. The connection request performs the coordination to set up and release connections and the allocation and de-allocation of resources to effect the connection.

Figure 6-7 illustrates the end-to-end progression of the signalling and connection request that sets up the resources to create a connection and complete a call.

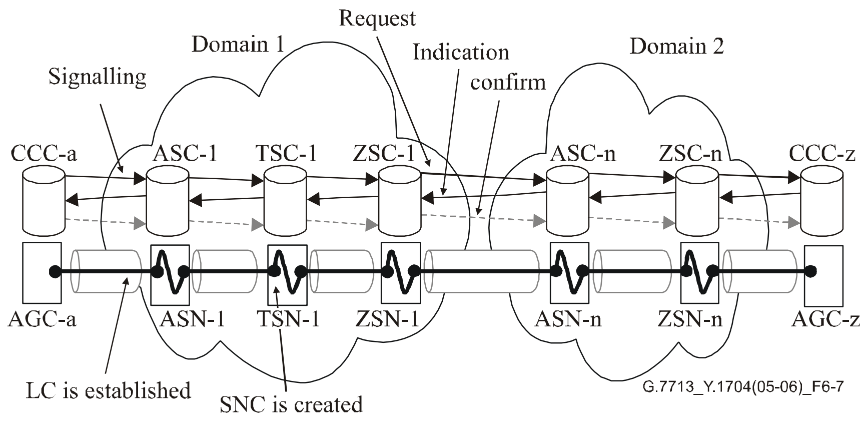


Figure 6-7/G.7713/Y.1704 – LC establishment and SNC creation for a connection setup request

Figure 6-8 illustrates the end-to-end progression of the signalling and connection request that de-allocates the resources to release a connection and release a call. Note that the sequence for the connection release, i.e., processes for de-allocating SNCs and LCs, may occur in different orders (e.g., de-allocate SNC-LC-SNC etc. in sequence, or de-allocate all LCs first, then all SNCs).

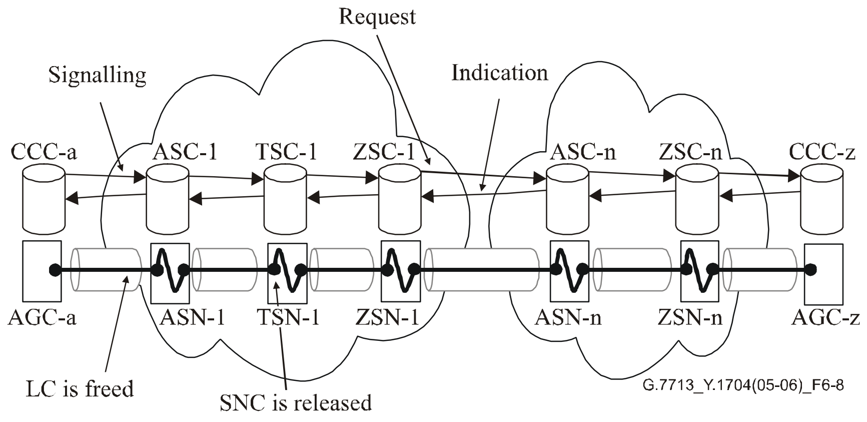


Figure 6-8/G.7713/Y.1704 – Freeing LC and releasing SNC for a connection release request

6.1.2.1 Process for setting up connections

The following processes are performed for the connection setup:

– According to 6.1.1.2, a setup call request has been verified and allowed to proceed.

– From the call request, CCC-a LRM specifies the LC negotiated to be established between AGC-a and ASN-1. This may be in the form of ASN-1's egress SNP ID.

– At ASC-1 CC, an ingress SNP is identified as determined by the ingress SNP ID (the ingress SNP ID is identified by mapping AGC-a's egress SNP to ASN-1's ingress SNP).

ASC-1 LRM negotiates with TSC-1 LRM to establish an LC connecting ASN-1 to TSN-1 (TSN-1 is determined by route information as provided by an RC or by received information from an upstream CC). Upon successful establishment of this LC, an egress SNP is identified by the LRM. An SNC is created that connects the ingress SNP to the egress SNP. The state of the SNP pair is updated to: provisioned. The CC then continues the connection setup process by communicating with the downstream CC.

– At TSC-1 CC, an ingress SNP is identified as determined by the ingress SNP ID. TSC-1 LRM negotiates with ZSC-1 LRM to establish an LC connecting TSN-1 to ZSN-1 (ZSN-1 is determined by route information as provided by an RC or by received information from upstream CC). Upon successful establishment of this LC, an egress SNP is identified by the LRM. An SNC is created that connects the ingress SNP to the egress SNP. The state of the SNP pair is updated to: provisioned. The CC then continues the connection setup process by communicating with the downstream CC.
– This process is continued until the connection request reaches CCC-z.
– At CCC-z CC, an ingress SNP is identified as determined by the ingress SNP ID. After the CC processes the connection request, a response message is sent to indicate that the connection is processed.
– Optionally, once the CCC-a CC receives the indication, a third message is sent from the CC to signal confirmation of the connection.

When setting up a connection, several high level behaviours may be specified for the network:

– If the route cannot be set up, then the subnetwork responds with a denied connection message.

For bidirectional connections, it shall be possible to specify the same SNP index values for a given CP handling both directions. For example, the same timeslot number in both directions on a port on a transport network element.

6.1.2.2 Process for releasing connections

Releasing a connection reverses the process for setting up a connection. A call release request is initially signalled and processed. The following processes are performed for the connection release:

– According to 6.1.1.3, a release call request has been verified and allowed to proceed. Upon indication to proceed with the release connection, the release process is initiated at the ASC-1 CC.
– From the call request, the agent that initiated the release call request identifies the call to be released.
– At ASC-1 CC, the SNC is released. This involves de-allocating the SNPs. ASC-1 LRM signals TSC-1 LRM to free the LC used by TSN-1 for the call. The state of the SNP pair is updated to: available. The CC then continues the connection release process by communicating with the downstream CC.
– At TSC-1 CC, the SNC is released. TSC-1 LRM signals ZSC-1 LRM to free the LC used by ZSN-1 for the call. The state of the SNP pair is updated to: available. The CC then continues the connection release process by communicating with the downstream CC.
– This process is continued until the connection release request reaches CCC-z CC.
– At CCC-z CC, the LC used for the call is freed. After the CC processes the connection release request, a response message is sent to indicate the connection release is processed.

6.2 Signalling controller resilience

Signalling controller resilience refers to its ability to continue operating under failure conditions. The signalling controller could be located at the UNI, I-NNI, E-NNI to support call and/or connection establishment and teardown procedures. Its operation depends upon the Data Communication Network (DCN), the transport plane, the management plane and the internal components of the control plane itself.

6.2.1 Signalling controller failure detection and indication

The signalling controller is considered to have failed if the control plane loses the communication to the transport plane, and/or the signalling controller loses the communication to its adjacent signalling controller. The failure could be caused by DCN failure or other reasons (e.g., adjacent signalling controller defect).

The control plane must have mechanisms to detect failures of the communications to transport plane, and the signalling controller should be informed of this failure. The failure of communications between control plane and transport plane is not necessary to cause signalling channel failure.

Signalling channel maintenance must be supported by a signalling protocol. Signalling channel failure could be caused by DCN failure or adjacent signalling controller failure. The failure detection mechanism in the signalling protocol should work for either case.

When any failure happens:

– The existing completed calls and their connections must not be altered during failure and recovery time.
– The signalling channel failure must be alarmed, and the failure should be notified to other signalling controllers. The management plane may be notified if the failure persists and requires operator intervention.
– No signalling messages are accepted or processed.
– If the failure is between control plane and transport plane, and the signalling controller is still reachable, the new call request/teardown request flows or new connection request/teardown flows will be failed with proper error indication.

If the failure is caused by signalling channel down, the new call request/call teardown flows will be lost due to signalling channel failure at the UNI, E-NNI or I-NNI. The new connection request/connection teardown flow will be lost due to signalling channel failure at I-NNI.

6.2.2 Signalling controller synchronization with transport plane

When communications between control plane and transport plane become available, the signalling controller must reconstruct the call and connection state corresponding to the connections in the transport plane. A possible resynchronization sequence is:

– The link resource manager synchronizes with the transport NE state information, including the cross connection information on NE and ports.
– The connection controller then synchronizes with link resource manager to recover the connection state.
– The call controller (if it applies) then synchronizes with connection controller to recover the call state.

During this time, the signalling protocol controller might rely on another system to report the error (e.g., zero bandwidth), or maintain a proper state, e.g., vertical recovery state. In this state, all signalling messages will be accepted, but not processed. The proper messages with error indication must be sent back.

6.2.3 Signalling controller synchronization with adjacent signalling controller

When communications between signalling controllers become available, the signalling controller must synchronize the call and connection state with its adjacent entities. The signalling protocol controller must have proper state, e.g., horizontal recovery state. In this state, the messages to setup the new calls/connections or teardown the existing calls/connections might be rejected, and the signalling controller must examine whether the connection information is consistent with the connection information by its adjacency. The check should include:

– Call and/or connection ID. The call and/or connection is considered as invalid if its ID is only found in one side.

– Resources allocation for the same connection. The connection is considered as invalid if the resources allocated by two ends are not consistent.

After the invalid calls and/or connections are identified, the signalling protocol controller must send proper messages to delete them, so that the partial calls and/or connections will be cleaned up.

This resynchronization must not occur until the synchronization between control plane and transport plane is done.

6.3 DCM signal flow – Exception handling

Different levels of exceptions may occur within a switched network, impacting both the transport plane and the control plane. For example, an exception may include defects of the signalling communication network, defects of the connection controller, and misbehaviour of the connection controller:

– Defects of the signalling communication network may result due to disruption of the communications channel.

– Defects of the connection controller may result due to failure of different agents that make up the controller, e.g., failure of the connection setup agent.

– Misbehaviour of the connection controller may be a result of incorrect decoding of messages. Mechanisms for detection of a misbehaved component are outside the scope of this Recommendation.

Defect information is communicated by the CallC and CC to the management plane, plus any information specific to the detected defects. Figure 5-1 provides the reference diagram for call requests handled by the CallC. Using this network model, the following subclauses provide different scenarios, as shown from Figures 6-9 to 6-30, the signalling flow due to setup and release of calls, and the various defects that may occur during these operations.

6.3.1 Setup connection

6.3.1.1 A-end CCC UNI defect (request message)

This signalling behaviour arises due to the following cause codes:

– Unrecognized signalling information.

– Defect of the RC (e.g., to determine a route to reach AGC-z).

– Defect of the CAC (e.g., to verify policy information).

– Link connection defect between AGC-a – ASN-1.

– Subnetwork connection defect within ASN-1.

– Defect of the LRM (e.g., to map the requested bandwidth to existing transport network resources).

– Expiration of the CC timer mechanism at the ASC-1.

– Expiration of the CC timer mechanism at the CCC-a.

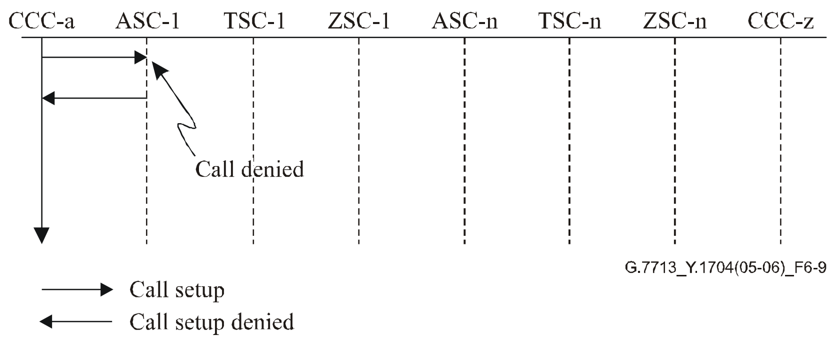


Figure 6-9/G.7713/Y.1704 – Setup → CCC-a UNI defect (call denied)

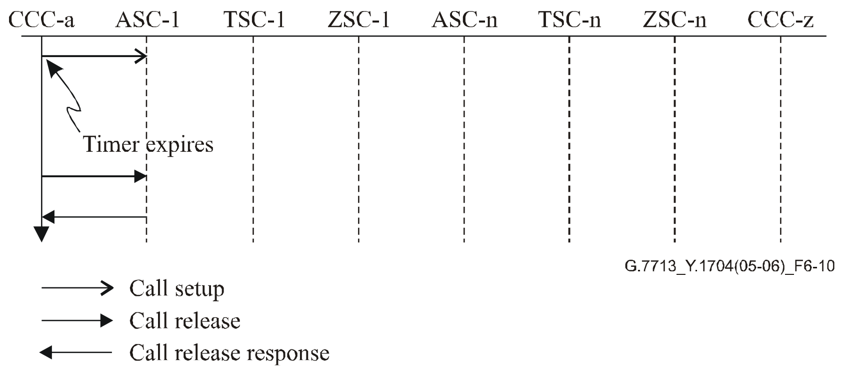


Figure 6-10/G.7713/Y.1704 – Setup → CCC-a UNI defect (timer expires)

Figures 6-9 and 6-10 show two examples of a setup rejection. In Figure 6-9, different defects occurred at ASC-1 that resulted in denied call setup.

The second example (Figure 6-10) shows a timer expiring before the user receives a response. In this case, the user drops the request. To clean up any states, and to prevent the network from setting up the call at some later time (e.g., due to error of request synchronization) an explicit release request is made to release the previous setup request.

6.3.1.2 A-end CCC UNI defect (response message)

This signalling behaviour arises due to the following cause codes:

– The response message did not reach the user requester agent.

– CC did not acknowledge connection.

Figures 6-11 to 6-14 illustrate the signal flows based on response to the defects. In the first case, from the CCC-a perspective, this is similar to a timeout; however, in this case, downstream connections have been established, i.e., resources have either been reserved or allocated. As with the above case, the user's "release" request will subsequently un-reserve or de-allocate any connections that have been committed.

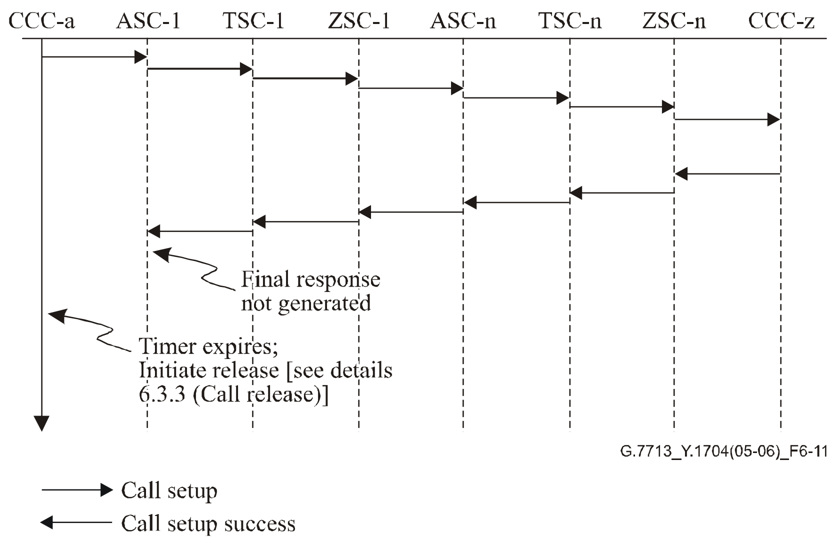

CCC-a
ASC-1
TSC-1
ZSC-1
ASC-n
TSC-n
ZSC-n
CCC-z
Final response not generated
Timer expires; Initiate release [see details 6.3.3 (Call release)]
G.7713_Y.1704(05-06)_F6-11
Call setup
Call setup success

Figure 6-11/G.7713/Y.1704 – Setup → Signalling defect of the response message (final response not generated)

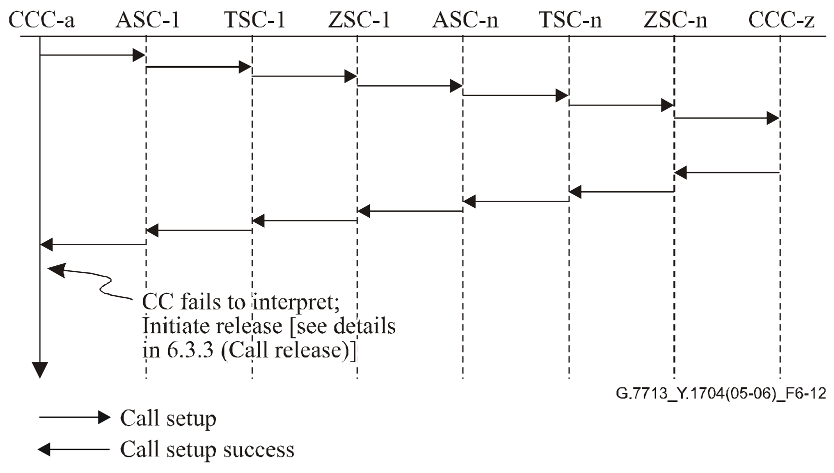

CCC-a
ASC-1
TSC-1
ZSC-1
ASC-n
TSC-n
ZSC-n
CCC-z
CC fails to interpret; Initiate release [see details in 6.3.3 (Call release)]
G.7713_Y.1704(05-06)_F6-12
Call setup
Call setup success

Figure 6-12/G.7713/Y.1704 – Setup → Signalling defect of the response message (CCC-a fails to interpret)

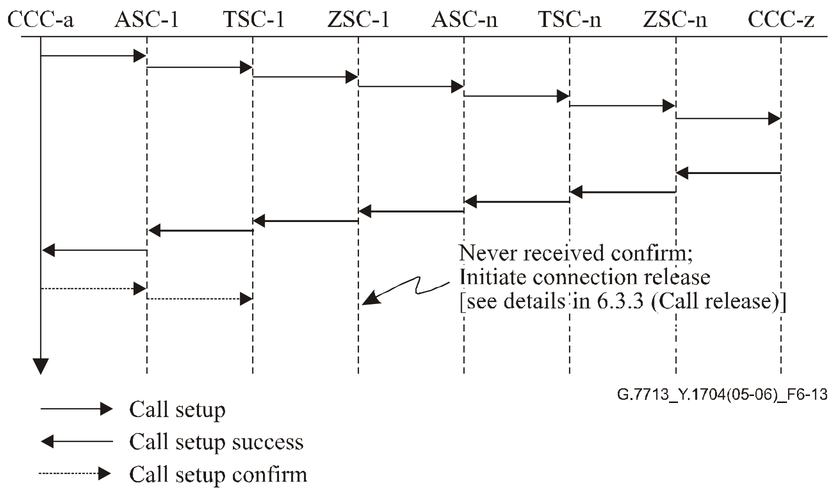

CCC-a
ASC-1
TSC-1
ZSC-1
ASC-n
TSC-n
ZSC-n
CCC-z
Never received confirm; Initiate connection release [see details in 6.3.3 (Call release)]
G.7713_Y.1704(05-06)_F6-13
Call setup
Call setup success
Call setup confirm

Figure 6-13/G.7713/Y.1704 – Setup → Signalling defect of the confirmation message (confirmation message lost during transmission)

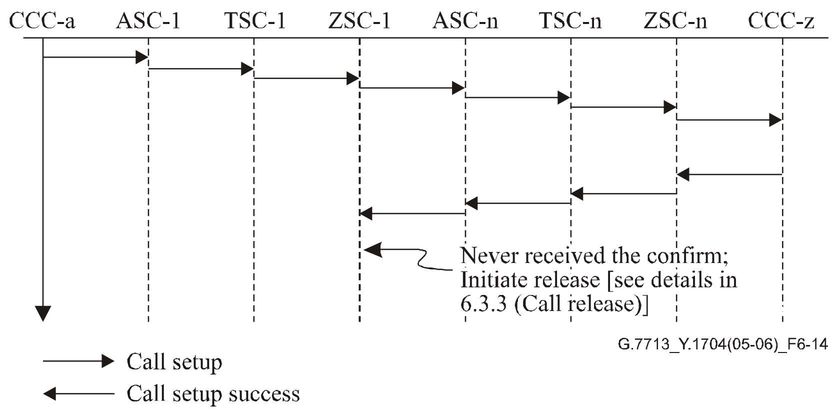


Figure 6-14/G.7713/Y.1704 – Setup → Signalling defect of the confirmation message (confirmation message never generated)

6.3.1.3 Intra-domain and inter-domain defects

This signalling behaviour arises due to the following cause codes:

- Defect of the RC (e.g., to determine a route to reach AGC-z).
- Defect of the CAC (e.g., to verify policy information).
- Link connection defect between subnetworks or between domains, e.g., ASN-1 – TSN-1 or ZSN-1 – ASN-n.
- Subnetwork connection defect in a subnetwork, e.g., TSN-1.
- Defect of the LRM (e.g., to map the requested bandwidth to existing transport network resources).
- Expiration of the CC timer mechanism.

