International Telecommunication Union

ITU-T G.7713/Y.1704

(11/2009)

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

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)

Recommendation ITU-T G.7713/Y.1704

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 AND OPTICAL SYSTEMS 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
MULTIMEDIA 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**
PACKET 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.

Recommendation ITU-T G.7713/Y.1704

Distributed call and connection management (DCM)

Summary

Recommendation ITU-T G.7713/Y.1704 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;

– management of DCM.

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

Revisions to this Recommendation include the following changes:

– Addition of interlayer signalling requirements following the ITU-T G.8080 interlayer architecture.

History

Edition	Recommendation	Approval	Study Group
1.0	ITU-T G.7713/Y.1704	2001-12-05	15
1.1	ITU-T G.7713/Y.1704 (2001) Amend. 1	2004-06-13	15
2.0	ITU-T G.7713/Y.1704	2006-05-07	15
3.0	ITU-T G.7713/Y.1704	2009-11-13	15

Keywords

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

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications, information and communication technologies (ICTs). The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

Compliance with this Recommendation is voluntary. However, the Recommendation may contain certain mandatory provisions (to ensure e.g., interoperability or applicability) and compliance with the Recommendation is achieved when all of these mandatory provisions are met. The words "shall" or some other obligatory language such as "must" and the negative equivalents are used to express requirements. The use of such words does not suggest that compliance with the Recommendation is required of any party.

INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU had not received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementers are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database at http://www.itu.int/ITU-T/ipr/.

CONTENTS

Introduction

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

Recommendation ITU-T G.7713/Y.1704

Distributed call and connection management (DCM)

1 Scope

This Recommendation covers the areas associated with the signalling aspects of the 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 support for multiple connections per call, call modification and calls across multiple layers. Areas covered include:

– attribute specifications;

– message specifications;

– signal flows;

– 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 G.8080] 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 G.7712].

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 G.7714].

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

[ITU-T G.709] Recommendation ITU-T G.709/Y.1331 (2009), *Interfaces for the Optical Transport Network*.

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

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

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

[ITU-T G.800] Recommendation ITU-T G.800 (2007), *Unified functional architecture of transport networks*.

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

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

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

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

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

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

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

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

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

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

[ITU-T G.7718] Recommendation ITU-T G.7718/Y.1709 (2005), *Framework for ASON management*.

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

[ITU-T G.8081] Recommendation ITU-T G.8081/Y.1353 (2008), *Terms and definitions for Automatically Switched Optical Networks (ASON)*.

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

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

[ITU-T Q.2931] Recommendation ITU-T 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 Q.2982] Recommendation ITU-T 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

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

– administrative domain;
– layer network;
– link connection;
– management domain;
– subnetwork;
– subnetwork connection.

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

– management information (MI) signal.

The following terms are defined in [ITU-T G.8081]:

– 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;
– signalling controller;
– soft rerouting;
– soft permanent connection;
– switched connection;
– subnetwork point;
– subnetwork point pool;
– transport domain;
– transport resource identifier.

The following term is defined in [ITU-T G.7712]:

– data communication network.

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

– connection termination point (examples of connection termination points for technology specific instances may be found in [ITU-T G.784] (for SDH) and [ITU-T G.874] (for OTN));

– trail termination point (examples of trail termination points for technology specific instances may be found in [ITU-T G.784] (for SDH) and [ITU-T G.874] (for OTN));
– resource.

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

– TPmode/PortMode.

4 Abbreviations

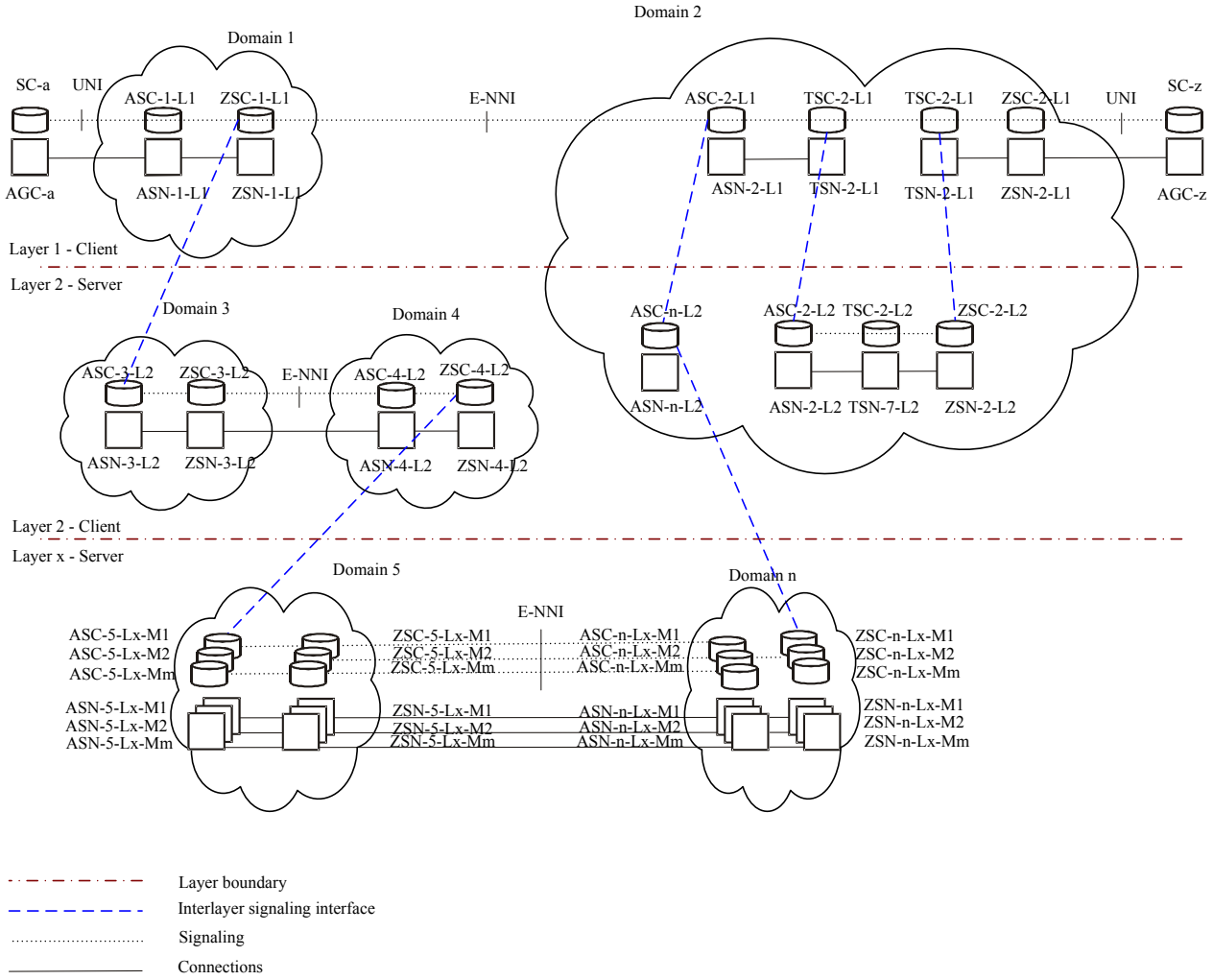
This Recommendation uses the following abbreviations:

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
CO-PS Connection Oriented Packet Switching
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
NE	Network Element
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
TRI	Transport Resource Identifier
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.



Subnetworks and AGCs

AGC-a	A-end location
AGC-z	Z-end location
ASN-n-[Lx]-[Mm]	A-end subnetwork in domain n, in layer x, m^{th} inverse multiplexed component
TSN-n-[Lx]-[Mm]	Transit subnetwork in domain n, in layer x, m^{th} inverse multiplexed component
ZSN-1-[Lx]-[Mm]	Z-end subnetwork in domain n, in layer x, m^{th} inverse multiplexed component

Signalling Controllers (CCC and CC)

SC-a	A-end user signalling controller
SC-z	Z-end user signalling controller
ASC-n-[Lx]-[Mm]	A-end signalling controller in domain n, in layer x, m^{th} inverse multiplexed component
TSC-n-[Lx]-[Mm]	Transit signalling controller in domain n, in layer x, m^{th} inverse multiplexed component
ZSC-n-[Lx]-[Mm]	Z-end signalling controller in domain n, in layer x, m^{th} inverse multiplexed component

Figure 5-1 – 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-n-[Lx]-[Mm], TSN-n-[Lx]-[Mm], ZSN-n-[Lx]-[Mm], and AGC-z.

When describing interlayer scenarios, a layer "Lx" may be added to the above abbreviations as follows, for an NCC in domain n at layer x, it is an "NCC-n-Lx". If inverse multiplexing is described, for the server layer providing m components that are multiplexed together to form the client layer, "Mm" may be added to the abbreviation. For example, the NCC for the mth component in the server layer is represented as "NCC-n-Lx-Mm".

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 G.8080], 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 set up 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.

Identifiers used for call and connection management must come from a different name space than the DCN identifiers to prevent changes in the DCN network from impacting call and connection management.

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 G.8080], 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 G.8080]).

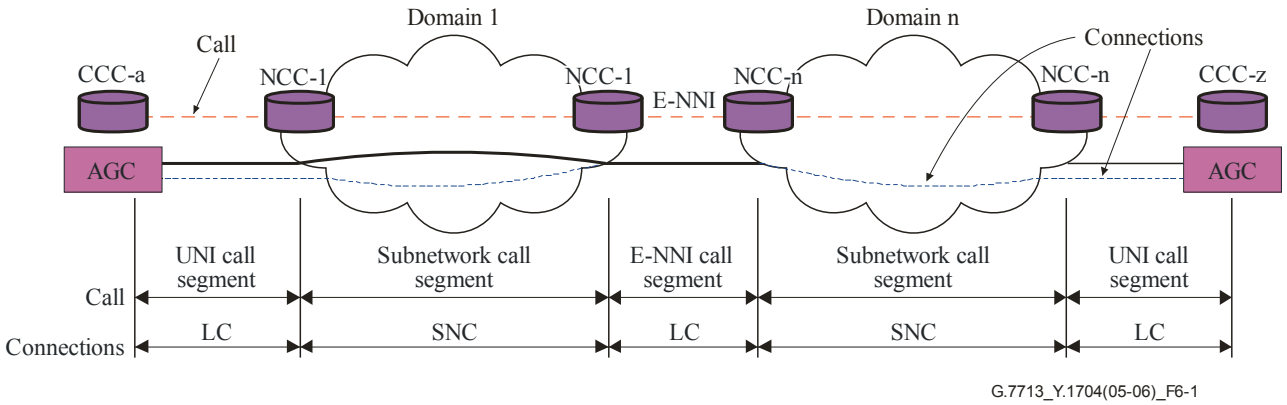


Figure 6-1 – Call segments and connections

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 G.8080], 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 set up of different types of resources to create a connection. Figure 6-2 illustrates the connection that is set up to support a call.

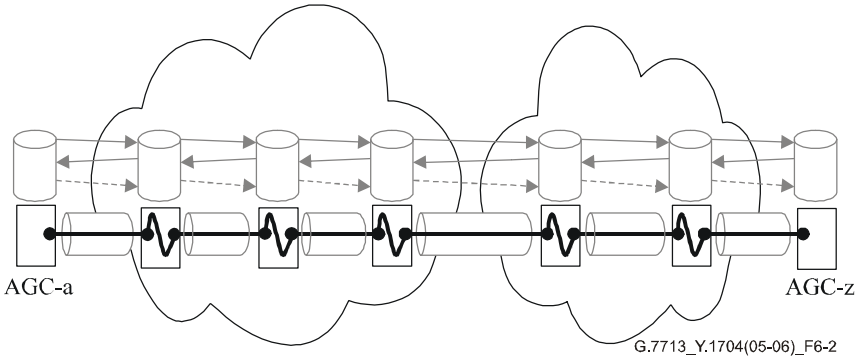


Figure 6-2 – LC and SNC establishment for a call request

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 G.8080].

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

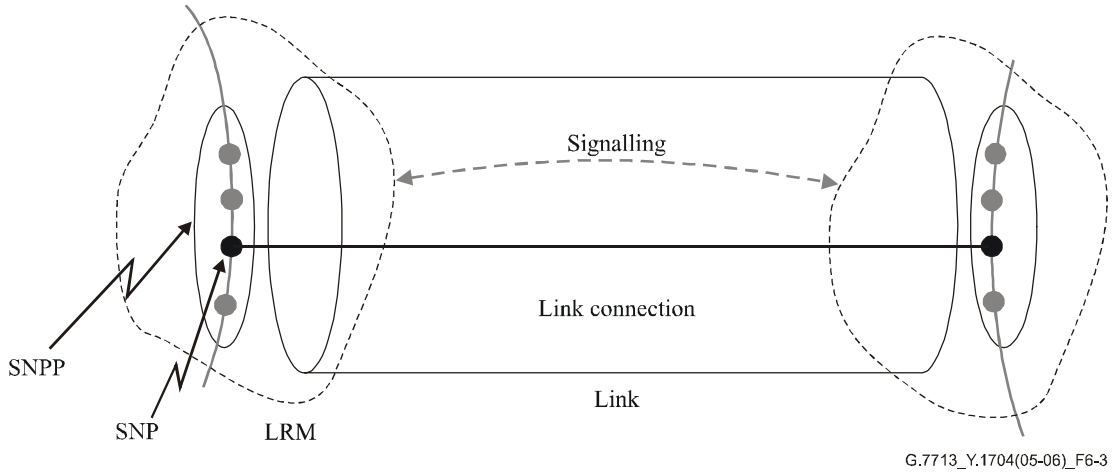


Figure 6-3 – Establishment of link connection by allocating the SNPs

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 set up 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 G.8080] 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.

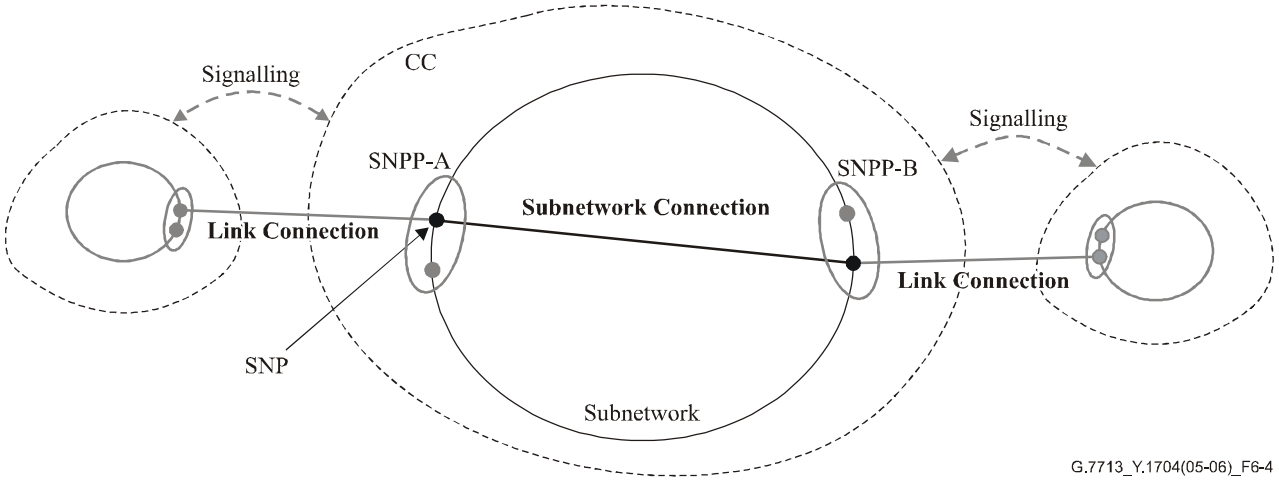


Figure 6-4 – Creation of subnetwork connection after link connection establishments

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 clause 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 set up of a call, and the associated signal flows between the relevant components.

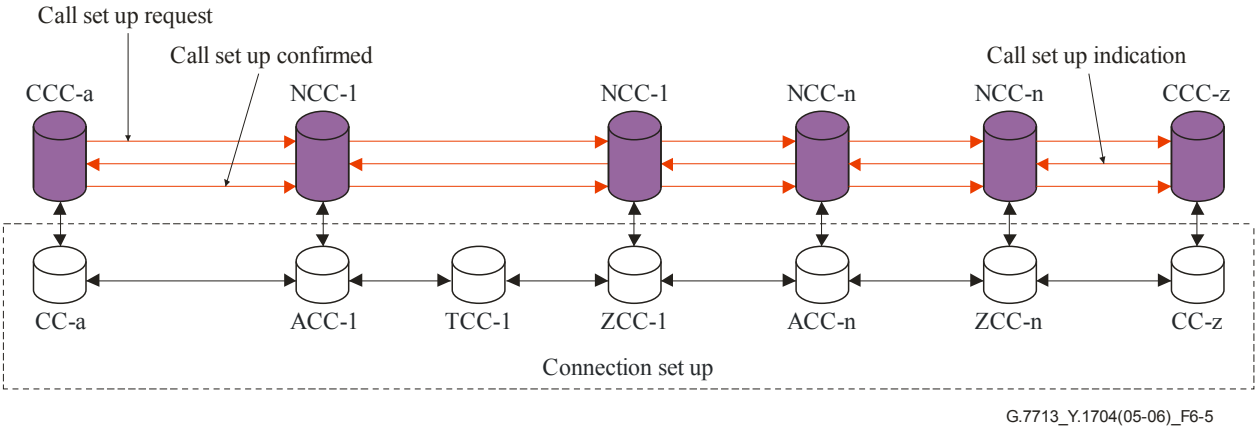


Figure 6-5 – Call set up request processing: Logical request progression

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

– The calling party call controller (CCC-a) requests call set up. 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 set up request by initiating a connection set up request to the CC. The process for connection set up request is described in clause 6.1.2. Note that, based on different protocol design decisions, initiation of connection set up 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 set up request process (across all call segments) the call set up request is successfully completed, and transfer of user characteristic information may begin.

