INTERNATIONAL TELECOMMUNICATION UNION

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

G.7713/Y.1704

(12/2001)

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

Digital terminal equipments – Operations, administration and maintenance features of transmission equipment

SERIES Y: GLOBAL INFORMATION INFRASTRUCTURE AND INTERNET PROTOCOL ASPECTS

Internet protocol aspects – Operation, administration and maintenance

Distributed call and connection management (DCM)

ITU-T Recommendation 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
TESTING EQUIPMENTS	G.500–G.599
TRANSMISSION MEDIA CHARACTERISTICS	G.600–G.699
DIGITAL TERMINAL EQUIPMENTS	G.700–G.799
DIGITAL NETWORKS	G.800–G.899
DIGITAL SECTIONS AND DIGITAL LINE SYSTEM	G.900–G.999
QUALITY OF SERVICE AND PERFORMANCE	G.1000–G.1999
TRANSMISSION MEDIA CHARACTERISCTICS	G.6000–G.6999
DIGITAL TERMINAL EQUIPMENTS	G.7000–G.7999
General	G.7000–G.7099
Coding of analogue signals by pulse code modulation	G.7100–G.7199
Coding of analogue signals by methods other than PCM	G.7200–G.7299
Principal characteristics of primary multiplex equipment	G.7300–G.7399
Principal characteristics of second order multiplex equipment	G.7400–G.7499
Principal characteristics of higher order multiplex equipment	G.7500–G.7599
Principal characteristics of transcoder and digital multiplication equipment	G.7600–G.7699
Operations, administration and maintenance features of transmission equipement	**G.7700–G.7799**
Principal characteristics of multiplexing equipment for the synchronous digital hierarchy	G.7800–G.7899
Other terminal equipment	G.7900–G.7999

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

ITU-T Recommendation G.7713/Y.1704

Distributed call and connection management (DCM)

Summary

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

– Attribute specifications;

– Message specifications;

– Signal flows;

– DCM state diagrams;

– Management of DCM.

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

Source

ITU-T Recommendation G.7713/Y.1704 was prepared by ITU-T Study Group 15 (2001-2004) and approved under the WTSA Resolution 1 procedure on 5 December 2001.

This Recommendation forms part of a suite of Recommendations covering the full functionality of the automatic switched transport network (ASTN).

History

Version	Notes
1.0	Version 1 of G.7713/Y.1704

Keywords

Automatic Switched Optical Network, Automatic Switched Transport Network, Connection Operations, Distributed Call and Connection Management, Exterior Network-Node Interface, Interior Network-Node Interface, User Network Interface.

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications. 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.

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, implementors are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database.

CONTENTS

Page

ITU-T Recommendation G.7713/Y.1704

Distributed call and connection management (DCM)

1 Scope

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

- Attribute specifications;
- Message specifications;
- Signal flows;
- DCM state diagrams;
- Management of DCM.

Other areas of ASTN such as routing mechanism, parameters associated with routing mechanisms, discovery, and naming and addressing are not in 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 Recs G.807/Y.1301 and G.8080/Y.1304 as the basis for the specification. This Recommendations aims to provide a protocol neutral approach to describe the capability sets of the DCM. Capabilities specified in this Recommendation include support for soft permanent connections and switched connections.

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

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

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

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent addition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published.

- ITU-T Recommendation G.707/Y.1322 (2000), *Network node interface for the synchronous digital hierarchy (SDH).*
- ITU-T Recommendation G.709/Y.1331 (2001), *Interfaces for the optical transport network.*
- ITU-T Recommendation G.783 (2000), *Characteristics of Synchronous Digital Hierarchy (SDH) equipment functional blocks.*

- ITU-T Recommendation G.784 (1999), *Synchronous Digital Hierarchy (SDH) management.*
- ITU-T Recommendation G.798 (2002), *Characteristics of optical transport network hierarchy equipment functional blocks*.
- ITU-T Recommendation G.803 (2000), *Architecture of transport networks based on the synchronous digital hierarchy*.
- ITU-T Recommendation G.805 (2000), *Generic functional architecture of transport networks*.
- ITU-T Recommendation G.806 (2000), *Characteristics of transport equipment – Description methodology and generic functionality*.
- ITU-T Recommendation G.807/Y.1302 (2001), *Requirements for the automatic switched transport network (ASTN)*.
- ITU-T Recommendation G.851.1 (1996), *Management of the transport network – Application of the RM-ODP framework.*
- ITU-T Recommendation G.852.2 (1999), *Enterprise viewpoint description of transport network resource model.*
- ITU-T Recommendation G.853.1 (1999), *Common elements of the information viewpoint for the management of a transport network.*
- ITU-T Recommendation G.872 (2001), *Architecture of optical transport networks*.
- ITU-T Recommendation G.874 (2001), *Management aspects of the optical transport network element.*
- ITU-T Draft Recommendation G.875, *Optical transport network (OTN) management information model for the network element view.*
- ITU-T Recommendation G.7712/Y.1703 (2001), *Architecture and specification of data communication network.*
- ITU-T Recommendation G.7714/Y.1705 (2001), *Generalized automatic discovery techniques*.
- ITU-T Recommendation G.8080/Y.1304 (2001), *Architecture of the automatic switched optical network (ASON).*
- ITU-T Recommendation M.3100 (1995), *Generic network information model.*

3 Terms and Definitions

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

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

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

- Management Information (MI) signal

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

- Agent

- Component
- Subnetwork controller
- Call controller
- Connection controller
- Connection admission control
- Route controller
- Neighbour discovery
- Link resource manager
- Policy
- Protocol controller
- Subnetwork point
- Subnetwork point pool

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

- Data communication network

The following terms are defined in ITU-T Rec. G.807/Y.1301:

- Soft permanent connection
- Switched connection

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

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

4 Abbreviations

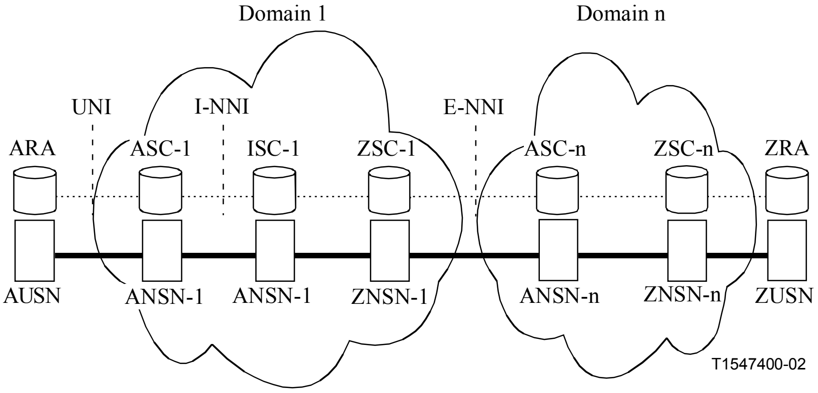
This Recommendation uses the following abbreviations:

AD	Administrative Domain
ANSN-n	A-end Network SubNetwork-domain n
ARA	A-end Requester Agent
ASC-n	A-end Subnetwork Controller-domain n
ASON	Automatic Switched Optical Network
ASTN	Automatic Switched Transport Network
AUSN	A-end User SubNetwork
CAC	Call Admission Control
CallC	Call Controller
CC	Connection Controller
CI	Characteristic Information

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
I-NNI	Internal NNI
INSN-n	Intermediate Network SubNetwork-domain n
ISC-n	Intermediate SubNetwork Controller-domain n
LRM	Link Resource Manager
MI	Management Information
ND	Neighbour Discovery
NNI	Network Node Interface
PC	Protocol Controller
PNNI	Private Network-to-Network Interface
RA	Requester Agent
RC	Route Controller
RSVP-TE	Resource Reservation Protocol-Traffic Engineering
SC	Switched Connection
SN	SubNetwork
SNC	SubNetwork Connection
SNCr	SubNetwork Controller
SNP	SubNetwork Point
SNPP	SubNetwork Point Pool
SPC	Soft Permanent Connection
TCP	Termination Connection Point
TTP	Trail Termination Point
UNI	User Network Interface
ZNSN-n	Z-end Network SubNetwork-domain n
ZRA	Z-end Requester Agent
ZSC-n	Z-end SubNetwork Controller-domain n
ZUSN	Z-end User SubNetwork

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.



ARA	A-end requester agent (A-end user CallC)
AUSN	A-end user subnetwork
ZRA	Z-end requester agent (Z-end user CallC)
ZUSN	Z-end user subnetwork
ASC-1	A-end subnetwork controller - domain 1 (CC)
ANSN-1	A-end network subnetwork - domain 1
ISC-1	Intermediate subnetwork controller - domain 1 (CC)
INSN-1	Intermediate network subnetwork - domain 1
ZSC-1	Z-end subnetwork controller - domain 1 (CC)
ZNSN-1	Z-end network subnetwork - domain 1
ASC-n	A-end subnetwork controller - domain n (CC)
ANSN-n	A-end network subnetwork - domain n
ZSC-n	Z-end subnetwork controller - domain n (CC)
ZNSN-n	Z-end network subnetwork - domain n

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

NOTE – This Recommendation uses "domain" synonymously with administrative domain, and uses requester agent synonymously with user Call Controller.

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 a LC or creating a SNC.
- De-allocate: de-allocate refers to either freeing a LC or releasing a 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 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.

As part of enabling automatic connection setup within the ASON network, the performance of the ASON network to set up call requests are based on different parameters. Characteristics that impact the signalling performance may include:

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

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

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 set up or release a connection:

- Route Controller (RC): Route controller provides route information as queried by the CC;
- Call Admission Control (CAC);
- Call Controller (CallC);
- Connection Controller (CC);
- Link Resource Manager (LRM).

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

The end-to-end establishment of a call involves requesting the call, requesting connections, and setup of different types of resources to create a connection. Figure 6-1 illustrates the connection that is set up to support a call.

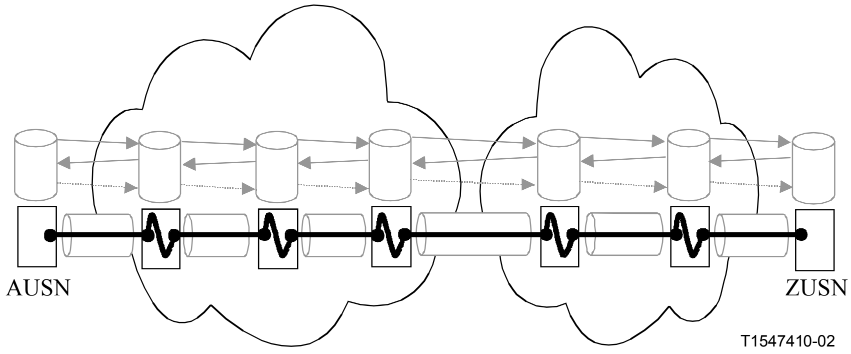


Figure 6-1/G.7713/Y.1704 – LC and SNC Establishment for a Call Request

The following resources are used for call setup:

– Subnetwork point (SNP);

– SNP Pool (SNPP);

– Link connection (LC).

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

Figure 6-2 illustrates the establishment of LC by the LRMs.

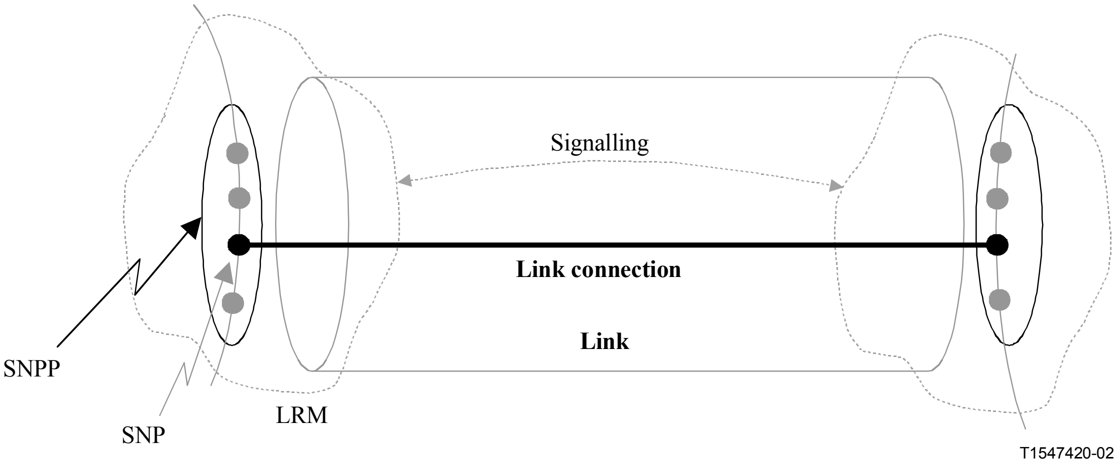


Figure 6-2G.7713/Y.1704 – Establishment of Link Connection by allocating the SNPs

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

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

Setting up of a SNC is a process that occurs internal to a subnetwork, and is controlled by the CC. A 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 6-3 depicts a SNC that has been set up along with the associated connection controller and the SNP pair that are involved with setting up a SNC.

