ETSI TECHNICAL REPORT

ETR 157-1

March 1996

Source: ETSI TC-SPS

Reference: DTR/SPS-03008-1

ICS: 33.080

Key words: B-ISDN, signalling, CS2

Broadband Integrated Services Digital Network (B-ISDN); Signalling requirements for B-ISDN services; Part 1: Capability Set 2 (CS2)

ETSI

European Telecommunications Standards Institute

ETSI Secretariat

Postal address: F-06921 Sophia Antipolis CEDEX - FRANCE
Office address: 650 Route des Lucioles - Sophia Antipolis - Valbonne - FRANCE
X.400: c=fr, a=atlas, p=etsi, s=secretariat - **Internet:** secretariat@etsi.fr

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

*

Contents

Blank page

Foreword

This ETSI Technical Report (ETR) has been produced by the Signalling Protocols and Switching (SPS) Technical Committee of the European Telecommunications Standards Institute (ETSI). It is based on contribution to ITU-T Study Group 11.

ETRs are informative documents resulting from ETSI studies which are not appropriate for European Telecommunication Standard (ETS) or Interim European Telecommunication Standard (I-ETS) status. An ETR may be used to publish material which is either of an informative nature, relating to the use or the application of ETSs or I-ETSs, or which is immature and not yet suitable for formal adoption as an ETS or an I-ETS.

This ETR is part 1 of a multi-part ETR covering the signalling requirements for B-ISDN services as described below:

Part 1: "Capability Set 2 (CS2)";

NOTE: Additional parts may cover the further development of the B-ISDN services.

Blank page

1 Scope

This ETSI Technical Report (ETR) contains a description of the Broadband Integrated Services Digital Network (B-ISDN) signalling requirements for Capability Set 2 (CS2). These requirements are to be used in the development of the appropriate peer-to-peer and layer-to-layer signalling protocols. These requirements are based on the requirements from ITU-T Study Groups 1 and 13, especially the available stage 1 definitions listed in subclause 3.1.

The following services/capabilities are identified. These will impose signalling requirements in B-ISDN.

a) Establishment of on demand connections.
   Demand a call between exactly two users which is supported by a single point-to-point connection.
b) Support of connections configurations on a point-to-point and point-to-multipoint basis.
c) Support of symmetric and asymmetric connections (e.g. low or zero bandwidth in one direction and high bandwidths in the other).
d) To be able to establish a call and then later add bearers.
e) Specification of Quality of Service (QoS) class, etc.

NOTE: For integration of 64 kbit/s based N-ISDN services and B-ISDN with a broadband signalling protocol, the supplementary services itemized in CCITT Recommendation Q.767 according to the most recent stage 1 and stage 2 recommendation should be supported.

2 Abbreviations

For the purposes of this ETR, the following abbreviations apply:

AAL	ATM Adaptation Layer
AC	Application Control
ATM	Asynchronous Transfer Mode
B-ISDN	Broadband Integrated Services Digital Network
BCLB	Broadband ConnectionLess Bearer
BCOB	Broadband Connection Oriented Bearer
CBR	Constant Bit Rate
CC	Call Control
CCR	Commitment, Concurrency and Recovery
CRCG	Common Route Connections Group
CS2	Capability Set 2
DC	enD-to-end Control
DCA	enD Control Agent
EC	Edge-to-edge Control
FE	Functional Entity
ID	IDentifier
IN	Intelligent Network
IWU	Inter Working Unit
LBA	Local Bearer Access
LC	Link-by-link Control
LCA	Link Control Agent
LEX	Local EXchange
LLC	Lower Layer Component
N-ISDN	Narrowband ISDN
NA	Not Applicable
NNI	Network-Network Interface
NSAP	Network Service Access Point
NT	Network Termination
OAM	Operations, Administration and Management
PC	Presentation Control
PCA	Presentation Control Agent
PDH	Pleisiochronous Digital Hierarchy
PSTN	Public Switched Telephone Network
QoS	Quality of Service
RLS	Release
SAP	Service Access Point

SAR	Segmentation And Reassembly
SC	Service Component
SDH	Synchronous Digital Hierarchy
SDL	Specification and Description Language
SDU	Service Data Unit
SM	Service Module
SMDS	Switched Multi-megabit Data Service
TE	Terminal Equipment
TEI	Terminal Equipment Identifier
TMN	Telecommunication Management Network
UNI	User-Network Interface
VBR	Variable Bit Rate
VC	Virtual Channel
VCC	Virtual Channel Connection
VCI	Virtual Channel Identifier
VP	Virtual Path
VPC	Virtual Path Connection
VPI	Virtual Path Identifier

3 B-ISDN signalling requirements objectives

3.1 B-ISDN Services Requirements

General service characteristics that need to be addressed in the development of B-ISDN Signalling CS2 are as follows:

- signalling Support for Bearer Service Classes X, A, B, C, and D for the connection types identified below. The connection types are defined in table 3.

Table 1: Bearer Service Classes and Connection Types

Bearer Service	Connection Types	
Sub-categories	**Type 1**	**Type 2**
Class X, A, B	☑	☑
Class C assured service	☑	--
Class C unassured service	☑	☑
Class D (dial up only)	☑	☑

- service Interworking with N-ISDN which is presently restricted to the 10 Supplementary Services defined in Broadband Release 1:

  - Calling Line Identification Presentation;
  - Calling Line Identification Restriction;
  - Connected Line Presentation;
  - Connected Line Restriction;
  - Direct Dialling In;
  - Multiple Subscriber Number;
  - Terminal Portability;
  - Closed User Group;
  - Subaddressing;
  - User to User Signalling Service 1;

- Interworking with Frame Relaying, X.25 and 64 kbit/s circuit mode bearer services of N-ISDN.

For additional information about the service specific requirements driving Broadband CS2, the following draft Recommendations for B-ISDN services should be considered:

F.310	(Broadband Videotex Service)
F.722	(Broadband Videotelephony)
F.732	(Broadband Videoconference)
F.811	(Broadband connection oriented bearer service)
F.812	(Broadband connectionless data bearer service)
F.821	(Broadband TV distribution service)
F.822	(Broadband HDTV)

On an interworking basis, the following services should also be considered:

I.122	(Frame Relay Service)
I.555	(Interworking with Frame Relay)
I.580	(Interworking with N-ISDN)
X.2	(Data Communications Services)

3.2 B-ISDN Bearer Services Signalling Requirements

The broadband connection oriented bearer service categories provide unrestricted transfer of user information over a B-ISDN Virtual Path, Virtual Channel, or Common Route Connection Group (CRCG) connection between two or more parties connected at the Sb/Tb reference points. The three forms of B-ISDN connections are illustrated in figure 1.

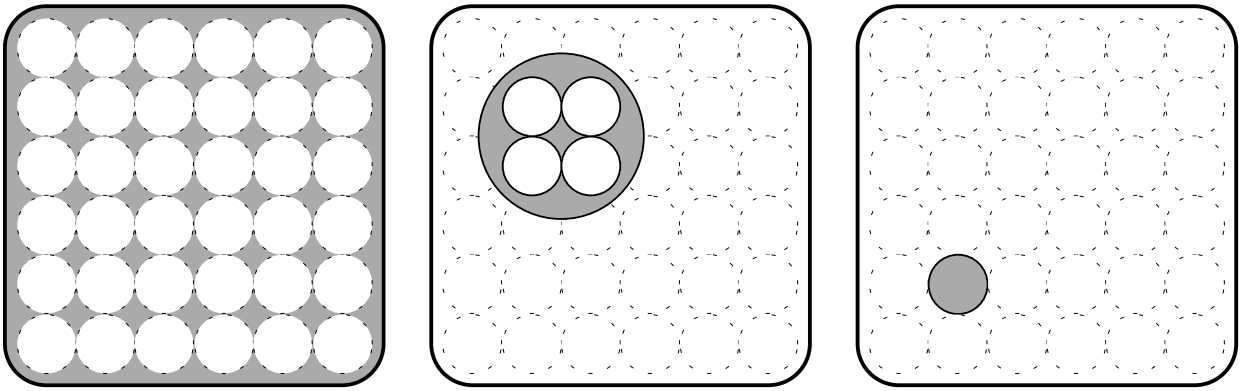


Figure 1: B-ISDN Bearer Connections

- Virtual Channel Connection: A single Virtual Channel is provided by the network as a separate entity.

- CRCG Bearer Connection: CRCGs provide a constrained differential delay service for a selected group of connections. This is accomplished by using the same route for VC connections in a CRCG from the requesting party to a requested party. Not all connections in a CRCG need to be established simultaneously. A user may wish to establish a video connection in addition to an already existing speech connection. It should be possible to indicate, that the video connection will be routed via the same route as the existing speech connection. Within Capability Set 2, the following rules apply:

  - no re-routeing is allowed when adding a party or a Network Connection Group; and
  - a CRCG may have VC connections of different connection types; however all roots of Type 2 connection must be attached at the same party;
  - a CRCG may consist of one connection only;
  - every connection belongs to only one CRCG.

- Virtual Path Bearer Connection: an entire Virtual Path is provided by the network. The Virtual Channels and their content is transparent to the network (Virtual Path Bearer Connections are not supported in Capability Set 2).

The Bearer Connection (Bearer Service) provides cell based communication in a bi-directional symmetric, bi-directional asymmetric or uni-directional fashion for various Bearer Connection topology types. Those connection topology types that are applicable for Capability Set 2 are illustrated in table 3; a more general representation showing possible future extensions is provided in annex B. The user specifies several parameters at connection establishment to characterize the communications. The Virtual Channel may carry one service component or may carry several service components multiplex together. When the Virtual Channel is used in this fashion, the method of multiplexing must be indicated to the addressed parties associated with this virtual channel at the time of connection establishment. The information rate may be characterized by a group of parameters such as peak cell rate or peak bit rate, average cell rate or bit rate, etc. When a connection is being requested, the admission control within the network will check if this connection type is allowed and then may employ "Look Ahead" procedures in order to determine if the addressed parties associated with the connection are willing to accept the connection. If the service class and the addressed parties are willing to accept the connection, the network will attempt to establish the connection. If there is sufficient physical bandwidth within the network to carry this connection, the connection will be established.

The following definitions about the ownerships capabilities will apply:

call owner: One that initiates a call is the call owner. There is only one call owner per call.

party owner: One that adds a party to a call is the owner of that party. There may be several party owners within a call.

connection owner: One that initiates a connection is the connection owner. There is only one connection owner per connection. There may be several connection owner per call. A connection owner may be associated with the root, a leaf, or be a party not attached to the connection.

branch owner: One that adds a branch to a connection is the owner of that branch. There may be several branch owners per connection. A branch owner may be associated with the root, a leaf, or be a party not attached to the connection.

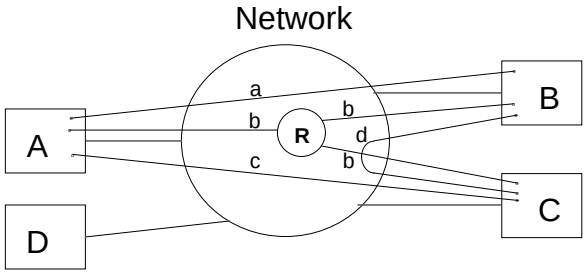
Table 2

Permission	Call Owner	Party Owner	Connection Owner	Branch Owner
Release a Call	☑			
Release Party from Call	☑	☑		
Release a Connection	☑			
Detach a Party from a Connection	☑		☑	☑

Any party has the permission to release itself from the call or to detach itself from the connection.

The transfer of ownership is not included in the Capability Set 2 requirements.

The B-ISDN services allows for the inclusion of several connections per "call." An example of a call with several connections is shown in figure 2.

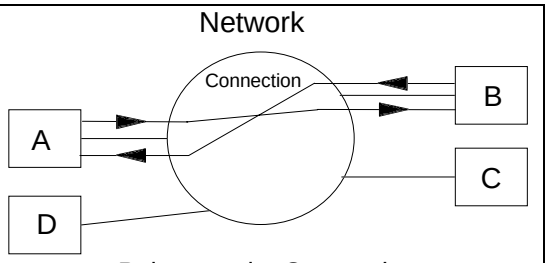
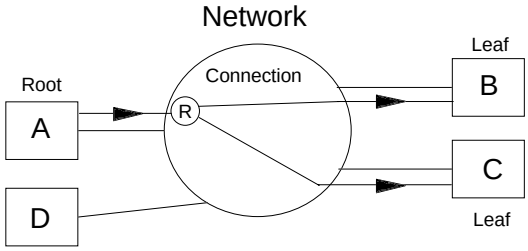


NOTE: Lines a, b, c, and d = connections.

Figure 2: Example Four Party Call containing Four Connections

- "R" is a replication function which may be implemented in multiple nodes within the network.
- Connection "a" is a Type 1 connection between Party "A" and Party "B".
- Connection "b" is a Type 2 connection between the "Root" Party "A" and the "Leaf" Parties "B" and "C".
- Connection "c" is a Type 1 connection between Party "A" and Party "C".
- Connection "d" is a Type 1 connection between Party "B" and Party "C".
- Party D is a member of the call but is not attached to any of the four connections.

Table 3: Connection Topology Types

Type 1: Point-to-point Connection - a unidirectional or bi-directional connection between two terminals. A Point-to-point Connection may provide uni-directional or Bi-directional asymmetric communications between parties "A" and "B".	 Point-to-point Connection
Type 2: Point-to-multi-point Connection - a unidirectional connection from a single source to two or more sinks (note 1). A Point-to-multi-point Connection provides uni-directional communications from the "Root" Party "A" to "Leaf" Parties "B" and "C." "R" = Replication Function (note 2).	 Uni-Directional Point-to-multi-point Connection

(continued)

Table 3 (concluded): Connection Topology Types

NOTE 1: Type 2 connections may be used to support multicast or broadcast services.
A multicast connection is one in which sink parties are specified before the connection is established, or by subsequent operations to add or remove parties from the connection. The source of the connection will always be aware of all parties to which the connection travels.

EXAMPLES:
A connection that is established to a list of end addresses.
A connection that broadcasts to a community of terminals by some criteria other than network parties.
A connection that is associated with a "distribution list" not visible to the source, such as an e-mail exploder.
A broadcast connection is one in which the sink parties are not always known to the source. The major difference from multicast is that for a broadcast connection, the connection to individual sink parties is not under the control of the source, but is by request of the each sink party.
A broadcast connection to which any terminal can subscribe without reference to the source, perhaps subject to subscription restrictions.
Broadcast connections are not required for Capability Set 2.

NOTE 2: Replication may occur in one or more network nodes to provide the multiple routes. A Replication point is a point in a connection where user plane data received from one incoming data flow is replicated on two or more outgoing data flows. Replication is possible in the ATM Layer, within the AAL or in higher layers.
Replication takes place within the ATM switch based on the VPI:VCI fields in the cell header. Each such cell arriving at the switch is copied onto one or more outgoing ATM cell streams, and within each cell stream onto one or more ATM virtual paths or virtual channels. Cell information field contents are not altered by this copying process. This form of Replication is simple to implement but is not suitable for those AALs that rely on point-to-point retransmission for recovery of lost data. ATM Replication points may occur in any node of the network. This type of Replication point is required for CS2.
Replication takes place within an AAL handler attached to or integrated with the ATM switch. Each AAL packet is assembled from incoming cells, the user information is extracted, and this is then segmented again onto two or more outgoing AAL connections. Because the AAL is terminated and regenerated in this case, retransmission is handled directly by the Replication point, not passed back to the origin of the information. This type of Replication point is **not** supported by Capability Set 2.
Higher Layer Replication takes places within a specialist server function of some kind. This type of Replication point is not supported by CS2.

Within CS2, the following general requirements apply:

1) the Network Connection owner shall be the connection initiating party;
2) the root party of a Type 2 connection is not necessarily the Network Connection owner;
3) a party that is not a connection owner can add a new party to the existing connection;
4) a third party may establish and add a Network Connection Group and attach an existing party to a Network Connection Group to which this party is not attached although it is the Network Connection Group owner;
5) it is possible for a party to have no connections;
6) when an attempt to add an additional Network Connection is made, the Network Connection will only be awarded to the terminal equipment which was awarded the first Network Connection within the call;
7) whenever type 2 connections are used, automatic answer machines are assumed;
8) when multiple parties are contained in an information flow, they are all considered mandatory;
9) the negotiation of ATM bearer connection characteristics (e.g. ATM cell rate) during the call establishment phase shall be provided;
10) a particular connection can belong to no more than one Network Connection Group at a certain point in time: A connection has one particular route through the network at a certain point of time, so it cannot belong at the same moment to multiple Network Connection Groups;
11) it should be possible to control the Network Connection Group as a whole for establishment;

12) it is necessary to be able to specify, within a single information flow, single or multiple parties, service components, service module and network connections;
13) all attached parties must be notified about type changes of the connection;
14) all serving nodes associated with parties that may invoke capabilities related to parties, connections or attachments that are modified should be notified of these modifications;
15) modification from type 1 to 2 connections are only allowed if the type 1 connection is unidirectional.

3.3 ITU-T SG 13 Proposed Schedule for B-ISDN Signalling Capabilities

Major areas of study for the initial B-ISDN access and network signalling Recommendations prepared by Study Group 13 are listed in table 4.

Table 4: Proposed Schedule for B-ISDN Signalling Capabilities

Release 1 (note 1)	Release 2 (note 2)
1 B-ISDN Bearer services (note 3)	<- as for Release 1 with additions:
BCOB-A (CBR, CO with end-to-end timing) peak traffic parameter, emulation (speech, 3,1 kHz audio & 64 kbit/s unrestricted and higher rates)	BCOB-B (VBR, CO, with end-to-end timing)
BCLB (VBR, CLS, no end-to-end timing) peak traffic parameter	BCOB-C (VBR, CO, no end-to-end timing)
BCOB-X (unrestricted, proprietary AAL) peak traffic parameter	Resource allocated according to statistical multiplexing scheme (VBR services)
Information transfer capability: unrestricted	Relationship of CLP to QoS & Indication of QoS by user
2 Network Architecture (refer I.311 par.2)	<- as for Release 1 with additions:
For both UNI and NNI	Connectionless servers (switched access)
VPC cross connect only	VP Resource management systems
VCC Switching (ATM switching)	Service Control Point (IN) access
non-intelligent multiplexing	Intelligent multiplexing (note 10)
Connectionless servers interconnected with semi-permanent VCC/VPC. Access to connectionless services with semi-permanent VCC/VPC (at UNI) (note 4)	Switched access to connectionless servers
3 Network Capabilities	<- as for Release 1 with additions:
VC switching (point-point)	Simple Multi-point VC & VP connections
VP cross connect (point-point) (note 8)	
VP establishment with OAM system (notes 5 and 6) VCC within a User-user VPC, establishment on demand	VP configuration with standard OAM CLP use
Indication of VPC and VCC peak bit-rate during establishment (note 7)	Negotiation of VPC and VCC traffic descriptor during establishment
	Re-negotiation of VPC/VCC traffic descriptor during active phase
	Indication of QoS
Uncontrolled GFC	GFC to support point-to-point star configuration
	Grouping of bearer services keeping timing relation
3.1 Traffic characteristics:	<- as for Release 1 with additions:
Peak cell rate descriptor	Source traffic parameters and network procedures for statistical multiplexing schemes
Traffic descriptor for circuit emulation including 64 kbit/s	N-ISDN interworking

(continued)

Table 4 (concluded): Proposed Schedule for B-ISDN Signalling Capabilities

Release 1 (note 1)	Release 2 (note 2)
3.2 Connection Configurations (User bearer services)	<- as for Release 1 with additions:
unidirectional, point-to-point	simple point to multipoint ATM bearers limited topologies,
bi-directional, point-to-point symmetrical asymmetrical	Capability for add/drop parties
single connection, simultaneous establishment of connections	multi-connection, delayed establishment
3.3 Connection Configurations (Signalling)	<- as for Release 1
uni-directional, point to multipoint, broadcast	
bi-directional, point-to-point symmetrical	
3.4 Interworking (note 9)	<- as for Release 1 with additions:
to narrowband ISDN using BCOB-A	for further study in Release 2 & 3
to other connectionless networks using BCLB 6	
to Frame Relay network using BCOB-X	
4 Other attributes	<- as for Release 1 with additions:
Common channel signalling transfer mode	for further study in Release 2 & 3
Meta-signalling channel	
Initial guidance on charging	Broadband aspects of charging and relationship to resource allocation
Limited supplementary services as per Q.767 (note 8)	Supplementary Services

NOTE 1: Mandatory for Release 1 Signalling Recommendations.
NOTE 2: As an objective for Release 1 Signalling Recommendations, to be included in Release 1 Recommendations where possible.
NOTE 3: This includes the support of narrowband ISDN services as defined in Q.767.
NOTE 4: No signalling impact expected, as the ATM bearer connections will be semi-permanent.
NOTE 5: The OAM system may be non-standard.
NOTE 6: Customer to VP service provider signalling relation to be considered in Release 2.
NOTE 7: A peak rate traffic parameter will be specified as a data request rate at the ATM layer SAP. An additional peak rate traffic parameter to be specified as a bit rate at the AAL SAP is for further study.
NOTE 8: Subject to minimal impact on connection configurations. Further study is required on the applicability of these supplementary services to B-ISDN services defined by SG 1.
NOTE 9: Further study is required on the bearer services for interworking.
NOTE 10: Standardization may not be required.

3.4 B-ISDN Signalling Capabilities

The Signalling Capabilities that are projected to be available in Broadband Capability Set 2 are contained in tables 5 to 22. The stage 2 requirements for this set of capabilities are contained in this Signalling Requirements release.

The term "Look Ahead", in CS2, is used to describe a procedure with two possibilities at the terminating signalling end point, named "Without State Change" and "With State Change" respectively.

The "Look Ahead Without State Change" is a procedure used to determine terminal and connection suitability before an intended call control action (e.g. call establishment or party addition).

The "Look Ahead With State Change" is a procedure used by the Network, with basically the same meaning of the previous one, but a state change is generated at the terminating signalling end point (e.g. Call and Bearer establishment or Addition of Bearer to a Call), whenever the network wishes to reserve the resources at the terminating signalling end point until the network request arrives.

In Capability Set 2, Gateway exchanges may optionally reject interrogation requests.

For both procedures, a selection mechanism for terminal capabilities will be used. This mechanism will offer sets of capabilities to the terminating end point. This end point will select one of these sets of capabilities or will refuse the call.

When reviewing the service requirements, note that some services may be realized through the combination of signalling flows and logical procedures defined for a number of other capabilities.

Table 5: Simultaneous Call and Network Connection Group Establishment

##	**Simultaneous Call and Network Connection Group Establishment** - with or without Network "Look Ahead". - a new call will be established to new parties with one new Network Connection Group.	Conn. Type
1	Establishment of a call containing a point-to-point Network Connection Group requested by a party that will be a connection endpoint.	1
2	Establishment of a call containing a point-to-multi-point Network Connection Group requested by the party that is the "root" of the Network Connection Group.	2
3	Establishment of a call containing a point-to-multi-point Network Connection Group when requested by a leaf party of the connection.	2
4	Establishment of a call containing a point-to-point Network Connection Group requested by a party that will not be a connection endpoint.	1
5	Establishment of a call containing a point-to-multi-point connection requested by a party that will not be a connection endpoint.	2

Table 6: Addition of a new Network Connection Group to an existing call with attachment of existing parties to the Network Connection Group

##	**Addition of a new Network Connection Group to an existing call with attachment of existing parties to the Network Connection Group** - with or without Network "Look Ahead". - a new Network Connection Group will be established to existing parties in an existing call, but not necessarily all parties.	Conn. Type
1	Addition of a point-to-point Network Connection Group requested by any party in the call that will be a connection end-point of the new Network Connection Group.	1
2	Addition of one point-to-multi-point Network Connection Group requested by the "root" of the new Network Connection Group.	2
3	Addition of a point-to-multi-point Network Connection Group when requested by a leaf party of the Network Connection Group.	2
4	Addition of a point-to-point Network Connection Group requested by a party that will not be a connection endpoint.	1
5	Addition of a point-to-multi-point Network Connection Group requested by a party that will not be a connection endpoint.	2

Table 7: Addition of a one Network Connection to an existing Network Connection Group with attachment of existing parties to the new Network Connection

##	**Addition of a one Network Connection to an existing Network Connection Group with attachment of existing parties to the new Network Connection** - with or without Network "Look Ahead". - a new Network Connection will be established to existing parties in an existing call, but not necessarily all parties.	Conn. Type
1	Addition of a point-to-point Network Connection requested by any party in the call that will be a Network Connection end-point of the new Network Connection.	1
2	Addition of one point-to-multi-point Network Connection requested by the "root" of the new Network Connection.	2
3	Addition of a point-to-multi-point Network Connection when requested by a leaf party of the Network Connection.	2
4	Addition of a point-to-point Network Connection requested by a party that will not be a connection endpoint.	1
5	Addition of a point-to-multi-point Network Connection requested by a party that will not be a connection endpoint.	2

Table 8: Addition of one or more new parties to an existing call with attachment to an existing Network Connection

##	**Addition of one or more new parties to an existing call with attachment to an existing Network Connection** - with or without Network "Look Ahead".	Conn. Type
1	Add one or more new parties requested by an endpoint of the original type 1 connection.	1->2
2	Add one or more new parties when requested by the root party of the Network Connection.	2
3	Add one or more new parties when requested by a leaf party of the Network Connection.	2
4	Add one or more new parties when requested by a party that will not be a connection endpoint.	1
5	Add one or more new parties when requested by a party that will not be a connection endpoint.	1->2

Table 9: Attachment of one or more existing parties to an existing Network Connection

##	**Attachment of one or more existing parties to an existing Network Connection** - with or without Network "Look Ahead".	Conn. Type
1	Attach one or more existing parties to an existing point-to-point Network Connection requested by an endpoint of the original type 1 connection.	1->2
2	Attachment of one or more existing parties to an existing point-to-multi-point Network Connection requested by the root party of the connection.	2
3	Attachment of one or more existing parties to an exiting point-to-multi-point Network Connection when requested by a leaf party of the connection.	2
4	Attachment of one or more existing parties to an exiting point-to-point Network Connection when requested by a party that will not be a connection endpoint.	1->2
5	Attachment of one or more existing parties to an exiting point-to-multi-point Network Connection when requested by a party that will not be a connection endpoint.	2

Table 10: Addition of one or more new parties and new Network Connection Group to an existing call with attachment of these new parties to the new Network Connection Group

##	**Addition of one or more new parties and new Network Connection Group to an existing call with attachment of these new parties to the new Network Connection Group** - with or without Network "Look Ahead".	Conn. Type
1	Add a new party and a new point-to-point Network Connection Group to the call, requested by any party of the existing call that will become a Network Connection Group endpoint of the new Network Connection Group:	1
2	Add two or more new parties and a new point-to-multi-point Network Connection Group, requested by any party that will become the root of the new Network Connection Group.	2
3	Add two or more new parties and a new point-to-multi-point Network Connection Group, requested by any party that will become the leaf of the new Network Connection Group.	2
4	Add two parties and a new Network Connection Group to the call when requested by a party that will not be a connection endpoint of the new Network Connection group.	1
5	Add three or more new parties and a new point-to-multi-point Network Connection Group when requested by a party that will not be a connection endpoint of the new Network Connection group.	2

Table 11: Addition of one or more new parties and a new Network Connection to an existing Network Connection Group of an existing call with attachment of the new parties to the new Network Connection

##	**Addition of one or more new parties and a new Network Connection to an existing Network Connection Group of an existing call with attachment of the new parties to the new Network Connection** - with or without Network "Look Ahead".	Conn. Type
1	Add a new party and a new point-to-point Network Connection to the call, requested by any party of the existing call that will become a Network Connection endpoint of the new Network Connection.	1
2	Add two or more new parties and a new point-to-multi-point Network Connection, requested by any party that will become the root of the new Network Connection.	2
3	Add two or more new parties and a new point-to-multi-point Network Connection, requested by any party that will become the leaf of the new Network Connection.	2
4	Add two parties and a new Network Connection to the call when requested by a party that will not be a connection endpoint of the new Network Connection.	1
5	Add three or more new parties and a new point-to-multi-point Network Connection when requested by a party that will not be a connection endpoint of the new Network Connection.	2

Table 12: Common Actions for Capabilities in tables 13, 14 and 15

When an action (automatic or requested by a party)...	**Then the following action occurs automatically...**
- detaches one end of a Type 1 Network Connection	the Network Connection is released
- leaves a Party with no Network Connections in Capability Set 2	the Party remains in the call
- leaves only one Party in a Call	the Call is released
- releases the owner of a Call	the Call is released
- releases the owner of one or more Network Connections	those Connections are released
- releases the owner of a Party	the Party is released
- releases the owner of a Branch	the Branch is released
- releases all Network Connections in a Network Connection Group	the Network Connection Group is released
- detaches the root of a Type 2 Network Connection	the Network Connection is released
- detaches a leaf party that is the Branch owner	the Branch he owns will not be released
- detaches a leaf party that is the owner of a Type 2 Network Connection	the modified Network Connection is not released
- detaches all leaf parties of a Type 2 Network Connection	the Network Connection is released
- detaches all but the root party and one leaf party of a Type 2 Network Connection	the connection is converted to a Type 1 Network Connection

Table 13: Detach one party from an existing Network Connection

##	**Detach one party from an existing Network Connection** See table 12	Conn. Type
1	Detach a party from an existing Network Connection when requested by the party itself	1&2
2	Detach a party from an existing Network Connection when requested by the call owner,	1&2
3	Detach a party from an existing Network Connection when requested by the connection owner.	1&2
4	Detach a party from an existing Network Connection when requested by the branch owner.	1&2

Table 14: Release of a Network Connection from an existing call

##	**Release of a Network Connection from an existing call** See table 12	Conn. Type
1	Release a Network Connection from an existing call requested by the call owner	1&2
2	Release a Network Connection from an existing call requested by the connection owner	1&2

Table 15: Release of a party from an existing call

##	**Release of a party from an existing call** See table 12	Conn. Type
1	Release a party from an existing call when requested by the call owner	1&2
2	Release a party from an existing call when requested by the party itself	1&2
3	Release a party from an existing call when requested by the party owner	1&2

Table 16: Release of a call

##	**Release of a call** - the call, all parties and all Network Connections or Network Connection Groups involved in the call, will be released.	Conn Type
1	Release an existing call requested by the call owner party	1&2

Table 17: Notification that parties have been added or released from the call

##	**Notification that parties have been added or released from the call**	Conn. Type
1	notify all parties that have ownership related to the affected parties	
2	notify all serving nodes associated with the parties that have special permission to invoke signalling capabilities	
3	notify a subset of the nodes associated with the parties that have special permission to invoke signalling capabilities	

Table 18: Notification that a Network Connection has been added, released or modified

##	**Notification that a Network Connection(s) has been added, released or modified**	Conn. Type
1	notify all parties that have ownership related to the affected connection(s)	
2	notify all serving nodes associated with the connection(s) that have special permission to invoke signalling capabilities	
3	notify a subset of the nodes associated with the connection(s) that have special permission to invoke signalling capabilities	

Table 19: Notification that a party has been attached or detached to or from a Network Connection

##	**Notification that a party has been attached or detached to or from a Network Connection**	Conn. Type
1	notify all parties that have ownership related to the affected parties	
2	notify all serving nodes associated with the parties that have special permission to invoke signalling capabilities	
3	notify a subset of the nodes associated with the parties that have special permission to invoke signalling capabilities	

Table 20: Renegotiation and modification of calling party/called party Network Connection characteristics during active phase

##	**Renegotiation and modification of calling party / called party Network Connection characteristics during active phase** - This capability is applicable to point-to-point and point-to-multipoint Network Connections. - This capability is invoked for a single Network Connection when it is in the active phase. - The changes to the traffic characteristics of a connection can either increase or decrease the traffic parameter values of the Network Connection. - For a point-to-point Network Connection, one of the two parties can request this capability. If the network or the other party rejects the request, the traffic parameter values in use prior to the invocation of the capability are still applicable. - For a point-to-multipoint Network Connection, only the root of the connection can request this capability. If the network or one of the leaves rejects the request, the traffic parameter values in use prior to the invocation of the capability are still applicable.	Conn. Type
1	Re-Negotiation of Network Connection characteristics associated with Network Connection without re-routeing and modification	1
2	Re-Negotiation of Network Connection characteristics associated with Network Connection without re-routeing and modification	2

Table 21: Call establishment without any Network Connections

##	**Call establishment without any Network Connections**	Conn. Type
1	Create two or more party call without Network Connections	

Table 22: Addition of a party to an existing call

##	**Addition of a party to an existing call**	Conn. Type
1	Add one or more new parties to an existing call requested by any party already associated with that call.	

4 Information model

4.1 General

The use of object-oriented philosophy is intended to help structure the information in the call-related information flows. This subclause addresses the information modelling aspects of such flows. It introduces the object types and clarifies the way that these objects are related in a telecommunication service.

4.1.1 Object Modelling and Signalling Requirements

Signalling requirements should guide the specification of protocol and procedures associated with the interoperation of independently designed systems to provide telecommunication services. Information flows and SDL have been used effectively in the past to define the requirements of relatively simple services. Traditional signalling requirements enhanced these components with descriptions of the parameters to be included within the information flows. The complexity and multiplicity of services which are enabled by Capability Set 2 make it difficult, if not impossible, to provide an exhaustive collection of information flows for each possible service. It is not only difficult to provide, it is difficult to use the results in Stage 3 specification of signalling protocol and procedures because the design of an efficient protocol first necessitates the reverse engineering of the flows to gain an understanding of the fundamental relationships of the information which is being exchanged and a conceptualization of the entities about which signalling is concerned.

Object modelling provides an opportunity to adjust the traditional methods to be responsive to the pressures to rapidly specify improved signalling protocols. This approach is useful because many of the capabilities of Capability Set 2 can be modelled with a small number of objects. Furthermore, object modelling partitions the complexity of the Stage 2 specification, which simplifies the Stage 2 specification without reducing the feature content of Capability Set 2. Lastly, many of the Stage 2 information flows for components of Capability Set 2 result in functional entity actions that can be modelled on creation and deletion of objects. By defining the creation and deletion actions (information flows and SDLs) for a small number of object classes, insight into a large number of capabilities has been accomplished.

Objects are defined by their behaviour, relationships to other objects and their attributes (referred to as parameters within signalling). Signalling protocols define the actions and attributes across an interface needed to provide a service. Defining the details of the call model objects will result in the definition of signalling protocol information element exchanges and actions needed to support the interactions between call control functional entities.

The information flows contain parameters needed to specify the objects in call and bearer interactions. Information flows cause objects to be created or deleted. The specific objects to be affected, and what operations are to be performed, are determined from the actions and parameters specified in the information flows.

All objects have a state field which reflects transient and stable states. The possible states are: creation-pending, modification pending and stable. A deleted object (e.g., after a release or detach) does not exist; therefore, there is no "deleted" state.

4.1.2 Views of a Call

The objects in the information model are used as generic building blocks to form telecommunications services and to represent calls. The information model is described from the local perspective. The *local view* is the picture of a telecommunication service as perceived by the local signalling endpoint. The local perspective is directly associated with the information flows that serve as the input to stage 3 and will be the only perspective used in clause 3 and beyond this ETR.

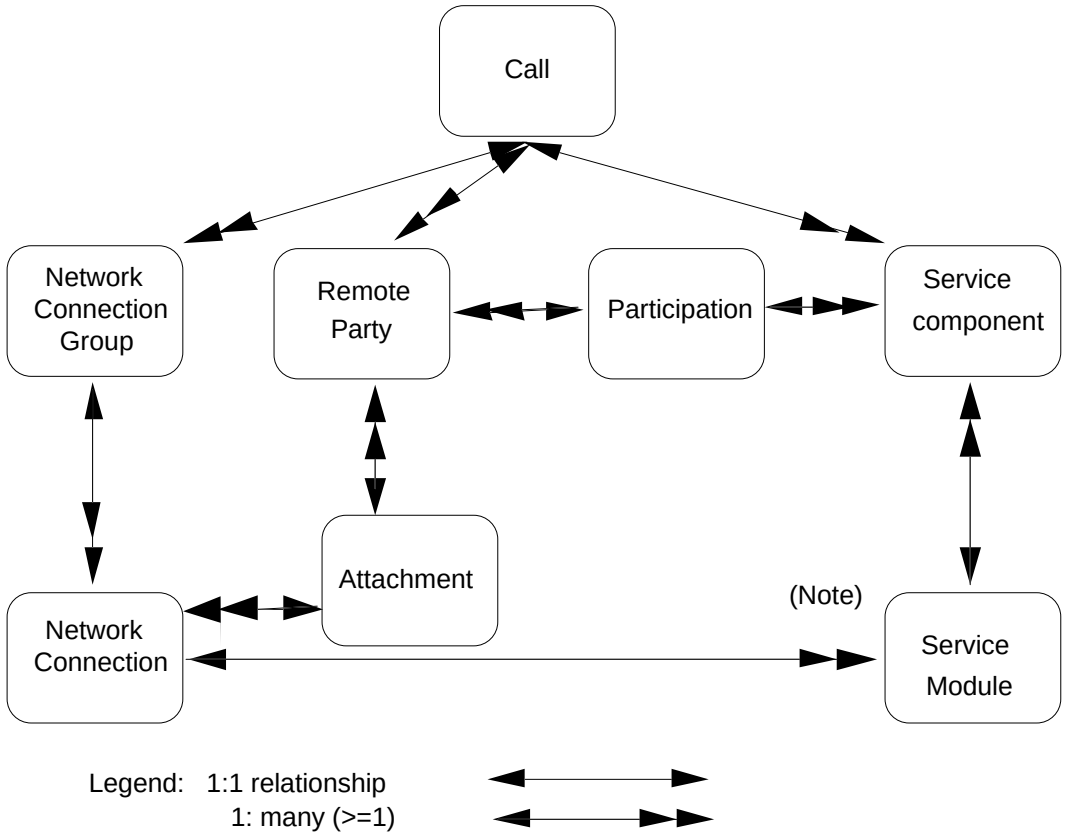
4.2 Description of Objects in Information Model

Table 23 describes the objects used as the elemental call objects to form telecommunication services and to establish a call. The attributes, which define the objects, are described in subclause 4.3.

Table 23: Elemental Call Objects

Object	Attributes of Object	Description of Object
Call	Call ID, List of Remote Parties, Local Party Address, Telecommunications Service Type, List of Service Components, List of Network Connection Groups	See subclause 4.2.1
Remote Party	Remote Party Reference ID, Remote Party Address, Type of Remote Party	See subclause 4.2.2
Attachment	cross reference of Remote Party and Network Connection	See subclause 4.2.3
Service Module	Service Module ID List of Service Components, Multiplexing Method	See subclause 4.2.4
Service Component	Service Component ID, Service Component Characteristics (incl. High Layer Information), Service Traffic descriptor requirements, Service QoS descriptor requirements	See subclause 4.2.5
Network Connection	Network Connection ID, List of Service Modules, User (TE)-Oriented Low Layer Information, Network-Oriented Low Layer Information, Directionality Indicator, Transit Network ID, ATM Connection Topology Type, Local Attachment	See subclause 4.2.6
Network Connection Group	Network Connection Group ID, list of Network Connections,	See subclause 4.2.7
Participation	cross reference of Remote Party and Service Component.	See subclause 4.2.8

Figure 3 shows the local view of the Basic Information Model in which a set of objects represents a call invoking a multimedia service with multiple parties and using multiple connections.



NOTE: The relationships between Service Modules and other object types reflects service requirements of ITU-T SGs 1 and 13 that allow AAL multiplexing (e.g., through the MID field of AAL Type 3/4) [the Network Connection relationship] and media multiplexing (e.g., in accordance with H.261) [at the Service Component relationship].

Figure 3: Basic Information Model

4.2.1 Call

This object is defined as the local view of an association of one or more parties using a telecommunication service to communicate through the network(s). This object also provides for the logic that is required for the interaction ("binding") of Service Component objects.

4.2.2 Remote Party

A party is an addressable signalling endpoint associated with a call. It may be a user or a service provider.

4.2.3 Attachment

This object provides the local view of which remote parties are associated with which Network Connections.

4.2.4 Service Module

A Service Module multiplexes one or more Service Components (one medium specified in a Service Component), and it specifies a multiplexing method for all Service Components within Service Module.

4.2.5 Service Component

A Service Component defines a single aspect of a telecommunication service related to a single medium such as voice, data, video, etc.

4.2.6 Network Connection

This object represents an AAL connection on top of an ATM connection, which could be a Type 1 connection topology or a Type 2 connection topology defined in table 3.

4.2.7 Network Connection Group

This object represents a set of Network Connections that have common network characteristics, such as a CRCG, or a Life and Death Connection Group.

4.2.8 Participation

This object resolves the many-to-many relationship between Remote Party type objects and Service Component type objects through a cross reference of Remote Parties and Service Components.

4.3 Descriptions of Attributes

4.3.1 Attributes of Call Objects

4.3.1.1 Call Identifier (Call ID)

The Call Identifier is used by all signalling entities to reference the instance of call service to which this signalling control information flow pertains.

4.3.1.2 List of Remote Parties

The Party Reference Identifiers are used to identify the parties in a call. At least more than one party exist in a call.

4.3.1.3 Local Party Address

The Local Party Address is to specify the directory number of the local party. This consists of the Local Party Number and the Local Party Subaddress.

4.3.1.4 Telecommunications Service Type

This attribute identifies a specific telecommunication service or a user-defined service.

4.3.1.5 List of Service Components

This attribute identifies each Service Component in a Service Module. One or more Service Components can be specified in a Service Module.

4.3.1.6 List Network Connection Groups

This attribute is a list of the Network Connection Groups within the call.

4.3.2 Attributes of Remote Party Objects

4.3.2.1 Remote Party Reference Identifier

The Remote Party Reference Identifier is used to identify a party in a call.

4.3.2.2 Remote Party Address

The Remote Party Address is to specify the directory number of the remote party. This consists of the Remote Party Number and the Remote Party Subaddress.

4.3.2.3 Remote Party Type

The Remote Party Type indicates the type of a party in a call, such as the call initiating party, the receiving-only party, the call owner, the network connection owner, etc.

4.3.3 Attributes of Attachment Objects

Attachment objects have a single attribute which is a cross reference of the Remote Parties and the Network Connections.

4.3.4 Attributes of Service Module Objects

4.3.4.1 Service Module Identifier

The Service Module Identifier is used by Edge-to-Edge or End-to-End Control entities to reference a virtual user plane Service Module connection between the Requesting Party and the Requested Party. A Service Module will be mapped into a single Network Connection in a call.

4.3.4.2 List of Service Components

This attribute identifies each Service Component in a Service Module. One or more Service Components could be specified in a Service Module.

4.3.4.3 Multiplexing Method

A multiplexing method may be specified to be used to multiplex all service components in the Service Module. Examples of multiplexing methods are H.221 and H.22x.

4.3.5 Attributes of Service Component Objects

4.3.5.1 Service Component Identifier

The Service Component Identifier is used to identify a Service Component in a call.

4.3.5.2 Service Component Characteristics

The Service Component Characteristics indicate type of media (e.g., voice, video, data, etc.), information transfer rate, symmetry, and media characteristics. This information allows End Control entities to determine if sufficient bandwidth and facilities exist to accept the Service Component. These characteristics include High Layer Information. High Layer Information provides the characteristics of the layers above the "Network Layer" information associated with the media characteristics. Examples of such information are Telephony, Facsimile Group 4, documentation Profiles (T.501, T.502, T.503, and T.504), Teletex, Telex, Message Handling Systems (X.400 Series), OSI Applications (X.200 Series), Video Broadcast, Video Telephony, etc. This attribute can be used by the Requesting Party for compatibility checking.

4.3.5.3 Service Traffic Descriptor Requirements

This attribute describes the service traffic requirements, such as bandwidth.

4.3.5.4 Service QoS Descriptor Requirements

This attribute describes the Quality of Service (QoS) requirements of the service.

4.3.6 Attributes of Network Connection Objects

4.3.6.1 Network Connection Identifier

This attribute is used to identify the bearer connection uniquely within the call. A bearer connection is mapped into one ATM connection.

4.3.6.2 List of Service Modules

This attribute identifies the Service Modules mapped to a Network Connection within a call.

4.3.6.3 User (TE) - Oriented Low Layer Information

This set of information specifies protocols and characteristics for the end-user's equipment of the network layer and below for a user connection to support particular services/applications.

4.3.6.3.1 Network Layer Information

Network layer information for the user terminal could be optionally present for certain services/applications as needed. This attribute specifies the network layer entity as well as user information layer 3 protocol (e.g., X.25, SMDS, etc.). This network layer information supports the high layer characteristics attribute specified in the service component. This attribute may also specify user information layer 2 protocol (Q.921/922, LAPB, etc.) above the AAL layer if needed.

4.3.6.3.2 AAL Layer Information

This attribute specifies the AAL layer characteristics at the user terminal, such as type of AAL, multiplexing method for multiple Service Modules, and control parameters associated with the particular AAL.

4.3.6.4 Network-Oriented Low Layer Information

This set of information specifies protocols and characteristics for the network of the AAL layer and below for a network connection to support particular services/applications.

4.3.6.4.1 Network Layer Information

Network layer information for the network could be optionally present for certain services/applications as needed. This attribute specifies the network layer entity as well as user information layer 3 protocol (e.g., X.25, SMDS, etc.). This attribute may also specify user information layer 2 protocol (Q.921/922, LAPB, etc.) above the AAL layer if needed. Network-oriented network layer information will always be equal to or a subset of user-oriented network layer information.

4.3.6.4.2 AAL Layer Information

This attribute specifies the AAL layer characteristics in the network, such as type of AAL, multiplexing method for multiple Service Modules, and control parameters associated with the particular AAL. (This attribute is empty if Bearer Service is Class X).

4.3.6.4.3 ATM Layer Information

This attribute indicates the ATM layer characteristics for the network, such as traffic descriptor.

4.3.6.4.4 Bearer Service Subcategory

This attribute indicates a bearer service classes, such as Class X. or Class A, which is the same as defined in F.811.

4.3.6.5 Directionality Indicator

This attribute indicates whether the endpoint equipment is the source of the user plane information, the sink, or both.

4.3.6.6 Transit Network Identifier

The Transit Network Identifier is used by Link Control entities to reference the transit network to be used in the execution of the Link Control action. This attribute may be repeated within the information flow.

4.3.6.7 ATM Connection Topology Type

This attribute defines ATM connection topology types for an ATM connection that is associated with the bearer service. The connection topology type would be one of those defined in table 3, such as point-to-point or point-to-multipoint.

4.3.6.8 Local Attachment

Local attachment indicates if the Local Party is attached/not attached when it sets up a network connection in which it is not an endpoint.

4.3.7 Attributes of Network Connection Group Objects

4.3.7.1 Network Connection Group Identifier

This attribute identifies a Network Connection Group.

4.3.7.2 List of Network Connection Identifiers

This attribute identifies each Network Connection in the Network Connection Group. One or more Network Connections will be specified in a Network Connection Group. All Network Connections in a Network Connection Group will be routed together.

4.3.8 Attributes of Participation Objects

Participation objects have a single attribute which is a cross reference of the Remote Parties and the Service Components.

5 Functional signalling model

5.1 Description of the Stage 2 Functional Model

The functional signalling model, illustrated in figure 4, that aids in the description of the target signalling requirements for B-ISDN Capability Set 2 is based on the separation of the Control Plane into several signalling functional entities. Co-ordination among these signalling functional entities is performed by Application Control (AC) functions.

NOTE: The discussion of the AC in this ETR treats the AC as if it were a single FE. This is not intended to preclude the future partitioning of the AC into several FEs. This latter approach is essential for the interworking of IN and Broadband and for the incorporation of TMN compatible layering principles in the architecture of B-ISDN.