If the connection set up 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 G.8080] 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.

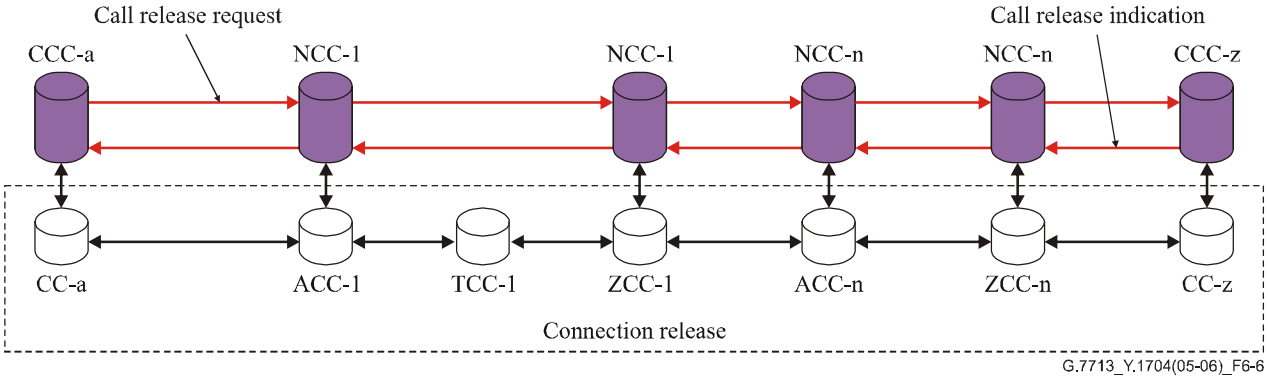


Figure 6-6 – Call release request processing: Logical request progression

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 clause 6.1.2.2. 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 set up 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 affect 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.

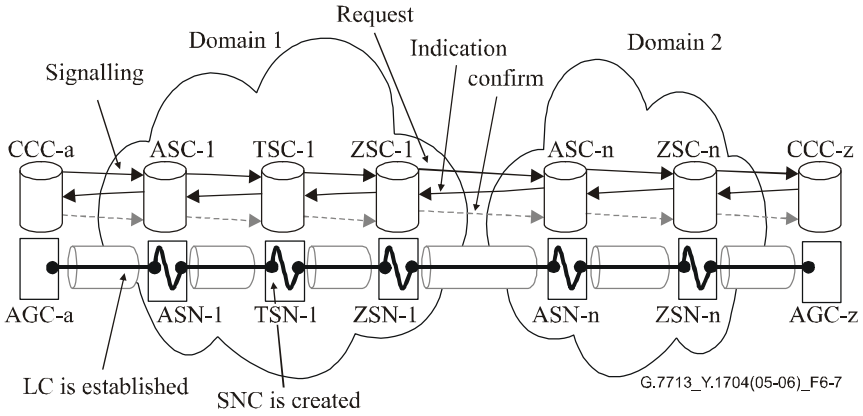


Figure 6-7 – LC establishment and SNC creation for a connection set up request

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.).

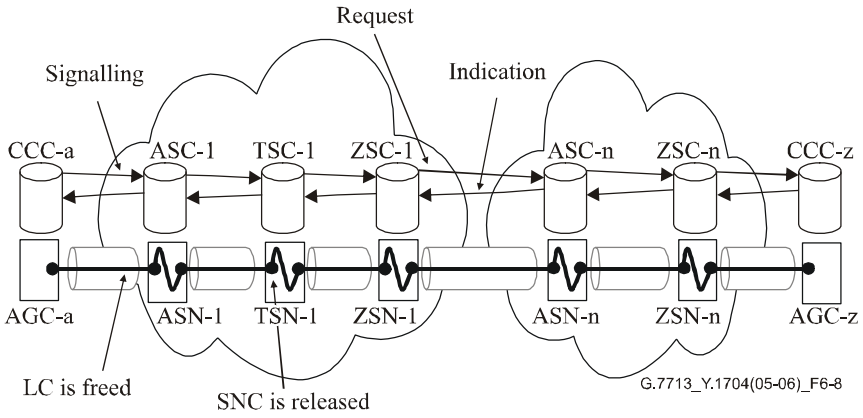


Figure 6-8 – Freeing LC and releasing SNC for a connection release request

6.1.2.1 Process for setting up connections

The following processes are performed for the connection set up:

– According to clause 6.1.1.2, a set up 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 set up 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 set up 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 time slot 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 clause 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.

6.1.2.3 Process for modifying connections

Modifying connections is limited to CO-PS connections that have a data rate associated with them, e.g., Ethernet. Two types of modification are supported: increase and decrease of connection CO-PS data rate. The following processes are performed for the CO-PS data rate connection modification:

– According to clause 6.1.4, a modify call request has been verified and allowed to proceed. Upon indication to proceed with the modify connection, the modify process is initiated at the ASC-1 CC.
– The modify connection request is carried from ASC-1 to ZSC-1, then similarly between ASC-n and ZSC-n until CCC-z CC is reached. At each CC, a modification is made to the CO-PS data rate associated with the LC or SCN. This process is continued until the connection modification reaches the CCC-z CC.
– When the connection request successfully reaches CCC-z CC, the connection modification request is complete and an indication is sent in the reverse direction. At each CC, a modification is made to the CO-PS data rate associated with the LC or SCN.

– After the CC processes the connection release request, a response message is sent to indicate the connection modify is processed.

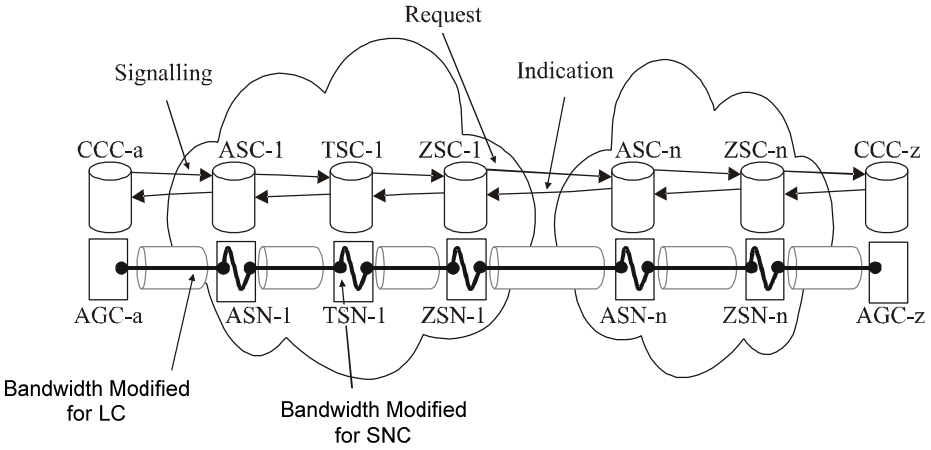


Figure 6-9 – CO-PS data rate modification in LC and SNC for a connection modify request

6.1.3 Interlayer signalling

A capability to map client characteristic information to a server layer is described in clause 6.6 of [ITU-T G.8080]. This capability may be used to provide for transfer of client characteristic information when the client layer is not continuous or connected between two access points. When the mapped CI is transported across the server layer, it is adapted back to the client layer where it may continue. The mapping can be controlled through ASON component interaction between network call controllers in adjacent layers. This pair wise relationship of NCCs is scoped to equipment that is capable of supporting multiple layers and adaptations between them.

6.1.3.1 Interlayer call set up

Steps for interlayer signalling control are described below assuming that a call/connection is progressing in the same signalling sequence. It is also assumed that client layer routing can compute a route that involves representations of server layer connectivity to which client CI can be mapped, and that it can be distinguished to connection controllers, when use of the server layer connectivity is included in a route.

1) When a connection request setup is received by a CC, it either includes a route or the CC obtains one from the routing component. The CC uses the route information to determine what the next CC is to pass the request to. If the next hop is an indicator that a server layer is needed, then perform step 2, otherwise perform connection set up as previously described and do not continue with the steps that follow.

2) An interlayer call is necessary because, according to the route, the next hop does not occur in the client layer network. The CC corresponds to the SNPP at the current hop position in the route which is at the present interlayer boundary. The CC should determine the SNPP in the route at which the next interlayer boundary occurs and this becomes the SNPP at which the mapped CI will be adapted back into the client layer.

3) The NCC corresponding to the CC is informed of the need for an interlayer call. An NCC must be present at this boundary.

4) The NCC uses the pair of SNPPs in the client layer that are to be connected in the server layer, as input to the interlayer address resolution function. This function returns a pair of UNI transport resource identifiers in the server layer. These identifiers are not associated with a UNI link but rather a set of access points in the server layer from which a connection can be created for the purpose of carrying mapped client CI.

5) Call the appropriate server layer NCC(s) associated with the adaptation "between" the client NCC (where this procedure is running) and server NCC(s). The call includes the UNI transport resource identifiers and any other parameters needed to obtain server layer connections that can satisfy the call parameters of the client call request.

6) If successful, the server layer NCC will return a pair of TCPs that can be used by the client/server adaptation for transferring client information. The request can be satisfied by creating server layer calls, or returning pairs of TCPs from existing server layer calls. From a client's perspective, the two cases cannot be distinguished.

7) When the client NCC receives the pairs of TCPs, it hands them to the CC to bind local TCPs to the adaptation function that accomplishes the mapping of the client CI to the server layer.

8) The call/connection setup continues at the interlayer boundary where the client CI is adapted back into the client layer. At this point, the other TCP is bound to the adaptation.

Figure 6-10 shows the resources of three adjacent layers. Note that domain 2 is not connected in that layer to domain 1.

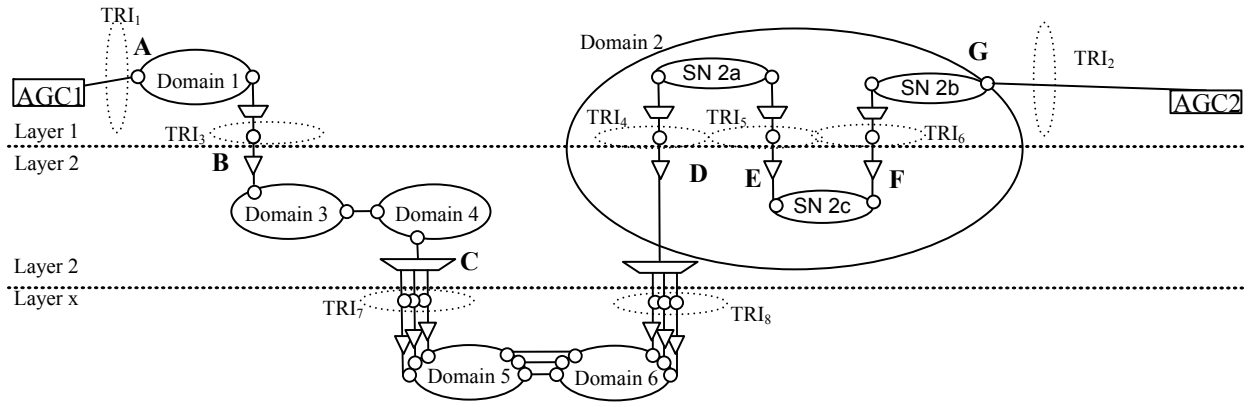


Figure 6-10 – Resource view for recursive interlayer with inverse multiplexing

Figure 6-11 shows a control plane organization of the same resources illustrated in Figure 6-10. At interlayer boundaries, there exists interlayer NCC relationships denoted by dotted lines. A UNI transport resource identifier, TRI_1 is assigned to the UNI link connecting Client1 (on AGC1) to domain1. Similarly, TRI_2 is assigned to the UNI link connecting Client2 (on AGC2) to domain2 (SN 2b). TRI_3 is associated with an AP in domain 3 and is used as the source of a layer 2 call in domain 3. TRI_4 is associated with an AP in domain 2 and is used as the destination of the first layer 2 call. TRI_7 is associated with a set of APs in domain 5 and is used as the source of the call in layer x. TRI_8 is associated with a set of APs in domain 6 and is used as the destination of the call in layer x. TRI_5 is associated with an AP in domain 2 (SN 2c) and is used as the source of the second layer 2 call. TRI_6 is associated with an AP in domain 2 (SN 2c) and is used as the destination of the second layer 2 call.

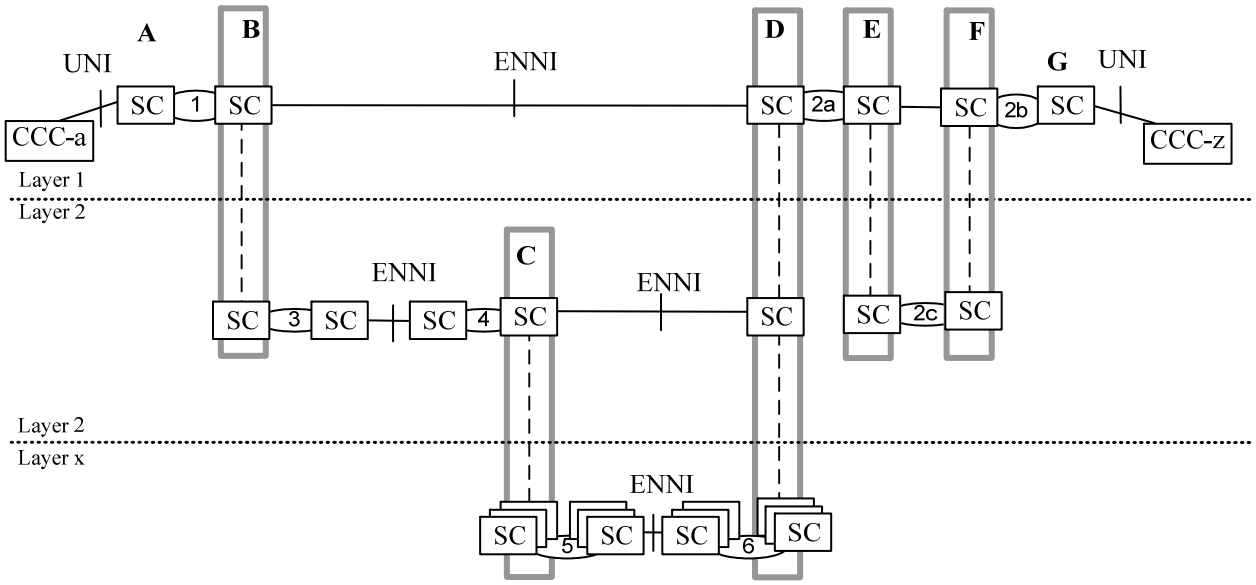


Figure 6-11 – Signalling controllers organization

Applying the interlayer procedure to Figures 6-10 and 6-11, setting up a call from Client1 to Client2 would have the high-level sequence illustrated in Figure 6-12 for the mapped server case, following alphabetical ordering of steps. To simplify the figure, 1:1 correspondence between CCs and NCCs is assumed at boundaries, SCs are represented. For the independent set up scenario, the order may vary depending on which layer is pre-established. For example, f through g could be performed first, followed by a through e and h through zb if layer x was independently established. Another example is d through k is performed first, followed by a through c and l through zb if layer 2 and x were established prior to layer 1. In all cases, the requirement is that the server layer connections are set up before the client layer call establishment completes. Interlayer SC-SC interactions are specifically NCC-NCC interactions. Within a layer, SC-SC interactions include both NCC-NCC and CC-CC interactions.