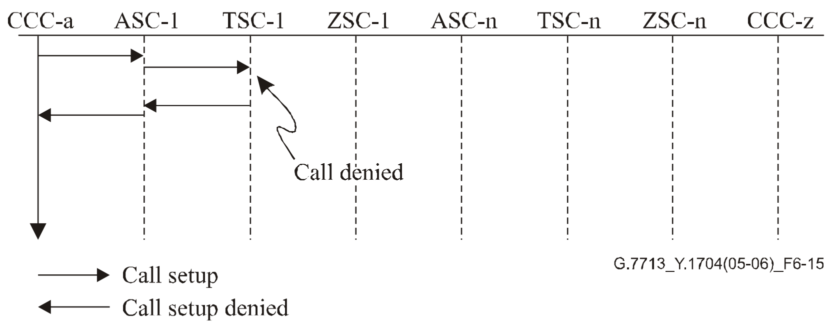


Figure 6-15/G.7713/Y.1704 – Signal flow: Setup → Intra-domain defect (call denied)

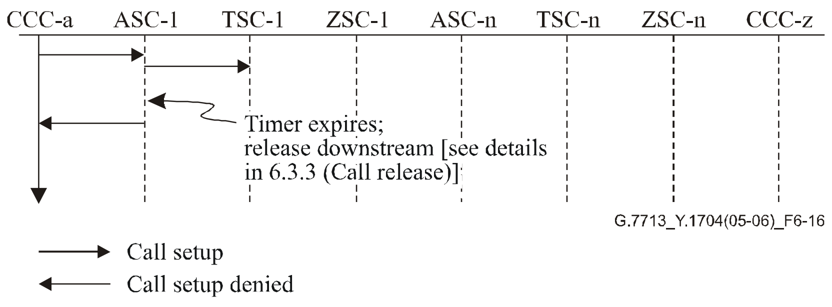


Figure 6-16/G.7713/Y.1704 – Signal flow: Setup → Inter-domain defect (timer expires)

In Figures 6-15 and 6-16, policy-based verification has succeeded; however, insufficient network resources (including failure of hard restoration actions if configured) resulted in denial of request. As a result of the service denial, the subnetworks that have reserved or allocated connections for the call will subsequently un-reserve or de-allocate the connections.

In the case where the timer expires, a setup denied request is sent upstream. This un-reserves/de-allocates the connections. In addition, a release request may also be sent downstream to prevent downstream elements from attempting to process the request.

6.3.1.4 Z-end CCC UNI defect

This signalling behaviour arises due to the following cause codes:

– Defect of the CAC (e.g., to verify policy information).

– Link connection defect between domains, e.g., ZSN-n – AGC-z.

– Subnetwork connection defect in a subnetwork, e.g., ZSN-n.

– Defect of the LRM (e.g., to map the requested bandwidth to existing transport network resources).

– Expiration of the CC timer mechanism at the ZSC-n.

– Expiration of the CC timer mechanism at the CCC-z.

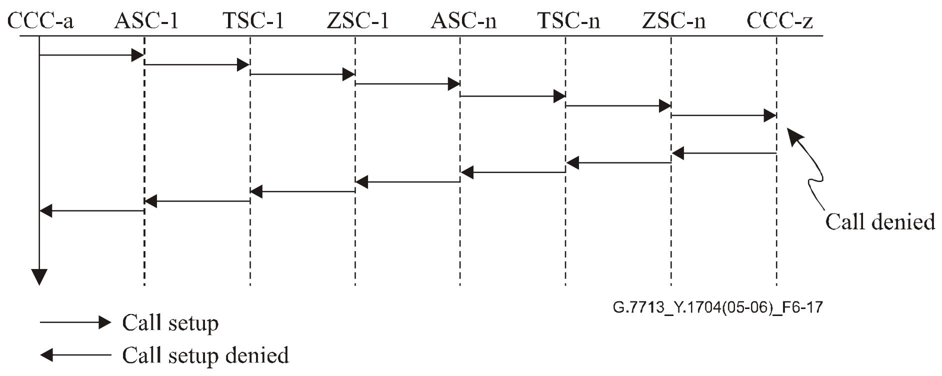


Figure 6-17/G.7713/Y.1704 – Signal flow: Setup → CCC-z UNI defect (call denied)

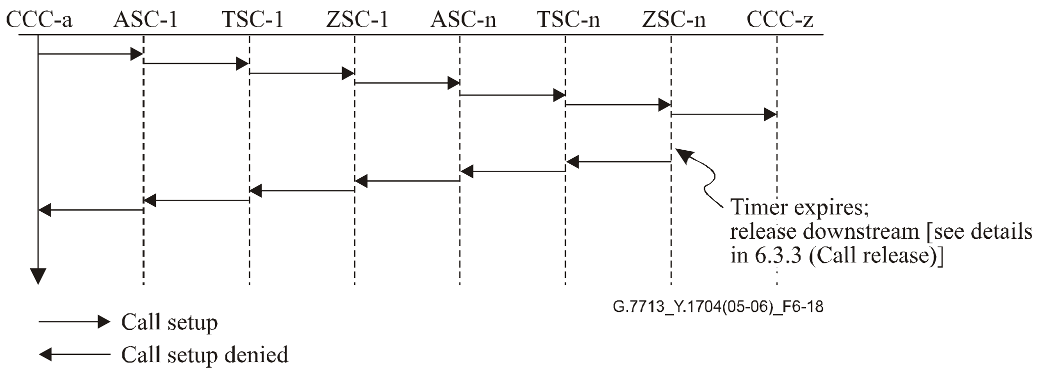


Figure 6-18/G.7713/Y.1704 – Signal flow: Setup → CCC-z UNI defect (timer expires)

In this defect scenario (see Figure 6-17), the CCC-z responds with denial of request. This may be caused by:

1) Failure to verify CCC-a's permission for connecting AGC-a with AGC-z (e.g., during call setup).

2) AGC-z has no resource (e.g., during connection setup).

As the denial propagates upstream, existing allocated or reserved resources in the subnetwork are released.

In the case where the timer expires (see Figure 6-18), a setup denied request is sent upstream. This un-reserves/de-allocates the connections. In addition, a release request may also be sent to the CCC-z to prevent the CCC-z from attempting to process the request.

6.3.2 Existing calls

Once a call has been set up, different defects may impact the call. These defects may occur to any transport network connections, and to the signalling channel. Defects and misbehaviours of the signalling network may also impact services.

In the case of a transport plane resource defect, two behaviours may be possible depending on the type of call requested:

– For a call setup using soft permanent connection, the control plane or the transport plane may attempt to recover the defected connection either via restoration or via protection switching (if protection/restoration is provided for the call). If the connection cannot be restored/protected within a time period, a notification is sent to the management system. The call remains active.

– For a call setup using switched connection, the call is released (via release of the respective connections) if hard restoration is not used or is not successful.

NOTE – During a bidirectional defect scenario, if actions need to be taken, then two CCs may be involved. A race condition may exist between the two CCs acting upon the same call. To resolve this condition, the CC that has a higher name value overrides the request of the CC with the lower name value.

6.3.2.1 Transport network connection defect (switched connection)

This signalling behaviour arises due to the following cause codes:

– Link connection defect. Link connection defect may occur as a result of a defect of SNP (point view), or as a result of a defect in the association (arc view).

– Subnetwork connection defect. Subnetwork connection defect may occur as a result of a defect of the SNP, or as a result of a defect in the association.

Figure 6-19 illustrates the case where the transport network resource failed. In this scenario, it is assumed that the call will be released as a result of the defect (i.e., no diversity/protection/restoration was specified for the connection).

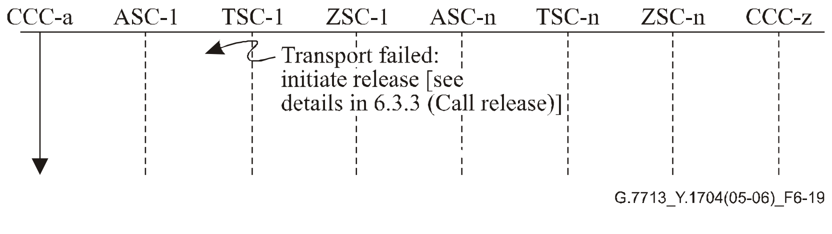

CCC-a
ASC-1
TSC-1
ZSC-1
ASC-n
TSC-n
ZSC-n
CCC-z
Transport failed: initiate release [see details in 6.3.3 (Call release)]
G.7713_Y.1704(05-06)_F6-19

Figure 6-19/G.7713/Y.1704 – Signal flow: Existing call → User transport network connection defect

Note that during the release request, a timeout may also occur (due to non-responsiveness of the receiving end). This case is handled separately (part of the exception handling for the release request). However, as the customer billing is associated with the "UP" or "DOWN" state of a call, a customer initiated release command should result in acknowledgement of the release. As there may be partial connections remaining due to defects during the release operation, the CC will notify the management plane regarding the exceptions, plus notification of the partial connections that are not released.

6.3.2.2 Transport network connection defect (soft permanent connection)

This signalling behaviour arises due to the following cause codes:

– Link connection defect. Link or link connection defect may occur as a result of a defect of the SNP or as a result of a defect in the association.

– Subnetwork connection defect. Subnetwork connection defect may occur as a result of a defect of the SNP, or as a result of a defect in the association.

Figure 6-20 illustrates the case where a connection defect was detected within the network. As a result of the defect, recovery procedures may be initiated to restore/protect the connection. This is dependent on the nature of the call, e.g., the CoS/GoS specified for the call and the type of routing applied for the call. If restoration/protection is not successful, a notification is sent to the management plane. This resulting call remains up until explicit release is received.

It is assumed that, if the connection cannot be recovered, the call remains up as a result of the defect.

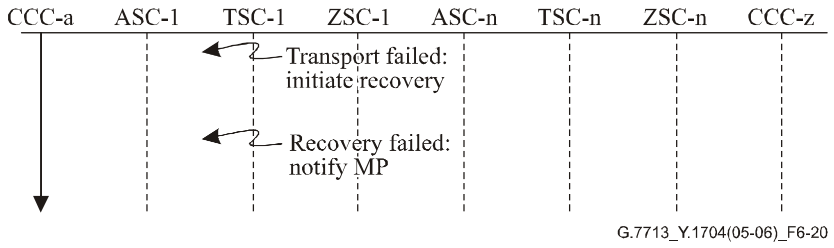

CCC-a
ASC-1
TSC-1
ZSC-1
ASC-n
TSC-n
ZSC-n
CCC-z
Transport failed: initiate recovery
Recovery failed: notify MP
G.7713_Y.1704(05-06)_F6-20

Figure 6-20/G.7713/Y.1704 – Signal flow: Existing call → Transport network connection defect

6.3.3 Call release

For exception scenarios, one of the subnetwork controllers within the network may also initiate a release request. A release request (upon verification) must always result in a successful release response to the user. However, any defects associated with the release request may be subsequently reported to a management system (including specific information regarding any partial connection not released), and procedures may be in place to prevent access/use of the connection that was unsuccessfully released.

6.3.3.1 A-end CCC or Z-end CCC initiated call release defect (request message)

This signalling behaviour arises due to the following cause codes:

– Defect of the CAC (e.g., to verify policy information).

– Defect of the LRM (e.g., to de-allocate the connection).

– Expiration of the CC timer mechanism.

– Congestion at ASC to process the message.

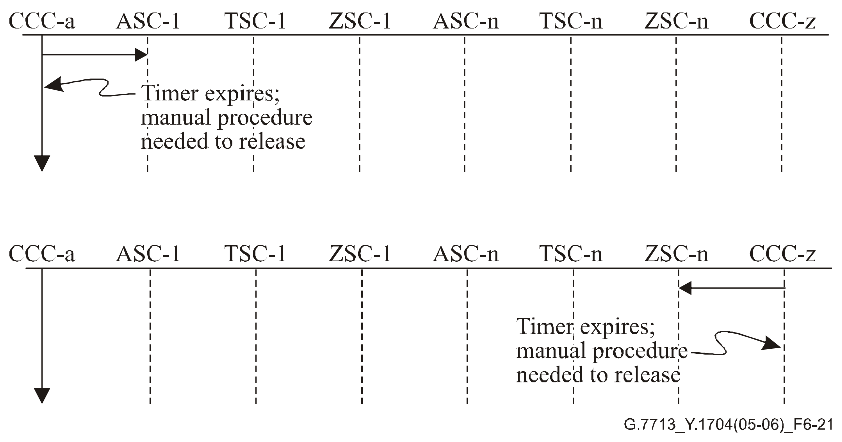

CCC-a
ASC-1
TSC-1
ZSC-1
ASC-n
TSC-n
ZSC-n
CCC-z
Timer expires; manual procedure needed to release
Timer expires; manual procedure needed to release
G.7713_Y.1704(05-06)_F6-21

Figure 6-21/G.7713/Y.1704 – Signal flow: Release call → User initiated release (user timer expires)

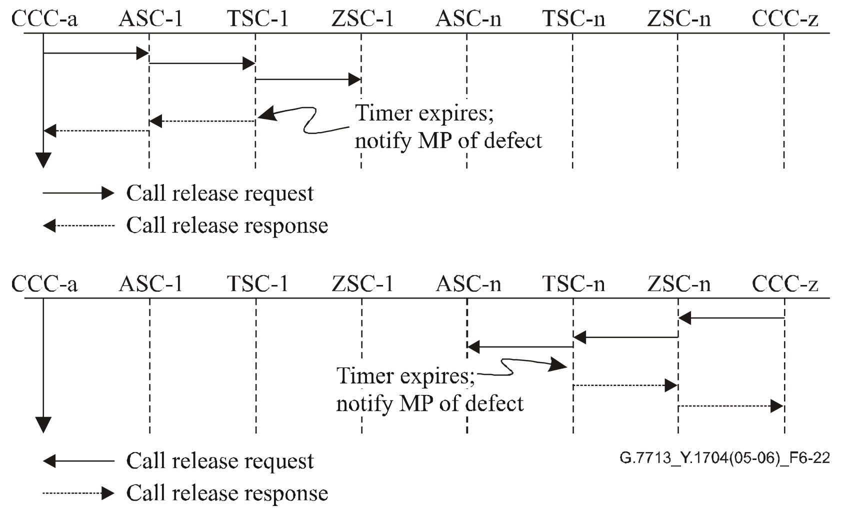


Figure 6-22/G.7713/Y.1704 – Signal flow: Release call → User initiated release (network timer expires)

In the case where the user timer expires (CCC-a or CCC-z) the user will require alternative means for releasing the call, e.g., manual process (see Figure 6-21). In the case where the network timer expires, the network sends a release confirmation response to the user (see Figure 6-22).

In addition, it notifies the management system of the defect. This allows any clean-up of the partially released connections.

Note that in the case of partial release of connections, mechanisms need to be in place to prevent AGC-a or AGC-z access to the call.

6.3.3.2 A-end CCC or Z-end CCC initiated call release (response message)

This signalling behaviour arises due to the following cause codes:

– The response message did not reach the user requester agent.

– The CC did not acknowledge release response.

Figure 6-23 illustrates the signals flows for the defects. In the first case, from the CCC-a perspective, this is similar to a timeout; however, in this case, downstream connections have been released. This introduces additional functions that the management system may need to perform as indications of the state of the connection, i.e., for timeout scenarios (in 6.3.3.1) the connections may still be up, while for this example the connection has been partially released.

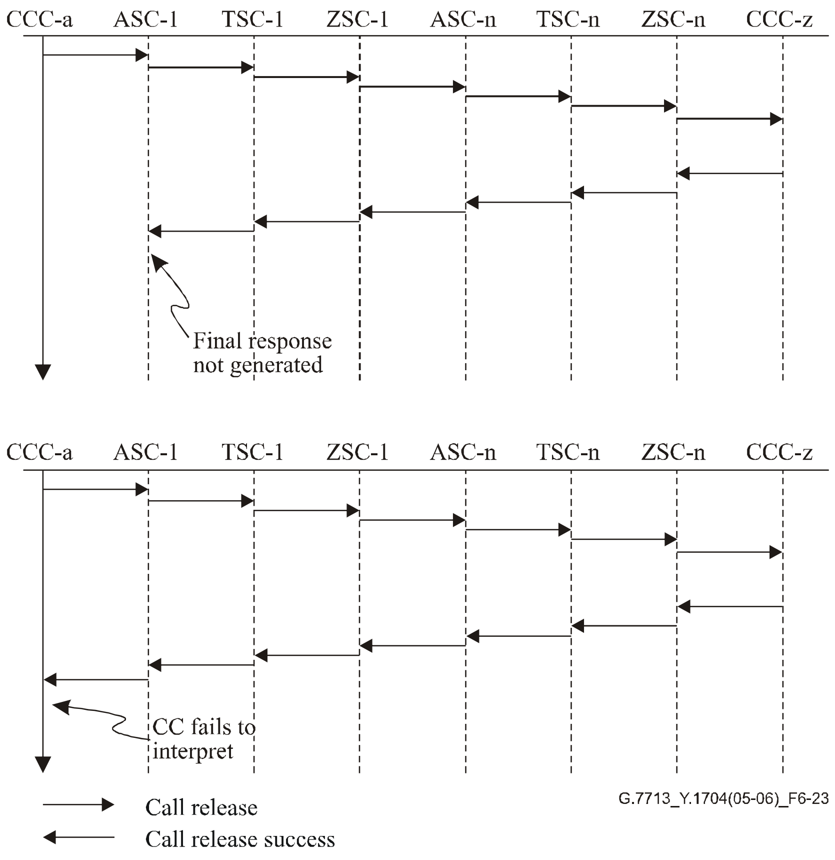


Figure 6-23/G.7713/Y.1704 – Signal flow: Release → Signalling defect of the response message

6.3.4 Connection resource release

As specified in 7.3.5.5/G.8080/Y.1304, if the network cannot setup all the connections of a new call, then any connections or partial connections that have been established will be torn down (deleted) and the call request will be rejected. This assumes that restoration actions (if permitted by policy) have also not succeeded. Cases for SCs and SPCs are shown in Figures 6-24 and 6-25 respectively.