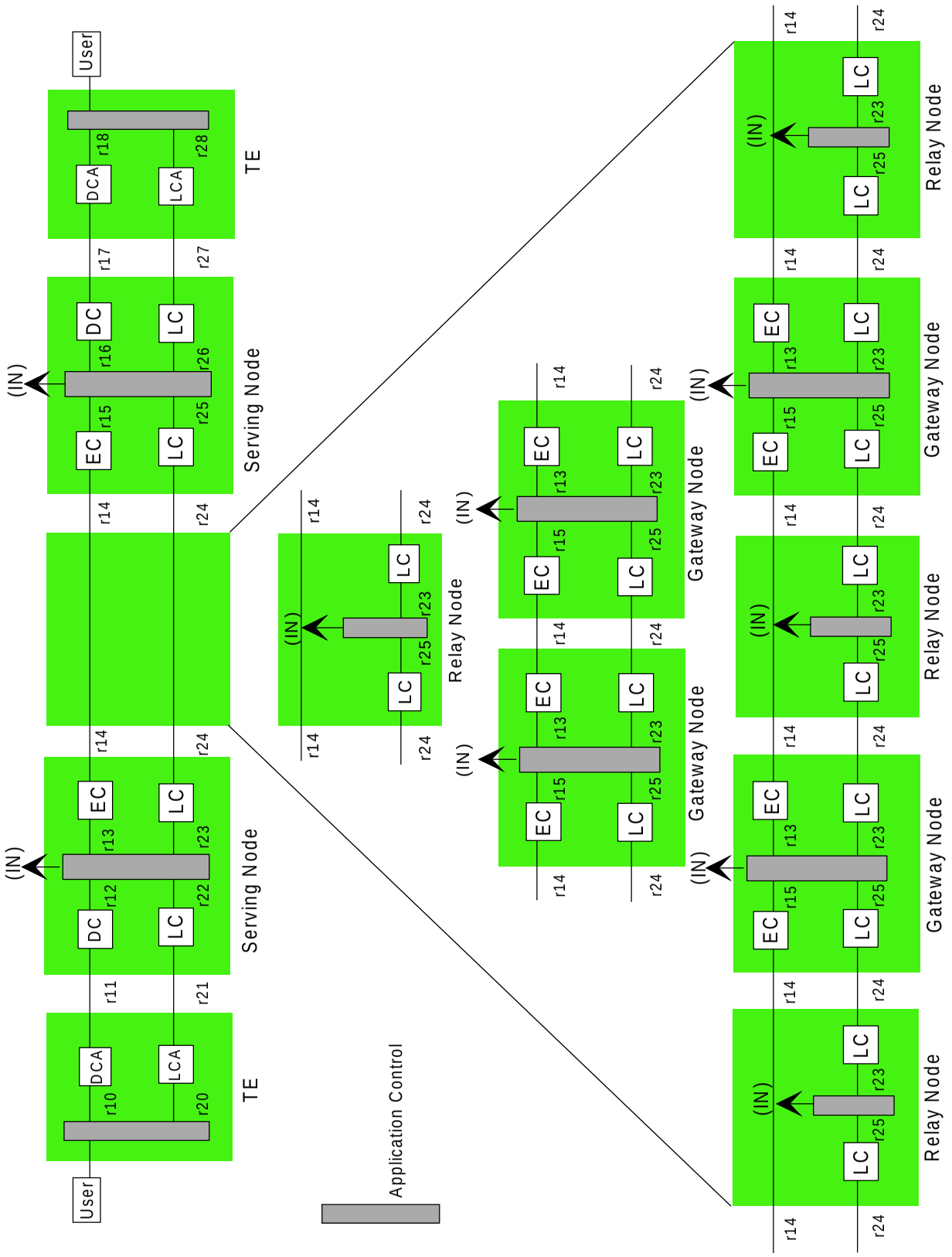
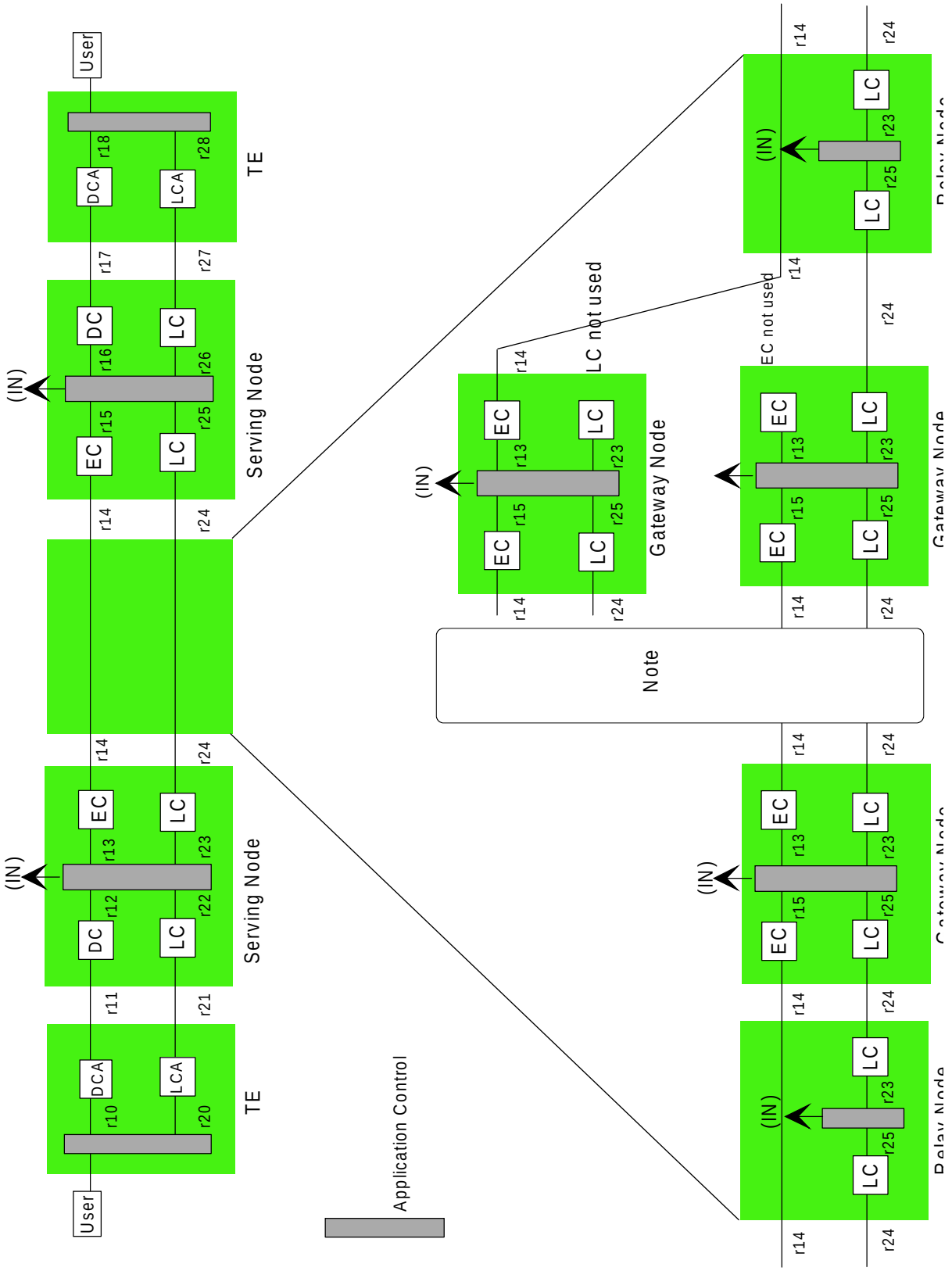
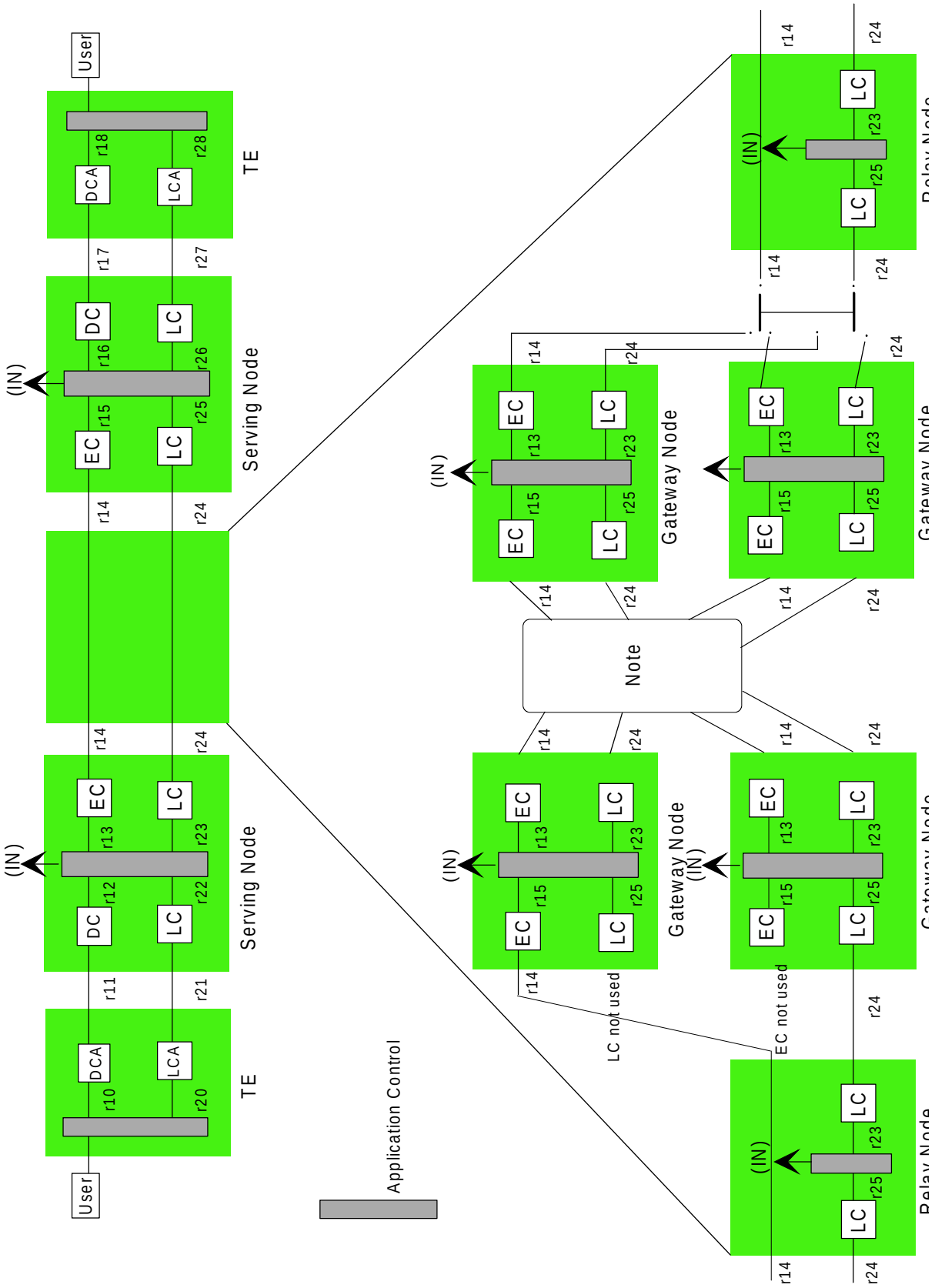


Figure 4 (1 of 3): Q.73 Functional Model for B-ISDN Capability Set 2



NOTE: Several interconnection scenarios are possible as long as:
r14 must be connected to r14 and
r24 must be connected to r24

Figure 4 (2 of 3): Q.73 Functional Model for B-ISDN Capability Set 2



NOTE: Several interconnection scenarios are possible as long as:
r14 must be connected to r14 and
r24 must be connected to r24

Figure 4 (3 of 3): Q.73 Functional Model for B-ISDN Capability Set 2

5.2 Description of Functional Entities

The functions performed by each functional entity within the functional signalling model are contained in the following tables.

Table 24

Functions within application control functional entity
- Call Control Functions 1) Overall Control of the "call" - selection of the Edge signalling controls to be used in the Edge/End signalling transactions - selection of the Link signalling controls to be used in Link signalling transactions - selection of action identifiers and call identifiers - maintenance of the state of the "call" 2) Service admission control - telecommunication service screening - user authentication, authorization and compatibility checking - service component and end to end QoS selection - user status checking and monitoring - functions for interworking with existing networks 3) Communications with Connection Control - give order to setup and release network resources associated with connections - provides co-ordination between connection controls - preparation of consolidated traffic reports 4) Charging information - collection and delivery of charging information
- Connection Control Functions 1) Overall control of the connection - trigger setup/release of the association(s) between two LC entities - maintenance of the state of the connection - selection of the connection identifier - network protection timers 2) Control network resources - reservation/release node resources - allocation/deallocation of node resources - node resource selection and negotiation - node resource interworking with existing networks - functions specific to the special resource handling of connection branches for multipoint connections and services - through connect/disconnect of connections 3) Collection of traffic count information
- Gateway Only Functions These functions are in addition to those for simple transit: - ensure that the routeing relevant parameters are set - significant digits in the called party number may be amended or omitted - include / delete the originating ISC point code parameter - protocol conversion, (e.g., national protocol to gateway protocol) - special arrangements for through connection (e.g. for fraud protection) - transit network selection - charging and accounting functions - bearer service interworking functions (note 1) - routeing and forwarding of Edge Control information (note 2) - Supplementary service specific functions
- Communications with Link and Edge/End Functional Entities
- Communications with Intelligent Network Functional Entities
NOTE 1: The inclusion and activation of the identified bearer service interworking of the Link Control entity are based on pre-determined characteristics of the network and the requested service. An example is a µ-law/A-law converter at a gateway exchange, whose enabling is based only on the country specific coding standard. Interworking functions that are controlled by the user on a per call or subscription basis are not identified explicitly.
NOTE 2: Link Control and Edge Control can be located in different gateways. The Edge Control entity at a gateway exchange may act as a "relay" for the Edge Control information flows. The routeing and forwarding function of the gateway Edge Control is identified for this purpose.

Table 25

Functions of link control functional entity
- Controls the link control association between two Link control functional entities
- Transport of Application Control information between two adjacent Application Control Functional entities
- Network protection timers
- Supplementary service specific functions
- Communications with the Application Control Functional entity

Table 26

Functions of edge control functional entity
- Controls the edge control association between two Edge control functional entities
- Transports Application Control information between two Application Control Functional entities located at the edge of the "network"
- Setup/Release of the information association between two Edge Control functional entities
- Transport of Application Control information between two Application Controls Functional Entities located at the edge of the "network"
- Network protection timers
- Supplementary service specific functions
- Communications with the Application Control Functional entity

5.3 Information Flow Strata for Capability Set 2

The functional signalling model discussed in subclause 3.1 allows a representation of the signalling requirements for B-ISDN which demonstrates a separation of the Control Plane into several Information Flow Strata. Co-ordination between these Strata is performed by an Application Control (AC) function.

The definitions of the Strata are given below:

- **End Control/Edge Control Stratum:** This stratum consist of two sections:

  - the section between the user equipment and the serving node;
  - within the network, the section between serving nodes or between a serving node and a gateway nodes.

  This stratum has been added in order to provide the signalling capabilities that are beyond Release 1 of B-ISDN. The definition of these two sections are as follows:

  - **End Control:** request/response information flows between a user equipment and it's serving network node (**DCA** = en**D** **C**ontrol **A**gent; **DC** = en**D** **C**ontrol);
  - **Edge Control**: request/response information flows between serving nodes, between serving nodes and gateway nodes, and between gateway nodes (**EC** = **E**dge **C**ontrol).

- **Link Control:** request/response information flows between the user equipment and the serving node, between the serving node and the relay node, between relay nodes, between relay nodes and gateway nodes, between gateway nodes, between serving node and gateways if relay nodes are not required, and between serving nodes if both relay nodes and gateway nodes are not required (**LCA** = **L**ink **C**ontrol **A**gent; **LC** = **L**ink **C**ontrol).

6 Information flow elements

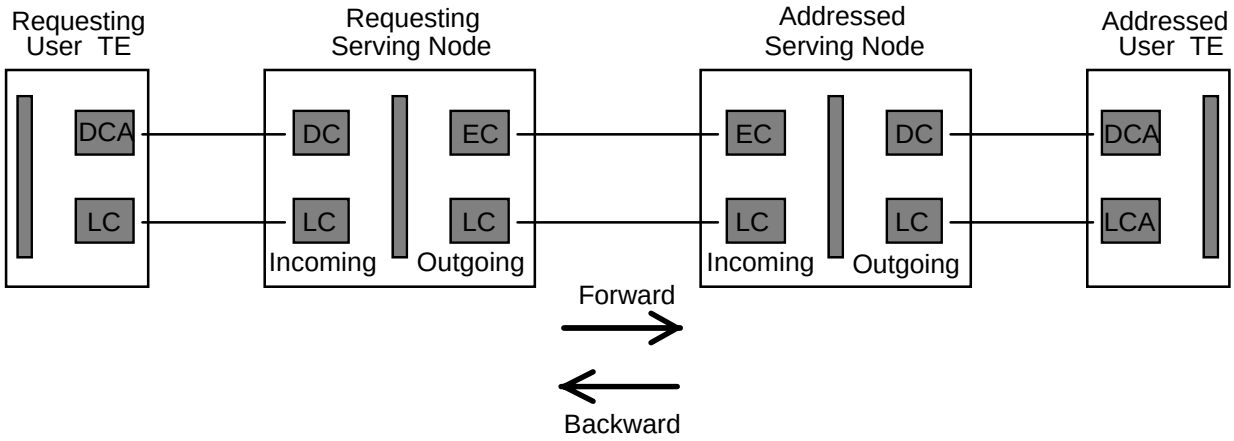
6.1 Information Flows and FE actions for Release 1 services

The information flows and functional entity actions are covered in Recommendation Q.71.

6.2 Information Flow Conventions for Capability Set 2 services

The drawing conventions associated with User Plane and Control Plane reference direction, ATM switch through and the symbols used to describe functional entity actions during connection establishment are contained in the following figures.

6.2.1 Convention of Reference Direction



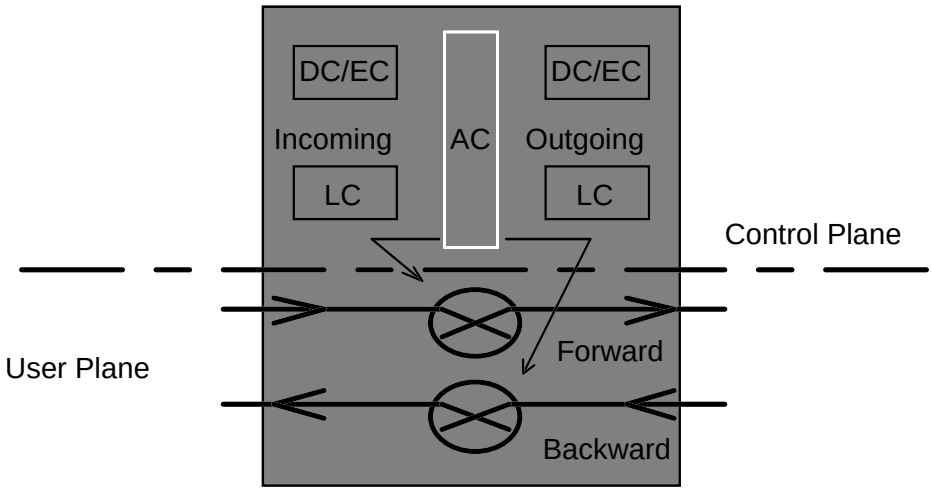
- Requesting Serving Node is associated with the user terminal requesting a signalling capability.
- Addressed Serving Node is associated with the user terminal addressed by the requested signalling capability.
- Incoming side of a Node is the side closest to the user terminal requesting a signalling capability.
- Outgoing side of a node is the side closest to the user terminal addressed by the requested signalling capability.
- Forward direction is from the requesting user terminal to the addressed user terminal.
- Backward direction is from the addressed user terminal to the user terminal requesting a signalling capability.

Figure 5: Reference Direction Convention

The reference to the incoming side and outgoing side is independent of the direction of transport of user data. When a call establishment signalling action has been requested, then the Requesting User TE becomes the Originating User TE and the Requesting Serving Node becomes the Originating Serving Node. In the same instance, the Addressed User TE becomes the Terminating User and the Addressed Serving Node becomes the Terminating Serving Node. However, when other signalling capabilities are invoked, the above convention is more useful since these actions could be invoked by a party that did not originate the call.

6.2.2 Convention for ATM Switch Through

The following figure illustrates the switch convention to be used in the handling of ATM connections within a network node.



- The "AC" is responsible for bandwidth reservation, bandwidth allocation, connection switch through, connection release, and bandwidth release in the forward and backward direction.

Figure 6: Convention for ATM switch through

6.3 Description of Information Flows

The terminology for information flows to be used in the Stage 2 representation of the Signalling Capability Sets are covered in this subclause. The signalling information flows are based on the two "Generic" information flow sequences illustrated below.

It is useful to conceive for the exchange of information between peer functional entities in terms of both one-phase and two phase interactions. The one phase mechanism is analogous to the two message request/response approach, while the two-phase is analogous to the CCR begin/ready/commit. Both of these mechanisms can be combined into a single one using CCR terminology as shown below.

All peer-to-peer exchanges are defined in terms of a superior/subordinate relationship according to the following definitions:

superior: The entity that ultimately decides on the commitment.

subordinate: The peer of the superior.

The type A request (one phase mechanism) consists of the following exchange:

The requesting functional entity, acting as a subordinate, signals its peer with X.Ready—where X represents the action to be performed. If the requested operation is successful, the superior functional entity shall complete the requested action and notify the subordinate functional entity that the action has been completed with X.Commit. If the action can not be completed, the superior functional entity will notify the subordinate functional entity that the action was not completed with X.Cancel. This is shown in the information flow in figure 7 below:

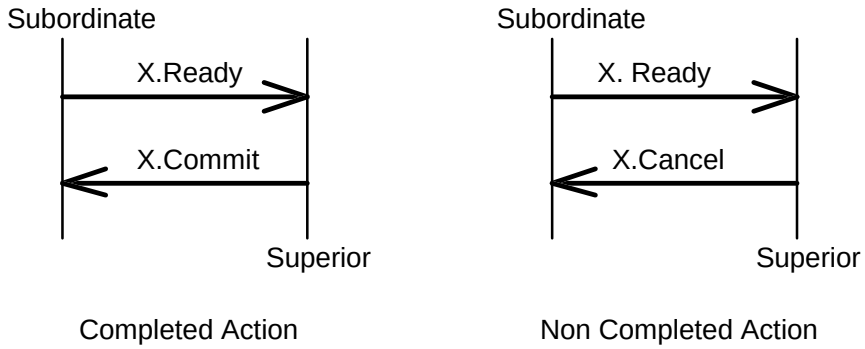


Figure 7: Information flows for single-phase request mechanism

The type B request (two phase mechanism) consists of the following exchange:

The requesting functional entity, acting as a superior, signals its peer with X.Begin. If the request will be able to be completed, the subordinate functional entity shall notify the superior functional entity of this condition with X.Ready and wait for further instructions from the superior functional entity before the requested action is completed. The superior may respond with either a flow that indicates that the action shall be completed (X.Commit) or with a flow that indicates that the receiver shall disregard the requested action (X.Cancel). If the requested operation is not successful, the subordinate functional entity shall notify the superior functional entity with X.Cancel. These flows are illustrated below:

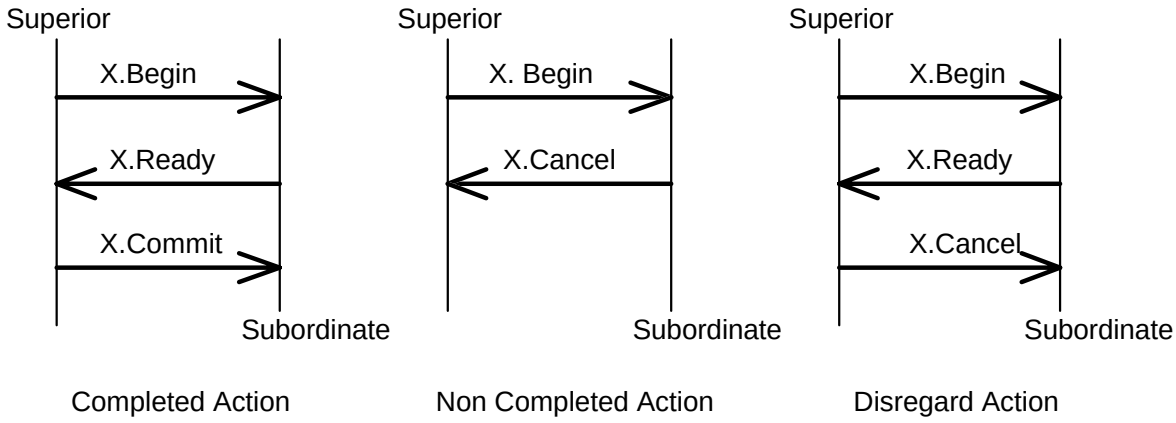


Figure 8: Information Flows for two-phase request mechanism

The following figures illustrate where the one phase and two phase mechanism are used.

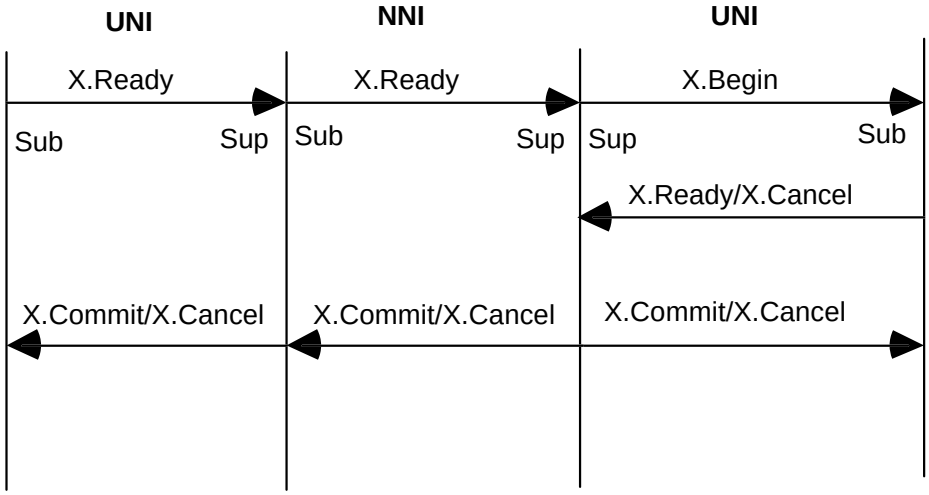


Figure 9: UNI/NNI flows with one phase request on NNI

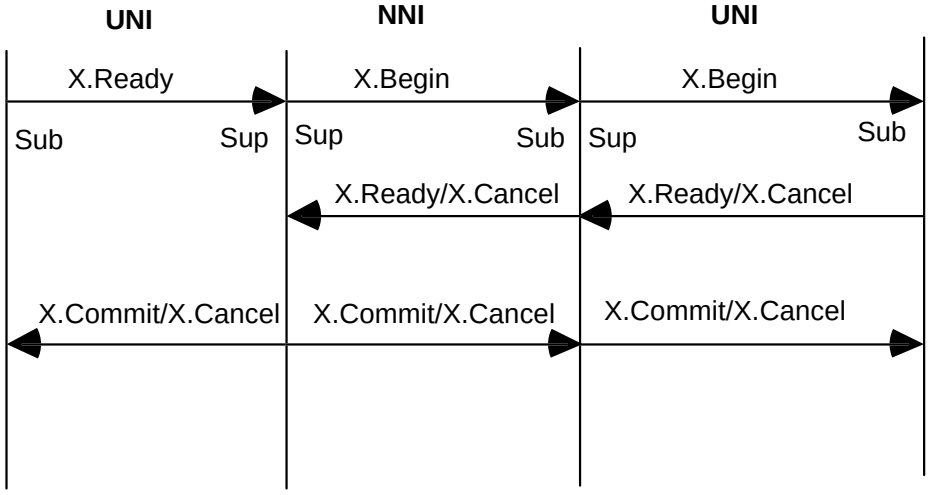


Figure 10: UNI/NNI flows with two phase request on NNI

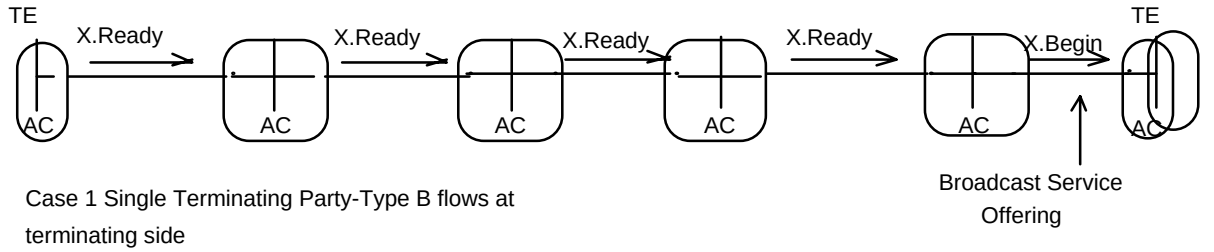
In figure 9, the one phase mechanism is used at the calling (or requesting) users side, and both the one phase and the two phase mechanisms are used at the called (or confirming) users side. Either mechanism can be employed at the NNI, depending on which side should have the ultimate authority to commit. The authority is generally transferred to the called user side when there is a single mandatory party.

It should be noted that for human non-interactive based services the generic information flow sequence is terminated by either a single X.Cancel, or one X.Commit information flow.

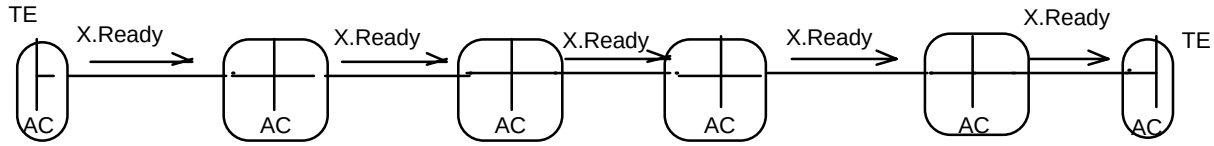
The following table and text contain the information flows sequences to be used in the specification of the stage 2 flows. The table associates each information flow "Action" with the signalling capabilities and the signalling plane which carries these information flow sequences.

Table 27: Information Flows used in Stage 2 Flow Diagrams

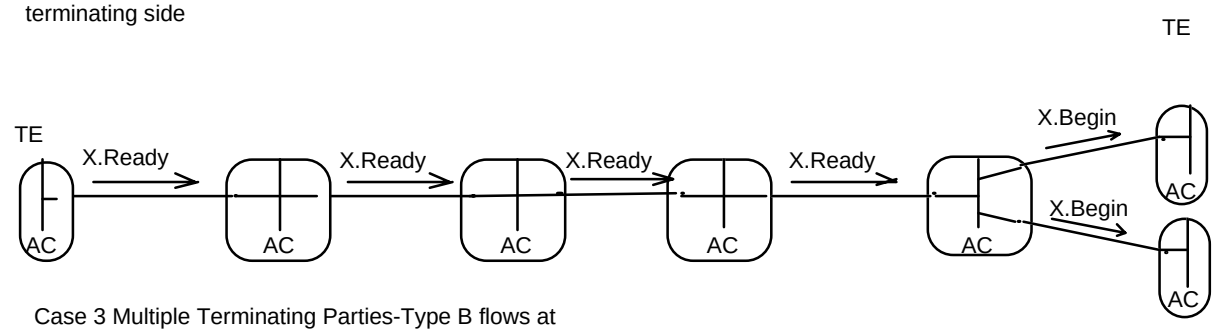
Signalling Flow Sequence	Begin	Ready	Commit	Cancel	Indication	Signalling Plane Association
Call-&-Bearer-Setup	☑	☑	☑	☑		Link
Call-Setup	☑	☑	☑	☑		enD/Edge
Add-Bearer-to Call	☑	☑	☑	☑		Link
Add-Party-to-Bearer	☑	☑	☑	☑		Link
Attach-Party-to-Bearer	☑	☑	☑	☑		Link
Add-Party-to-Call	☑	☑	☑	☑		enD/Edge
Add-Party-&-Bearer-to-Call	☑	☑	☑	☑		Link
Detach-Party-from-Bearer		☑	☑	☑		Link
Release-Bearer		☑	☑	☑		Link
Release-Party-from-Call		☑	☑	☑		enD/Edge
Release-Call		☑	☑	☑		enD/Edge
Re-negotiate-Bearer	☑	☑	☑	☑		enD/Edge
Modify-Bearer	☑	☑	☑	☑		Link
Call-&-Bearer-Setup-Remote	☑	☑	☑	☑		enD/Edge
Add-Bearer-to-Call-Remote	☑	☑	☑	☑		enD/Edge
Add-Party-to-Bearer-Remote	☑	☑	☑	☑		enD/Edge
Attach-Party-to-Bearer-Remote	☑	☑	☑	☑		enD/Edge
Add-Party-&-Bearer-to-Call-Remote	☑	☑	☑	☑		enD/Edge
Detach-Party-from-Bearer-Remote		☑	☑	☑		enD/Edge
Release-Bearer-Remote		☑	☑	☑		enD/Edge
Interrogation-Terminating-End-Point		☑	☑			enD/Edge
Look-Ahead	☑	☑		☑		enD/Edge
Report-enD/Edge					☑	enD/Edge
Report-Link					☑	Link
Notify-Party-Change					☑	enD/Edge
Notify-Bearer-Change					☑	enD/Edge
Notify-Bearer-Modified					☑	enD/Edge



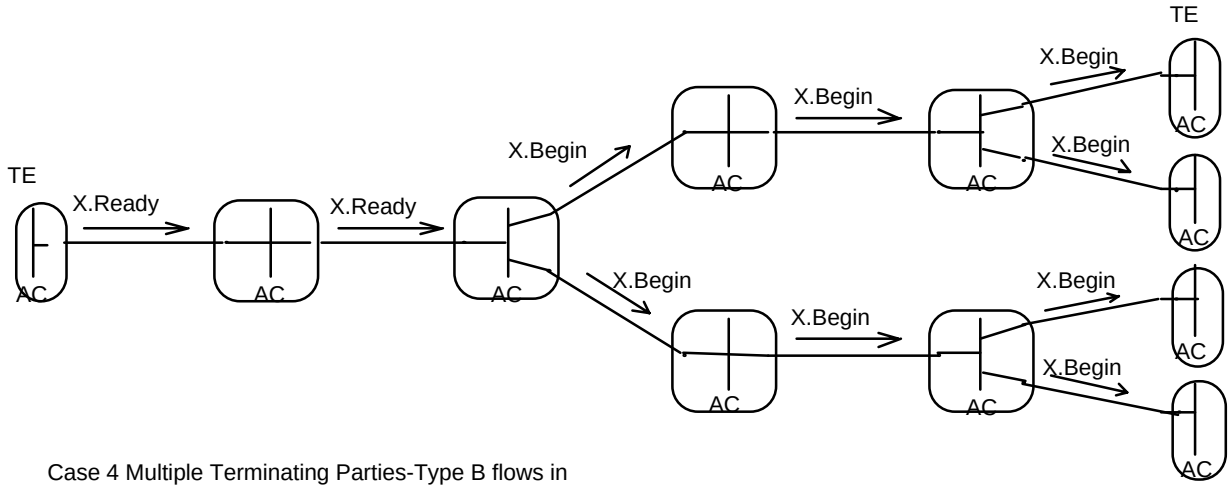
Case 1 Single Terminating Party-Type B flows at terminating side



Case 2 Single Terminating Party-Type A flow at terminating side



Case 3 Multiple Terminating Parties-Type B flows at terminating side



Case 4 Multiple Terminating Parties-Type B flows in the NNI

Figure 11: Flows illustrating conversion of X.Ready to X.Begin

An unconfirmed information flow has also to be considered e.g. in order to support the notification capability items. The set of mechanisms may be executed e.g. in order to cover user to user negotiation of U-plane attributes.

The following subclauses contain the description of the Information Flow sequences listed in table 27. In case that more than one addressed party is specified in an information flow all parties are mandatory.

An information flow will handle only one Network Connection or Network Connection Group per request.

6.3.1 Call-&-Bearer-Setup Information Flow Sequence

The action request (either Begin or Ready) information flow requests that a call and one or more network connection group is established between the transmitting Functional Entity and the Parties contained within the information flow.

6.3.2 Call-Setup Information Flow Sequence

The action request (either Begin or Ready) information flow requests that a call is established between the transmitting Functional Entity and the Parties contained within the information flow.

6.3.3 Add-Bearer-to-Call Information Flow Sequence

The action request (either Begin or Ready) information flow requests that one Network Connection Group or one Network Connection is established between the transmitting Functional Entity and the Parties contained within the information flow.

6.3.4 Add-Party-to-Bearer Information Flow Sequence

The action request (either Begin or Ready) information flow requests that one or more new Parties be added to the call and attached to the specified Network Connection contained in the information flow.

6.3.5 Attach-Party-to-Bearer Information Flow Sequence

The action request (either Begin or Ready) information flow requests that one or more existing Parties be attached to the specified existing Network Connection contained in the information flow.

6.3.6 Add-Party-to-Call Information Flow Sequence

The action request (either Begin or Ready) information flow requests that one or more Parties be added to the call.

6.3.7 Add-Party-&-Bearer-to-Call Information Flow Sequence

The action request (either Begin or Ready) information flow requests that one or more Parties and one new Network Connection Group or Network Connection is established between the transmitting Functional Entity and the Parties contained within the information flow.

6.3.8 Detach-Party-from-Bearer

The action request (Ready) information flow requests that one Party is detached from a specified Network Connection contained in the information flow.

6.3.9 Release-Bearer

The action request (Ready) information flow requests that one Network Connection specified in the information flow should be released from the call.

6.3.10 Release-Party-from-Call

The action request (Ready) information flow requests that one Party specified in the information flow should be released from the call.

6.3.11 Release-Call

The action request (Ready) information flow requests that the call should be released.

6.3.12 Re-Negotiate-Bearer

The action request (either Begin or Ready) information flow requests that the receiver reviews the contents of the flow and determine if the designated Network Connection characteristics can be accepted.

6.3.13 Modify-Bearer

The action request (either Begin or Ready) information flow requests that the receiver modifies the Network Connection to the specified new characteristics being requested.

6.3.14 Call-&-Bearer-Setup-Remote

When this action is requested by a serving node, the serving node is requesting the addressed serving node to establish one Network Connection Group between the parties specified within the information flow. The addressed node will be the "root" node associated with the specified Network Connections.

6.3.15 Add-Bearer-to-Call-Remote

When this action is requested by a serving node, the serving node is requesting the addressed serving node to establish the connections between the parties specified within the information flow. The addressed node is will be the "root" node associated with the specified Network Connections.

6.3.16 Add-Party-to-Bearer-Remote

When this action is requested by a serving node, the serving node is requesting the addressed serving node to add one or more new parties to the call and attach them to the specified existing Network Connections identified within the information flow. The addressed node is the "root" node associated with the specified Network Connections.

6.3.17 Attach-Party-to-Bearer-Remote

When this action is requested by a serving node, the serving node is requesting the addressed serving node to attach one or more existing parties to the specified existing Network Connections identified within the information flow. The addressed node is the "root" node associated with the specified Network Connections.

6.3.18 Add-Party-&-Bearer-to-Call-Remote

When this action is requested by a serving node, the serving node is requesting the addressed serving node to add two or more parties to the new Network Connections specified within the information flow. The addressed node will be the "root" node associated with the specified Network Connections.

6.3.19 Detach-Party-from-Bearer-Remote

When this action is invoked by a serving node, the serving node is requesting the addressed node that it should detach one or more parties from the specified Network Connections. The addressed node is the "root" node associated with the specified Network Connections.

6.3.20 Release-Bearer-Remote

When this action is requested by a serving node, the serving node is requesting that the addressed node release one or more Network Connections from the call. The addressed node is the "root" associated with the specified Network Connections.

6.3.21 Interrogation-Terminating-End-Point

This action requests the receiving entity to determine if the parties can accept the indicated operation designated by the contents of the information flow. This operation does not require any signalling state change of the parties.

6.3.22 Look-Ahead

This action requests the receiving service node to determine if the parties can accept the indicated operation designated by the contents of the information flow. The operations include either adding a bearer or establishing call & bearer, which will cause the signalling state changes of the parties.

6.3.23 Report-End/Edge

This action is information regarding the call that indicates that some event has taken place. An example of the type of information that is delivered is that the transmitting entity has received a Call Setup request and this request is being processed (Report.Ind.{Call Proceeding}). No response is expected by the transmitting entity.

6.3.24 Report-Link

This action is information regarding the Network Connection Group or Network Connection that indicates that some event has taken place. Two examples of this action can be illustrated by a Report.Ind {Network Connection Proceeding} information flow and a Report.Ind{Alerting} information flow. No response is expected by the transmitting entity.

6.3.25 Notify-Party-Change

Whenever a Party is added or released from the call parties associated with the call can be notified by the End/Edge node signalling node entity that initiated the change. This is done so that the local view of the call and its associated parties is kept up to date in each End/Edge signalling node. No Information Flow is expected in response.

6.3.26 Notify-Bearer-Change

Whenever a Network Connection Group or Network Connection is changed, such as attachment or detachment of parties to the Network Connection, Parties associated with the call can be notified by the End/Edge signalling node entity initiating the change. This is done so that the local view of the call and its associated parties is kept up to date in each End/Edge signalling node. No information flow is expected in response.

6.3.27 Notify-Bearer-Modified

Whenever a Network Connection is modified, such as a change in bandwidth, or configuration, Parties associated with the call can be notified by the End/Edge signalling node entity initiating the change. This is done so that the local view of the call and its associated Network Connections in each End/Edge signalling node entity is kept up to date. No information flow is expected in response.

6.4 Relationship between Parameters of Information Flows and Stage 2 model

This subclause specifies parameters to be used in information flows of the Stage 2 functional model. This subclause only discusses those external flows between two signalling entities of TE/service/relay nodes, such as r11, r14, r17 for edge-to-edge or end-to-end signalling control, and r21, r24, r27 for link-to-link signalling control (see the following figure).

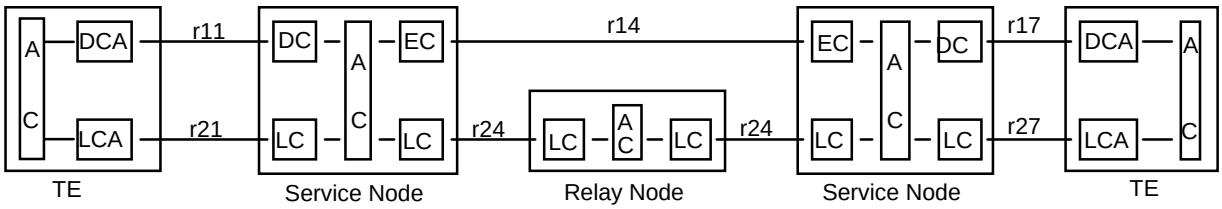


Figure 12: Information Relationships in Broadband Functional Signalling Model

The following table illustrates parameters to be used for either or both link and edge/end-to-end signalling control. All the parameters discussed in this subclause may not have one-to-one mapping to Stage 3 protocol information elements.

Table 28

		Information Flow					
Items of Information	Function	r11	r14	r17	r21	r24	r27
Call Object	Both LC/EC	M	M	M	M	M	M
Call ID	Both LC/EC	M	M	M	M	M	M
Party Reference ID	Both LC/EC	M/R	M/R	M/R	M/R	M/R	M/R
Requesting Party Address	both LC/EC	M	M	M	M	M	M
Telecom Service Type	both LC/EC	O	O	O	O	O	O
Service Component ID	both LC/EC	M/R	M/R	M/R	M/R	M/R	M/R
Network Connection Group ID	both LC/EC	M	M	M	O	O	O
Remote Party Object	both LC/EC	M	M	M	M	M	M
Party Reference ID	both LC/EC	M	M	M	M	M	M
Party Address	both LC/EC	O	O	O	O	O	O
Type of Remote Party	both LC/EC	M	M	M	M	M	M
Service Module Object	both LC/EC	M/R	M/R	M/R	M/R	M/R	M/R
Service Module ID	both LC/EC	M	M	M	O	O	O
Service Component ID	both LC/EC	M/R	M/R	M/R	O/R	O/R	O/R
Multiplexing Method	both LC/EC	O	O	O	O	O	O
Service Component Object	both LC/EC	M/R	M/R	M/R	O/R	O/R	O/R
Service Component ID	both LC/EC	M	M	M	O	O	O
Service Component Characteristics	both LC/EC	M	M	M	O	O	O
High Layer Information	both LC/EC	M	M	M	O	O	O
Service Traffic Descriptor	both LC/EC	M	M	M	O	O	O
Service QoS Descriptor	both LC/EC	M	M	M	O	O	O
Network Connection Object	LC				R	R	R
Network Connection ID	LC				M	M	M
Service Module ID	LC				M	M	M
Low Layer Information	LC				M	M	M
Directionality Indicator	LC				M	M	M
Transit Network ID	LC				O/R	O/R	O/R
ATM Connection Topology Type	LC				M	M	M
Local Attachment indicator	LC				-	-	-
Network Connection Group Object					R	R	R
Network Connection Group ID	LC				M	M	M
Network Connection ID	LC				M/R	M/R	M/R
Attachment Object							
Remote Party ID	LC				R	R	R
Network Connection ID	LC				R	R	R
Participation Object							
Remote Party ID	EC	R	R	R			
Service Component ID	EC	R	R	R			

Legend:

EC/DC =	Edge-to-Edge Control (EC) or EnD-to-End Control (DC)
LC =	Link-to-Link Control (LC)
Both of LC/EC =	Attribute used in both EC/DC and LC Information Flows
M =	Mandatory information
M/R =	Mandatory and Repeat information
O =	Optional information to some flows
O/R =	Optional and Repeat information
R =	Repeat information for multiple parties or connections

6.5 Information Flow and Stage 2 Model

This subclause shows relationship of each information flow name and the flow identifiers (r11, r14, r17, r21, r24, and r27) of Stage 2 functional model in the following table.

Table 29

Name of Information Flow	DCA-DC (r11)	EC-EC (r14)	DC-DCA (r17)	LCA-LC (r21)	LC-LC (r24)	LC-LCA (r27)
CALL & BEARER SETUP				☑	☑	☑
CALL SETUP	☑	☑	☑			
ADD BEARER TO CALL				☑	☑	☑
ADD PARTY TO BEARER				☑	☑	☑
ATTACH PARTY TO BEARER				☑	☑	☑
ADD PARTY TO CALL	☑	☑	☑			
ADD PARTY & BEARER TO CALL				☑	☑	☑
DETACH PARTY FROM BEARER				☑	☑	☑
RELEASE BEARER				☑	☑	☑
RELEASE PARTY FROM CALL	☑	☑	☑			
RELEASE CALL	☑	☑	☑			
RE-NEGOTIATE BEARER	☑	☑	☑			
MODIFY BEARER				☑	☑	☑
CALL & BEARER SETUP REMOTE		☑				
ADD BEARER TO CALL REMOTE		☑				
ADD PARTY TO BEARER REMOTE		☑				
ATTACH PARTY TO BEARER REMOTE		☑				
ADD PARTY & BEARER TO CALL REMOTE		☑				
DETACH PARTY FROM BEARER REMOTE		☑				
RELEASE BEARER REMOTE		☑				
INTERROGATION-TERMINATING-END-POINT		☑	☑			
LOOK-AHEAD		☑				
REPORT END/EDGE	☑	☑	☑			
REPORT LINK				☑	☑	☑
NOTIFY-PARTY-CHANGE	☑	☑	☑			
NOTIFY-BEARER-CHANGE	☑	☑	☑			
NOTIFY-BEARER-MODIFIED	☑	☑	☑			

6.6 Information Parameters for Information Flows

In the following tables the information relevant for each elemental flow is listed.