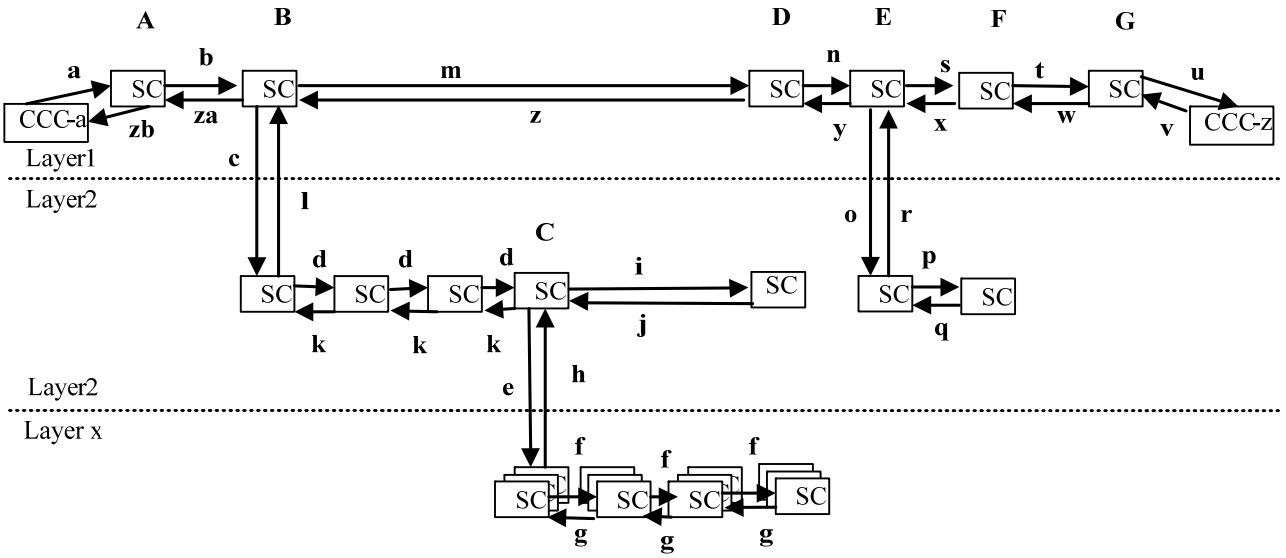


Figure 6-12 – High-level sequence

6.1.3.2 Interlayer call release

In the case of interlayer call release, there is a requirement that the client layer call is informed of the release prior to removing the server layer call. In Figure 6-12, assuming the arrows represent the call release request (→) and call release indication (←), this would correspond to a-b-k-l-q-r-s-t-u-v-w completing prior to m through p, x completing prior to c through j.

When a call on the client side of an interlayer boundary is released, the association with its server layer call or calls is removed. The server NCC coordination out interface from the ITU-T G.8080 NCC component is used to inform the server layer(s). This is similar to initiating a connection release in the non-interlayer case. Server layer calls involved in the removal of the association may or may not be released and this is subject to server layer policy. In cases where the server layer does not release the call, it is then available for other client interlayer associations.

When a call on the server side of an interlayer boundary is released, the association with its client layer call or calls is removed. The client NCC coordination out interface from the ITU-T G.8080 NCC component is used to inform the client layer(s). This is similar to initiating a connection release in the non-interlayer case. Client layer calls involved in the removal of the association may or may not be released and this is subject to client layer policy. The client layer call may make another server layer call as a part of a recovery process or it may also choose to release itself.

6.1.4 Call modification for non-disruptive bandwidth modification

Non-disruptive bandwidth modification is defined as a call modification that does not impact user traffic. In order for the control plane to support non-disruptive bandwidth modification, the underlying transport plane technology must support the ability to non-disruptively modify the bandwidth. There are three types of non-disruptive bandwidth modification to consider:

- Type 1: Connection modification: Call modification that increases/decreases the data rate of an existing CO-PS connection. A change in data rate of a user flow is made that is within the available resources for the connection. For example, an Ethernet connection is modified by increasing its CO-PS data rate from 40 Mbit/s to 60 Mbit/s.
- Type 2: Connection resizing: Call modification that increases/decreases the number of connections associated with the call. For example, a call is decreased from three STM-16 connections to two STM-16 connections. This is not an inverse multiplexing case in the layer being considered. Mechanisms in the client layer that use the modified call are expected to adjust to the change in number of connections.
- Type 3: Group resizing: Call modification that increases/decreases the group size. This is an inverse multiplexing case. For example, a call is increased from a VC-4-5v to a VC-4-6v .

There are similarities between the last two bullets in that the number of connections is modified. There is a major difference: the number of layers involved in the network.

Scenario 1: Single layer – Connection modification (type 1) and connection resizing (type 2)

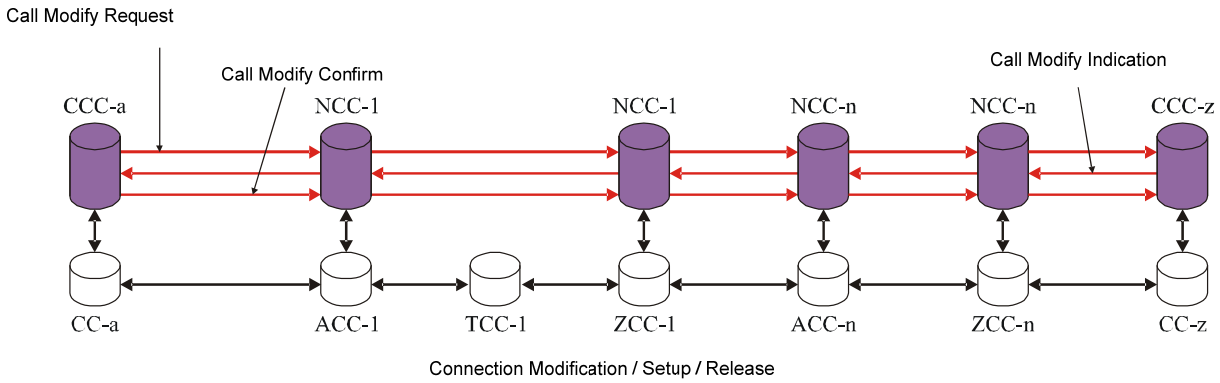


Figure 6-13 – Connection modification and connection resizing

For a call modify request for non-disruptive bandwidth modification, the steps include:

– The calling party call controller (CCC-a) requests call modification. At the ingress NCC-1, processes are initiated to check the call modification (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 6-13) may include verifying that the call modification is accepted end-to-end (e.g., request for CCC-z call verification).

– Upon successful indication by the call modification request process, the calling party call controller (CCC-a) continues the call modification request by initiating a connection modification request to the CC or by requesting the set-up of a new connection or release of an existing connection. The process for connection set up request and connection release is described in clause 6.1.2. The process for connection modification is described in clause 6.1.2.3. Note that, based on different protocol design decisions, initiation of connection modification/setup/release request may occur in a different order as shown in Figure 6-13. The only requirement is that a network connection modification/setup/release is completed before the call modification is completed.

– Upon successful indication by the connection modification/setup/release request process (across all call segments), the call modification request is successfully completed.

An interface exists between a call controller and connection controller in the [ITU-T G.8080] component description for initiating a connection setup/release/modification request, but is not described in this Recommendation.

Scenario 2: Interlayer group resizing (type 3)

This scenario describes a case where a bandwidth modification at one layer triggers a group size modification at server layer, in turn triggering the set up or release of its server layer calls. Figure 6-14 illustrates a group size modification for an inverse multiplexing scenario.

NOTE – An example of this scenario occurs when a virtual concatenated signal is operational and link capacity adjustment scheme (LCAS) in ITU-T G.7042 is applied to the virtual concatenation group (VCG) to change the size of the VCG without affecting the traffic in the virtual concatenated signals that are not added or removed from the VCG. For a decrease, LCAS is run prior to connection release in control plane; and for an increase, LCAS is run after a new connection is set up in control plane.

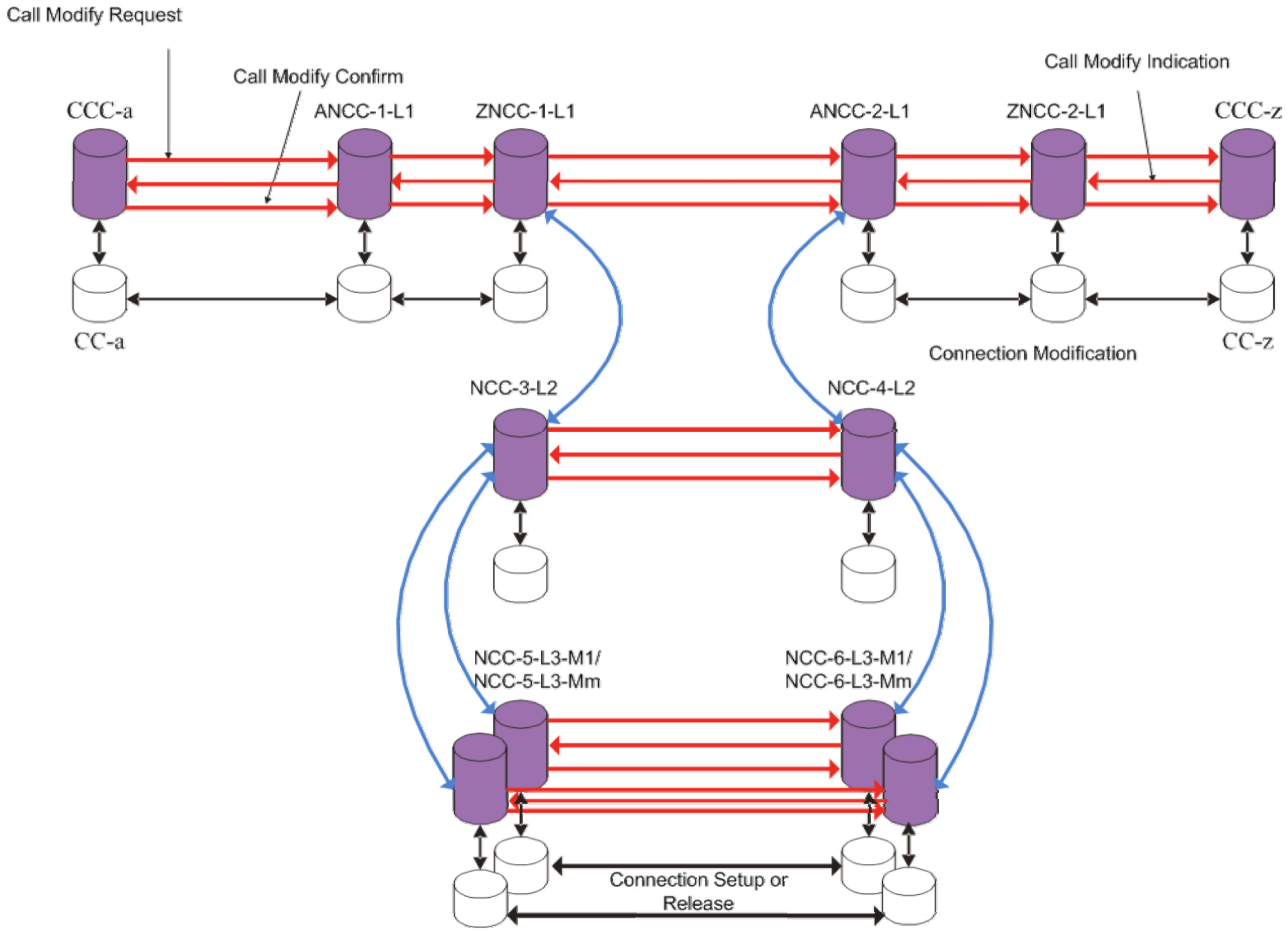


Figure 6-14 – Interlayer group resizing

For a call modify request to increase/decrease the group size, the steps include:

– The calling party call controller (CCC-a) requests call modification. At ANCC-1-L1, processes are initiated to check the call modification (this may include checking for authentication and integrity of the request as well as constraints placed by policy decisions). Processes included in ZNCC-2-L1 may include verifying that the call modification is accepted end-to-end (e.g., request for CCC-z call verification). The request is also sent to the intermediate network call controllers. At interlayer call boundaries, ZNCC-1-L1, an interlayer call modification request is made to NCC-3-L2 to modify the group size. In turn, NCC-3-L2 requests a new call set up to NCC-5-L3-M1 to NCC-5-L3-Mm for a group size increase or a call release for a group size decrease.

– Upon successful indication by the call modification request process, the calling party call controller (CCC-a) continues the call modification request by initiating a connection modification request to the CC. Similarly, the call controllers at interlayer boundaries continue the modification request by initiating a connection modification request, connection set up request or connection release request to the corresponding connection controller.

– Upon successful indication by the connection modification request process (across all call segments at all layers) the call modification request is successfully completed.

– Note that, based on different protocol design decisions, initiation of connection modification/setup/release request and interlayer call requests may occur in a different order as shown in Figure 6-14. The only requirements are as follows:
  - Overall network connection modification/setup/release is completed before the call modification is completed.
  - For increases, call and connection modifications at server layers must be completed before the client layer call modification completes.
  - For decrease, the request must be validated at the client layer before the interlayer call modification request is made, i.e., the client layer must be in the Call Modify Indication or Call Modify Confirm phase.

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 set up 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 clauses provide different

scenarios, as shown from Figures 6-15 to 6-37, the signalling flow due to set up and release of calls, and the various defects that may occur during these operations.

6.3.1 Set up 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.

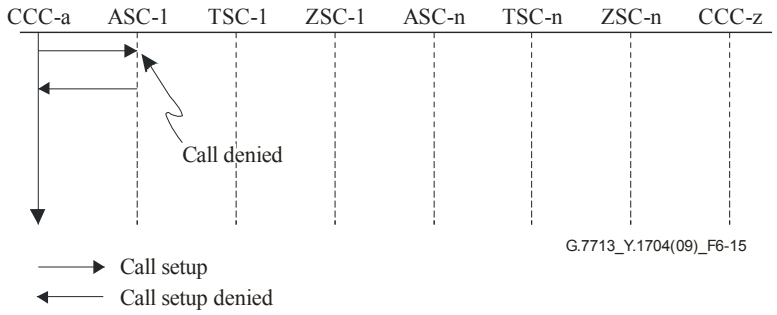


Figure 6-15 – Set up → CCC-a UNI defect (call denied)

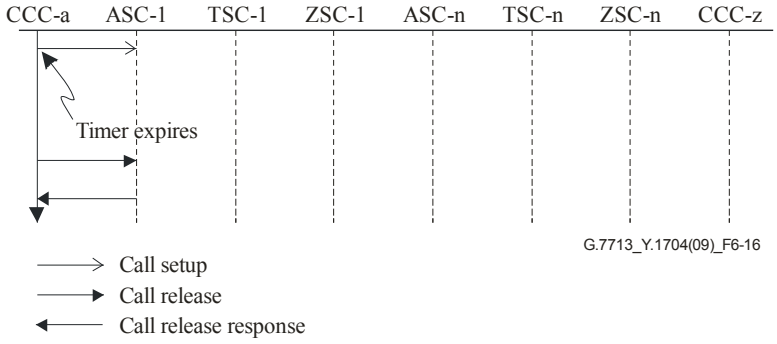


Figure 6-16 – Set up → CCC-a UNI defect (timer expires)

Figures 6-15 and 6-16 show two examples of a set up rejection. In Figure 6-15, different defects occurred at ASC-1 that resulted in denied call set up.

The second example (Figure 6-16) 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 set up 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 6-17 to 6-20 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.

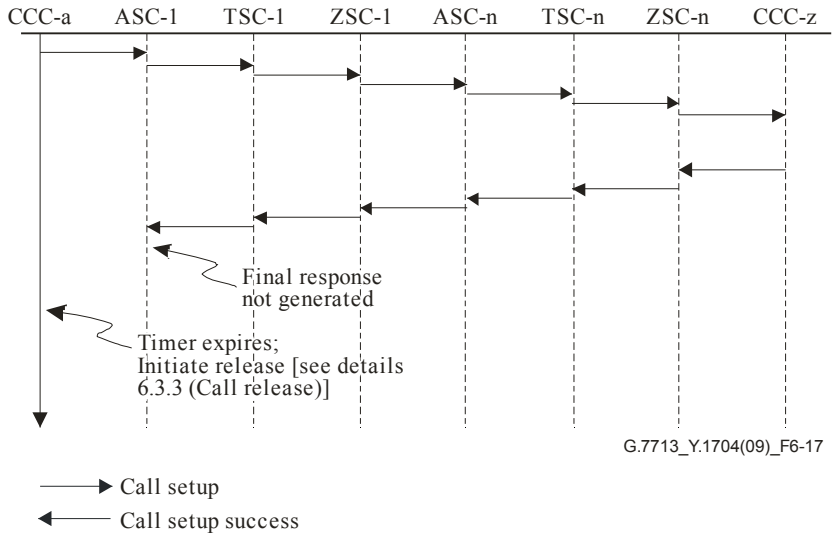


Figure 6-17 – Set up → Signalling defect of the response message (final response not generated)

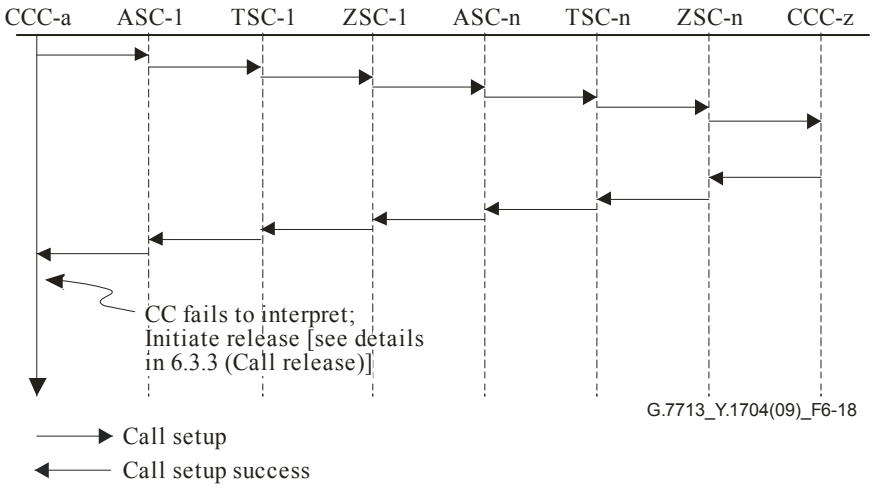


Figure 6-18 – Set up → Signalling defect of the response message (CCC-a fails to interpret)