In the scenarios described in Figures 6-24 and 6-25, the connection of a new call is denied at ASC-n. A failure notification is sent to CCC-a and the connection is released in the same time. After receiving the failure notification, CCC-a sends a call release request to release the call. In Figure 6-24, CCC-a is located at the user end (in an Access Group Container) and in Figure 6-25 it is located in the management plane. In both cases, complete call/connection separation is used in signalling and it is assumed that the call progressed to the called party. In Figure 6-25, the connection is released up to the edge of the network but the LC to user is retained. This is because that LC is configured by the management plane.

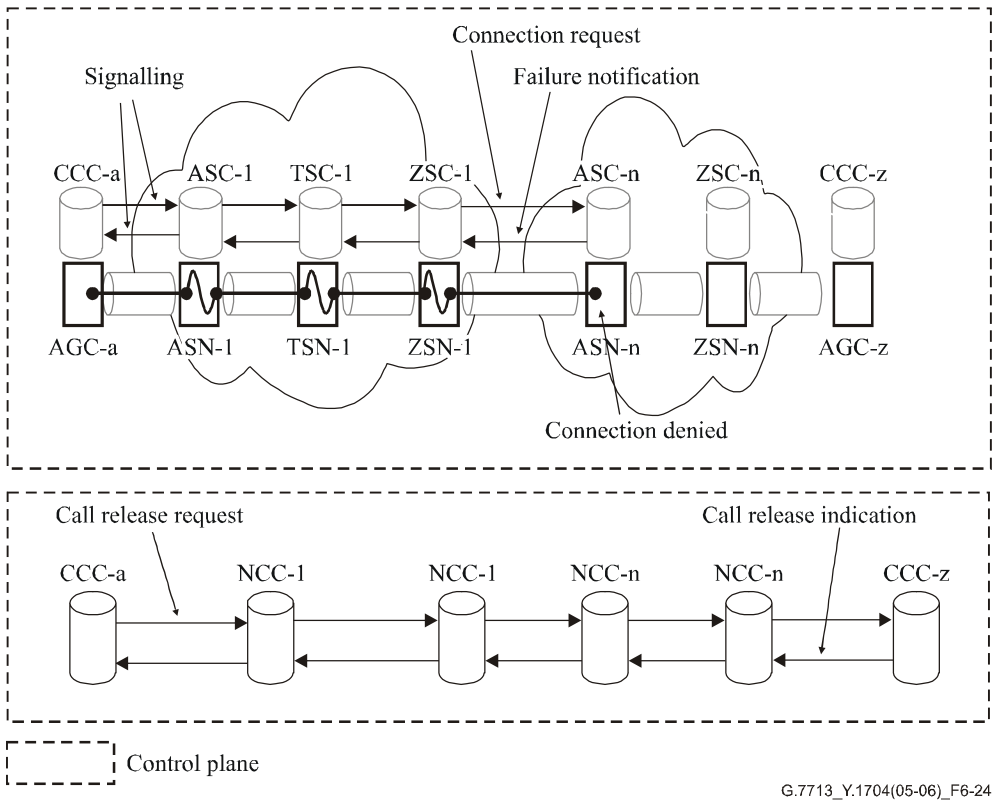


Figure 6-24/G.7713/Y.1704 – Call release after connection setup failure (SC)

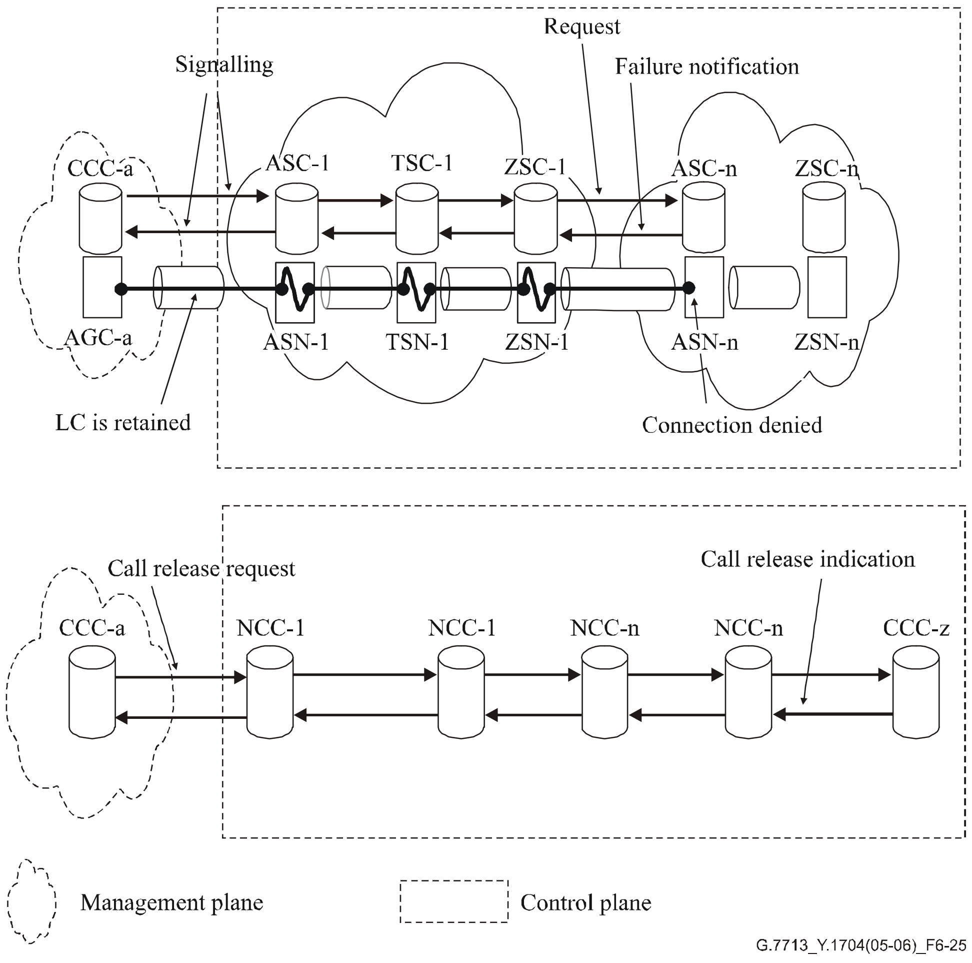


Figure 6-25/G.7713/Y.1704 – Call release after connection setup failure (SPC)

For call modification specified in 7.3.5.4/G.8080/Y.1304, if the network is unable to modify the connections, then call modification is considered to have failed. Any modified connections or partial connections will be removed and no changes will be made to the existing call. Cases for SCs and SPCs are shown in Figures 6-26 and 6-27 respectively.

In the scenarios illustrated in Figures 6-26 and 6-27, a call has been modified that results in a new connection request. When the new connection is denied at ASC-n, a failure notification is sent to the CCC-a and the connection is released. After receiving the failure notification, CCC-a sends a connection release request to release the failed connection in order to keep the state of the call unchanged.

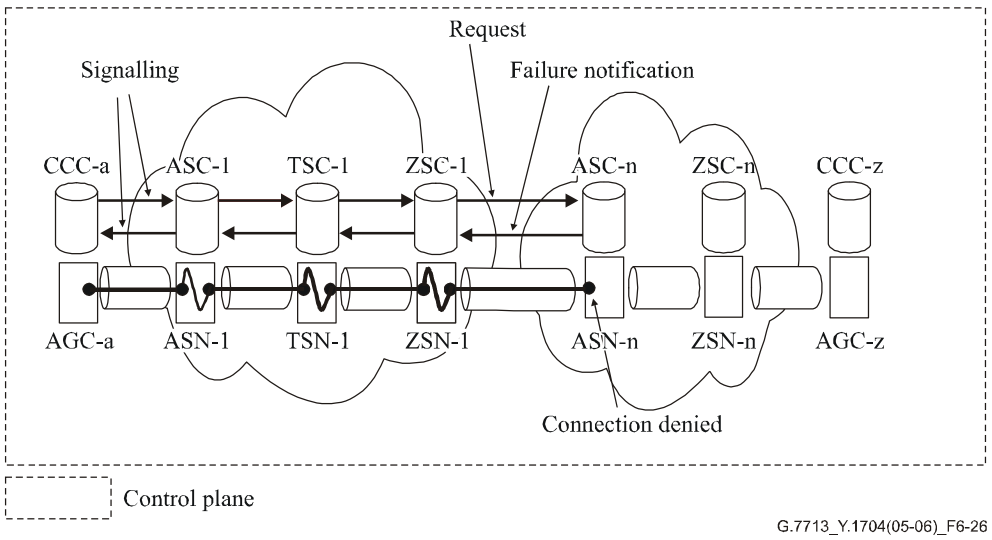


Figure 6-26/G.7713/Y.1704 – Call state unchanged after connection setup failure (SC)

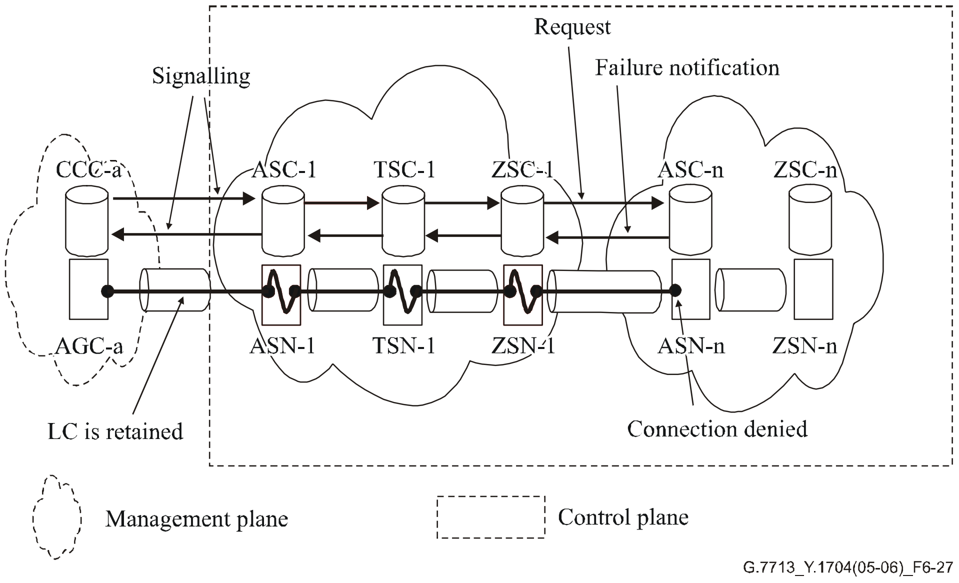


Figure 6-27/G.7713/Y.1704 – Call state unchanged after connection setup failure (SPC)

6.4 Restoration

Restoration of a call is the replacement of a connection by rerouting the connection using spare capacity. This action occurs within a rerouting domain and is enabled by policy for each call. The rerouting action consists of the establishment of a separate connection between two call controllers that are part of the same call. Once established, the new ("rerouted") connection is used in place of an existing connection for that same call segment. Note that the call itself is maintained while the connection is rerouted.

There are two types of restoration, soft rerouting and hard rerouting. In soft rerouting, the connection to be replaced is in-service and the connection is rerouted for administrative purposes. The rerouting request is made between two call controllers at the edge of a rerouting domain and once the new connection is established it can replace an existing connection in the call. If the soft rerouting action is revertive, the replaced connection is not released. Otherwise it can be released and its call and connection controller state removed.

Hard rerouting is a failure recovery function that attempts to create another connection to the destination at the edge of a rerouting domain. This is performed in response to the failure of an existing connection, and the rerouted connection replaces the connection that contained the failure. When the failure is signalled to the source call controller in the rerouting domain, the call is not released but instead, a rerouted connection is requested. The rerouting request is made between two call controllers and, once the new connection is established, it can replace the failed connection in the call. If the hard rerouting action is revertive, the resources of the failed connection are not released, nor are the connection controller states for it. The state of the failed connection is monitored and, when it recovers, the call is restored to the original connection and the rerouted connection is released. If the hard rerouting action is not revertive, the failed connection is released and its connection controller state removed.

Both soft and hard rerouting mechanisms can be used by Switched Connection and Soft Permanent Connection services. They are applied between Network Call Controllers related by a call, and not at UNI reference points.

A rerouting action can occur between two call controllers that are related by one call segment. It may also occur between two call controllers that are on either end of several contiguous call segments, in which case those call segments may be changed as a result of the path of the rerouted connection. In both cases, call parameters are sent and the call name remains the same as the call being rerouted.

6.4.1 Hard rerouting – Single rerouting domain

For a call that is provisioned with a hard rerouting service within a domain, a connection failure results in the signalling of the failure to the NCCs at the edge of the domain. The failure is not propagated outside of those NCCs and a hard restoration action is initiated. Using the example of Figure 6-7, suppose a failure occurred on a connection within Domain 1. The hard rerouting sequence is shown in Figure 6-28. When the failure notification reaches ASC-1, the call is not released but a new call/connection to ZSC-1 is initiated within the context of the same call. Note that ASC-1 and ZSC-1 contain both call and connection controllers. After a new connection is established in Domain 1, the call uses it. The call segment between ASC-1 and ZSC-1 remains before and after the rerouting action.

If the action is not revertive, ASC-1 initiates the release of the failed connection. This action may occur before or after the new connection is established. In Figure 6-28, it is shown as occurring after the new connection is established.

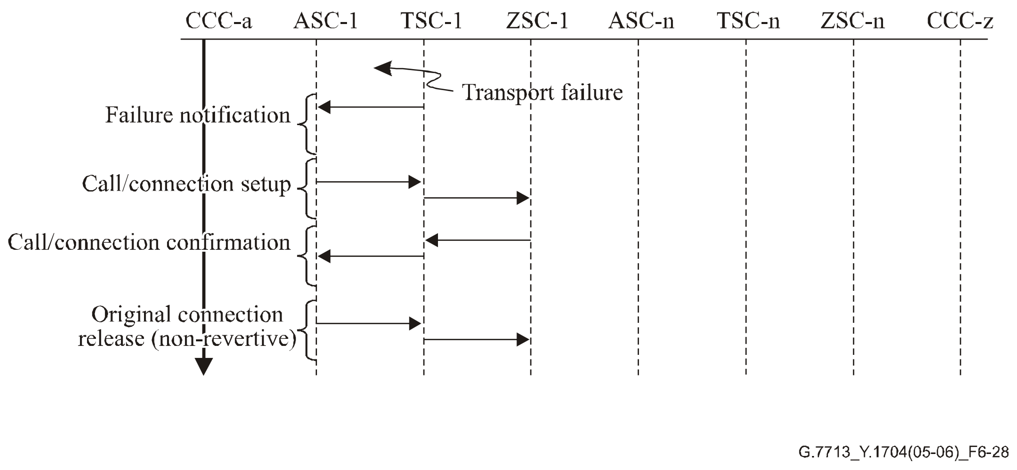


Figure 6-28/G.7713/Y.1704 – Hard rerouting – Single domain

6.4.2 Hard rerouting – Multiple rerouting domains

When a call traverses multiple domains, those domains are within a larger domain that is able to route between its containing domains. This is a property of hierarchical routing. Hard rerouting can be enabled for the call at the highest domain so that rerouting can occur between the enclosed domains. If the failed connection is to be released (non-revertive service) and the new connection involves new call controllers (i.e., new call segments), then the intermediate call controllers for the failed connection must release their call state.

In Figure 6-29 there are three domains. Two are connected by two E-NNI links (Domains 1 and 2) and are contained within a larger encompassing rerouting domain shown as Domain 0. A call is illustrated that traverses E-NNI-1. Assume that its connection traverses AGC-a, ASN-1, TSN-1, ZSN-1, ASN-2, ZSN-2, AGC-z. If a failure occurs on the link connection for the call at E-NNI-1, the rerouting sequence begins by notifying ASC-1 and is an action of the encompassing rerouting domain, Domain 0. Note that ZSC-1 cannot reroute the connection as there is only one link from its corresponding subnetwork ZSN-1 to Domain 2. Also, Domain 1 cannot reroute the connection as it does not have E-NNI-2 in its scope.

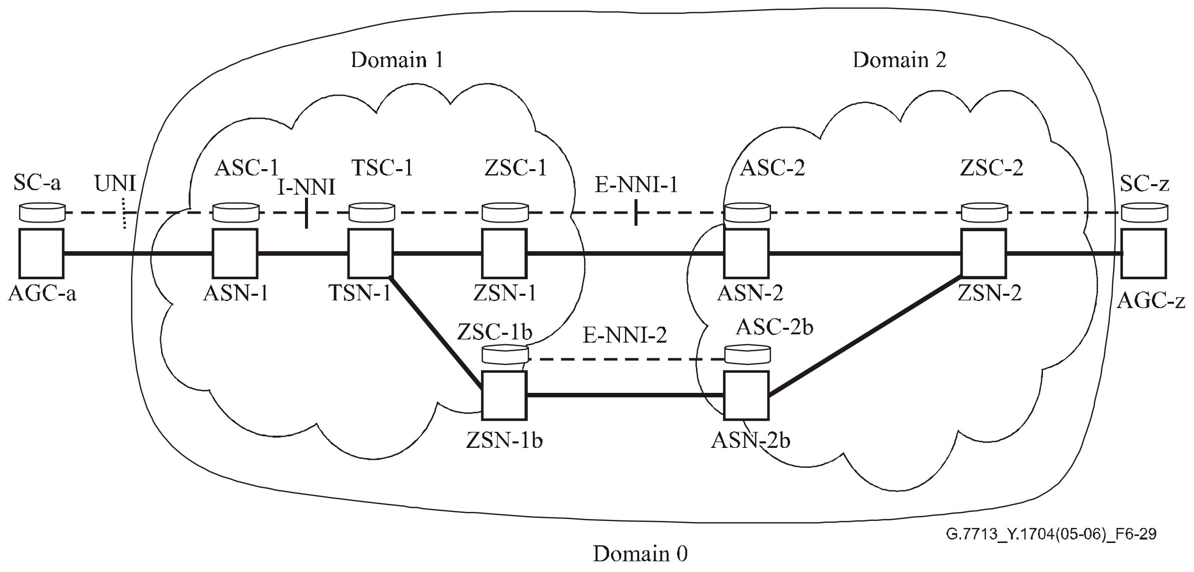


Figure 6-29/G.7713/Y.1704 – Hard rerouting – Multiple E-NNIs

At ASC-1, the call controller is configured for hard rerouting with non-revertive service. In the context of the encompassing rerouting domain, a path for a new connection is determined to traverse Domains 1 and 2 via E-NNI-2. To get to E-NNI-2, rerouting in Domain 1 is performed, and rerouting in Domain 2 is used between E-NNI-2 and the destination ZSC-2. A connection is established through ASN-1, TSN-1, ZSN-1b, ASN-2b, and ZSN-2. Three call segments are established as a result of the new connection and are ASC-1 to ZSC-1b, ZSC-1b to ASC-2b, and ASC-2b to ZSC-2. The rerouting sequence is shown in Figure 6-30.

If the action is not revertive, the call controller at ASC-1 initiates the release of the failed connection and its associated call segments that are no longer used. This removes call state at ZSC-1 and ASC-2. This action may occur before or after the new connection is established. In Figure 6-30, it is shown as occurring after the new connection is established.