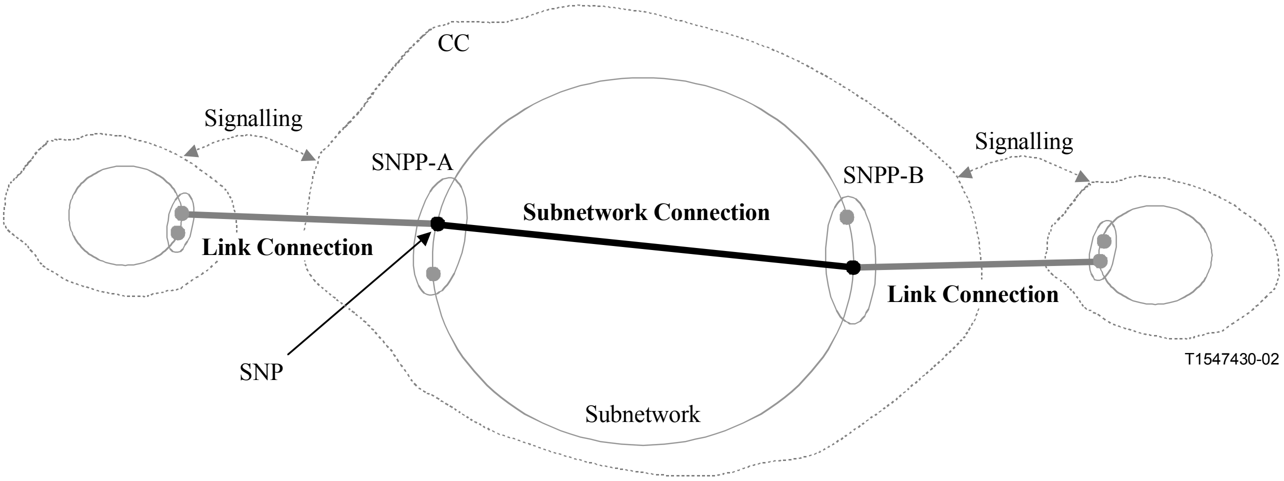


Figure 6-3/G.7713/Y.1704 – Creation of Subnetwork Connection after Link Connection Establishments

6.1.1 Process for Call Request

To support switched connection (SC) service, a call request is initiated by the A-end user request agent (ARA) via a "call setup request" message sent by the ARA (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 call process is handled by the management plane, with management requesting the CC to establish connections in support of the call.

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

6.1.1.1 Setting Up a Call

Figure 6-4 illustrates the setup of a call, and the associated signal flows between the relevant components.

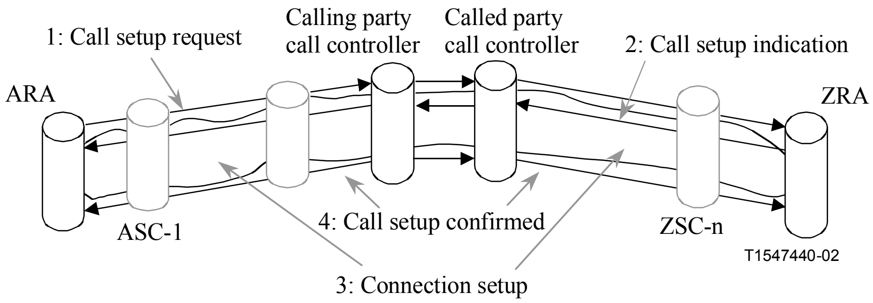


Figure 6-4/G.7713/Y.1704 – Call Setup Request Processing: Logical Request Progression

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

- The ARA requests call setup. At the calling party CallC, 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 called party CallC. Processes included in the called party CallC may include verifying that the call request is accepted end-to-end (e.g. request for ZRA call verification).
- Upon successful checking, the calling party CallC continues the call setup request by initiating a connection setup request to the CC. The process for connection setup request is described in 6.1.2. Note that, based on different protocol design decisions, initiation of connection setup request may occur in different orders as shown in Figure 6-4. The requirement is that a connection is set up before a call is completed.
- Upon successful indication by the connection setup request process, the call setup request is successfully completed, and transfer of user characteristic information may begin.

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

6.1.1.2 Releasing a Call

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

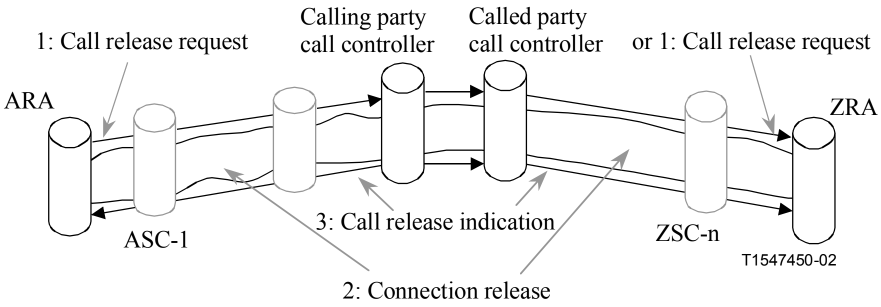

1: Call release request
Calling party call controller
Called party call controller
or 1: Call release request
ARA
ZRA
ASC-1
3: Call release indication
ZSC-n
T1547450-02
2: Connection release

Figure 6-5/G.7713/Y.1704 – Call Release Request Processing: Logical Request Progression

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

– Check the call release request. This may include checking for authentication and integrity of the request, as well as constraints placed by policy decisions.

– Upon successful checking, the call release request continues by initiating a connection release request. The process for connection release request is described in 6.1.2. Note that based on different protocol design decisions, initiation of connection release request may occur in different orders as shown in Figure 6-5. The requirement is that a connection is released before a call is released.

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

While connection setup may be denied and thus result in call denial, a connection release that is denied (e.g. due to inability to de-allocate resources, release a SNC or free a 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 is 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 ARA to ZRA, and the transport signal (e.g. unequipped or OCI) progression from AUSN to ZUSN, 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 suppression may be needed to prevent triggering the protection/restoration process.

6.1.2 Process for Connection Request

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

Figure 6-6 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.

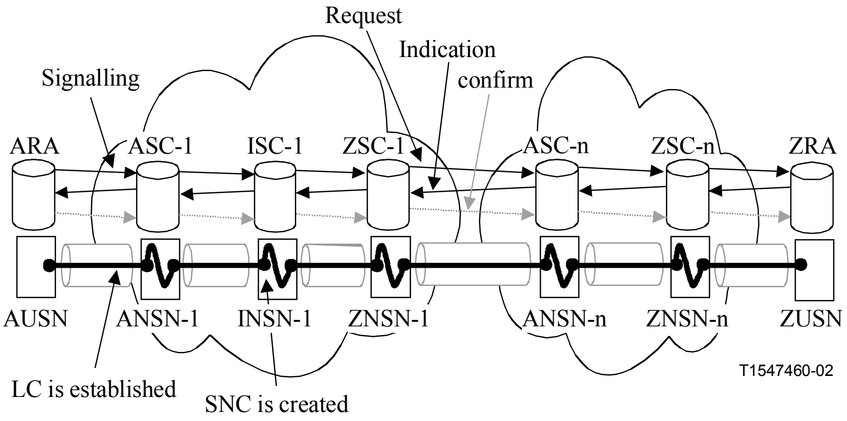


Figure 6-6/G.7713/Y.1704 – LC Establishment and SNC Creation for a Connection Setup Request

Figure 6-7 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).

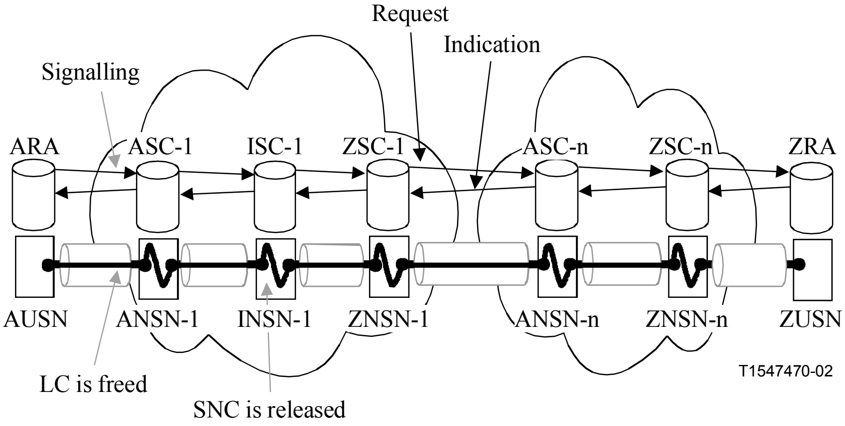


Figure 6-7G.7713/Y.1704 – Freeing LC and Releasing SNC for a Connection Release Request

6.1.2.1 Process for Setting up Connections

The following processes are performed for the connection setup:

- According to 6.1.1.1, a setup call request has been verified and allowed to proceed.
- From the call request, ARA LRM specifies the LC negotiated to be established between AUSN and ANSN-1. This may be in the form of AUSN'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 AUSN's egress SNP to ANSN-1's ingress SNP). ASC-1 LRM negotiates with ISC-1 LRM to establish a LC connecting ANSN-1 to INSN-1 (INSN-1 is determined by a route information as provided by a RC or by received information from upstream CC). Upon successful establishment of this LC, an egress SNP is identified by the LRM. A SNC is created that connects the ingress SNP to the egress SNP. The state of the SNP pair is updated to: provisioned. The CC then continues the connection setup process by communicating with the downstream CC.
- At ISC-1 CC, an ingress SNP is identified as determined by the ingress SNP ID. ISC-1 LRM negotiates with ZSC-1 LRM to establish a LC connecting INSN-1 to ZNSN-1 (ZNSN-1 is determined by a route information as provided by a RC or by received information from upstream CC). Upon successful establishment of this LC, an egress SNP is identified by the LRM. A SNC is created that connects the ingress SNP to the egress SNP. The state of the SNP pair is updated to: provisioned. The CC then continues the connection setup process by communicating with the downstream CC.
- This process is continued until the connection request reaches ZRA.
- At ZRA 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 the connection is processed.
- Optionally once ARA 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 initial determined route cannot be set up (for any reason), then the subnetwork that experienced the block may respond to the initiating subnetwork with status regarding the block (plus any additional information that identifies the cause of the block). Any reserved or allocated resources will be unreserved or de-allocated. This "crankback" process may be performed to allow the initiating subnetwork additional information that may be used for subsequent requests to avoid the "blocked" subnetwork.

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

6.1.2.2 Process for Releasing Connections

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

- According to 6.1.1.2, 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 ISC-1 LRM to free the LC used by INSN-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 ISC-1 CC, the SNC is released. ISC-1 LRM signals ZSC-1 LRM to free the LC used by ZNSN-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 request reaches ZRA CC.
- At ZRA CC, the LC used for the call is freed. After the CC processes the connection request, a response message is sent to indicate the connection is processed.

6.2 Signalling Network Resilience

The signalling network supports reliable transport for transferring signalling information between call and connection controllers. A signalling link is a connection in the signalling network that provides the delivery of messages between signalling entities. The following behaviour applies to a signalling link defect:

- After detection of a defect on a signalling link, the signalling network attempts to recover from the defect. This may be by means of redundancy of the signalling link. During this recovery period, no messages are accepted or processed (messages can be processed for those signalling links that are not impacted, only the messages that need to reach the call or connection controller connected by the impacted link are not processed). Existing calls remain up.
- If the defect is not recoverable (i.e. CallC or CC cannot communicate with other CallC or CC) then a notification is sent to the management system; however, existing calls remain up. Release of these impacted calls will be provided via other means, e.g. via manual processes.
- After recovery succeeds, the CallC or CC may need to resynchronize the status of existing calls/connections.

NOTE – If both the signalling network (as described in ITU-T Rec. G.7712/Y.1703) and the signalling channel supports recovery, a defect of the signalling network may result in both the DCN and the control plane trying to recover from the defect. Coordination may be needed between these potential recovery schemes to avoid both responding to a single defect.

In addition, defects of various components (e.g. defects or misbehaviours of CallC, CC, LRM) may result in defects of the control plane. As such, some of these defects may not be recoverable;

however, a notification is sent to the management plane that provides information regarding the defect.

6.2.1 User signalling defect

This signalling behaviour arises due to the following cause codes:

- Signalling channel disrupted between ARA and ASC-1;
- Signalling channel disrupted between ZSC-n and ZRA.

Note that other cause codes that currently have not been considered, may also contribute to a defect. When additional cause codes have been identified which can be considered to contribute to the defect, the cause codes will be added.

Figure 6-8 illustrates the ARA – ASC-1 defect. The behaviour due to a ZSC-n – ZRA defect results in similar actions, i.e. ZRA or ZSC-n initiates recovery of the signalling channel.

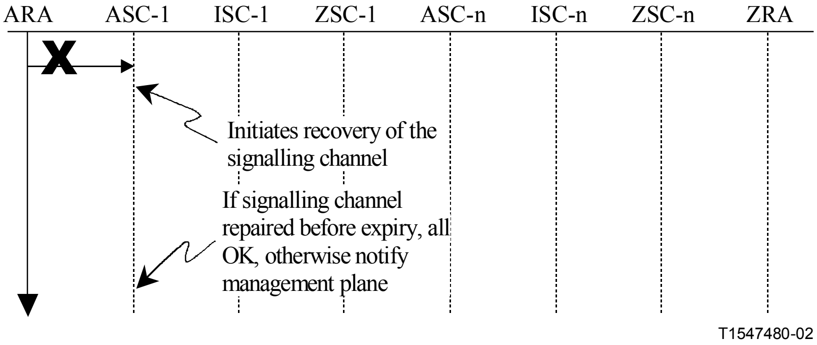

ARA
ASC-1
ISC-1
ZSC-1
ASC-n
ISC-n
ZSC-n
ZRA
X
Initiates recovery of the signalling channel
If signalling channel repaired before expiry, all OK, otherwise notify management plane
T1547480-02

Figure 6-8/G.7713/Y.1704 – UNI Signalling defect

In this example, a defect is detected for the signalling connection from ARA to ASC-1. Since ARA has no way of signalling call release request, alternate means need to be specified. Existing calls are not impacted.

6.2.2 Network signalling defect

This signalling behaviour arises due to the following cause codes:

- Signalling channel disrupted between CCs within a domain;
- Signalling channel disrupted between CCs across a domain.

Note that other cause codes, that currently have not been considered, may also contribute to a defect. When additional cause codes have been identified which can be considered to contribute to the defect, the cause codes will be added.