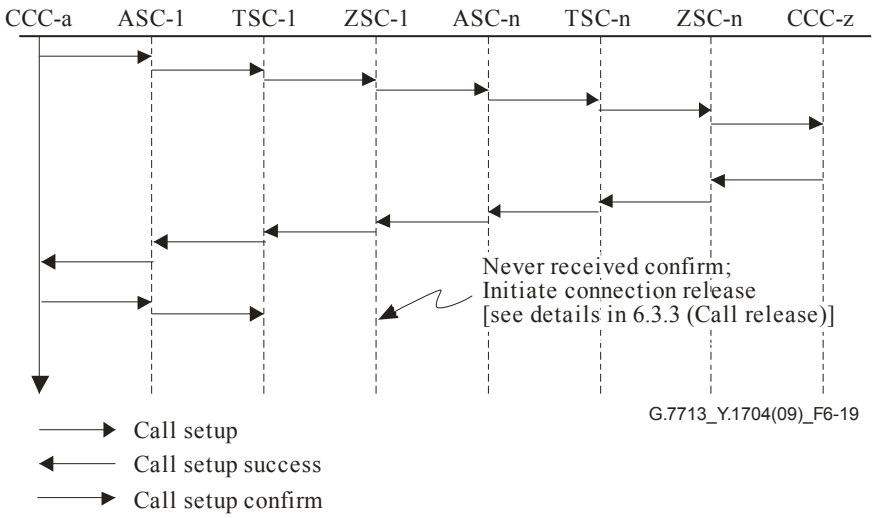


Figure 6-19 – Set up → Signalling defect of the confirmation message (confirmation message lost during transmission)

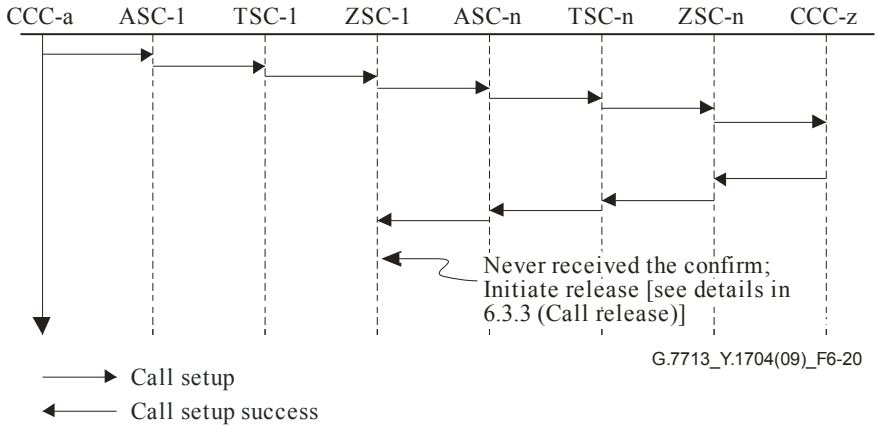


Figure 6-20 – Set up → Signalling defect of the confirmation message (confirmation message never generated)

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.

– Unrecognized signalling information.

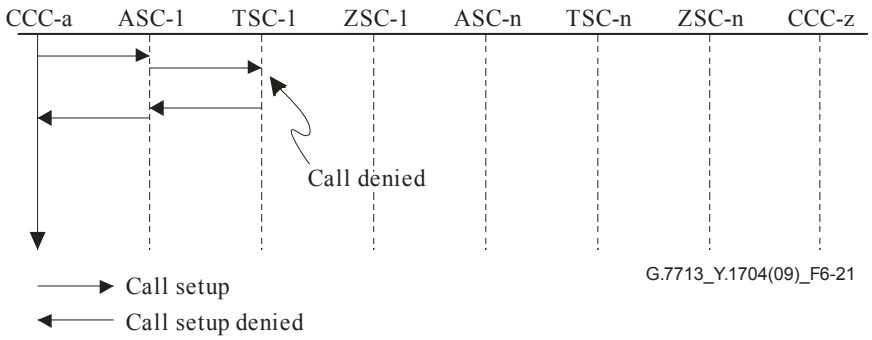


Figure 6-21 – Signal flow: Set up → Intra-domain defect (call denied)

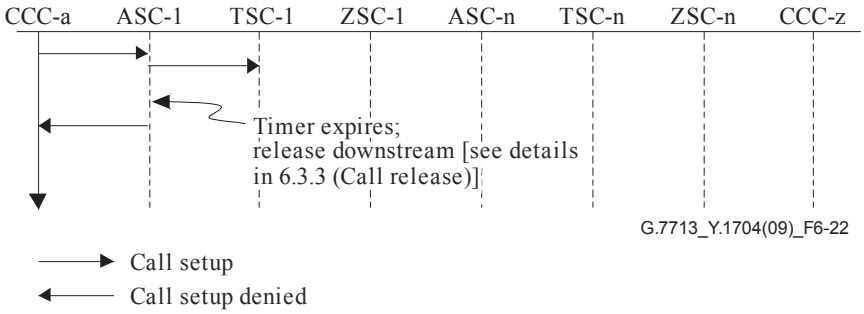


Figure 6-22 – Signal flow: Set up → Inter-domain defect (timer expires)

In Figures 6-21 and 6-22, 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.

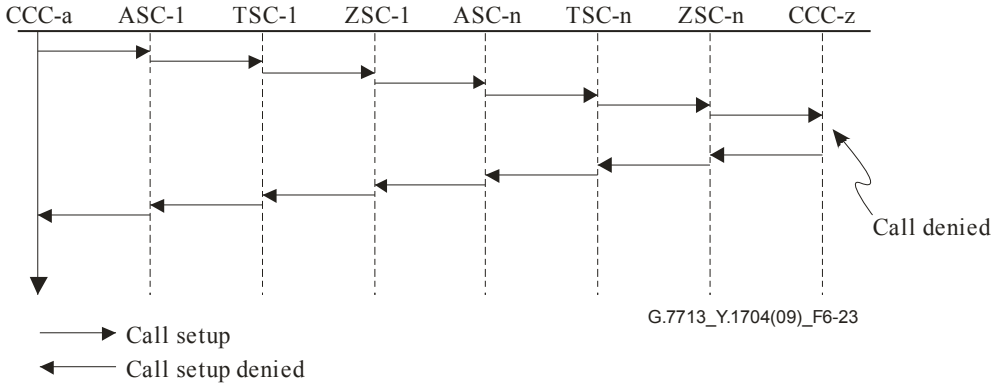


Figure 6-23 – Signal flow: Set up → CCC-z UNI defect (call denied)

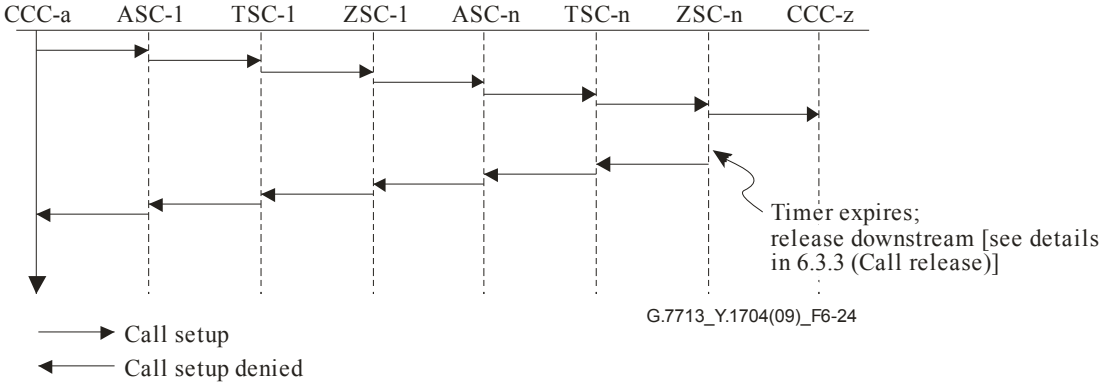


Figure 6-24 – Signal flow: Set up → CCC-z UNI defect (timer expires)

In this defect scenario (see Figure 6-23), 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 set up).

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

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

In the case where the timer expires (see Figure 6-24), 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, three behaviours may be possible depending on the type of call requested:

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

– For an interlayer call, the server layer connections are released if hard restoration is not used or is not useful. The server layer call remains in place even if no connections remain. The requesting client layer call controller is notified of the failure in order to allow for restoration to be attempted at a higher layer. The client layer call controller may release the server layer call.

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 6-25 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).

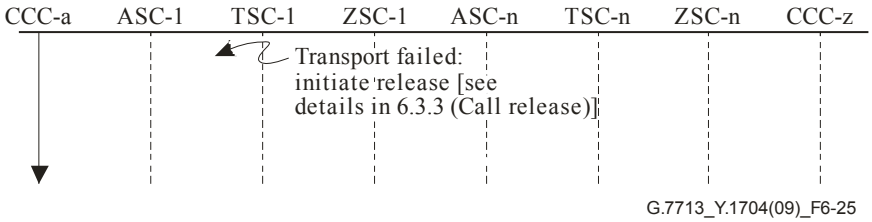


Figure 6-25 – Signal flow: Existing call → User transport network connection defect

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 6-26 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.

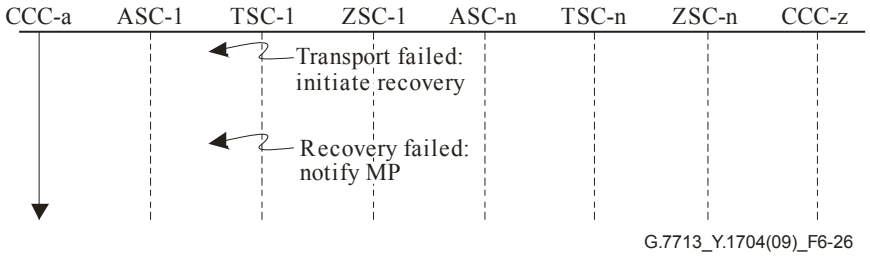


Figure 6-26 – Signal flow: Existing call → Transport network connection defect

6.3.2.3 Transport network connection defect (interlayer)

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 6-27 illustrates the case where a connection defect was detected within the network in a server layer. As a result of the defect, recovery procedures may be initiated to restore/protect the connection in the server layer. 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 client layer call controller. The failed call remains up until explicit release is received from the client call controller.

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

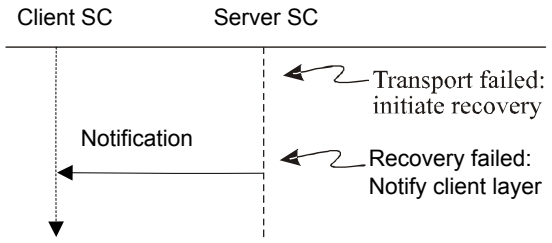


Figure 6-27 – Signal flow: Existing interlayer call → Transport network connection defect

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.

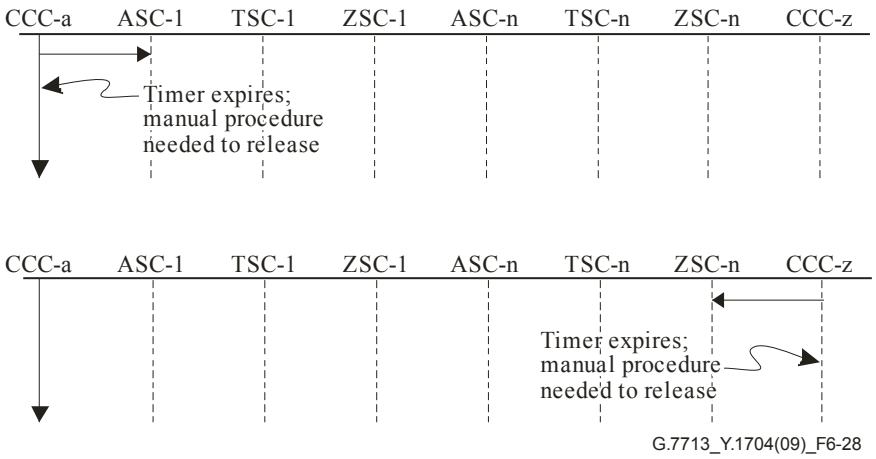


Figure 6-28 – Signal flow: Release call → User initiated release (user timer expires)

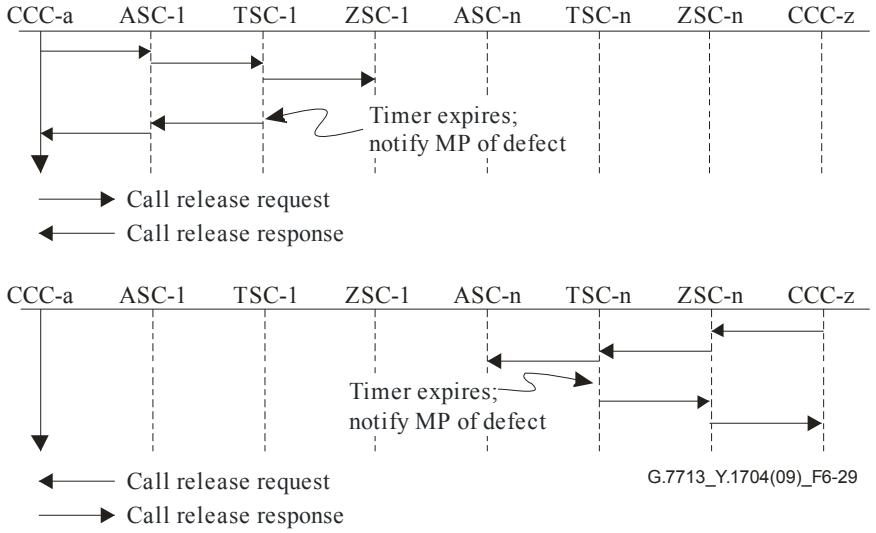


Figure 6-29 – Signal flow: Release call → User initiated release (network timer expires)

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 6-28). In the case where the network timer expires, the network sends a release confirmation response to the user (see Figure 6-29).

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 6-30 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 clause 6.3.3.1) the connections may still be up, while for this example the connection has been partially released.

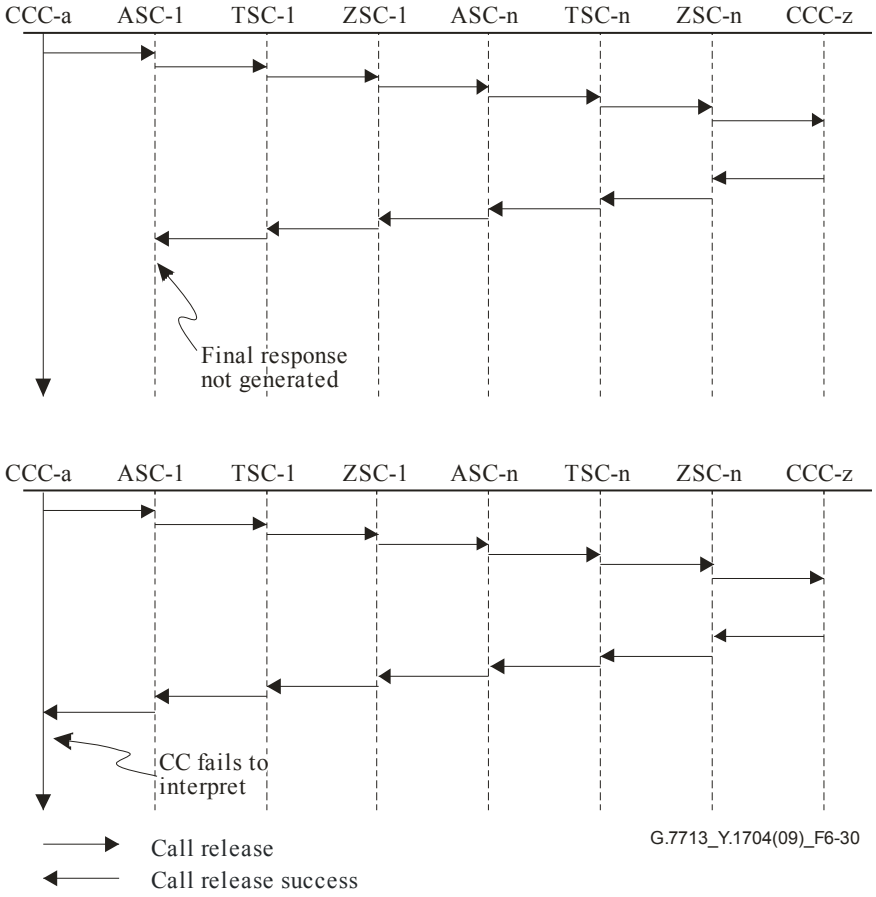


Figure 6-30 – Signal flow: Release → Signalling defect of the response message

6.3.4 Connection resource release

As specified in clause 7.3.5.5 of [ITU-T G.8080], if the network cannot set up 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 6-31 and 6-32 respectively.

In the scenarios described in Figures 6-31 and 6-32, 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 6-31, CCC-a is located at the user end (in an Access Group Container) and in Figure 6-32 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 6-31, 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.