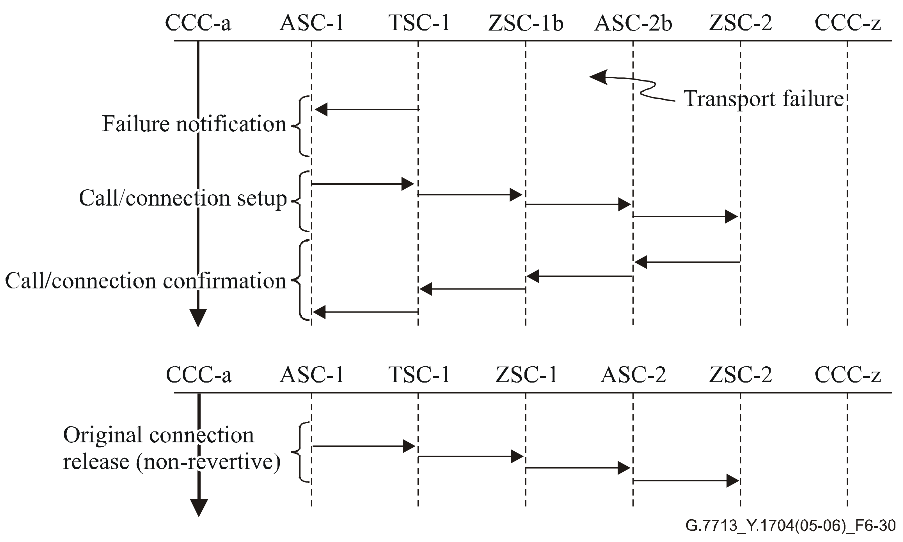


Figure 6-30/G.7713/Y.1704 – Hard rerouting – Multiple domains

7 DCM attributes list

The DCM attributes list may be separated into attributes associated with the call and attributes associated with the connection. Tables 7-1, 7-2 and 7-3 summarize a list of attributes that are considered for UNI, I-NNI and E-NNI signalling processing.

UNI signalling processing includes call attributes, and also connection attributes for setting up link connection(s) on user-to-network access links.

I-NNI signalling processing includes connection attributes. Call attributes must be exchanged between call controllers (e.g., ASC-n to ZSC-n in Figure 5-1). Many mechanisms to achieve this are not part of this architecture. I-NNI signalling might be used to exchange call attributes by piggybacking them on connection-related messages but, if so, they do not form part of I-NNI processing.

E-NNI signalling processing includes call attributes and also connection attributes for setting up link connection(s) on network-to-network access links.

All attributes represent the logical information that is exchanged across the respective interfaces to support the CCC/NCC, CC, and LRM. Note that protocol design decisions may result in aggregation (or segmentation) of some of this logical information; however, the functions supported by the information shall be present.

Table 7-1/G.7713/Y.1704 – UNI attributes list

	Attributes	**Scope**	**Call vs connection**
Identity attributes	Calling UNI Transport Resource Identifier	End-to-end	Call
	Called UNI Transport Resource Identifier	End-to-end	Call
	Initiating CC name	Local	Connection
	Initiating CallC name	Local	Call
	Terminating CC name	Local	Connection
	Terminating CallC name	Local	Call
	Connection name	Local	Connection
	Call name	End-to-end	Call

Table 7-1/G.7713/Y.1704 – UNI attributes list

	Attributes	Scope	Call vs connection
Service attributes	Calling AGC SNP ID	Local	Connection
	Calling AGC SNPP ID	Local	Connection
	Called AGC SNP ID	Local at remote end	Connection
	Called AGC SNPP ID	Local at remote end	Connection
	Directionality	Local	Call/connection
Policy attributes	CoS	End-to-end (Note)	Call
	GoS	End-to-end (Note)	Call
	Security	Local	Call/connection
NOTE – Although CoS and GoS are end-to-end in scope, their values may be changed as they cross domains. However, the policy associated with the requested service should be met.			

Table 7-2/G.7713/Y.1704 – I-NNI attributes list

	Attributes	Scope	Call vs connection
Identity attributes	Calling UNI Transport Resource Identifier	Carry transparently	Call
	Called UNI Transport Resource Identifier	Carry transparently	Call
	Initiating CC name	Local	Connection
	Terminating CC name	Local	Connection
	Connection name	Global in one domain	Connection
	Call name	End-to-end	Call
Service attributes	SNP ID	Local	Connection
	SNPP ID	Local	Connection
	Called AGC SNP ID	Carry transparently	Connection
	Called AGC SNPP ID	Carry transparently	Connection
	Directionality	Global in one domain	Call/connection
Policy attributes	CoS	Carry transparently	Call
	GoS	Carry transparently	Call
	Connection CoS	Global in one domain	Connection
	Connection GoS	Global in one domain	Connection
	Explicit resource list	Global in one domain	Connection
	Recovery	Global in one domain	Connection

Table 7-3/G.7713/Y.1704 – E-NNI attributes list

	Attributes	Scope	Call vs Connection
Identity attributes	Calling UNI Transport Resource Identifier	End-to-end or carry transparently	Call
	Called UNI Transport Resource Identifier	End-to-end or carry transparently	Call
	Initiating CC name	Local	Connection
	Initiating CallC name	Local	Call
	Terminating CC name	Local	Connection
	Terminating CallC name	Local	Call
	Connection name	Local	Connection
	Call name	End-to-end	Call
Service attributes	SNP ID	Local	Connection
	SNPP ID	Local	Connection
	Called AGC SNP ID	Carry transparently	Connection
	Called AGC SNPP ID	Carry transparently	Connection
	Directionality	Local	Call/connection
Policy attributes	CoS	End-to-end	Call
	GoS	End-to-end	Call
	Security	Local	Call/connection
	Explicit resource list	Local	Connection

7.1 UNI attributes list

7.1.1 Identity attributes

7.1.1.1 Calling UNI Transport Resource Identifier

This attribute is the ITU-T Rec. G.8080/Y.1304 UNI Transport Resource Identifier used to reach the A-end call controller. The value of this attribute has to be globally unique and assigned by the service provider. For example, a user name may be an NSAP address assigned by service provider #1, while another user name may be an IPv6 address assigned by service provider #2. As the user name provides a globally unique identification of the users, different formats may co-exist.

7.1.1.2 Called UNI Transport Resource Identifier

This attribute is the ITU-T Rec. G.8080/Y.1304 UNI Transport Resource Identifier used to reach the Z-end call controller. Characteristics are the same as those specified for the Calling UNI Transport Resource Identifier.

7.1.1.3 Initiating CC/CallC name

This attribute specifies the name associated with the CC and CallC that initiated the explicit signalling message.

7.1.1.4 Terminating CC/CallC name

This attribute specifies the name associated with the CC and CallC that terminates the explicit signalling message.

7.1.1.5 Connection name

This attribute uniquely identifies a link connection that has been chosen to be used for the connection. The value of this attribute is locally unique, and may be assigned by either the user or the network.

7.1.1.6 Call name

This attribute uniquely identifies the requested call. The value of this attribute is globally unique and assigned by the network.

7.1.2 Service attributes

7.1.2.1 SNP ID

This attribute represents the subnetwork point used to establish a link connection in a connection request. This is also the SNP used to create the SNC. For specific connection requests, the SNP ID is chosen from a set of SNPs specified within an SNPP. The value of this attribute is locally unique, is automatically discovered or provisioned (as per ITU-T Rec. G.7714/Y.1705), and is independent between layer networks. For calls that request bidirectional connections, or request multiple connections, this attribute may contain multiple values ordered as downstream IDs first followed by upstream IDs. The SNP ID(s) may be specified by either the initiating or terminating LRM. To prevent contention, the LRM with the higher name value overrides the choice for SNP IDs to use.

The SNP ID in signalling messages include:

– Calling AGC SNP ID – This ID is used to setup the LC on the calling AGC to network element access links.

– Called AGC SNP ID – This ID is used to setup the LC on the called AGC to network element access links.

– SNP ID – This ID is used to setup an LC on network element to network element access links.

7.1.2.2 SNPP ID

This attribute represents the subnetwork point pool used to request establishing a connection. SNPP IDs uniquely identify the set of SNPs that may be used for requesting a connection between subnetworks, and are independent between layer networks. Multiple SNPPs may exist between subnetworks. For calls that request bidirectional connections or request multiple connections, this attribute may contain multiple values ordered as downstream IDs first, followed by upstream IDs. The SNPP ID(s) may be specified by the initiating LRM, with the terminating LRM choosing an SNP within the SNPP to establish the link connection. To prevent contention, the LRM with the higher name value overrides the choice for SNPP ID to use, subject to meeting the constraints requested by the call.

The SNPP ID in signalling messages include:

– Calling AGC SNPP ID – This ID is used to setup the LC on the calling AGC to network element access links.

– Called AGC SNPP ID – This ID is used to setup the LC on the called AGC to network element access links.

– SNPP ID – This ID is used to setup an LC on the network element to network element access links.

7.1.2.3 Directionality

This attribute specifies the directionality of the connection. The directionality attribute supports requests for unidirectional, symmetric bidirectional and asymmetric bidirectional connections. In the case of asymmetric bidirectional connections, additional information is provided in the attribute to specify the number of connections requested for both downstream and upstream.

7.1.3 Policy attributes

7.1.3.1 CoS and connection CoS

This attribute specifies the CoS for the requested call. The value of this attribute is unique per domain, assigned by the network, and may differ per user-network relationship (e.g., the CoS that is used by user #1 may be different from the CoS that is used by user #2). Across domains, a translation function that translates one network's CoS to another network's CoS value may be needed to support end-to-end CoS requests. This call CoS attribute is part of the SLA (Service Level Agreement) of the calling party to specify the CoS. A translation function that translates the call CoS to a domain-specific CoS (connection CoS) value is required.

The connection CoS might be different from domain to domain, but the connection CoS in each domain has to meet the SLA to support end-to-end CoS requests. It is applicable only to the I-NNI.

Example list of information that may be contained within CoS includes: enumerated list of service classes.

7.1.3.2 GoS and connection GoS

This attribute specifies the GoS for the requested call and is used to further specify the GoS associated with each GoS requested. The value of this attribute is unique per domain, assigned by the network, and may differ per user-network relationship. Across domains, a translation function that translates one network's GoS to another network's GoS value may be needed to support end-to-end GoS requests. This call GoS attribute is part of the SLA (Service Level Agreement) of the calling party to specify the GoS. At the domain boundary a translation function that translates call GoS to a domain-specific GoS (connection GoS) value is required.

The connection GoS will be different from domain to domain, but the connection GoS in each domain has to meet the SLA to support end-to-end GoS requests. It is applicable only to the I-NNI.

Example list of information that may be contained within GoS includes:

– diversity information; or
– a list of subnetwork controller name(s), SNPP(s), SNP(s), or 3 SNPP(s) and SNP(s) to avoid (or include).

7.1.3.3 Security

This attribute specifies information necessary to allow verification of the call request (e.g., this may include information to allow authentication of the call request, and possibly integrity checking of the call request). The value of this attribute is locally unique.

7.1.4 Status codes

7.1.4.1 Response codes for calls

Call setup – success;

Call setup – failed: called party busy;

Call setup – failed: calling party busy;

Call setup – failed: network busy;

Call setup – failed: message error;

Call setup – failed: identity error: invalid A-end user name;

Call setup – failed: identity error: invalid Z-end user name;

Call setup – failed: identity error: invalid connection name;

Call setup – failed: service error: invalid SNP ID;

Call setup – failed: service error: unavailable SNP ID;

Call setup – failed: service error: invalid SNPP ID;

Call setup – failed: service error: unavailable SNPP ID;

Call setup – failed: policy error: invalid CoS;

Call setup – failed: policy error: unavailable CoS;

Call setup – failed: policy error: invalid GoS;

Call setup – failed: policy error: unavailable GoS;

Call setup – failed: policy error: failed security check;

Call release – success;

Call release – failed: message error;

Call release – failed: identity error: invalid call name;

Call release – failed: policy error: failed security check.

7.1.4.2 Notification codes

Call error – non-service affecting;

Call error – service affecting.

7.2 I-NNI attributes list

7.2.1 Identity attributes

7.2.1.1 Calling UNI transport resource identifier

See definition in 7.1.1.1.

7.2.1.2 Called UNI transport resource identifier

See definition in 7.1.1.2.

7.2.1.3 Initiating CC/CallC name

See definition in 7.1.1.3.

7.2.1.4 Terminating CC/CallC name

See definition in 7.1.1.4.

7.2.1.5 Connection name

This attribute uniquely identifies a subnetwork connection that has been chosen to be used for the connection in the domain that contains this I-NNI. The value of this attribute is unique within the domain.

7.2.1.6 Call name

See definition in 7.1.1.6.

7.2.2 Service attributes

7.2.2.1 SNP ID

See definition in 7.1.2.1.

7.2.2.2 SNPP ID

See definition in 7.1.2.2.

7.2.2.3 Directionality

See definition in 7.1.2.3.

7.2.3 Policy attributes

7.2.3.1 CoS and connection CoS

See definition in 7.1.3.1.

7.2.3.2 GoS and connection GoS

See definition in 7.1.3.2.

7.2.3.3 Explicit resource list

This attribute specifies the explicit resources used in setting up the connection. The ordered list may include zero or more instances of CC names, SNPP ID and/or SNP ID that are within the domain of responsibility of the RC, i.e., routing area.

7.2.3.4 Recovery

This attribute specifies the type of recovery method used for the connection. For example, the information contained in this attribute may include:

– Indication of connection type (e.g., working or protection connection).

– Type of recovery (e.g., 1+1, 1:1, auto-reroute).

– Behaviour of recovery (e.g., revertive, non-revertive, or adaptive non-revertive).

7.2.4 Status codes

7.2.4.1 Response codes for connections

Connection setup – success;

Connection setup – failed: message error;

Connection setup – failed: called party busy;

Connection setup – failed: calling party busy;

Connection setup – failed: timeout;

Connection setup – failed: identity error: invalid connection name;

Connection setup – failed: service error: invalid SNP ID;

Connection setup – failed: service error: unavailable SNP ID;

Connection setup – failed: service error: invalid SNPP ID;

Connection setup – failed: service error: unavailable SNPP ID;

Connection setup – failed: policy error: invalid explicit resource list;

Connection setup – failed: policy error: invalid recovery;

Connection setup – failed: connection error: failed to create SNC;

Connection setup – failed: connection error: failed to establish LC;

Connection release – success;

Connection release – failed: message error;

Connection release – failed: timeout;

Connection release – failed: identity error: invalid call name;

Connection release – failed: connection error: failed to release SNC;

Connection release – failed: connection error: failed to free LC.

7.2.4.2 Notification codes

Connection error – non-service affecting;

Connection error – service affecting;

Connection error – unexpected call release.

7.3 E-NNI attributes list

7.3.1 Identity attributes

7.3.1.1 Calling UNI transport resource identifier

See definition in 7.1.1.1.

7.3.1.2 Called UNI transport resource identifier

See definition in 7.1.1.2.

7.3.1.3 Initiating CC/CallC name

See definition in 7.1.1.3.

7.3.1.4 Terminating CC/CallC name

See definition in 7.1.1.4.

7.3.1.5 Connection name

See definition in 7.1.1.5.

7.3.1.6 Call name

See definition in 7.1.1.6.

7.3.2 Service attributes

7.3.2.1 SNP ID

See definition in 7.1.2.1.

7.3.2.2 SNPP ID

See definition in 7.1.2.2.

7.3.2.3 Directionality

See definition in 7.1.2.3.

7.3.3 Policy attributes

7.3.3.1 CoS

See definition in 7.1.3.1.

7.3.3.2 GoS

See definition in 7.1.3.2.

7.3.3.3 Security

See definition in 7.1.3.3.

7.3.3.4 Explicit resource list

See definition in 7.2.3.3.

7.3.4 Status codes

7.3.4.1 Response codes for connections

Connection setup – success;

Connection setup – failed: message error;

Connection setup – failed: called party busy;

Connection setup – failed: calling party busy;

Connection setup – failed: timeout;

Connection setup – failed: identity error: invalid A-end user name;

Connection setup – failed: identity error: invalid Z-end user name;

Connection setup – failed: identity error: invalid connection name;

Connection setup – failed: service error: invalid SNP ID;

Connection setup – failed: service error: unavailable SNP ID;

Connection setup – failed: service error: invalid SNPP ID;

Connection setup – failed: service error: unavailable SNPP ID;

Connection setup – failed: policy error: invalid CoS;

Connection setup – failed: policy error: unavailable CoS;

Connection setup – failed: policy error: invalid GoS;

Connection setup – failed: policy error: unavailable GoS;

Connection setup – failed: policy error: failed security check;

Connection setup – failed: policy error: invalid explicit resource list;

Connection setup – failed: policy error: invalid recovery;

Connection setup – failed: connection error: failed to create SNC;

Connection setup – failed: connection error: failed to establish LC;

Connection release – success;

Connection release – failed: message error;

Connection release – failed: timeout;

Connection release – failed: identity error: invalid call name;

Connection release – failed: policy error: failed security check;

Connection release – failed: connection error: failed to release SNC;

Connection release – failed: connection error: failed to free LC.

7.3.4.2 Notification codes

Connection error – non-service affecting;

Connection error – service affecting;

Connection error – unexpected call release.

8 DCM message sets

The DCM messages may be separated into messages associated with the UNI connection operations and messages associated with the NNI (I-NNI and E-NNI) connection operations. Tables 8-1 to 8-3 summarize a candidate list of messages that are considered for signalling processing. They represent the logical messages that are exchanged across the respective interfaces to support the CallC, CC, and LRM. Note that protocol design decisions may result in aggregation (or segmentation) of some of these logical messages; however, the functions supported by the message exchange shall be present.

NOTE – The **modify** operation is for further study and is not included as part of the message sets.

Table 8-1/G.7713/Y.1704 – UNI messages

	UNI messages
Call setup messages	CallSetupRequest CallSetupIndication CallSetupConfirm
Call release messages	CallReleaseRequest CallReleaseIndication
Call query messages	CallQueryRequest CallQueryIndication
Call notification message	CallNotify

Table 8-2/G.7713/Y.1704 – I-NNI messages

	I-NNI messages
Connection setup messages	ConnectionSetupRequest ConnectionSetupIndication ConnectionSetupConfirm
Connection release messages	ConnectionReleaseRequest ConnectionReleaseIndication
Connection query messages	ConnectionQueryRequest ConnectionQueryIndication
Connection notification message	ConnectionNotify

Table 8-3/G.7713/Y.1704 – E-NNI messages

	E-NNI messages
Connection setup messages	ConnectionSetupRequest ConnectionSetupIndication ConnectionSetupConfirm
Connection release messages	ConnectionReleaseRequest ConnectionReleaseIndication
Connection query messages	ConnectionQueryRequest ConnectionQueryIndication
Connection notification message	ConnectionNotify

8.1 UNI messages

8.1.1 Call setup

A two-phase (and optional third phase) approach is defined for setting up a call. A call name (**callName**) is created by the network and sent to the user for reference to the requested call. In addition, a connection name (**connName**) is created by the initiating call requester for reference to the requested connection.

8.1.1.1 Request: Setup call

A **callSetupRequest** message is defined to set up the call. Attributes sent for call setup request are shown in Table 8-4.