Figure 6-9 illustrates the ASC-1 – ISC-1 defect.

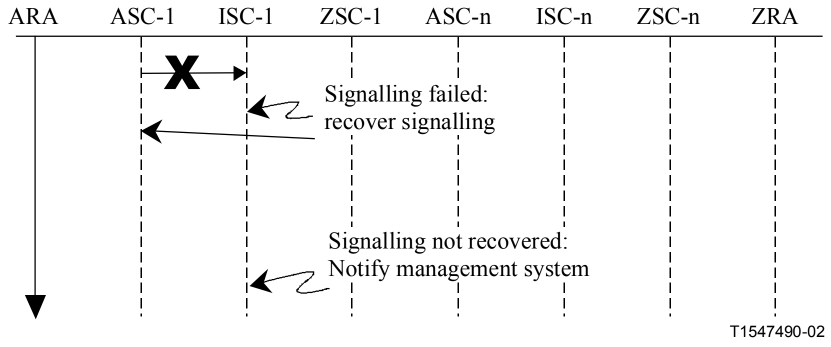


Figure 6-9/G.7713/Y.1704 – Network signalling defect

A signalling channel defect does not result in releasing a call. The network attempts to recover the signalling channel. Upon recovery of the signalling channel, the control plane continues servicing new requests. However, if the signalling channel is unrecoverable, then the CC notifies the management plane regarding the defect.

6.3 DCM Signal Flow – Exception Handling

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

- Defects of the signalling communication network may result due to disruption of the communications channel;
- Defects of the connection controller may result due to failure of different agents that make up the controller, e.g. failure of the connection setup agent;
- Misbehaviour of the connection controller may be a result of incorrect decoding of messages. Mechanisms for detection of a misbehaved component are outside the scope of this Recommendation.

Defect information is communicated by the CallC and CC to the management plane, plus any information specific to the detected defects. Figure 5-1 provides the reference diagram for call requests handled by the CallC. Using this network model, the following subclauses provide different scenarios, as shown from Figure 6-10 to Figure 6-24, the signalling flow due to setup and release of calls, and the various defects that may occur during these operations.

6.3.1 Setup Connection

6.3.1.1 ARA UNI defect (request message)

This signalling behaviour arises due to the following cause codes:

- Unrecognised signalling information;
- Defect of the RC (e.g. to determine a route to reach ZUSN);
- Defect of the CAC (e.g. to verify policy information);
- Link connection defect between AUSN – ANSN-1;
- Subnetwork connection defect within ANSN-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 ARA.

Note that other cause codes that currently have not been considered may also contribute to a defect. When additional cause codes have been identified, which can be considered to contribute to the defect, the cause codes will be added.

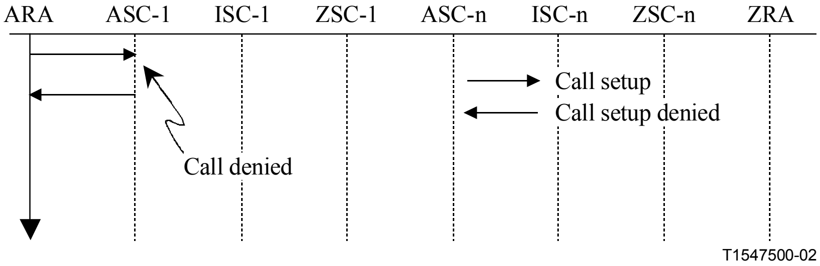


Figure 6-10/G.7713/Y.1704 – Setup → ARA UNI defect (Call denied)

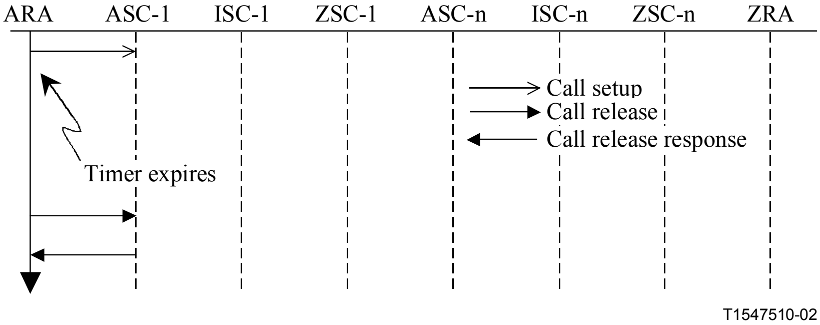


Figure 6-11/G.7713/Y.1704 – Setup → ARA UNI defect (Timer expires)

Figures 6-10 and 6-11 show two examples of a setup rejection. In Figure 6-10, different defects occurred at ASC-1 that resulted in denied call setup.

The second example (Figure 6-11) shows a timer expiring before the user receives a response. In this case, the user drops the request. To clean up any states, and to prevent the network from setting up the call at some later time (e.g. due to error of request synchronization) an explicit release request is made to release the previous setup request.

6.3.1.2 ARA 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 6-12 to 6-15 illustrate the signal flows based on response to the defects. In the first case, from the ARA 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.

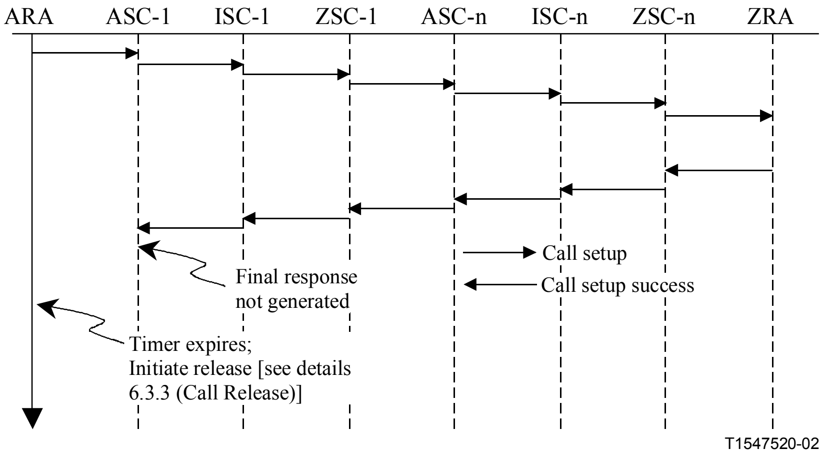


Figure 6-12/G.7713/Y.1704 – Setup → Signalling defect of the response message (final response not generated)

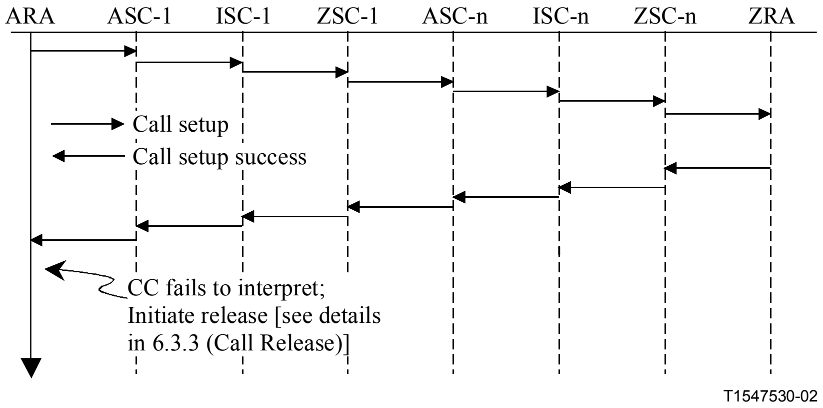


Figure 6-13/G.7713/Y.1704 – Setup → Signalling defect of the response message (ARA fails to interpret)

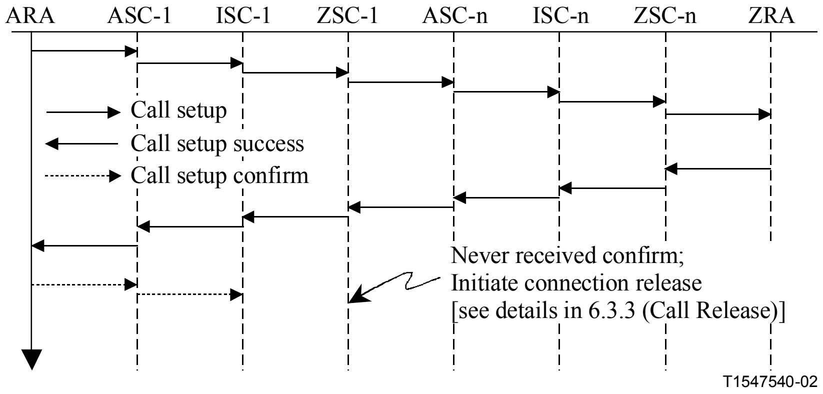


Figure 6-14/G.7713/Y.1704 – Setup → Signalling defect of the confirmation message (Confirmation message lost during transmission)

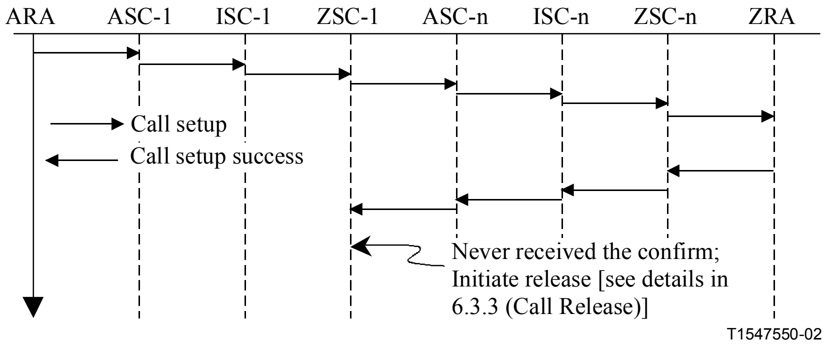


Figure 6-15/G.7713/Y.1704 – Setup → Signalling defect of the confirmation message (Confirmation message never generated)

6.3.1.3 Intra-domain and Inter-domain defects

This signalling behaviour arises due to the following cause codes:

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

Note that other cause codes that currently have not been considered may also contribute to a defect. When additional cause codes have been identified which can be considered to contribute to the defect, the cause codes will be added.

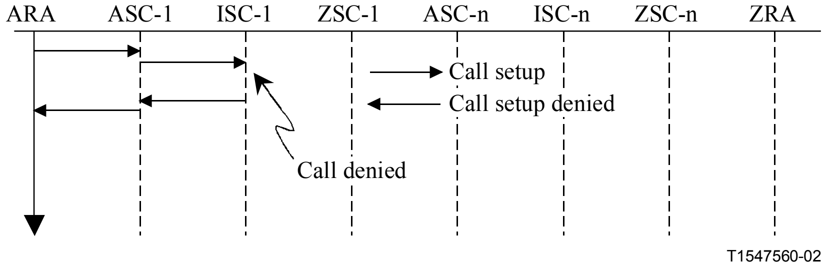


Figure 6-16/G.7713/Y.1704 – Signal flow: Setup → Intra-domain defect (Call denied)

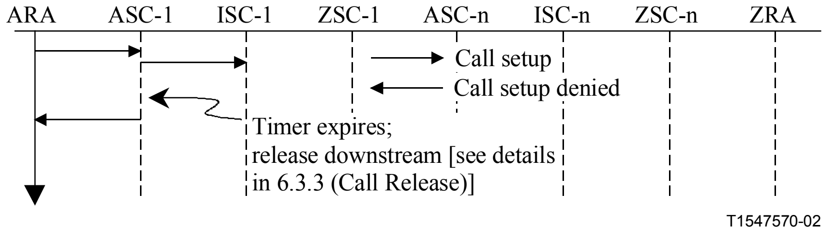


Figure 6-17/G.7713/Y.1704 – Signal flow: Setup → Intra-domain defect (Timer expires)

In this example, policy-based verification has succeeded; however, insufficient network resource 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 unreserve or de-allocate the connections.

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

6.3.1.4 ZRA 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. ZNSN-n – ZUSN;
- Subnetwork connection defect in a subnetwork, e.g. ZNSN-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 ZRA.

Note that other cause codes, that currently have not been considered, may also contribute to a defect. When additional cause codes have been identified which can be considered to contribute to the defect, the cause codes will be added.

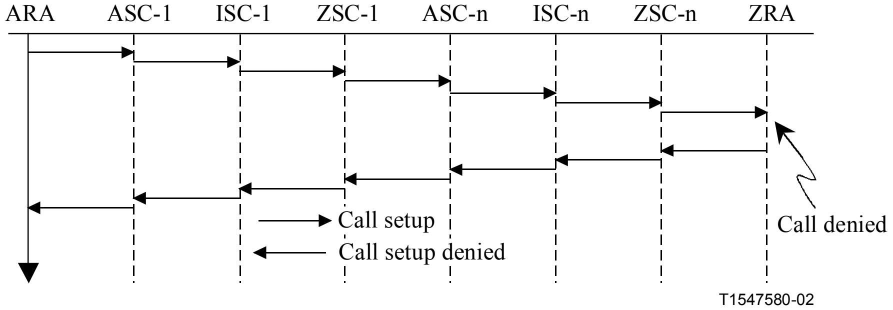


Figure 6-18/G.7713/Y.1704 – Signal flow: Setup → ZRA UNI defect (Call denied)

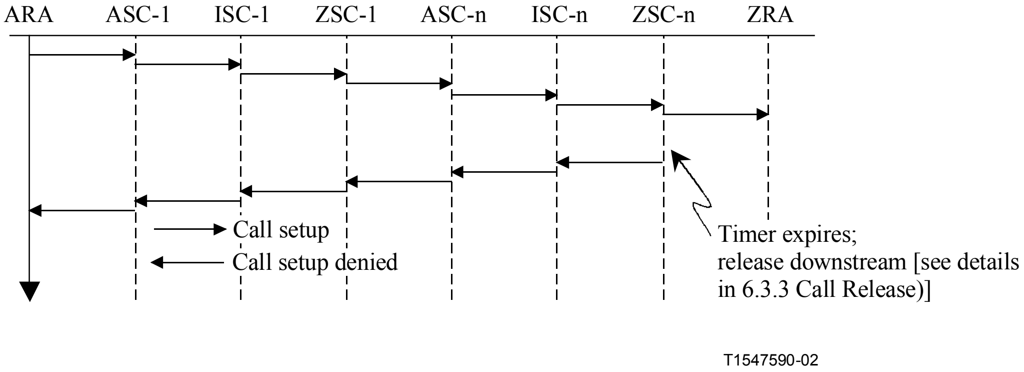


Figure 6-19/G.7713/Y.1704 – Signal flow: Setup → ZRA UNI defect (Timer expires)

In this defect scenario, the ZRA responds with denial of request. This may be caused by

1) failure to verify ARA's permission for connecting AUSN with ZUSN (e.g. during call setup),

2) ZUSN has no resource (e.g. during connection setup).