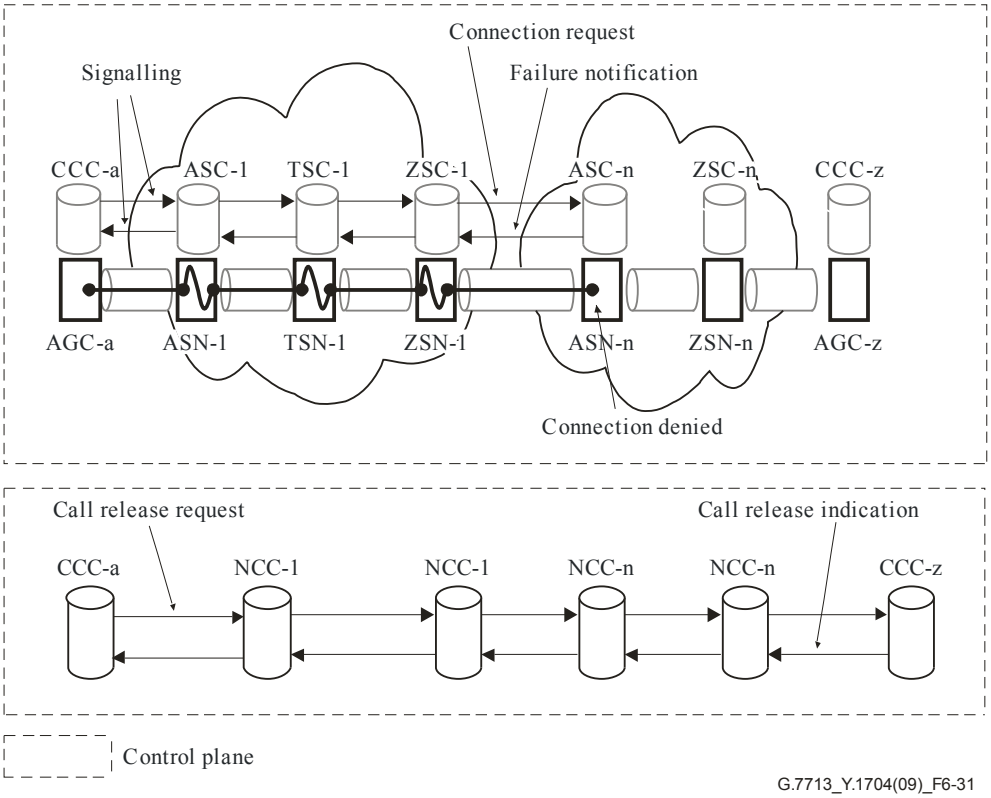


Figure 6-31 – Call release after connection set up failure (SC)

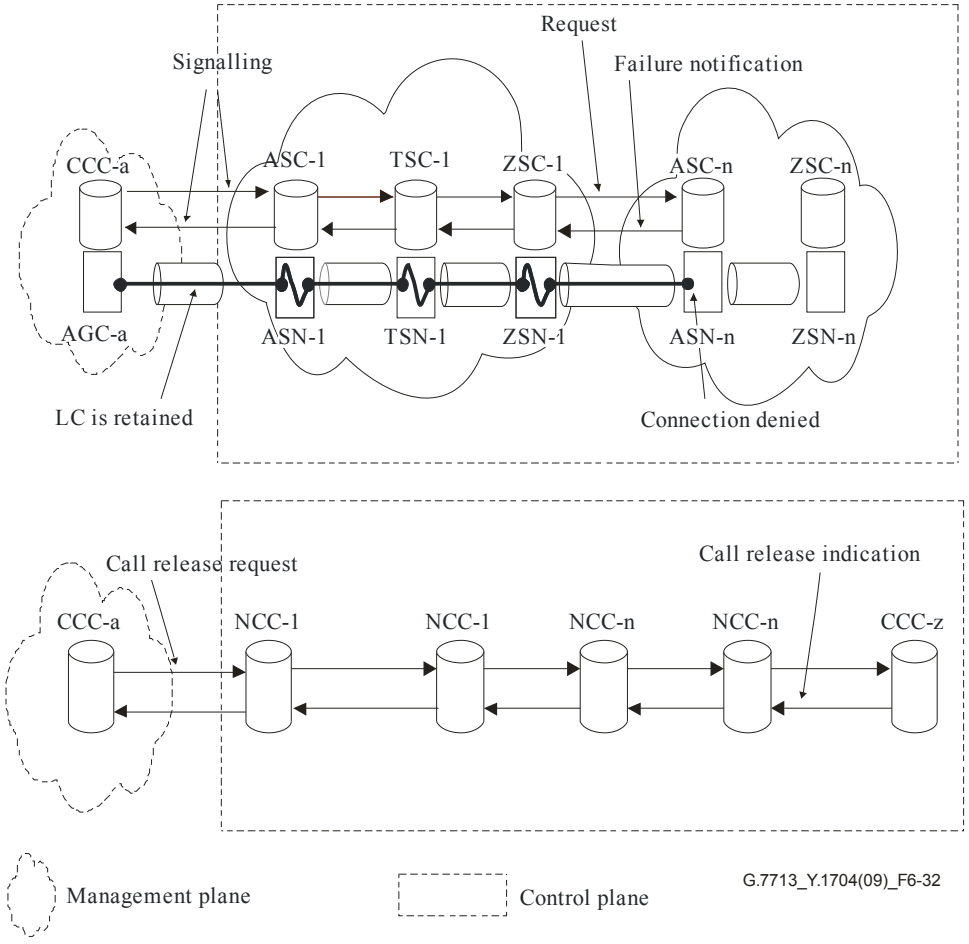


Figure 6-32 – Call release after connection set up failure (SPC)

6.3.5 Call modification

For the call modification specified in clause 7.3.5.4 of [ITU-T G.8080], if the network is unable to modify the connections, then the 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 6-33 and 6-34 respectively.

In the scenarios illustrated in Figures 6-33 and 6-34, 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.

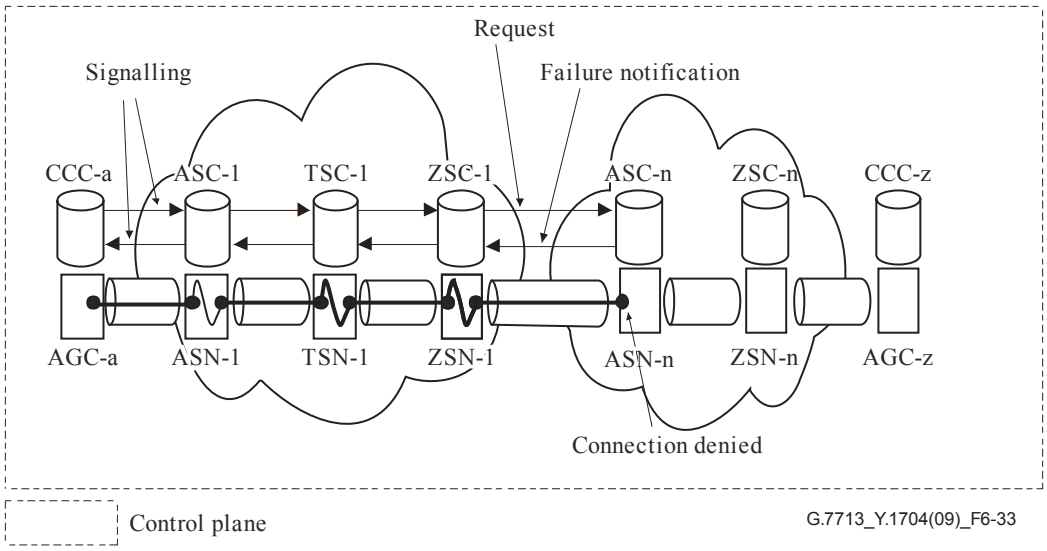


Figure 6-33 – Call state unchanged after connection set up failure (SC)

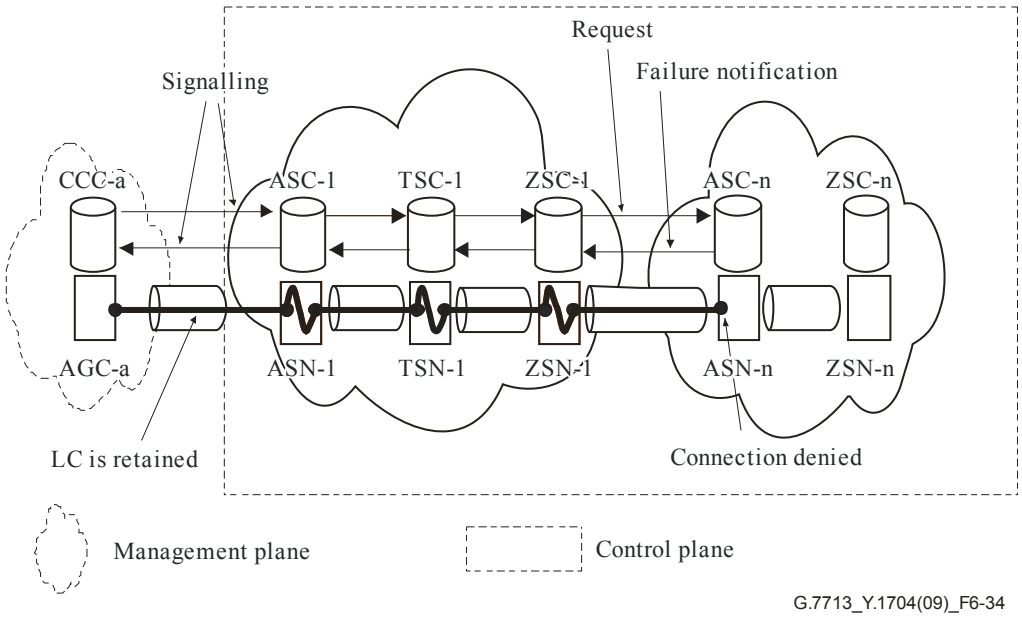


Figure 6-34 – Call state unchanged after connection set up failure (SPC)

6.3.6 Connection modification

There are two types of modification that can be applied to a connection: increases and decreases of the CO-PS data rate. They are covered separately below as exception handling differs.

6.3.6.1 Connection modification that increases CO-PS data rate

If a connection modification that increases the CO-PS data rate associated with a connection was unsuccessful, the call modification request will be rejected.

6.3.6.2 Connection modification that decreases CO-PS data rate

If a connection modification that decreases the CO-PS data rate associated with a connection was unsuccessful, a successful call modification indication is sent to the user and the MP is notified of the failure to decrease the CO-PS data rate of a connection. This is necessary to avoid billing the client for services it no longer requires. The source network call controller may reattempt to delete the connection or decrease the CO-PS data rate of the existing connection when requested by the MP or after a timeout period expires.

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 6-35. 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 6-35, it is shown as occurring after the new connection is established.

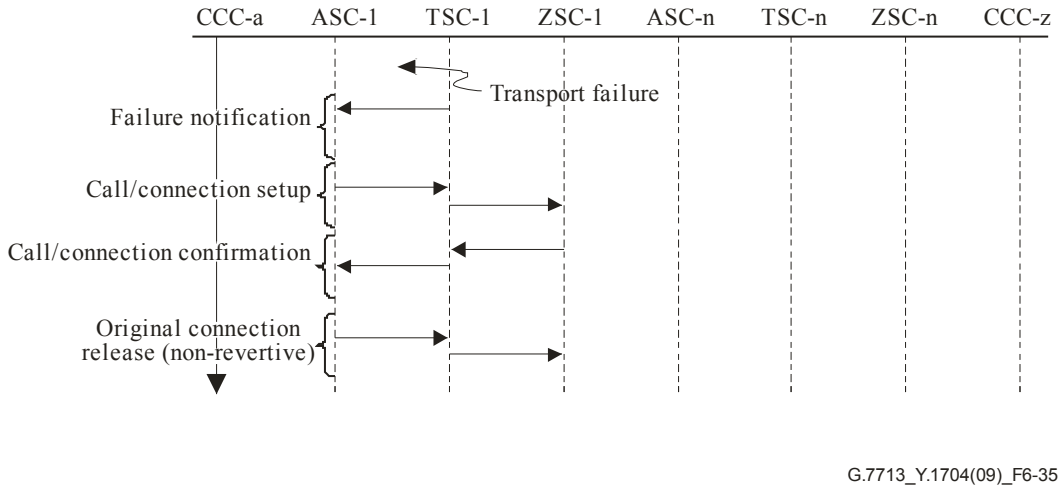


Figure 6-35 – Hard rerouting – Single domain

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 6-36 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.

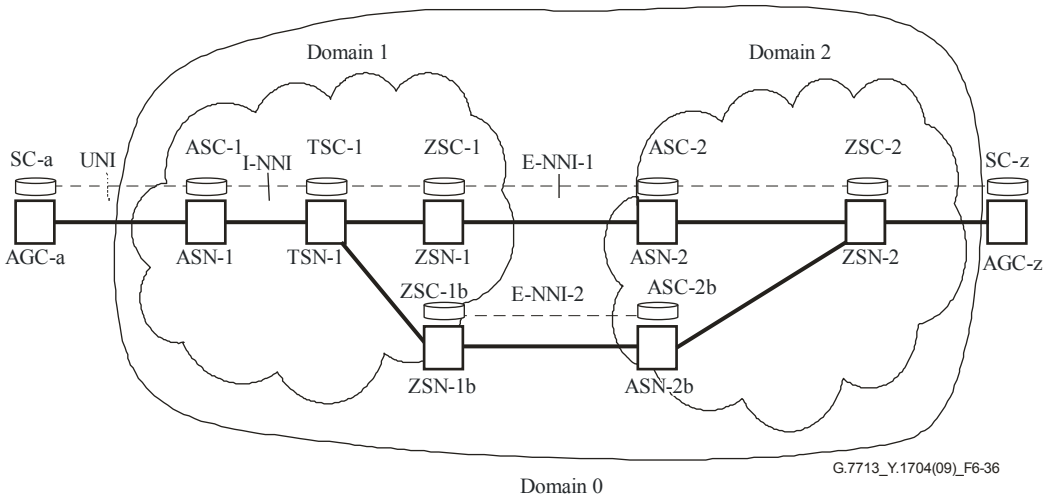


Figure 6-36 – Hard rerouting – Multiple E-NNIs

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 6-37.

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 6-37, it is shown as occurring after the new connection is established.

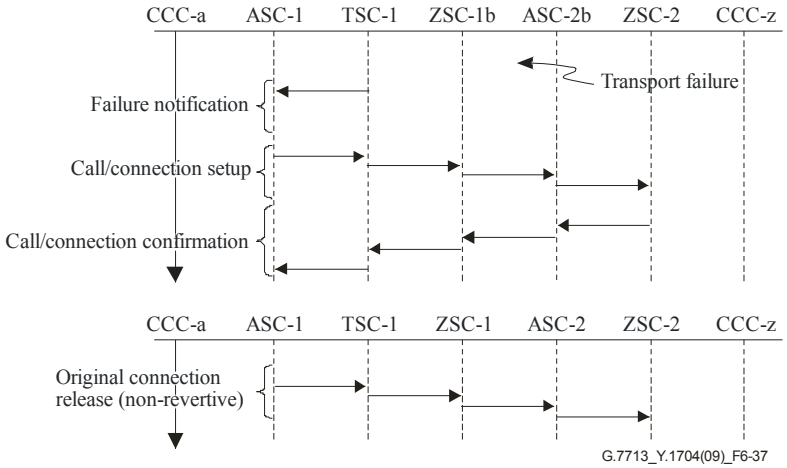


Figure 6-37 – Hard rerouting – Multiple domains

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 through 7-4 summarize a list of attributes that are considered for UNI, I-NNI, E-NNI and interlayer 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 – 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
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
	Group Size	End-to-end	Call
	CO-PS data rate	End-to-end	Call
	Non-disruptive bandwidth modification	End-to-end	Call
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 – 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
	Group size	End-to-end	Call
	CO-PS data rate	End-to-end	Call
	Non-disruptive bandwidth modification	End-to-end	Call
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 – 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

Table 7-3 – E-NNI attributes list

	Attributes	Scope	Call vs Connection
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
	Group Size	End-to-end	Call
	CO-PS data rate	End-to-end	Call
	Non-disruptive bandwidth modification	End-to-end	Call
Policy attributes	CoS	End-to-end	Call
	GoS	End-to-end	Call
	Security	Local	Call/Connection
	Explicit resource list	Local	Connection

Table 7-4 – Interlayer attributes list

	Attributes	Call/Connection
Identity attributes	Calling Server Layer Transport Resource Identifier	Call
	Called Server Layer Transport Resource Identifier	Call
	Server Layer CallC Name	Call
	Client Layer CallC Name	Call
	Server Layer Call Name	Call
	Client Layer Call Name	Call
Service attributes	Calling Server Layer SNP ID	Connection
	Calling Server Layer SNPP ID	Connection
	Called Server Layer SNP ID	Connection
	Called Server Layer SNPP ID	Connection
	Directionality	Call/Connection
	Group Size	Call
	CO-PS data rate	Call
	Non-disruptive bandwidth modification	Call
Policy attributes	CoS	Call
	GoS	Call
	Explicit resource list	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 G.8080] 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 G.8080] 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 G.7714]), 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 set up the LC on the calling AGC to network element access links.

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

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

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

– SNPP ID – This ID is used to set up 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.

7.1.2.4 Group size

The number of server layer connections that are multiplexed into the inverse multiplexing adaptation. This attribute is only applicable to inverse multiplexed layers.