Table 8-4/G.7713/Y.1704 – UNI call setup request message

User sent attributes	Network sent attributes
Calling UNI Transport Resource Identifier	Calling UNI Transport Resource Identifier
Called UNI Transport Resource Identifier	Called UNI Transport Resource Identifier
Initiating CallC name	Initiating CC/CCC name
Terminating CallC name	Terminating CC/CCC name
Calling AGC SNP ID	Calling AGC SNP ID
Calling AGC SNPP ID	Calling AGC SNPP ID
Called AGC SNP ID	Called AGC SNP ID
Called AGC SNPP ID	Called AGC SNPP ID
Directionality	Directionality
CoS	CoS
GoS	GoS
Security	Security
connName	connName
	callName

8.1.1.2 Indication: Setup call

A **callSetupIndication** message is defined as a response to **callSetupRequest**. This message may be sent by either the Z-end user (or third party) or the network. Attributes sent for call setup indication are shown in Table 8-5.

Table 8-5/G.7713/Y.1704 – UNI call setup indication message

User sent attributes	Network sent attributes
connName	connName
callName	callName
Status	Status

8.1.1.3 Confirm: Setup call

An optional **callSetupConfirm** message is defined as a response to **callSetupIndication**. This message may be sent by either the A-end user (or third party) or the network. Attributes sent for call setup confirm are shown in Table 8-6.

Table 8-6/G.7713/Y.1704 – UNI call setup confirm message

User sent attributes	Network sent attributes
connName	connName
callName	callName
Status	Status

8.1.2 Call release

8.1.2.1 Request: Release call

A **callReleaseRequest** message is defined to release the call. This message may be sent by either the user or the network. Attributes sent for call release request are shown in Table 8-7.

Table 8-7/G.7713/Y.1704 – UNI link call release request message

User sent attributes	Network sent attributes
callName	callName
Security	Security

8.1.2.2 Indication: Release call

A **callReleaseIndication** message is defined as a response to **callReleaseRequest**. This message may be sent by either the user or the network. Attributes sent for call release indication are shown in Table 8-8.

Table 8-8/G.7713/Y.1704 – UNI call release indication message

User sent attributes	Network sent attributes
callName	callName
Status	Status

8.1.3 Call query

The following status codes are valid for a call query:

– Call active.

– Call does not exist.

– Call unavailable.

– Call pending.

8.1.3.1 Request: Query per call characteristics

A **callQueryRequest** message is defined to query an existing call. This message may be sent by either the user or the network. Attributes sent for call query request are shown in Table 8-9.

Table 8-9/G.7713/Y.1704 – UNI call query request message

User sent attributes	Network sent attributes
callName	callName
Security	Security

8.1.3.2 Response: Query per call characteristics

A **callQueryResponse** message is defined as a response to **callQueryRequest**. This message may be sent by either the user or the network. Attributes sent for call query response are shown in Table 8-10.

Table 8-10/G.7713/Y.1704 – UNI call query response message

User sent attributes	Network sent attributes
callName	callName
CoS	CoS
GoS	GoS
For each connName: SNP ID SNPP ID	For each connName: SNP ID SNPP ID
Status	Status

8.1.3.3 Request: Query all calls

A **callQueryAllRequest** message is defined to query all existing calls associated with a particular signalling agent or Network Call Controller name. This message may be sent by either the user or the network. Attributes sent for all call query request are shown in Table 8-11.

Table 8-11/G.7713/Y.1704 – UNI all call query request message

User sent attributes	Network sent attributes
CC/NCC name	CC/NCC name
Security	Security

8.1.3.4 Response: Query all calls

A **callQueryAllResponse** message is defined as a response to **callQueryAllRequest**. This message may be sent by either the user or the network. Attributes sent for all call query response are shown in Table 8-12.

Table 8-12/G.7713/Y.1704 – UNI all call query response message

User sent attributes	Network sent attributes
CC/NCC name	CC/NCC name
List of callNames	List of callNames
Status	Status

8.1.4 Notification

A **notification** message is specified to allow exchange of information related to the connection status. The notification message may be sent to notify call or connection status. Attributes sent for notification are shown in Table 8-13.

Table 8-13/G.7713/Y.1704 – UNI notification message

User sent attributes	Network sent attributes
callName	callName
connName	connName
Error code	Error code
Security	Security

The notification message is one-phase.

8.2 I-NNI messages

8.2.1 Connection setup

A two-phase (and optional third phase) approach is defined for setting up connections. A connection name (**connName**) is created for reference to the requested connection.

8.2.1.1 Request: Setup connection

A **connSetupRequest** message is defined to set up a connection. Attributes sent for connection setup request are shown in Table 8-14.

Table 8-14/G.7713/Y.1704 – I-NNI connection setup request message

Attributes
Calling UNI Transport Resource Identifier
Called UNI Transport Resource Identifier
Initiating CC name
Terminating CC name
connName
callName
Local SNP ID

Table 8-14/G.7713/Y.1704 – I-NNI connection setup request message

Attributes
Local SNPP ID
Called AGC SNP ID
Called AGC SNPP ID
Directionality
CoS
GoS
Connection CoS
Connection GoS
Explicit resource list
Recovery

8.2.1.2 Indication: Setup connection

A **connSetupIndication** message is defined as a response to **connSetupRequest**. Attributes sent for connection setup indication are shown in Table 8-15.

Table 8-15/G.7713/Y.1704 – I-NNI connection setup indication message

Attributes
connName
callName
Status

8.2.1.3 Confirm: Setup connection

An optional **connSetupConfirm** message is defined as a response to **connSetupIndication**. Attributes sent for connection setup confirm are shown in Table 8-16.

Table 8-16/G.7713/Y.1704 – I-NNI connection setup confirm message

Attributes
connName
callName
Status

8.2.2 Connection release

8.2.2.1 Request: Release connection

A **connReleaseRequest** message is defined to release a connection. Attributes sent for connection release request are shown in Table 8-17.

Table 8-17/G.7713/Y.1704 – I-NNI connection release request message

Attributes
callName
connName

8.2.2.2 Indication: Release connection

A **connReleaseIndication** message is defined as a response to **connReleaseRequest**. Attributes sent for connection release indication are shown in Table 8-18.

Table 8-18/G.7713/Y.1704 – I-NNI connection release indication message

Attributes
callName
connName
Status

8.2.3 Connection query

The following status codes are valid for a connection query:

– Connection active.

– Connection does not exist.

– Connection unavailable.

– Connection pending.

8.2.3.1 Request: Query per connection characteristics

A **connQueryRequest** message is defined to query an existing connection. Attributes sent for connection query request are shown in Table 8-19.

Table 8-19/G.7713/Y.1704 – I-NNI connection query request message

Attributes
callName
connName

8.2.3.2 Response: Query per connection characteristics

A **connQueryResponse** message is defined as a response to **connQueryRequest**. Attributes sent for connection query response are shown in Table 8-20.

Table 8-20/G.7713/Y.1704 – I-NNI connection query response message

Attributes
callName
connName SNP ID SNPP ID
Directionality
Explicit resource list
Recovery
Status

8.2.3.3 Request: Query all connections

A **connQueryAllRequest** message is defined to query all existing connections associated with a connection controller name. Attributes sent for all connection query request are shown in Table 8-21.

Table 8-21/G.7713/Y.1704 – I-NNI all connection query request message

Attributes
CC name

8.2.3.4 Response: Query all connections

A **connQueryAllResponse** message is defined as a response to **connQueryAllRequest**. This message may be sent by either the user or the network. Attributes sent for all connection query response are shown in Table 8-22.

Table 8-22/G.7713/Y.1704 – I-NNI all connection query response message

Attributes
CC name
List of connName with their associated callName
Status

8.2.4 Notification

A **notification** message is specified to allow exchange of information related to the connection status. The notification message may be sent to notify call or connection status. Attributes sent for notification are shown in Table 8-23.

Table 8-23/G.7713/Y.1704 – I-NNI notification message

Attributes
callName
connName
Error code

The notification message is one-phase.

8.3 E-NNI messages

8.3.1 Connection setup

A two-phase (and optional third phase) approach is defined for setting up connections. A connection name (**connName**) is created for reference to the requested connection.

8.3.1.1 Request: Setup connection

A **connSetupRequest** message is defined to set up a connection. Attributes sent for connection setup request are shown in Table 8-24.

Table 8-24/G.7713/Y.1704 – E-NNI connection setup request message

Attributes
Calling UNI Transport Resource Identifier
Called UNI Transport Resource Identifier
Initiating CC/NCC name
Terminating CC/NCC name
connName
callName
Local SNP ID
Local SNPP ID
Called AGC SNP ID
Called AGC SNPP ID
Directionality
CoS
GoS
Explicit resource list
Recovery

8.3.1.2 Indication: Setup connection

A **connSetupIndication** message is defined as a response to **connSetupRequest**. Attributes sent for connection setup indication are shown in Table 8-25.

Table 8-25/G.7713/Y.1704 – E-NNI connection setup indication message

Attributes
connName
callName
Status

8.3.1.3 Confirm: Setup connection

An optional **connSetupConfirm** message is defined as a response to **connSetupIndication**. Attributes sent for connection setup confirm are shown in Table 8-26.

Table 8-26/G.7713/Y.1704 – E-NNI connection setup confirm message

Attributes
connName
callName
Status

8.3.2 Connection release

8.3.2.1 Request: Release connection

A **connReleaseRequest** message is defined to release a connection. Attributes sent for connection release request are shown in Table 8-27.

Table 8-27/G.7713/Y.1704 – E-NNI connection release request message

Attributes
callName
connName
Security

8.3.2.2 Indication: Release connection

A **connReleaseIndication** message is defined as a response to **connReleaseRequest**. Attributes sent for connection release indication are shown in Table 8-28.

Table 8-28/G.7713/Y.1704 – E-NNI connection release indication message

Attributes
callName
connName
Status

8.3.3 Connection query

The status codes are identified in 8.2.3.

8.3.3.1 Request: Query per connection characteristics

A **connQueryRequest** message is defined to query an existing connection. Attributes sent for connection query request are shown in Table 8-29.

Table 8-29/G.7713/Y.1704 – E-NNI connection query request message

Attributes
callName
connName
Security

8.3.3.2 Response: Query per connection characteristics

A **connQueryResponse** message is defined as a response to **connQueryRequest**. Attributes sent for connection query response are shown in Table 8-30.

Table 8-30/G.7713/Y.1704 – E-NNI connection query response message

Attributes
callName
connName SNP ID SNPP ID Directionality
Explicit resource list
Recovery
Status

8.3.3.3 Request: Query all connections

A **connQueryAllRequest** message is defined to query all existing connections associated with a Network Call Controller name. Attributes sent for all connection query request are shown in Table 8-31.

Table 8-31/G.7713/Y.1704 – E-NNI all connection query request message

Attributes
NCC name
Security

8.3.3.4 Response: Query all connections

A **connQueryAllResponse** message is defined as a response to **connQueryAllRequest**. This message may be sent by either the user or the network. Attributes sent for all connection query response are shown in Table 8-32.

Table 8-32/G.7713/Y.1704 – E-NNI all connection query response message

Attributes
NCC name
For each callName: List of associated connName
Status

8.3.4 Notification

A **notification** message is specified to allow exchange of information related to the connection status. The notification message may be sent to notify call or connection status. Attributes sent for notification are shown in Table 8-33.

Table 8-33/G.7713/Y.1704 – E-NNI notification message

Attributes
callName
connName
Error code
Security

The notification message is one-phase.

9 DCM state diagrams

This clause provides the detail specifications for the state transitions of the connection controller (CC) entity based on state transition events. The state diagrams specified in this clause apply on a per-CC basis. It describes the progression of the state of a connection, along with the associated events and actions that take place at each communication entity during a transition from one state to the next state.

The state transitions are specified for:

– Initiating user: the user that initiated the operation. For setup operations this is the A-end user (CCC-a), while for release operations this may be either the A-end or the Z-end user.

– Terminating user: the user that terminates the call. For setup operations this is the Z-end user (CCC-z), while for release operations this may be either the Z-end or the A-end user.

– Call controller.

– Connection controller.

The following states are defined for call operations:

– **Idle (S_i)**: This is the default state. In this state, the signalling communication entity is available to accept call setup request and take action.

– **Verify call setup request (S_{svreq})**: In this state, the CallC is in the process of verifying the setup request (e.g., this may include security and policy checking).

- **Call setup request initiated (S_{sreq})**: In this state, the CallC has generated and transmitted a setup request, and is waiting for a response to the setup request.
- **Setup connection state (S_{sconn})**: In this state, the CC performs the connection setup to support the accepted call.
- **Call setup accepted (S_{sacpt})**: In this state, the CallC has generated and transmitted a setup indication as a response to a setup request and is waiting for a final confirmation of call established.
- **Verify call (S_{svcall})**: In this state, the CallC is in the process of verifying that the call has been set up correctly.
- **Active (S_a)**: In this state, the call setup process is done and the respective connections to support the call have been set up and are ready for transferring user characteristic information.
- **Verify call release request (S_{rvreq})**: In this state, the CallC is in the process of verifying the release request. Positive verification allows the release request to proceed and negative verification refuses the release request. This verification may include authentication and integrity checking, and may also include policy checking.
- **Call release request initiated (S_{rreq})**: In this state, the CallC has generated and transmitted a release request, and is waiting for a response to the release request.
- **Release connection state (S_{rconn})**: In this state, the CC performs the connection release to support the call release operation.
- **Signalling error (S_{sigerr})**: In this state, the signalling communication channel has been interrupted.

The following states are defined for connection operations, and represent the detailed states that occur for the "**connection state**" described above in the call state:

- **Idle (S_i)**: This is the default state. In this state, the signalling communication entity is available to accept connection setup request and take action.
- **Verify connection setup request (S_{svreq})**: In this state, the CC is in the process of verifying the setup request (e.g., this may include security and policy checking).
- **Connection setup request initiated (S_{sreq})**: In this state, the CC has generated and transmitted a setup request, and is waiting for a response to the setup request.
- **Connection setup accepted (S_{sacpt})**: In this state, the CC has generated and transmitted a setup indication as a response to a setup request, and is waiting for a final confirmation of connection established.
- **Verify connection (S_{svconn})**: In this state, the CC is in the process of verifying that the connection has been set up correctly.
- **Active (S_a)**: In this state, the connection setup process is done and notification is sent to the CallC to indicate completion of connection setup.
- **Verify connection release request (S_{rvreq})**: In this state, the CC is in the process of verifying the release request. Positive verification allows the release request to proceed and negative verification refuses the release request. This verification may include authentication and integrity checking, and may also include policy checking.
- **Connection release request initiated (S_{rreq})**: In this state, the CC has generated and transmitted a release request, and is waiting for a response to the release request.
- **Signalling error (S_{sigerr})**: In this state, the signalling communication channel has been interrupted.

Events that may cause a state transition include:

– External triggered events, for example, a user decides to request a call, traffic engineering management issues a new call, a failed connection leads to request to release a call, etc. These may be triggered via a user interface or application interface.
– Reception of message events.
– A result of verification events.
– Timeout events.

Unexpected or unknown messages do not cause state transitions. They may be ignored or an error message may be sent back to inform of the unexpected/unknown message.

9.1 Call state

The following events are associated with the user CallC call state (Table 9-1):

Table 9-1/G.7713/Y.1704 – User CallC call state events

Event	Event description
Unk	An unknown or unexpected message was received.
SetReq	The user CallC received a request to set up a call.
SetVer	The user CallC successfully verified the call set up request.
SetNVer	The user CallC unsuccessfully verified the call set up request.
SetSuc	The user CallC received a response that the call was successfully established.
SetNSuc	The user CallC received a response that the call was not established.
SetExp	Call setup timer expired.
RelReq	The user CallC received a request to release a call.
RelVer	The user CallC successfully verified the call release request.
RelNVer	The user CallC unsuccessfully verified the call release request.
RelSuc	The user CallC received a response that the call was released.
RelExp	Call release timer expired.
SigErr	Defect detected at the signalling communication channel.
SigNErr	Signalling communication channel defect repaired.

The following events are associated with the network CallC (calling party and called party) call state (Table 9-2):

Table 9-2/G.7713/Y.1704 – Network CallC call state events

Event	Event description
Unk	An unknown or unexpected message was received.
SetReq	The network CallC received a request to set up a call.
SetVer	The network CallC successfully verified the call set up request.
SetNVer	The network CallC unsuccessfully verified the call set up request.
SetAcp	The network CallC received a response that the call setup request was accepted.
SetNAcp	The network CallC received a response that the call setup request was denied.

Table 9-2/G.7713/Y.1704 – Network CallC call state events

Event	Event description
SetCon	The network CallC received a response that the connection supporting the call was successfully established.
SetNCon	The network CallC received a response that the connection supporting the call was not established.
SetCallVer	The network CallC successfully verified the established call.
SetCallNVer	The network CallC unsuccessfully verified the established call.
SetExp	Call setup timer expired.
RelReq	The network CallC received a request to release call.
RelVer	The network CallC successfully verified the call release request.
RelNVer	The network CallC unsuccessfully verified the call release request.
RelCon	The network CallC received a response that the connection supporting the call was successfully released.
RelNCon	The network CallC received a response that the connection supporting the call was not released.
RelExp	Call release timer expired.
SigErr	Defect detected at the signalling communication channel.
SigNErr	Signalling communication channel defect repaired.

9.1.1 Initiating user CallC call state

9.1.1.1 Initiating user call state: Setup

The state transitions shown in this clause apply to call setup for the initiating user CallC (Table 9-3, Figure 9-1).

Table 9-3/G.7713/Y.1704 – Initiating user CallC call setup state transitions

Current state	Event	Actions taken	Next state
*	Unk	• Send a notification message to the sender to inform of the error. No state transition occurs.	*
S_i	SetReq	• Verify the request.	S_{svreq}
S_{svreq}	SetVer	• Send call setup request message. • Initiate call setup timer (T_{call_setup}).	S_{sreq}
S_{svreq}	SetNVer	• Notify call initiator of denied call setup request.	S_i
S_{sreq}	SetSuc	• Delete call setup timer (T_{call_setup}). • Send call setup request confirmation (optional).	S_a
S_{sreq}	SetNSuc	• Delete call setup timer (T_{call_setup}).	S_i
S_{sreq}	SetExp	• Notify call initiator of call setup request timeout. • Send call release request message. • Initiate call release timer ($T_{call\_release}$).	S_{rreq}
S_a	SigErr	• None.	S_{sigerr}
S_{sigerr}	SigNErr	• None.	S_a

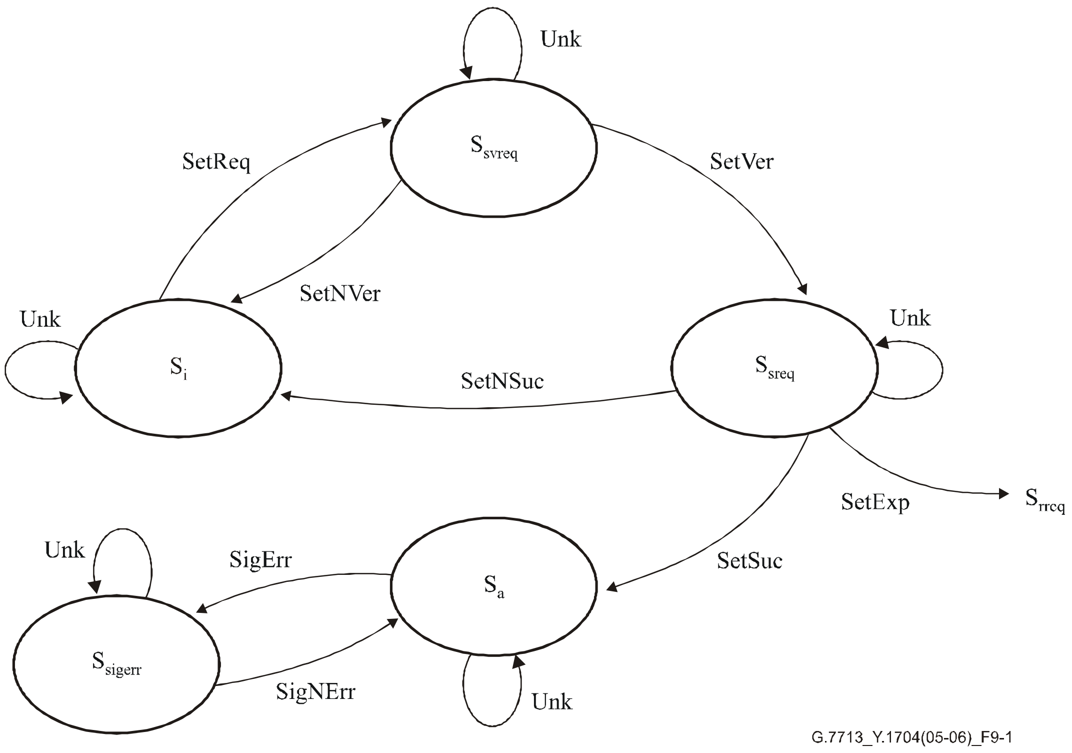


Figure 9-1/G.7713/Y.1704 – Initiating user CallC call setup state diagram

9.1.1.2 Initiating user call state: Release

The state transitions shown in this clause apply to call release for the initiating user CallC (Table 9-4, Figure 9-2).