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

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

6.3.2 Existing Calls

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

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

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

– For a call setup using switched connection, the call is released (via release of the respective connections).

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.

Note that other cause codes that currently have not been considered may also contribute to a defect. When additional cause codes have been identified, which can be considered to contribute to the defect, the cause codes will be added.

This scenario 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).

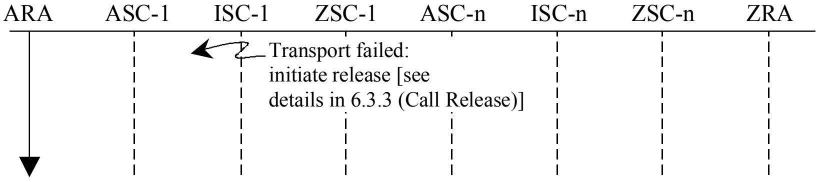

ARA
ASC-1
ISC-1
ZSC-1
ASC-n
ISC-n
ZSC-n
ZRA
Transport failed: initiate release [see details in 6.3.3 (Call Release)]
T1547600-02

Figure 6-20/G.7713/Y.1704 – Signal Flow: Existing call → user transport network connection defect

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

6.3.2.2 Transport network connection defect (Soft permanent connection)

This signalling behaviour arises due to the following cause codes:

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

Note that other cause codes that currently have not been considered may also contribute to a defect. When additional cause codes have been identified, which can be considered to contribute to the defect, the cause codes will be added.

This scenario 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.

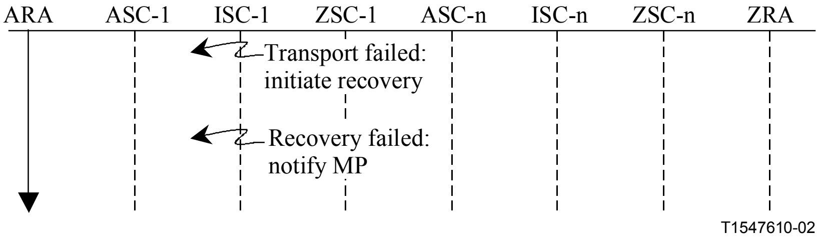

ARA
ASC-1
ISC-1
ZSC-1
ASC-n
ISC-n
ZSC-n
ZRA
Transport failed: initiate recovery
Recovery failed: notify MP
T1547610-02

Figure 6-21/G.7713/Y.1704. Signal flow: Existing call → transport network connection defect

6.3.3 Call Release

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

6.3.3.1 ARA or ZRA 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.

Note that other cause codes that currently have not been considered may also contribute to a defect. When additional cause codes have been identified which can be considered to contribute to the defect, the cause codes will be added.

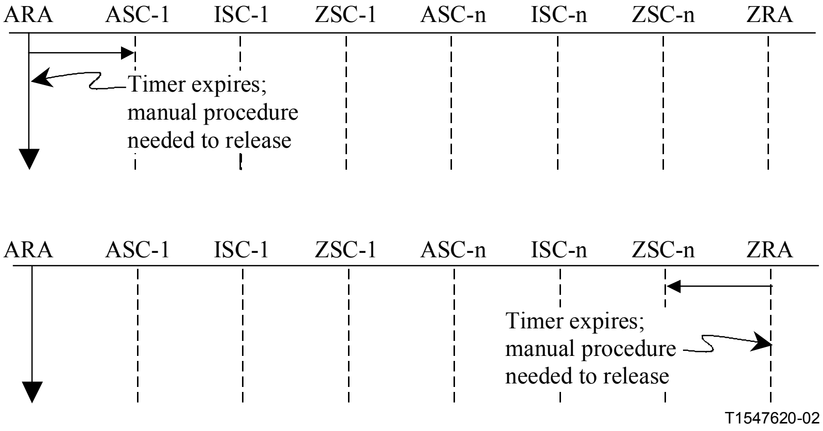

ARA
ASC-1
ISC-1
ZSC-1
ASC-n
ISC-n
ZSC-n
ZRA
Timer expires; manual procedure needed to release
ARA
ASC-1
ISC-1
ZSC-1
ASC-n
ISC-n
ZSC-n
ZRA
Timer expires; manual procedure needed to release
T1547620-02

Figure 6-22/G.7713/Y.1704 – Signal flow: Release call → User Initiated Release (User Timer expires)

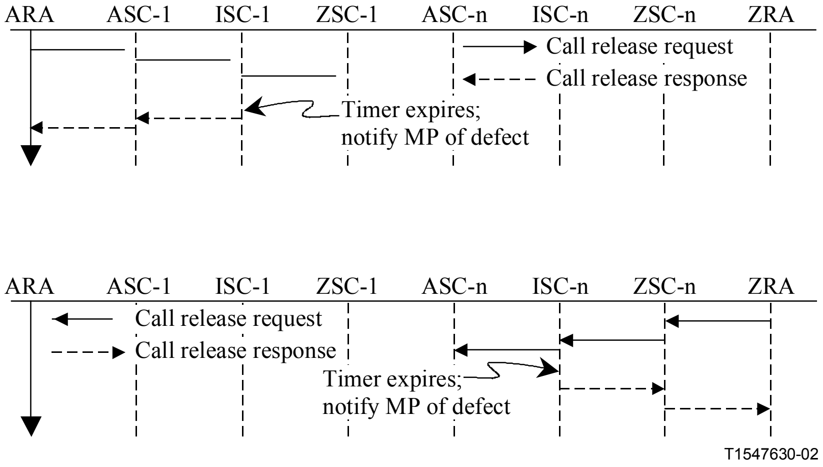


Figure 6-23/G.7713/Y.1704 – Signal flow: Release call → User Initiated Release (Network timer expires)

ARA (or ZRA) timer expires. The user will require alternative means for releasing the call, e.g. manual process. In the case where the network timer expires, the network sends a release confirmation response to the user. In addition, it notifies the management system of the defect. This allows any cleanup of the partially released connections.

Note that in the case of partial release of connections, mechanisms need to be in place to prevent AUSN or ZUSN access to the call.

6.3.3.2 ARA or ZRA initiated call release (response message)

This signalling behaviour arise due to the following cause codes:

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

– The CC did not acknowledge release response.

Figure 6-24 illustrates the signals flows for the defects. In the first case, from the ARA 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 scenario (in 6.3.3.1) the connections may still be up, while for this example connection has been partially released.

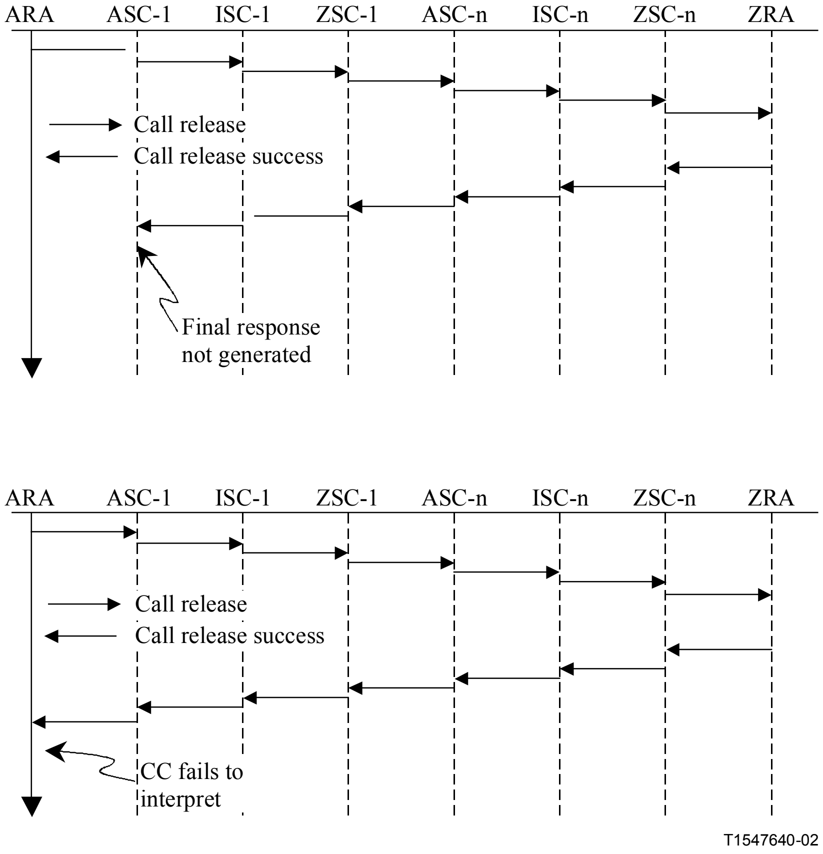


Figure 6-24/G.7713/Y.1704 – Signal Flow: Release → Signalling defect of the response message

7 DCM Attributes List

The DCM attributes list may be separated into attributes associated with the UNI request and attributes associated with the NNI (I-NNI and E-NNI). Tables 7-1, 7-2 and 7-3 summarise a candidate list of attributes that are considered for signalling processing. They represent the logical information that is exchanged across the respective interfaces to support the CallC, CC, and LRM. Note that protocol design decisions may result in aggregation (or segmentation) of some of this logical information; however, the functions supported by the information shall be present.

Table 7-1/G.7713/Y.1704 – UNI Attributes List

	Attributes	Format	Scope
Identity Attributes	A-end user name	String	End-to-end
	Z-end user name	String	End-to-end
	Initiating CC/CallC name	String	Local
	Terminating CC/CallC name	String	Local
	Connection name	String	Local
	Call name	String	End-to-end
Service attributes	SNP ID	String	Local
	SNPP ID	String	Local
	Directionality	String	Local
Policy attributes	CoS	String	End-to-end (Note)
	GoS	String	End-to-end (Note)
	Security	String	Local
NOTE 1 – Although CoS and GoS are end-to-end in scope; its values may be changed as it cross domains. However, the policy associated with the requested service should be met.			

Table 7-2/G.7713/Y.1704 – I-NNI Attributes List

	Attributes	Format	Scope
Identity Attributes	A-end user name	String	End-to-end
	Z-end user name	String	End-to-end
	Initiating CC/CallC name	String	Local
	Terminating CC/CallC name	String	Local
	Connection name	String	Local
	Call name	String	End-to-end
Service attributes	SNP ID	String	Local
	SNPP ID	String	Local
	Directionality	String	Local
Policy attributes	CoS	String	End-to-end
	GoS	String	End-to-end
	Explicit resource list	String	Local
	Recovery	String	Local

Table 7-3/G.7713/Y.1704 – E-NNI Attributes List

	Attributes	Format	Scope
Identity attributes	A-end user name	String	End-to-end
	Z-end user name	String	End-to-end
	Initiating CC/CallC name	String	Local
	Terminating CC/CallC name	String	Local
	Connection name	String	Local
	Call name	String	End-to-end
Service attributes	SNP ID	String	Local
	SNPP ID	String	Local
	Directionality	String	Local
Policy attributes	CoS	String	End-to-end
	GoS	String	End-to-end
	Security	String	Local
	Explicit resource list	String	Local
	Recovery	String	Local

7.1 UNI Attributes List

7.1.1 Identity Attributes

7.1.1.1 A-end user name

This attribute specifies the name of the A-end user. 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 Z-end user name

This attribute specifies the name of the Z-end user. Characteristics of the Z-end user name are the same as that specified for the A-end user name.

7.1.1.3 Initiating CC/CallC name

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

7.1.1.4 Terminating CC/CallC name

This attribute specifies the name associated with the CC/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 a SNPP. The value of this attribute is locally unique, is automatically discovered or provisioned (as per ITU-T Rec. G.7714/Y.1705), and 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.

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.

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. In the case of asymmetric bidirectional, additional information is provided in the attribute to specify number of connections requested for both downstream and upstream.

7.1.3 Policy Attributes

7.1.3.1 Class of Service

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

7.1.3.2 Grade of Service

This attribute specifies the GoS for the requested call and is used to further specify the GoS associated with each CoS 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 CoS to another network's CoS value may be needed to support end-to-end CoS requests. Example list of information that may be contained within GoS includes:

– Diversity information, or

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

7.1.4 Status Attributes

7.1.4.1 Response code for call

Call setup – success;

Call setup – failed: called party busy;

Call setup – failed: calling party busy;

Call setup – failed: network busy;

Call setup – failed: message error;

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

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

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

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

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

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

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

Call setup – failed: policy error: invalid CoS;

Call setup – failed: policy error: unavailable CoS;

Call setup – failed: policy error: invalid GoS;

Call setup – failed: policy error: unavailable GoS;

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

Call release – success;

Call release – failed: message error;

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

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

7.1.4.2 Notification codes

Call error – non-service affecting;

Call error –service affecting.

7.2 I-NNI Attributes List

7.2.1 Identity Attributes

7.2.1.1 A-end user name

See definition in 7.1.1.1.

7.2.1.2 Z-end user name

See definition in 7.1.1.2.