7.1.2.5 CO-PS data rate

The rate associated with each call, e.g., bit/s. This attribute is only applicable to packet transport layers.

7.1.2.6 Non-disruptive bandwidth modification

This attribute is used to set hitless bandwidth modification.

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;

Call modify – success;

Call modify – failed: called party busy;
Call modify – failed: calling party busy;
Call modify – failed: network busy;
Call modify – failed: message error;
Call modify – failed: identity error: invalid call name;
Call modify – failed: identity error: invalid A-end user name;
Call modify – failed: identity error: invalid Z-end user name;
Call modify – failed: identity error: invalid connection name;
Call modify – failed: service error: invalid SNP ID;
Call modify – failed: service error: unavailable SNP ID;
Call modify – failed: service error: invalid SNPP ID;
Call modify – failed: service error: unavailable SNPP ID;
Call modify – failed: policy error: invalid CoS;
Call modify – failed: policy error: unavailable CoS;
Call modify – failed: policy error: invalid GoS;
Call modify – failed: policy error: unavailable GoS;
Call modify – 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 clause 7.1.1.1.

7.2.1.2 Called UNI transport resource identifier

See definition in clause 7.1.1.2.

7.2.1.3 Initiating CC/CallC name

See definition in clause 7.1.1.3.

7.2.1.4 Terminating CC/CallC name

See definition in clause 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 clause 7.1.1.6.

7.2.2 Service attributes

7.2.2.1 SNP ID

See definition in clause 7.1.2.1.

7.2.2.2 SNPP ID

See definition in clause 7.1.2.2.

7.2.2.3 Directionality

See definition in clause 7.1.2.3.

7.2.2.4 Group size

See definition in clause 7.1.2.4.

7.2.2.5 CO-PS data rate

See definition in clause 7.1.2.5.

7.2.2.6 Non-disruptive bandwidth modification

See definition in clause 7.1.2.6.

7.2.3 Policy attributes

7.2.3.1 CoS and connection CoS

See definition in clause 7.1.3.1.

7.2.3.2 GoS and connection GoS

See definition in clause 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;
Connection modify – success;
Connection modify – failed: called party busy;
Connection modify – failed: calling party busy;
Connection modify – failed: network busy;
Connection modify – failed: message error;
Connection modify – failed: identity error: invalid call name;
Connection modify – failed: identity error: invalid A-end user name;
Connection modify – failed: identity error: invalid Z-end user name;
Connection modify – failed: identity error: invalid connection name;
Connection modify – failed: service error: invalid SNP ID;
Connection modify – failed: service error: unavailable SNP ID;
Connection modify – failed: service error: invalid SNPP ID;
Connection modify – failed: service error: unavailable SNPP ID;
Connection modify – failed: policy error: invalid CoS;
Connection modify – failed: policy error: unavailable CoS;
Connection modify – failed: policy error: invalid GoS;
Connection modify – failed: policy error: unavailable GoS;
Connection modify – failed: policy error: failed security check.

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 clause 7.1.1.1.

7.3.1.2 Called UNI transport resource identifier

See definition in clause 7.1.1.2.

7.3.1.3 Initiating CC/CallC name

See definition in clause 7.1.1.3.

7.3.1.4 Terminating CC/CallC name

See definition in clause 7.1.1.4.

7.3.1.5 Connection name

See definition in clause 7.1.1.5.

7.3.1.6 Call name

See definition in clause 7.1.1.6.

7.3.2 Service attributes

7.3.2.1 SNP ID

See definition in clause 7.1.2.1.

7.3.2.2 SNPP ID

See definition in clause 7.1.2.2.

7.3.2.3 Directionality

See definition in clause 7.1.2.3.

7.3.2.4 Group size

See definition in clause 7.1.2.4.

7.3.2.5 CO-PS data rate

See definition in clause 7.1.2.5.

7.3.2.6 Non-disruptive bandwidth modification

See definition in clause 7.1.2.6.

7.3.3 Policy attributes

7.3.3.1 CoS

See definition in clause 7.1.3.1.

7.3.3.2 GoS

See definition in clause 7.1.3.2.

7.3.3.3 Security

See definition in clause 7.1.3.3.

7.3.3.4 Explicit resource list

See definition in clause 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;
Connection modify – success;
Connection modify – failed: called party busy;
Connection modify – failed: calling party busy;
Connection modify – failed: network busy;
Connection modify – failed: message error;
Connection modify – failed: identity error: invalid call name;
Connection modify – failed: identity error: invalid A-end user name;
Connection modify – failed: identity error: invalid Z-end user name;
Connection modify – failed: identity error: invalid connection name;
Connection modify – failed: service error: invalid SNP ID;

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

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

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

Connection modify – failed: policy error: invalid CoS;

Connection modify – failed: policy error: unavailable CoS;

Connection modify – failed: policy error: invalid GoS;

Connection modify – failed: policy error: unavailable GoS;

Connection modify – failed: policy error: failed security check.

7.3.4.2 Notification codes

Connection error – non-service affecting;

Connection error – service affecting;

Connection error – unexpected call release.

7.4 Interlayer attributes list

7.4.1 Identity attributes

7.4.1.1 Calling server layer transport resource identifier

This attribute is the [ITU-T G.8080] transport resource identifier used to identify SNPs that reference resources at the interlayer boundary at the A-end of the request.

7.4.1.2 Called server layer transport resource identifier

This attribute is the [ITU-T G.8080] transport resource identifier used to identify SNPs that reference resources at the interlayer boundary at the Z-end of the request.

7.4.1.3 Server layer CallC name

This attribute specifies the name associated with the server layer CallC across the interlayer signalling interface.

7.4.1.4 Client layer CallC name

This attribute specifies the name associated with the client layer CallC across the interlayer signalling interface.

7.4.1.5 Server layer call name

This attribute uniquely identifies a call in the server layer. It is assigned by the calling server layer call controller.

7.4.1.6 Client layer call name

This attribute uniquely identifies the client call requesting the server layer call(s). It is assigned by the A-end network call controller in the client layer.

7.4.2 Service attributes

7.4.2.1 Calling server layer SNP ID

This attribute represents the adaptation point used at the A-end adaptation to establish the adaptation between the client and server layers.

7.4.2.2 Calling server layer SNPP ID

This attribute represents the pool of adaptation points used to request a call across an interlayer boundary at the A-end adaptation.

7.4.2.3 Called server layer SNP ID

This attribute represents the adaptation point used at the Z-end adaptation to establish the adaptation between the client and server layers.

7.4.2.4 Called server layer SNPP ID

This attribute represents the pool of adaptation points used to request a call across an interlayer boundary at the Z-end adaptation.

7.4.2.5 Directionality

See definition in clause 7.1.2.3.

7.4.2.6 Group size

The number of server layer connections that are multiplexed into the inverse multiplexing adaptation. This attribute is only applicable when the server layer is an inverse multiplexed layer.

7.4.2.7 CO-PS data rate

The rate associated with each call, e.g., bits/s. This attribute is only applicable when the server layer is a packet transport layer.

7.4.2.8 Non-disruptive bandwidth modification

See definition in clause 7.1.2.6.

7.4.3 Policy attributes

7.4.3.1 CoS

This attribute specifies the CoS for the requested server layer call.

7.4.3.2 GoS

This attribute specifies the GoS for the requested server layer call.

7.4.3.3 Explicit resource list

This attribute specifies the explicit resources used in setting up the connection in the server layer. The ordered list may include zero or more instances of SNPP ID and/or SNP ID in the server layer.

7.4.4 Status codes

7.4.4.1 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;
Call modify – success;
Call modify – failed: called party busy;
Call modify – failed: calling party busy;
Call modify – failed: network busy;
Call modify – failed: message error;
Call modify – failed: identity error: invalid call name;
Call modify – failed: identity error: invalid A-end user name;
Call modify – failed: identity error: invalid Z-end user name;
Call modify – failed: identity error: invalid connection name;
Call modify – failed: service error: invalid SNP ID;
Call modify – failed: service error: unavailable SNP ID;
Call modify – failed: service error: invalid SNPP ID;
Call modify – failed: service error: unavailable SNPP ID;
Call modify – failed: policy error: invalid CoS;
Call modify – failed: policy error: unavailable CoS;
Call modify – failed: policy error: invalid GoS;
Call modify – failed: policy error: unavailable GoS;
Call modify – failed: policy error: failed security check.

7.4.4.2 Notification codes

Call error – non-service affecting;
Call error – service affecting.

8 DCM message sets

The DCM messages may be separated into messages associated with the call operations and messages associated with the connection operations. Table 8-1 summarizes a candidate list of messages that are considered for signalling processing. They represent the logical messages that are exchanged across the UNI, I-NNI, E-NNI and interlayer interfaces to support the CallC and CC. 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.

Table 8-1 – DCM message set

	UNI and E-NNI call messages	Interlayer call message	UNI, I-NNI and E-NNI connection messages
Set up messages	CallSetupRequest CallSetupIndication CallSetupConfirm	CallSetupRequest Client → Server CallSetupIndication Client ← Server	ConnectionSetupRequest ConnectionSetupIndication ConnectionSetupConfirm
Release messages	CallReleaseRequest CallReleaseIndication	CallReleaseRequest Client → Server CallReleaseIndication Client ← Server	ConnectionReleaseRequest ConnectionReleaseIndication
Query messages	CallQueryRequest CallQueryIndication	CallQueryRequest Client → Server CallQueryIndication Client ← Server	ConnectionQueryRequest ConnectionQueryIndication
Notification message	CallNotify	CallNotify Client ← Server	ConnectionNotify
Modification messages	CallModifyRequest CallModifyIndication CallModifyConfirm	CallModifyRequest Client → Server CallModifyIndication Client ← Server	ConnectionModifyRequest ConnectionModifyIndication ConnectionModifyConfirm

8.1 Call messages

8.1.1 Call setup

Across the UNI and E-NNI, 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.

Across the interlayer interface, a two-phase approach is defined for setting up a call. A call name is created by the A-end server layer network call controller and, combined with the client layer call name, is used for reference to the requested server layer call.

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 8-2.

Table 8-2 – Call setup request message

UNI and E-NNI attributes	Interlayer attributes
Calling UNI transport resource identifier	Calling Server Layer TRI
Called UNI transport resource identifier	Calling Server Layer TRI
Initiating CC/CCC name	
Terminating CC/CCC name	
Calling AGC SNP ID	[Calling Server Layer SNP ID]
Calling AGC SNPP ID	[Calling Server Layer SNPP ID]
Called AGC SNP ID	[Called Server Layer SNP ID]
Called AGC SNPP ID	[Called Server Layer SNPP ID]
Directionality	Directionality
CoS	CoS
GoS	GoS
Security	
connName	
callName – Not present in the UNI call request from the source user. Present in all other messages.	Client Layer Call Name
[Group Size]	[Group Size]
[CO-PS data rate]	[CO-PS data rate]
Security	

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 8-3.

Table 8-3 – Call setup indication message

UNI and E-NNI attributes	Interlayer attributes
connName	Client Layer Call Name
callName	Server Layer Call Name
Status	Status
Security	Calling Server Layer SNP ID
	Calling Server Layer SNPP ID
	Called Server Layer SNP ID
	Called Server Layer SNPP ID

8.1.1.3 Confirm: Setup call

An optional callSetupConfirm message is defined as a response to callSetupIndication. This message may be generated by either the A-end user (or third party) for SC calls or the network for SPC calls. Attributes sent for call setup confirm are shown in Table 8-4.

Table 8-4 – Call setup confirm message

Attributes
connName
callName
Status
Security

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 generated by either the user or the network. Attributes sent for call release request are shown in Table 8-5.

Table 8-5 – Call release request message

UNI and E-NNI attributes	Interlayer attributes
callName	Client layer call name
Security	Server layer call name
	Server layer SNP ID
	Server layer SNPP ID

8.1.2.2 Indication: Release call

A callReleaseIndication message is defined as a response to callReleaseRequest. This message may be generated by either the user or the network. Attributes sent for call release indication are shown in Table 8-6.

Table 8-6 – Call release indication message

UNI and E-NNI attributes	Interlayer attributes
callName	Client layer call name
Status	Server layer call name
Security	Server layer SNP ID
	Server layer SNPP ID
	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 generated by either the user or the network. Attributes sent for call query request are shown in Table 8-7.

Table 8-7 – Call query request message

UNI and E-NNI attributes	Interlayer attributes
callName	Client layer call name(s)
Security	Client layer call name(s)

8.1.3.2 Response: Query per call characteristics

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

Table 8-8 – Call query response message

UNI and E-NNI attributes	Interlayer attributes
callName	Server layer call name
CoS	Client layer call name
GoS	CoS
For each connName:	GoS
SNP ID	Server layer SNP ID
SNPP ID	Server layer SNPP ID
Status	Status
Security	

8.1.3.3 Request: Query all calls

A callQueryAllRequest message is defined to query all existing calls associated with a particular client call controller 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 8-9.

Table 8-9 – All call query request message

UNI and E-NNI attributes	Interlayer attributes
CCC/NCC name	Client layer call name
Security	

8.1.3.4 Response: Query all calls

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

Table 8-10 – All call query response message

UNI and E-NNI attributes	Interlayer attributes
CCC/NCC name	Client layer call name
List of callNames	List of server layer call names
Status	Status
Security	

8.1.4 Notification

A call notification message is specified to allow exchange of information related to the call status. Attributes sent for notification are shown in Table 8-11.

Table 8-11 – Call notification message

UNI and E-NNI attributes	Interlayer attributes
callName	Client layer call name
connName	Server layer call name
Error code	Error code
Security	

The notification message is one-phase.

8.1.5 Call modification

A two-phase (and optional third phase) approach is defined for modifying a call. The call is either modified by modifying the number of connections or the call is modified by modifying parameters of existing connections.

8.1.5.1 Request: Modify call

A callModifyRequest message is defined to modify the call. Attributes sent for call modify request are shown in Table 8-12.

Table 8-12 – Call modify request message

UNI and E-NNI attributes	Interlayer attributes
Calling UNI transport resource identifier	Calling Server layer TRI
Called UNI transport resource identifier	Called Server layer TRI
Initiating CC/CCC name	
Terminating CC/CCC name	
Calling AGC SNP ID	[Calling Server Layer SNP ID]
Calling AGC SNPP ID	[Calling Server Layer SNPP ID]
Called AGC SNP ID	[Called Server Layer SNP ID]
Called AGC SNPP ID	[Called Server Layer SNPP ID]
Directionality	Directionality
CoS	CoS
GoS	GoS
Security	

Table 8-12 – Call modify request message

UNI and E-NNI attributes	Interlayer attributes
connName – the connection(s) being added, removed or modified within the call	Server layer call name – the call modified by this request
callName	Client layer call name
[Group Size]	[Group Size]
[CO-PS data rate]	[CO-PS data rate]
Non-disruptive bandwidth modification	Non-disruptive bandwidth modification
Security	

8.1.5.2 Indication: Modify call

A callModifyIndication message is defined as a response to callModifyRequest. This message may be sent by either the Z-end user (or third party) or the network. Attributes sent for call modify indication are shown in Table 8-13.

Table 8-13 – Call modify indication message

UNI and E-NNI attributes	Interlayer attributes
connName	Client layer call name
callName	Server layer call name
Status	Server layer SNP ID
Security	Server layer SNPP ID
	Status

8.1.5.3 Confirm: Modify call

An optional callModifyConfirm message is defined as a response to callModifyIndication. This message may be sent by either the A-end user (or third party) or the network. Attributes sent for call modify confirm are shown in Table 8-14.

Table 8-14 – UNI call modify confirm message

Attributes
connName
callName
Status
Security

8.2 Connection 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 8-15.

Table 8-15 – 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
Local SNPP ID
Called AGC SNP ID
Called AGC SNPP ID
Directionality
Call CoS
Call GoS
Connection CoS
Connection GoS
Explicit resource list
Recovery
Security

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 8-16.

Table 8-16 – Connection setup indication message

Attributes
connName
callName
Status
Security

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 8-17.

Table 8-17 – Connection setup confirm message

Attributes
connName
callName
Status
Security

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 8-18.

Table 8-18 – Connection release request message

Attributes
callName
connName
Security

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 8-19.

Table 8-19 – Connection release indication message

Attributes
callName
connName
Status
Security

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 8-20.

Table 8-20 – Connection query request message

Attributes
callName
connName
Security

8.2.3.2 Response: Query per connection characteristics

A connQueryIndication message is defined as a response to connQueryRequest. Attributes sent for connection query response are shown in Table 8-21.

Table 8-21 – Connection query response message

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

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 8-22.

Table 8-22 – All connection query request message

Attributes
CC name
Security

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 8-23.

Table 8-23 – All connection query response message

Attributes
CC name
List of connName with their associated callName
Status
Security

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 connection notification are shown in Table 8-24.

Table 8-24 – Notification message

Attributes
callName
connName
Error code
Security

The notification message is one-phase.

8.2.5 Connection modify

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

8.2.5.1 Request: Modify connection

A connModifyRequest message is defined to modify a connection. Attributes sent for connection modify request are shown in Table 8-25.