Table 9-4/G.7713/Y.1704 – Initiating user CallC call release state transitions

Current state	Event	Actions taken	Next state
*	Unk	• Send a notification message to the sender to inform of the error. No state transition occurs.	*
S_a	RelReq	• Verify the request.	S_{rvreq}
S_{rvreq}	RelVer	• Send call release request message. • Initiate call release timer ($T_{call\_release}$).	S_{rreq}
S_{rvreq}	RelNVer	• Notify call initiator of denied call release request.	S_a
S_{rreq}	RelSuc	• Delete call release timer ($T_{call\_release}$).	S_i
S_{rreq}	RelExp	• Notify call initiator of call release request timeout.	S_i

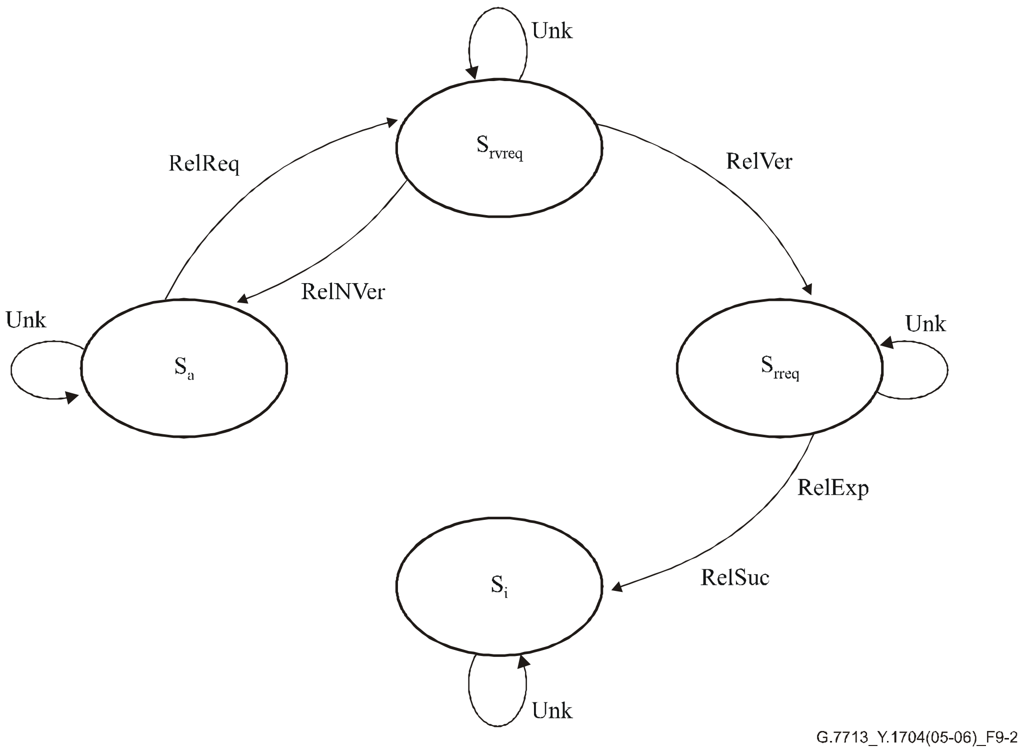


Figure 9-2/G.7713/Y.1704 – Initiating user CallC call release state diagram

9.1.2 Terminating user CallC call state

9.1.2.1 Terminating user call state: Setup

The state transitions shown in this clause apply to call setup for the terminating user CallC (Table 9-5, Figure 9-3).

Table 9-5/G.7713/Y.1704 – Terminating user CallC call setup state transitions

Current state	Event	Actions taken	Next state
*	Unk	• Send a notification message to the sender to inform of the error. No state transition occurs.	*
S_i	SetReq	• Verify the request.	S_{svreq}
S_{svreq}	SetVer	• Send call accepted message. • Initiate call setup timer (T_{call_setup}).	S_{sacpt}
S_{svreq}	SetNVer	• Notify call initiator of denied call setup request.	S_i
S_{sacpt}	SetSuc	• Delete call setup timer (T_{call_setup}).	S_a
S_{sacpt}	SetNSuc	• Delete call setup timer (T_{call_setup}).	S_i
S_{sacpt}	SetExp	• Notify call initiator of call setup request timeout. • Send call release request message. • Initiate call release timer ($T_{call\_release}$).	S_{rreq}
S_a	SigErr	• None.	S_{sigerr}
S_{sigerr}	SigNErr	• None.	S_a

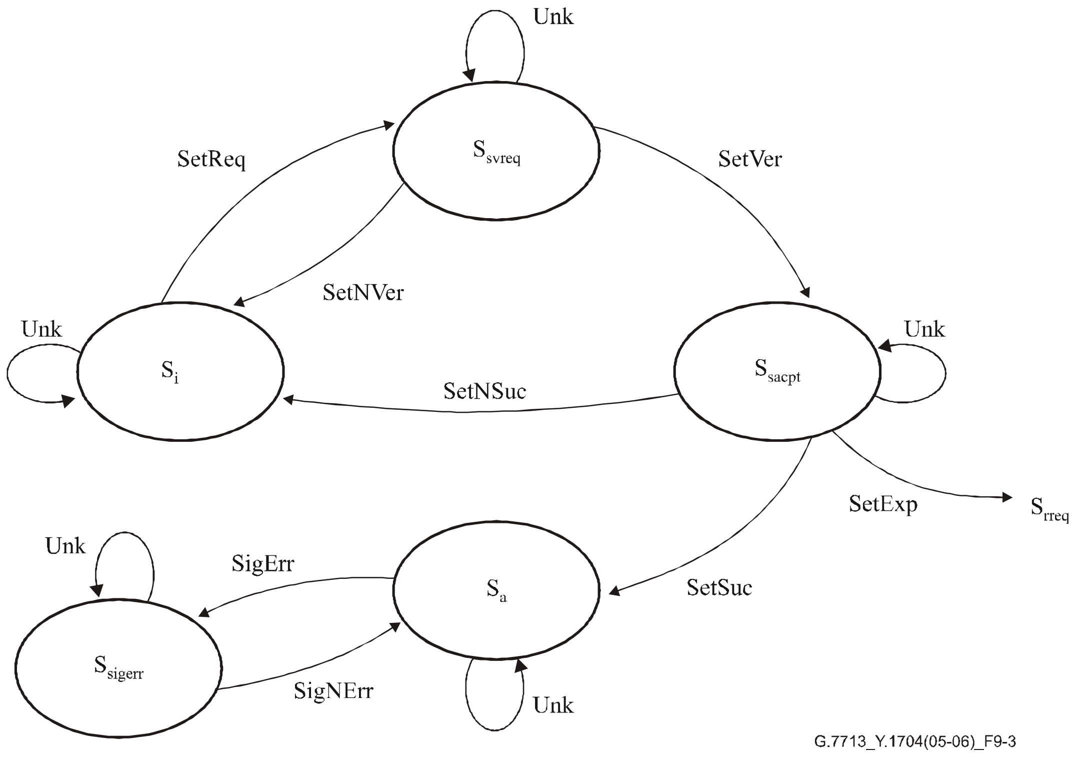


Figure 9-3/G.7713/Y.1704 – Terminating user CallC call setup state diagram

9.1.2.2 Terminating User call state: Release

The state transitions shown in this clause apply to call release for the terminating user CallC (Table 9-6, Figure 9-4).

Table 9-6/G.7713/Y.1704 – Terminating user CallC call release state transitions

Current state	Event	Actions taken	Next state
*	Unk	• Send a notification message to the sender to inform of the error. No state transition occurs.	*
S_a	RelReq	• None.	S_{rreq}
S_{rreq}	RelSuc	• None.	S_i

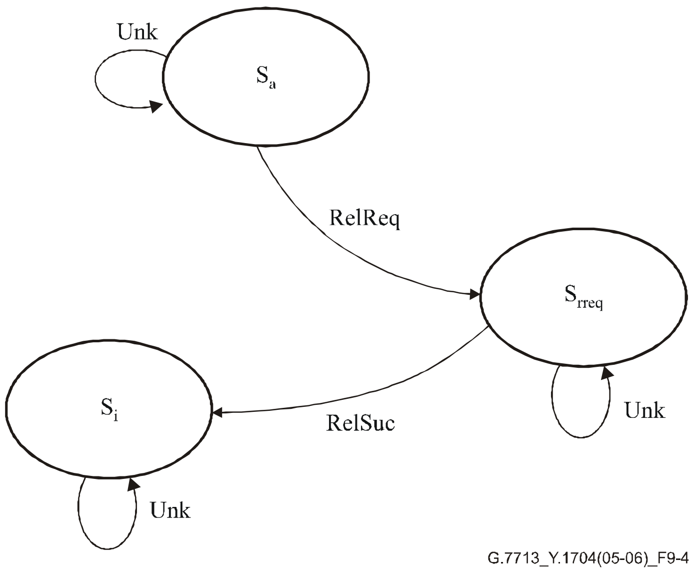


Figure 9-4/G.7713/Y.1704 – Terminating user CallC call release state diagram

9.1.3 Network CallC call state

The following transitions apply to both the calling party and called party CallC.

9.1.3.1 Network CallC call state: Setup

The state transitions shown in this clause apply to call setup for the network CallC (Table 9-7, Figure 9-5).

Table 9-7/G.7713/Y.1704 – Network CallC call setup state transitions

Current state	Event	Actions taken	Next state
*	Unk	• Send a notification message to the sender to inform of the error. No state transition occurs.	*
S_i	SetReq	• Verify the request.	S_{svreq}
S_{svreq}	SetVer	• Send call setup request message. • Initiate call setup timer (T_{call_setup}).	S_{sreq}
S_{svreq}	SetNVer	• Notify call initiator of denied call setup request.	S_i
S_{sreq}	SetAcp	• Delete call setup timer (T_{call_setup}). • Send call setup accepted message. • Initiate connection setup process (for calling party CallC). • Initiate connection setup timer (T_{conn_setup}).	S_{sconn}
S_{sreq}	SetNAcp	• Delete call setup timer (T_{call_setup}). • Notify call initiator of denied call setup request.	S_i
S_{sreq}	SetExp	• Notify call initiator of denied call setup request.	S_i
S_{sconn}	SetCon	• Delete connection setup timer (T_{conn_setup}).	S_{svcall}
S_{sconn}	SetNCon	• Notify call initiator of denied call setup request.	S_i

Table 9-7/G.7713/Y.1704 – Network CallC call setup state transitions

Current state	Event	Actions taken	Next state
S_{sconn}	SetExp	• Notify call initiator of denied call setup request. • Initiate connection release process (for calling party CallC). • Initiate connection release timer ($T_{conn\_release}$).	S_{rconn}
S_{svcall}	SetCallVer	• Send call established message.	S_a
S_{svcall}	SetCallNVer	• Notify call initiator of denied call setup request. • Initiate connection release process (for calling party CallC). • Initiate connection release timer ($T_{conn\_release}$).	S_{rconn}
S_a	SigErr	• None.	S_{sigerr}
S_{sigerr}	SigNErr	• None.	S_a

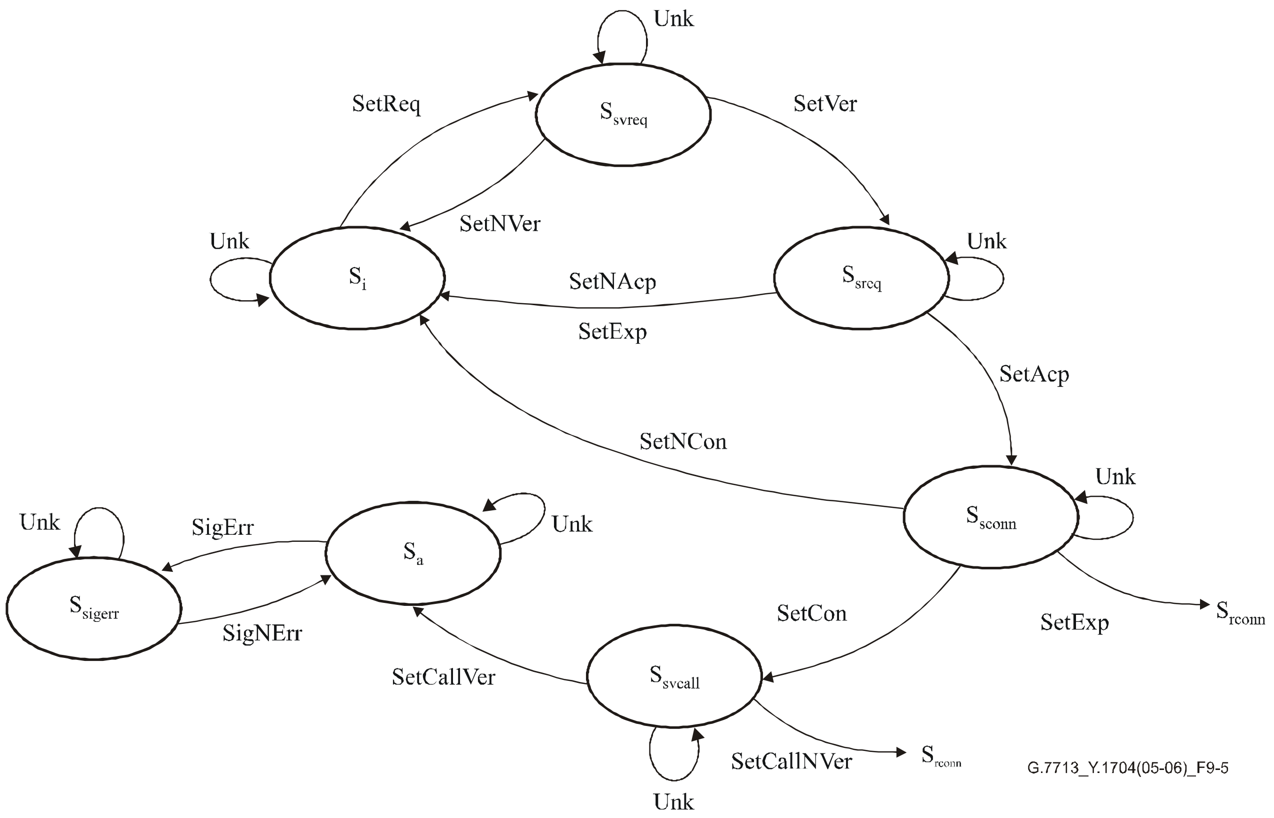


Figure 9-5/G.7713/Y.1704 – Network CallC call setup state diagram

9.1.3.2 Network CallC call state: Release

The state transitions shown in this clause apply to call release for the network CallC (Table 9-8, Figure 9-6).

Table 9-8/G.7713/Y.1704 – Network CallC call release state transitions

Current state	Event	Actions taken	Next state
*	Unk	• Send a notification message to the sender to inform of the error. No state transition occurs.	*
S_a	RelReq	• Verify the request.	S_{rvreq}
S_{rvreq}	RelVer	• Send call release request message. • Initiate connection release process (for calling party CallC). • Initiate connection release timer ($T_{conn\_release}$).	S_{rconn}
S_{rvreq}	RelNVer	• Notify call initiator of denied call release request.	S_a
S_{rconn}	RelCon	• Delete call release timer ($T_{call\_release}$). • Notify call controllers of call release.	S_i
S_{rconn}	RelNCon, RelExp	• Notify call initiator of call release request timeout.	S_i

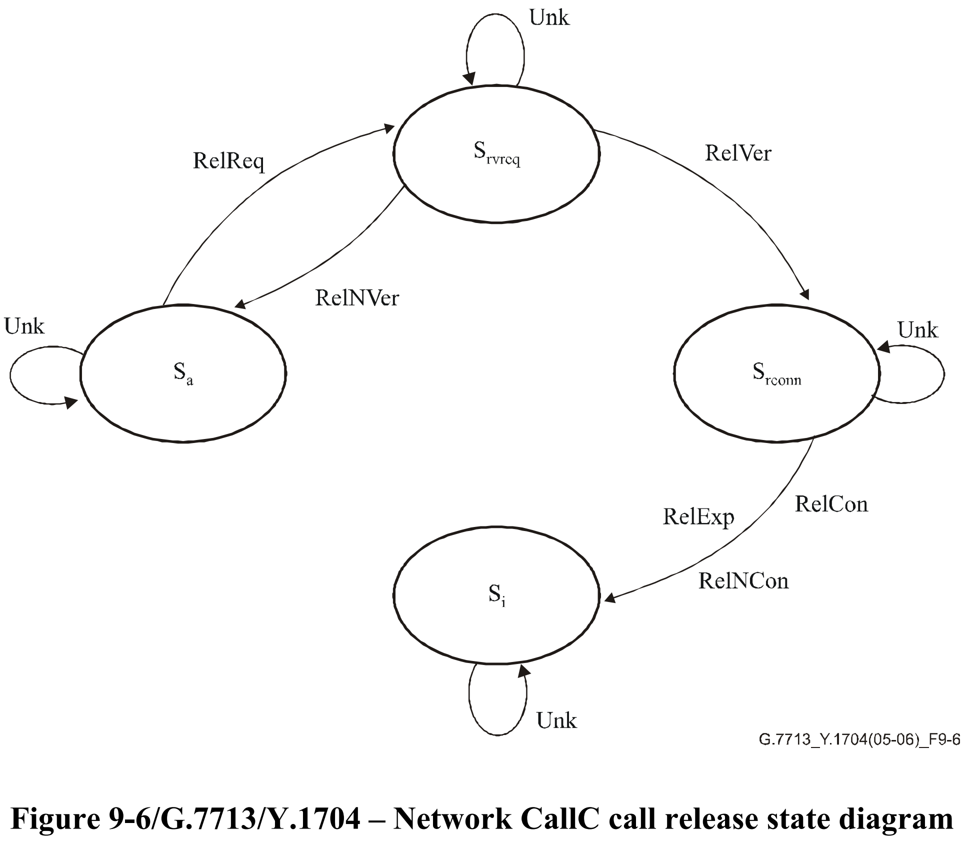


Figure 9-6/G.7713/Y.1704 – Network CallC call release state diagram

9.2 Connection state

The following events are associated with the user CC connection state (Table 9-9):

Table 9-9/G.7713/Y.1704 – User CC connection state events

Event	Event description
Unk	An unknown or unexpected message was received.
SetReq	The user CC received a request to set up a connection.
SetVer	The user CC successfully verified the connection setup request.
SetNVer	The user CC unsuccessfully verified the connection setup request.
SetInd	The user CC received an indication that the connection setup request was successfully processed.
SetNInd	The user CC received an indication that the connection setup request was not processed.
SetCnfm	The user CC received a confirmation message that the connection was successfully established.
SetNCnfm	The user CC received a confirmation message that the connection was not established.
SetExp	Connection setup timer expired.
RelReq	The user CC received a request to release a connection.
RelVer	The user CC successfully verified the connection release request.
RelNVer	The user CC unsuccessfully verified the connection release request.
RelInd	The user CC received an indication that the connection was released.
RelExp	Call release timer expired.
SigErr	Defect detected at the signalling communication channel.
SigNErr	Signalling communication channel defect repaired.