[Remark: REPORT, NOTIFY and REMOTE flows still need to be added]

Legend for the table entries:

"blank":	Not contained
X:	Contained (may be optional or mandatory)
R:	Contained, may be repeated, (may be optional or mandatory)
[...]:	Indicates that the table entry refers not to an information item, but to the category
(SL):	Indicates the presence of a Selection List
(SR):	Indicates a Selection Result (only applicable if a Selection List is used)

[Remark: which items are optional and which are mandatory still needs to be clarified]

Table 30

CALL & BEARER SETUP	Type A: READY Type B: BEGIN	Type A: COMMIT Type B: READY	Type B: COMMIT	Type A: CANCEL Type B: CANCEL
call_information	[X]	[X]	[X]	[X]
{ Call ID }	X	X	X	X
{ Requesting Party Address }	X			
{ List of Parties }	X			
{ Telecommunication Service Type }	X			
{ List of Service Components }	X			
{ Network Connection Group IDs }	X			
remote_party_information	[R]			
{ Remote Party Reference ID }	X			
{ Party Address }	X			
{ Type of Party }	X			
service_module_information	[R]			
{ Service Module ID }	X			
{ List of Service Components }	X			
{ Multiplexing Method }	X			
service_component_information	[R]	[R](SR)		
{ Service Component ID }	X	X		
{ Service Component Characteristics: High Layer Information }	X			
{ Service Traffic Descriptor }	X			
{ Service QoS Descriptor }	X			
network_connection_information	[R]	[R](SR)		
{ Network Connection ID }	X	X		
{ List of Service Modules}	R(SL)	X(SR)		
{ Low Layer Information }	R(SL)	X(SR)		
{ Directionality Indicator }	X	X		
{ Transit Network ID}	X			
{ ATM Connection Topology Type}	X			
{ Local Attachment}	X			
network_connection_group_information	[X]			
{ Network Connection Group ID }	X			
{ List of Network Connections }	X			

Table 31

ADD BEARER TO CALL	Type A: READY Type B: BEGIN	Type A: COMMIT Type B: READY	Type B: COMMIT	Type A: CANCEL Type B: CANCEL
call_information	[X]	[X]	[X]	[X]
{ Call ID }	X	X	X	X
{ Requesting Party Address }	X			
{ List of Parties }	X			
{ Telecommunication Service Type }	X			
{ List of Service Components }	X			
{ Network Connection Group IDs }	X			
remote_party_information	[R]			
{ Remote Party Reference ID }	X			
{ Party Address }	X			
{ Type of Party }	X			
service_module_information	[R]			
{ Service Module ID }	X			
{ List of Service Components }	X			
{ Multiplexing Method }	X			
service_component_information	[R]	[R](SR)		
{ Service Component ID }	X	X		
{ Service Component Characteristics: High Layer Information }	X			
{ Service Traffic Descriptor }	X			
{ Service QoS Descriptor }	X			
network_connection_information	[R]	[R](SR)		
{ Network Connection ID }	X	X		
{ List of Service Modules}	R(SL)	X(SR)		
{ Low Layer Information }	R(SL)	X(SR)		
{ Directionality Indicator }	X	X		
{ Transit Network ID}	X			
{ ATM Connection Topology Type}	X			
{ Local Attachment}	X			
network_connection_group_information	[X]			
{ Network Connection Group ID }	X			
{ List of Network Connections }	X			

Table 32

ADD PARTY TO BEARER	Type A: READY Type B: BEGIN	Type A: COMMIT Type B: READY	Type B: COMMIT	Type A: CANCEL Type B: CANCEL
call_information	[X]	[X]	[X]	[X]
{ Call ID }	X	X	X	X
{ Requesting Party Address }	X			
{ List of Parties }	X			
{ Telecommunication Service Type }	X			
{ List of Service Components }	X			
{ Network Connection Group IDs }	X			
remote_party_information	[R]			
{ Remote Party Reference ID }	X			
{ Party Address }	X			
{ Type of Party }	X			
service_module_information	[R]			
{ Service Module ID }	X			
{ List of Service Components }	X			
{ Multiplexing Method }	X			
service_component_information	[R]	[R](SR)		
{ Service Component ID }	X	X		
{ Service Component Characteristics: High Layer Information }	X			
{ Service Traffic Descriptor }	X			
{ Service QoS Descriptor }	X			
network_connection_information	[R]	[R](SR)		
{ Network Connection ID }	X	X		
{ List of Service Modules}	R(SL)	X(SR)		
{ Low Layer Information }	R(SL)	X(SR)		
{ Directionality Indicator }	X	X		
{ Transit Network ID}	X			
{ ATM Connection Topology Type}	X			
{ Local Attachment}	X			
network_connection_group_information	[X]			
{ Network Connection Group ID }	X			
{ List of Network Connections }	X			

Table 33

ATTACH PARTY TO BEARER	Type A: READY Type B: BEGIN	Type A: COMMIT Type B: READY	Type B: COMMIT	Type A: CANCEL Type B: CANCEL
call_information	[X]	[X]	[X]	[X]
{ Call ID }	X	X	X	X
{ Requesting Party Address }	X			
{ List of Parties }	X			
{ Telecommunication Service Type }	X			
{ List of Service Components }	X			
{ Network Connection Group IDs }	X			
remote_party_information	[R]			
{ Remote Party Reference ID }	X			
{ Party Address }	X			
{ Type of Party }	X			
service_module_information	[R]			
{ Service Module ID }	X			
{ List of Service Components }	X			
{ Multiplexing Method }	X			
service_component_information	[R]	[R](SR)		
{ Service Component ID }	X	X		
{ Service Component Characteristics: High Layer Information }	X			
{ Service Traffic Descriptor }	X			
{ Service QoS Descriptor }	X			
network_connection_information	[R]	[R](SR)		
{ Network Connection ID }	X	X		
{ List of Service Modules}	R(SL)	X(SR)		
{ Low Layer Information }	R(SL)	X(SR)		
{ Directionality Indicator }	X	X		
{ Transit Network ID}	X			
{ ATM Connection Topology Type}	X			
{ Local Attachment}	X			
network_connection_group_information	[X]			
{ Network Connection Group ID }	X			
{ List of Network Connections }	X			

Table 34

ADD PARTY & BEARER TO CALL	Type A: READY Type B: BEGIN	Type A: COMMIT Type B: READY	Type B: COMMIT	Type A: CANCEL Type B: CANCEL
call_information	[X]	[X]	[X]	[X]
{ Call ID }	X	X	X	X
{ Requesting Party Address }	X			
{ List of Parties }	X			
{ Telecommunication Service Type }	X			
{ List of Service Components }	X			
{ Network Connection Group IDs }	X			
remote_party_information	[R]			
{ Remote Party Reference ID }	X			
{ Party Address }	X			
{ Type of Party }	X			
service_module_information	[R]			
{ Service Module ID }	X			
{ List of Service Components }	X			
{ Multiplexing Method }	X			
service_component_information	[R]	[R](SR)		
{ Service Component ID }	X	X		
{ Service Component Characteristics: High Layer Information }	X			
{ Service Traffic Descriptor }	X			
{ Service QoS Descriptor }	X			
network_connection_information	[R]	[R](SR)		
{ Network Connection ID }	X	X		
{ List of Service Modules}	R(SL)	X(SR)		
{ Low Layer Information }	R(SL)	X(SR)		
{ Directionality Indicator }	X	X		
{ Transit Network ID}	X			
{ ATM Connection Topology Type}	X			
{ Local Attachment}	X			
network_connection_group_information	[X]			
{ Network Connection Group ID }	X			
{ List of Network Connections }	X			

Table 35

DETACH PARTY FROM BEARER	READY	COMMIT
call_information	[X]	[X]
{ Call ID }	X	X
{ Requesting Party Address }	X	
{ List of Parties }	X	
{ Telecommunication Service Type }		
{ List of Service Components }		
{ Network Connection Group IDs }		
remote_party_information	[R]	
{ Remote Party Reference ID }	X	
{ Party Address }		
{ Type of Party }		
service_module_information		
{ Service Module ID }		
{ List of Service Components }		
{ Multiplexing Method }		
service_component_information		
{ Service Component ID }		
{ Service Component Characteristics: High Layer Information }		
{ Service Traffic Descriptor }		
{ Service QoS Descriptor }		
network_connection_information	[X]	
{ Network Connection ID }	X	
{ List of Service Modules}		
{ Low Layer Information }		
{ Directionality Indicator }		
{ Transit Network ID}		
{ ATM Connection Topology Type}		
{ Local Attachment}		
network_connection_group_information		
{ Network Connection Group ID }		
{ List of Network Connections }		

Table 36

RELEASE BEARER	READY	COMMIT
call_information	[X]	[X]
{ Call ID }	X	X
{ Requesting Party Address }	X	
{ List of Parties }		
{ Telecommunication Service Type }		
{ List of Service Components }		
{ Network Connection Group IDs }		
remote_party_information		
{ Remote Party Reference ID }		
{ Party Address }		
{ Type of Party }		
service_module_information		
{ Service Module ID }		
{ List of Service Components }		
{ Multiplexing Method }		
service_component_information		
{ Service Component ID }		
{ Service Component Characteristics: High Layer Information }		
{ Service Traffic Descriptor }		
{ Service QoS Descriptor }		
network_connection_information	[X]	
{ Network Connection ID }	X	
{ List of Service Modules}		
{ Low Layer Information }		
{ Directionality Indicator }		
{ Transit Network ID}		
{ ATM Connection Topology Type}		
{ Local Attachment}		
network_connection_group_information		
{ Network Connection Group ID }		
{ List of Network Connections }		

Table 37

RELEASE PARTY FROM CALL	READY	COMMIT
call_information	[X]	[X]
{ Call ID }	X	X
{ Requesting Party Address }	X	
{ List of Parties }	X	
{ Telecommunication Service Type }		
{ List of Service Components }		
{ Network Connection Group IDs }		
remote_party_information	[X]	
{ Remote Party Reference ID }	X	
{ Party Address }		
{ Type of Party }		
service_module_information		
{ Service Module ID }		
{ List of Service Components }		
{ Multiplexing Method }		
service_component_information		
{ Service Component ID }		
{ Service Component Characteristics: High Layer Information }		
{ Service Traffic Descriptor }		
{ Service QoS Descriptor }		
network_connection_information		
{ Network Connection ID }		
{ List of Service Modules}		
{ Low Layer Information }		
{ Directionality Indicator }		
{ Transit Network ID}		
{ ATM Connection Topology Type}		
{ Local Attachment}		
network_connection_group_information		
{ Network Connection Group ID }		
{ List of Network Connections }		

Table 38

RELEASE CALL	READY	COMMIT
call_information	[X]	[X]
{ Call ID }	X	X
{ Requesting Party Address }	X	
{ List of Parties }		
{ Telecommunication Service Type }		
{ List of Service Components }		
{ Network Connection Group IDs }		
remote_party_information		
{ Remote Party Reference ID }		
{ Party Address }		
{ Type of Party }		
service_module_information		
{ Service Module ID }		
{ List of Service Components }		
{ Multiplexing Method }		
service_component_information		
{ Service Component ID }		
{ Service Component Characteristics: High Layer Information }		
{ Service Traffic Descriptor }		
{ Service QoS Descriptor }		
network_connection_information		
{ Network Connection ID }		
{ List of Service Modules}		
{ Low Layer Information }		
{ Directionality Indicator }		
{ Transit Network ID}		
{ ATM Connection Topology Type}		
{ Local Attachment}		
network_connection_group_information		
{ Network Connection Group ID }		
{ List of Network Connections }		

Table 39

RE-NEGOTIATE BEARER	Type A: READY Type B: BEGIN	Type A: COMMIT Type B: READY	Type B: COMMIT	Type A: CANCEL Type B: CANCEL
call_information	[X]	[X]	[X]	[X]
{ Call ID }	X	X	X	X
{ Requesting Party Address }	X			
{ List of Parties }				
{ Telecommunication Service Type }				
{ List of Service Components }				
{ Network Connection Group IDs }				
remote_party_information				
{ Remote Party Reference ID }				
{ Party Address }				
{ Type of Party }				
service_module_information				
{ Service Module ID }				
{ List of Service Components }				
{ Multiplexing Method }				
service_component_information				
{ Service Component ID }				
{ Service Component Characteristics: High Layer Information }				
{ Service Traffic Descriptor }				
{ Service QoS Descriptor }				
network_connection_information	[R]	[R](SR)		
{ Network Connection ID }	X	X		
{ List of Service Modules}				
{ Low Layer Information }	R(SL)	X(SR)		
{ Directionality Indicator }				
{ Transit Network ID}				
{ ATM Connection Topology Type}				
{ Local Attachment}				
network_connection_group_information				
{ Network Connection Group ID }				
{ List of Network Connections }				

Table 40

MODIFY BEARER	Type A: READY Type B: BEGIN	Type A: COMMIT Type B: READY	Type B: COMMIT	Type A: CANCEL Type B: CANCEL
call_information	[X]	[X]	[X]	[X]
{ Call ID }	X	X	X	X
{ Requesting Party Address }	X			
{ List of Parties }				
{ Telecommunication Service Type }				
{ List of Service Components }				
{ Network Connection Group IDs }				
remote_party_information				
{ Remote Party Reference ID }				
{ Party Address }				
{ Type of Party }				
service_module_information				
{ Service Module ID }				
{ List of Service Components }				
{ Multiplexing Method }				
service_component_information				
{ Service Component ID }				
{ Service Component Characteristics: High Layer Information }				
{ Service Traffic Descriptor }				
{ Service QoS Descriptor }				
network_connection_information	[R]			
{ Network Connection ID }	X			
{ List of Service Modules}				
{ Low Layer Information }	R(SL)			
{ Directionality Indicator }	X			
{ Transit Network ID}	X			
{ ATM Connection Topology Type}	X			
{ Local Attachment}				
network_connection_group_information				
{ Network Connection Group ID }				
{ List of Network Connections }				

Table 41

INTERROGATION-TERMINATING-END-POINT LOOK-AHEAD	BEGIN	READY	CANCEL
call_information	[X]	[X]	[X]
{ Call ID }	X	X	X
{ Requesting Party Address }	X		
{ List of Parties }			
{ Telecommunication Service Type }	X		
{ List of Service Components }	X		
{ Network Connection Group IDs }			
remote_party_information	[X]		
{ Remote Party Reference ID }			
{ Party Address }	X		
{ Type of Party }	X		
service_module_information	[R]		
{ Service Module ID }	X		
{ List of Service Components }	X		
{ Multiplexing Method }	X		
service_component_information	[R]	[R](SR)	
{ Service Component ID }	X	X	
{ Service Component Characteristics: High Layer Information }	X		
{ Service Traffic Descriptor }	X		
{ Service QoS Descriptor }	X		
network_connection_information	[R]	[R](SR)	
{ Network Connection ID }	X	X	
{ List of Service Modules}	R(SL)	X(SR)	
{ Low Layer Information }	R(SL)	X(SR)	
{ Directionality Indicator }	X	X	
{ Transit Network ID}	X		
{ ATM Connection Topology Type}	X		
{ Local Attachment}	X		
network_connection_group_information	[X]		
{ Network Connection Group ID }	X		
{ List of Network Connections }	X		

Table 42

CALL SETUP, ADD PARTY TO CALL	READY	COMMIT	CANCEL
call_information	[X]	[X]	[X]
{ Call ID }	X	X	X
{ Requesting Party Address }	X		
{ List of Parties }			
{ Telecommunication Service Type }	X		
{ List of Service Components }	X		
{ Network Connection Group IDs }			
remote_party_information	[X]		
{ Remote Party Reference ID }			
{ Party Address }	X		
{ Type of Party }	X		
service_module_information	[R]		
{ Service Module ID }	X		
{ List of Service Components }	X		
{ Multiplexing Method }	X		
service_component_information	[R]	[R](SR)	
{ Service Component ID }	X	X	
{ Service Component Characteristics: High Layer Information }	X		
{ Service Traffic Descriptor }	X		
{ Service QoS Descriptor }	X		
network_connection_information	[R]	[R](SR)	
{ Network Connection ID }	X	X	
{ List of Service Modules}	R(SL)	X(SR)	
{ Low Layer Information }	R(SL)	X(SR)	
{ Directionality Indicator }	X	X	
{ Transit Network ID}	X		
{ ATM Connection Topology Type}	X		
{ Local Attachment}	X		
network_connection_group_information	[X]		
{ Network Connection Group ID }	X		
{ List of Network Connections }	X		

7 Functional entity actions

Stage 2 flows for each Signalling Capability contained in Capability Set 2 is illustrated via a high level overview. The over view model does not illustrate all possible configurations which could exist within an actual instant of the service, however, the examples have been chosen in order to illustrate the general principles. The overview will employ the network configuration shown in the following figure. The actions illustrated in this figure can be used to describe signalling control actions associated with establishment, bandwidth modification, or release of one or more Network Connections. The figure illustrates the actions necessary to establish a point-to-point Network Connection, when the requesting party is or will be associated with the specified connection.

Note that the signalling flows and actions illustrate the establishment of a Network Connection Group (defined in subclause 4.2.7). In many cases a Network Connection Group may consist of just one Network Connection. However, the examples shown here illustrate the general case of a Network Connection Group (with possibly several Network Connections) being established in a single signalling transaction.

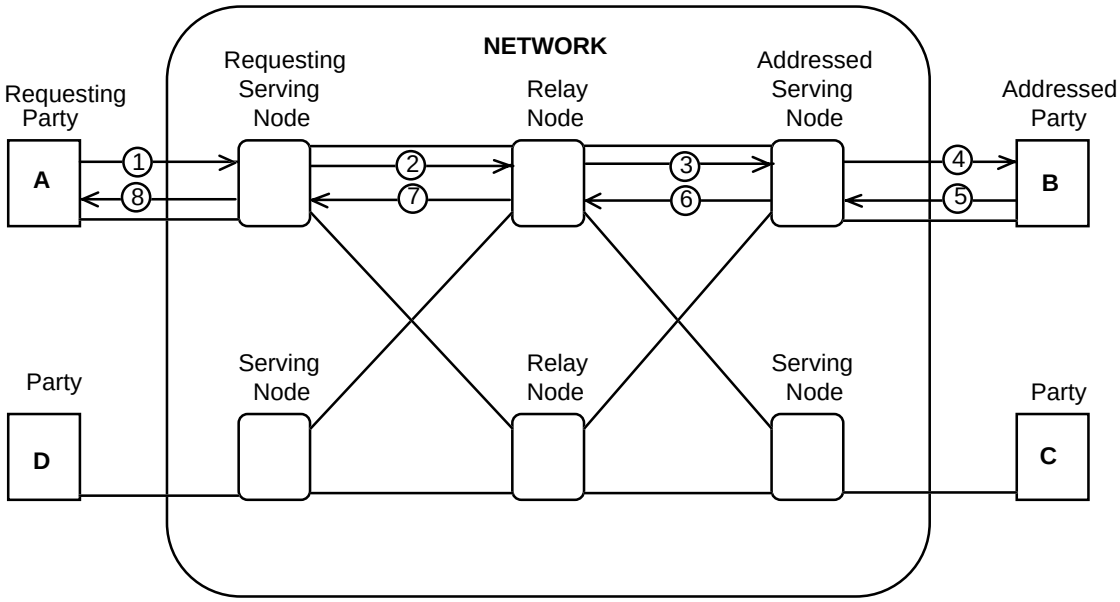


Figure 13: Point-to-point Network Connection (Type 1 Between A and B)

The actions illustrated in the above figure are described as follows;

1) Signalling Service Request issued by service requester: Receiving entity validates request, modifies internal state information, and then issues action 2.
2) Relayed Signalling Service Request issued by requester's serving node: Receiving entity validates request, modifies internal state information, and then relays request to next node as action 3.
3) Relayed Signalling Service Request issued by relay node: Receiving entity validates request, modifies internal state information, and then issues the request on the addressed party's interface as action 4.
4) Signalling Service Request issued by addressed party's serving node. Receiving entity validates request, modifies internal state information, and then issues its response as action 5.
5) Signalling Service Response issued by addressed party. Receiving entity records response, modifies internal state information and then relays response as action 6.
6) Relayed Signalling Service Response issued by addressed party's serving node. Receiving entity records response, modifies internal state information and then relays response as action 7.
7) Relayed Signalling Service Response issued by relay exchange. Receiving entity records response, modifies internal state information and relays response to the service requester as action 8.
8) Signalling Service Response issued by requester's serving node. Requester records response, modifies internal state information, and notifies user of the outcome of the requested service.

The purpose of this overview model is that it provides a end to end pictorial representation of the signalling capability in one figure. Again, note that the model does not present all possible network topologies, however, it illustrates the general configurations that would be encountered in intra-network operation. The extension to multiple networks can be extrapolated by replacing the serving modes and relay nodes with local serving networks and transit networks. The resulting information flows, generally would not be changed.

The following subclauses will describe the Signalling Capabilities contained in Capability Set 2 using this model.

7.1 Simultaneous Call and Connection Establishment

The simultaneous establishment of a call and Network Connection Groups can be separated into several categories such as those that are associated with Type 1 and Type 2 connections. For Signalling Capability Set 2 the following categories have been agreed upon.

1) Call and Network Connection Group Establishment of Type 1 Connections;
2) Call and Network Connection Group Establishment of Type 2 Connections.

The following subclauses describe these signalling services.

7.1.1 Call and Network Connection Group Establishment of Type 1 Connections

Three example variations of this capability will be illustrated in this subclause. These three variations are as follows;

1) Call and Network Connection Group Establishment of a single Network Connection Group without network initiated "Look Ahead";
2) Call and Network Connection Group Establishment of a single Network Connection Group with network initiated "Look Ahead"; and
3) Call and Network Connection Group Establishment of a Single Network Connection Group by a Party other than one of the two parties to be attached to the Network Connection Group without network initiated "Look Ahead".

The overview of the Type 1 simultaneous call and Network Connection Group establishment capabilities are contained in the following subclauses.

7.1.1.1 Call and Network Connection Group Establishment - Single Network Connection Group - Without "Look Ahead"

The User (Party A) requests a two party call between Party A and Party B. One point-to-point Network Connection Group is to be associated with this call. Parties A and B are to be attached to the Network Connection Group. The User also specifies the Higher Layer service to be carried on this Network Connection Group and the desired Network Bearer service that should be established. The requested service is of the non-human interactive type. Therefore, immediate answer can be performed by the Party B's equipment. If the Requested Party B equipment can accept the requested service, the designated attachment method, and specified bearer service, the equipment will indicate acceptance of the call and Network Connection Group request. This example also assumes that the Requested Party is connected to a Multi-signalling entity interface. In addition, the network does not perform a "Look Ahead" procedure before progressing with the Network Connection Group establishment.

The signalling capability of simultaneously establishing this call and Network Connection Group between the two requested parties without network "Look Ahead" is illustrated below.

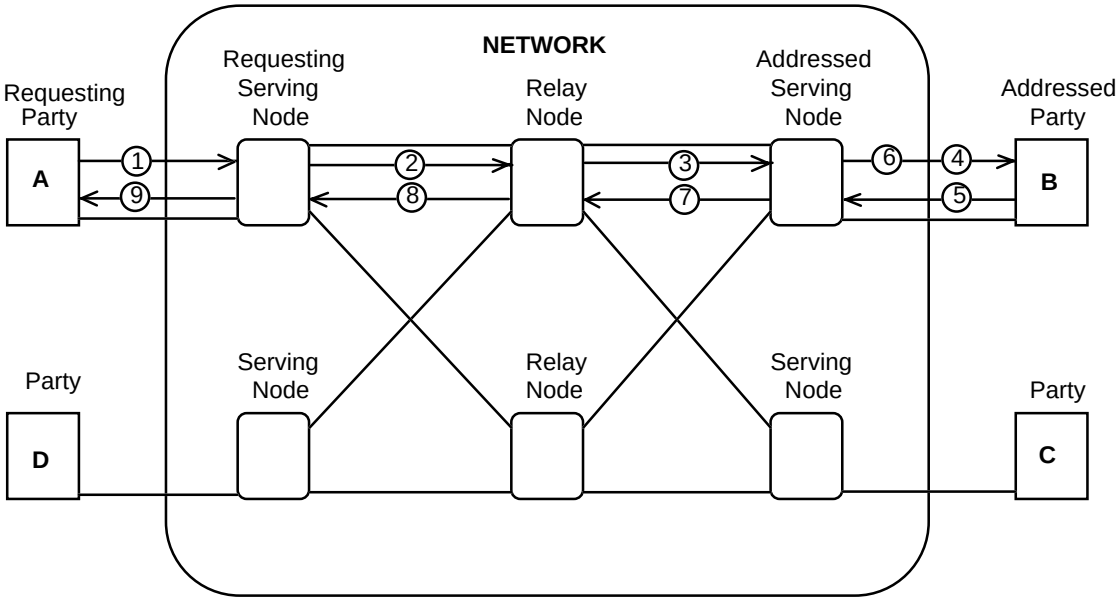


Figure 14: Point-to-point Network Connection Group (Type 1 Between A and B) - (no Look Ahead)

The actions illustrated in the above figure are as follows.

Requesting Party's terminal equipment issues the following information flow towards it's serving node. The terminal equipment then attaches to the backward portion of the Network Connection Group.

1) **CALL-&-BEARER SETUP.ready:** [Requesting Party = A, Addressed Party=B]

The requester's serving node validates the request and determines the route and outgoing trunk facility. Since only one outgoing port is needed, the serving node can commit to the request and therefore issues the following information flow towards the selected relay node. The Network Connection Group is backward through connected.

2) **CALL-&-BEARER SETUP.ready:** [Requesting Party = A, Addressed Party=B]

The selected relay node validates the request and determines the route and outgoing trunk facility. Since only one outgoing port is needed, the serving node can commit to the request and therefore issues the following information flow towards the addressed serving node. The Network Connection Group is backward through connected.

3) **CALL-&-BEARER SETUP.ready:** [Requesting Party = A, Addressed Party=B]

The addressed serving node validates the request and selects the terminating interface facility. Since the interface is classified as a multiple signalling entity interface, the serving node can not commit to the addressed end point and therefore issues the following information flow towards the selected interface facility. The Network Connection Group is backward through connected.

4) **CALL-&-BEARER SETUP.begin:** [Requesting Party = A, Addressed Party=B]

The addressed terminal equipment determines that it can accept the requested and issues the following information flow towards it's associated serving node.

5) **CALL-&-BEARER SETUP.ready**

The addressed serving node records the responses to the action request and selects of the responding terminals. The selected terminal is sent information flow number 6. The serving node then clears the non-selected terminals (note this action is not illustrated for simplicity), and issues information flow 7 toward the requesting relay node. The Network Connection Group is forward connected.

6) **CALL-&-BEARER SETUP.commit**

When the Terminal receives this information flow, it records the commitment, and connects in both the forward and backward directions.

7) **CALL-&-BEARER SETUP.commit**

When the relay node receives this information flow, it records the commitment, and relays this commitment to the requesting serving node by issuing information flow number 8, and performs forward through connect of the Network Connection Group.

8) **CALL-&-BEARER SETUP.commit**

When the requesting service node receives this information flow, it records the commitment, and relays this commitment to the requesting user equipment by issuing information flow number 9, and performs forward through connect of the Network Connection Group.

9) **CALL-&-BEARER SETUP.commit**

When the user equipment receives this information flow, it records the commitment, and notifies the user of this commitment, and connects to the forward portion of the Network Connection Group.

7.1.1.2 Call and Network Connection Group Establishment - Single Network Connection Group - With "Look Ahead"

The User (Party A) requests a two party call between Party A and Party B. One point-to-point Network Connection Group is to be associated with this call. Parties A and B are to be attached to the Network Connection Group. The User also specifies the Higher Layer service to be carried on this Network Connection Group and the desired Network Bearer service that should be established. The requested service is of the non-human interactive type. Therefore, immediate answer can be performed by the Party B's equipment. If the Requested Party B equipment can accept the requested service, the designated attachment method, and specified bearer service, the equipment will indicate acceptance of the call and Network Connection Group request. This example also assumes that the Requested Party is connected to a Multi-signalling entity interface. In addition, the network does perform a "Look Ahead" procedure before progressing with the Network Connection Group establishment. If this "Look Ahead" is successful then the network progresses the establishment of the call and Network Connection Group in the same manner as if network "Look Ahead" was not employed.

The signalling capability of simultaneously establishing this call and Network Connection Group between the two requested parties with network "Look Ahead" is illustrated below.

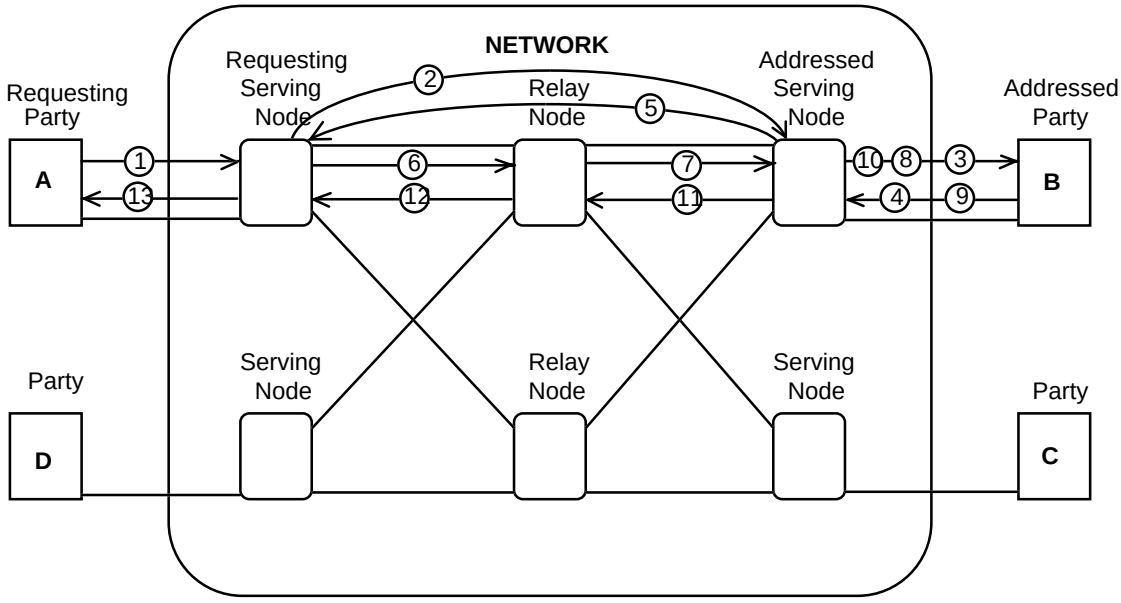


Figure 15: Point-to-point Network Connection Group (Type 1 Between A and B) - (Look Ahead)

The actions illustrated in the above figure are as follows.

Requesting Party's terminal equipment issues the following information flow towards it's serving node. The terminal equipment then attaches to the backward portion of the Network Connection Group.

1) **CALL-&-BEARER SETUP.ready:** [Requesting Party = A, Addressed Party=B]

The requester's serving node validates the request and determines the edge to edge route to the addressed serving node. It then issues the following information flow to the addressed serving node in order to determine if Party B will be willing to accept the call and Network Connection Group when it is offered at a latter time. This network capability reduces the possibility of establishing a Network Connection Group and having it rejected by the addressed serving node or addressed party.

2) **INTERROGATION-TERMINATING-END-POINT.ready:** [Requesting Party = A, Addressed Party=B]

The Addressed serving node validates the request, selects the interface associated with Party B and broadcasts the following information flow to all terminals connected to the selected interface.

3) **INTERROGATION-TERMINATING-END-POINT.ready:** [Requesting Party = A, Addressed Party=B]

When this information flow is received, the equipment associated with Party B determines if the requested service, Attachment, and bearer characteristics can be accepted. If the request can be accepted, the terminal equipment will issue the following information flow.

4) **INTERROGATION-TERMINATING-END-POINT.commit**

When the addressed serving node receives at least one copy of the above information flow, it will issue the following information flow. If multiple terminals respond, only one commit message will be sent to the requesting serving node.

5) **INTERROGATION-TERMINATING-END-POINT.commit**

When the requesting service node receives the commit information flow, it determines the route and outgoing trunk facility. Since only one outgoing port is needed, the serving node can commit to the request and therefore issues the following information flow towards the selected relay node. The Network Connection Group is backward through connected.

6) **CALL-&-BEARER SETUP.ready:** [Requesting Party = A, Addressed Party=B]

The selected relay node validates the request and determines the route and outgoing trunk facility. Since only one outgoing port is needed, the serving node can commit to the request and therefore issues the following information flow towards the addressed serving node. The Network Connection Group is backward through connected.

7) **CALL-&-BEARER SETUP.ready:** [Requesting Party = A, Addressed Party=B]

The addressed serving node validates the request and selects the terminating interface facility. Since the interface is classified as a multiple signalling entity interface, the serving node can not commit to the addressed end point and therefore issues the following information flow towards the selected interface facility. The Network Connection Group is backward through connected.

8) **CALL-&-BEARER SETUP.begin:** [Requesting Party = A, Addressed Party=B]

The addressed terminal equipment determines that it can accept the request and issues the following information flow towards it's associated serving node.

9) **CALL-&-BEARER SETUP.ready**

The addressed serving node records the responses to the action request and selects of the responding terminals. The selected terminal is sent information flow number 10. The serving node then clears the non-selected terminals (note this action is not illustrated for simplicity), and issues information flow 11 toward the requesting relay node. The Network Connection Group is forward connected.

10) **CALL-&-BEARER SETUP.commit**

When the Terminal receives this information flow, it records the commitment, and connects in both the forward and backward directions.

11) **CALL-&-BEARER SETUP.commit**

When the relay node receives this information flow, it records the commitment, and relays this commitment to the requesting serving node by issuing information flow number 12, and performs forward through connect of the Network Connection Group.

12) **CALL-&-BEARER SETUP.commit**

When the requesting service node receives this information flow, it records the commitment, and relays this commitment to the requesting user equipment by issuing information flow number 13, and performs forward through connect of the Network Connection Group.

13) **CALL-&-BEARER SETUP.commit**

When the user equipment receives this information flow, it records the commitment, and notifies the user of this commitment, and connects to the forward portion of the Network Connection Group.

7.1.1.3 Call and Network Connection Group Establishment -Single Network Connection Group - Third Party

The User (Party D) requests a three party call between Party A, B, and Party D. One point-to-point Network Connection Group is to be associated with this call. Parties A and B are to be attached to the Network Connection Group. The User (Party D) also specifies the Higher Layer service to be carried on this Network Connection Group and the desired Network Bearer service that should be established. The requested service is of the non-human interactive type. Therefore, immediate answer can be performed by the Party A's and Party B's equipment. If both the Requested Parties equipment can accept the requested service, the designated attachment method, and specified bearer service, the equipment will indicate acceptance of the call and Network Connection Group request. This example also assumes that both Requested Parties are connected to a Multi-signalling entity interface. In addition, the network does not perform a "Look Ahead" procedure before progressing with the Network Connection Group establishment.

The signalling capability of simultaneously establishing a call between three parties and establishing a point-to-point Network Connection Group between two of these parties without network "Look Ahead" is illustrated below. This request was issued by Party D that will not be attached to the request Network Connection Group.

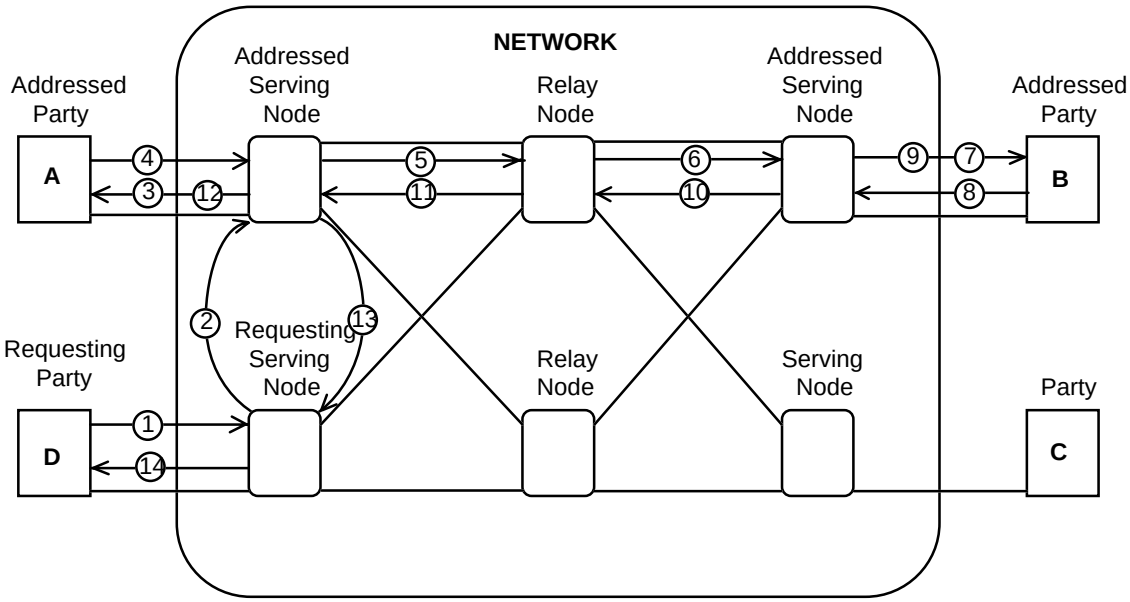


Figure 16: Point-to-point Network Connection Group (Type 1 Between A and B) Third Party

The actions illustrated in the above figure are as follows.

Requesting Party's (Party D) terminal equipment issues the following information flow towards it's serving node. The terminal equipment then attaches to the backward portion of the Network Connection Group.

1) **CALL-&-BEARER SETUP.ready:** [Requesting Party = D, Addressed Party = A,B]

The requester's serving node validates the request and determines which party will be designated the "root" Party for this Network Connection Group (for this example Party A is chosen) and the edge signalling route to the serving node associated with selected "root" Party. Since Party D is not attached to the requested Network Connection Group, and the "root" of the Network Connection Group is located in another serving node, a Remote operation request needs to be invoked. In addition, only one outgoing signalling port is needed, therefore Party D's serving node can commit to the request and therefore issues the following information flow towards the selected "root" serving node.

2) **CALL-&-BEARER SETUP-REMOTE.ready:** [Requesting Party = D, Addressed Party = A,B]

The selected serving node validates the request and determines the interface associated with Party A. Since this interface is a multi-signalling entity interface and that the Network Connection Group shall be established only after Party B also is willing to commit to the Network Connection Group, the serving node can not commit to the request and therefore issues the following information flow towards the addressed party (Party A).

3) **CALL-&-BEARER SETUP.begin:** [Requesting Party = D, Addressed Party = A,B]

When Party A receives the above information flow, it determines if it can accept the request contained in the flow. If it can accept the call and Network Connection Group, it responses with the following message. The terminal equipment then attaches to the backward portion of the Network Connection Group.

4) **CALL-&-BEARER SETUP.ready:** [Requesting Party = D, Addressed Party = A,B]

When the serving node associated with Party A receives the above information flow, it then determines the route and the outgoing trunk facility. Since only one outgoing port is needed, and that Party A can accept the request, the serving node can also commit to the request and therefore issues the following information flow towards the selected relay node. The Network Connection Group is backward through connected.

5) **CALL-&-BEARER SETUP.ready:** [Requesting Party = D, Addressed Party = A,B]

The selected relay node validates the request and determines the route and outgoing trunk facility. Since only one outgoing port is needed, the serving node can commit to the request and therefore issues the following information flow towards the addressed serving node. The Network Connection Group is backward through connected.

6) **CALL-&-BEARER SETUP.ready:** [Requesting Party = D, Addressed Party = A,B]

The addressed serving node validates the request and selects the terminating interface facility. Since the interface is classified as a multiple signalling entity interface, the serving node can not commit to the addressed end point and therefore issues the following information flow towards the selected interface facility. The Network Connection Group is backward through connected.

7) **CALL-&-BEARER SETUP.begin:** [Requesting Party = D, Addressed Party = A,B]

The addressed terminal equipment determines that it can accept the request and issues the following information flow towards it's associated serving node.

8) **CALL-&-BEARER SETUP.ready**

The addressed serving node records the responses to the action request and selects of the responding terminals. The selected terminal is sent information flow number 6. The serving node then clears the non-selected terminals (note this action is not illustrated for simplicity), and issues information flow 7 toward the requesting relay node. The Network Connection Group is forward connected.

9) **CALL-&-BEARER SETUP.commit**

When the Terminal receives this information flow, it records the commitment, and connects in both the forward and backward directions.

10) **CALL-&-BEARER SETUP.commit**

When the relay node receives this information flow, it records the commitment, and relays this commitment to the requesting serving node by issuing information flow number 11, and performs forward through connect of the Network Connection Group.

11) **CALL-&-BEARER SETUP.commit**

When the service node associated with Party A receives this information flow, it records the commitment, and notifies this commitment to Party A's user equipment by issuing information flow number 12, and performs forward through connect of the Network Connection Group. In addition, it issues the notification of the completion of the remote request by issuing information flow 13 towards the requesting serving node associated with Party D.

12) **CALL-&-BEARER SETUP.commit**

When the user equipment receives this information flow, it records the commitment, and notifies the user of this commitment, and connects to the forward portion of the Network Connection Group.

13) **CALL-&-BEARER-SETUP-REMOTE.commit**

When the requesting service node receives this information flow, it records the commitment, and relays the commitment to the requesting Party (Party D) by issuing the following information flow.

14) **CALL-&-BEARER-SETUP.commit**

When the requesting party's user equipment receives this information flow, it records the commitment and notifies the user, thereby completing the requested action.

7.1.2 Call and Network Connection Group Establishment of Type 2 Network Connection Groups

Four example variations of this capability will be illustrated in this subclause. These four variations are as follows:

1) Call and Network Connection Group Establishment of a single Network Connection Group with branching occurring at the originating exchange and without network initiated "Look Ahead". The requesting Party is to be the "root" of all the Network Connections;
2) Call and Network Connection Group Establishment of a single Network Connection Group with branching occurring at the relay node and without network initiated "Look Ahead". The requesting Party is to be the "root" of all the Network Connections;
3) Call and Network Connection Group Establishment of a single Network Connection Group with branching occurring at the relay node and without network initiated "Look Ahead". The requesting Party is to be a "leaf of all the Network Connections;
4) Call and Network Connection Group Establishment of a single Network Connection Group with branching occurring at the relay node and without network initiated "Look Ahead". The requesting Party is not to be a attached to the Network Connections.

The overview of the Type 2 simultaneous call and Network Connection Group establishment capabilities are contained in the following subclauses.

7.1.2.1 Call and Network Connection Group Establishment Single Network Connection Group - Originating Node Branch Root Party

The User (Party A) requests a three party call between Party A, B, and Party C. One point-to-multi-point Network Connection Group is to be associated with this call. Parties A, B, and C are to be attached to the Network Connection Group. Party A is to be the "root" of the Network Connection Group. The User also specifies the Higher Layer service to be carried on this Network Connection Group and the desired Network Bearer service that should be established. The requested service is of the non-human interactive type. Therefore, immediate answer can be performed by both Party B's and Party C's equipment. If the Requested Parties equipment can accept the requested service, the designated attachment method, and specified bearer service, the equipment will indicate acceptance of the call and Network Connection Group request. This example also assumes that the Requested Parties are connected to a Multi-signalling entity interface. In addition, the network does not perform a "Look Ahead" procedure before progressing with the Network Connection Group establishment.

The signalling capability of simultaneously establishing this call and Network Connection Group between the three requested parties without network "Look Ahead" is illustrated below.

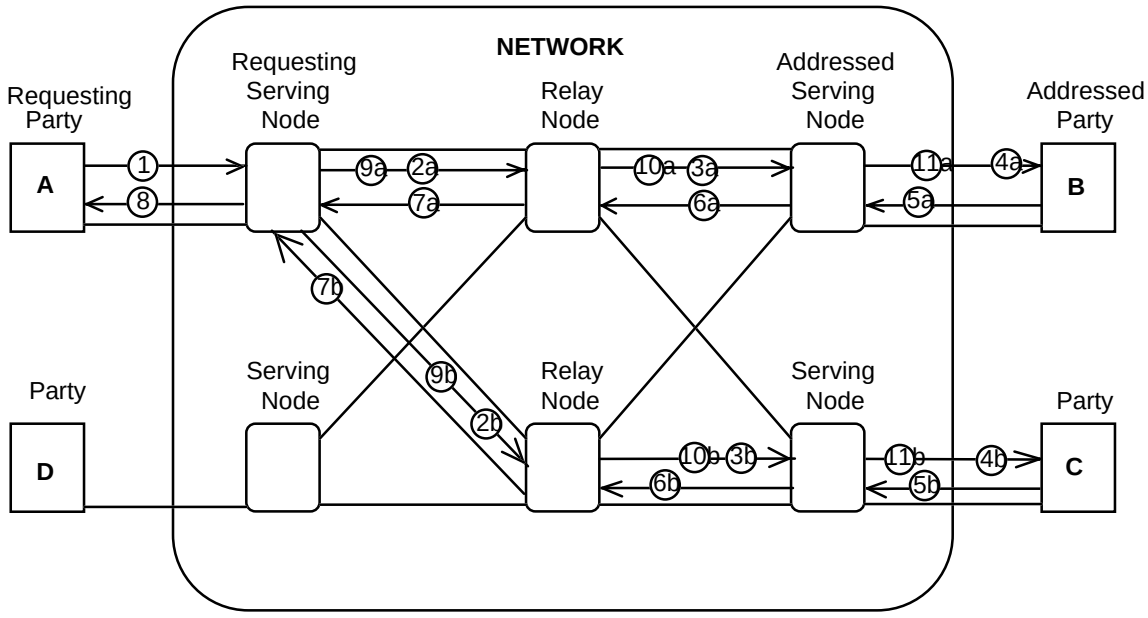


Figure 17: Type 2 Network Connection Group Between A, B, and C) - (no Look Ahead: Orig. Branch-Root)

The actions illustrated in the above figure are as follows.

Requesting Party's terminal equipment issues the following information flow towards it's serving node. The terminal equipment then attaches to the backward portion of Network Connection Group.

1) **CALL-&-BEARER SETUP.ready:** [Requesting Party = A, Addressed Party = B,C]

The requester's serving node validates the request and determines the route and outgoing trunk facility for the Network Connection Group. For this example, the Network Connection Group will be routed through separate relay nodes, two signalling ports are needed, the serving node can not commit to the request and therefore issues the following information flows towards the selected relay nodes. The Network Connection Group are backward through connected.

2a) **CALL-&-BEARER SETUP.begin:** [Requesting Party = A, Addressed Party = B,C]

2b) **CALL-&-BEARER SETUP.begin:** [Requesting Party = A, Addressed Party = B,C]

Each selected relay node validates the request and determines the route and outgoing trunk facility. Each selected relay node issues the following information flow towards the addressed serving nodes. The Network Connection Group in each relay node is backward through connected.

3a) **CALL-&-BEARER SETUP.begin:** [Requesting Party = A, Addressed Party = B,C]

3b) **CALL-&-BEARER SETUP.begin:** [Requesting Party = A, Addressed Party = B,C]

Each addressed serving node validates the information flow, selects the terminating interface facility. The serving node relays the following information flow towards the selected interface facility. The Network Connection Group is backward through connected.

4a) **CALL-&-BEARER SETUP.begin:** [Requesting Party = A, Addressed Party = B,C]

4b) **CALL-&-BEARER SETUP.begin:** [Requesting Party = A, Addressed Party = B,C]

The addressed terminal equipment determines that it can accept the request and issues the following information flow towards it's associated serving node.

5a) **CALL-&-BEARER SETUP.ready**

5b) **CALL-&-BEARER SETUP.ready**

The addressed service nodes receives the above flow and it records the responses to the action request and issues the following responses to the request from its associated two relay node

6a) **CALL-&-BEARER SETUP.ready**

6b) **CALL-&-BEARER SETUP.ready**

When the selected relay nodes receive the above responses it records them and relays the responses to the requesting service node in the form illustrated by the following information flows.

7a) **CALL-&-BEARER SETUP.ready**

7b) **CALL-&-BEARER SETUP.ready**

When the requesting service node receives these information flows, it records the willingness to accept the call and Network Connection Groups, and send commitment information flows towards the requesting user equipment (flow 8) and the relay nodes (flows 9a and 9b), and performs forward and backward through connect of the Network Connection Groups.

8) **CALL-&-BEARER SETUP.commit**

When the user equipment receives this information flow, it records the commitment, and notifies the user of this commitment, and connects to the forward portion of the Network Connection Group.

9a) **CALL-&-BEARER SETUP.commit**

9b) **CALL-&-BEARER SETUP.commit**

When the selected relay nodes receive the above information flows, it records the commitment, and relays this commitment to the addressed service nodes by issuing information flow number 10a and 10b, and performs forward and backward through connect of the Network Connection Group.

10a) **CALL-&-BEARER SETUP.commit**

10b) **CALL-&-BEARER SETUP.commit**

When the addressed service node receives these information flows, it records the commitment, selects one of the terminals that have answered request 4a or 4b, sends a commitment information flow to the selected terminal, and clears the other terminals that have responded to the requests. (for simplicity the clearing flows are not shown). The addressed serving node then through connects both Network Connection Groups in the forward and backward directions.

11a) **CALL-&-BEARER SETUP.commit**

11b) **CALL-&-BEARER SETUP.commit**

When the user equipment receives one of the above information flows, it records the commitment, and notifies the user of this commitment, and through connects in both directions.

7.1.2.2 Call and Network Connection Group Establishment Single Network Connection Group - Relay Node Branch Root

The User (Party A) requests a three party call between Party A, B, and Party C. One point-to-multi-point Network Connection Group is to be associated with this call. Parties A, B, and C are to be attached to the Network Connection Group. Party A is to be the "root" of the Network Connection Group. The User also specifies the Higher Layer service to be carried on this Network Connection Group and the desired Network Bearer service that should be established. The requested service is of the non-human interactive type. Therefore, immediate answer can be performed by both Party B's and Party C's equipment. If the Requested Parties equipment can accept the requested service, the designated attachment method, and specified bearer service, the equipment will indicate acceptance of the call and Network Connection Group request. This example also assumes that the Requested Parties are connected to a Multi-signalling entity interface. In addition, the network does not perform a "Look Ahead" procedure before progressing with the Network Connection Group establishment.

The signalling capability of simultaneously establishing this call and Network Connection Group between the three requested parties without network "Look Ahead" is illustrated below.

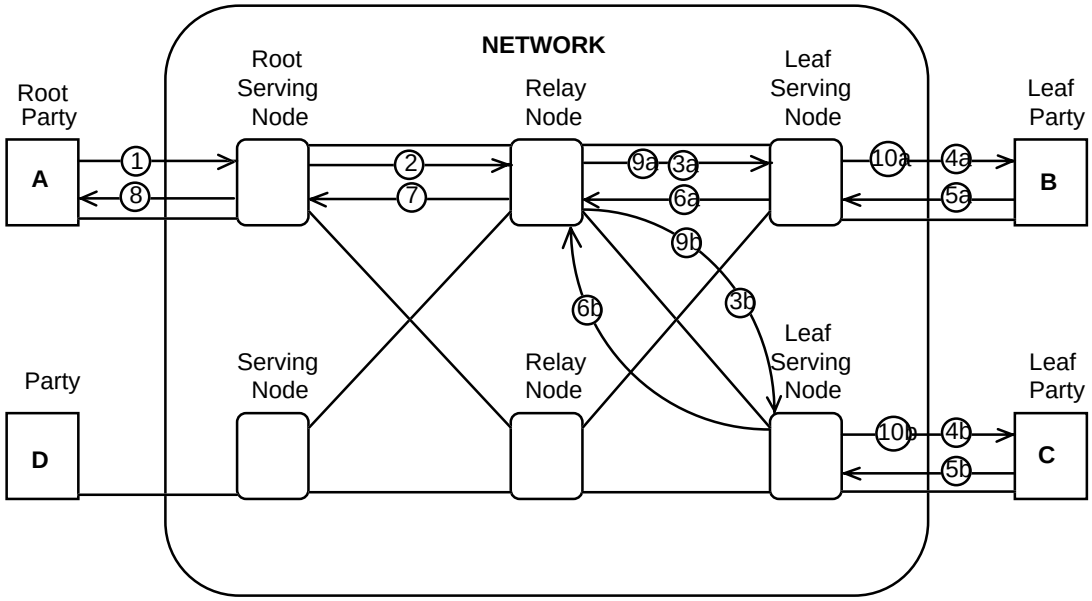


Figure 18: Type 2 Network Connection Group Between A, B, and C) - (no Look Ahead: Relay Branch-Root)

The actions illustrated in the above figure are as follows.

Requesting Party's terminal equipment issues the following information flow towards it's serving node. The terminal equipment then attaches to the backward portion of Network Connection Group.

1) **CALL-&-BEARER SETUP.ready:** [Requesting Party = A, Addressed Party = B,C]

The requester's serving node validates the request and determines the route and outgoing trunk facility for the Network Connection Group. For this example, the Network Connection Group will be routed through single relay node, the serving node can commit to the request and therefore issues the following information flows towards the selected relay node. The Network Connection Group are backward through connected.

2) **CALL-&-BEARER SETUP.ready:** [Requesting Party = A, Addressed Party = B,C]

The selected relay node validates the request and determines the route and outgoing trunk facility. As a result of this routeing, two separate routes are required to get to Parties B and C. The selected relay node issues the following information flows towards the addressed serving nodes. The Network Connection Group in the relay node is backward through connected.

3a) **CALL-&-BEARER SETUP.begin:** [Requesting Party = A, Addressed Party = B,C]

3b) **CALL-&-BEARER SETUP.begin:** [Requesting Party = A, Addressed Party = B,C]

Each addressed serving node validates the information flow and selects the terminating interface facility. The serving node relays the following information flow towards the selected interface facility. The Network Connection Group is backward through connected.

4a) **CALL-&-BEARER SETUP.begin:** [Requesting Party = A, Addressed Party = B,C]

4b) **CALL-&-BEARER SETUP.begin:** [Requesting Party = A, Addressed Party = B,C]

The addressed terminal equipment determines that it can accept the request and issues the following information flow towards it's associated serving node.

5a) **CALL-&-BEARER SETUP.ready**

5b) **CALL-&-BEARER SETUP.ready**

The addressed service nodes receives the above flow and it records the responses to the action request and issues the following responses to the request from its associated relay node

6a) **CALL-&-BEARER SETUP.ready**

6b) **CALL-&-BEARER SETUP.ready**

When the selected relay node receive the above responses it records them and relays the responses to the associated service nodes in the form illustrated by the following information flows.