Table 8-25 – Connection modify request message

Attributes
Calling UNI Transport Resource Identifier
Called UNI Transport Resource Identifier
Initiating CC name
Terminating CC name
connName
callName
Local SNP ID
Local SNPP ID
Called AGC SNP ID
Called AGC SNPP ID
Directionality
Call CoS
Call GoS
Connection CoS
Connection GoS
Explicit resource list
Recovery
Non-disruptive bandwidth modification
[Group Size]
[CO-PS data rate]
Security

8.2.5.2 Indication: Modify connection

A connModifyIndication message is defined as a response to connModifyRequest. Attributes sent for connection modify indication are shown in Table 8-26.

Table 8-26 – Connection modify indication message

Attributes
connName
callName
Status
Security

8.2.5.3 Confirm: Modify connection

An optional connModifyConfirm message is defined as a response to connModifyIndication. Attributes sent for connection modify confirm are shown in Table 8-27.

Table 8-27 – Connection modify confirm message

Attributes
connName
callName
Status
Security

9 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 set up/release, dynamic protection/restoration allocations/assignments and restoration.

– ASON management requirements are described in [ITU-T G.7718] including general protocol controller management.

The call controller, connection controller and link resource manager components defined in [ITU-T G.8080] provide for the capability to supervise and manage calls that are set up and released via distributed connection set up 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 9-1 illustrates an interface between these components and MP, where the MI information described concerns only the DCM function.

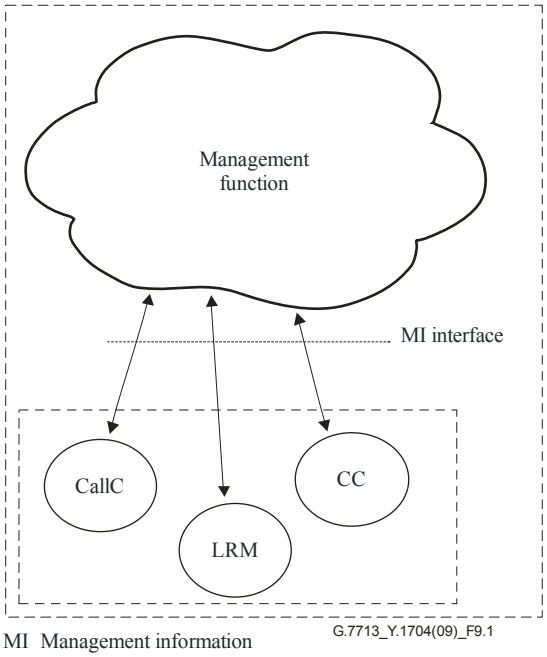


Figure 9-1 – Interface between control plane components and management plane

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 information reported 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 Appendix I.
  - MI_CCconnectionState provides information about the state of a particular connection based on the states defined in Appendix I.
  - MI_LRMconnectionState provides information about the state of a particular link connection based on the states defined in [ITU-T G.8080].
  - 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 G.8080] 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 set up and release of connections. Clauses 9.1 and 9.2 describe the processes related to set up and release, plus the management communications applicable at different stages of the connection operations.

Table 9-1 provides a summary of the candidate MI signals defined in this Recommendation.

Table 9-1 – 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
MI_CallCcallDuration	Time zone, start time and date, end time and date
MI_CallCcallState	As per states defined in Appendix I
MI_CCconnectionState	As per states defined in Appendix I
MI_LRMconnectionState	As per states defined in [ITU-T G.8080]

Table 9-1 – Candidate MI signals

MI signal	Attributes of signal
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_ConnectionSetupRequest	Attributes are the same as attributes for ConnectionSetupRequest message
MI_ConnectionSetupIndication	Attributes are the same as attributes for ConnectionSetupIndication message
MI_requestSetupNewRoute	callName connName A-end user name Z-end user name SNP ID SNPP ID Directionality CoS GoS [CO-PS data rate] [group size]
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 [CO-PS data rate] [group size]
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

Table 9-1 – Candidate MI signals

MI signal	Attributes of signal
MI_CallReleaseIndication	Attributes are the same as attributes for CallReleaseIndication message
MI_ConnectionReleaseRequest	Attributes are the same as attributes for ConnectionReleaseRequest message
MI_ConnectionReleaseIndication	Attributes are the same as attributes for ConnectionReleaseIndication message
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 is the same as specified in clause 7.2.4.1 for connection release
MI_CallModifyRequest	Attributes are the same as attributes for CallModifyRequest message
MI_CallModifyIndication	Attributes are the same as attributes for CallModifyIndication Message
MI_ConnectionModifyRequest	Attributes are the same as attributes for ConnectionModifyRequest message
MI_ConnectionModifyIndication	Attributes are the same as attributes for ConnectionModifyIndication message
MI_modifyError	Attributes are the same as specified in clause 7.2.4.1 for connection modification

9.1 Setting up a call and associated connections

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

1) Two types of connections are supported, as per [ITU-T G.8080]: switched connections and soft permanent connections (SPCs). In the case of SPCs, the call and connection set up request is initiated by the management plane. As such, a message is needed across the MI interface, MI_CallSetupRequest.

2) After receipt of the set up 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.

3) 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.

4) 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.

5) 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.

6) 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).

   a) 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).

7) 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 set up response is sent to the management plane. As such, a message is needed across the MI interface, MI_CallSetupIndication.

   a) As part of the MI_CallSetupIndication, if the connection was not set up, or errors occurred that resulted in partial set up, 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.

8) 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.

9.2 Releasing a call and all associated connections

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

1) Two types of connections are supported, as per [ITU-T G.8080]: switched connections and soft permanent connections (SPCs). In the case of SPC, the connection release request is initiated by the management plane. As such, a message is needed across the MI interface, MI_CallReleaseRequest.

2) 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.

3) 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.

4) 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).

a) 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.

5) 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.

a) 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.

9.3 Modifying a call

The following candidate list of MI signals provides interactions between the management function and the control plane function for modifying a call by adding or removing connections or modifying existing connections.

1) Two types of connections are supported, as per [ITU-T G.8080]: switched connections and soft permanent connections (SPCs). In the case of SPCs, the call modify request is initiated by the management plane. As such, a message is needed across the MI interface. The exact message used reflects the type of modification required: MI_ConnectionModifyRequest to modify the CO-PS data rate of an existing connection, MI_ConnectionSetupRequest to add a connection to an existing call or MI_ConnectionReleaseRequest to remove a connection from an existing call.

2) When adding connections to an existing call, the processes are as described in clause 9.1 steps 2 to 6, followed by the last step below:

a) 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_ConnectionSetupIndication.

– As part of the MI_ConnectionSetupIndication, 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.

3) When removing connections from an existing call, the processes are as described in clause 9.2, steps 2 to 4, followed by the last step below:

a) 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_ConnectionReleaseIndication.

– As part of the MI_ConnectionReleaseIndication, 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.

4) When modifying existing connections from an existing call, the processes are as follows:

a) After receipt of the modify 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.

b) Upon verification of the request, resources need to be allocated new CO-PS data rate parameters. An additional message is needed for communicating that resources have been allocated new CO-PS data rate, MI_responseReleaseResourcesAllocated.

c) 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 modify operation. This communication allows notification to the management plane of the control plane's inability to successfully modify an existing connection, MI_modifyError.

a) Upon determining the status of the connection modification, 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 connection modify response is sent to the management plane. As such a message is needed across the MI interface, MI_ConnectionModifyIndication.

– As part of the MI_ConnectionModifyIndication, if the connection was not modified, or errors occurred that resulted in partial increase or decrease, 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 modified but cannot be unmodified by the control plane.

Appendix I

DCM state diagrams

(This appendix does not form an integral part of this Recommendation)

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.

Note that this clause is limited to the basic call/connection set up/release, and does not cover call/connection modification and interlayer signalling procedures.

The state transitions are specified for:

– Initiating user: the user that initiated the operation. For set up 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 set up 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.

I.1 Call state

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

Table I.1 – 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.

Table I.1 – User CallC call state events

Event	Event description
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 I.2):

Table I.2 – 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.
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.

I.1.1 Initiating user CallC call state

I.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 I.3, Figure I.1).

Table I.3 – 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

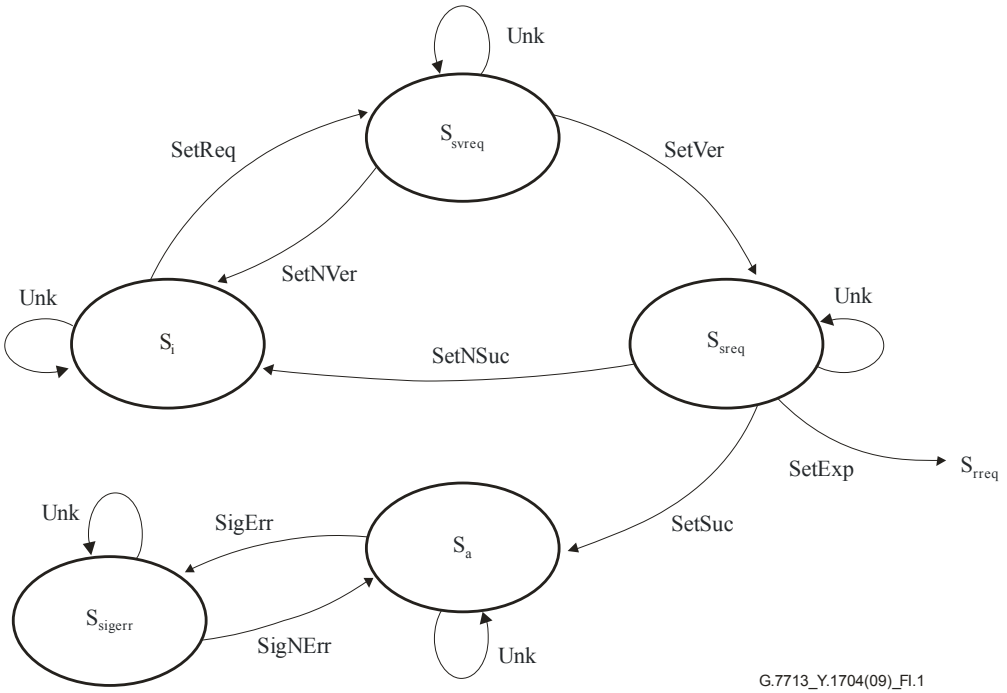


Figure I.1 – Initiating user CallC call setup state diagram

I.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 I.4, Figure I.2).

Table I.4 – 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

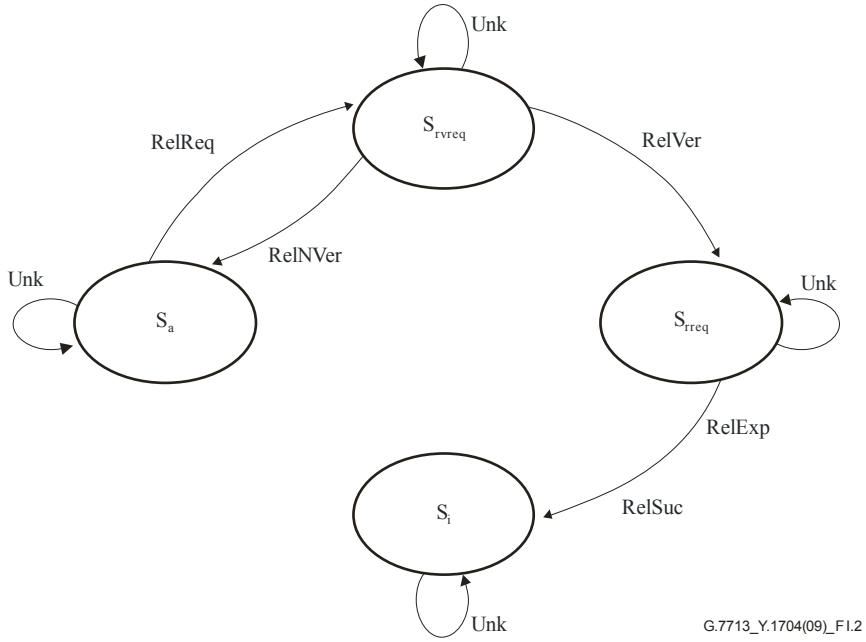


Figure I.2 – Initiating user CallC call release state diagram

I.1.2 Terminating user CallC call state

I.1.2.1 Terminating user call state: Setup

The state transitions shown in this clause apply to call set up for the terminating user CallC (Table I.5, Figure I.3).

Table I.5 – 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

Table I.5 – Terminating user CallC call setup state transitions

Current state	Event	Actions taken	Next state
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

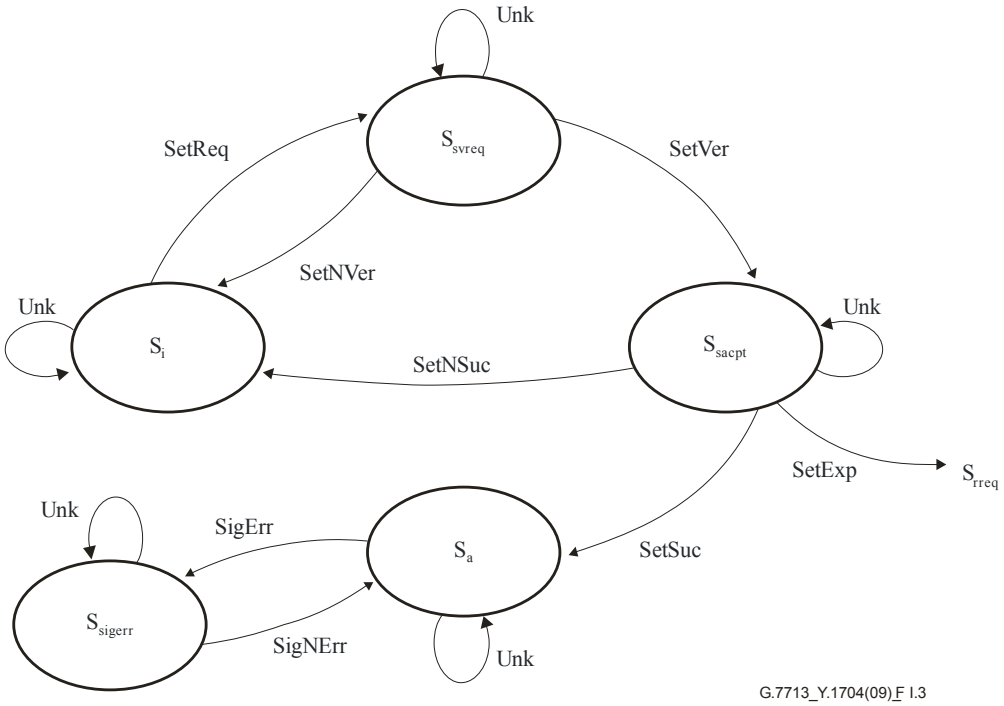


Figure I.3 – Terminating user CallC call setup state diagram

I.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 I.6, Figure I.4).