The following events are associated with the network CC connection state (Table 9-10):

Table 9-10/G.7713/Y.1704 – Network CC connection state events

Event	Event description
Unk	An unknown or unexpected message was received.
SetReq	The network CC received a request to set up a connection.
SetVer	The network CC successfully verified the connection setup request.
SetNVer	The network CC unsuccessfully verified the connection setup request.
SetInd	The network CC received an indication that the connection setup request was successfully processed.
SetNInd	The network CC received an indication that the connection setup request was not processed.
SetCnfm	The network CC received a confirmation message that the connection was successfully established.
SetNCnfm	The network CC received a confirmation message that the connection was not established.
SetConnVer	The network CC successfully verified the established connection.
SetConnNVer	The network CC unsuccessfully verified the established connection.
SetExp	Connection setup timer expired.
RelReq	The network CC received a request to release a connection.
RelVer	The network CC successfully verified the connection release request.

Table 9-10/G.7713/Y.1704 – Network CC connection state events

Event	Event description
RelNVer	The network CC unsuccessfully verified the connection release request.
RelInd	The network CC received an indication that the connection was successfully released.
RelNInd	The network CC received an indication that the connection was not released.
RelExp	Connection release timer expired.
SigErr	Defect detected at the signalling communication channel.
SigNErr	Signalling communication channel defect repaired.

9.2.1 Initiating user CC (for SC) or initiating network CC (for SPC) connection state

For the SC service, the following state transition tables apply to the initiating user CC. For the SPC service, the following state transition tables apply to the initiating network CC.

9.2.1.1 Initiating CC connection state: Setup

The state transitions shown in this clause apply to connection setup for the initiating CC (Table 9-11, Figure 9-7).

Table 9-11/G.7713/Y.1704 – Initiating CC connection setup state transitions

Current state	Event	Actions taken	Next state
*	Unk	• Send a notification message to the sender to inform of the error. No state transition occurs.	*
S_i	SetReq	• Verify the request.	S_{svreq}
S_{svreq}	SetVer	• Send connection setup request message. • Initiate connection setup timer (T_{conn_setup}).	S_{sreq}
S_{svreq}	SetNVer	• Notify connection initiator of denied connection setup request.	S_i
S_{sreq}	SetInd	• Delete connection setup timer (T_{conn_setup}). • Optionally send connection confirmation message.	S_a
S_{sreq}	SetNInd	• Delete connection setup timer (T_{conn_setup}).	S_i
S_{sreq}	SetExp	• Notify connection initiator of connection setup request timeout. • Send connection release request message. • Initiate connection release timer ($T_{conn\_release}$).	S_{rreq}
S_a	SigErr	• None.	S_{sigerr}
S_{sigerr}	SigNErr	• None.	S_a

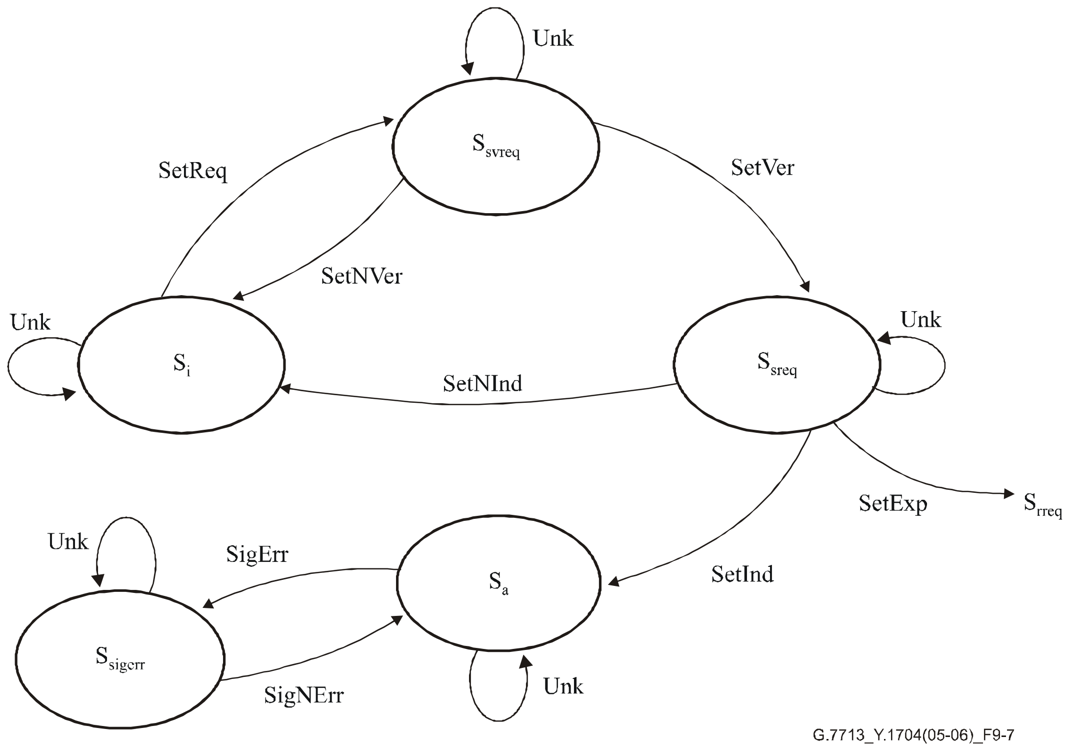


Figure 9-7/G.7713/Y.1704 – Initiating CC connection setup state diagram

9.2.1.2 Initiating CC connection state: Release

The state transitions shown in this clause apply to connection release for the initiating CC (Table 9-12, Figure 9-8).

Table 9-12/G.7713/Y.1704 – Initiating CC connection release state transitions

Current state	Event	Actions taken	Next state
*	Unk	• Send a notification message to the sender to inform of the error. No state transition occurs.	*
S_a	RelReq	• Verify the request.	S_{rvreq}
S_{rvreq}	RelVer	• Send connection release request message. • Initiate connection release timer ($T_{conn\_release}$).	S_{rreq}
S_{rvreq}	RelNVer	• Notify connection initiator of denied connection release request.	S_a
S_{rreq}	RelInd	• Delete connection release timer ($T_{conn\_release}$).	S_i
S_{rreq}	RelExp	• Notify connection initiator of connection release request timeout.	S_i

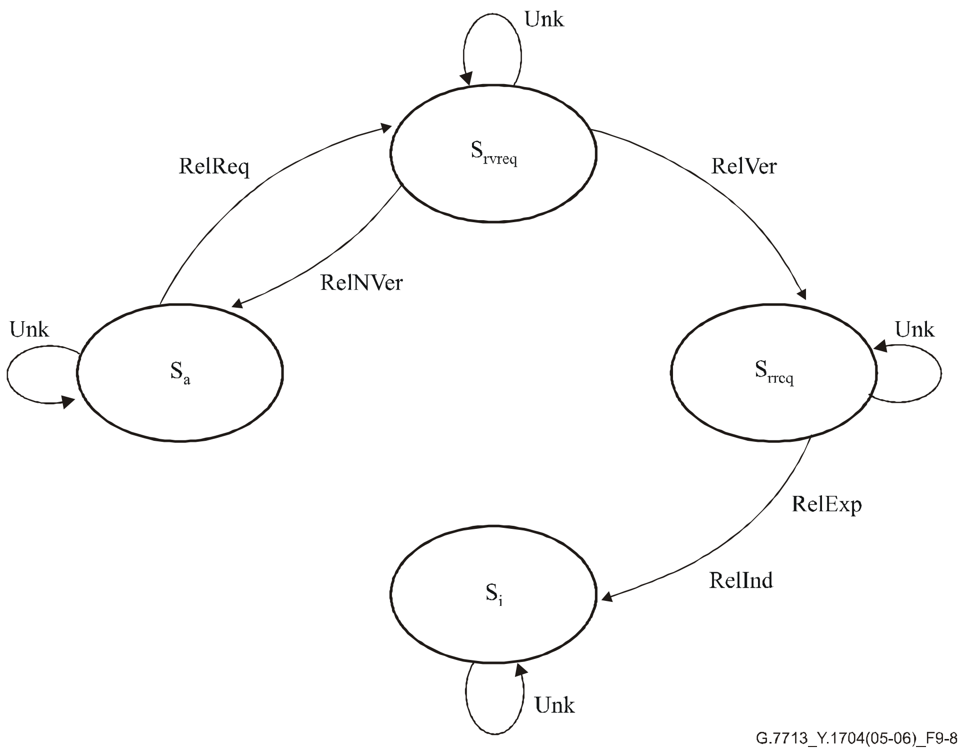


Figure 9-8/G.7713/Y.1704 – Initiating CC connection release state diagram

9.2.2 Terminating user CC (for SC) or terminating network CC (for SPC) connection state

For the SC service, the following state transition tables apply to the terminating user CC. For the SPC service, the following state transition tables apply to the terminating network CC.

9.2.2.1 Terminating CC connection state: Setup

The state transitions shown in this clause apply to connection setup for the terminating CC (Table 9-13, Figure 9-9).

Table 9-13/G.7713/Y.1704 – Terminating CC connection setup state transitions

Current state	Event	Actions taken	Next state
*	Unk	• Send a notification message to the sender to inform of the error. No state transition occurs.	*
S_i	SetReq	• Verify the request.	S_{svreq}
S_{svreq}	SetVer	• Send connection indication message. • Initiate connection setup timer (T_{conn_setup}).	S_{sacpt}
S_{svreq}	SetNVer	• Notify connection initiator of denied connection setup request.	S_i
S_{sacpt}	SetCnfm	• Delete connection setup timer (T_{conn_setup}).	S_a
S_{sacpt}	SetNCnfm	• Delete connection setup timer (T_{conn_setup}).	S_i
S_{sacpt}	SetExp	• Notify connection initiator of connection setup request timeout. • Send connection release request message. • Initiate connection release timer ($T_{conn\_release}$).	S_{rreq}
S_a	SigErr	• None.	S_{sigerr}
S_{sigerr}	SigNErr	• None.	S_a

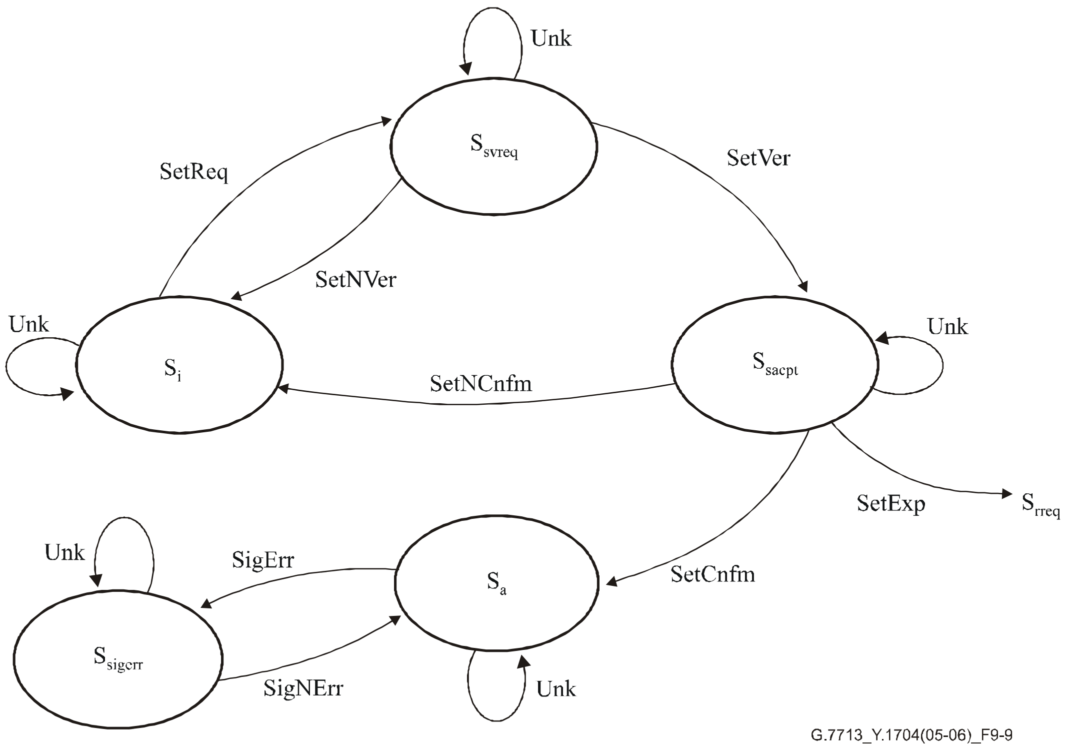


Figure 9-9/G.7713/Y.1704 – Terminating CC connection setup state diagram

9.2.2.2 Terminating CC connection state: Release

The state transitions shown in this clause apply to connection release for the terminating CC (Table 9-14, Figure 9-10).

Table 9-14/G.7713/Y.1704 – Terminating CC connection release state transitions

Current state	Event	Actions taken	Next state
*	Unk	• Send a notification message to the sender to inform of the error. No state transition occurs.	*
S_a	RelReq	• Send connection indication message.	S_i

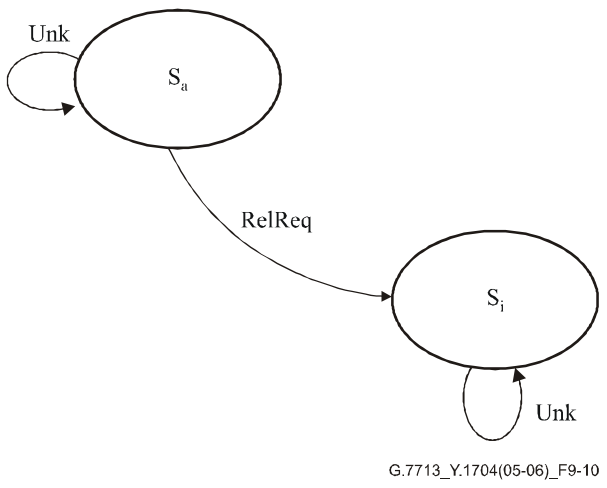


Figure 9-10/G.7713/Y.1704 – Terminating CC connection release state diagram

9.2.3 Intermediate CC connection state

For the SC service, the following state transition tables apply to network CC. For the SPC service, the following state transition tables apply to the intermediate network CC.

9.2.3.1 Intermediate CC connection state: Setup

The state transitions shown in this clause apply to connection setup for the intermediate CC (Table 9-15, Figure 9-11).

Table 9-15/G.7713/Y.1704 – Intermediate CC connection setup state transitions

Current state	Event	Actions taken	Next state
*	Unk	• Send a notification message to the sender to inform of the error. No state transition occurs.	*
S_i	SetReq	• Verify the request.	S_{svreq}
S_{svreq}	SetVer	• Send connection setup request message. • Initiate connection setup timer (T_{conn_setup}).	S_{sreq}
S_{svreq}	SetNVer	• Notify connection initiator of denied connection setup request.	S_i
S_{sreq}	SetInd	• Delete connection setup timer (T_{conn_setup}). • Send connection indication message. • Initiate connection setup timer (T_{conn_setup}).	S_{sacpt}
S_{sreq}	SetNInd	• Delete connection setup timer (T_{conn_setup}). • Notify connection initiator of denied connection setup request.	S_i
S_{sreq}	SetExp	• Notify connection initiator of denied connection setup request. • Send connection release request message. • Initiate connection release timer ($T_{conn\_release}$).	S_{rreq}
S_{sacpt}	SetCnfm	• Delete connection setup timer (T_{conn_setup}).	S_{svconn}
S_{sacpt}	SetNCnfm	• Delete connection setup timer (T_{conn_setup}). • Notify connection initiator of denied connection setup request.	S_i
S_{sacpt}	SetExp	• Notify connection initiator of denied connection setup request. • Send connection release request message. • Initiate connection release timer ($T_{conn\_release}$).	S_{rreq}
S_{svconn}	SetConnVer	• Send connection confirmation message.	S_a
S_{svconn}	SetConnNVer	• Notify connection initiator of denied connection setup request. • Send connection release request message. • Initiate connection release timer ($T_{conn\_release}$).	S_{rreq}
S_a	SigErr	• None.	S_{sigerr}
S_{sigerr}	SigNErr	• None.	S_a

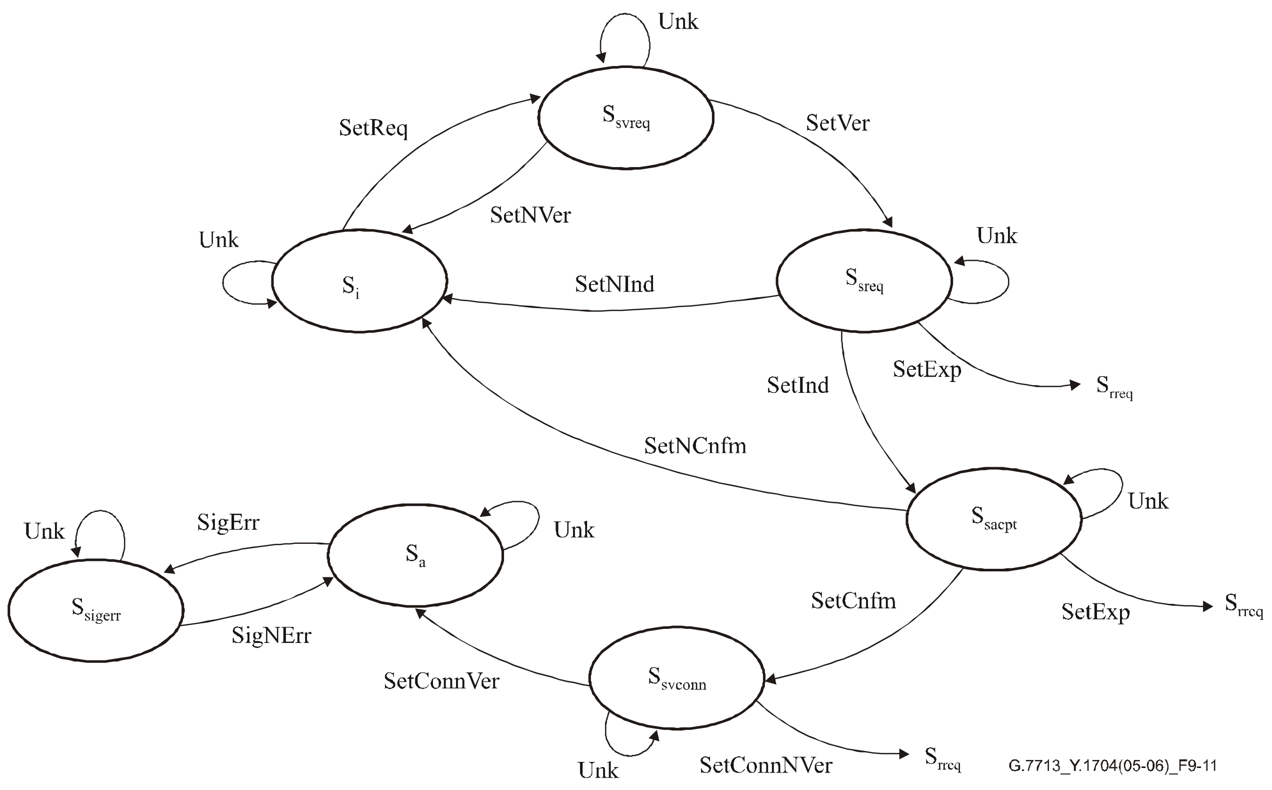


Figure 9-11/G.7713/Y.1704 – Intermediate CC connection setup state diagram

9.2.3.2 Intermediate CC connection state: Release

The state transitions shown in this clause apply to connection release for the intermediate CC (Table 9-16, Figure 9-12).