7) **CALL-&-BEARER SETUP.Commit**

9a) **CALL-&-BEARER SETUP.commit**

9b) **CALL-&-BEARER SETUP.commit**

When the requesting service node receives these information flows, it records the willingness to accept the call and Network Connection Groups, and send commitment information flows towards the requesting user equipment (flow 8) and the relay nodes (flows 9a and 9b), and performs forward and backward through connect of the Network Connection Groups.

8) **CALL-&-BEARER SETUP.commit**

When the user equipment receives this information flow, it records the commitment, and notifies the user of this commitment, and connects to the forward portion of the Network Connection Group.

When the addressed service nodes receives either information flows 9a or 9b, it records the commitment, selects one of the terminals that have answered request 4a or 4b, sends a commitment information flow to the selected terminal, and clears the other terminals that have responded to the requests. (for simplicity the clearing flows are not shown). The addressed serving node then through connects both Network Connection Groups in the forward and backward directions.

10a) **CALL-&-BEARER SETUP.commit**

10b) **CALL-&-BEARER SETUP.commit**

When the user equipment receives one of the above information flows, it records the commitment, and notifies the user of this commitment, and through connects in both directions.

7.1.2.3 Call and Network Connection Group Establishment Single Network Connection Group - Relay Node Branch Leaf

The User (Party B) requests a three party call between Party A, B, and Party C. One point-to-multi-point Network Connection Group is to be associated with this call. Parties A, B, and C are to be attached to the Network Connection Group. Party A is to be the "root" of the Network Connection Group. The User also specifies the Higher Layer service to be carried on this Network Connection Group and the desired Network Bearer service that should be established. The requested service is of the non-human interactive type. Therefore, immediate answer can be performed by both Party A's and Party C's equipment. If the Requested Parties equipment can accept the requested service, the designated attachment method, and specified bearer service, the equipment will indicate acceptance of the call and Network Connection Group request. This example also assumes that the Requested Parties are connected to a Multi-signalling entity interface. In addition, the network does not perform a "Look Ahead" procedure before progressing with the Network Connection Group establishment.

The signalling capability of simultaneously establishing this call and Network Connection Group between the three requested parties without network "Look Ahead" is illustrated below.

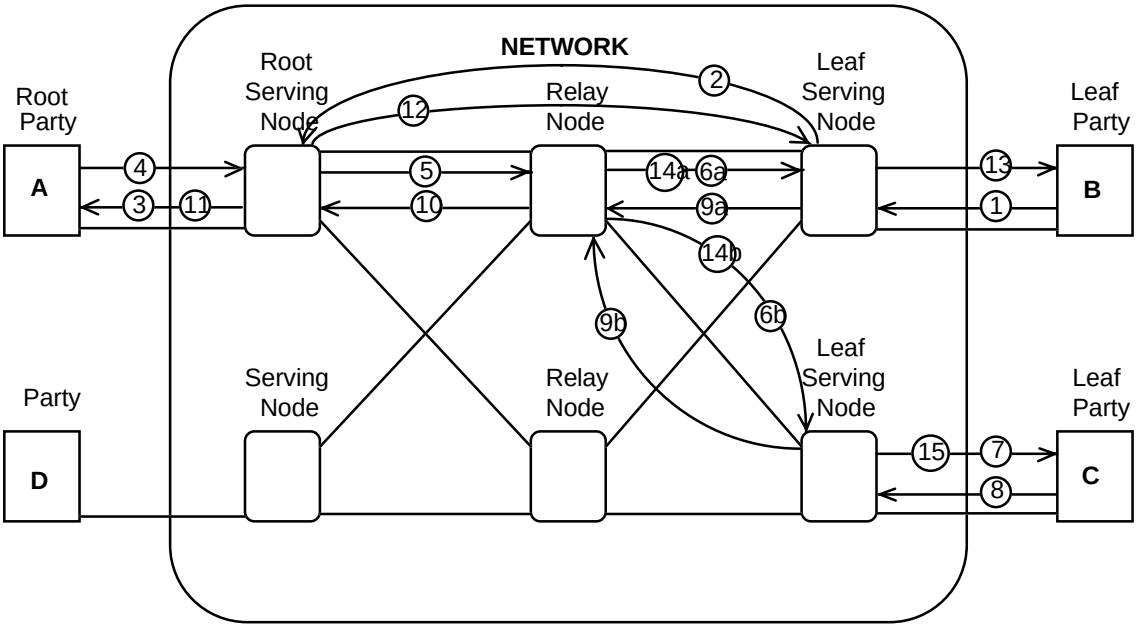


Figure 19: Type 2 Network Connection Group Between A, B, and C) - (no Look Ahead: Relay Branch-Leaf)

The actions illustrated in the above figure are as follows.

Requesting Party B's terminal equipment issues the following information flow towards it's serving node. The terminal equipment then attaches to the backward portion of Network Connection Group.

1) **CALL-&-BEARER SETUP.ready:** [Requesting Party = B, Addressed Party = A,C]

The requester's serving node validates the request and determines the route to the serving node associated with the Network Connection Group. Since the root of the Network Connection Group is located in another serving node, a Remote request needs to be invoked. The following information flow would be issued towards the serving node associated with Party A.

2) **CALL-&-BEARER SETUP-REMOTE.ready:** [Requesting Party = B, Addressed Party = A,C]

When the above information flow is received by the "Root" serving node, it will validate the request. The serving node will determine the interface which is associated with Party A and issue the following information flow.

3) **CALL-&-BEARER SETUP.begin:** [Requesting Party = B, Addressed Party = A,C]

Party A's terminal equipment determines that it can accept the request and issues the following information flow towards it's associated serving node.

4) **CALL-&-BEARER SETUP.ready**

When the serving node associated with Party A receives the above information flow, it records Party A willingness to accept the call and Network Connection Group, determines the route to the requested Parties B and C. For this example, the Network Connection Group will be routed through a single relay node, the serving node can commit to the request and therefore issues the following information flows towards the selected relay node. The Network Connection Group is backward through connected.

5) **CALL-&-BEARER SETUP.ready:** [Requesting Party = B, Addressed Party = A,C]

The selected relay node validates the request and determines the route and outgoing trunk facility. As a result of this routeing, two separate routes are required to get to Parties B and C. The selected relay node issues the following information flows towards the addressed serving nodes. The Network Connection Group in the relay node is backward through connected.

6a) **CALL-&-BEARER SETUP.begin:** [Requesting Party = B, Addressed Party = A,C]

6b) **CALL-&-BEARER SETUP.begin:** [Requesting Party = B, Addressed Party = A,C]

When the serving node associated with Party B receives the information flow 6a, it associates this flow with the remote action request illustrated as flow 2. It records the reception of this flow and issues the flowing information flow response towards the requesting relay node. The Network Connection Group is backward through connected.

9a) **CALL-&-BEARER SETUP.ready**

When the serving node associated with Party C receives the flow 6b, it validates the information flow and selects the terminating interface facility. The serving node relays the following information flow towards the selected interface facility. The Network Connection Group is backward through connected.

7) **CALL-&-BEARER SETUP.begin:** [Requesting Party = B, Addressed Party = A,C]

Party C's terminal equipment determines that it can accept the request and issues the following information flow towards it's associated serving node.

8) **CALL-&-BEARER SETUP.ready**

The addressed service node associated with Party C receives the above flow and it records the responses to the action request and issues the following responses to the request from its associated relay node.

9b) **CALL-&-BEARER SETUP.ready**

When the selected relay node receive the information flow responses 9a and 9b it records them and relays the responses to the associated service nodes in the form illustrated by the following information flows.

10) **CALL-&-BEARER SETUP.commit**

When the service node associated with Party A receives information flow 10, it records the willingness to accept the call and Network Connection Groups, and send commitment information flows towards the requesting service node (flow 12) and Party A's terminal equipment (flow 11), and performs forward and backward through connect of the Network Connection Groups.

11) **CALL-&-BEARER SETUP.commit**

12) **CALL-&-BEARER SETUP-REMOTE.commit**

When the Party A's terminal equipment receives this information flow, it records the commitment, and notifies the user of this commitment, and connects to the forward portion of the Network Connection Group.

When the service node associated with Party B receives information flow 14a, it records the commitment, and relays the commitment to the requesting Party B that the service request was completed. However, the action state machine is not cleared until information flow 12 is received. The information flow sent to Party B is as follows;

13) **CALL-&-BEARER SETUP.commit**

When the Leaf service node associated with Party C receives information flow 14b, it records the commitment, selects one of the terminals that have answered request 7, sends a commitment information flow to the selected terminal, and clears the other terminals that have responded to the requests. (for simplicity the clearing flows are not shown). The addressed serving node then through connects both Network Connection Groups in the forward and backward directions.

14a) **CALL-&-BEARER SETUP.commit**

14b) **CALL-&-BEARER SETUP.commit**

15) **CALL-&-BEARER SETUP.commit**

When the user equipment receives either information flow 13 or 15, it records the commitment, and notifies the user of this commitment, and through connects in both directions.

7.1.2.4 Call and Network Connection Group Establishment Single Network Connection Group - Relay Node Branch Third

The User (Party D) requests a three party call between Party A, B, C, and Party D. One point-to-multi-point Network Connection Group is to be associated with this call. Parties A, B, and C are to be attached to the Network Connection Group. Party A is to be the "root" of the Network Connection Group. The User also specifies the Higher Layer service to be carried on this Network Connection Group and the desired Network Bearer service that should be established. The requested service is of the non-human interactive type. Therefore, immediate answer can be performed by Party A's Party B's, and Party C's equipment. If the Requested Parties equipment can accept the requested service, the designated attachment method, and specified bearer service, the equipment will indicate acceptance of the call and Network Connection Group request. This example also assumes that the Requested Parties are connected to a Multi-signalling entity interface. In addition, the network does not perform a "Look Ahead" procedure before progressing with the Network Connection Group establishment.

The signalling capability of simultaneously establishing this call and Network Connection Group between the three requested parties without network "Look Ahead" is illustrated below.

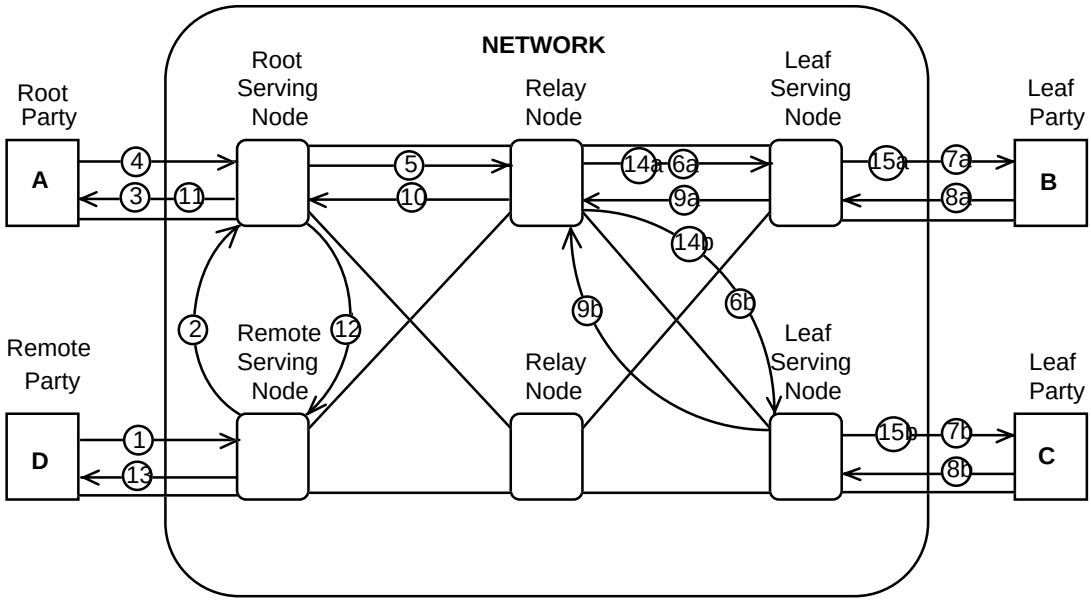


Figure 20: Type 2 Network Connection Group Between A, B, and C) - (no Look Ahead: Relay Branch-Third)

The actions illustrated in figure 20 are as follows.

Requesting Party D's terminal equipment and issues the following information flow towards it's serving node. The terminal equipment then attaches to the backward portion of Network Connection Group.

1) **CALL-&-BEARER SETUP.ready:** [Requesting Party = D, Addressed Party = A, B, C]

The requester's serving node validates the request and determines the route to the serving node associated with the "root" of the Network Connection Group. Since the root of the Network Connection Group is located in another serving node, a Remote request needs to be invoked. The following information flow would be issued towards the serving node associated with Party A.

2) **CALL-&-BEARER SETUP-REMOTE.ready:** [Requesting Party = D, Addressed Party = A, B, C]

When the above information flow is received by the "Root" serving node, it will validate the request. The serving node will determine the interface which is associated with Party A and issue the following information flow.

3) **CALL-&-BEARER SETUP.begin:** [Requesting Party = D, Addressed Party = A, B, C]

Party A's terminal equipment determines that it can accept the request and issues the following information flow towards it's associated serving node.

4) **CALL-&-BEARER SETUP.ready**

When the serving node associated with Party A receives the above information flow, it records Party A willingness to accept the call and Network Connection Group, determines the route to the requested Parties B and C. For this example, the Network Connection Group will be routed through a single relay node, the serving node can commit to the request and therefore issues the following information flows towards the selected relay node. The Network Connection Group is backward through connected.

5) **CALL-&-BEARER SETUP.ready:** [Requesting Party = D, Addressed Party = A, B, C]

The selected relay node validates the requests and determines the route and outgoing trunk facility. As a result of this routeing, two separate routes are required to get to Parties B and C. The selected relay node issues the following information flows towards the addressed serving nodes. The Network Connection Group in the relay node is backward through connected.

6a) **CALL-&-BEARER SETUP.begin:** [Requesting Party = D, Addressed Party = A, B, C]

6b) **CALL-&-BEARER SETUP.begin:** [Requesting Party = D, Addressed Party = A, B, C]

When the serving nodes associated with Party B and C receive the above flows, they validates the information flow and select the terminating interface facilities associated with Party B and Party C. The serving nodes relay the following information flows toward the selected interface facilities. The Network Connection Group is backward through connected.

7a) **CALL-&-BEARER SETUP.begin:** [Requesting Party = D, Addressed Party = A, B, C]

7b) **CALL-&-BEARER SETUP.begin:** [Requesting Party = D, Addressed Party = A, B, C]

Party B's and Party C's terminal equipment determine that they can accept the request and both issue the following information flows towards their associated serving node.

8a) **CALL-&-BEARER SETUP.ready**

8b) **CALL-&-BEARER SETUP.ready**

The addressed service nodes associated with Party B and Party C receive the above flows and they record the response to the action request and issue the following response to the request from its associated relay node

9a) **CALL-&-BEARER SETUP.ready**

9b) **CALL-&-BEARER SETUP.ready**

When the selected relay node receive the information flow responses 9a and 9b it records them and relays the responses to the associated service nodes in the form illustrated by the following information flows.

10) **CALL-&-BEARER SETUP.commit**

When the remote service node associated with Party A receives information flow 10, it records the willingness to accept the call and the Network Connection Group, and send commitment information flows towards the requesting service node (flow 12) and Party A's terminal equipment (flow 11), and performs forward and backward through connect of the Network Connection Groups.

11) **CALL-&-BEARER SETUP.commit**

12) **CALL-&-BEARER SETUP-REMOTE.commit**

When the Party A's terminal equipment receives this information flow, it records the commitment, and notifies the user of this commitment, and connects to the forward portion of the Network Connection Group.

When the service node associated with Party D receives information flow 12, it records the commitment, and relays the commitment to the requesting Party D that the service request was completed. The information flow sent to Party D is as follows;

13) **CALL-&-BEARER SETUP.commit**

When the Leaf service nodes associated with Party B and C receive information flows 14a or 14b, they record the commitment, select one of the terminals that have answered request 7a or 7b, send a commitment information flow to the selected terminal, and clears the other terminals on that interface that have responded to the request. (for simplicity the clearing flows are not shown). The Leaf serving nodes then through connects both Network Connection Groups in the forward and backward directions.

14a) **CALL-&-BEARER SETUP.commit**

14b) **CALL-&-BEARER SETUP.commit**

15a) **CALL-&-BEARER SETUP.commit**

15b) **CALL-&-BEARER SETUP.commit**

When the user equipment receives either information flow 15a or 15b, it records the commitment, notifies the user of this commitment, and through connects in both directions.

7.2 Addition of a new Network Connection (Group) to an existing Call with attachment of existing Parties to the Network Connection (Group)

The addition of and one or more Network Connection Groups to an existing call can be separated into several categories such as those that are associated with Type 1 and Type 2 Network Connection Groups. For Signalling Capability Set 2 the following categories have been agreed upon:

1) Addition of a Type 1 Network Connection Group to an existing call;
2) Addition of a Type 2 Network Connection Group to an existing call.

The following subclause describes these signalling services.

7.2.1 Addition of a Type 1 Network Connection Group to an existing call

Three example variations of this capability will be illustrated in this subclause. These three variations are as follows:

1) addition of an Network Connection Group to an existing Call without network "Look Ahead" and without Notifying the other parties associated with the call;

2) addition of an Network Connection Group to an existing Call without network "Look Ahead" and with the notification of the Bearer Change to the other Parties associated with the Call; and
3) addition of an Network Connection Group to an existing Call requested by a third Party associated with the Call without network "Look Ahead" and without the Notification of the other Parties associated with the call.

The overview of the capabilities for addition of a Type 1 Network Connection Group to an existing call are contained in the following subclauses.

7.2.1.1 Addition of one point-to-point Network Connection Group requested by the Call Initiating Party

The user (Party A) requests an addition of one point-to-point Network Connection group with one single connection between itself and the Party B. Parties A and B are to be attached to the Network Connection. The user also specifies the High Layer service to be carried on this connection and the desired Bearer service that should be established. The requested service is of the human interactive type. No immediate answer can be performed by the Party B's terminal. If the requested Party B's terminal can accept the requested service it will indicate user alerting. If the human user can accept the requested service also, then the terminal will indicate acceptance of the connection request. The network does not perform a Look Ahead procedure before progressing with the connection establishment.

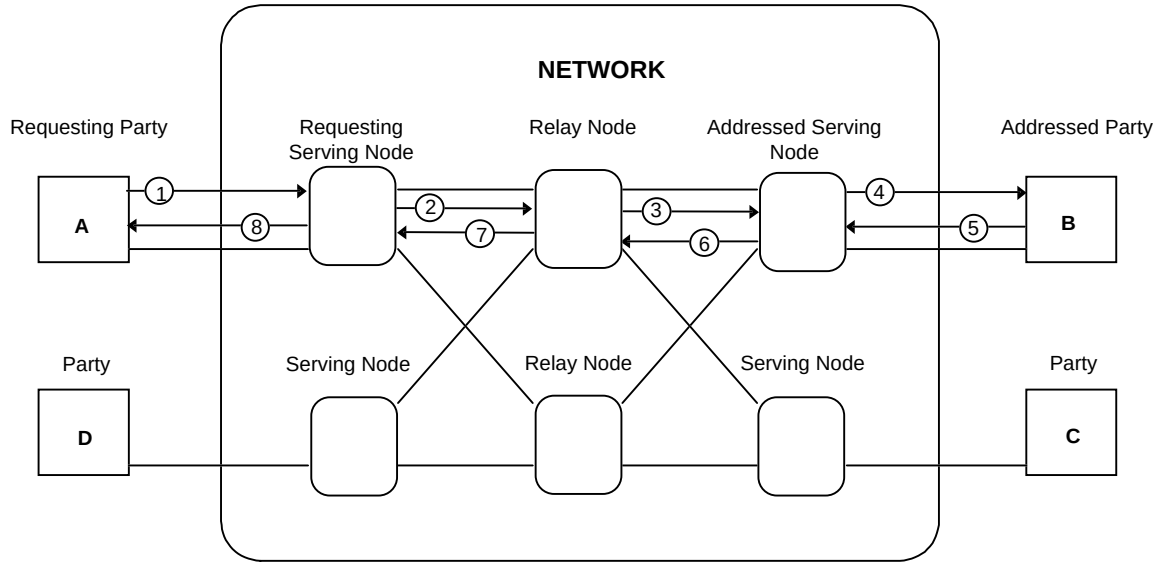


Figure 21: Addition of one point-to-point Network Connection Group to an existing Call

The actions illustrated in the above figure are as follows:

After accepting the request from the user the terminal issues the following information flow towards its serving node.

1) **ADD-BEARER-TO-CALL.ready:** [Requesting Party = A, Addressed Party = B]

The requesting serving node validates the request and determines the route and outgoing trunk facility of the Network Connection group. Since only one outgoing connection is needed, the serving node commits to the request and therefore issues the following information flow towards the selected relay node. The connection may be backward through connected.

2) **ADD-BEARER-TO-CALL.ready:** [Requesting Party = A, Addressed Party = B]

The selected relay node validates the request and determines the route and outgoing trunk facility of the network group. Since only one outgoing connection is needed, the relay node commits to the request and therefore issues the following information flow towards the selected serving node. The connection may be through connected in both directions.

3) **ADD-BEARER-TO-CALL.ready:** [Requesting Party = A, Addressed Party = B]

The addressed serving node validates the request and selects the terminating interface. Since only one outgoing connection is needed, the addressed serving node commits to the request and therefore issues the following information flow towards the selected interface.

4) **ADD-BEARER-TO-CALL.ready:** [Requesting Party = A, Addressed Party = B]

The selected terminal determines that it can commit to the request and issues the following information flow towards the addressed serving node. The terminal then connects to the connection.

5) **ADD-BEARER-TO-CALL.commit**

When the addressed serving node receives this information flow, it records commitment and issues the following information flow towards the requesting relay node. The connection is through connected in both directions.

6) **ADD-BEARER-TO-CALL.commit**

When the relay node receives this information flow, it records commitment and issues the following information flow towards the requesting serving node. The connection is through connected in both directions if not already done.

7) **ADD-BEARER-TO-CALL.commit**

When the requesting serving node receives this information flow, it records commitment and issues the following information flow towards the requesting terminal. The connection is through connected in both directions if not already done.

8) **ADD-BEARER-TO-CALL.commit**

When the requesting terminal receives this information flow, it records commitment and notifies the user of this commitment. The terminal then connects in both directions if not already done.

7.2.1.2 Addition of an Network Connection Group to an existing Call - Without "Look Ahead" and with Notify - With Establishment of a new Route in the Network

The User (Party A) requests that an additional Network Connection Group be added to the call. The request is for a point-to-point Network Connection Group between Party A and Party B. The requested service is of the non-human interactive type. Therefore, immediate answer can be performed by the Party B's equipment. If the Addressed Party B equipment can accept the requested service, the equipment will indicate acceptance of the Network Connection Group request. This example also assumes that the Addressed Party is connected to a Multi-signalling entity interface. In addition, the network does not perform a "Look Ahead" procedure before progressing with the Network Connection Group establishment and does notify other serving nodes and Parties already associated with the call that a new Network Connection Group has been added. The Notify procedure is service dependent and this example assumes that the service does require notification whenever a new Network Connection Group is added to the call. For this example, assume that Parties C and D are associated with the call. The route of this new Network Connection Group is assumed to be through a Relay Node that has not been associated with the call.

The signalling capability of adding an additional Network Connection Group between two parties without network "Look Ahead" and with notify is illustrated below.

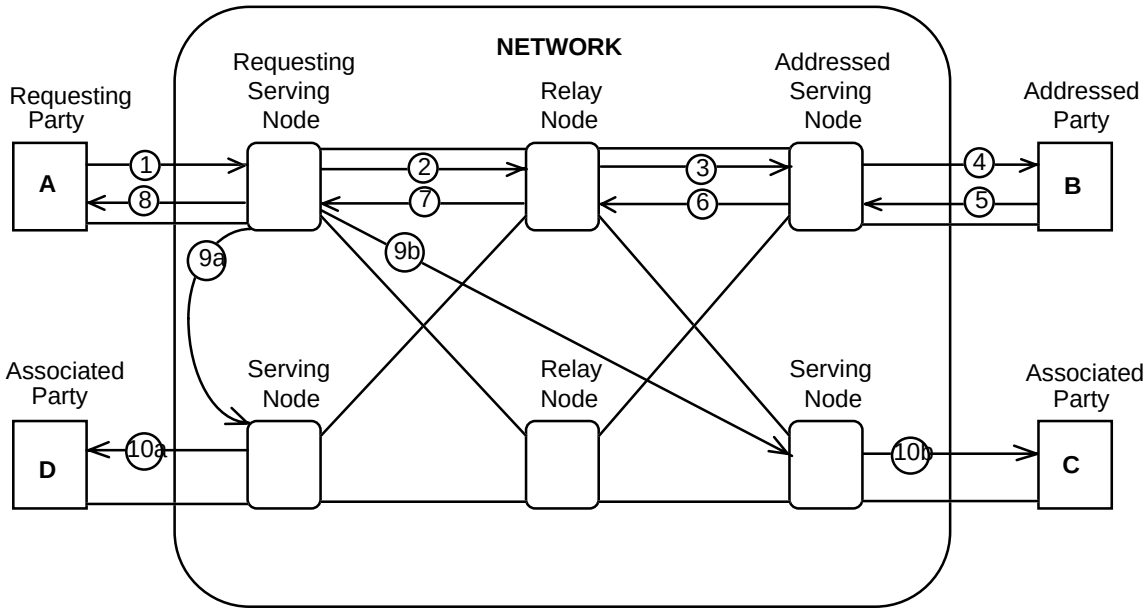


Figure 22: Addition of a Point-to-point Network Connection Group (Type 1 Between A and B) - (no Look Ahead, with Notify) (Network Connection Group routed through a Relay Node not associated with the call)

The actions illustrated in the above figure are as follows.

Requesting Party's terminal equipment issues the following information flow towards its serving node. The terminal equipment then attaches to the backward portion of the Network Connection Group.

1) **ADD-BEARER-TO-CALL.ready:** [Requesting Party = A, Addressed Party = B]

The requester's serving node validates the request and determines the route and outgoing trunk facility. Since only one outgoing port is needed, the serving node can commit to the request and since the Relay Node is not already associated with the call, the serving node issues the following information flow towards the selected relay node. The Network Connection Group is backward through connected.

2) **CALL-&-BEARER SETUP.ready:** [Requesting Party = A, Addressed Party = B]

The selected relay node validates the request and determines the route and outgoing trunk facility. Since only one outgoing port is needed, the Relay node can commit to the request and therefore issues the following information flow towards the addressed serving node. The Network Connection Group is backward through connected.

3) **CALL-&-BEARER SETUP.ready:** [Requesting Party = A, Addressed Party = B]

The addressed serving node determines that the received request is associated with a call that is already in progress and proceeds to validate the request and issue the following information flow to end point already associated with the call. Since the information flow is offered to a single device, the serving node can commit to the addressed end point and therefore issues the following information flow towards the selected interface facility. The Network Connection Group is backward through connected.

4) **ADD-BEARER-TO-CALL.ready:** [Requesting Party = A, Addressed Party = B]

The addressed terminal equipment determines that it can accept the requested action and issues the following information flow towards it's associated serving node and connects in both the forward and backward directions.

5) **ADD-BEARER-TO-CALL.commit**

The addressed serving node records the response to the action request and completes the action by sending the following information flow. The Network Connection Group is forward connected.

6) **CALL-&-BEARER SETUP.commit**

When the relay node receives this information flow, it records the commitment, and relays this commitment to the requesting serving node by issuing information flow number 7, and performs forward through connect of the Network Connection Group.

7) **CALL-&-BEARER SETUP.commit**

When the requesting service node receives this information flow, it records the commitment, and relays this commitment to the requesting user equipment by issuing information flow number 8, and performs forward through connect of the Network Connection Group. In addition the requesting service node determines that a service requirement exists to notify other Parties associated with the call that a new Network Connection Group has been added to the call. In this example, Parties C and D are associated with the call. The serving node issues information flows 9a and 9b.

8) **ADD-BEARER-TO-CALL.commit**

9a) **NOTIFY-BEARER-CHANGE.indication:** [Network Connection Group Added]

9b) **NOTIFY-BEARER-CHANGE.indication:** [Network Connection Group Added]

When the user equipment receives information flow number 8, it records the commitment, and notifies the user of this commitment, and connects to the forward portion of the Network Connection Group.

When the serving nodes associated with Parties C and D receive information flows 9b and 9a, they record the addition of the Network Connection Group and issue information flows 10b and 10a respectively. (Note: Service Requirements could state that they is no need to notify end equipment of the addition, however, this example assumes that the end equipment will be notified. If there is no need to notify the end points, information flows 10a and 10b would not be issued.)

10a) **NOTIFY-BEARER-CHANGE.indication:** [Network Connection Group Added]

10b) **NOTIFY-BEARER-CHANGE.indication:** [Network Connection Group Added]

When the end equipment associated with parties C and D receive information flows 10b and 10a, the equipment will record the addition of a new Network Connection Group.

7.2.1.3 Addition of an Network Connection Group to an existing Call - Third Party

The User (Party D) requests that an Network Connection Group be added to the call. This Network Connection Group is to connect Parties A and B. The requested service is of the non-human interactive type. Therefore, immediate answer can be performed by the Party A's and Party B's equipment. If both the Addressed Parties equipment can accept the requested service, the equipment will indicate acceptance of the Network Connection Group request. This example also assumes that both Addressed Parties are connected to a Multi-signalling entity interface. In addition, the network does not perform a "Look Ahead" procedure before progressing with the Network Connection Group establishment.

The signalling capability of adding a point-to-point Network Connection Group between two parties already associated with the call requested by a third party is illustrated below. This request was issued by Party D that will not be attached to the request Network Connection Group. The example will not invoke the network "Look Ahead" procedure. In addition, the other parties associated with the call will not be notified (Service Requirement Assumption).

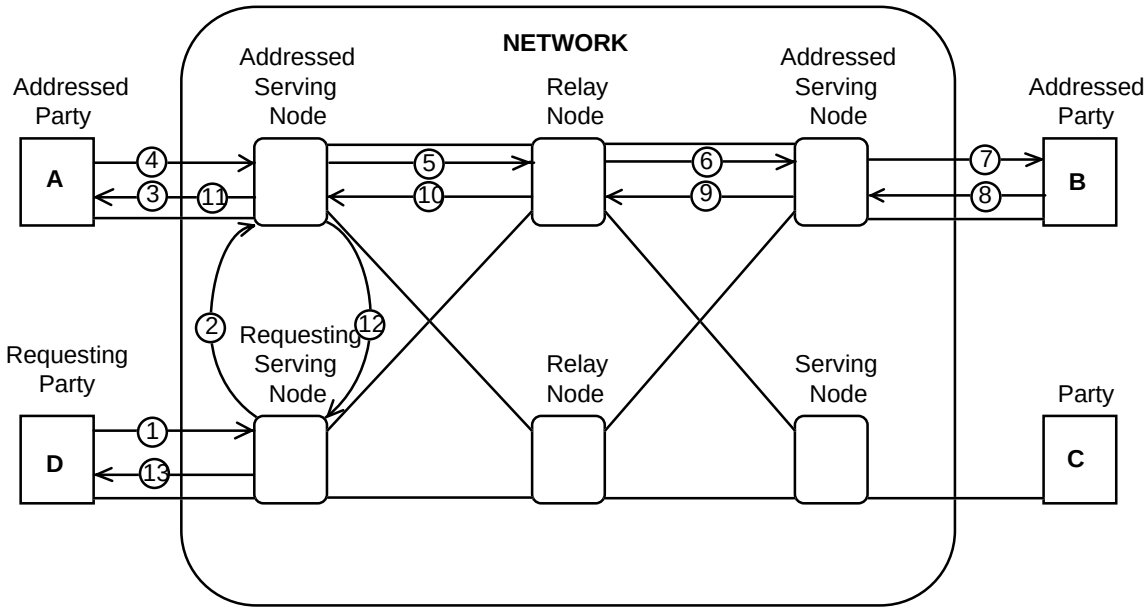


Figure 23: Addition of Point-to-point Network Connection Group (Type 1 Between A and B) Third Party

The actions illustrated in the above figure are as follows.

Requesting Party's (Party D) terminal equipment issues the following information flow towards its serving node.

1) **ADD-BEARER-TO-CALL.ready:** [Requesting Party = D, Addressed Party = A, B]

The requester's serving node validates the request and determines which party will be designated the "root" Party for this Network Connection Group (for this example Party A is chosen) and the edge signalling route to the serving node associated with selected "root" Party. Since Party D is not attached to the requested Network Connection Group, and the "root" of the Network Connection Group is located in another serving node, a Remote operation request needs to be invoked. In addition, only one outgoing signalling port is needed, Party D's serving node can commit to the request and therefore issues the following information flow towards the selected "root" serving node.

2) **ADD-BEARER-TO-CALL-REMOTE.ready:** [Requesting Party = D, Addressed Party = A, B]

The selected serving node validates the request and determines the interface associated with Party A. Even though Party A is already a member of the call, the Network Connection Group shall be established only after Party B indicates that it is willing to commit to the Network Connection Group, the serving node can not commit to the request and therefore issues the following information flow towards the addressed party (Party A).

3) **ADD-BEARER-TO-CALL.begin:** [Requesting Party = D, Addressed Party = A, B]

When Party A receives the above information flow, it determines if it can accept the request contained in the flow. If it can accept the call and Network Connection Group, it responses with the following information flow. The terminal equipment then attaches to the backward portion of the Network Connection Group.

4) **ADD-BEARER-TO-CALL.ready**

When the serving node associated with Party A receives the above information flow, it then determines the route and the outgoing trunk facility. Since only one outgoing port is needed, and that Party A can accept the request, the serving node can also commit to the request and therefore issues the following information flow towards the selected relay node that is already associated with the call. The Network Connection Group is backward through connected.

5) **ADD-BEARER-TO-CALL.ready:** [Requesting Party = D, Addressed Party = A, B]

The selected relay node validates the request and determines the route and outgoing trunk facility. Since only one outgoing port is needed, the serving node can commit to the request and therefore issues the following information flow towards the addressed serving node. The Network Connection Group is backward through connected.

6) **ADD-BEARER-TO-CALL.ready:** [Requesting Party = D, Addressed Party = A, B]

The addressed serving node validates the request and issues the following information flow to the end point already associated with the call. Since the information flow is offered to a single device, the serving node can commit to the addressed end point and therefore issues the following information flow towards the selected interface facility. The Network Connection Group is backward through connected.

7) **ADD-BEARER-TO-CALL.ready:** [Requesting Party = D, Addressed Party = A, B]

The addressed terminal equipment determines that it can accept the requested action and issues the following information flow towards it's associated serving node and connects in both the forward and backward directions.

8) **ADD-BEARER-TO-CALL.commit**

The addressed serving node records the response to the action request and completes the action by sending the following information flow. The Network Connection Group is forward connected.

9) **ADD-BEARER-TO-CALL.commit**

When the relay node receives this information flow, it records the commitment, and relays this commitment to the requesting serving node by issuing information flow number 10, and performs forward through connect of the Network Connection Group.

10) **ADD-BEARER-TO-CALL.commit**

When the service node associated with Party A receives this information flow, it records the commitment, and notifies this commitment to Party A's user equipment by issuing information flow number 11, and performs forward through connect of the Network Connection Group. In addition, it issues the notification of the completion of the remote request by issuing information flow 12 towards the requesting serving node associated with Party D.

11) **ADD-BEARER-TO-CALL.commit**

When the user equipment receives this information flow, it records the commitment, and notifies the user of this commitment, and connects to the forward portion of the Network Connection Group.

12) **ADD-BEARER-TO-CALL-REMOTE.commit**

When the requesting service node receives this information flow, it records the commitment, and relays the commitment to the requesting Party (Party D) by issuing the following information flow.

13) **ADD-BEARER-TO-CALL.commit**

When the requesting party's user equipment receives this information flow, it records the commitment and notifies the user, thereby completing the requested action.

7.2.2 Addition of a Type 2 Network Connection Group to an existing call

Two example variations of this capability will be illustrated in this subclause. These two variations are as follows:

1) Addition of a single Network Connection Group or common route Network Connection Group with branching occurring at the originating exchange and without network initiated "Look Ahead". The requesting Party is to be the "root" of the Network Connection Group; and
2) Addition of a single Network Connection Group or common route Network Connection Group with branching occurring at the relay node and without network initiated "Look Ahead". The requesting Party is to be the "root" of the Network Connection Group.

The overview of the Type 2 simultaneous call and Network Connection Group establishment capabilities are contained in the following subclauses.

7.2.2.1 Addition of an Network Connection Group to an existing Call - Requesting Serving Node Branch Root Party

The User (Party A) requests that a new point-to-multi-point Network Connection Group between Parties A, B, and C be added to an existing call. Party A is to be the "root" of the Network Connection Group The requested service is of the non-human interactive type. Therefore, immediate answer can be performed by both Party B's and Party C's equipment. If the Addressed Parties equipment can accept the requested service, the equipment will indicate acceptance of the call and Network Connection Group request. This example also assumes that the Addressed Parties are connected to a Multi-signalling entity interface. In addition, the network does not perform a "Look Ahead" procedure before progressing with the Network Connection Group establishment. The example assumes that Relay Node R-1 is already associated with the call while Relay Node R-2 is not associated with the call. The example also assumes that Party D is party of the call and the service requirements specifies that all changes to the call shall be notified to all Parties associated with the Call.

The signalling capability of adding a point-to-multi-point Network Connection Group to an existing call without network "Look Ahead" is illustrated below.

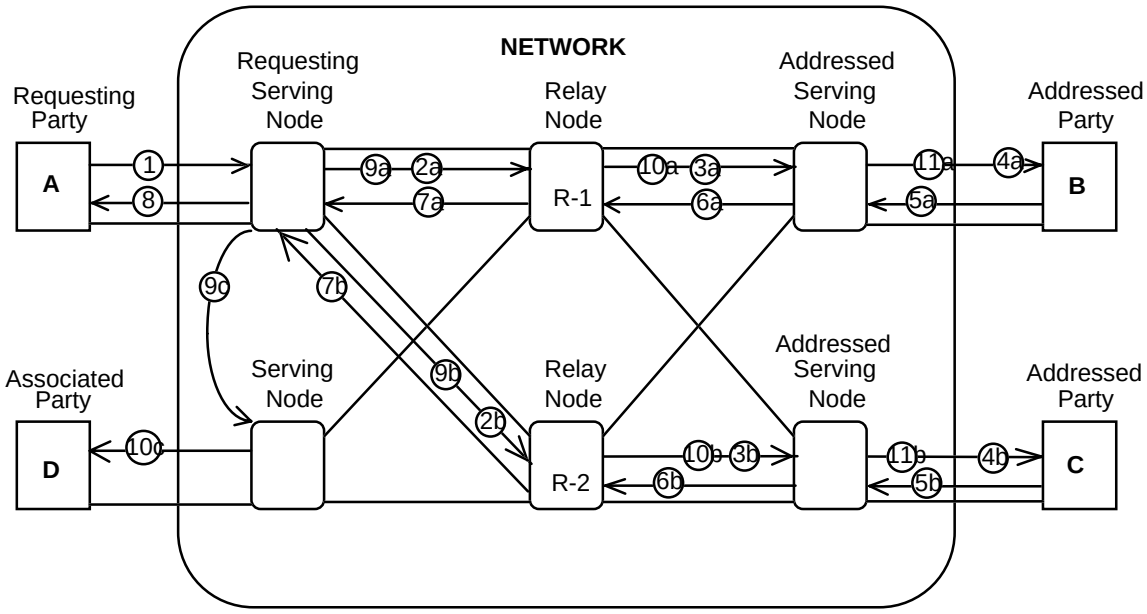


Figure 24: Addition of Type 2 Network Connection Group Between A, B, and C) - (no Look Ahead: Orig. Branch-Root: with Notify Function)

The actions illustrated in the above figure are as follows.

Requesting Party's terminal equipment issues the following information flow towards its serving node. The terminal equipment then attaches to the backward portion of Network Connection Group.

1) **ADD-BEARER-TO-CALL.ready:** [Requesting Party = A, Addressed Party = B, C]

The requester's serving node validates the request and determines the route and outgoing trunk facility for the Network Connection Group. For this example, the Network Connection Group will be routed through separate relay nodes, two signalling ports are needed, therefore the serving node can not commit to the request. In addition, the two Relay nodes have different status with respect to the call. Relay Node R-1 is already associated with the call while Relay Node R-2 is not associated with the call. Therefore the serving node issues the following information flows towards the selected relay nodes. The Network Connection Group is backward through connected.

2a) **ADD-BEARER-TO-CALL.begin:** [Requesting Party = A, Addressed Party = B, C]

2b) **CALL-&-BEARER SETUP.begin:** [Requesting Party = A, Addressed Party = B, C]

Each selected relay node validates the requests and determines the route and outgoing trunk facility. Each selected relay node issues the following information flows towards the addressed serving nodes. The Network Connection Group in each relay node is backward through connected.

3a) **ADD-BEARER-TO-CALL.begin:** [Requesting Party = A, Addressed Party = B, C]

3b) **CALL-&-BEARER SETUP.begin:** [Requesting Party = A, Addressed Party = B, C]

Each addressed serving node validates the information flow and offers the request to the end equipment already associated with the call. The serving node relays the following information flow towards the selected end equipment. The Network Connection Group is backward through connected.

4a) **ADD-BEARER-TO-CALL.begin:** [Requesting Party = A, Addressed Party = B, C]

4b) **ADD-BEARER-TO-CALL.begin:** [Requesting Party = A, Addressed Party = B, C]

The addressed terminal equipment determines that it can accept the request and issues the following information flow towards it's associated serving node.

5a) **ADD-BEARER-TO-CALL.ready**

5b) **ADD-BEARER-TO-CALL.ready**

The addressed service nodes receives the above flow and it records the response to the action request and issues the following response to the request from its associated two relay node.

6a) **ADD-BEARER-TO-CALL.ready**

6b) **CALL-&-BEARER SETUP.ready**

When the selected relay nodes receive the above response it records it and relays the response to the requesting service node in the form illustrated by the following information flows.

7a) **ADD-BEARER-TO-CALL.ready**

7b) **CALL-&-BEARER SETUP.ready**

When the requesting service node receives these information flows, it records the willingness to accept the call and Network Connection Groups, and send commitment information flows towards the requesting user equipment (flow 8) and the relay nodes (flow 9a and 9b), notifies Party D of the addition of an Network Connection Group to the call via information flow 9c, and performs forward and backward through connect of the Network Connection Groups.

8) **ADD-BEARER-TO-CALL.commit**

When the user equipment receives this information flow, it records the commitment, and notifies the user of this commitment, and connects to the forward portion of the Network Connection Group.

9a) **ADD-BEARER-TO-CALL.commit**

9b) **CALL-&-BEARER SETUP.commit**

When the selected relay nodes receive either 9a or 9b information flow, it records the commitment, and relays this commitment to the addressed service nodes by issuing information flow number 10a and 10b, and performs forward and backward through connect of the Network Connection Group.

10a) **ADD-BEARER-TO-CALL.commit**

10b) **CALL-&-BEARER SETUP.commit**

When the addressed service nodes receive these information flows, they record the commitments, and send commitment information flows to the selected terminals. The addressed serving nodes then through connect each Network Connection Group in the forward and backward directions.

11a) **ADD-BEARER-TO-CALL.commit**

11b) **ADD-BEARER-TO-CALL.commit**

When the user equipment receives one of the above information flows, it records the commitment, and notifies the user of this commitment, and through connects in both directions.

When the serving node associated with Party D receives information flow 9c, it relays the information flow to Party D's end equipment already associated with the call via the following information flow.

9c) **NOTIFY-BEARER-CHANGE.indication: [Network Connection Group Added]**

10c) **NOTIFY-BEARER-CHANGE.indication: [Network Connection Group Added]**

7.2.2.2 Addition of an Network Connection Group to an existing Call - Relay Node Branch Root

The User (Party A) requests that a new point-to-multi-point Network Connection Group be added to the call. Parties A, B, and C are to be attached to the Network Connection Group. The three parties are already associated with the call. Party A is to be the "root" of the Network Connection Group The requested service is of the non-human interactive type. Therefore, immediate answer can be performed by both Party B's and Party C's equipment. If the Requested Parties equipment can accept the requested service, the equipment will indicate acceptance of the Network Connection Group request. This example also assumes that the Requested Parties are connected to a Multi-signalling entity interface. In addition, the network does not perform a "Look Ahead" procedure before progressing with the Network Connection Group establishment.

The signalling capability of establishing this Network Connection Group between the three parties without network "Look Ahead" is illustrated below.

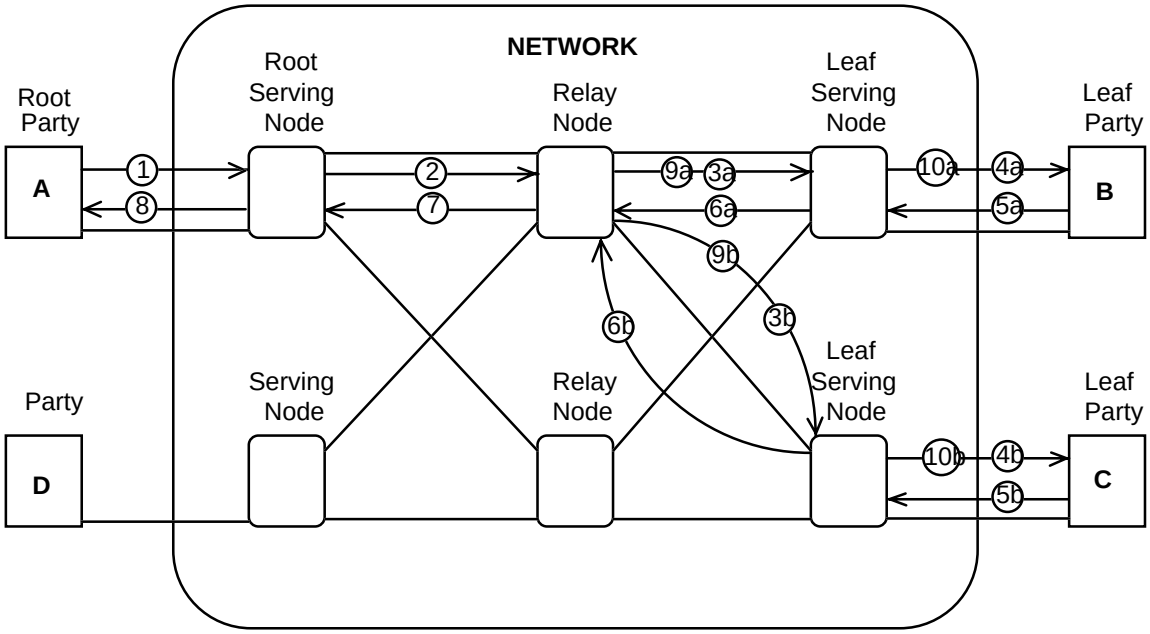


Figure 25: Addition of Type 2 Network Connection Group Between A, B, and C) - (no Look Ahead: Relay Branch-Root)

The actions illustrated in the above figure are as follows.

Requesting Party's terminal equipment issues the following information flow towards its serving node. The terminal equipment then attaches to the backward portion of Network Connection Group.

1) **ADD-BEARER-TO-CALL.ready:** [Requesting Party = A, Addressed Party = B, C]

The requester's serving node validates the request and determines the route and outgoing trunk facility for the Network Connection Group. For this example, the Network Connection Group will be routed through single relay node, the serving node can commit to the request and therefore issues the following information flows towards the selected relay node. The Network Connection Group are backward through connected.

2) **ADD-BEARER-TO-CALL.ready:** [Requesting Party = A, Addressed Party = B, C]

The selected relay node validates the request and determines the route and outgoing trunk facility. As a result of this routeing, two separate routes are required to get to Parties B and C. The selected relay node issues the following information flows towards the addressed serving nodes. The Network Connection Group in the relay node is backward through connected.

3a) **ADD-BEARER-TO-CALL.begin:** [Requesting Party = A, Addressed Party = B, C]

3b) **ADD-BEARER-TO-CALL.begin:** [Requesting Party = A, Addressed Party = B, C]

Each addressed serving node validates the information flow and offers the Network Connection Group request to the end equipment already associated with the call. The serving node relays the following information flow towards the selected terminal. The Network Connection Group is backward through connected.

4a) **ADD-BEARER-TO-CALL.begin:** [Requesting Party = A, Addressed Party = B, C]

4b) **ADD-BEARER-TO-CALL.begin:** [Requesting Party = A, Addressed Party = B, C]

The addressed terminal equipment determines that it can accept the request and issues the following information flow towards it's associated serving node.

5a) **ADD-BEARER-TO-CALL.ready**

5b) **ADD-BEARER-TO-CALL.ready**

The addressed service nodes receives the above flow and it records the responses to the action request and issues the following responses to the request from its associated relay node

6a) **ADD-BEARER-TO-CALL.ready**

6b) **ADD-BEARER-TO-CALL.ready**

When the selected relay node receive the above responses it records them and relays the responses to the associated service nodes in the form illustrated by the following information flows.

7) **ADD-BEARER-TO-CALL.commit**

9a) **ADD-BEARER-TO-CALL.commit**

9b) **ADD-BEARER-TO-CALL.commit**

When the requesting service node receives information flow 7, it records the willingness to accept the Network Connection Group, and sends a commitment information flow towards the requesting user equipment (flow 8) and performs forward and backward through connect of the Network Connection Group.

8) **ADD-BEARER-TO-CALL.commit**

When the user equipment receives this information flow, it records the commitment, and notifies the user of this commitment, and connects to the forward portion of the Network Connection Group.

When the addressed service nodes receives either information flows 9a or 9b, it records the commitment and sends a commitment information flow to the selected terminal. The addressed serving node then through connects in the forward and backward directions.

10a) **ADD-BEARER-TO-CALL.commit**

10b) **ADD-BEARER-TO-CALL.commit**

When the user equipment receives one of the above information flows, it records the commitment, and notifies the user of this commitment, and through connects in both directions.