Table I.6 – 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

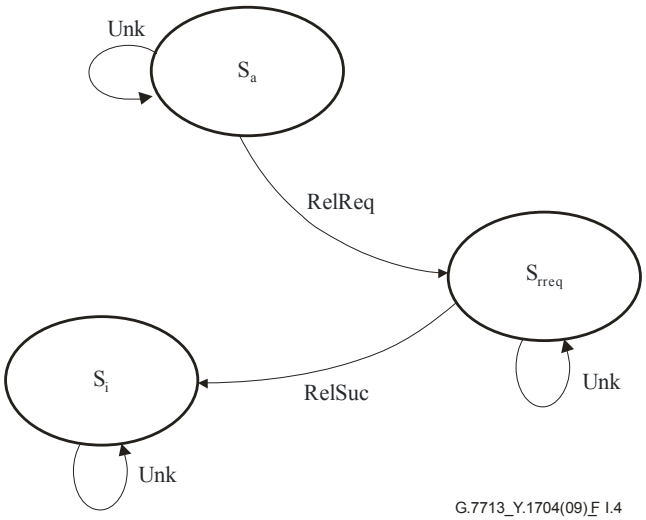


Figure I.4 – Terminating user CallC call release state diagram

I.1.3 Network CallC call state

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

I.1.3.1 Network CallC call state: Setup

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

Table I.7 – 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
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

Table I.7 – Network CallC call setup state transitions

Current state	Event	Actions taken	Next state
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

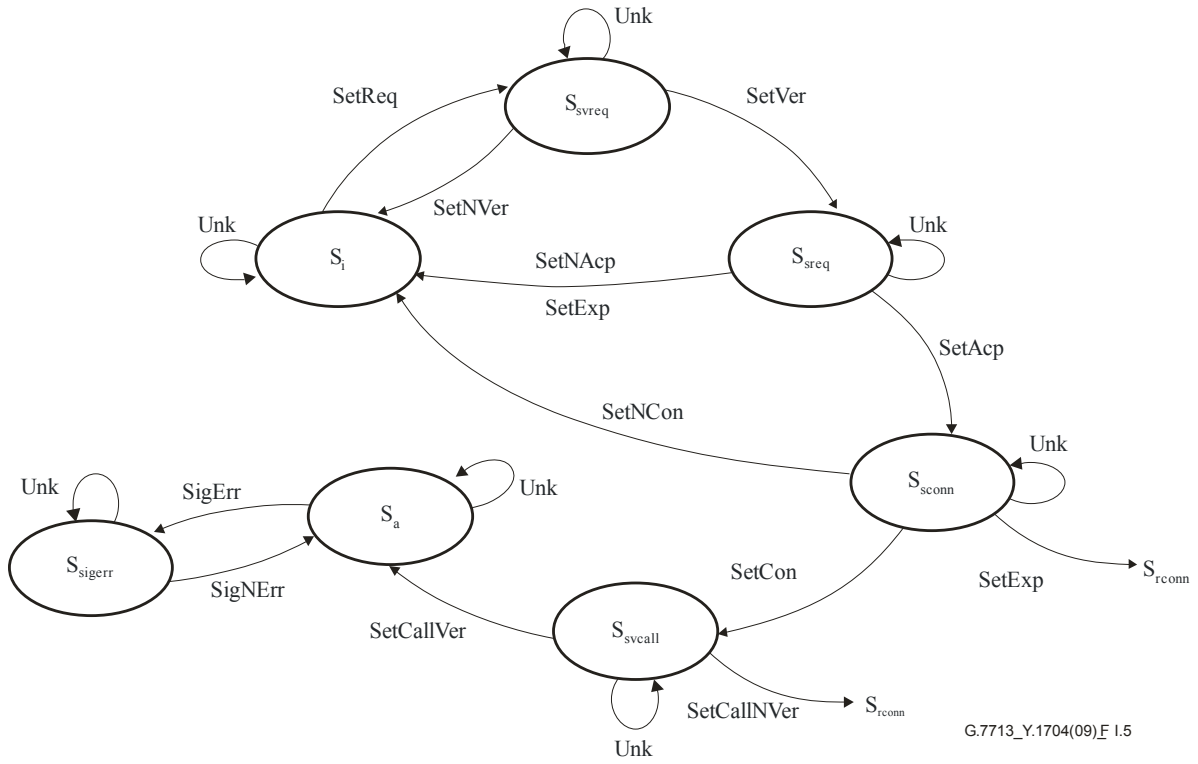


Figure I.5 – Network CallC call setup state diagram

I.1.3.2 Network CallC call state: Release

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

Table I.8 – 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}

Table I.8 – Network CallC call release state transitions

Current state	Event	Actions taken	Next state
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

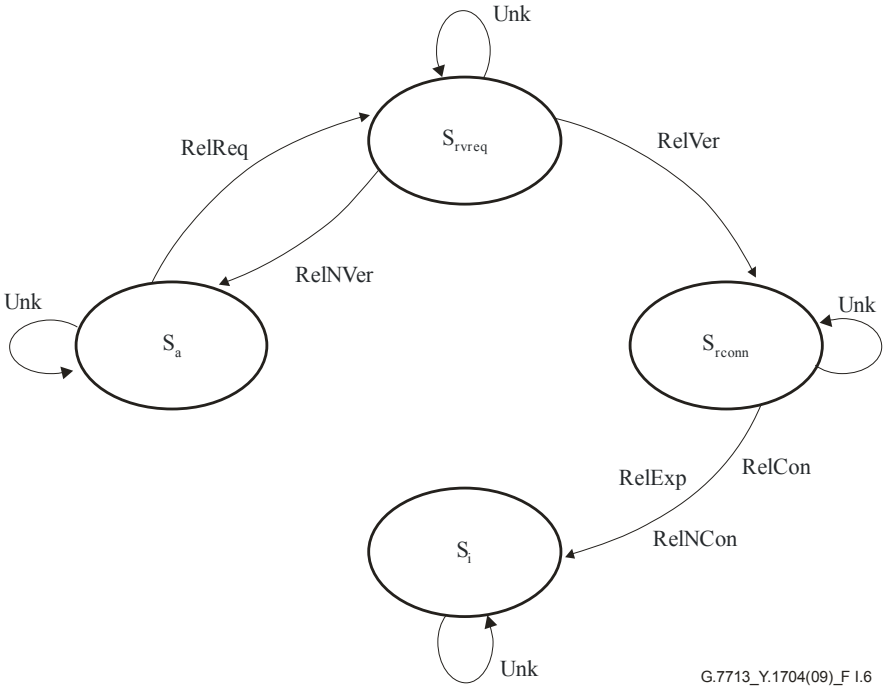


Figure I.6 – Network CallC call release state diagram

I.2 Connection state

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

Table I.9 – 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.

Table I.9 – User CC connection state events

Event	Event description
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 I.10):

Table I.10 – 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.
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.

I.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.

I.2.1.1 Initiating CC connection state: Setup

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

Table I.11 – 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

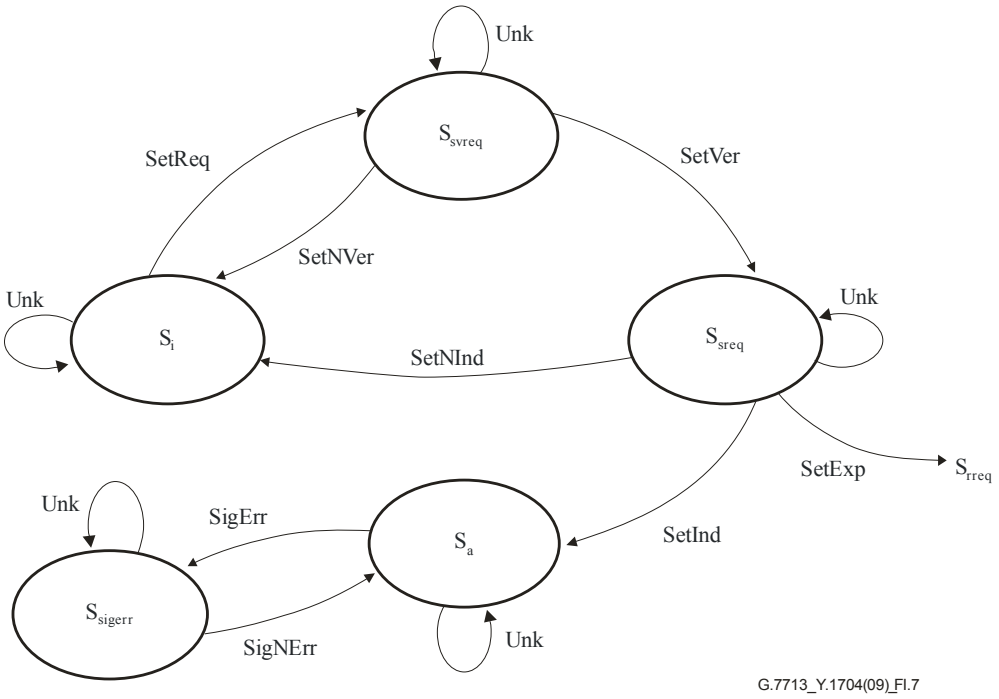


Figure I.7 – Initiating CC connection setup state diagram

I.2.1.2 Initiating CC connection state: Release

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

Table I.12 – 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

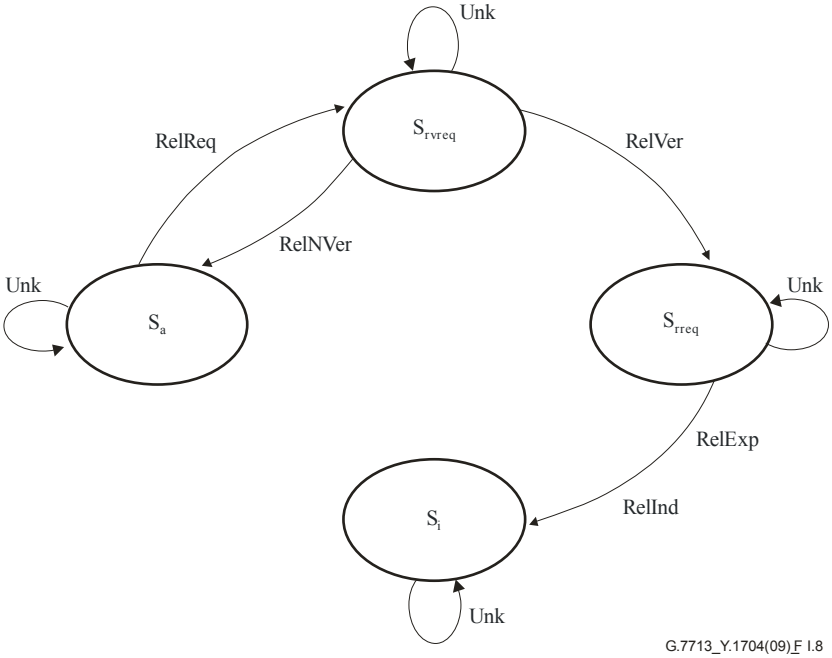


Figure I.8 – Initiating CC connection release state diagram

I.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.

I.2.2.1 Terminating CC connection state: Setup

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

Table I.13 – 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

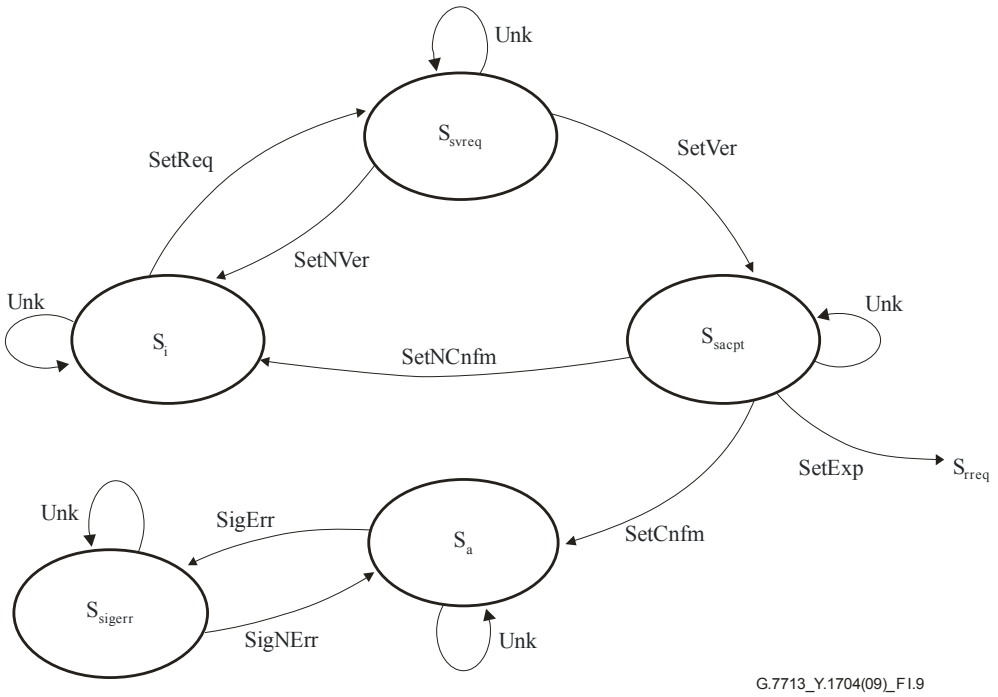


Figure I.9 – Terminating CC connection setup state diagram

I.2.2.2 Terminating CC connection state: Release

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

Table I.14 – 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

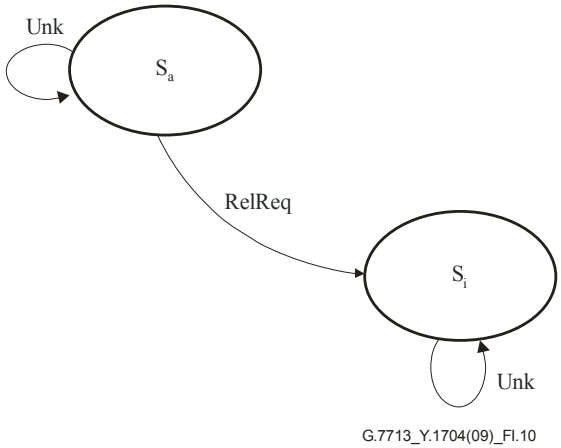


Figure I.10 – Terminating CC connection release state diagram

I.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.

I.2.3.1 Intermediate CC connection state: Setup

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

Table I.15 – 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

Table I.15 – Intermediate CC connection setup state transitions

Current state	Event	Actions taken	Next state
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

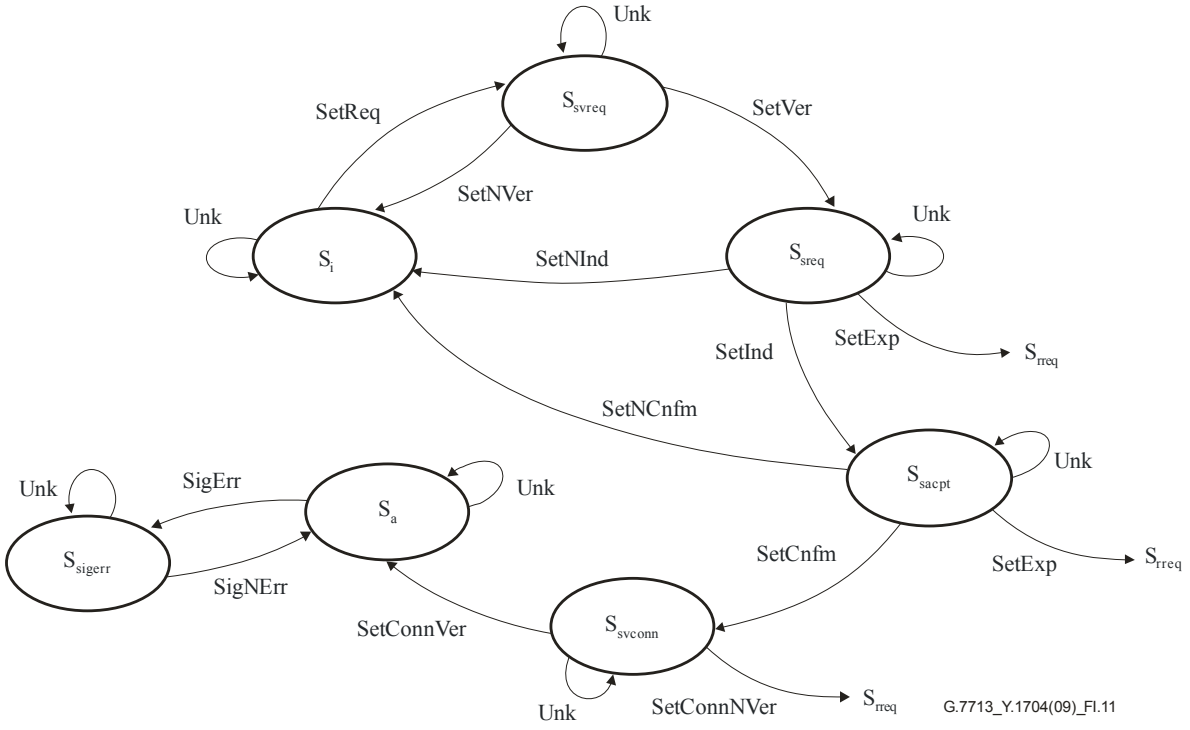


Figure I.11 – Intermediate CC connection setup state diagram

I.2.3.2 Intermediate CC connection state: Release

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

Table I.16 – 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

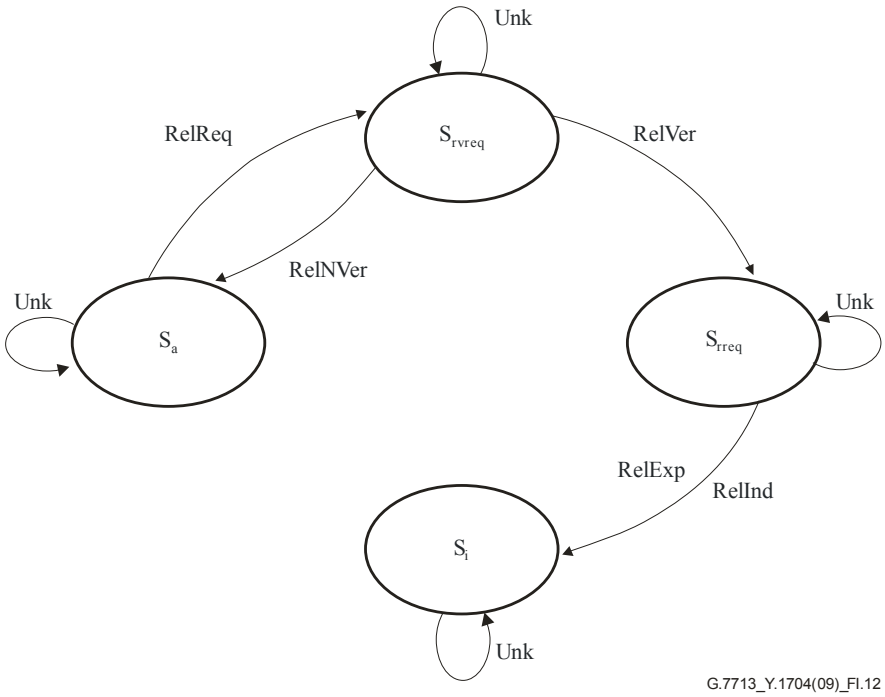


Figure I.12 – Intermediate CC connection release state diagram

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
IPTV over NGN	Y.1900–Y.1999
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
Future networks	Y.2600–Y.2699
Security	Y.2700–Y.2799
Generalized mobility	Y.2800–Y.2899
Carrier grade open environment	Y.2900–Y.2999

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	Terminals and subjective and objective assessment methods
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, 2010