7.2.1.3 Initiating CC/CallC name

See definition in 7.1.1.3.

7.2.1.4 Terminating CC/CallC name

See definition in 7.1.1.4.

7.2.1.5 Connection name

See definition in 7.1.1.5.

7.2.1.6 Call name

See definition in 7.1.1.6.

7.2.2 Service Attributes

7.2.2.1 SNP ID

See definition in 7.1.2.1.

7.2.2.2 SNPP ID

See definition in 7.1.2.2.

7.2.2.3 Directionality

See definition in 7.1.2.3.

7.2.3 Policy Attributes

7.2.3.1 CoS

See definition in 7.1.3.1.

7.2.3.2 GoS

See definition in 7.1.3.2.

7.2.3.3 Explicit resource list

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

7.2.3.4 Recovery

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

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

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

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

7.2.4 Status Attributes

7.2.4.1 Response code for connection

Connection setup – success;

Connection setup – failed: message error;

Connection setup – failed: called party busy;

Connection setup – failed: calling party busy;

Connection setup – failed: timeout;

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

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

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

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

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

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

Connection setup – failed: policy error: invalid recovery;

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

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

Connection release – success;

Connection release – failed: message error;

Connection release – failed: timeout;

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

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

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

7.2.4.2 Notification codes

Connection error – non-service affecting;

Connection error –service affecting;

Connection error – unexpected call release.

7.3 E-NNI Attributes List

7.3.1 Identity Attributes

7.3.1.1 A-end user name

See definition in 7.1.1.1.

7.3.1.2 Z-end user name

See definition in 7.1.1.2.

7.3.1.3 Initiating CC/CallC name

See definition in 7.1.1.3.

7.3.1.4 Terminating CC/CallC name

See definition in 7.1.1.4.

7.3.1.5 Connection name

See definition in 7.1.1.5.

7.3.1.6 Call name

See definition in 7.1.1.6.

7.3.2 Service Attributes

7.3.2.1 SNP ID

See definition in 7.1.2.1.

7.3.2.2 SNPP ID

See definition in 7.1.2.2.

7.3.2.3 Directionality

See definition in 7.1.2.3.

7.3.3 Policy Attributes

7.3.3.1 CoS

See definition in 7.1.3.1.

7.3.3.2 GoS

See definition in 7.1.3.2.

7.3.3.3 Security

See definition in 7.1.3.3.

7.3.3.4 Explicit resource list

See definition in 7.2.3.3.

7.3.3.5 Recovery

See definition in 7.2.3.4.

7.3.4 Status Attributes

7.3.4.1 Response code for connection

Connection setup – success;
Connection setup – failed: message error;
Connection setup – failed: called party busy;
Connection setup – failed: calling party busy;
Connection setup – failed: timeout;
Connection setup – failed: identity error: invalid A-end user name;
Connection setup – failed: identity error: invalid Z-end user name;
Connection setup – failed: identity error: invalid connection name;
Connection setup – failed: service error: invalid SNP ID;
Connection setup – failed: service error: unavailable SNP ID;
Connection setup – failed: service error: invalid SNPP ID;
Connection setup – failed: service error: unavailable SNPP ID;
Connection setup – failed: policy error: invalid CoS;
Connection setup – failed: policy error: unavailable CoS;
Connection setup – failed: policy error: invalid GoS;
Connection setup – failed: policy error: unavailable GoS;

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

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

Connection setup – failed: policy error: invalid recovery;

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

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

Connection release – success;

Connection release – failed: message error;

Connection release – failed: timeout;

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

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

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

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

7.3.4.2 Notification codes

Connection error – non-service affecting;

Connection error –service affecting;

Connection error – unexpected call release.

8 DCM Message Sets

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

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

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

	UNI Messages
Call Setup messages	CallSetupRequest
	CallSetupIndication
	CallSetupConfirm
Call Release messages	CallReleaseRequest
	CallReleaseIndication
Call Query messages	CallQueryRequest
	CallQueryIndication
Call notification message	CallNotify

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

	I-NNI Messages
Connection Setup messages	ConnectionSetupRequest ConnectionSetupIndication ConnectionSetupConfirm
Connection Release messages	ConnectionReleaseRequest ConnectionReleaseIndication
Connection Query messages	ConnectionQueryRequest ConnectionQueryIndication
Connection notification message	ConnectionNotify

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

	E-NNI Messages
Connection Setup messages	ConnectionSetupRequest ConnectionSetupIndication ConnectionSetupConfirm
Connection Release messages	ConnectionReleaseRequest ConnectionReleaseIndication
Connection Query messages	ConnectionQueryRequest ConnectionQueryIndication
Connection notification message	ConnectionNotify

8.1 UNI Messages

8.1.1 Call Setup

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

8.1.1.1 Request: Setup Call

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

Table 8-4/G.7713/Y.1704 – UNI Call Setup Request Message

User sent attributes	Network sent attributes
A-end user name	A-end user name
Z-end user name	Z-end user name
Initiating CC/CallC name	Initiating CC/CallC name
Terminating CC/CallC name	Terminating CC/CallC name
SNP ID	SNP ID
SNPP ID	SNPP ID
Directionality	Directionality
CoS	CoS
GoS	GoS
Security	Security
connName	connName
	callName

8.1.1.2 Indication: Setup Call

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

Table 8-5/G.7713/Y.1704 – UNI Call Setup Indication Message

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

8.1.1.3 Confirm: Setup Call

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

Table 8-6/G.7713/Y.1704 – UNI Call Setup Confirm Message

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

8.1.2 Call Release

8.1.2.1 Request: Release Call

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

Table 8-7/G.7713/Y.1704 – UNI Link Connection Release Request Message

User sent attributes	Network sent attributes
callName	callName
Security	Security

8.1.2.2 Indication: Release Call

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

Table 8-8/G.7713/Y.1704 – UNI Call Release Indication Message

User sent attributes	Network sent attributes
callName	callName
Status	Status

8.1.3 Call Query

The following status codes are valid for query

Call active;

Call does not exist;

Call unavailable;

Call pending.

8.1.3.1 Request: Query per call characteristics

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

Table 8-9/G.7713/Y.1704 – UNI Call Query Request Message

User sent attributes	Network sent attributes
callName	callName
Security	Security

8.1.3.2 Response: Query per call characteristics

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

Table 8-10/G.7713/Y.1704 – UNI Call Query Response Message

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

8.1.3.3 Request: Query all call

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

Table 8-11/G.7713/Y.1704 – UNI All Call Query Request Message

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

8.1.3.4 Response: Query all call

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

Table 8-12/G.7713/Y.1704 – UNI All Link Connection Query Response Message

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

8.1.4 Notification

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

Table 8-13/G.7713/Y.1704 – UNI Notification Message

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

The notification message is one-phase.

8.2 I-NNI Messages

8.2.1 Connection Setup

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

8.2.1.1 Request: Setup Connection

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

Table 8-14/G.7713/Y.1704 – I-NNI Connection Setup Request Message

Attributes
A-end user name
Z-end user name
Initiating CC/CallC name
Terminating CC/CallC name
connName
callName
SNP ID
SNPP ID
Directionality
CoS
GoS
Explicit resource list
Recovery

8.2.1.2 Indication: Setup Connection

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

Table 8-15/G.7713/Y.1704 – I-NNI Connection Setup Indication Message

Attributes
connName
callName
Status

8.2.1.3 Confirm: Setup Connection

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

Table 8-16/G.7713/Y.1704 – I-NNI Connection Setup Confirm Message

Attributes
connName
callName
Status

8.2.2 Connection Release

8.2.2.1 Request: Release Connection

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

Table 8-17/G.7713/Y.1704 – I-NNI Connection Release Request Message

Attributes
callName
connName

8.2.2.2 Indication: Release Connection

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

Table 8-18/G.7713/Y.1704 – I-NNI Connection Release Indication Message

Attributes
callName
connName
Status

8.2.3 Connection Query

The following status codes are valid for query:

Connection active;

Connection does not exist;

Connection unavailable;

Connection pending.

8.2.3.1 Request: Query per connection characteristics

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

Table 8-19/G.7713/Y.1704 – I-NNI Connection Query Request Message

Attributes
callName
connName

8.2.3.2 Response: Query per connection characteristics

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

Table 8-20/G.7713/Y.1704 – I-NNI Connection Query Response Message

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

8.2.3.3 Request: Query all connection

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

Table 8-21/G.7713/Y.1704 – I-NNI All Connection Query Request Message

Attributes
SNCr name

8.2.3.4 Response: Query all connection

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

Table 8-22/G.7713/Y.1704 – I-NNI All Connection Query Response Message

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

8.2.4 Notification

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

Table 8-23/G.7713/Y.1704 – I-NNI Notification Message

Attributes
callName
connName
Error code

The notification message is one-phase.

8.3 E-NNI Messages

8.3.1 Connection Setup

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

8.3.1.1 Request: Setup Connection

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

Table 8-24/G.7713/Y.1704 – E-NNI Connection Setup Request Message

Attributes
A-end user name
Z-end user name
Initiating CC/CallC name
Terminating CC/CallC name
connName
callName
SNP ID
SNPP ID
Directionality
CoS
GoS
Security
Explicit resource list
Recovery

8.3.1.2 Indication: Setup Connection

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

Table 8-25/G.7713/Y.1704 – E-NNI Connection Setup Indication Message

Attributes
connName
callName
Status

8.3.1.3 Confirm: Setup Connection

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

Table 8-26/G.7713/Y.1704 – E-NNI Connection Setup Confirm Message

Attributes
connName
callName
Status

8.3.2 Connection Release

8.3.2.1 Request: Release Connection

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

Table 8-27/G.7713/Y.1704 – E-NNI Connection Release Request Message

Attributes
callName
connName
Security

8.3.2.2 Indication: Release Connection

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

Table 8-28/G.7713/Y.1704 – E-NNI Connection Release Indication Message

Attributes
callName
connName
Status

8.3.3 Connection Query

The status codes are identified in 8.2.3.

8.3.3.1 Request: Query per connection characteristics

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

Table 8-29/G.7713/Y.1704 – E-NNI Connection Query Request Message

Attributes
callName
connName
Security

8.3.3.2 Response: Query per connection characteristics

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

Table 8-30/G.7713/Y.1704 – E-NNI Connection Query Response Message

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

8.3.3.3 Request: Query all connection

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

Table 8-31/G.7713/Y.1704 – E-NNI All Connection Query Request Message

Attributes
SNCr name
Security

8.3.3.4 Response: Query all connection

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

Table 8-32/G.7713/Y.1704 – E-NNI All Connection Query Response Message

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

8.3.4 Notification

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

Table 8-33/G.7713/Y.1704 – E-NNI Notification Message

Attributes
callName
connName
Error code
Security

The notification message is one-phase.

9 DCM State Diagrams

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

The state transitions are specified for:

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

– Terminating user: the user that terminates the call. For setup operation this is the Z-end user (ZRA), while for release operation this may be either Z-end or 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 actions.

– **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 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 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 actions.
- **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 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, user decides to request a call, traffic engineering management issues a new call, a failed connection leading to request to release a call, etc. These may be triggered via a user interface or application interface.
- Reception of message events.

– A result of verification events.