7.3 Addition of one Network Connection to an Existing Network Connection Group with attachment of existing Parties to the new Network Connection

With and without network Look Ahead.

The following capabilities will be illustrated:

1) Network Connection Establishment of Type 1 Connections;
2) Network Connection Establishment of Type 2 Connections.

7.3.1 Addition of one Type 1 Network Connection

The following capabilities will be illustrated:

1) Addition of one point-to-point Network Connection requested by the Call Initiating Party.

7.3.1.1 Addition of one point-to-point Network Connection requested by the Call Initiating Party

The user (Party A) requests an addition of one single point-to-point Network Connection to an already existing Network Connection group between itself and the Party B. Parties A and B are to be attached to the connection. The user also specifies the High Layer service to be carried on this connection and the desired Bearer service that should be established. The requested service is of the human interactive type. No immediate answer can be performed by the Party B's terminal. If the requested Party B's terminal can accept the requested service it will indicate user alerting. If the human user can accept the requested service also, then the terminal will indicate acceptance of the connection request. The network does not perform a Look Ahead procedure before progressing with the connection establishment.

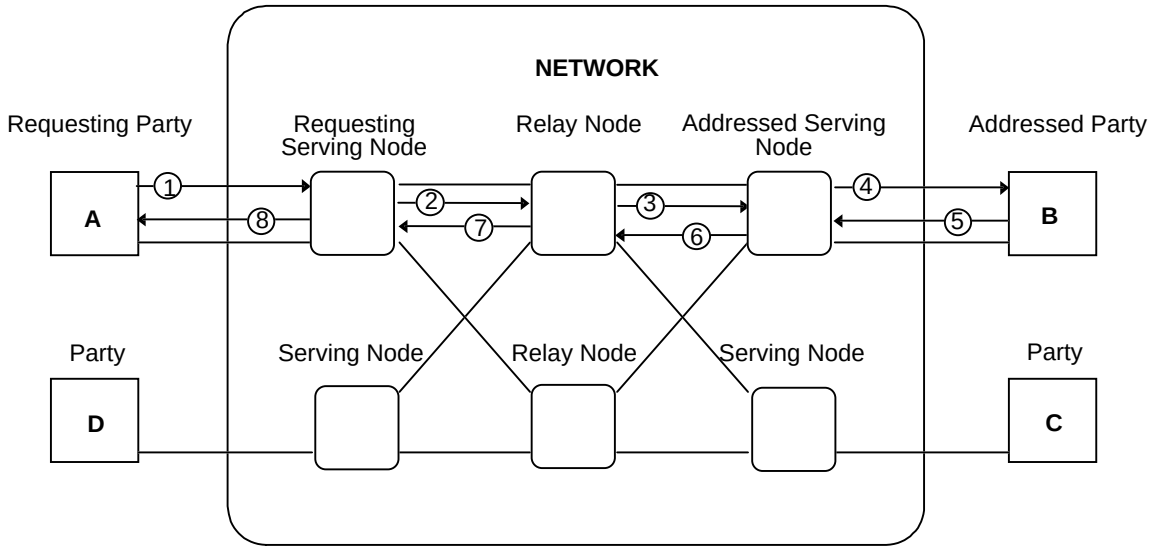


Figure 26: Addition of one point-to-point Network Connection to an existing Network Connection Group

The actions illustrated in the above figure are as follows:

After accepting the request from the user the terminal issues the following information flow towards its serving node.

1) **ADD-BEARER-TO-CALL.ready:** [Requesting Party = A, Addressed Party = B]

The requesting serving node validates the request and determines the route and outgoing trunk facility of the network connection group to which the new Network Connection is to be added. Since only one outgoing connection is needed, the serving node commits to the request and therefore issues the following information flow towards the selected relay node. The connection may be backward through connected.

2) **ADD-BEARER-TO-CALL.ready:** [Requesting Party = A, Addressed Party = B]

The selected relay node validates the request and determines the route and outgoing trunk facility of the network group to which the new Network Connection is to be added. Since only one outgoing connection is needed, the relay node commits to the request and therefore issues the following information flow towards the selected serving node. The connection may backward through connected.

3) **ADD-BEARER-TO-CALL.ready:** [Requesting Party = A, Addressed Party = B]

The addressed serving node validates the request and selects the terminating interface. Since only one outgoing connection is needed, the addressed serving node commits to the request and therefore issues the following information flow towards the selected interface.

4) **ADD-BEARER-TO-CALL.ready:** [Requesting Party = A, Addressed Party = B]

The selected terminal determines that it can commit to the request and issues the following information flow towards the addressed serving node. The terminal then connects to the connection.

5) **ADD-BEARER-TO-CALL.commit**

When the addressed serving node receives this information flow, it records commitment and issues the following information flow towards the requesting relay node. The connection is through connected in both directions.

6) **ADD-BEARER-TO-CALL.commit**

When the relay node receives this information flow, it records commitment and issues the following information flow towards the requesting serving node. The connection is through connected in both directions if not already done.

7) **ADD-BEARER-TO-CALL.commit**

When the requesting serving node receives this information flow, it records commitment and issues the following information flow towards the requesting terminal. The connection is through connected in both directions if not already done.

8) **ADD-BEARER-TO-CALL.commit**

When the requesting terminal receives this information flow, it records commitment and notifies the user of this commitment. The terminal then connects in both directions if not already done.

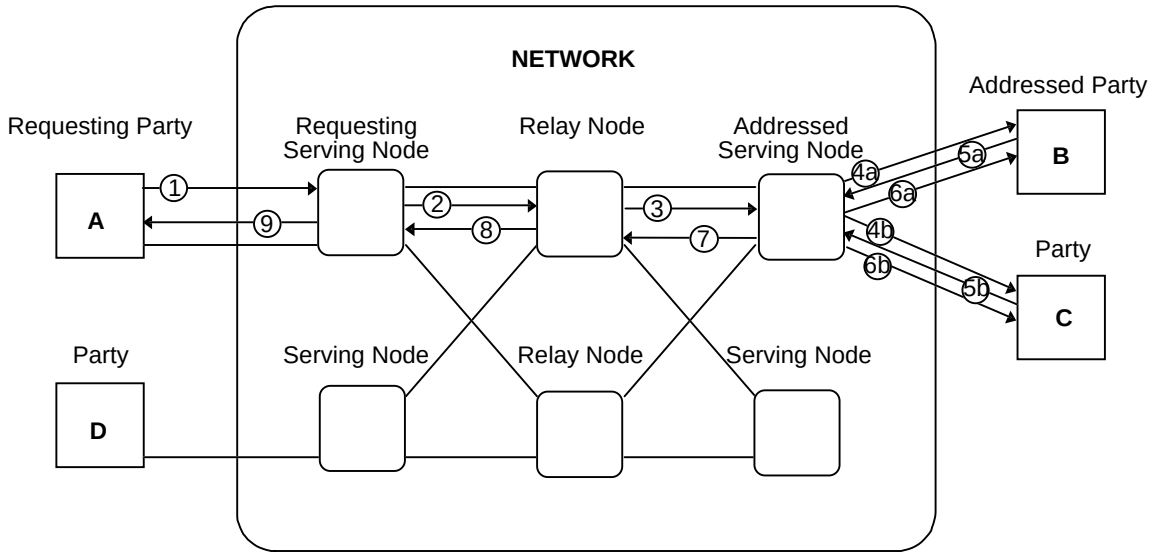
7.3.2 Addition of one Type 2 Network Connection

The following capabilities will be illustrated:

1) Addition of one point to multipoint Network Connection requested by the Root of the new Network Connection.

7.3.2.1 Addition of one point to multipoint Network Connection requested by the Root of the new Network Connection

The user (Party A) requests an addition of one single point to multipoint Network Connection to an already existing Network Connection group between itself and the Parties B and C which are mandatory for this new connection. Parties A, B and C are to be attached to the connection. The user also specifies the High Layer service to be carried on this connection and the desired Bearer service that should be established. The requested service is of the human interactive type. If the requested Parties B's and C's terminal can accept the requested service they will commit. The branching point of the Type 2 connection is at the addressed serving node. The network does not perform a Look Ahead procedure before progressing with the connection establishment.



NOTE: Party B and Party C are connected to different accesses.

Figure 27: Addition of one point to multipoint Network Connection to a Network Connection Group requested by the Root of the new Network Connection

The actions illustrated in the above figure are as follows:

After accepting the request from the user the terminal issues the following information flow towards its serving node.

1) **ADD-BEARER-TO-CALL.ready:** [Requesting Party = A, Addressed Party = B, C]

The requesting serving node validates the request and determines the route and outgoing trunk facility of the network connection group to which the new Network Connection is to be added. Determining that only one outgoing connection is needed, the serving node commits to the request and therefore issues the following information flow towards the selected relay node.

2) **ADD-BEARER-TO-CALL.ready:** [Requesting Party = A, Addressed Party = B, C]

The selected relay node validates the request and determines the route and outgoing trunk facility of the network group to which the new Network Connection is to be added. Determining that only one outgoing connection is needed, the relay node commits to the request and therefore issues the following information flow towards the selected serving node.

3) **ADD-BEARER-TO-CALL.ready:** [Requesting Party = A, Addressed Party = B, C]

The addressed serving node validates the request and selects the terminating interfaces. Determining that multiple outgoing connections are needed, the addressed serving node can not commit to the request and therefore issues the following information flow towards the selected interfaces.

4a) **ADD-BEARER-TO-CALL.begin:** [Requesting Party = A, Addressed Party = B, C]

4b) **ADD-BEARER-TO-CALL.begin:** [Requesting Party = A, Addressed Party = B, C]

Each selected terminal determines that it can commit to the request and issues the following information flow backwards to the addressed serving node.

5a) **ADD-BEARER-TO-CALL.ready**

5b) **ADD-BEARER-TO-CALL.ready**

When the addressed serving node receives both information flows, it records commitment and issues the following information flow towards the addressed terminals. The connection is through connected in forward direction.

6a) **ADD-BEARER-TO-CALL.commit**

6b) **ADD-BEARER-TO-CALL.commit**

When the terminal receives this information flow, it records commitment. The connection is through connected in forward direction.

The addressed serving node issues the following information flow towards the requesting relay node.

7) **ADD-BEARER-TO-CALL.commit**

When the relay node receives this information flow, it records commitment and issues the following information flow towards the requesting serving node. The connection is through connected in forward direction.

8) **ADD-BEARER-TO-CALL.commit**

When the requesting serving node receives this information flow, it records commitment and issues the following information flow towards the requesting terminal. The connection is through connected in forward direction.

9) **ADD-BEARER-TO-CALL.commit**

When the requesting terminal receives this information flow, it records commitment and notifies the user of this commitment. The connection is through connected in forward direction.

7.4 Addition of one or more new Parties to an Existing Call with Attachment to an Existing Network Connection - With or without network Look-ahead.

The following capabilities will be illustrated:

1) Addition of one or more new Parties and attachment to a Type 1 Network Connection;
2) Addition of one or more new Parties and attachment to a Type 2 Network Connection.

7.4.1 Addition of one or more new Parties and attachment to a Type 1 Network Connection

The following capabilities will be illustrated:

1) Add one new Party requested by the call initiating Party.

7.4.1.1 Add one new Party requested by a Party which will be the Root of the Type 1 Network Connection (without network Look Ahead)

The user (Party A) that will become the root of the Type 1 Network Connection requests that a new party C is to be added to the call. There is already a Type 1 unidirectional Network Connection between the party A and the party B. The party C is to be attached to this connection too. The old Type 1 Network Connection will become a Type 2 Network Connection. This example also assumes that the requested party is connected to a point-to-multipoint signalling interface. The network does not perform a Look Ahead procedure before progressing with the connection establishment. It is assumed, that the new branching point will be at the relay node. Notification may be necessary in some cases. This may be service dependent.

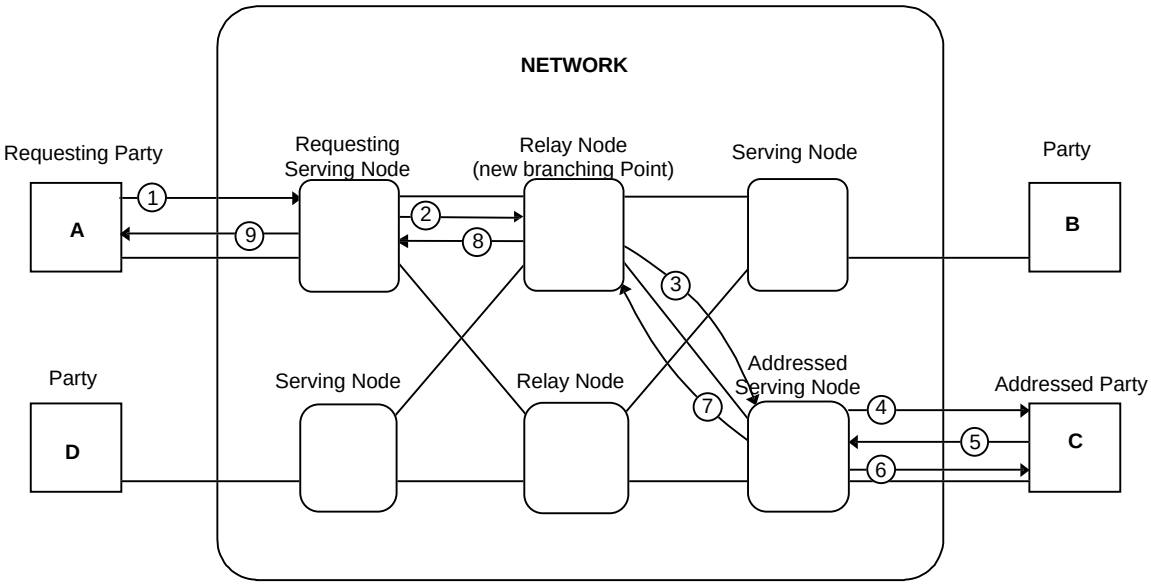


Figure 28: Add one new Party requested by a Party which will be the Root of the Type 1 Network Connection

The actions illustrated in the above figure are as follows:

After accepting the request from the user the terminal issues the following information flow towards its serving node.

1) **ADD-PARTY-TO-BEARER.ready:** [Requesting Party = A, Addressed Party = C]

The requesting serving node validates the request and determines the route and outgoing trunk of the Network Connection. It determines that it will not be the new branching point of the Network Connection. Therefore it commits to the request and issues the following information flow towards the selected relay node.

2) **ADD-PARTY-TO-BEARER.ready:** [Requesting Party = A, Addressed Party = C]

The selected relay node validates the request and determines the route and outgoing trunk facility of the Network Connection. It determines that it will be the new branching point of the existing connection. The type of the Network Connection will be changed from Type 1 to a Type 2 Network Connection. The relay node commits to the request and issues the following information flow towards the addressed serving node of the new party. The new connection branch may be through connected in both directions.

3) **CALL-&-BEARER-SETUP.ready:** [Requesting Party = A, Addressed Party = C]

The addressed serving node validates the request and selects the terminating interface. Since the interface is classified as a point-to-multipoint signalling interface, the addressed serving node can not commit to the request and issues the following information flow towards the selected interface.

4) **CALL-&-BEARER-SETUP.begin:** [Requesting Party = A, Addressed Party = C]

The addressed terminal determines that it can commit to the request and issues the following information flow towards the addressed serving node.

5) **CALL-&-BEARER-SETUP.ready**

The addressed serving node records the responses, selects of the responding terminals and issues the following information flow towards the selected terminal. The non-selected terminals will be cleared. The connection is through connected in both directions.

6) **CALL-&-BEARER-SETUP.commit**

When the selected terminal receives this information flow, it records commitment and through connects in both directions.

Furthermore, the addressed serving node issues the following information flow towards the requesting relay node.

7) **CALL-&-BEARER-SETUP.commit**

When the relay node receives this information flow, it records commitment. Since the relay node has the function of branching it issues the following information flow towards the requesting serving node. The connection is through connected in both directions if not already done.

8) **ADD-PARTY-TO-BEARER.commit**

When the requesting serving node receives this information flow, it records commitment and issues the following information flow towards the requesting terminal.

9) **ADD-PARTY-TO-BEARER.commit**

When the requesting terminal receives this information flow, it records commitment and notifies the user of this commitment.

7.4.2 Addition of one or more new Parties and attachment to a Type 2 Network Connection

The following capabilities are illustrated:

1) Add one or more new Parties requested by the call initiating Party.

7.4.2.1 Add one or more new Parties requested by the call initiating Party

The same flow as for subclause 7.4.1.1 is valid, except that a branching node may already exist in advance. No additional branching node may be needed in this case. Furthermore no change of type of the Network Connection (i.e. from Type 1 to Type 2) is necessary.

7.5 Attachment of one or more existing Parties to an existing Network Connection

The following capabilities will be illustrated:

1) Attachment of one or more existing Parties to a Type 1 Network Connection;
2) Attachment of one or more existing Parties to a Type 2 Network Connection.

7.5.1 Attachment of one or more existing Parties to a Type 1 Network Connection

The following capabilities will be illustrated:

1) Attachment of one existing Party requested by the call initiating Party.

7.5.1.1 Attachment of one existing Party requested by a Party which will be the Root of the Type 1 Network Connection

The user (Party A) that will become the root of the Type 1 Network Connection requests that an existing party C is to be attached to an already existing Type 1 Network Connection between the party A and the party B. The old Type 1 Network Connection will become a Type 2 Network Connection. Notification may be necessary in some cases. This may be service dependent.

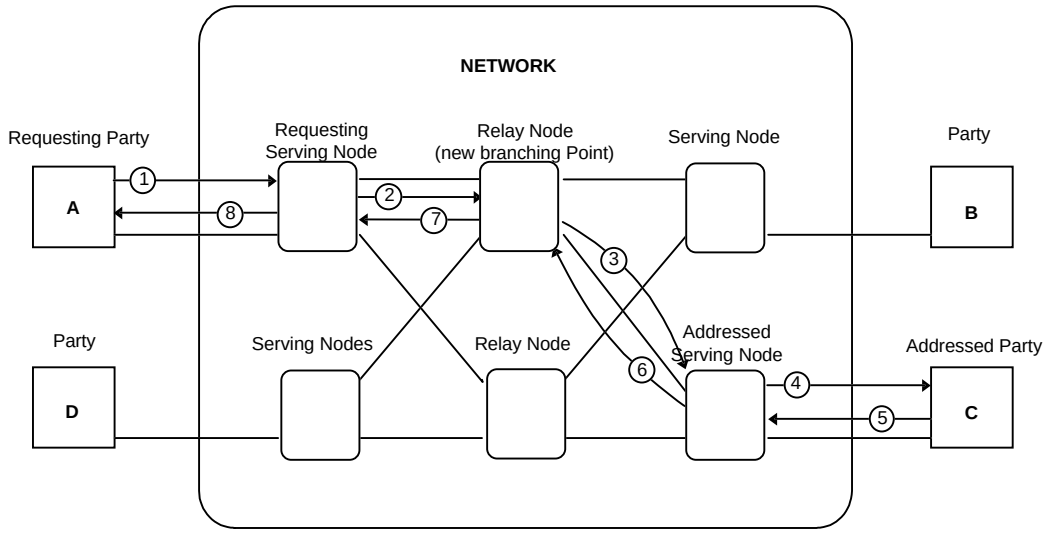


Figure 29: Attachment of one existing Party requested by a Party which will be the Root of the Type 1 Network Connection

The actions illustrated in the above figure are as follows:

After accepting the request from the user the terminal issues the following information flow towards it's serving node.

1) **ATTACH-PARTY-TO-BEARER.ready:** [Requesting Party = A, Addressed Party = C]

The requesting serving node validates the request and determines the route and outgoing trunk facility of the Network Connection. It determines that it will not be the new branching point of the Network Connection. Therefore it commits to the request and issues the following information flow towards the selected relay node.

2) **ATTACH-PARTY-TO-BEARER.ready:** [Requesting Party = A, Addressed Party = C]

The selected relay node validates the request and determines the route and outgoing trunk facility of the Network Connection. It determines that it will be the new branching point of the existing connection. The type of the Network Connection will be changed from Type 1 to a Type 2 Network Connection. The relay node commits to the request and issues the following information flow towards the addressed serving node of the existing party C. The new connection branch may be through connected in both directions.

3) **ADD-BEARER-TO-CALL.ready:** [Requesting Party = A, Addressed Party = C]

The addressed serving node validates the request and selects the terminating interface. Since only one outgoing connection is needed, the addressed serving node commits to the request and therefore issues the following information flow towards the selected interface.

4) **ADD-BEARER-TO-CALL.ready:** [Requesting Party = A, Addressed Party = C]

The selected terminal determines that it can commit to the request and issues the following information flow towards the addressed serving node. The terminal then connects to the connection.

5) **ADD-BEARER-TO-CALL.commit**

When the addressed serving node receives this information flow, it records commitment and issues the following information flow towards the requesting relay node. The connection is through connected in both directions.

6) **ADD-BEARER-TO-CALL.commit**

When the relay node receives this information flow, it records commitment and issues the following information flow towards the requesting serving node. The connection is through connected in both directions if not already done.

7) **ATTACH-PARTY-TO-BEARER.commit**

When the requesting serving node receives this information flow, it records commitment and issues the following information flow towards the requesting terminal.

8) **ATTACH-PARTY-TO-BEARER.commit**

When the requesting terminal receives this information flow, it records commitment and notifies the user of this commitment.

7.5.2 Attachment of one or more existing Parties to a Type 2 Network Connection

The following capabilities will be illustrated:

1) Attachment of one existing Party requested by the call initiating Party.

7.5.2.1 Attachment of one existing Party requested by the call initiating Party

The same flow as for subclause 7.5.1.1 is valid, except that a branching node may be already exists in advance. No additional branching node may be needed in this case. Furthermore no change of type of the Network Connection (i.e. from Type 1 to Type 2) is necessary.

7.6 Call Establishment without any Network Connection Group(s)

The establishment of a call with no Network Connection Groups has two variations: establishing a call between two parties, and establishment of a call with more than two parties. The number of called parties determines whether the single- or two-phase request mechanism is utilized.

7.6.1 Two-Party Call-Only Establishment

Party A requests a two party call without connection(s), and with Party B. If Party B, or Party B's agent functional entity determines that the call-only establishment request can be accepted, the agent functional entity will indicate acceptance of the call-only establishment request. This is shown in the figure below.

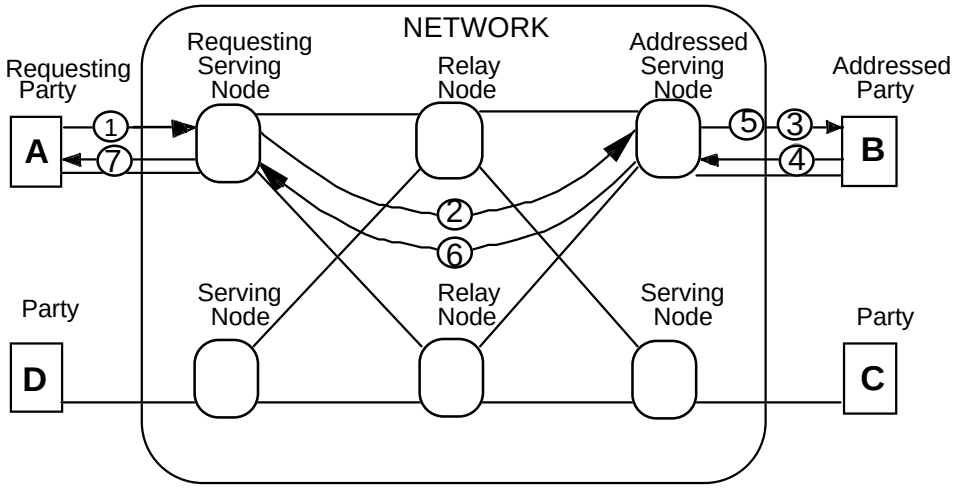


Figure 30: Call-Only Establishment

The actions illustrated in the above figure are:

Party A's terminal equipment issues the following information flow towards it's serving node.

1) **CALL SETUP.ready:** [Requesting Party = A, Addressed Party = B]

The requester's serving node validates the request. Since only one party is involved, the serving node delegates responsibility for committing the request and therefore issues the following information flow towards the addressed serving node.

2) **CALL SETUP.ready:** [Requesting Party = A, Addressed Party = B]

The addressed serving node validates the request. It then issues the following information flow towards the addressed party, Party B.

3) **CALL SETUP.begin:** [Requesting Party = A, Addressed Party = B]

The addressed terminal equipment issues the following information flow towards it's associated serving node.

4) **CALL SETUP.ready**

The addressed serving node records the responses to the action request and selects one of the responding terminals. The selected terminal is sent information flow number 5. The serving node then clears the non-selected terminals (note this action is not illustrated for simplicity), and issues information flow 6 toward the requesting serving node. When the Terminal receives information flow 5, it records the commitment.

5) **CALL SETUP.commit**

6) **CALL SETUP.commit**

When the requesting service node receives information flow 6, it records the commitment, and relays this commitment to the requesting user equipment by issuing information flow 7.

7) **CALL SETUP.commit**

When the user equipment receives information flow 7, it records the commitment, and notifies the user of this commitment.

7.6.2 Three-or-More-Party Call-Only Establishment

Party A requests a three (or more) party call without connection(s), and with Party B and Party C. If Party B, or Party B's agent functional entity determines that the call-only establishment request can be accepted, and Party C, or Party C's agent functional entity determines that the call-only establishment request can be accepted, then the requesting serving node functional entity will indicate acceptance of the call-only establishment request. This is shown in the figure below. The procedures for a call with more than 3 parties would be extensions of this example.

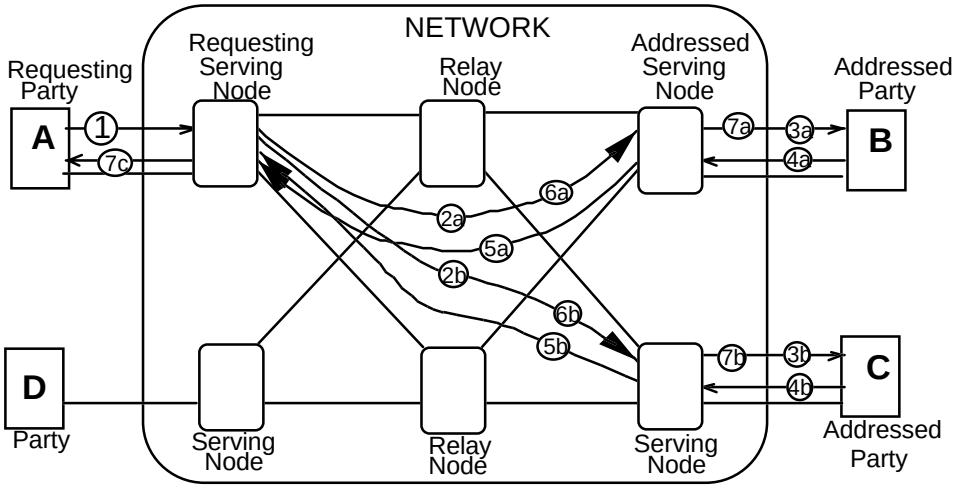


Figure 31: Call-only establishment of a 3-party call

The actions illustrated in the above figure are as follows:

Party A's terminal equipment issues the following information flow towards it's serving node.

1) **CALL SETUP.ready:** [Requesting Party = A, Addressed Party = B, C]

The requester's serving node validates the request. Since two parties are involved, the serving node retains responsibility for committing the request and therefore issues the following information flows towards the addressed serving nodes.

2a) **CALL SETUP.begin:** [Requesting Party = A, Addressed Party = B, C]

2b) **CALL SETUP.begin:** [Requesting Party = A, Addressed Party = B, C]

Each addressed serving node validates the request. Each then issues the following information flow towards the addressed party.

3a) **CALL SETUP.begin:** [Requesting Party = A, Addressed Party = B, C]

3b) **CALL SETUP.begin:** [Requesting Party = A, Addressed Party = B, C]

Each set of addressed terminal equipment issues the information flow 4a (or 4b) towards it's associated serving node.

4a) **CALL SETUP.ready**

4b) **CALL SETUP.ready**

Each addressed serving node records the responses to the action request and selects one of the responding terminals. The selected terminal is recorded for use in issuing flows 7a and 7b.. Each serving node then clears the non-selected terminals (note this action is not illustrated for simplicity), and issues information flow 5a (or 5b) toward the requesting serving node.

5a) **CALL SETUP.ready**

5b) **CALL SETUP.ready**

When the requesting serving node receives all of the "ready" information flows, it sends flows 6a and 6b to the addressed serving nodes, and flows 7c to the requesting party.

6a) **CALL SETUP.commit**

6b) **CALL SETUP.commit**

7c) **CALL SETUP.commit**

When the addressed serving nodes receive flows 6a and 6b, then each sends the commit flows (7a or 7b) to the addressed party.

7a) **CALL SETUP.commit**

7b) **CALL SETUP.commit**

When each address party receives information flow 7, it records the commitment and informs the user.

7.7 Addition of a Party to a Call without Attachment to any Network Connection Group(s)

This will not be described by the Baseline Text.

7.8 Detachment of a Party from an Existing Connection

The detachment of a party from an existing connection is illustrated in this subclause of the signalling requirements. The illustrations are organized in three groups based on the manner in which the requesting party is attached to the connection. The three groups are as follows:

1) Detachment Requested by the Root of the Connection;
2) Detachment Requested by a Leaf of the Connection;
3) Detachment Requested by a Party not associated with the Connection.

The following subclauses contain the chosen examples used to illustrate the detachment signalling requirements.

7.8.1 Detachment Requested by the Root of the Connection

A party that is associated with the root of a connection may request that it be is detached, or that a leaf be detached from the connection. The network will action the request based on the Call ownership, the Connection Ownership and Branch Ownership status of the requesting party. The actions taken by the network are shown in the following table.

Table 43

Command	Location of Requester	Requester Ownership Type	Location of Party to be Detached	ACTION
Detach-Party	Root	---	Only itself	Connection released by the Root serving node
	Root	Call, Connection, or Branch Owner of the specified Leaf	Leaf of the connection	Branch to the Specified Party released by the Root serving node
	Root	Not Branch Owner of the specified Leaf	Leaf of the connection	Denied

The following subclauses illustrate the two allowed cases shown in the above table.

7.8.1.1 Root Party Requests Detachment from the Connection

The party associated with the root of a connection may at any time request detachment. When this occurs the connection is to be released. The Root Serving Node issues the commands to release the connection and notifies all parties which are not attached to the connection but are associated with the call that the connection has been dropped. The following diagram illustrates this example.

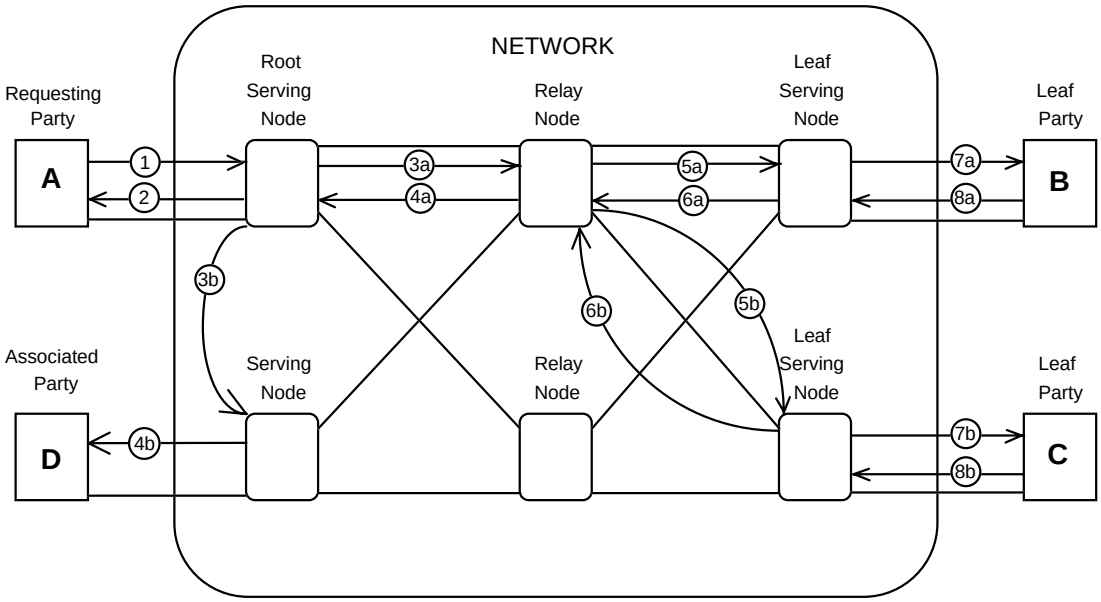


Figure 32: Detach Party "A" from Connection requested by Party "A" Point-to-multi-point Connection with Root = Party "A" Single Relay Exchange Example

The information flows shown above are as follows:

1) Detach-Party-from-Bearer.ready:{Requester = Party A, Addressed Party = Party A}
2) Detach-Party-from-Bearer.commit
3a) Release-Bearer.ready:
3b) Notify-Bearer-Change.indication:(Connection Released) to Party D
4a) Release-Bearer.commit
4b) Notify-Bearer-Change.indication:(Connection Released) to Party D
5a) Release-Bearer.ready:
5b) Release-Bearer.ready:
6a) Release-Bearer.commit
6b) Release-Bearer.commit
7a) Release-Bearer.ready:
7b) Release-Bearer.ready:
8a) Release-Bearer.commit
8b) Release-Bearer.commit

7.8.1.2 Root Party Requests that a Leaf be Detached from the Connection

If the party associated with the root of a connection requests that a leaf party is to be removed, the requesting party needs to be the owner of the branch associated with the requested party or be the call or connection owner before the network will detach the specified party. If the requesting party is not allowed to detach the specified party, the network will return a Cancel response flow. The following example illustrates the flows within the network for the case where the requesting party is allowed to detach the party that was specified in the requesting flow.

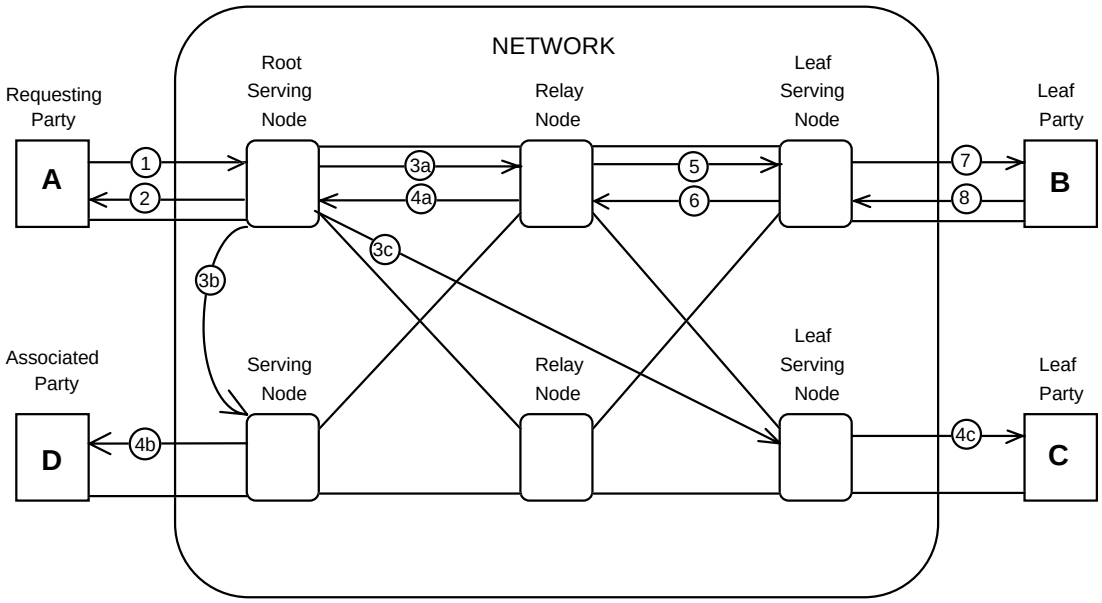


Figure 33: Detach Party "B" from Connection requested by Party "A" Point-to-multi-point Connection with Root = Party "A"

The information flows shown above are as follows:

1) Detach-Party-from-Bearer.ready:{Requester = Party A, Addressed Party = Party B};
2) Detach-Party-from-Bearer.commit;
3a) Detach-Party-from-Bearer.ready:{Requester = Party A, Addressed Party = Party B};
3b) Notify-Bearer-Change.indication:(Party B detached from connection) to Party D;
3c) Notify-Bearer-Change.indication:(Party B detached from connection) to Party C;
4a) Detach-Party-from-Bearer.commit;
4b) Notify-Bearer-Change.indication:(Party B detached from connection) to Party D;
4c) Notify-Bearer-Change.indication:(Party B detached from connection) to Party C;
5) Detach-Party-from-Bearer.ready;
6) Detach-Party.from.Bearer.commit;
7) Detach-Party.from.Bearer.ready;
8) Detach-Party.from.Bearer.commit.

7.8.2 Detachment Requested by a Leaf of the Connection

A party that is associated with a leaf of a connection may request that it be is detached, the root be detached, or a leaf be detached from the connection. The network will action the request based on the Call ownership, Connection Ownership and Branch Ownership status of the requesting party. The actions taken by the network are shown in the following table.

Table 44

Command	Location of Requester	Requester Ownership Type	Location of Party to be Detached	ACTION
Detach-Party	Leaf	---	Only itself	Release Branch to Requester, Message sent towards Root serving node
	Leaf	Call Owner, or Connection Owner	Root	Connection released by the Root serving node
	Leaf	Call, Connection or Branch Owner of the specified leaf	A Leaf of the connection	Branch to the Specified Party released by the Root serving node.
	Leaf	Not Branch Owner of the specified leaf	A Leaf of the connection	Denied
	Leaf	Not Call or not Connection Owner	Root or a Leaf of the connection	Denied

7.8.2.1 A Leaf Party Requests that it be Detached from a Connection

A Leaf Party may detach itself from a connection at any time. The following diagram illustrates this action.

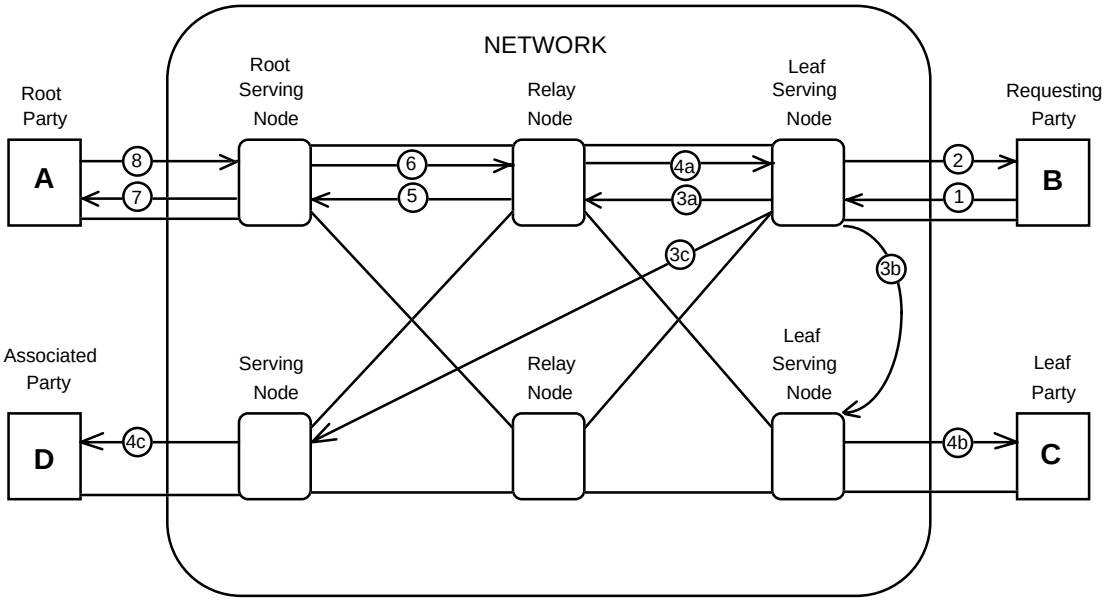


Figure 34: Detach Party "B" requested by Party "B" Point-to-multi-point Connection with Root = Party "A"

The information flows shown above are as follows:

1) Detach-Party-from-Bearer.ready:{Requester = Party B, Addressed Party = Party B};
2) Detach-Party-from-Bearer.commit;
3a) Detach-Party-from-Bearer.ready:{Requester = Party B, Addressed Party = Party B};
3b) Notify-Bearer-Change.indication:(Party B detached from connection) to Party C;
3c) Notify-Bearer-Change.indication:(Party B detached from connection) to Party D;
4a) Detach-Party-from-Bearer.commit;
4b) Notify-Bearer-Change.indication:(Party B detached from connection) to Party C;
4c) Notify-Bearer-Change.indication:(Party B detached from connection) to Party D;
5) Detach-Party-from-Bearer.ready:{Requester = Party B, Addressed Party = Party B};
6) Detach-Party-from-Bearer.commit;
7) Detach-Party-from-Bearer.ready:{Requester = Party B, Addressed Party = Party B};
8) Detach-Party-from-Bearer.commit.

7.8.2.2 A Leaf Party Requests that the Root be Detached from the Connection

A Leaf Party may request that a root be detached from a connection. In the example shown below Party B requests that the root party be detached. When the Leaf Serving node receives this request, it determines that Party B is allowed to request this disconnection and forwards the request to the Root Serving Node. The Root Serving node performs the disconnection action.

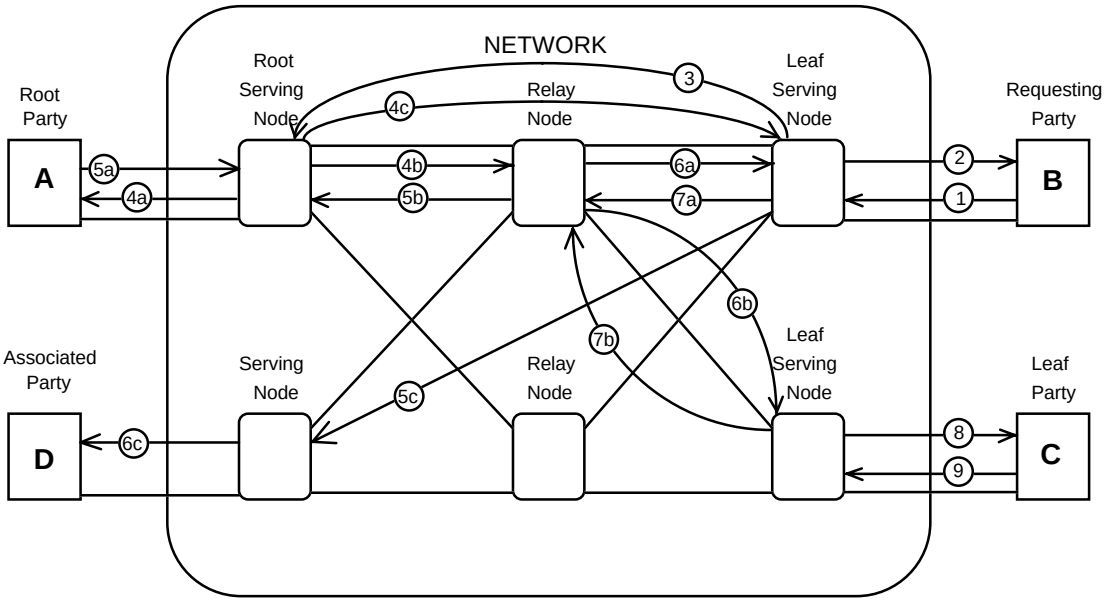


Figure 35: Detach Party "A" from Connection requested by Party "B" Point-to-multi-point Connection with Root = Party "A" Single Relay Example

The information flows shown above are as follows:

1) Detach-Party-from-Bearer.ready:{Requester = Party B, Addressed Party = Party A;
2) Detach-Party-from-Bearer.commit;
3) Detach-Party-from-Bearer-Remote.ready:{Requester = Party B, Addressed Party = Party A};
4a) Release-Bearer.ready;
4b) Release-Bearer.ready;
4c) Detach-Party-from-Bearer-Remote.commit;
5a) Release-Bearer.commit;
5b) Release-Bearer.commit;
5c) Notify-Bearer-Change.indication:(Connection Released) to Party D;
6a) Release-Bearer.ready;
6b) Release-Bearer.ready;
6c) Notify-Bearer-Change.indication:(Connection Released) to Party D;
7a) Release-Bearer.commit;
7b) Release-Bearer.commit;
8) Release-Bearer.ready;
9) Release-Bearer.commit.

7.8.2.3 A Leaf Party Requests a Leaf be Detached from the Connection

A Leaf party may request that a leaf be detached from a connection.

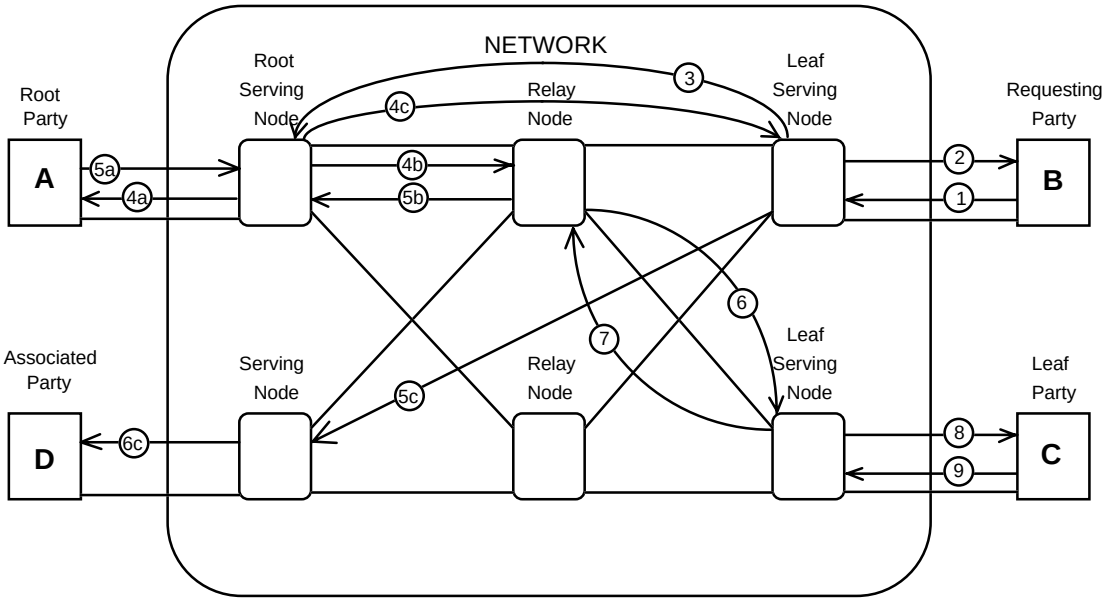


Figure 36: Detach Party "C" from Connection requested by Party "B" Point-to-multi-point Connection with Root = Party "A"

The information flows shown above are as follows:

1) Detach-Party-from-Bearer.ready:{Requester = Party B, Addressed Party = Party C};
2) Detach-Party-from-Bearer.commit;
3) Detach-Party-from-Bearer-Remote.ready:{Requester = Party B, Addressed Party = Party C};
4a) Detach-Party-from-Bearer.ready:{Requester = Party B, Addressed Party = Party C};
4b) Detach-Party-from-Bearer.ready:{Requester = Party B, Addressed Party = Party C};
4c) Detach-Party-from-Bearer-Remote.commit;
5a) Detach-Party-from-Bearer.commit;
5b) Detach-Party-from-Bearer.commit;
5c) Notify-Bearer-Change.indication:(Party C detached from the Connection) to Party D;
6) Detach-Party.from.Bearer.ready;
6c) Notify-Bearer-Change.indication:(Party C detached from the Connection) to Party D;
7) Detach-Party.from.Bearer.commit;
8) Detach-Party.from.Bearer.ready:{Requester = Party B, Addressed Party = Party C};
9) Detach-Party.from.Bearer.commit.

7.8.3 Detachment Requested by a Party not Attached to the Connection

A party that is not attached to a connection may request that the root of the connection be detached, or a leaf be detached from the connection. The network will action the request based on the Call Ownership, Connection Ownership and Branch Ownership status of the requesting party. The actions taken by the network are shown in the following table.

Table 45

Command	Location of Requester	Requester Ownership Type	Location of Party to be Detached	ACTION
Detach-Party	3rd	Call Owner, or Connection Owner	Root of the connection	Connection released by the Root serving node
	3rd	Call Owner, Connection Owner or Branch Owner of the specified leaf	A Leaf of the connection	Branches to Specified Parties released by the Root serving node
	3rd	Not Branch Owner of the specified leaf	A Leaf of the connection	Denied
	3rd	Not Call Owner or not Connection Owner	Root or Leaf of connection	Denied

7.8.3.1 A Party Not Attached to a Connection Requests that the Root Party be Detached

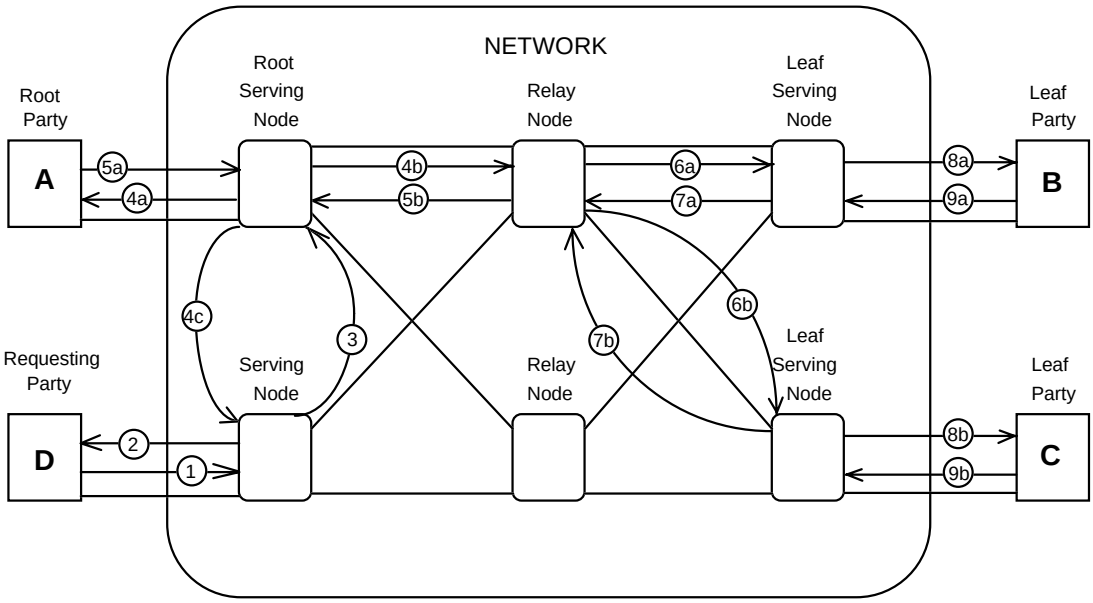


Figure 37: Detach Party "A" from Connection requested by Party "D" Point-to-multi-point Connection with Root = Party "A" Single Relay Exchange Example

The information flows shown above are as follows:

1) Detach-Party-from-Bearer.ready:{Requester = Party D, Addressed Party = Party A};
2) Detach-Party-from-Bearer.commit;
3) Detach-Party-from-Bearer-Remote.ready:{Requester = Party D, Addressed Party = Party A};
4a) Release-Bearer.ready;
4b) Release-Bearer.ready;
4c) Detach-Party-from-Bearer-Remote.commit.