Table 9-16/G.7713/Y.1704 – Intermediate CC connection release state transitions

Current state	Event	Actions taken	Next state
*	Unk	• Send a notification message to the sender to inform of the error. No state transition occurs.	*
S_a	RelReq	• Verify the request.	S_{rvreq}
S_{rvreq}	RelVer	• Send connection release request message. • Initiate connection release timer ($T_{conn\_release}$).	S_{rreq}
S_{rvreq}	RelNVer	• Notify connection initiator of denied connection release request.	S_a
S_{rreq}	RelInd	• Delete connection release timer ($T_{conn\_release}$). • Send connection indication message.	S_i
S_{rreq}	RelExp	• Notify connection initiator of connection release request timeout.	S_i

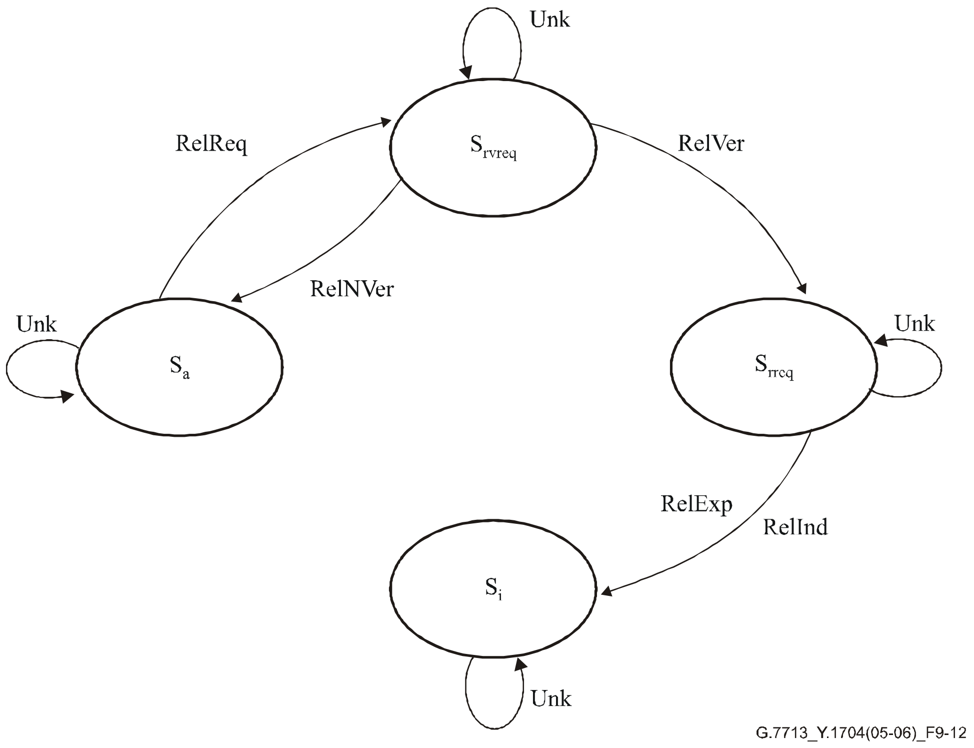


Figure 9-12/G.7713/Y.1704 – Intermediate CC connection release state diagram

10 Management of the call and connection controller function

In this engineering viewpoint selection, this Recommendation has assumed the following distribution of functions:

– Some of the functions of the management plane are to support the capabilities of fault management, configuration management (including resource allocation/de-allocation), performance management, security management and accounting.

– The transport plane supports the capabilities of payload transport, performance monitoring, fault detection, and protection switch.

– The control plane supports the capabilities of dynamic path computation, dynamic distributed call and connection setup/release, dynamic protection/restoration allocations/assignments and restoration.

The call controller, connection controller and link resource manager components defined in ITU-T Rec. G.8080/Y.1304 provide for the capability to supervise and manage calls that are set up and released via distributed connection setup and release. As part of the management and supervision of these requests, communication needs to take place between the CallC, CC and LRM and the management plane. Communications between these components and the MP may include specific information exchanges for configuration of the components, as well as for ensuring the health of the components. This may include specifying various behaviours of the component, to supporting management information signals for exchanging information between the component and the MP. Figure 10-1 illustrates an interface between these components and MP, where the MI information described concerns only the DCM function.

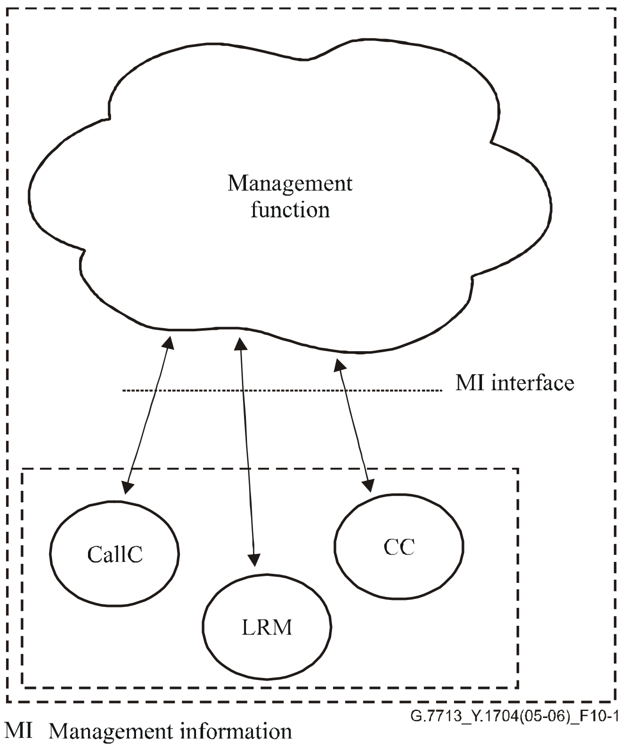


Figure 10-1/G.7713/Y.1704 – Interface between control plane components and management plane

The management system is able to set certain information regarding the components' behaviour and is able to communicate such information to the components. The components may also report certain information over which the components have control. This reported information by the components may be categorized as information that is autonomously communicated by the component to the management system, or information that is communicated to the management system via the management system querying the component. In the case of autonomous communication, an additional signal is described to either enable or disable the automated reporting capability. Note that status information should always be available for retrieval by the MP, even when status reporting is disabled. The following candidate MI signals are controlled by the management function:

– Component status: The MI signals **MI_CallCstatusReporting**, **MI_CCstatusReporting**, and **MI_LRMstatusReporting** allow the management plane to specify whether status reporting is enabled for the identified components, while the MI signal **MI_CallCoperationalState**, **MI_CCoperationalState**, and **MI_LRMoperationalState** autonomously provide information from the components to the management plane regarding the health of the component (i.e., "enabled", capable of performing its functions or "disabled", incapable of performing its functions). The MI signal **MI_CallCproblemList**, **MI_CCproblemList**, and **MI_LRMproblemList** autonomously provides detailed information about the health of the component and its associated interfaces (i.e., list of problems either detected at the component interface or within a component).

  If the **MI_XXstatusReporting** signal is not provisioned by the management plane, then the default is "disabled".

– Signalling mechanism used: For the case where multiple signalling mechanisms may be supported across an interface of the signalling protocol controller (e.g., RSVP-TE, CR-LDP, PNNI), the MI signal **MI_DCMsigMode** allows the management plane to determine which signalling mechanism to use for the signalling protocol controller. The protocol controller may also specify a particular signalling mechanism to use, however, the management plane overrides the decision.

The following candidate MI signals are controlled by the components:

– Tracking of the component performance: The CallC and CC provide the supervision and management of call and connection requests. As such, various parameters may be tracked by these components, including service usage information and call attempt status information. Other types of information such as connection usage are not considered part of these components, and is thus not described. If the component provides tracking of connection requests, then the MI signals are needed to support management query of the control plane. These MI signals are specified for the CallC, CC and LRM components.
  - **MI_CallCcallDuration** provides tracking of the duration of a call sourced by the CallC.
  - **MI_CallCcallState** provides information about the state of a particular call based on the states defined in clause 9.
  - **MI_CCconnectionState** provides information about the state of a particular connection based on the states defined in clause 9.
  - **MI_LRMconnectionState** provides information about the state of a particular link connection based on the states defined in ITU-T Rec. G.8080/Y.1304.
  - **MI_CallCcallAttempt** provides information about the number of call setup requests (or attempts) received by the CallC.
  - **MI_CallCcallBlocked** provides information about the number of call setup requests received by the CallC that are blocked.

– Enhanced call option (monitoring): ITU-T Rec. G.8080/Y.1304 defines a basic connection as a connection without any monitoring capability, and an enhanced connection as a connection with monitoring capability. Monitoring capabilities and impact upon the CC are for further study.

In addition to these MI signals, other types of communications between the control plane and the management plane are applicable. These include communications during setup and release of connections. Clauses 10.1 and 10.2 describe the processes related to setup and release, plus the management communications applicable at different stages of the connection operations.

Table 10-1 provides a summary of the candidate MI signals defined in this Recommendation.

Table 10-1/G.7713/Y.1704 – Candidate MI signals

MI signal	Attributes of signal
MI_CallCstatusReporting MI_CCstatusReporting MI_LRMstatusReporting	"Enabled", "Disabled"
MI_CallCoperationalState MI_CCoperationalState MI_LRMoperationalState	"Up", "Down"
MI_CallCproblemList MI_CCproblemList MI_LRMproblemList	e.g., "Peer component communication failed", "Excessive error interpreting messages"
MI_DCMsigMode	e.g., RSVP-TE, CR-LDP, PNNI

Table 10-1/G.7713/Y.1704 – Candidate MI signals

MI signal	Attributes of signal
MI_CallCcallDuration	Time zone, start time and date, End time and date
MI_CallCcallState	As per states defined in clause 9
MI_CCconnectionState	As per states defined in clause 9
MI_LRMconnectionState	As per states defined in ITU-T Rec. G.8080/Y.1304
MI_CallCcallAttempt	Total number of call setup requests per time period
MI_CallCcallBlocked	Total number of call requests blocked
MI_CallSetupRequest	Attributes are the same as attributes for **CallSetupRequest** message
MI_CallSetupIndication	Attributes are the same as attributes for **CallSetupIndication** message
MI_requestSetupNewRoute	callName connName A-end user name Z-end user name SNP ID SNPP ID Directionality CoS GoS
MI_responseSetupNewRoute	callName connName SNP ID SNPP ID Directionality Explicit resource list Recovery
MI_verifyCAC	callName connName A-end user name Z-end user name SNP ID SNPP ID CoS GoS Explicit resource list Recovery
MI_responseCAC	"Accepted", "Denied"
MI_responseSetupResourcesReserved	"Reserved", "Not reserved"
MI_responseSetupResourcesAllocated	"Allocated", "Not allocated"
MI_responseCallResourceFail	"Resource Failed"
MI_CallReleaseRequest	Attributes are the same as attributes for **CallReleaseRequest** message
MI_CallReleaseIndication	Attributes are the same as attributes for **CallReleaseIndication** message

Table 10-1/G.7713/Y.1704 – Candidate MI signals

MI signal	Attributes of signal
MI_requestExistingRoute	callName connName
MI_responseExistingRoute	callName connName SNP ID SNPP ID Directionality Explicit resource list Recovery
MI_responseReleaseResourcesDeallocated	"De-allocated"
MI_releaseError	Status code the same as specified in 7.2.4.1 for connection release

10.1 Setting up a connection

The following candidate list of MI signals provides interactions between the management function and the control plane function for setting up a connection.

– Two types of connections are supported as per ITU-T Rec. G.8080/Y.1304: switched connections and soft permanent connections (SPCs). In the case of SPCs, the connection setup request is initiated by the management plane. As such, a message is needed across the MI interface, **MI_CallSetupRequest**.

– After receipt of the setup request, the internal process may occur in different ways to establish the resources. This may include processes for authentication of the request, determining route information, verification of resource/route information, and allocation of the resources (allocation of resources may include allocations for protection or restoration if the service profile requires it). Note that some of these processes may not be required because the management system has performed these processes. When the CC requests an outgoing route from the RC (where the RC is distributed in the management plane), additional messages are needed for communication of the route information, **MI_requestSetupNewRoute**, **MI_responseSetupNewRoute**.

  NOTE – To support protection/diversity constraints, multiple routes may be communicated within the signal, i.e., the explicit resource list attribute may contain multiple routes for the protected/diverse connections.

– In the case where the CAC function is distributed to the management plane, additional messages are needed for communication with this function, **MI_verifyCAC**, **MI_responseCAC**.

– Upon verification of the request, resource reservation may need to be performed. An additional message is needed for communicating to the CC that resources have been reserved, **MI_responseSetupResourcesReserved**.

– Upon verification of the request, resource allocation needs to be performed. An additional message is needed for communicating to the CC that resources have been allocated, **MI_responseSetupResourcesAllocated**.

– Once the CC processes are complete, the connection request continues to the downstream CC. The downstream CC is determined by the route established by the RC. Upon completion of the request, a response is received from the downstream CC on the status of the request (e.g., confirmed or denied).

- If the received response is denied, additional processes will be needed to de-allocate and/or un-reserve the resource all the way back to the ingress node (i.e., the call originating node).

– Upon determining the status of the connection operation, a response is sent to the CC. This may be the upstream CC, the network CallC, or a management plane (for the SPC). In the case of SPC, connection setup response is sent to the management plane. As such, a message is needed across the MI interface, **MI_CallSetupIndication**.

  - As part of the **MI_CallSetupIndication**, if the connection was not set up, or errors occurred that resulted in partial setup, the error details are sent to the management plane. This may include the cause of the error as well as any partial links (or link connections) that were set up but cannot be released by the control plane.

– After the call's connection(s) have been established, the CallC relies upon the CC and LRM to monitor the connections. If a connection fails, the CallC must know about it in order to take appropriate actions such as restoration or interrupting usage collection for that call (i.e., in the event that the call was released). As such, a message is needed for communicating to the CallC, **MI_responseCallResourceFail**.

10.2 Releasing a connection

In releasing a call, certain information needs to be exchanged both within the internal and external functions of the various components.

– Two types of connections are supported as per ITU-T Rec. G.8080/Y.1304: switched connection and soft permanent connection (SPC). In the case of SPC, the connection release request is initiated by the management plane. As such, a message is needed across the NMI interface, **MI_CallReleaseRequest**.

– After receipt of the release request, the CC requests RC for the route of the existing connection. In the case where RC is distributed to the management plane, additional messages are needed for communication of the route information, **MI_requestExistingRoute**, **MI_responseExistingRoute**.

– Upon verification of the request, resources need to be un-reserved and de-allocated. An additional message is needed for communicating that resources have been de-allocated, **MI_responseReleaseResourcesDeallocated**.

– Once the CC processes are complete, the connection request continues to the downstream CC. The downstream CC is determined by the route specified by the RC. Upon completion of the request, a response is received from the downstream CC on the status of the request (e.g., confirmed or denied).

  - If the received response is denied, additional processes may be needed to inform the management plane of reasons for denial of the release operation. This communication allows notification to the management plane of the control plane's inability to successfully release an existing call, **MI_releaseError**.

– Upon determining the status of the connection operation, a response is sent to the requesting component. This may be the upstream CC, the network CallC, or a management plane (for the SPC). In the case of SPC, the call release response is sent to the management plane. As such a message is needed across the MI interface, **MI_CallReleaseIndication**.

  - As part of the **MI_CallReleaseIndication**, if the call was not released, or errors occurred that resulted in partial release, the error details are sent to the management plane. This may include the cause of the error as well as any partial links (or link connections) that cannot be released.

ITU-T Y-SERIES RECOMMENDATIONS

GLOBAL INFORMATION INFRASTRUCTURE, INTERNET PROTOCOL ASPECTS AND NEXT-GENERATION NETWORKS

GLOBAL INFORMATION INFRASTRUCTURE	
General	Y.100–Y.199
Services, applications and middleware	Y.200–Y.299
Network aspects	Y.300–Y.399
Interfaces and protocols	Y.400–Y.499
Numbering, addressing and naming	Y.500–Y.599
Operation, administration and maintenance	Y.600–Y.699
Security	Y.700–Y.799
Performances	Y.800–Y.899
INTERNET PROTOCOL ASPECTS	
General	Y.1000–Y.1099
Services and applications	Y.1100–Y.1199
Architecture, access, network capabilities and resource management	Y.1200–Y.1299
Transport	Y.1300–Y.1399
Interworking	Y.1400–Y.1499
Quality of service and network performance	Y.1500–Y.1599
Signalling	Y.1600–Y.1699
Operation, administration and maintenance	**Y.1700–Y.1799**
Charging	Y.1800–Y.1899
NEXT GENERATION NETWORKS	
Frameworks and functional architecture models	Y.2000–Y.2099
Quality of Service and performance	Y.2100–Y.2199
Service aspects: Service capabilities and service architecture	Y.2200–Y.2249
Service aspects: Interoperability of services and networks in NGN	Y.2250–Y.2299
Numbering, naming and addressing	Y.2300–Y.2399
Network management	Y.2400–Y.2499
Network control architectures and protocols	Y.2500–Y.2599
Security	Y.2700–Y.2799
Generalized mobility	Y.2800–Y.2899

For further details, please refer to the list of ITU-T Recommendations.

SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series D	General tariff principles
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	**Transmission systems and media, digital systems and networks**
Series H	Audiovisual and multimedia systems
Series I	Integrated services digital network
Series J	Cable networks and transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Construction, installation and protection of cables and other elements of outside plant
Series M	Telecommunication management, including TMN and network maintenance
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Telephone transmission quality, telephone installations, local line networks
Series Q	Switching and signalling
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	Data networks, open system communications and security
Series Y	**Global information infrastructure, Internet protocol aspects and next-generation networks**
Series Z	Languages and general software aspects for telecommunication systems

Printed in Switzerland
Geneva, 2006