– Timeout events

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

9.1 Call State

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

Table 9-1/G.7713/Y.1704 – User CallC Call State Events

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

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

Table 9-2/G.7713/Y.1704 – Network CallC Call State Events

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

Table 9-2/G.7713/Y.1704 – Network CallC Call State Events

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

9.1.1 Initiating User CallC Call State

9.1.1.1 Initiating User Call State: Setup

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

Table 9-3/G.7713/Y.1704 – Initiating User CallC Call Setup State Transitions

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

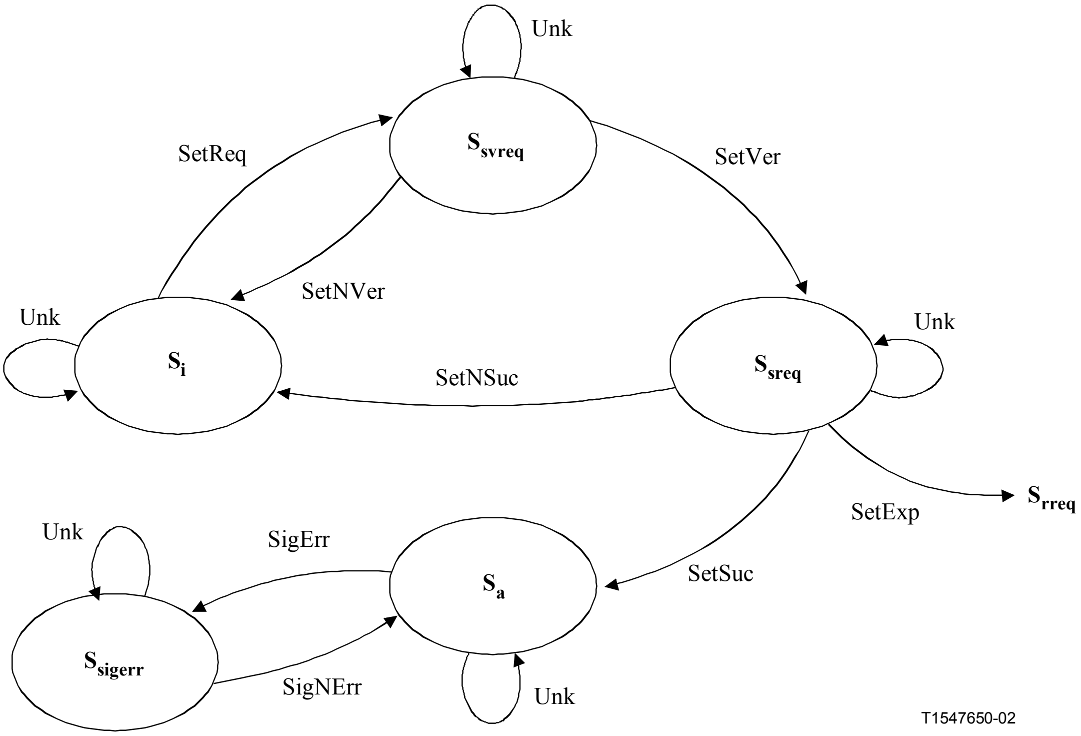


Figure 9-1/G.7713/Y.1704 – Initiating User CallC Call Setup State Diagrams

9.1.1.2 Initiating User Call State: Release

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

Table 9-4/G.7713/Y.1704 – Initiating User CallC Call Release State Transitions

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

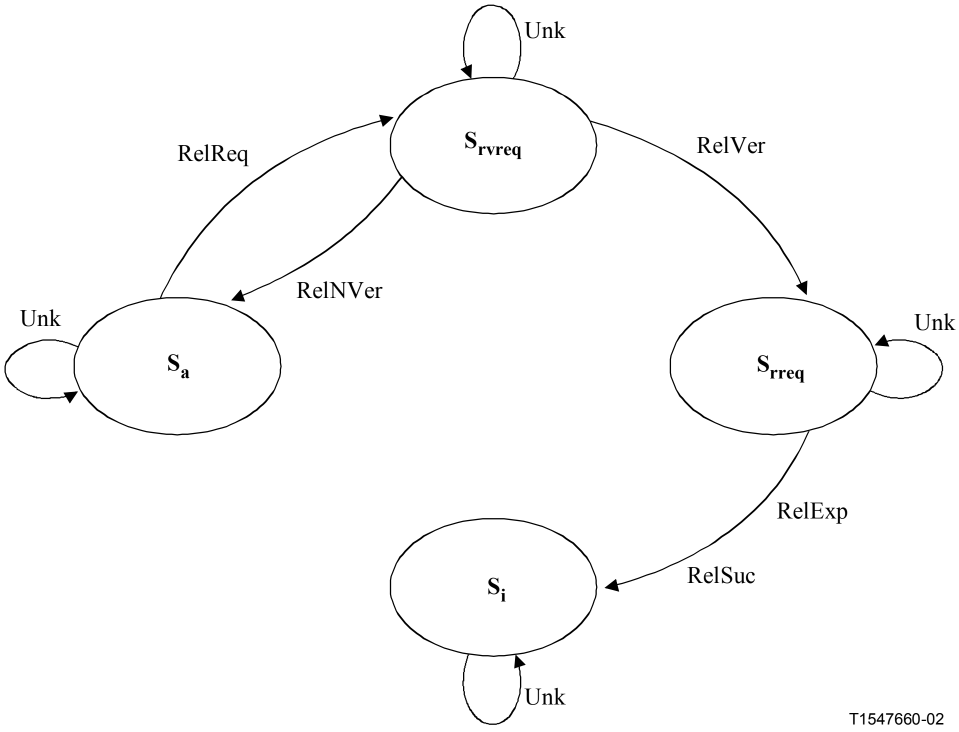


Figure 9-2/G.7713/Y.1704 – Initiating User CallC Call Release State Diagrams

9.1.2 Terminating User CallC Call State

9.1.2.1 Terminating User Call State: Setup

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

Table 9-5/G.7713/Y.1704 – Terminating User CallC Call Setup State Transitions

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

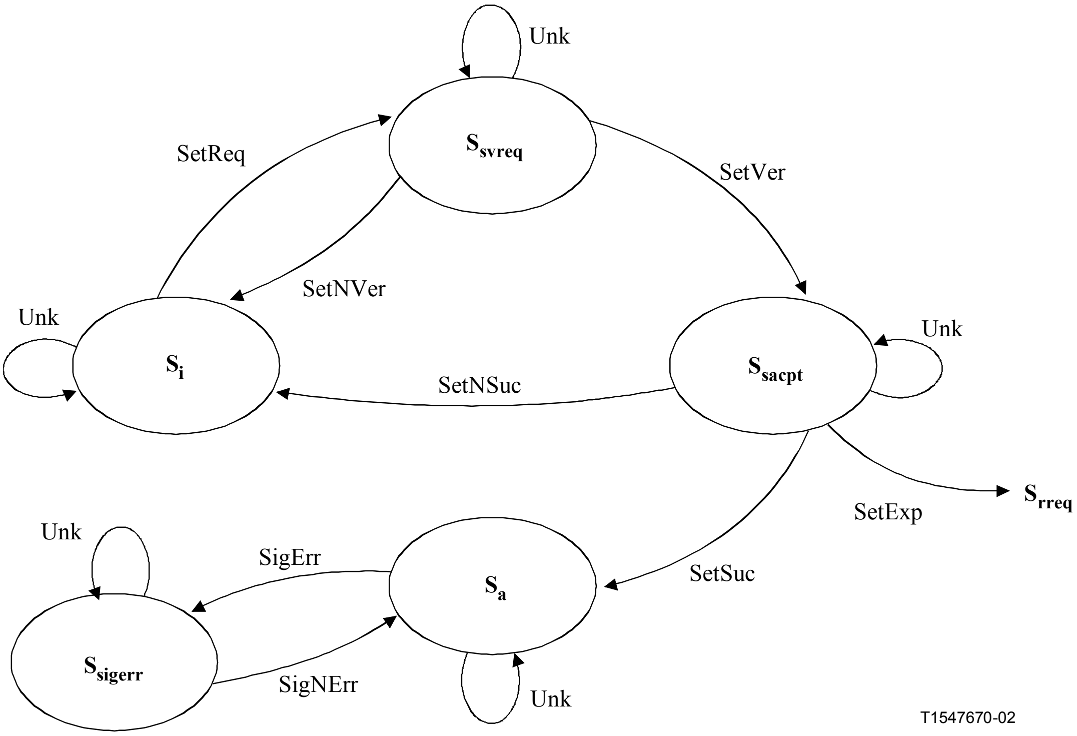


Figure 9-3/G.7713/Y.1704 – Terminating User CallC Call Setup State Diagrams

9.1.2.2 Terminating User Call State: Release

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

Table 9-6/G.7713/Y.1704 – Terminating User CallC Call Release State Transitions

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

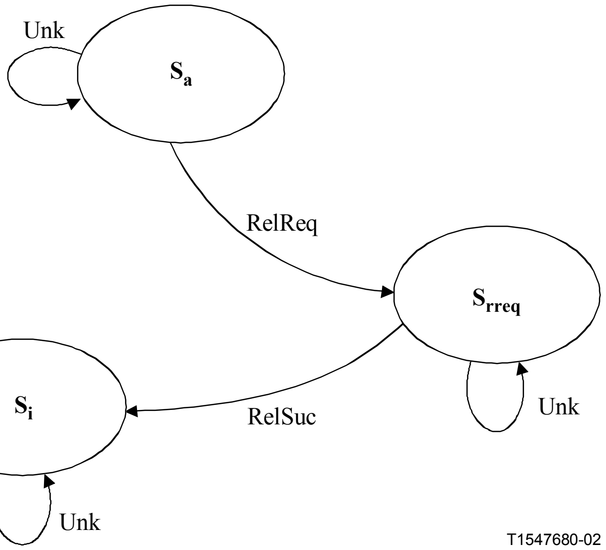


Figure 9-4/G.7713/Y.1704 – Terminating User CallC Call Release State Diagrams

9.1.3 Network CallC Call State

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

9.1.3.1 Network CallC Call State: Setup

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

Table 9-7/G.7713/Y.1704 – Network CallC Call Setup State Transitions

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

Table 9-7/G.7713/Y.1704 – Network CallC Call Setup State Transitions

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

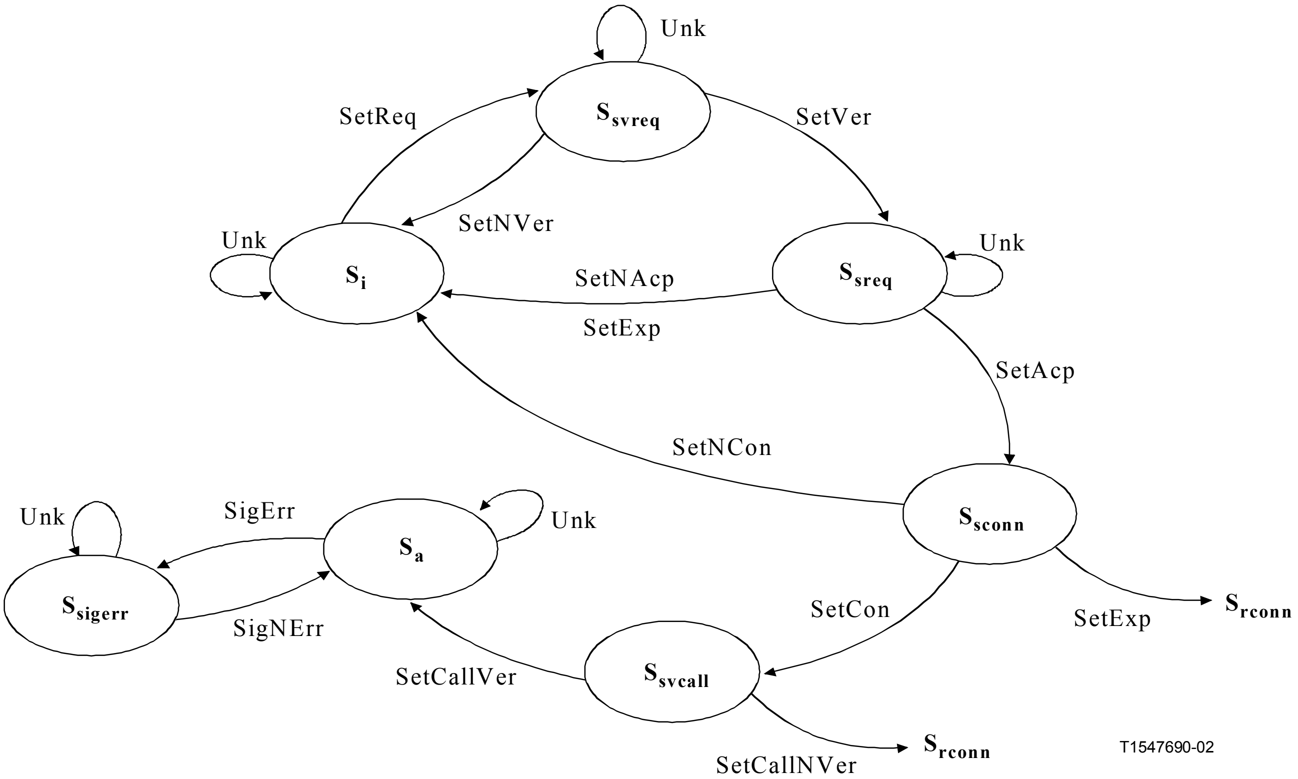


Figure 9-5/G.7713/Y.1704 – Network CallC Call Setup State Diagrams

9.1.3.2 Network CallC Call State: Release

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

Table 9-8/G.7713/Y.1704 – Network CallC Call Release State Transitions

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

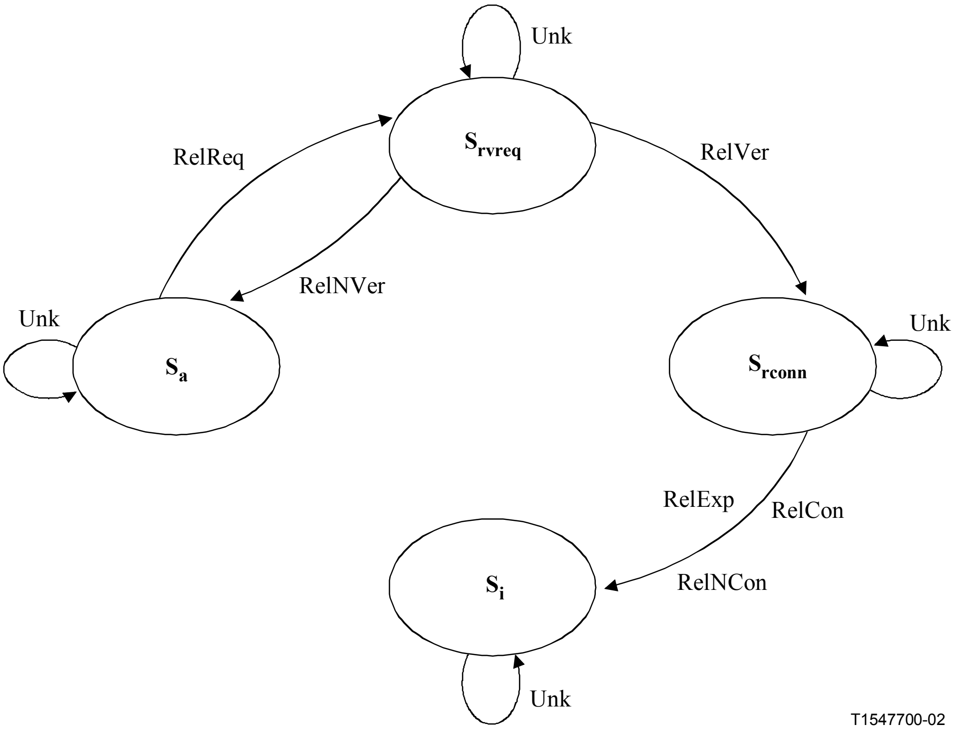


Figure 9-6/G.7713/Y.1704 – Network CallC Call Release State Diagrams

9.2 Connection State

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

Table 9-9/G.7713/Y.1704 – User CC Connection State Events

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

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

Table 9-10/G.7713/Y.1704 – Network CC Connection State Events

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

Table 9-10/G.7713/Y.1704 – Network CC Connection State Events

Event	Event description
SigErr	Defect detected at the signalling communication channel.
SigNErr	Signalling communication channel defect repaired.