When the Serving Node associated with Party D receives information flow 4c it will notify all serving nodes associated with other parties associated with the call and not the connection that the connection has been released.

5a) Release-Bearer.commit
5b) Release-Bearer.commit
6a) Release-Bearer.ready
6b) Release-Bearer.ready
7a) Release-Bearer.commit
7b) Release-Bearer.commit
8a) Release-Bearer.ready
8b) Release-Bearer.ready
9a) Release-Bearer.commit
9b) Release-Bearer.commit

7.8.3.2 A Party Not Attached to a Connection Requests that a Leaf be Detached

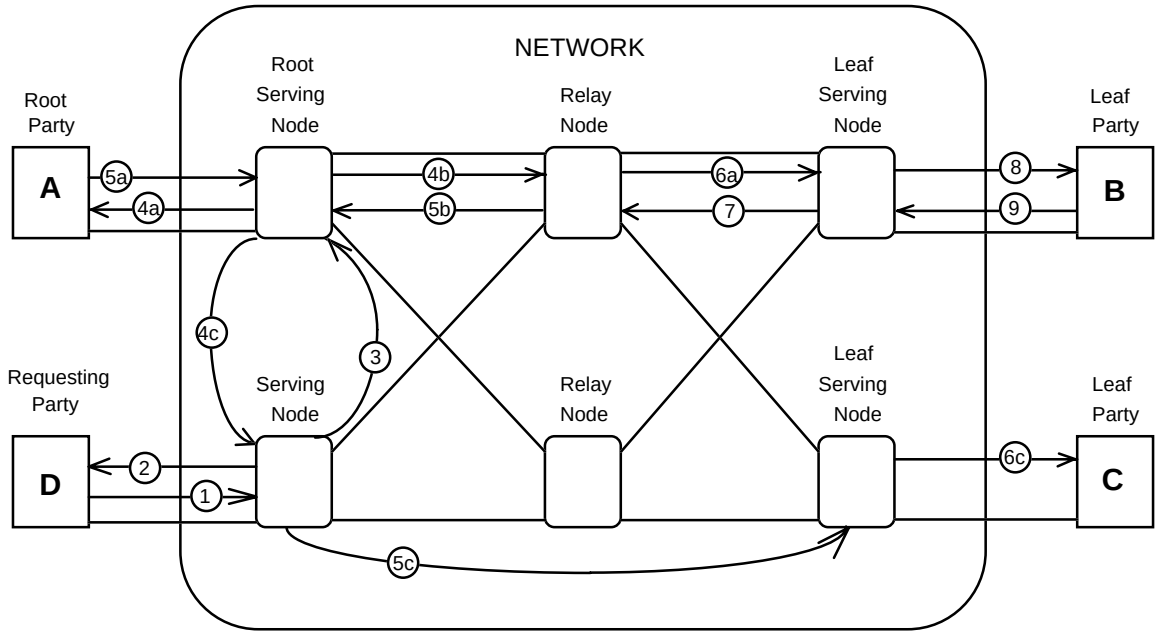


Figure 38: Detach Party "B" from Connection requested by Party "D" Point-to-multi-point Connection with Root = Party "A"

The information flows shown above are as follows:

1) Detach-Party-from-Bearer.ready:{Requester = Party D, Addressed Party = Party B};
2) Detach-Party-from-Bearer.commit;
3) Detach-Party-from-Bearer-Remote.ready:{Requester = Party D, Addressed Party = Party B};
4a) Detach-Party-from-Bearer.ready:{Requester = Party D, Addressed Party = Party B};
4b) Detach-Party-from-Bearer.ready:{Requester = Party D, Addressed Party = Party B};
4c) Detach-Party-from-Bearer-Remote.commit;
5a) Detach-Party-from-Bearer.commit;
5b) Detach-Party-from-Bearer.commit;
5c) Notify-Bearer-Change.indication:(Party B detached from Connection) to Party C;
6a) Detach-Party.from.Bearer.ready;
6b) Detach-Party.from.Bearer.ready;
6c) Notify-Bearer-Change.indication:(Party B detached from Connection) to Party C;
7) Detach-Party.from.Bearer.commit;
8) Detach-Party.from.Bearer.ready;
9) Detach-Party.from.Bearer.commit.

7.9 Release of the Connection from an Existing Call

Examples of the release of a connection from an existing call are contained in the following two subclauses:

1) Release of a Point-to-point Connection;
2) Release of a Point-to-multi-point Connection.

In all of the examples, it is assumed that the connection that is being removed does not cause the call to be cleared.

7.9.1 Release of a Point-to-point Connection from an existing Call

The following diagram illustrates the action taken by the network when either party of a point-to-point connection requests the release of the connection. The example also assumes that the other parties associated with the call are notified that the connection has been removed.

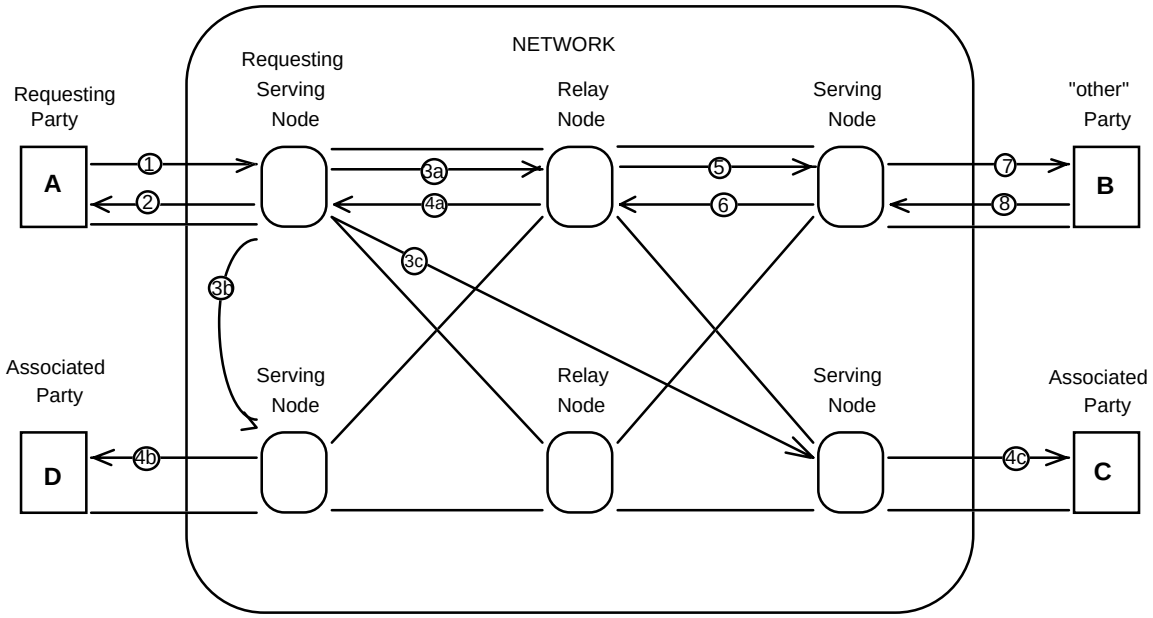


Figure 39: Release of a Point-to-point Connection Between Party "A" and Party "B"

1) Release-Bearer.Ready
2) Release-Bearer.commit
3a) Release-Bearer.Ready
3b) Notify-Bearer-Change.Indication:(Bearer released) to Party D
3c) Notify-Bearer-Change.Indication:(Bearer released) to Party C
4a) Release-Bearer.commit
4b) Notify-Bearer-Change.Indication:(Bearer released) to Party D
4c) Notify-Bearer-Change.Indication:(Bearer released) to Party C
5) Release-Bearer.Ready
6) Release-Bearer.commit
7) Release-Bearer.Ready
8) Release-Bearer.commit

7.9.2 Release of a Point-to-multi-point Connection from an Existing Call

The release of a point-to-multi-point connection associated with an existing call can be requested by the party associated with the root of the connection, a party that is a leaf of the connection, or by a party that is not attached to the connection (3rd Party Request). This request may be honoured by the network depending on the Call Ownership, Connection Ownership status of the requesting party. The following table illustrates the actions taken by the node receiving the release request.

Table 46

Command	Location of Requester	Requester Ownership Type	ACTION
Release-Bearer	Root	---	Connection released by the Root serving node
	Leaf	Call Owner, or Connection Owner	Connection released by the Root serving node
	Leaf	Not Call Owner or not Connection Owner	Release Branch, Message sent towards root, Note: connection may be released if two party connection
	3rd Party	Call Owner, or Connection Owner	Connection released by the Root serving node
	3rd Party	Not Call Owner or not Connection Owner	Denied

7.9.2.1 Release of a Connection requested by the Party Associated with the Root

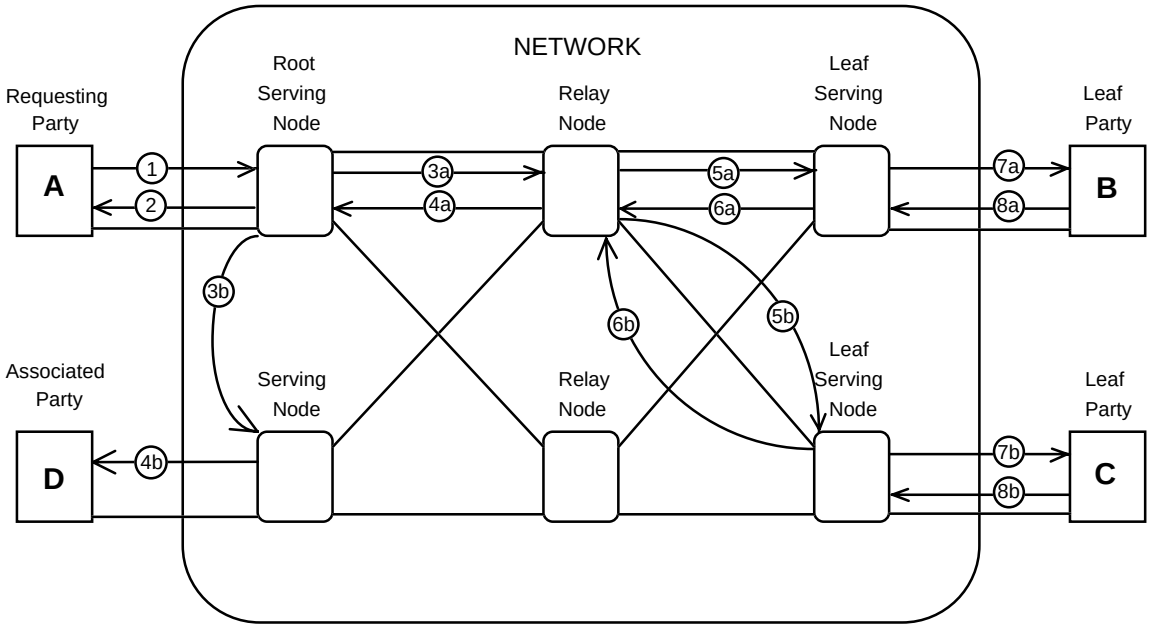


Figure 40: Release Bearer requested by Party "A" Point-to-multi-point Connection with Root = Party "A" Single Relay Exchange Example

1) Release-Bearer.Ready
2) Release-Bearer.commit
3a) Release-Bearer.Ready
3b) Notify-Bearer-Change.indication:(Bearer Release) to Party D
4a) Release-Bearer.commit
4b) Notify-Bearer-Change.indication:(Bearer Release) to Party D
5a) Release-Bearer.Ready
5b) Release-Bearer.Ready
6a) Release-Bearer.commit
6b) Release-Bearer.commit
7a) Release-Bearer.Ready
7b) Release-Bearer.Ready
8a) Release-Bearer.commit
8b) Release-Bearer.commit

7.9.2.2 Release of a Connection Requested by a Leaf Party that is the Connection Owner

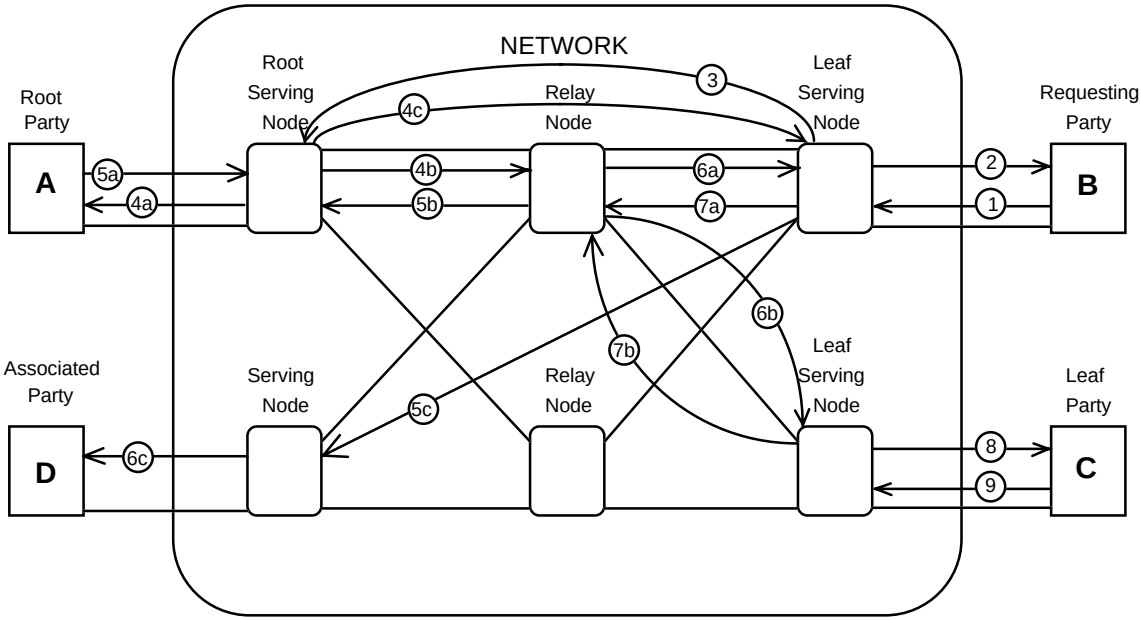


Figure 41: Release Bearer requested by Party "B" Point-to-multi-point Connection with Root = Party "A" Single Relay Exchange Example

1) Release-Bearer.Ready
2) Release-Bearer.commit
3) Release-Bearer-Remote.Ready
4a) Release-Bearer.Ready
4b) Release-Bearer.Ready
4c) Release-Bearer-Remote.commit
5a) Release-Bearer.commit
5b) Release-Bearer.commit
5c) Notify-Bearer-Change.indication:(Bearer Release) to Party D
6a) Release-Bearer.Ready
6b) Release-Bearer.Ready
6c) Notify-Bearer-Change.indication:(Bearer Release) to Party D
7a) Release-Bearer.commit
7b) Release-Bearer.commit
8) Release-Bearer.Ready
9) Release-Bearer.commit

7.9.2.3 Release of a Connection Requested by a Leaf Party that is not the Connection Owner

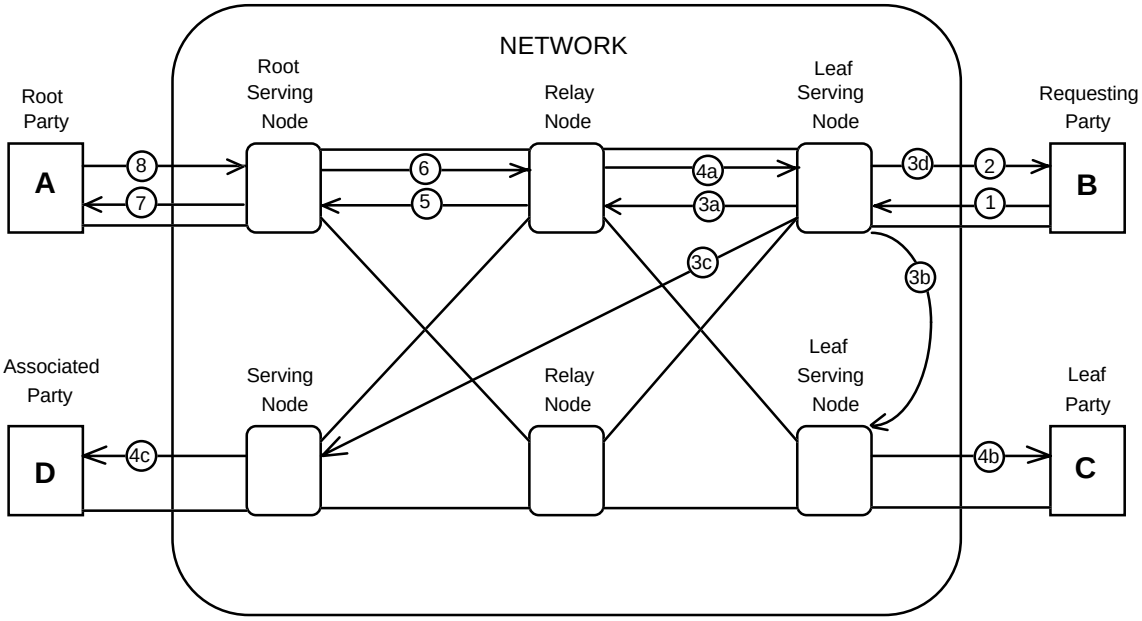


Figure 42: Release Bearer requested by Party "B" Party "B" is not the Connection Owner Point-to-multi-point Connection with Root = Party "A" Single Relay Exchange Example

1) Release-Bearer.Ready
2) Release-Bearer.Ready
3a) Detach-Party-from-Bearer.Ready:{Requester = Party B, Addressed Party = Party B}
3b) Notify-Bearer-Change.indication:(Party B detached from Connection) to Party C
3c) Notify-Bearer-Change.indication:(Party B detached from Connection) to Party D
3d) Notify-Bearer-Change.indication:(connection Added between Parties....) to Party B
4a) Detach-Party-from-Bearer.commit
4b) Notify-Bearer-Change.indication:(Party B detached from Connection) to Party C
4c) Notify-Bearer-Change.indication:(Party B detached from Connection) to Party D
5) Detach-Party-from Bearer.Ready:{Requester = Party B; Addressed Party = Party B}
6) Detach-Party-from-Bearer.commit
7) Detach-Party-from-Bearer.Ready:{Requester = Party B; Addressed Party = Party B}
8) Detach-Party-from-Bearer.commit

7.9.2.4 Release of a Connection Requested by a Party not attached to the Connection

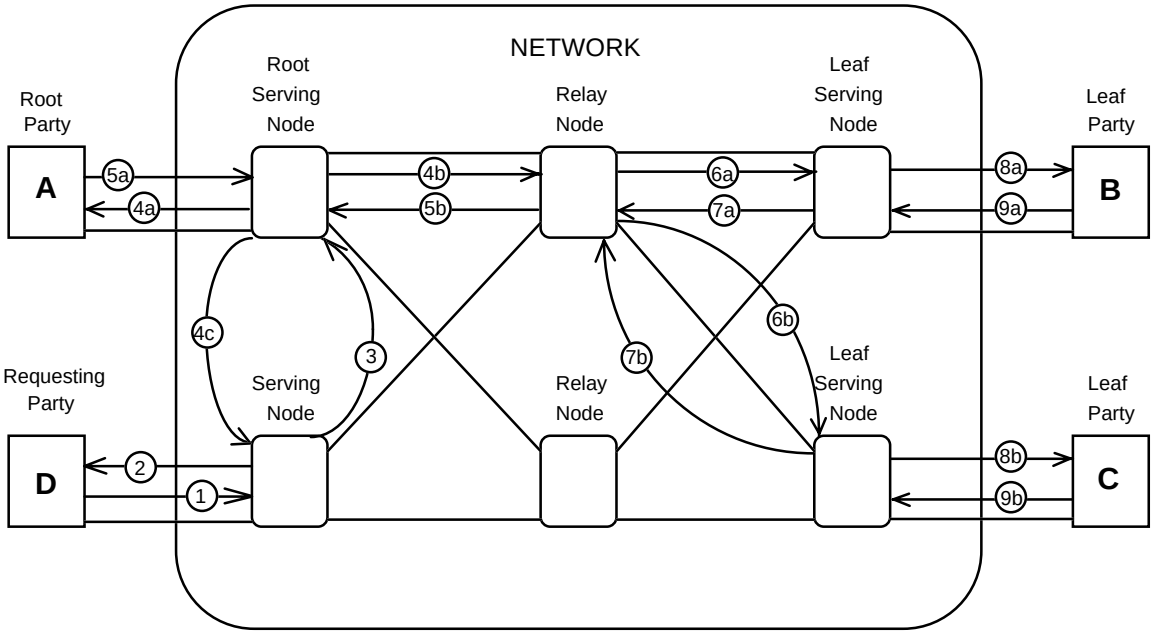


Figure 43: Release Bearer requested by Party "D" Point-to-multi-point Connection with Root = Party "A" Single Relay Exchange Example

1) Release-Bearer.Ready
2) Release-Bearer.commit
3) Release-Bearer-Remote.Ready
4a) Release-Bearer.Ready
4b) Release-Bearer.Ready
4c) Release-Bearer-Remote.commit

When the Serving Node associated with Party D receives information flow 4c it will notify any other parties associated with the call but not associated with the released connection.

5a) Release-Bearer.commit
5b) Release-Bearer.commit
6a) Release-Bearer.Ready
6b) Release-Bearer.Ready
7a) Release-Bearer.commit
7b) Release-Bearer.commit
8a) Release-Bearer.Ready
8b) Release-Bearer.Ready
9a) Release-Bearer.commit
9b) Release-Bearer.commit

7.10 Release of a Party from an existing call

7.10.1 General rules for release of a Party

- If the call owner releases himself, the general rules for releasing a call apply (see next subclause);
- If a non-call owner releases himself, the following actions will take place.

1) Release all the connections he owns
2) Release all the branches he owns from the remaining connections
3) Release all the parties he owns
4) Detach itself from all the remaining connections it is attached to
5) Notify all relevant parties

The procedures for 1, 2 and 4 are from the service point of view equivalent to the corresponding procedures.

- If a party releases another party, the serving node of the requesting party will send a Release-Party-from-Call.ready to the serving node of the party to be released. The receiving serving node will acknowledge this flow and invoke 1, 2, 3, 4 and 5 as described above.

- If a not call owner (release 1 or 2) is attached to a release 1 serving node and releases itself, the Release-Party flow need not to be send because it owns no other party (not the call owner).

- If a not call owner (release 1 or 2) is attached to a release 1 serving node, the Release-Party flow will not be received. The connection however will be released and with it the party.

7.10.2 Release of a Party from a Call requested by the Call or Party Owner

7.10.2.1 Release of a Party that is the leaf party of a Type 2 Connection by the Call Owner

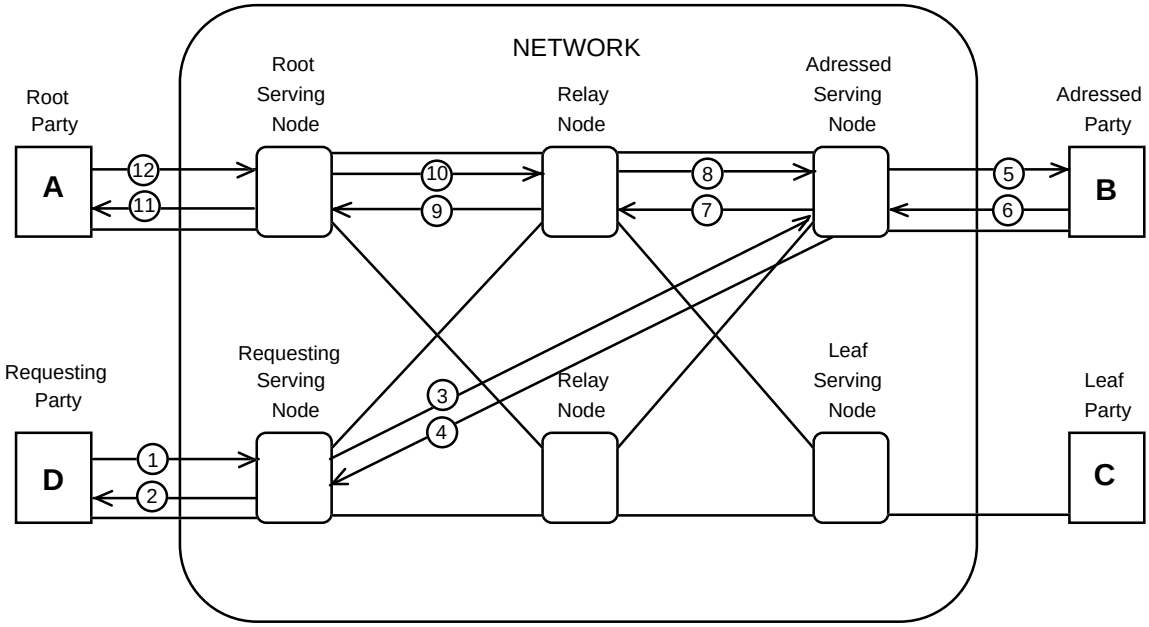


Figure 44: Release Party "B" from Call requested by Party "D" One type 2 connection between Parties A, B and C with Party A as root and connection owner Party D that is call owner is not attached to the Type 2 connection

1) Release-Party-from-Call.Ready
2) Release-Party-from-Call.Commit
3) Release-Party-from-Call.Ready
4) Release-Party-from-Call.Commit
5) Release-Party-from-Call.Ready
6) Release-Party-from-Call.Commit
7) Detach-Party-from-Bearer.Ready
8) Detach-Party-from-Bearer.Commit
9) Detach-Party-from-Bearer.Ready
10) Detach-Party-from-Bearer.Commit
11) Detach-Party-from-Bearer.Ready
12) Detach-Party-from-Bearer.Commit

7.10.3 Release of a Party from a Call requested by the Party to be released

7.10.3.1 Release of a Party that is the Connection owner but not the root of a Type 2 Connection

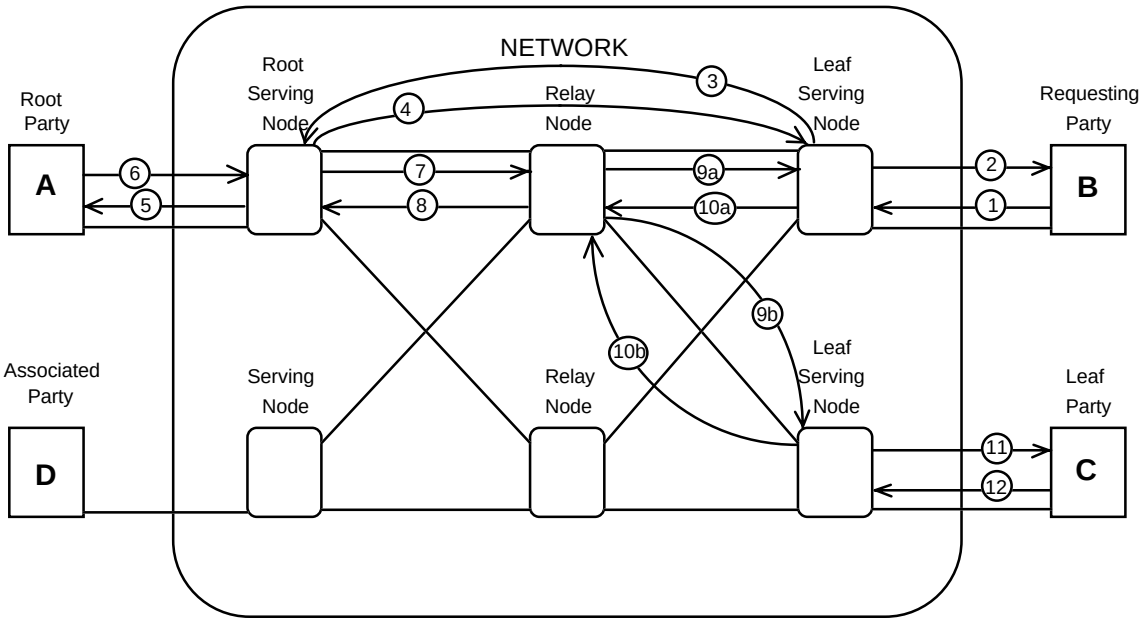


Figure 45: Release Party from Call requested by Party "B" Point-to-multi-point Connection with Root = Party "A" The Connection Owner is Party B that is to be released Party A is the Call Owner

1) Release-Party-from-Call.Ready
2) Release-Party-from-Call.Commit
3) Release-Bearer-Remote.Ready
4) Release-Bearer-Remote.Commit
5) Release-Bearer.Ready
6) Release-Bearer.Commit
7) Release-Bearer.Ready
8) Release-Bearer.Commit
9a) Release-Bearer.Ready
9b) Release-Bearer.Ready
10a) Release-Bearer.Commit
10b) Release-Bearer.Commit
11) Release-Bearer.Ready
12) Release-Bearer.Commit

7.11 Release of a Call

7.11.1 General rules for releasing a call

- Only the call owner is allowed to invoke this operation.

1) The call owner will inform all the parties he owns that the call will be released. He will use an Edge to edge flow (Release-Call.ready).
2) The serving node of the call owner will detach all connections to which it is attached.

- If the serving node of an owned party receives a Release-Call.ready, the following actions will take place:

1) Acknowledge the Release-Call.ready by a Release-Call.commit.
2) Inform all the parties it owns that the call will be released. It will use an Edge to edge flow (Release-Call.ready).
3) The serving node will detach all connections to which it is attached.

- If a call owner (release 1 or 2) is attached to a release 1 serving node and releases itself, the Release-Call flow can not be send. The connection however will be released, so the other end will be informed and can invoke action 1 and 2 if it is a release 2 serving node.

- If a non-call owner (release 1 or 2) is attached to a release 1 serving node, the Release-Call flow will not be received. The connection however will be released and with it the party.

7.11.2 Release of a Call without any connections

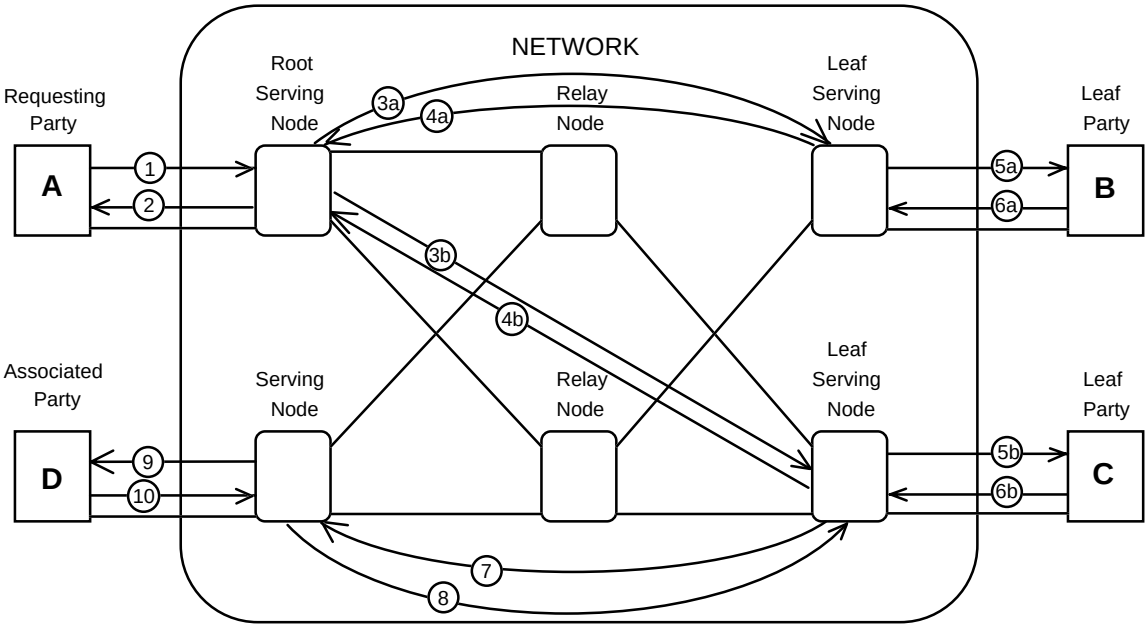


Figure 46: Release of a Call with four Parties but no connections Party "A" is the call owner and owns Parties B and C Party C is the Party owner of Party D

1) Release-Call.Ready
2) Release-Call.Commit
3a) Release-Call.Ready
3b) Release-Call.Ready
4a) Release-Call.Commit
4b) Release-Call.Commit
5a) Release-Call.Ready
5b) Release-Call.Ready
6a) Release-Call.Commit
6b) Release-Call.Commit
7) Release-Call.Ready
8) Release-Call.Commit
9) Release-Call.Ready
10) Release-Call.Commit

7.11.3 Release of a Call with multiple Type 1 connections

7.11.3.1 Release of a Call with two Type 1 connections between different Parties

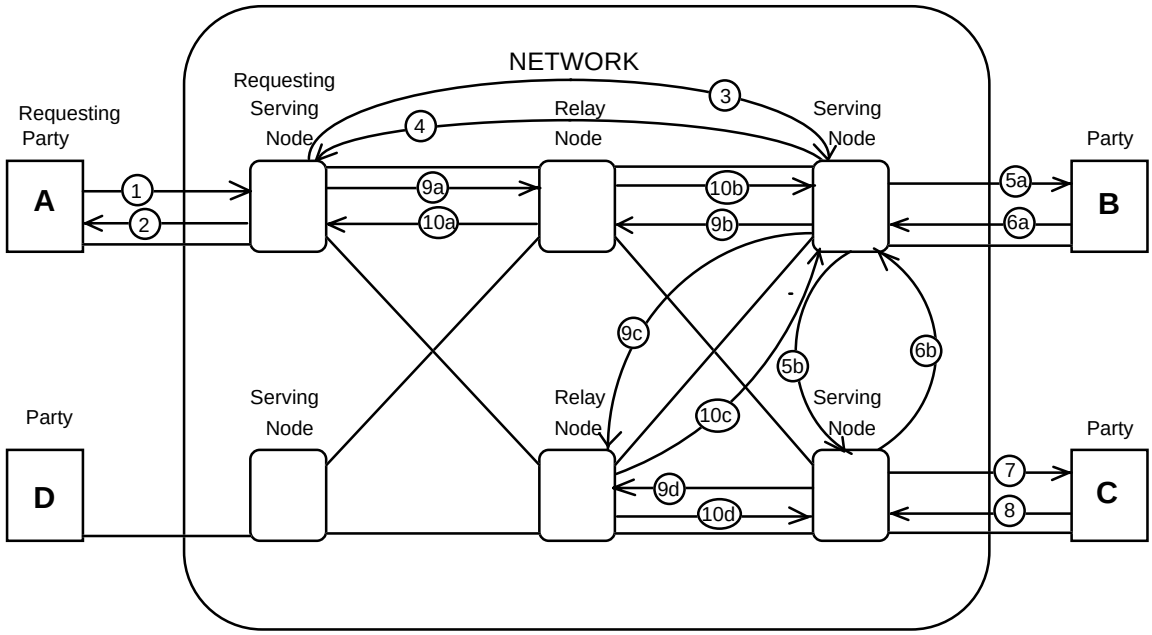


Figure 47: Release Call requested by Party "A" Call owner A own Party B Party C is owned by party B Two Type 1 Connections: One between A and B and one between B and C

1) Release-Call.Ready
2) Release-Call.Commit
3) Release-Call.Ready
4) Release-Call.Commit
5a) Release-Call.Ready
5b) Release-Call.Ready
6a) Release-Call.Commit
6b) Release-Call.Commit
7) Release-Call.Ready
8) Release-Call.Commit
9a) Detach-Party-from-Bearer.Ready
9b) Detach-Party-from-Bearer.Ready
9c) Detach-Party-from-Bearer.Ready
9d) Detach-Party-from-Bearer.Ready
10a) Detach-Party-from-Bearer.Commit
10b) Detach-Party-from-Bearer.Commit
10c) Detach-Party-from-Bearer.Commit
10d) Detach-Party-from-Bearer.Commit

7.11.3.2 Release of a Call Requested by the Connection owner

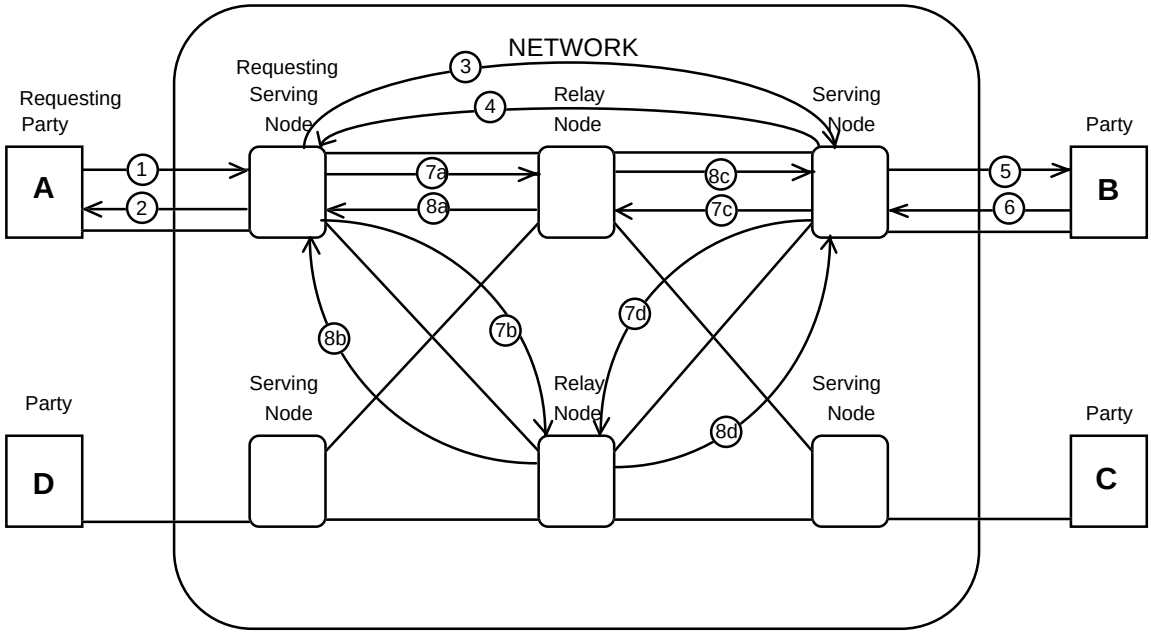


Figure 48: Release Call requested by the call owner Party "A" Two Type 1 Connections between A and B routed over different relay nodes

1) Release-Call.Ready
2) Release-Call.Commit
3) Release-Call.Ready
4) Release-Call.Commit
5) Release-Call.Ready
6) Release-Call.Commit
7a) Detach-Party-from-Bearer.Ready
7b) Detach-Party-from-Bearer.Ready
7c) Detach-Party-from-Bearer.Ready
7d) Detach-Party-from-Bearer.Ready
8a) Detach-Party-from-Bearer.Commit
8b) Detach-Party-from-Bearer.Commit
8c) Detach-Party-from-Bearer.Commit
8d) Detach-Party-from-Bearer.Commit

7.11.4 Release of a Call with Point to multipoint connections

7.11.4.1 Release of a Call where the call owner is not the root of the connection

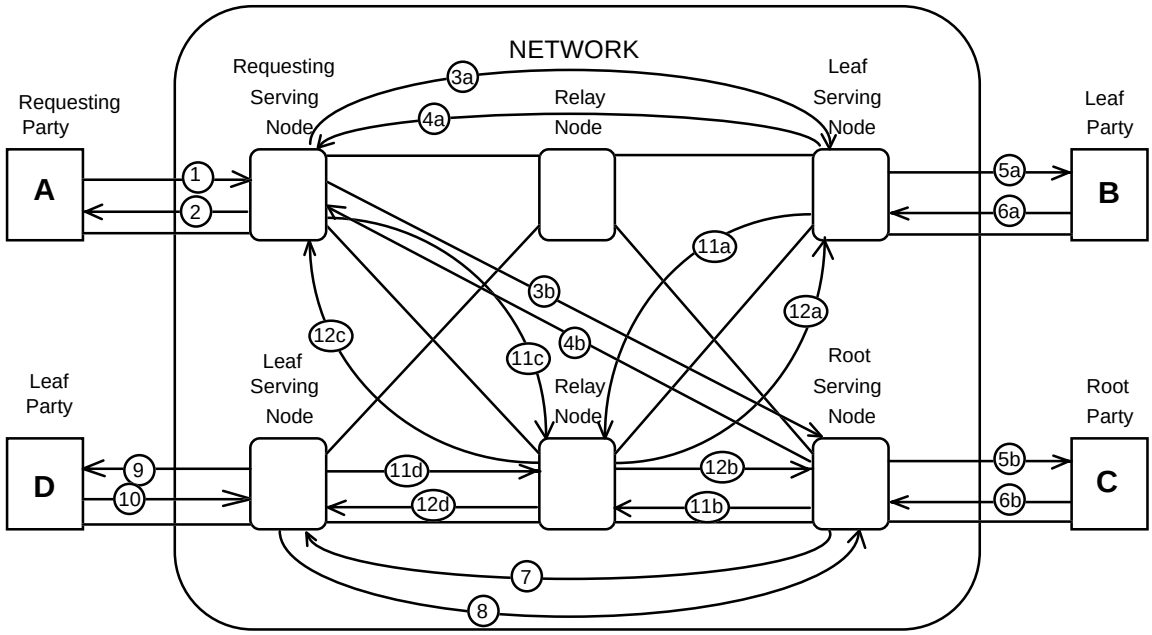


Figure 49: Release Call requested by the call owner Party "A" One Type 2 Connection between Parties A, B, C and D Party C is the root of the connection Branching is taking place in the relay node The call owner A owns Parties B and C Party C is Party owner of Party D

1) Release-Call.Ready
2) Release-Call.Commit
3a) Release-Call.Ready
3b) Release-Call.Ready
4a) Release-Call.Commit
4b) Release-Call.Commit
5a) Release-Call.Ready
5b) Release-Call.Ready
6a) Release-Call.Commit
6b) Release-Call.Commit
7) Release-Call.Ready
8) Release-Call.Commit
9) Release-Call.Ready
10) Release-Call.Commit
11a) Detach-Party-from-Bearer.Ready
11b) Release-Bearer.Ready
11c) Detach-Party-from-Bearer.Ready
11d) Detach-Party-from-Bearer.Ready
12a) Detach-Party-from-Bearer.Commit
12b) Release-Bearer.Commit
12c) Detach-Party-from-Bearer.Commit
12d) Detach-Party-from-Bearer.Commit

7.12 Look Ahead Procedures

This subclause defines the Look Ahead procedures to be defined for CS-2.

Two variations exist to provide this capability. They are called:

1) Look Ahead without state change at the Addressed Party's terminal;
2) Look Ahead with state change at the Addressed Party's terminal.

The Human User Version defers the X.Ready until after an indication that end-to-end has been successfully allocated.

The Non-Human User version allows a terminal to indicate it's X.Ready before the network has allocated resources.

7.12.1 Look Ahead without State Change

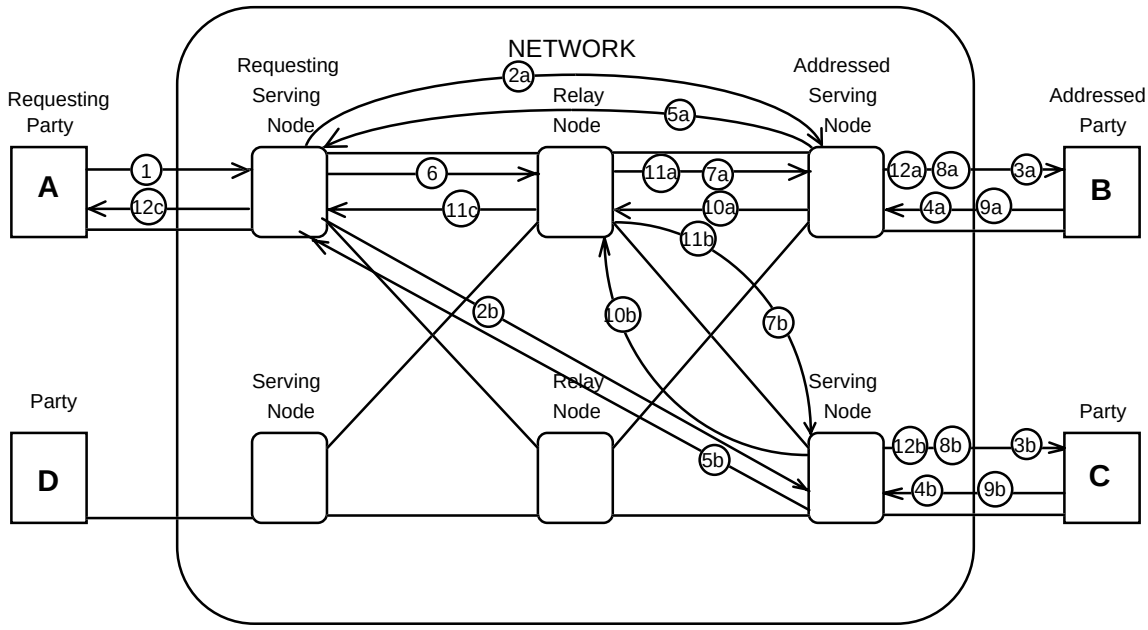


Figure 50: Look Ahead without State Change

The actions illustrated in the above figure are as follows:

Table 47

Flow No.	Call-&-Bearer Setup	Add-Bearer-to-Call	Add-Party-to-Bearer	Attach-Party-to-Bearer
1)	Call-&-Bearer-Setup.Ready	Add-Bearer-to-Call.Ready	Add-Party-to-Bearer.Ready	Attach-Party-to-Bearer.Ready
2a)	Interogation-Terminating-End-Point.Ready	Interogation-Terminating-End-Point.Ready	Interogation-Terminating-End-Point.Ready	Interogation-Terminating-End-Point.Ready
2b)	Interogation-Terminating-End-Point.Ready	Interogation-Terminating-End-Point.Ready	Interogation-Terminating-End-Point.Ready	Interogation-Terminating-End-Point.Ready
3a)	Interogation-Terminating-End-Point.Ready	Interogation-Terminating-End-Point.Ready	Interogation-Terminating-End-Point.Ready	Interogation-Terminating-End-Point.Ready
3b)	Interogation-Terminating-End-Point.Ready	Interogation-Terminating-End-Point.Ready	Interogation-Terminating-End-Point.Ready	Interogation-Terminating-End-Point.Ready
4a)	Interogation-Terminating-End-Point.Commit	Interogation-Terminating-End-Point.Commit	Interogation-Terminating-End-Point.Commit	Interogation-Terminating-End-Point.Commit
4b)	Interogation-Terminating-End-Point.Commit	Interogation-Terminating-End-Point.Commit	Interogation-Terminating-End-Point.Commit	Interogation-Terminating-End-Point.Commit

(continued)

Table 47 (concluded)

Flow No.	Call-&-Bearer Setup	Add-Bearer-to-Call	Add-Party-to-Bearer	Attach-Party-to-Bearer
5a)	Interogation-Terminating-End-Point.Commit	Interogation-Terminating-End-Point.Commit	Interogation-Terminating-End-Point.Commit	Interogation-Terminating-End-Point.Commit
5b)	Interogation-Terminating-End-Point.Commit	Interogation-Terminating-End-Point.Commit	Interogation-Terminating-End-Point.Commit	Interogation-Terminating-End-Point.Commit
6)	Call-&-Bearer-Setup.Ready	Add-Bearer-to-Call.Ready note 1	Add-Party-to-Bearer.Ready note 3	Attach-Party-to-Bearer.Ready note 5
7a)	Call-&-Bearer-Setup.Begin	Add-Bearer-to-Call.Begin note 2	Add-Party-to-Bearer.Begin note 4	Attach-Party-to-Bearer.Begin note 6
7b)	Call-&-Bearer-Setup.Begin	Add-Bearer-to-Call.Begin note 2	Add-Party-to-Bearer.Begin note 4	Attach-Party-to-Bearer.Begin note 6
8a)	Call-&-Bearer-Setup.Begin	Add-Bearer-to-Call.Begin	Add-Bearer-to-Call.Begin	Add-Bearer-to-Call.Begin
8b)	Call-&-Bearer-Setup.Begin	Add-Bearer-to-Call.Begin	Add-Bearer-to-Call.Begin	Add-Bearer-to-Call.Begin
9a)	Call-&-Bearer-Setup.Ready	Add-Bearer-to-Call.Ready	Add-Bearer-to-Call.Ready	Add-Bearer-to-Call.Ready
9b)	Call-&-Bearer-Setup.Ready	Add-Bearer-to-Call.Ready	Add-Bearer-to-Call.Ready	Add-Bearer-to-Call.Ready
10a)	Call-&-Bearer-Setup.Ready	Add-Bearer-to-Call.Ready note 2	Add-Party-to-Bearer.Ready note 4	Attach-Party-to-Bearer.Ready note 6
10b)	Call-&-Bearer-Setup.Ready	Add-Bearer-to-Call.Ready note 2	Add-Party-to-Bearer.Ready note 4	Attach-Party-to-Bearer.Ready note 6
11a)	Call-&-Bearer-Setup.Commit	Add-Bearer-to-Call.Commit note 2	Add-Party-to-Bearer.Commit note 4	Attach-Party-to-Bearer.Commit note 6
11b)	Call-&-Bearer-Setup.Commit	Add-Bearer-to-Call.Commit note 2	Add-Party-to-Bearer.Commit note 4	Attach-Party-to-Bearer.Commit note 6
11c)	Call-&-Bearer-Setup.Commit	Add-Bearer-to-Call.Commit note 1	Add-Party-to-Bearer.Commit note 3	Attach-Party-to-Bearer.Commit note 5
12a)	Call-&-Bearer-Setup.Commit	Add-Bearer-to-Call.Commit	Add-Bearer-to-Call.Commit	Add-Bearer-to-Call.Commit
12b)	Call-&-Bearer-Setup.Commit	Add-Bearer-to-Call.Commit	Add-Bearer-to-Call.Commit	Add-Bearer-to-Call>Commit
12c)	Call-&-Bearer-Setup.Commit	Add-Bearer-to-Call.Commit	Add-Party-to-Bearer.Commit	Attach-Party-to-Bearer.Commit

NOTE 1: If relay node was not already associated with the call use, Call-&-Bearer-Setup.
NOTE 2: If leaf serving nodes are not already associated with call or no call association existed between relay node and the leaf serving node, use Call-&-Bearer-Setup.
NOTE 3: If relay node was not already associated with call, use Call-&-Bearer-Setup.
NOTE 4: If relay node was not already associated with call, or no call association existed between the relay node and the leaf serving node, use Add-Party-to-Bearer.
NOTE 5: If relay node was not already associated with the call use, Call-&-Bearer-Setup.
NOTE 6: If relay node was not already associated with call, or no call association existed between the relay node and the leaf serving node, use Add-Party-to-Bearer.