9.2.1 Initiating User CC (for SC) or Initiating Network CC (for SPC) Connection State

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

9.2.1.1 Initiating CC Connection State: Setup

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

Table 9-11/G.7713/Y.1704 – Initiating CC Connection Setup State Transitions

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

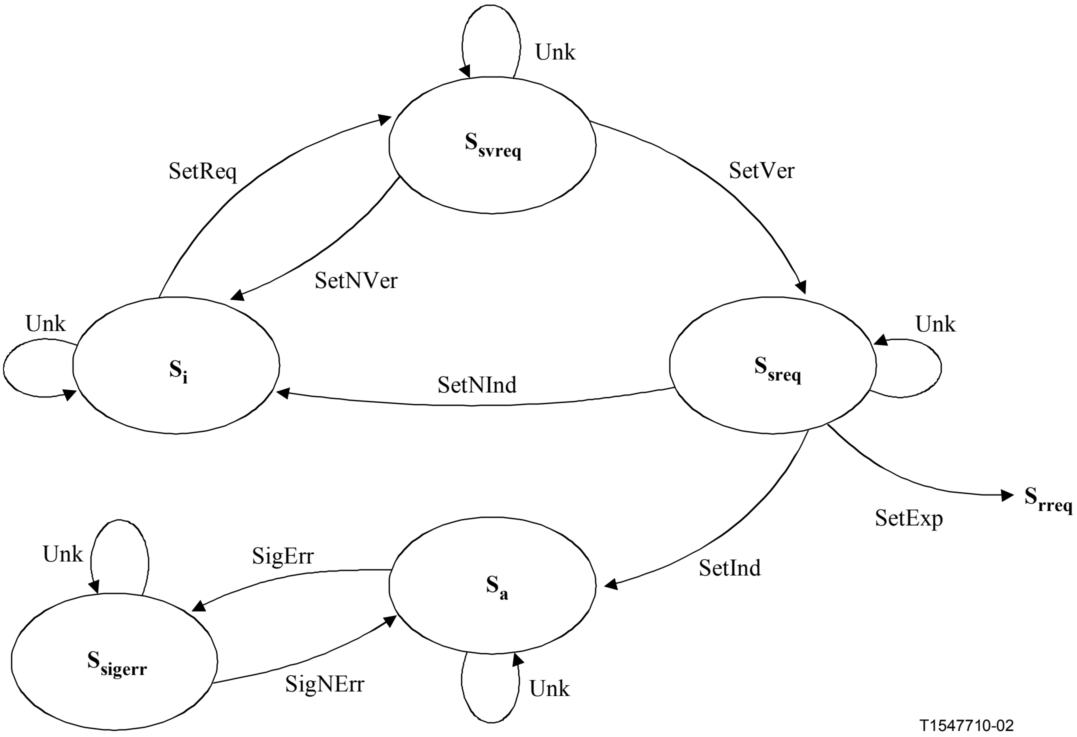


Figure 9-7/G.7713/Y.1704 – Initiating CC Connection Setup State Diagrams

9.2.1.2 Initiating CC Connection State: Release

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

Table 9-12/G.7713/Y.1704 – Initiating CC Connection Release State Transitions

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

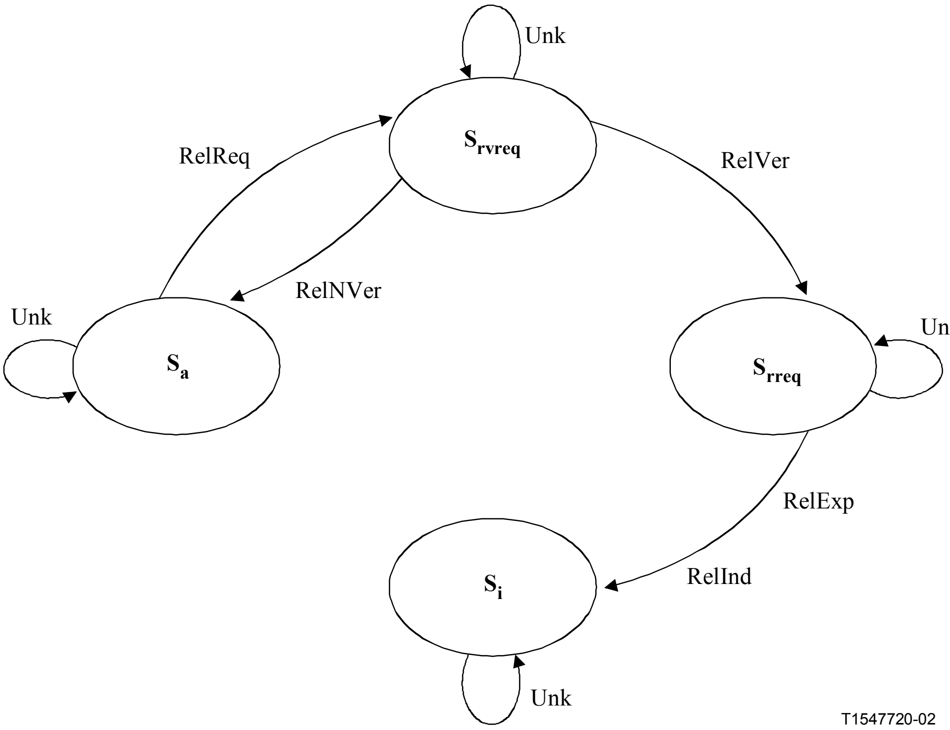


Figure 9-8/G.7713/Y.1704 – Initiating CC Connection Release State Diagrams

9.2.2 Terminating User CC (for SC) or Terminating Network CC (for SPC) Connection State

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

9.2.2.1 Terminating CC Connection State: Setup

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

Table 9-13/G.7713/Y.1704 – Terminating CC Connection Setup State Transitions

Current state	Event	Actions taken	Next state
*	Unk	• Send a notification message to the sender and inform 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}

Table 9-13/G.7713/Y.1704 – Terminating CC Connection Setup State Transitions

Current state	Event	Actions taken	Next state
S_a	SigErr	• none.	S_{sigerr}
S_{sigerr}	SigNErr	• none.	S_a

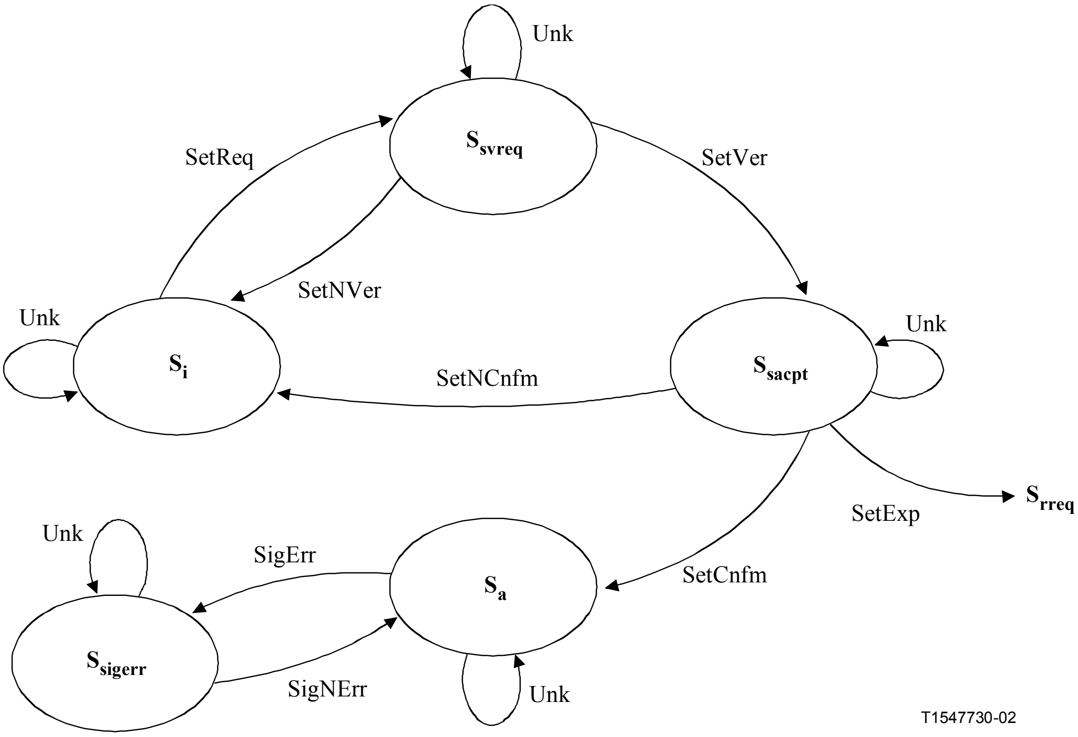


Figure 9-9/G.7713/Y.1704 – Terminating CC Connection Setup State Diagrams

9.2.2.2 Terminating CC Connection State: Release

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

Table 9-14/G.7713/Y.1704 – Terminating CC Connection Release State Transitions

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

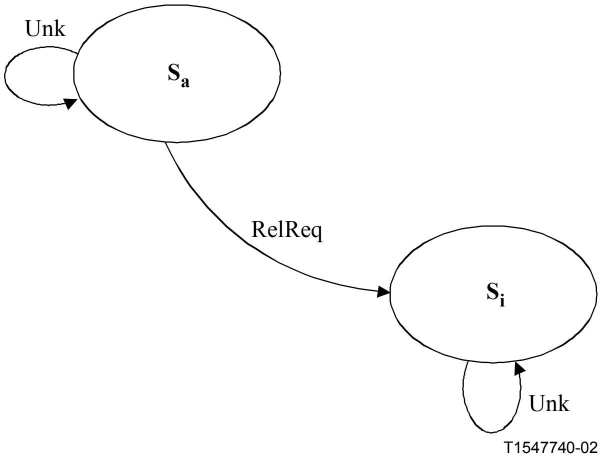


Figure 9-10/G.7713/Y.1704 – Terminating CC Connection Release State Diagrams

9.2.3 Intermediate CC Connection State

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

9.2.3.1 Intermediate CC Connection State: Setup

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

Table 9-15/G.7713/Y.1704 – Intermediate CC Connection Setup State Transitions

Current state	Event	Actions taken	Next state
*	Unk	• Send a notification message to the sender and inform the error. No state transition occurs.	*
S_i	SetReq	• Verify the request.	S_{svreq}
S_{svreq}	SetVer	• Send connection setup request message. • Initiate connection setup timer (T_{conn_setup}).	S_{sreq}
S_{svreq}	SetNVer	• Notify connection initiator of denied connection setup request.	S_i
S_{sreq}	SetInd	• Delete connection setup timer (T_{conn_setup}). • Send connection indication message. • Initiate connection setup timer (T_{conn_setup}).	S_{sacpt}
S_{sreq}	SetNInd	• Delete connection setup timer (T_{conn_setup}). • Notify connection initiator of denied connection setup request.	S_i
S_{sreq}	SetExp	• Notify connection initiator of denied connection setup request. • Send connection release request message. • Initiate connection release timer ($T_{conn\_release}$).	S_{rreq}
S_{sacpt}	SetCnfm	• Delete connection setup timer (T_{conn_setup}).	S_{svconn}

Table 9-15/G.7713/Y.1704 – Intermediate CC Connection Setup State Transitions

Current state	Event	Actions taken	Next state
S_{sacpt}	SetNCnfm	• Delete connection setup timer (T_{conn_setup}). • Notify connection initiator of denied connection setup request.	S_i
S_{sacpt}	SetExp	• Notify connection initiator of denied connection setup request. • Send connection release request message. • Initiate connection release timer ($T_{conn\_release}$).	S_{rreq}
S_{svconn}	SetConnVer	• Send connection confirmation message.	S_a
S_{svconn}	SetConnNVer	• Notify connection initiator of denied connection setup request. • Send connection release request message. • Initiate connection release timer ($T_{conn\_release}$).	S_{rreq}
S_a	SigErr	• none.	S_{sigerr}
S_{sigerr}	SigNErr	• none.	S_a