7.12.2 Look Ahead with State Change

This subclause defines the Look Ahead procedures with state change at the addressed party.

Two options exist to provide this capability:

1) Human User Version;
2) Non-Human User Version.

7.12.2.1 Human User Version

This subclause is for the human user version.

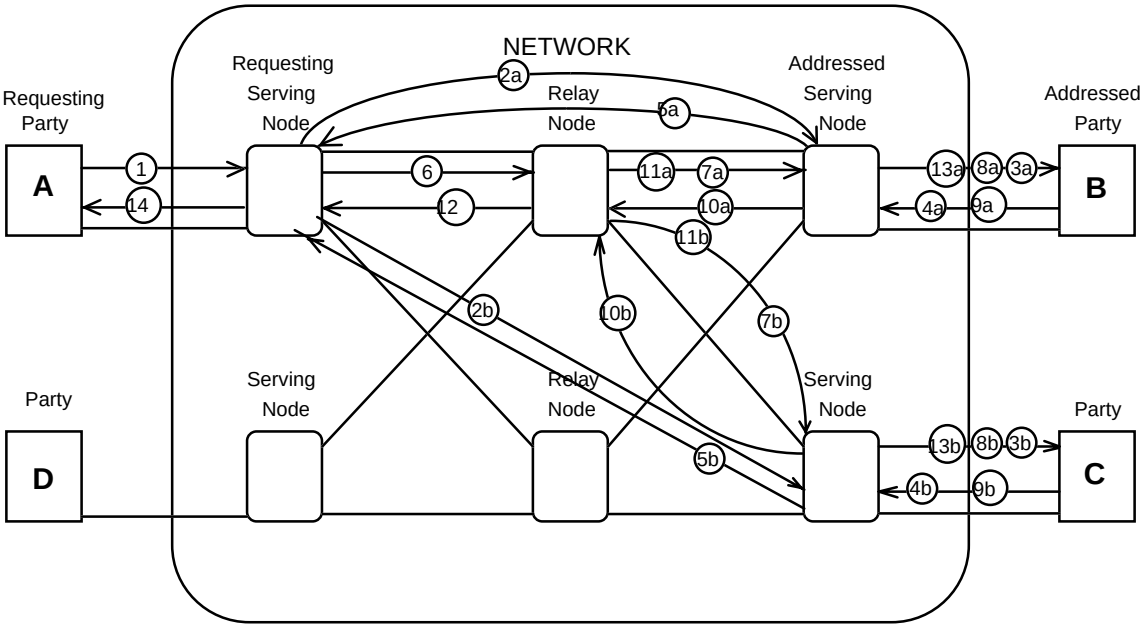


Figure 51: Look Ahead with State Change, human user version

Table 48

Flow No.	Call-&-Bearer Setup	Add-Bearer-to-Call	Add-Party-to-Bearer	Attach-Party-to-Bearer
1)	Call-&-Bearer-Setup.Ready	Add-Bearer-to-Call.Ready	Add-Party-to-Bearer.Ready	Attach-Party-to-Bearer.Ready
2a)	Lookahead.Begin	Lookahead.Begin	Lookahead.Begin	Lookahead.Begin
2b)	Lookahead.Begin	Lookahead.Begin	Lookahead.Begin	Lookahead.Begin
3a)	Call-&-Bearer-Setup.Begin	Add-Bearer-to-Call.Begin	Call-&-Bearer-Setup.Begin	Add-Bearer-to-Call.Begin
3b)	Call-&-Bearer-Setup.Begin	Add-Bearer-to-Call.Begin	Call-&-Bearer-Setup.Begin	Add-Bearer-to-Call.Begin
4a)	ReportLink (Selection)	ReportLink (Selection)	ReportLink (Selection)	ReportLink (Selection)
4b)	ReportLink (Selection)	ReportLink (Selection)	ReportLink (Selection)	ReportLink (Selection)
5a)	Lookahead.Ready	Lookahead.Ready	Lookahead.Ready	Lookahead.Ready
5b)	Lookahead.Ready	Lookahead.Ready	Lookahead.Ready	Lookahead.Ready
6)	Call-&-Bearer-Setup.Ready	Add-Bearer-to-Call.Ready note 1	Add-Party-to-Bearer.Ready note 3	Attach-Party-to-Bearer.Ready note 5

(continued)

Table 48 (concluded)

Flow No.	Call-&-Bearer Setup	Add-Bearer-to-Call	Add-Party-to-Bearer	Attach-Party-to-Bearer
7a)	Call-&-Bearer-Setup.Begin	Add-Bearer-to-Call.Begin note 2	Add-Party-to-Bearer.Begin note 4	Attach-Party-to-Bearer.Begin note 6
7b)	Call-&-Bearer-Setup.Begin	Add-Bearer-to-Call.Begin note 2	Add-Party-to-Bearer.Begin note 4	Attach-Party-to-Bearer.Begin note 6
8a)	ReportLink (Selection Ack)	ReportLink (Selection Ack)	ReportLink (Selection Ack)	ReportLink (Selection Ack)
8b)	ReportLink (Selection Ack)	ReportLink (Selection Ack)	ReportLink (Selection Ack)	ReportLink (Selection Ack)
9a)	Call-&-Bearer-Setup.Ready	Add-Bearer-to-Call.Ready	Call-&-Bearer-Setup.Ready	Add-Bearer-to-Call.Ready
9b)	Call-&-Bearer-Setup.Ready	Add-Bearer-to-Call.Ready	Call-&-Bearer-Setup.Ready	Add-Bearer-to-Call.Ready
10a)	Call-&-Bearer-Setup.Ready	Add-Bearer-to-Call.Ready note 2	Add-Party-to-Bearer.Ready note 4	Attach-Party-to-Bearer.Ready note 6
10b)	Call-&-Bearer-Setup.Ready	Add-Bearer-to-Call.Ready note 2	Add-Party-to-Bearer.Ready note 4	Attach-Party-to-Bearer.Ready note 6
11a)	Call-&-Bearer-Setup.Commit	Add-Bearer-to-Call.Commit note 2	Add-Party-to-Bearer.Commit note 4	Attach-Party-to-Bearer.Commit note 6
11b)	Call-&-Bearer-Setup.Commit	Add-Bearer-to-Call.Commit note 2	Add-Party-to-Bearer.Commit note 4	Attach-Party-to-Bearer.Commit note 6
12)	Call-&-Bearer-Setup.Commit	Add-Bearer-to-Call.Commit note 1	Add-Party-to-Bearer.Commit note 3	Attach-Party-to-Bearer.Commit note 7
13a)	Call-&-Bearer-Setup.Commit	Add-Bearer-to-Call.Commit	Call-&-Bearer-Setup.Commit	Add-Bearer-to-Call.Commit
13b)	Call-&-Bearer-Setup.Commit	Add-Bearer-to-Call.Commit	Call-&-Bearer-Setup.Commit	Add-Bearer-to-Call.Commit
14)	Call-&-Bearer-Setup.Commit	Add-Bearer-to-Call.Commit	Add-Party-to-Bearer.Commit	Attach-Party-to-Bearer.Commit

NOTE 1: If relay node was not already associated with the call use, Call-&-Bearer-Setup.
NOTE 2: If leaf serving nodes are not already associated with call or no call association existed between relay node and the leaf serving node, use Call-&-Bearer-Setup.
NOTE 3: If relay node was not already associated with call, use Call-&-Bearer-Setup.
NOTE 4: If relay node was not already associated with call, or no call association existed between the relay node and the leaf serving node, use Add-Party-to-Bearer.
NOTE 5: If relay node was not already associated with the call use, Call-&-Bearer-Setup.
NOTE 6: If relay node was not already associated with call, or no call association existed between the relay node and the leaf serving node, use Add-Party-to-Bearer.

7.12.2.2 Non-Human User Version

This subclause is for the non-human user version.

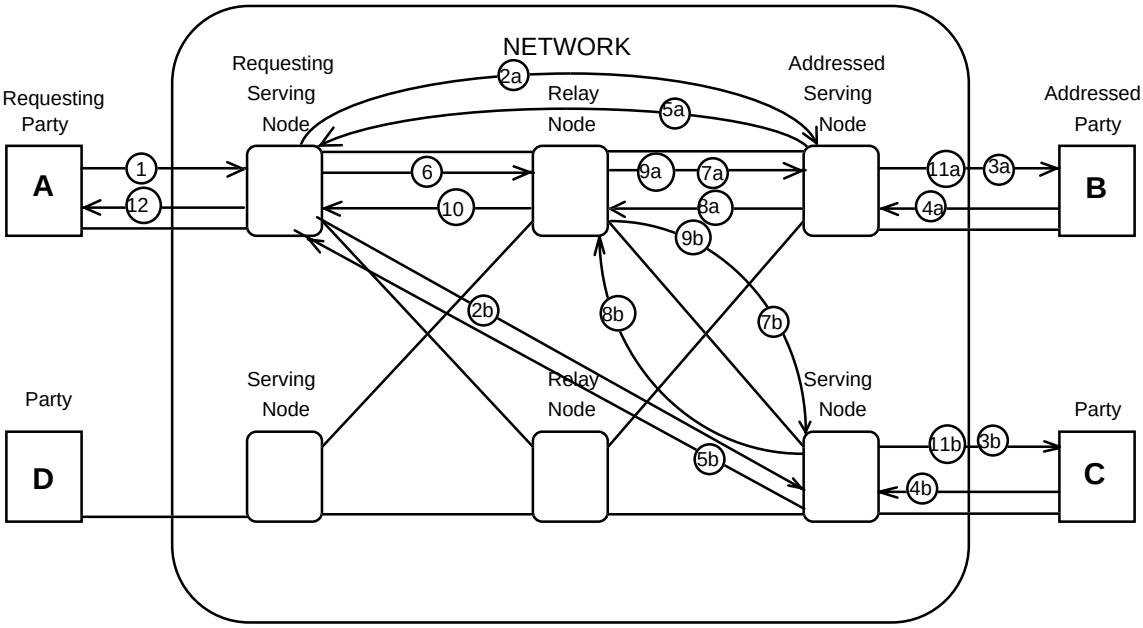


Figure 52: Look Ahead without State Change, non-human user version

Table 49

Flow No.	Call-&-Bearer Setup	Add-Bearer-to-Call	Add-Party-to-Bearer	Attach-Party-to-Bearer
1)	Call-&-Bearer-Setup.Ready	Add-Bearer-to-Call.Ready	Add-Party-to-Bearer.Ready	Attach-Party-to-Bearer.Ready
2a)	Lookahead.Begin	Lookahead.Begin	Lookahead.Begin	Lookahead.Begin
2b)	Lookahead.Begin	Lookahead.Begin	Lookahead.Begin	Lookahead.Begin
3a)	Call-&-Bearer-Setup.Begin	Add-Bearer-to-Call.Begin	Call-&-Bearer-Setup.Begin	Add-Bearer-to-Call.Begin
3b)	Call-&-Bearer-Setup.Begin	Add-Bearer-to-Call.Begin	Call-&-Bearer-Setup.Begin	Add-Bearer-to-Call.Begin
4a)	Call-&-Bearer-Setup.Ready	Add-Bearer-to-Call.Ready	Call-&-Bearer-Setup.Ready	Add-Bearer-to-Call.Ready
4b)	Call-&-Bearer-Setup.Ready	Add-Bearer-to-Call.Ready	Call-&-Bearer-Setup.Ready	Add-Bearer-to-Call.Ready
5a)	Lookahead.Ready	Lookahead.Ready	Lookahead.Ready	Lookahead.Ready
5b)	Lookahead.Ready	Lookahead.Ready	Lookahead.Ready	Lookahead.Ready
6)	Call-&-Bearer-Setup.Ready	Add-Bearer-to-Call.Ready (note 1)	Add-Party-to-Bearer.Ready (note 3)	Attach-Party-to-Bearer.Ready (note 5)

(continued)

Table 49 (concluded)

Flow No.	Call-&-Bearer Setup	Add-Bearer-to-Call	Add-Party-to-Bearer	Attach-Party-to-Bearer
7a)	Call-&-Bearer-Setup.Begin	Add-Bearer-to-Call.Begin (note 2)	Add-Party-to-Bearer.Begin (note 4)	Attach-Party-to-Bearer.Begin (note 6)
7b)	Call-&-Bearer-Setup.Begin	Add-Bearer-to-Call.Begin (note 2)	Add-Party-to-Bearer.Begin (note 4)	Attach-Party-to-Bearer.Begin (note 6)
8a)	Call-&-Bearer-Setup.Ready	Add-Bearer-to-Call.Ready (note 2)	Add-Party-to-Bearer.Ready (note 4)	Attach-Party-to-Bearer.Ready (note 6)
8b)	Call-&-Bearer-Setup.Ready	Add-Bearer-to-Call.Ready (note 2)	Add-Party-to-Bearer.Ready (note 4)	Attach-Party-to-Bearer.Ready (note 6)
9a)	Call-&-Bearer-Setup.Commit	Add-Bearer-to-Call.Commit (note 2)	Add-Party-to-Bearer.Commit (note 4)	Attach-Party-to-Bearer.Commit (note 6)
9b)	Call-&-Bearer-Setup.Commit	Add-Bearer-to-Call.Commit (note 2)	Add-Party-to-Bearer.Commit (note 4)	Attach-Party-to-Bearer.Commit (note 6)
10)	Call-&-Bearer-Setup.Commit	Add-Bearer-to-Call.Commit (note 1)	Add-Party-to-Bearer.Commit (note 3)	Attach-Party-to-Bearer.Commit (note 6)
11a)	Call-&-Bearer-Setup.Commit	Add-Bearer-to-Call.Commit	Call-&-Bearer-Setup.Commit	Add-Bearer-to-Call.Commit
11b)	Call-&-Bearer-Setup.Commit	Add-Bearer-to-Call.Commit	Call-&-Bearer-Setup.Commit	Add-Bearer-to-Call.Commit
12)	Call-&-Bearer-Setup.Commit	Add-Bearer-to-Call.Commit	Add-Party-to-Bearer.Commit	Attach-Party-to-Bearer.Commit

NOTE 1: If relay node was not already associated with the call use, Call-&-Bearer-Setup.
NOTE 2: If leaf serving nodes are not already associated with call or no call association existed between relay node and the leaf serving node, use Call-&-Bearer-Setup.
NOTE 3: If relay node was not already associated with call, use Call-&-Bearer-Setup.
NOTE 4: If relay node was not already associated with call, or no call association existed between the relay node and the leaf serving node, use Add-Party-to-Bearer.
NOTE 5: If relay node was not already associated with the call use, Call-&-Bearer-Setup.
NOTE 6: If relay node was not already associated with call, or no call association existed between the relay node and the leaf serving node, use Add-Party-to-Bearer.

8 Signalling interworking requirements

8.1 Introduction

When a new or enhanced signalling protocol is introduced into a network it is important that existing operations are not impaired. It is important that any signalling requirements for B-ISDN, should take into account the need to maintain the stability of existing standards.

8.1.1 Possible solutions

Since the target solution is longer term the probability is that a completely stand alone network will support B-ISDN; however two possible options can be applied (the selection will depend on the needs of a network operator):

a) integration with the existing network;
b) a stand alone overlay network.

With the second option contact with the existing network will be through clearly identified interworking nodes and there will be a requirement to clearly specify the interworking mechanism to be employed. Impact on existing operations will be minimal and policing at those nodes common to both networks to ensure no "overflowing" of signalling should suffice. The solution itself being new should not impact on the existing standard.

The first option will present greater problems in minimising the impact of an enhanced protocol. To avoid abortive processing and signalling, "signalling route" policing will be required, i.e. B-ISDN signalling will only be allowed on to the appropriate routes. In addition the enhancements to the existing protocol and procedures should be in accordance with the rules for compatibility contained in current recommendations.

In the last case, not only should existing operations be protected but the standards will also need to be secured form any possible de-stabilisation, as a result of enhancement. This can be achieved by using a similar approach to producing the enhancement as that employed for the existing standard. As stated above any enhancement produced should align with the specified compatibility requirements for procedures, messages etc., for the standard being enhanced.

Protection of existing standards, compatibility and ease of interworking are important requirements of any signalling solution produced for B-ISDN.

8.1.2 Communication scenarios

Considering the interworking configuration of ISDN having both 64 kbit/s based ISDN capabilities and broadband capabilities, the communication scenarios are identified as shown in figure 53. These scenarios also apply to any 2 networks with different signalling protocols.

- Scenario I is an interexchange scenario between B-ISDN and 64 kbit/s based ISDN.
- Scenario II is a network concatenation interworking scenario, but the interfaces and services are the same as those which are currently provided by 64 kbit/s based ISDN.
- In Scenario III, the service capabilities provided between broadband user access points are restricted to 64 kbit/s based ISDN capabilities.
- In Scenario IV, end-to-end access has broadband capabilities and can provide the services currently provided by 64 kbit/s based ISDN. The 64 kbit/s based ISDN services provided by this scenario are similar to those provided by scenario I and III.

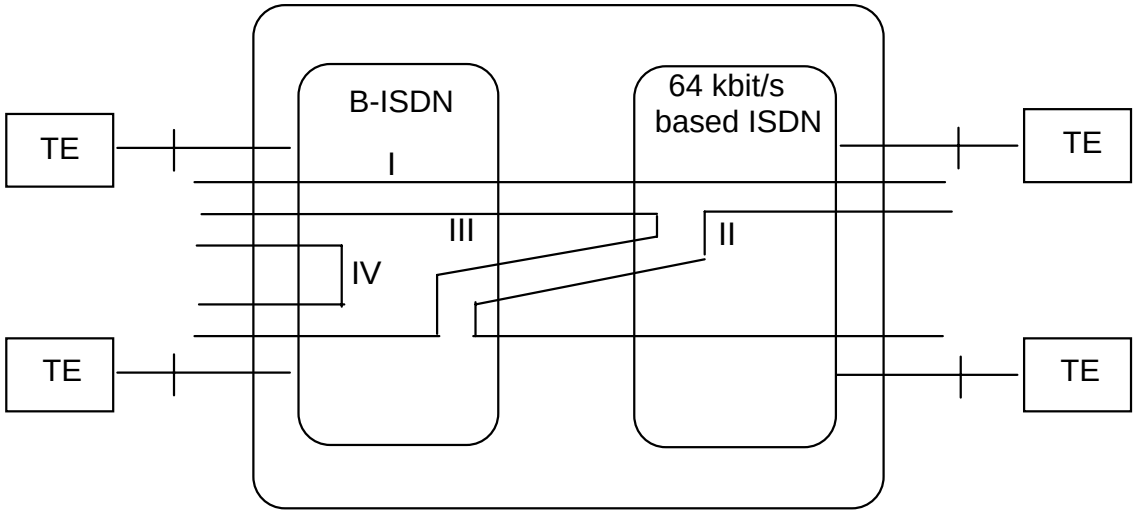


Figure 53: Communication scenarios between B-ISDN and 64 kbit/s based ISDN

8.2 N-ISDN (64 kbit/s based) Interface Requirements

From the B-ISDN user part view, 64 kbit/s based ISDN services are accessible to them without any restriction. In order to fulfil this requirement, two possible interface scenarios may be considered:

Scenario A: To provide access to 64 kbit/s based ISDN through a B-ISDN (figure 54), or;
Scenario B: To integrate all 64 kbit/s based ISDN services in B-ISDN with a broadband signalling protocol (figure 55).

1) Scenario A.
In this scenario a transparent ATM connection, either permanent, semi-permanent or on demand is used. The corresponding bearer service is Broadband Connection Oriented Bearer Service-A as described in Recommendation F.811.
The introduction of ATM in B-ISDN should allow the functionality's operated in the existing ISDN be re-used.
It should be possible to allow the N-TEs via B-ISDN premises to get access to ISDN services. One possible arrangement may require the B-ISDN to provide transparent connections to connect these N-TEs to an IWU.
This scenario may have two possible cases:

Case 1: Emulation of the B and D channels of basic and primary rate access;
Case 2: Emulation of the basic and primary rate access.

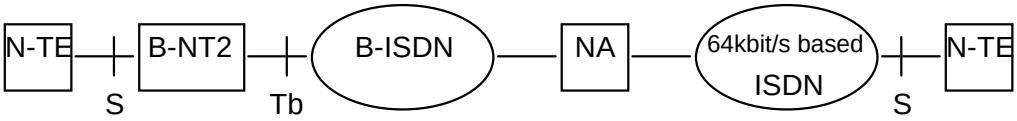


Figure 54: Scenario A

2) Scenario B.
In this scenario, network interworking takes place by interconnecting trunk line with broadband signalling protocol.
As a basis for interworking with N-ISDN, the support of N-ISDN services as defined in Rec. Q.767 according to the most recent stage 1 and stage 2 recommendations will be supported. The number of interworking units for one connection should be minimized to prevent degradation of QoS.

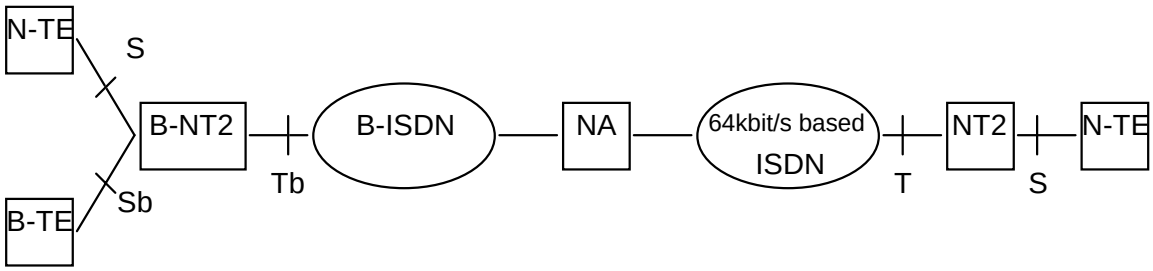
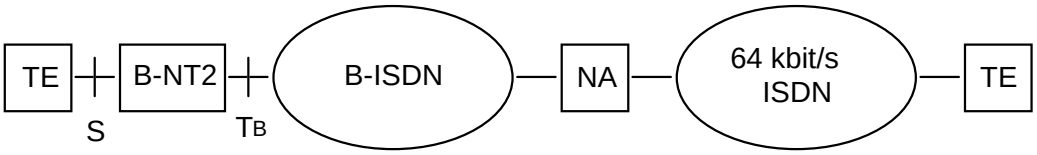


Figure 55: Scenario B

Scenario A

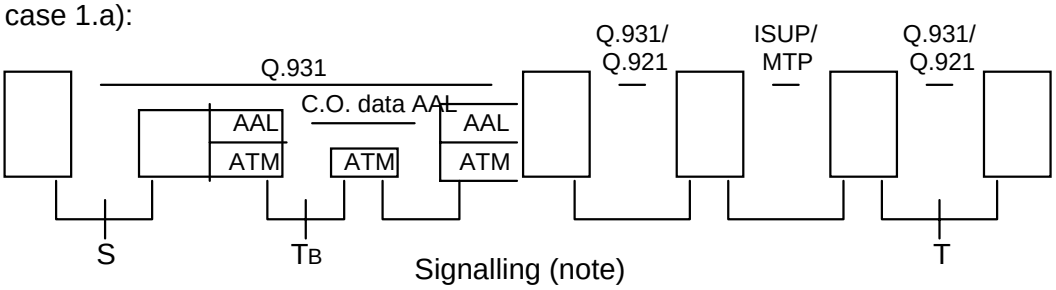
This scenario, as depicted in figure 56, may have two possible cases:

* Case 1: Emulation of the B and D channels
  a) B channels supported by AAL Type 1
     D channel supported by "connection oriented data AAL"
  b) B channels supported by AAL Type 1
     D channel supported by AAL Type 1

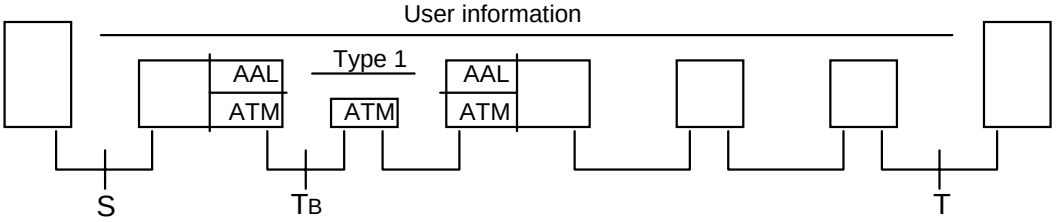


Interworking configuration

case 1.a):

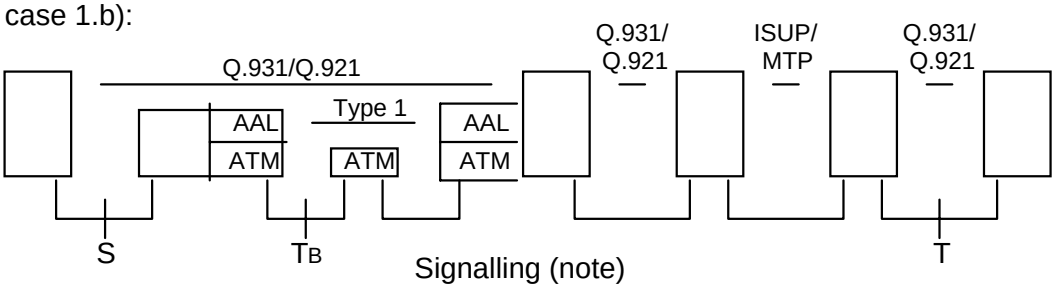


Signalling (note)

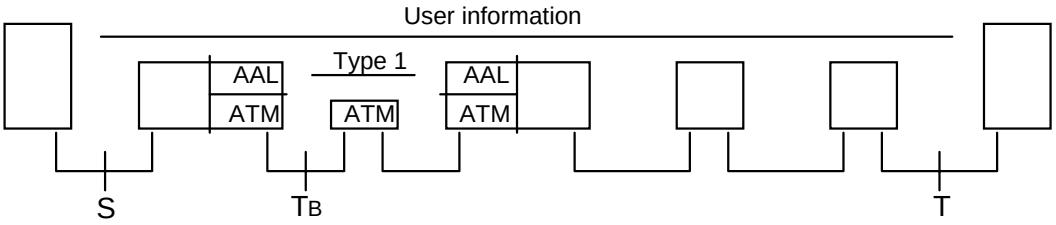


User information transfer (interworking in the U-plane)

case 1.b):



Signalling (note)



User information transfer (interworking in the U-plane)

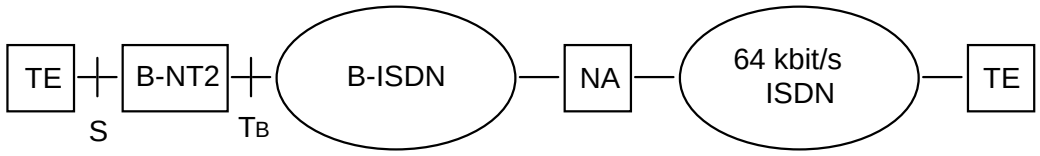
NOTE: In this Scenario C-Plane is not used for signalling for call establishment within B-ISDN. The signalling protocol will be transferred through ATM connection as user information.

Figure 56 (part 1 of 2): Example of interworking Scenario A

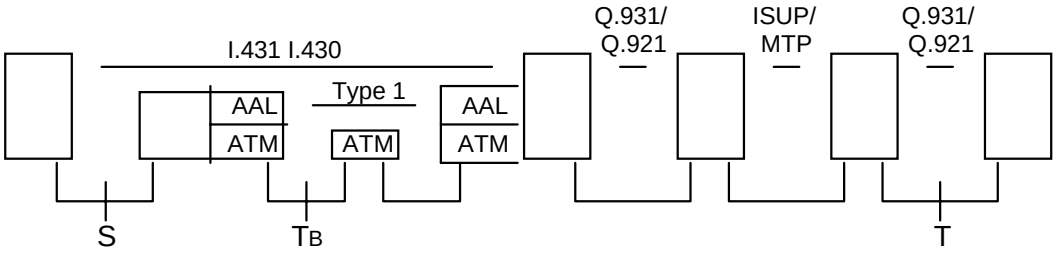
* Case 2: Emulation of the Basic and Primary rate access

1) case 1: Emulating the B/D channels of basic and primary rate access
   A B-NT2 is used to connect the TEs (64 kbit/s TEs) to the B-ISDN. At the physical layer activation of the TEs, one permanent connection (semi-permanent or demand ATM connection) will link the TEs to the NA. The B-NT2 will packetize/depacketize information from/to D channel in ATM cell flow. The Data Link Connection will be established between the TEs and the NA and the Q.921 TEI procedures may then take place under the control of the NA. The NA may then play the role of the NT2 or ET. Following that step the circuit connection control protocol of Q.931 may take place. During the circuit connection establishment phase, the allocation of the B channels will result in dynamic establishment and allocation of ATM connections between the B-NT2 and the NA.
2) case 2: Emulating the basic and primary rate access
   In this scheme the B-T2 will have a very generic function which is reduced to the circuit emulation function foreseen in AAL class A. The 64 kbit/s S interface will be prolonged from the B-NT2 to the NA. In this Scenario the B-ISDN is transparent to the 64 kbit/s ISDN circuit switched call control. The NA is seen from the B-ISDN as a specialized Service provider (Server for 64 kbit/s ISDN services access). It is viewed as the NT2 or the ET from the 64 kbit/s based ISDN network.
   Two subcases may be distinguished:
   a) I.431 emulation: In this case the whole physical layer I.431 is emulated by the ALL connection class A;
   b) I.430 emulation: in this case only the 2B + D (144 kbit/s) is emulated by the AAL connection class A. The Echo channel E handling will be taken over by the B-NT2 device.

case 2):



Interworking configuration



Signalling (note) and User information transfer

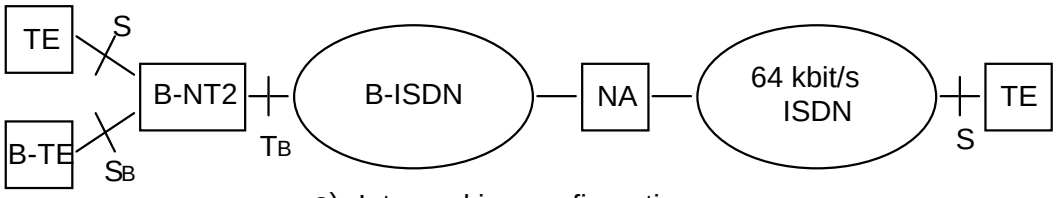
NOTE: In this Scenario C-Plane is not used for signalling for call establishment within B-ISDN. The signalling protocol will be transferred through ATM connection as user information.

Figure 56 (part 2 of 2): Example of interworking Scenario A

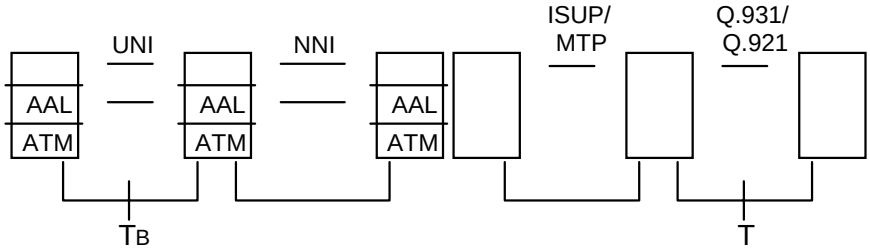
Scenario B

This scenario is depicted in figure 57. In this Scenario B-ISDN shall support not only Broadband capabilities but also 64 Kbit/s based ISDN capabilities.

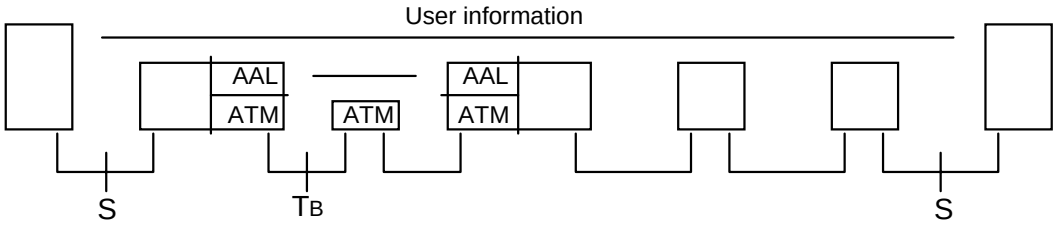
Interworking functions must take into account the mapping of protocols with respect to coding, sequencing, timing, etc. These mapping functions employed for interworking between Broadband service and 64 kbit/s ISDN services require further study.



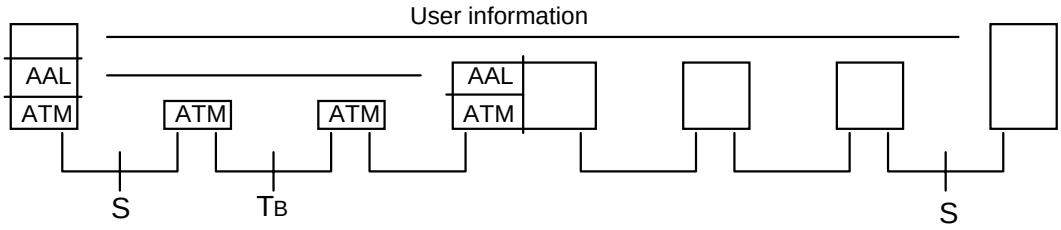
a) Interworking configuration



b) Signalling (interworking in the C-plane)



c-1) User information transfer (interworking in the U-plane)
64 kbit/s based ISDN TE connected to B-ISDN



c-2) User information transfer (interworking in the U-plane)
B-TE supporting 64 kbit/s based ISDN services

Figure 57: Example of interworking Scenario B

8.2.1 Interworking requirements for Access signalling

In broadband ISDN en bloc sending and receiving is recommended. For access signalling in the case of narrowband ISDN and PSTN interworking overlap sending and receiving may be applied.

8.2.2 Interworking requirements for Network signalling

Network signalling will support both overlap (sending and receiving) and en-bloc sending receiving.

8.3 Interworking of Release 1 and Release 2

8.3.1 Requirements for B-ISDN signalling protocol evolution

Taking into account the investment made for release 1 equipment, the requirements for B-ISDN signalling protocol evolution are:

a) Release 1 protocols applicable across the UNI and NNI are based on the corresponding narrowband ISDN protocols, i.e. the existing Q.931 and ISUP protocols respectively.
b) Release 1 terminals must be able to be connected to a Release 2 network and retain the release 1 services.
c) The network will allow Release 1 and Release 2 terminals simultaneously to be used on the same access in Release 2 networks.
d) Future B-ISDN signalling protocols shall support the use of Release 1 terminals which use the Release 1 UNI protocol.

8.3.2 Scenarios for Interworking

It is desirable that peer-to-peer compatibility between protocols for release 1 and future releases is achieved (for release 1 services).

It is required that Release 1 call/bearer control has sufficient functionality to provide the minimum set of Release 2/3 bearer control functionality.

Several scenario's of peer-to-peer communication are envisaged of how protocol evolution from Release 1 protocol to Release 2 protocol could take place by fully reusing the Release 1 protocol in a context sensitive manner controlled by the service.

8.3.2.1 Scenario A

Communication between a Release 1 exchange/terminal and a Release 2/3 exchange/terminal.

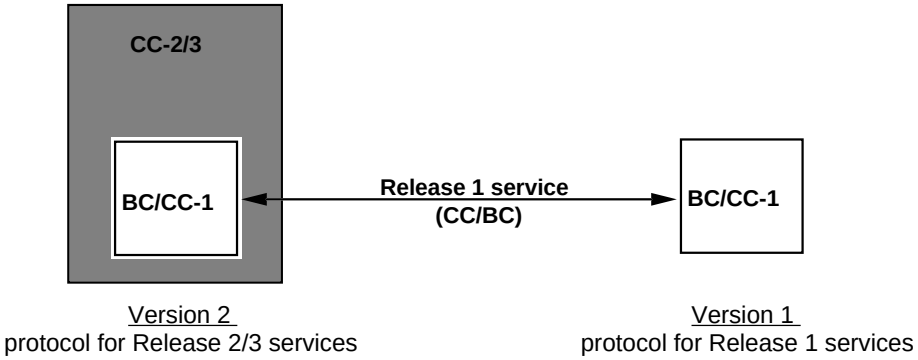


Figure 58: Scenario A

To provide Release 1 services, the call control function CC-2/3 is not used and would not be addressed by a call setup from a "version 1 exchange". For a call setup from a "version 2 exchange" to a "version 1 exchange" requesting a Release 2/3 service, the forward compatibility rules, which have to be included in Release 1 protocol, will result in normal unsuccessful case.

8.3.2.2 Scenario B

Communication between two Release 2 exchange/terminal in case of a Release 1 service request.

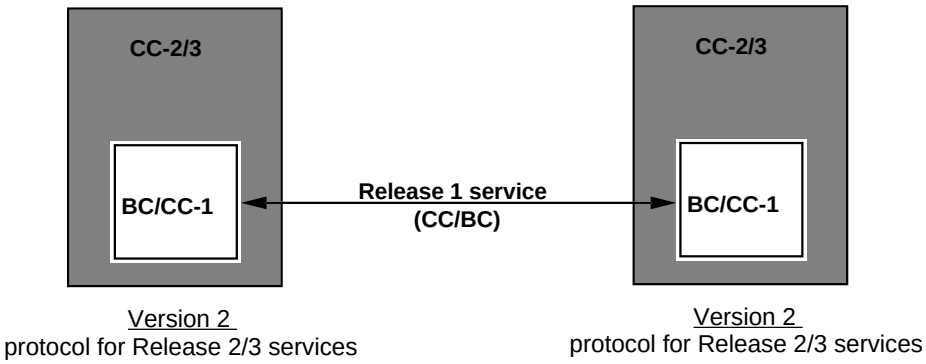


Figure 59: Scenario B

8.3.2.3 Scenario C

Communication between two Release 2 exchange/terminal in case of a Release 2/3 service request.

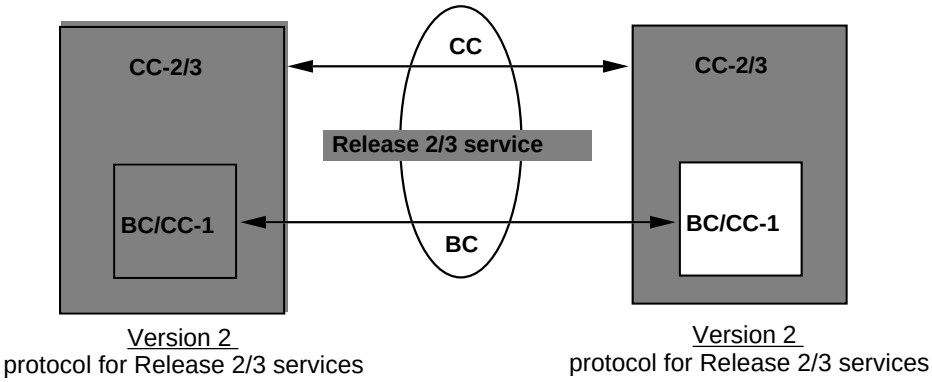


Figure 60: Scenario C

In this case from a context sensitive manner controlled by the service, the CC-2/3 function (supported by CC-2/3 protocol) is used instead of the CC-1 function. However, the bearer control protocol remains the same as in the previous cases shown above.

8.3.2.4 Scenario D

Communication via a version 1 transit node for Release 2/3 services.

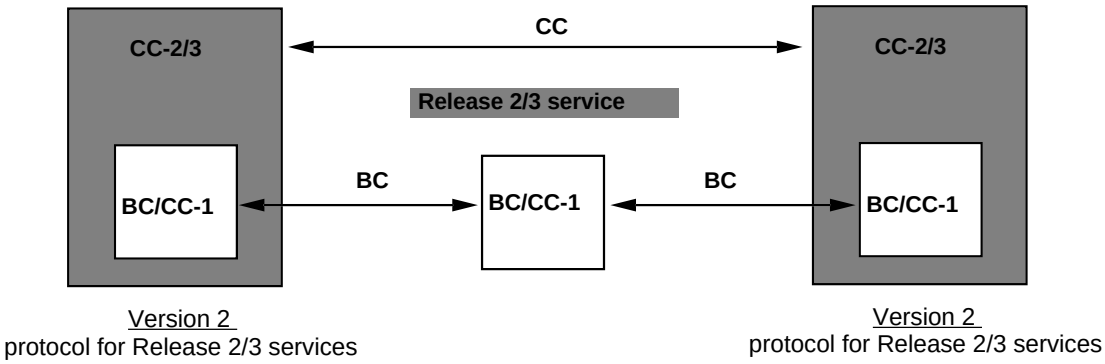


Figure 61: Scenario D

A "version 1 exchange" can serve as transit node for the bearer connection. It was agreed that there is the requirement for a Release 1 protocol to transfer information which is not understood transparently, or instruct appropriately.

8.3.3 Support of Release 2 services by using Release 1 equipment

For a number of Release 2 specific services, no specific Release 2 bearer control is needed, e.g. bandwidth modification or Common Route group. Therefore, in principle, it is possible to use Release 1 TEXs to establish the bearers for these services. This allows a smooth introduction of Release 2. This is depicted in figure 62.

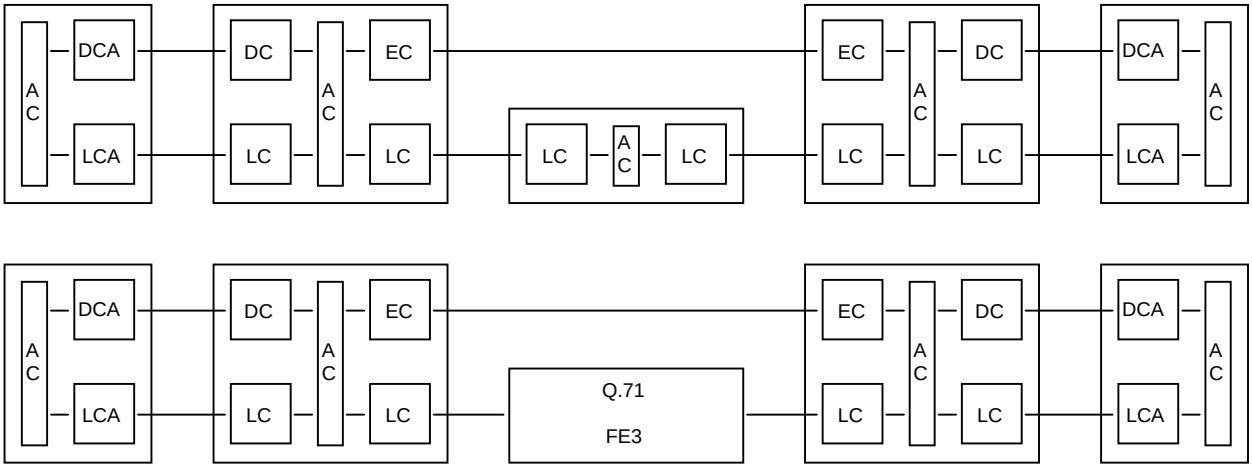


Figure 62: For some Release 2 specific services both Release 1 and 2 TEXs may be used

Annex A: SDL diagrams supporting atomic actions

A.1 Introduction

Specifications for CS2 are broken into several related SDL descriptions. Functional entities can be discriminated according to their function in the Functional Signalling Model. Those that are directly associated with a user are called a Functional Entity Agent (FEA). Others that are more closely associated with networking aspects are simply referred to as Functional Entities (FE).

Further discrimination can be made in terms of an entities role in a specific service request. The CCR model is useful in differentiating these roles. The roles determine which aspects of procedures a FE needs to perform. The SDLs in this subclause have been defined generically so that it is possible to reuse some in the FE actions.

Figure A.1 shows a FEA and its directly related FE. In this figure, these entities are not currently in the process of negotiating a service request, so they do not assume any CCR-related roles at this time and CCR-related nodes do not appear. The Local Views contain Information Model Objects, that persists beyond an atomic action. This information is structured as a set of related objects. Each Information Model Object has a separate identity and its own state. In this example, because there is no atomic action request in progress, all of the state of the FE an FEA entities is contained by their Information Model Objects. Other state machines residing in nodes of an atomic action structure are absent from this figure.

Figure A.2 shows additional state machines created to track a user-generated service request. These SDLs are based on CCR atomic action concepts. Four major types of SDLs are used. The state machine labelled "fea" is defined only in a FEA. The state machine labelled "sup" is defined by the SDL for a superior. The machine labelled "sub" is for a subordinate. The fourth is labelled "coord" for co-ordinator. This machine is subdivided into a Master Co-ordinator and an Intermediate Co-ordinator parts to simplify presentation, though these are really part of the same main Co-ordinator SDL. It is possible for a Co-ordinator to start out as a Master and, by delegating this authority to another FE, assume the role of an Intermediate. However, it is not currently envisioned as being useful to be able to go in the other direction.

CCR-based state machines exist for the duration of an atomic action, which has a lifetime that coincides with the negotiations that define or change Information Model Objects in the local view. They keep track of an instance of a request to change the contents of related Local Views. These machines do not exist for the lifetime of a telecommunication service.

Figure A.2 contains a snapshot of a portion of the nodes of a generic atomic action tree for a service request updating some Local Views. It illustrates the roles that FE and FEA entities can assume and the state machines associated with these roles.

FEA nodes require only fea machines, subordinate machines and Information Model Objects. This is because a FEA can only be a subordinate in an atomic action, the leaf of an action tree. FEs can require all machines and Information Model Objects, except the "fea"; however, a node assuming the master role does not use a subordinate machine.

Figure A.3 is an SDL for the lifecycle of a Generic Information Model Object. Figure A.4 shows an SDL for the application part of the FEA. Figure A.5 is the specification for the main part of the co-ordination, figure A.6 is the part of the Co-ordinator after a FE decides to retain the role of Master and figure A.7 is the part for co-ordination by an Intermediate. Figure A.8 specifies the subordinate role and figure A.9 defines the superior.

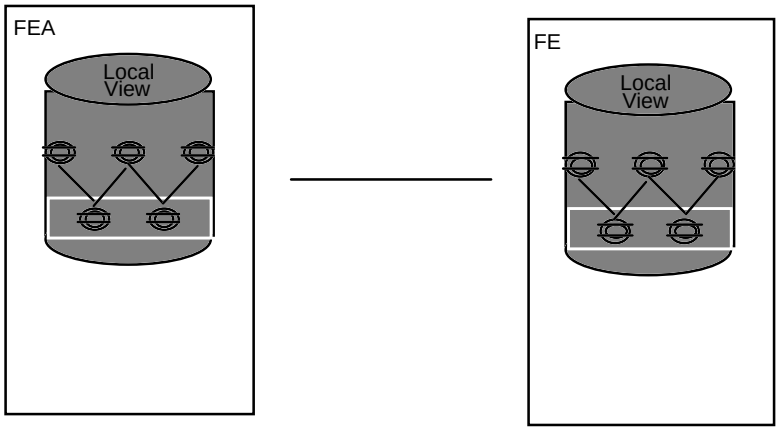


Figure A.1

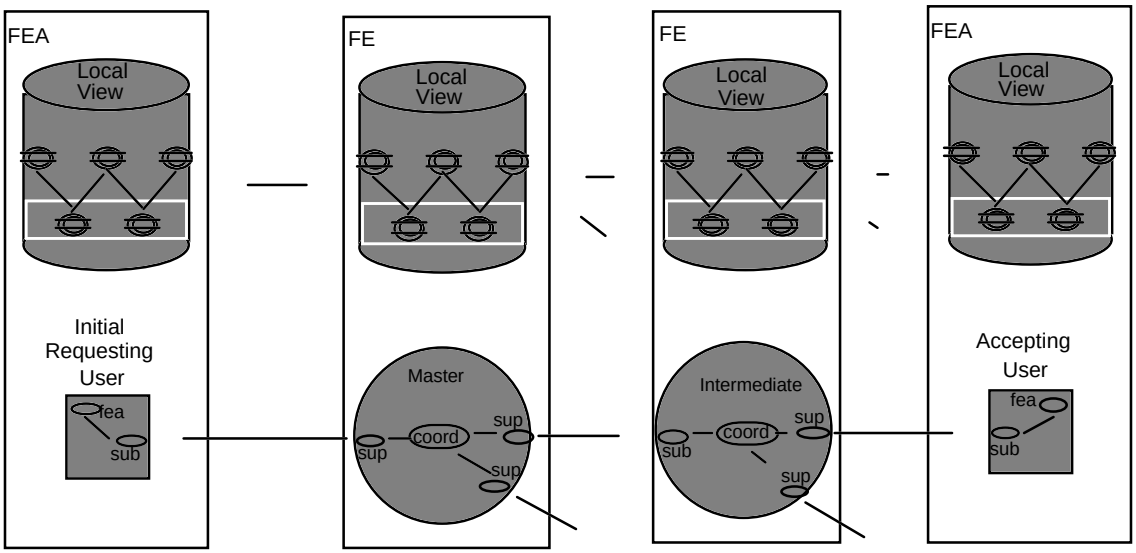


Figure A.2

A.2 Generic Information Model Object

The SDL for Generic Information Model Object (see figure A.3) describes procedures associated with Information Model Object in the Local View. The Local View may be that of a FE or a FEA. The views of a FEA and its associated FE should be essentially equivalent in topology and state, except for some transitional periods when they are exchanging call control information as part of an atomic action.

Instructions to invoke "signals" on Generic Information Model Objects depicted in this SDL are not transmitted independently in an instance of an information flow. These instructions are passed between call control entities in a "container" provided by the generic atomic action mechanism. These containers are defined in terms of CCR primitives (BEGIN, READY, etc.).

Processing of operations on the Generic Information Model Objects is performed within the context of the atomic action functionality of the control plane. Locks on Generic Information Model Objects are associated with all states except active and null. A locking mechanism is needed to isolate the affects of different atomic actions.

This SDL for Generic Information Model Objects is a simplification. It assumes all operations complete successfully. For example, it does not take into account what happens if a signal is sent to operate on an object that is locked, or if an unexpected signal is received, or the operation cannot be successfully carried out for other reasons.

There are a number of procedures that may be considered for dealing with contention for locks on Generic Information Model Objects. Complex serialisations mechanisms, such as those used by distributed database systems could be defined. But telecommunications signalling applications may not warrant these. A simple solution is to fail an operation that attempts to seize a locked object. This would result in the failure of the atomic action if the operation is mandatory. If the failure is propagated back to the FEA, this could be considered to be like a busy signal. It may attempt to repeat the action later. A variation is to have the FE entity retry the failed atomic action. The advantage of lock detection, rollback and retry strategies is that no deadlock detection is needed. Of course, there are other mechanisms to be considered.

The SDL in figure A.3 applies to all subclasses of the Generic Information Model Objects.

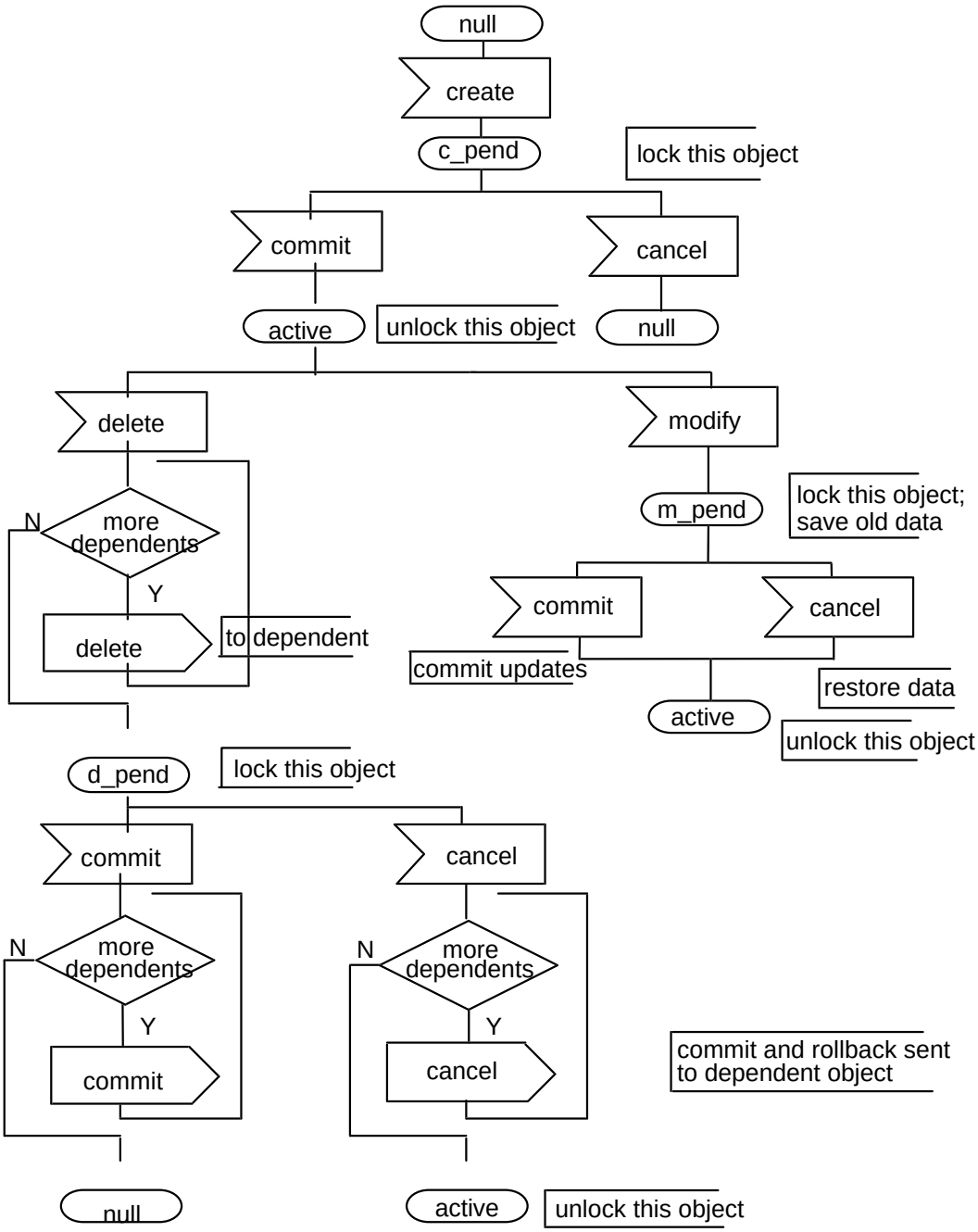

null
create
c_pend
lock this object
commit
cancel
active
unlock this object
null
delete
modify
N
more dependents
Y
delete
to dependent
m_pend
lock this object; save old data
commit
cancel
commit updates
restore data
active
unlock this object
d_pend
lock this object
commit
cancel
N
more dependents
Y
commit
N
more dependents
Y
cancel
commit and rollback sent to dependent object
null
active
unlock this object
SDL for Generic Information Model Object

Figure A.3

A.3 FEA Application Part

A FEA application part machine is created when a FEA becomes involved in an atomic action. There are two circumstances under which it is instantiated. One is when directly a user initiate an atomic action. This is called an Initial Requesting FEA (IRFEA). The second is when the local user needs to confirm an atomic action originated by another user. These are Intermediate Accepting or Final Accepting FEA (IAFEA or FAFEA), sometimes referred to as Accepting FEA as a group (AFEA).

An IRFEA first sends signals to the Generic Information Model object, updating its local view. These objects are locked at this time and enter a pending state, subject to commitment of the atomic action. The IRFEA then invokes a subordinate machine, have these operations also executed by its peer Fe. These operations are transferred in X.ready information flow from a subordinate in the FEA to a superior in the FE. The FE carries out the specified operations on the Generic Information Model objects too, keeping its local view and its FEAs in synchronization.

An AFEA is invoked by a Begin Indication, which contains a set of operations from its FE that need confirmation by its user. The FEA updates its local view; sending appropriate signals to lock the Generic Information Model objects that reflect the operations generated by its peer FE. It then passes the Begin Indication to its user and enters the Begin state, waiting for a response from the user. The user, assuming it is a mandatory party may accept or reject the atomic action. If the user rejects the request, FEA must undo the updates from the Begin as well.

If it accepts, the user may also be able to invoke additional operations within the scope of the atomic action. A benefit of using a generic mechanism, such as described here, is that it makes it easier for a user to make acceptance contingent on the successful completion of additional mandatory operations specified as part of its Accept[1)]. Optional operations can also be specified in the Accept. These additional operations on Generic Information Model objects are also reflected in a second updating to the Local View that is part of the same atomic action.

If either the FEA sends an Initiate or an Accept, the FEA Application Part must wait for a Commit Indication or a Cancel Indication from its subordinate. Based on the outcome, it finalises the pending updates upon Commit or undoes them on Cancel.

---

1) Existing protocols in general limit the "called party's ability to specify operations or define service attributes. Some service requests, such as those requiring pre-allocation of resources for "simultaneous setup" may operate in the traditional way. Also, users with some classes of service may be restricted in their ability to invoke such operations. However, they should be defined in the generic information flows, which should aim to provide a flexible mechanism, not imposing restrictions.

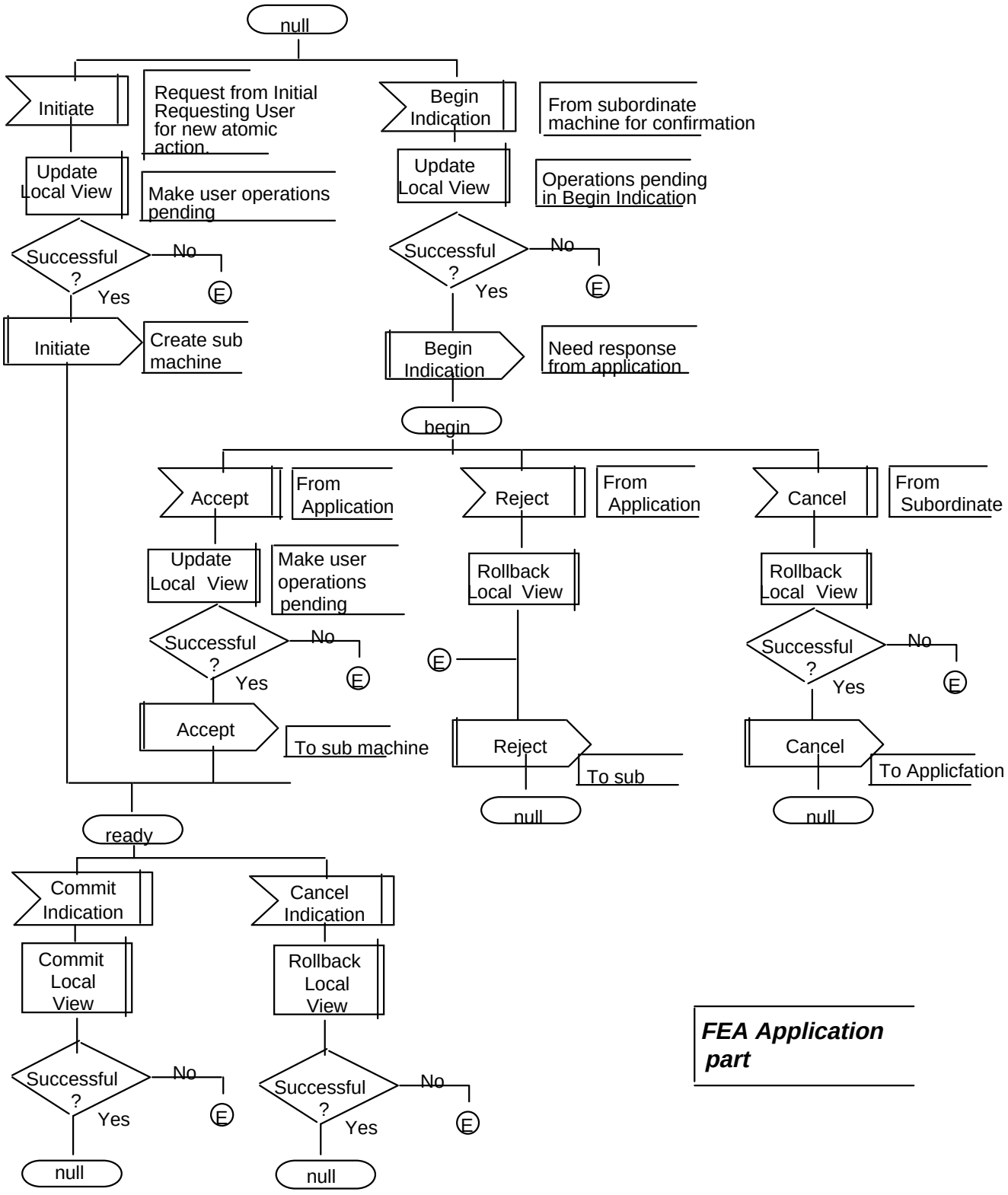

null
Initiate
Request from Initial Requesting User for new atomic action.
Update Local View
Make user operations pending
Successful ?
No
Yes
E
Initiate
Create sub machine
Begin Indication
From subordinate machine for confirmation
Update Local View
Operations pending in Begin Indication
Successful ?
No
Yes
E
Begin Indication
Need response from application
begin
Accept
From Application
Update Local View
Make user operations pending
Successful ?
No
Yes
E
Accept
To sub machine
Reject
From Application
Rollback Local View
E
Reject
To sub
null
Cancel
From Subordinate
Rollback Local View
Successful ?
No
Yes
E
Cancel
To Applicfation
null
ready
Commit Indication
Commit Local View
Successful ?
No
Yes
E
null
Cancel Indication
Rollback Local View
Successful ?
No
Yes
E
null
FEA Application part

Figure A.4

A.4 SDL for Co-ordinators

Figure A.5, figure A.6 and figure A.7 are SDLs for the Coronations functions in the Fes. It has been broken into three parts to simplify comprehension. Figure A.5 shows the Main Part, where the Co-ordinator is instantiated and remains until it passes the point at which it must decide to remain a Master for life if it has been initiated as a Master or if it to be become an Intermediate in an atomic action. Figure A.6 shows the part after a FE has decided to retain the Master role. Figure A.7 is the part after a FE has become an Intermediate. The main difference between the Master and Intermediate is that an Intermediate has a subordinate machine and must report back to its superior to commence the second phase of an atomic action. Only a Master has the authority to begin the second phase.

As can be seen in figure A.5, a Co-ordination machine can be invoked by a Ready Indication from a superior machine or a Begin Indication from a subordinate machine. If it is a Ready Indication as a result of a X.ready from an IRFEA or a FE delegating its master role, the Co-ordinator starts out at least initially as a master. If it is invoked by a Begin Indication, it can never be a master and must resign itself to being an intermediate for life.

If the Ready Indication came from a delegating FE, the Co-ordinator updates its local view based on the instructions sent with the X.ready. A X.ready from this other FE can contain operations on e.g. confirmed objects. It then invokes a Begin to create a superior machine to send a X.begin to its FEA containing operations requiring confirmation by its user and goes into the Begin state. Its waits for an expected Ready or Refuse Indication from the FEA. If it gets a Refuse, it rolls back any pending changes to its local view and rolls back the FE that delegated the master role by invoking its superior machine to the subordinate FE with a Cancel.

If it receives a Ready Indication from its FEA, it also updates its local view in accord with the operations on local and confirmed objects invoked by the FEA. At this point in time, if the FEA has invoked operations requiring confirmation by another user, it must decide if it is to stay master. At this point, the Co-ordinator also has an opportunity to interact with other Fes if there is a need based on the service request. For example, for some services, it may need to have resources pre-allocated in order to proceed.

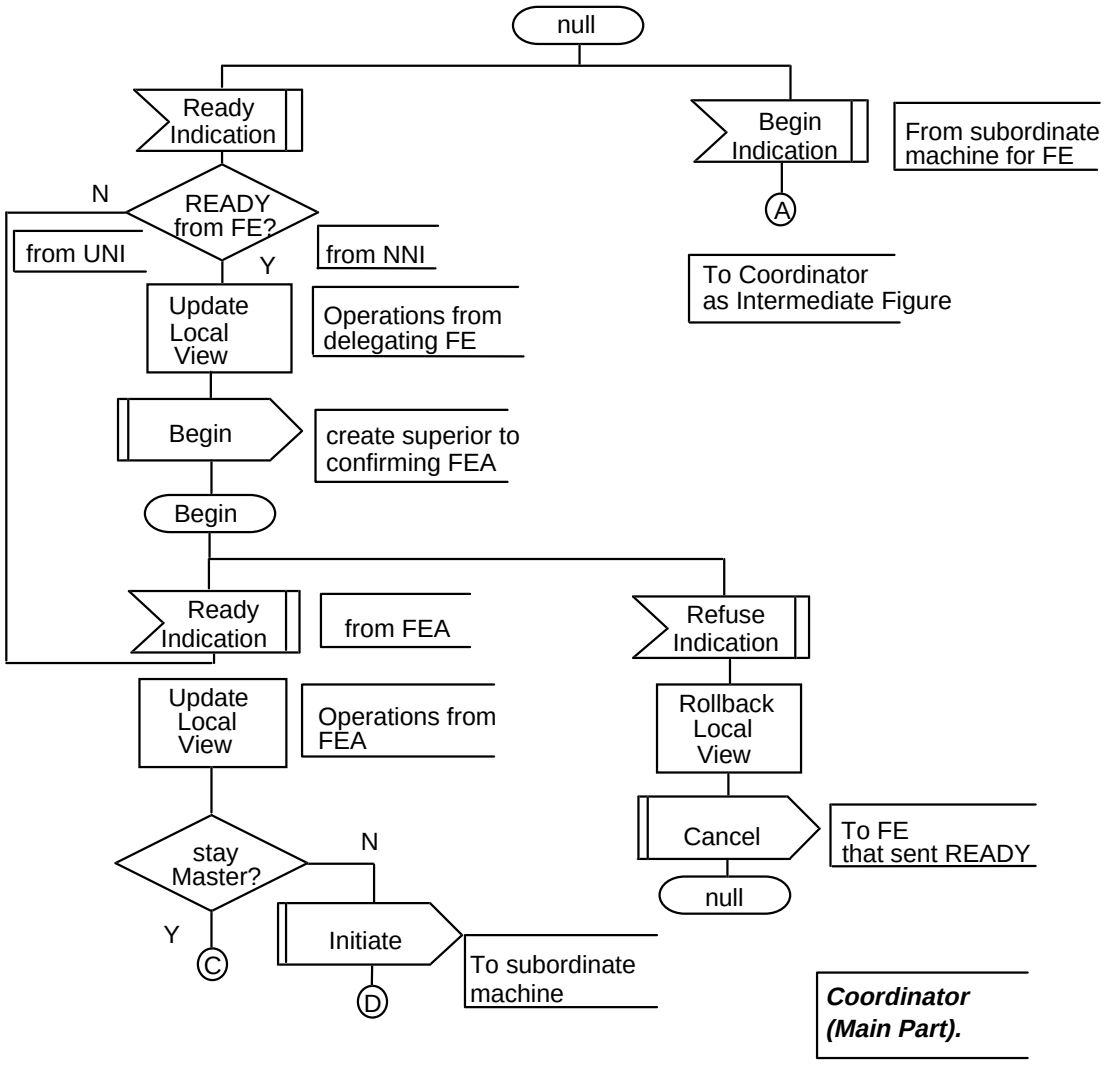


Figure A.5

Continuing in figure A.6, if the Co-ordinator remains master, it examines the operations invoked by its FEA to determine required interaction with other FE entities.

This is done by examining the operations in the Ready Indication for operations on confirmed objects, which require acceptance by the remote user. The Co-ordinator must invoke a Begin to create a superior machine for each confirming user it detects. The master, using operations in the Ready Indication and information in its local view, formats a set of operations to be sent in a X.begin to each confirming FE. After completing this, it goes into the wait state, awaiting responses for these outstanding Begins.

If it gets a Refuse Indications from any of the superior machines attached to its subordinates, it can immediate invoke its Cancel rollback) procedures for the atomic action. This entails rolling back its local views and generating cancel primitives to its other superior machines attached to its subordinates, including the one for its FEA.

If all of the superior machines for all of the outstanding Begins other FEs respond with Ready Indications, the Co-ordinator can invoke its Commit procedures, committing all the changes, unlocking Generic Information Model objects in the local view and sending Commits to all of its subordinates via its superior machines. The commits can contain operations on objects resulting from analysis of the Ready Indications.

Commit procedures also need to generate information flows to update local views of FE and FEA entities not directly involved in the atomic action but participants in an ongoing telecommunication service that is affected by the successful atomic action.

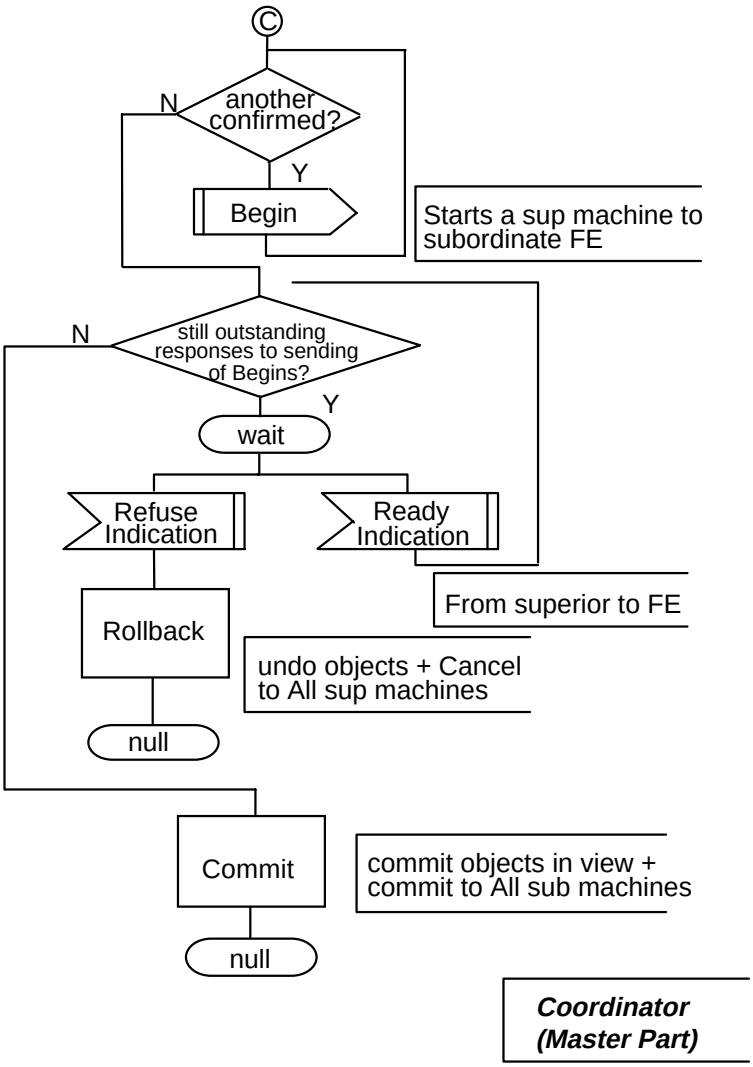


Figure A.6

The procedures for Co-ordination by an Intermediate are complicated by its interact with its superior via a subordinate machine (see figure A.7). The machine for Co-ordination by an Intermediate is instantiated in response to a Begin Indication from a superior that comes from its subordinate machine. The Begin Indication contains operations on one or more objects to be confirmed by the local user. These operations generate messages to Generic Functional Model objects affecting the local view.

The Co-ordinator may interact with other FE entities at this point. For example, it may need to co-ordinate the allocation of resources before preceding if the operations show that it is a simultaneous service request.

Then the Co-ordinator invokes a Begin to create a superior machine that has the local FEA as its subordinate, much like in the Main Part before the master has decided whether or not to retain its role. It then goes into the Begin state, waiting for a response from the local FEA. If the user does not accept, the Co-ordinator gets back a Refuse Indication from the superior machine created for the FEA. It then sends signals to the Generic Functional Model objects to rollback the local view and a Reject invocation to its subordinate machine so that it can X.cancel the atomic action to its superior.

If the user accepts the atomic action, the Co-ordinator gets a Ready Indication via the superior machine created for the FEA. This brings the Intermediate Co-ordinator to pretty much the same point as the Master Co-ordinator was when getting the Ready Indication from its FEA. It must update its local view. It can again interact with other FEs at this point and examine the operations to see if additional confirming users must have branches in this atomic action, if it is allowed. It may then invoke appropriate Begins to create superior machine for subordinate FEs if required. Then it waits for responses for outstanding Begins.

An important difference from the Master Co-ordinator is that when an Intermediate Co-ordinator gets back a Refuse Indication it invokes its subordinate machines with a Reject so that the subordinate machine can send a X.cancel to its superior FE. Also, if all the outstanding Begins get Ready Indication responses, it is not empowered to commit. It must generate a X.ready to its superior FE by invoking its subordinate machine with an Accept. It then goes into the Ready state, waiting for a Commit indication or Cancel Indication to come. Then it executes its Commit or Rollback process like the Master Co-ordinator.

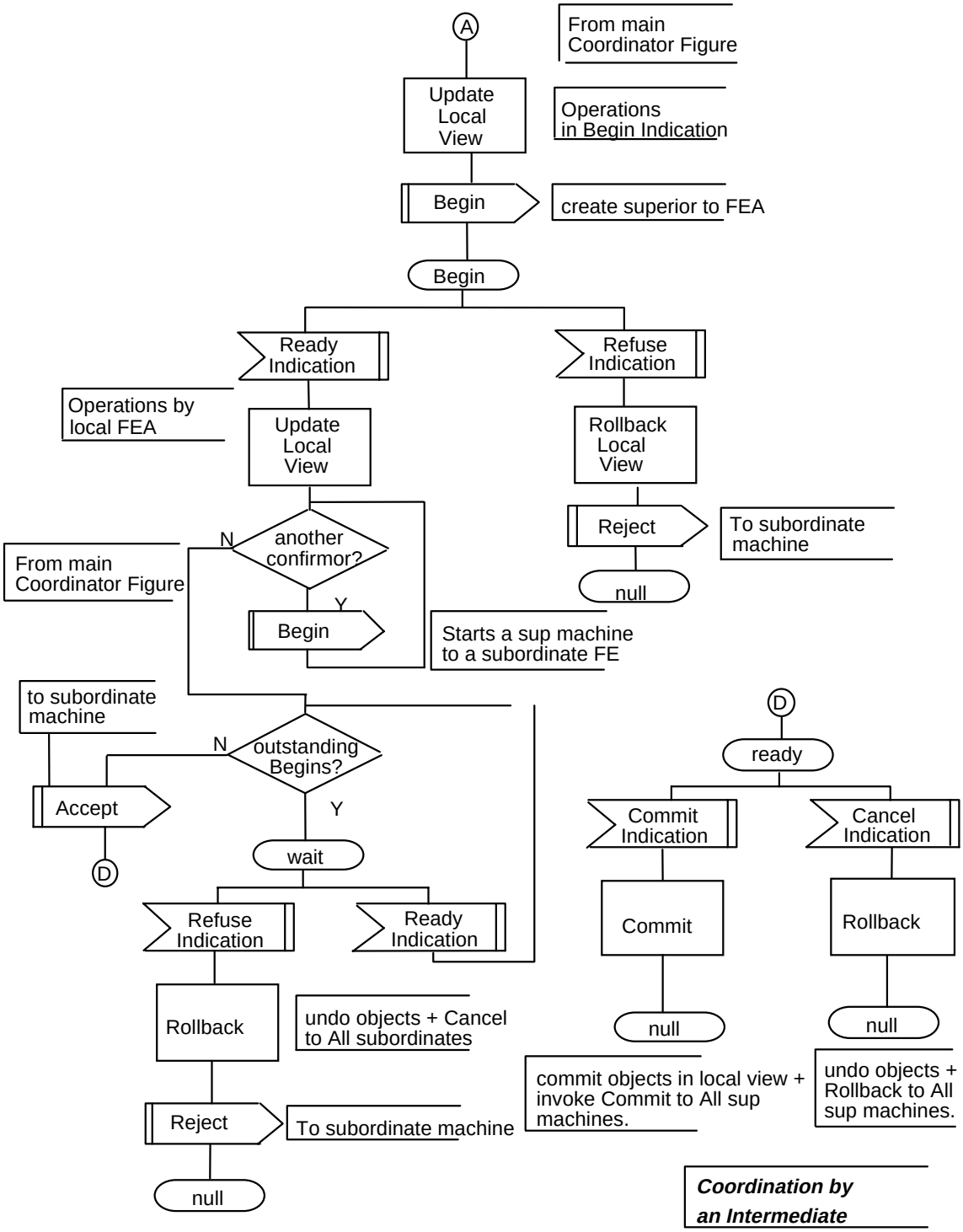


Figure A.7

A.5 SDL for a Subordinate

A subordinate machine is created when a FEA or FE becomes involved in an atomic action. An instance of a subordinate state machine exists within a FE that is an Intermediate but not a Master.

Figure A.8 illustrates that the state machine can be instantiated under two circumstances. One is when it is invoked with an Initiate primitive, bypassing the "Begin" part of the procedure. This can occur in a FEA when it is the IRFEA or in a FE when it is delegating its Master role to the another FE that becomes its superior.

In the second scenario, a superior sends a X.begin creating a branch to a subordinate. The superior may be in a FE sending a X.begin to its peer AFEA that requires a subordinate machine. Or another FE may send a X.begin to cause the receiver to generate a subordinate machine. Upon receiving a X.begin, there is a Begin Indication to instantiate a fea machine or intermediate co-ordinator.

Normally, the subordinate machine for a Co-ordinator remains in the Begin state until all recursively generated activity of entities at lower levels of the atomic action structure have completed the first phase of the two phase commit procedure. Based on the outcome of this activity, the intermediate co-ordinator sends to its subordinate machine an Accept or Reject response, which is reported back to its superior at a higher level in the action tree.

If the response to a Begin Indication is Accept or if the machine was invoked via the Initiate, a X.ready is sent by the subordinate machine to its superior. If the subordinate is a FE, the X.ready can contain operations on other objects. The X.ready from a FE can also contain information about results of optional operations. Of course, mandatory operations cannot fail in a X.ready, but negotiable parameters selected by the confirming side in a successful operation may also be reported.

In general, the subordinate state machine goes into the Ready state at this point, waiting for instructions from its superior to X.commit or X.rollback.

A X.commit may contain operations on other object that are passed in the Commit Indication. These operations update the local view immediately, as alluded to in the introductory remarks, bypassing the locking and pending states shown in figure A.3 and becoming active immediately.

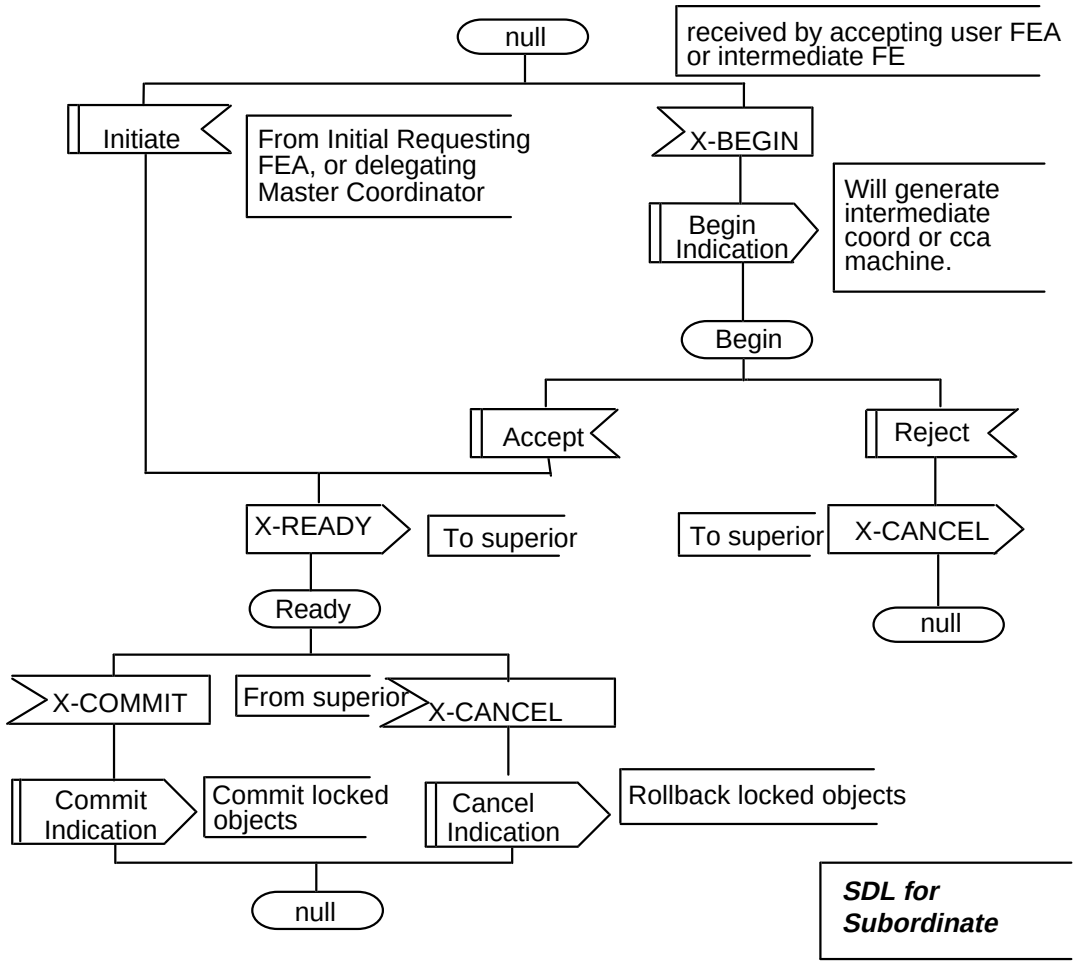

null
received by accepting user FEA or intermediate FE
Initiate
From Initial Requesting FEA, or delegating Master Coordinator
X-BEGIN
Will generate intermediate coord or cca machine.
Begin Indication
Begin
Accept
Reject
X-READY
To superior
To superior
X-CANCEL
Ready
null
X-COMMIT
From superior
X-CANCEL
Commit Indication
Commit locked objects
Cancel Indication
Rollback locked objects
null
SDL for Subordinate

Figure A.8

A.6 SDL for Superior

The SDL for a superior is depicted in figure A.9. As previously mentioned, it only exists in a FE, never a FEA. There are two circumstances causing a superior state machine to come into existence. First, it can be created upon receipt of a X.ready from either a IRFEA or a FE that is delegating its Master role. Second, it can be invoked by a Co-ordinator to create a branch to a subordinate that is either its associated FEA or a FE for another user. The X.begin creating the branch to the subordinate contains operations on Generic Information Model Objects to be performed by the entity receiving the X.begin.

After sending the X.begin, the superior machine goes into the X.begin state waiting for its subordinate to respond with a X.ready or X.cancel. A X.cancel causes the superior machine to generate a Refuse Indication to its Co-ordinator, which causes the atomic action to rollback.

When a X.ready is from a FEA, it may contain additional operations on local or confirmed object. When from another FE, it may contain results on optional operations, selected negotiable parameters and operations.

Receipt of a X.ready causes generation of a Ready Indication. When the X.ready is from an IRFEA or FE delegating its Master role, processing the Ready Indication requires instantiation of a Co-ordinator that can service as a Master (see figure A.5 and figure A.6). When the X.ready is a response to a X.begin, the Ready Indication is sent to the Intermediate Co-ordinator (see figure A.7) that invoked the Begin primitive instantiating this superior machine.

After the Ready Indication, the machine goes into the Ready state waiting for its Co-ordinator to invoke either a Commit or Cancel, sending its subordinate a X.commit or X.rollback respectively.

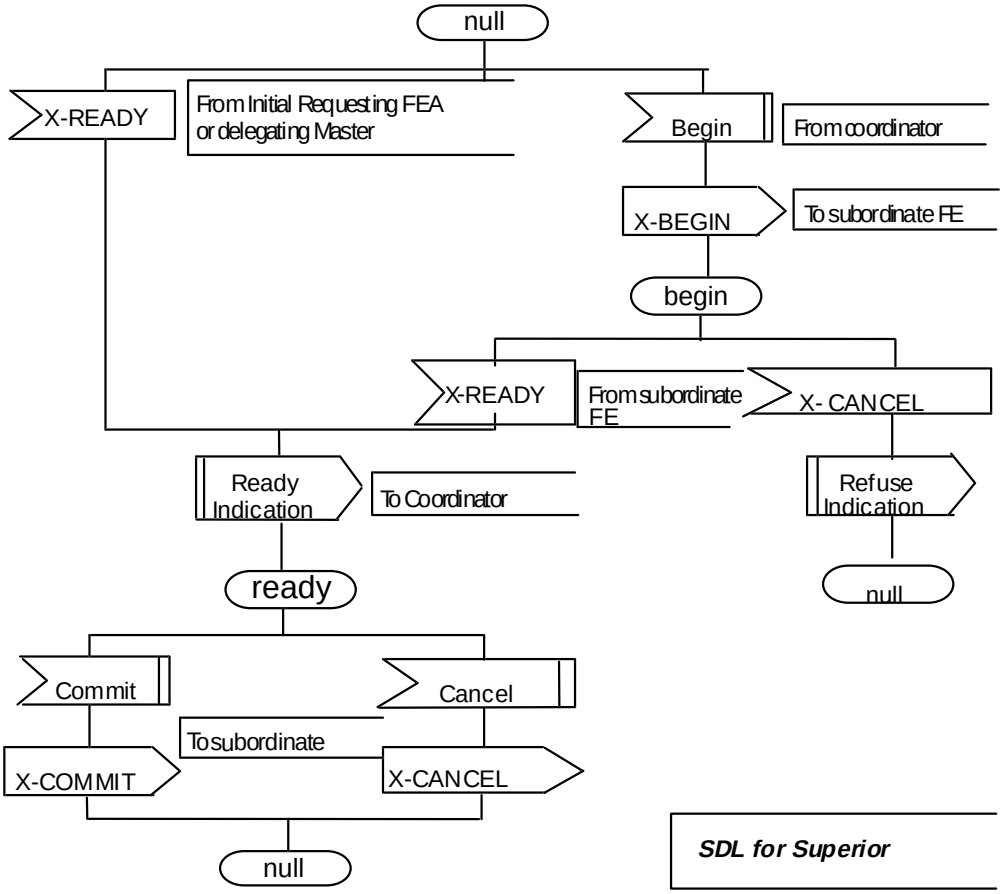


Figure A.9

Annex B: General connection types for possible future application

Clause 1 of this ETR defined the Connection Types that are applicable for Capability Set 2. This annex provides a more general representation of this information in the context of additional connection types.

Table B.1: Connection Topology Types

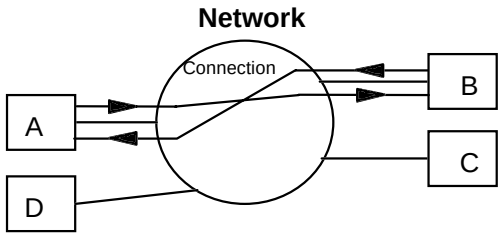
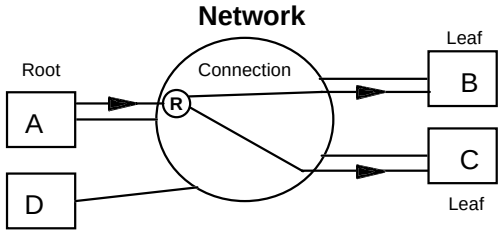
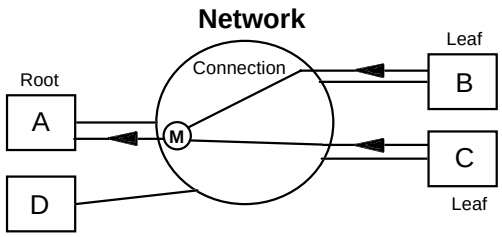
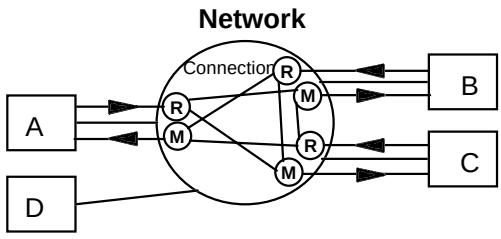
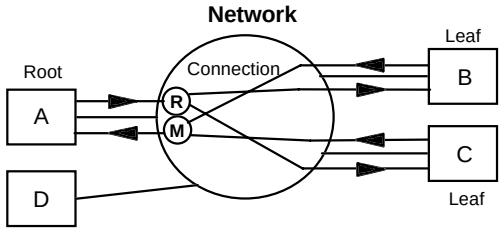
Type 1: Point-to-point Connection - a unidirectional or bi-directional connection between two terminals. A Point-to-point Connection may provide uni-directional or Bi-directional asymmetric communications between parties "A" and "B". This connection could be established, modified, or released in one of two ways:: Party "A" or "B" requests the action, or Party "C" or "D" may requests the action.	 Point-to-point Connection
Type 2: Point-to-multi-point Connection - a unidirectional connection from a single source to two or more sinks (note 4) A Point-to-multi-point Connection provides uni-directional communications from the "Root" Party "A" to "Leaf" Parties "B" and "C." This connection may be established, modified, or released in one of three ways: The "Root" party may requests the action, Either "Leaf" party may requests the action, or Party D may requests the action. The requester may be allowed to specify which called parties are required to agree to be part of the connection before the connection can be executed. "R" = Replication Function (note 2)	 Uni-Directional Point-to-multi-point Connection
Type 3: Multi-Point-to-point Connection - a unidirectional connection from two or more sources to a single sink (notes 1, 5) A Multi-Point-to-point Connection provides uni-directional communications from the "Leaf" Parties "B" and "C" to the "Root" Party "A". The bandwidth transmitted by the "Leaf" parties may be different. In addition, the bandwidth received by the "Root" Party may be different than the sum of the Transmitter's bandwidth. This connection may be established, modified, or released in one of three ways: The "Root" party may requests the action, Either "Leaf" party may requests the action, or Party D may requests the action. The requester may be allowed to specify which called parties are required to agree to be part of the connection before the connection can be executed. "M" = Merged Function (note 3)	 Uni-Directional Multi-Point-to-point Connection
(continued)	

Table B.1 (continued): Connection Topology Types

Type 4: Multi-Point-to-multi-point Connection - a connection in which each of the parties act as both source and sink so that each is receiving an orderly combination of information sent by every other party (notes 1, 6) A Multi-Point-to-multi-point connection provides the capability that all Parties can communicate. The receiver bandwidth of each party may be different that the sum of the bandwidths transmitted and may be different for each party. The bandwidths transmitted by each party may be different from each other and different than the bandwidth received. This connection may be established, modified, or released in one of two ways: Any party that will be associated with the connection may request the action, or Party D may request the action. The requester may be allowed to specify which called parties are required to agree to the connection before the connection can be executed.	 Multi-Point-to-multi-point Connection
Type 5 Bi-directional Point-to-multi-point Connection (notes 1, 7) A Bi-directional Point-to-multi-point connection provides communication between the "Root" Party "A" and between the "Leaf" Parties "B" and "C". This connection allows the "Root" Party to sent information to the "Leaf" Parties, while "Leaf" Parties only communicate with the "Root" Party. The Bandwidth received by the "Leaf" Parties may be different than the bandwidth transmitted by the "Root". The bandwidth transmitted by the "Leaf" Parties may also be different. In addition, the bandwidth received by the "Root" may be different than the sum of the bandwidth transmitted by the "Leaf" Parties. This connection may be established, modified, or released in one of three ways: The "Root" party may requests the action, Either "Leaf" party may request the action. or Party D may requests the action. The requester may be allowed to specify which called parties are required to agree to be part of the connection before the connection can be executed.	 Bi-directional Point-to-multi-point Connection
NOTE 1: Not Included in Capability set 2. NOTE 2: Replication may occur in one or more network nodes to provide the multiple routes. A Replication point is a point in a connection where user plane data received from one incoming data flow is replicated on two or more outgoing data flows. Replication is possible in the ATM Layer, within the AAL or in higher layers. Replication takes place within the ATM switch based on the VPI:VCI fields in the cell header. Each such cell arriving at the switch is copied onto one or more outgoing ATM cell streams, and within each cell stream onto one or more ATM virtual paths or virtual channels. Cell information field contents are not altered by this copying process. This form of Replication is simple to implement but is not suitable for those AALs that rely on point-to-point retransmission for recovery of lost data. ATM Replication points may occur in any node of the network. This type of Replication point is **required** for Capability Set 2. Replication takes place within an AAL handler attached to or integrated with the ATM switch. Each AAL packet is assembled from incoming cells, the user information is extracted, and this is then segmented again onto two or more outgoing AAL connections. Because the AAL is terminated and regenerated in this case, retransmission is handled directly by the Replication point, not passed back to the origin of the information. This type of Replication point is **not** supported by Capability Set 2. Higher Layer Replication takes places within a specialist server function of some kind. This type of Replication point is **not** supported by Capability Set 2.	

(continued)

Table B.1 (continued): Connection Topology Types

NOTE 3: Merging of source streams may occur in one or more network nodes to provide the multiple routes. A merging point is a point in a connection where user-plane data received from two or more incoming data flows is combined in a single outgoing data flow. A merging point may make use o ATM Layer, AAL, or higher Layer functionality. Since Type 3 connections are not required by Capability Set 2, none of the Merging Point sub-classes are supported by Capability Set 2.

In an ATM Layer merging point, cells from two different incoming cell streams are interleaved onto a single ATM virtual channel of a single outgoing cell stream. Cell information field contents are not altered by this interleaving process. This function does not work where the user information packets are more than a single cell, unless some form of multiplexing field is present in the SAR sub-layer of the AAL (i.e. in each cell). Where user information can be carried within a single cell, some form of identification of the point of origin.

An AAL merging point that operates such that user information from two or more ATM virtual channels are assembled into AAL SDUs. These are then multiplexed onto a single outgoing virtual channel if multiplexing is supported by the AAL, or sent sequentially on the outgoing channel if multiplexing is not supported (or is disabled).

A higher layer merging point is one that operates above the AAL. User information received from two or more incoming ATM virtual channels (via appropriate AAL handlers) is combined in some way, then enclosed in a new outgoing AAL for transmission via a single ATM virtual channel.

Examples of this are:

- combination of digitised audio signals into a single digitised audio signal;
- combination of multiple video pictures into a single "windowed" display;
- selection of a single video image from several images received, the choice being based on activity in an associated audio channel (to show the face of the current speaker in a conference);
- combination of music audio from one source with video images from another to form a single audio-visual presentation to the user.

NOTE 4: Type 2 connections may be used to support multicast or broadcast services:

A multicast connection is one in which sink parties are specified before the connection is established, or by subsequent operations to add or remove parties from the connection. The source of the connection will always be aware of all parties to which the connection travels.

Examples:

- A connection that is established to a list of end addresses.
- A connection that broadcasts to a community of terminals by some criteria other than network parties.
- A connection that is associated with a "distribution list" not visible to the source, such as an e-mail exploder.

A broadcast connection is one in which the sink parties are not always known to the source. The major difference from multicast is that for a broadcast connection, the connection to individual sink parties is not under the control of the source, but is by request of the each sink party.

Example:

- A broadcast connection to which any terminal can subscribe without reference to the source, perhaps subject to subscription restrictions.

- Broadcast connections are not required for Capability Set 2.

NOTE 5: Such connections may exist at the ATM layer if AAL Type 3/4 or Type X is used.

NOTE 6: This connection may be implemented in the ATM layer using AAL Type 3/4 or Type X, or it may be implemented at a higher layer such as a telephony conference bridge. The important point is that all terminals can both send and receive in this configuration. The sender may or may not receive an echo of his own information depending on the service.

(continued)

Table B.1 (concluded): Connection Topology Types

NOTE 7: A Type 5 connection is formed when a Type 2 connection is overlaid on a Type 3 connection such that the source of the Type 2 is the same endpoint as the sink for the Type 3 (the "root"), and the sinks for the Type 2 can act as source for the Type 3 (the "leaves"), all source/sink parties being specified by the "root" endpoint.
This connection type may be implemented at the ATM layer using AAL Type 3/4 or Type X. The source of a feedback cell can be obtained from the MID field or equivalent, if necessary.
The feedback (Type 3) connection may have a non-uniform bandwidth allocation, perhaps increasing at each cell-interleaving point. Alternatively, the terminals use of the feedback route may be controlled by an in-band flow control mechanism (such as token passing). This may not be necessary where the feedback function is of very low bandwidth, such as one cell at intervals of several minutes.
Example of multicast Type 5 connection:
- Data broadcast with opportunity for recipients to request retransmission.
  Examples of broadcast Type 5 connections:
- Television show with viewers voting.
- Tele-shopping where customer identification (customer account number, etc.) is embedded in the feedback cell.

Annex C: Bibliography

- Draft new ITU-T Recommendation F.310: "Broadband Videotex Service".
- Draft new ITU-T Recommendation F.722: "Broadband Videotelephony".
- Draft new ITU-T Recommendation F.732: "Broadband Videoconference".
- Draft new ITU-T Recommendation F.811: "Broadband connection oriented bearer service".
- Draft new ITU-T Recommendation F.812: "Broadband connectionless data bearer service".
- Draft new ITU-T Recommendation F.821: "Broadband TV distribution service".
- Draft new ITU-T Recommendation F.822: "Broadband HDTV".
- Draft new ITU-T Recommendation I.122: "Frame Relay Service".
- Draft new ITU-T Recommendation I.555: "Frame relaying bearer service interworking".
- Draft new ITU-T Recommendation I.580: "General arrangements for interworking between B-ISDN and 64 kbit/s based ISDN".
- ITU-T Recommendation Q.71 (1993): "ISDN circuit mode switched bearer services".
- CCITT Recommendation Q.767 (1991): "Application of the ISDN user part of CCITT signalling system No. 7 for international ISDN interconnections".
- Draft new ITU-T Recommendation X.2: "Data Communications Services".

History

Document history	
March 1996	First Edition

ISBN 2-7437-0562-0 - Edition complète
ISBN 2-7437-0171-4 - Partie 1
Dépôt légal : Mars 1996