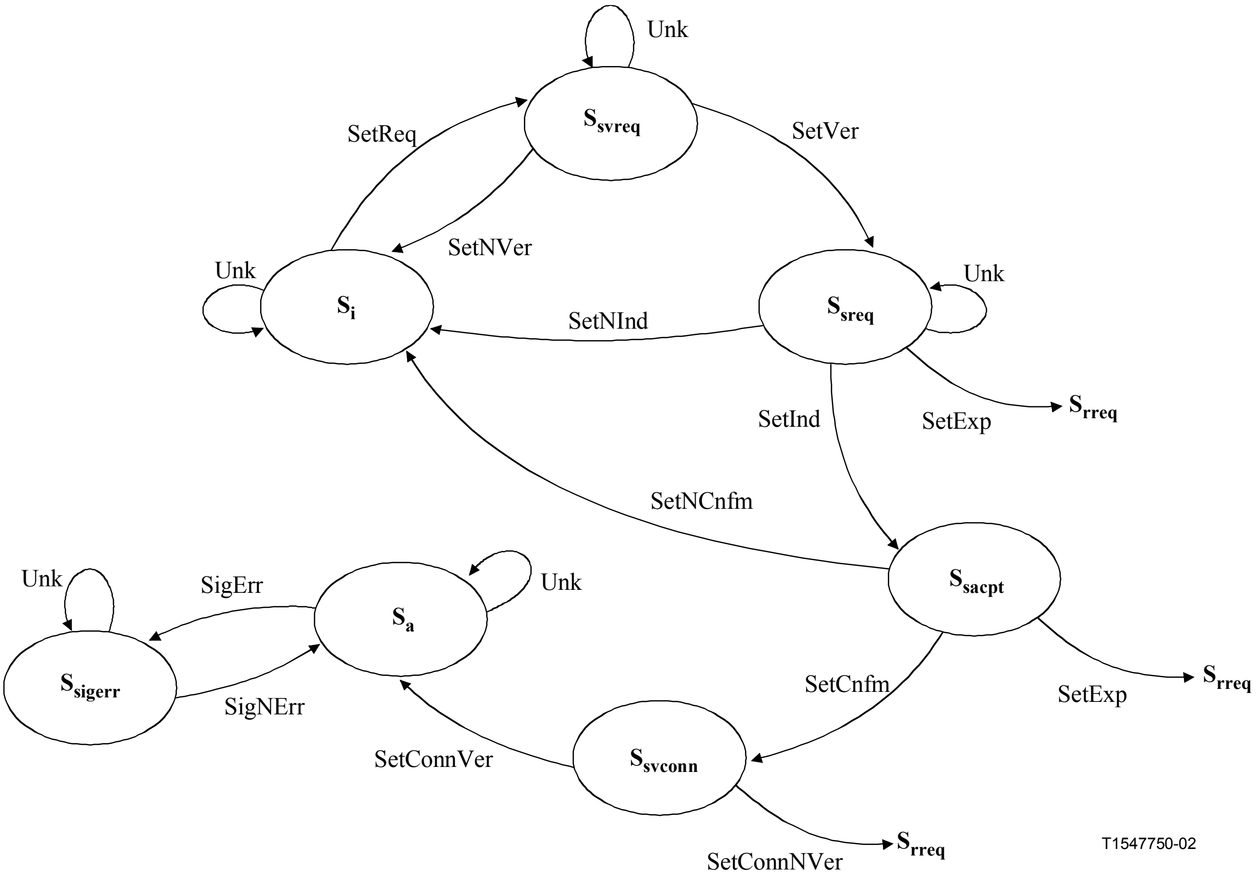


Figure 9-11/G.7713/Y.1704 – Intermediate CC Connection Setup State Diagrams

9.2.3.2 Intermediate CC Connection State: Release

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

Table 9-16/G.7713/Y.1704 – Intermediate CC Connection Release State Transitions

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

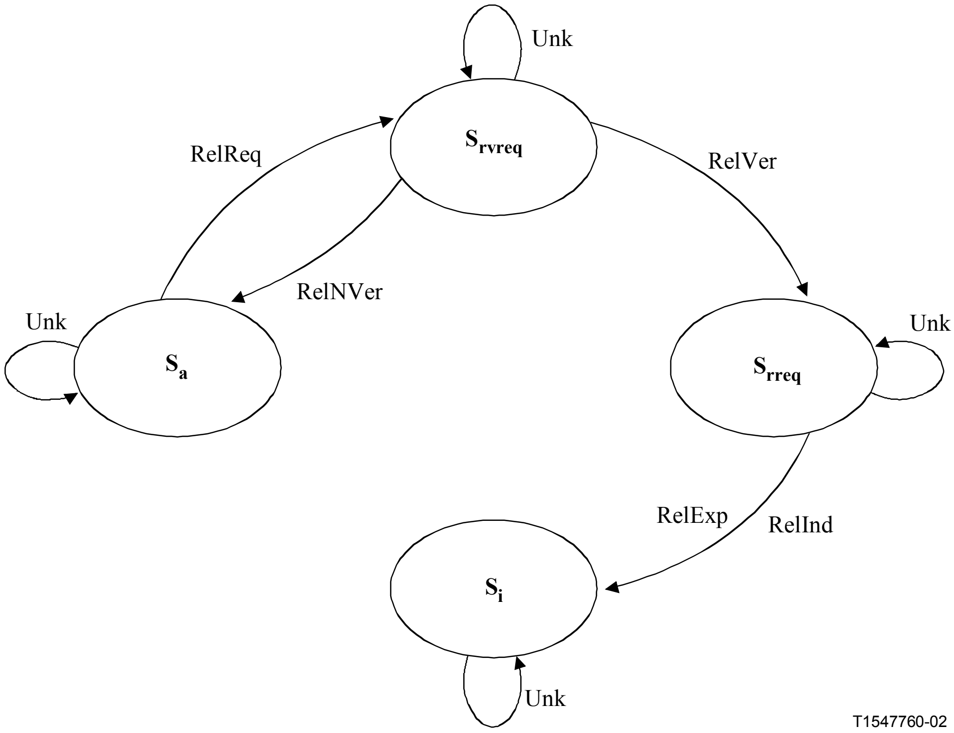


Figure 9-12/G.7713/Y.1704 – Intermediate CC Connection Release State Diagrams

10 Management of the Call and Connection Controller Function

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

- Some of the functions of the management plane are to support the capabilities of fault management, configuration management (including resource allocation/de-allocation), performance management, security management and accounting.
- The transport plane supports the capabilities of payload transport, performance monitoring, fault detection, and protection switch.
- The control plane supports the capabilities of dynamic path computation, dynamic distributed call and connection setup/release, dynamic protection/restoration allocations/ assignments and restoration.

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

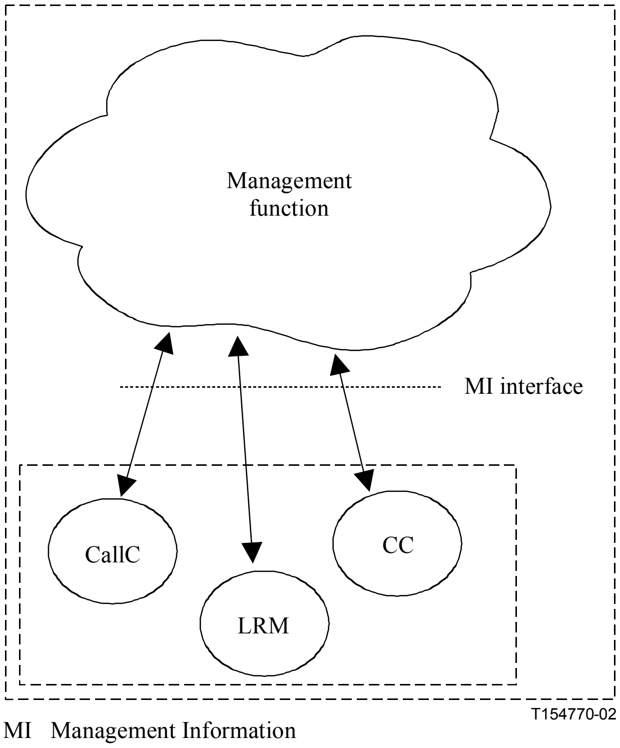


Figure 10-1/G.7713/Y.1704 – Interface between Control Plane Components and Management Plane

The management system is able to set certain information regarding the components' behaviour and is able to communicate such information to the components. The components may also report certain information over which the components have control. This reported information by the components may be categorised as information that is autonomously communicated by the component to the management system, or information that is communicated to the management system via 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 management plane regarding the health of the component (i.e. "enabled", capable of performing its functions or "disabled", incapable of performing its functions). The MI signal **MI_CallCproblemList**, **MI_CCproblemList**, and **MI_LRMproblemList** autonomously provides detailed information about the health of the component and its associated interfaces (i.e. list of problems either detected at the component interface or within a component).

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

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

The following candidate MI signals are controlled by the components:

– Tracking of the component performance: The CallC and CC provide the supervision and management of call and connection requests. As such, various parameters may be tracked by these components, including service usage information and call attempt status information. Other types of information such as connection usage are not considered part of these components, and is thus not described. If the component provides tracking of connection requests, then the MI signals are needed to support management query of the control plane. These MI signals are specified for the CallC, CC and LRM components.

  - **MI_CallCcallDuration** provides tracking of the duration of a call sourced by the CallC.
  - **MI_CallCcallState** provides information about the state of a particular call based on the states defined in clause 9.
  - **MI_CCconnectionState** provides information about the state of a particular connection based on the states defined in clause 9.
  - **MI_LRMconnectionState** provides information about the state of a particular link connection based on the states defined in ITU-T Rec. G.8080/Y.1304.
  - **MI_CallCcallAttempt** provides information about the number of call setup requests (or attempts) received by the CallC.
  - **MI_CallCcallBlocked** provides information about the number of call setup requests received by the CallC that are blocked .

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

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

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

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

MI Signal	Attributes of Signal
MI_CallCstatusReporting MI_CCstatusReporting MI_LRMstatusReporting	Enabled, Disabled
MI_CallCoperationalState MI_CCoperationalState MI_LRMoperationalState	Up, Down
MI_CallCproblemList MI_CCproblemList MI_LRMproblemList	e.g. "Peer component communication failed", "Excessive error interpreting messages"
MI_DCMsigMode	e.g. RSVP-TE, CR-LDP, PNNI
MI_CallCcallDuration	Time Zone, Start time & date, End time & date
MI_CallCcallState	As per states defined in clause 9
MI_CCconnectionState	As per states defined in clause 9
MI_LRMconnectionState	As per states defined in ITU-T Rec. G.8080/Y.1304
MI_CallCcallAttempt	Total number of call setup requests per time period
MI_CallCcallBlocked	Total number of call requests blocked
MI_CallSetupRequest	Attributes the same as attributes for **CallSetupRequest** message
MI_CallSetupIndication	Attributes the same as attributes for **CallSetupIndication** message
MI_requestSetupNewRoute	callName connName A-end user name Z-end user name SNP ID SNPP ID Directionality CoS GoS
MI_responseSetupNewRoute	callName connName SNP ID SNPP ID Directionality Explicit resource list Recovery
MI_verifyCAC	callName connName A-end user name Z-end user name

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

MI Signal	Attributes of Signal
	SNP ID SNPP ID CoS GoS Explicit resource list Recovery
MI_responseCAC	"Accepted", "Denied"
MI_responseSetupResourcesReserved	"Reserved", "Not reserved"
MI_responseSetupResourcesAllocated	"Allocated", "Not allocated"
MI_responseCallResourceFail	"Resource Failed"
MI_CallReleaseRequest	Attributes the same as attributes for **CallReleaseRequest** message
MI_CallReleaseIndication	Attributes the same as attributes for **CallReleaseIndication** 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 the same as specified in 7.2.4.1 for connection release

10.1 Setting Up a Connection

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

- Two types of connections are supported as per ITU-T Rec. G.8080/Y.1304: switched connections and soft permanent connections (SPCs). In the case of SPCs, the connection setup request is initiated by the management plane. As such, a message is needed across the MI interface, **MI_CallSetupRequest**.
- After receipt of the setup request, the internal process may occur in different ways to establish the resources. This may include processes for authentication of the request, determining route information, verification of resource/route information, and allocation of the resources (allocation of resources may include allocations for protection or restoration if service profile requires it). Note that some of these processes may not be required because management system has performed these processes. When the CC requests an outgoing route from the RC (where RC is distributed in the management plane), additional messages are needed for communication of the route information, **MI_requestSetupNewRoute**, **MI_responseSetupNewRoute**.

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

- In the case where CAC function is distributed to the management plane, additional messages are needed for communication with this function, **MI_verifyCAC**, **MI_responseCAC**.
- Upon verification of the request, resource reservation may need to be performed. An additional message is needed for communicating to the CC that resources have been reserved, **MI_responseSetupResourcesReserved**.
- Upon verification of the request, resource allocation needs to be performed. An additional message is needed for communicating to the CC that resources have been allocated, **MI_responseSetupResourcesAllocated**.
- Once the CC processes are complete, the connection request continues to the downstream CC. The downstream CC is determined by the route established by the RC. Upon completion of the request, a response is received from the downstream CC on the status of the request (e.g. confirmed or denied).
  - If the received response is denied, additional processes will be needed to de-allocate and/or unreserve the resource all the way back to the ingress node (i.e. or call originating node).
- Upon determining the status of the connection operation, a response is sent to the CC. This may be the upstream CC, the network CallC, or a management plane (for the SPC). In the case of SPC, connection setup response is sent to the management plane. As such, a message is needed across the MI interface, **MI_CallSetupIndication**.
  - As part of the **MI_CallSetupIndication**, if the connection was not set up, or errors occurred that resulted in partial setup, the error details are sent to the management plane. This may include the cause of the error as well as any partial links (or link connections) that was set up but cannot be released by the control plane.
- After the call's connection(s) have been established, the CallC relies upon the CC and LRM to monitor the connections. If a connection fails, the CallC must know about it in order to take appropriate actions such as restoration or interrupting usage collection for that call (i.e. in the event that the call was released). As such, a message is needed for communicating to the CallC, **MI_responseCallResourceFail**.

10.2 Releasing a Connection

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

- Two types of connections are supported as per ITU-T Rec. G.8080/Y.1304: switched connection and soft permanent connection (SPC). In the case of SPC, connection release request is initiated by the management plane. As such, a message is needed across the NMI interface, **MI_CallReleaseRequest**.
- After receipt of the release request, the CC requests RC for the route of the existing connection. In the case where RC is distributed to the management plane, additional messages are needed for communication of the route information, **MI_requestExistingRoute**, **MI_responseExistingRoute**.
- Upon verification of the request, resources need to be unreserved and de-allocated. An additional message is needed for communicating that resources have been de-allocated, **MI_responseReleaseResourcesDeallocated**.
- Once the CC processes are complete, the connection request continues to the downstream CC. The downstream CC is determined by the route specified by the RC. Upon completion

of the request, a response is received from the downstream CC on the status of the request (e.g. confirmed or denied).

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

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

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

ITU-T Y-SERIES RECOMMENDATIONS

GLOBAL INFORMATION INFRASTRUCTURE AND INTERNET PROTOCOL ASPECTS

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

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 B	Means of expression: definitions, symbols, classification
Series C	General telecommunication statistics
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	TMN and network maintenance: international transmission systems, telephone circuits, telegraphy, facsimile and leased circuits
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Telephone transmission quality, telephone installations, local line networks
Series Q	Switching and signalling
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	Data networks and open system communications
Series Y	**Global information infrastructure and Internet protocol aspects**
Series Z	Languages and general software aspects for telecommunication systems

22912

Printed in Switzerland
Geneva, 2002