International Telecommunication Union

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

G.8080/Y.1304

(02/2012)

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

Packet over Transport aspects – Ethernet over Transport aspects

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

Internet protocol aspects – Transport

Architecture for the automatically switched optical network

Recommendation ITU-T G.8080/Y.1304

International
Telecommunication
Union

ITU-T G-SERIES RECOMMENDATIONS

TRANSMISSION SYSTEMS AND MEDIA, DIGITAL SYSTEMS AND NETWORKS

INTERNATIONAL TELEPHONE CONNECTIONS AND CIRCUITS	G.100–G.199
GENERAL CHARACTERISTICS COMMON TO ALL ANALOGUE CARRIER-TRANSMISSION SYSTEMS	G.200–G.299
INDIVIDUAL CHARACTERISTICS OF INTERNATIONAL CARRIER TELEPHONE SYSTEMS ON METALLIC LINES	G.300–G.399
GENERAL CHARACTERISTICS OF INTERNATIONAL CARRIER TELEPHONE SYSTEMS ON RADIO-RELAY OR SATELLITE LINKS AND INTERCONNECTION WITH METALLIC LINES	G.400–G.449
COORDINATION OF RADIOTELEPHONY AND LINE TELEPHONY	G.450–G.499
TRANSMISSION MEDIA AND OPTICAL SYSTEMS CHARACTERISTICS	G.600–G.699
DIGITAL TERMINAL EQUIPMENTS	G.700–G.799
DIGITAL NETWORKS	G.800–G.899
DIGITAL SECTIONS AND DIGITAL LINE SYSTEM	G.900–G.999
MULTIMEDIA QUALITY OF SERVICE AND PERFORMANCE – GENERIC AND USER-RELATED ASPECTS	G.1000–G.1999
TRANSMISSION MEDIA CHARACTERISTICS	G.6000–G.6999
DATA OVER TRANSPORT – GENERIC ASPECTS	G.7000–G.7999
PACKET OVER TRANSPORT ASPECTS	G.8000–G.8999
Ethernet over Transport aspects	**G.8000–G.8099**
MPLS over Transport aspects	G.8100–G.8199
Quality and availability targets	G.8200–G.8299
Service Management	G.8600–G.8699
ACCESS NETWORKS	G.9000–G.9999

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

Recommendation ITU-T G.8080/Y.1304

Architecture for the automatically switched optical network

Summary

Recommendation ITU-T G.8080/Y.1304 describes the reference architecture for the control plane of the automatically switched optical network as applicable to connection-oriented circuit or packet transport networks, as defined in Recommendation ITU-T G.805. This reference architecture is described in terms of the key functional components and the interactions between them.

History

Edition	Recommendation	Approval	Study Group
1.0	ITU-T G.8080/Y.1304	2001-11-29	15
1.1	ITU-T G.8080/Y.1304 (2001) Amd. 1	2003-03-16	15
1.2	ITU-T G.8080/Y.1304 (2001) Amd. 2	2005-02-22	15
2.0	ITU-T G.8080/Y.1304	2006-06-06	15
2.1	ITU-T G.8080/Y.1304 (2006) Cor. 1	2007-09-06	15
2.2	ITU-T G.8080/Y.1304 (2006) Amd. 1	2008-03-29	15
2.3	ITU-T G.8080/Y.1304 (2006) Amd. 2	2010-09-06	15
3.0	ITU-T G.8080/Y.1304	2012-02-13	15

FOREWORD

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

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

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

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

NOTE

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

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

INTELLECTUAL PROPERTY RIGHTS

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

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

Table of Contents

Page

Recommendation ITU-T G.8080/Y.1304

Architecture for the automatically switched optical network

1 Scope

This Recommendation specifies the architecture and requirements for the automatic switched transport network as applicable to connection-oriented circuit or packet transport networks, as defined in [ITU-T G.805].

This Recommendation describes the set of control plane components that are used to manipulate transport network resources in order to provide the functionality of setting up, maintaining and releasing connections. The use of components allows for the separation of call control from connection control and the separation of routing and signalling.

For the purposes of this Recommendation, components are used to represent abstract entities rather than instances of implementable software. UML-like notation is used to describe components of the architecture of the automatically switched optical network.

2 References

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

[ITU-T G.705]	Recommendation ITU-T G.705 (2000), *Characteristics of plesiochronous digital hierarchy (PDH) equipment functional blocks*.
[ITU-T G.800]	Recommendation ITU-T G.800 (2012), *Unified functional architecture of transport networks*.
[ITU-T G.805]	Recommendation ITU-T G.805 (2000), *Generic functional architecture of transport networks*.
[ITU-T G.7718]	Recommendation ITU-T G.7718/Y.1709 (2005), *Framework for ASON management*.
[ITU-T G.8081]	Recommendation ITU-T G.8081/Y.1353 (2010), *Terms and definitions for automatically switched optical networks*.
[ITU-T M.3100]	Recommendation ITU-T M.3100 (2005), *Generic network information model*.
[ITU-T X.25]	Recommendation ITU-T X.25 (1996), *Interface between Data Terminal Equipment (DTE) and Data Circuit-terminating Equipment (DCE) for terminals operating in the packet mode and connected to public data networks by dedicated circuit*.
[ITU-T Y.1311]	Recommendation ITU-T Y.1311 (2002), *Network-based VPNs – Generic architecture and service requirements*.
[ITU-T Y.1312]	Recommendation ITU-T Y.1312 (2003), *Layer 1 Virtual Private Network generic requirements and architecture elements*.

[ITU-T Y.1313]	Recommendation ITU-T Y.1313 (2004), *Layer 1 Virtual Private Network service and network architectures*.

3 Definitions

This Recommendation uses the following terms defined elsewhere:

3.1 **access group (AG)**: [ITU-T G.805].

3.2 **adaptation**: [ITU-T G.805].

3.3 **address**: [ITU-T G.8081].

3.4 **administrative domain**: [ITU-T G.805].

3.5 **agent**: [ITU-T G.8081].

3.6 **allocated (resource) label range**: [ITU-T G.8081].

3.7 **assigned SNPs**: [ITU-T G.8081].

3.8 **call**: [ITU-T G.8081].

3.9 **call segment**: [ITU-T G.8081].

3.10 **characteristic information**: [ITU-T G.800].

3.11 **closed user group**: [ITU-T X.25].

3.12 **component**: [ITU-T G.8081].

3.13 **configured (resource) label**: [ITU-T G.8081].

3.14 **connection**: [ITU-T G.8081].

3.15 **connection point (CP)**: [ITU-T G.8081].

3.16 **connection termination point (CTP)**: [ITU-T G.8081].

3.17 **control domain**: [ITU-T G.8081].

3.18 **control plane**: [ITU-T G.8081].

3.19 **control plane configured protection**: [ITU-T G.8081].

3.20 **E-NNI**: [ITU-T G.8081], See also clause 8.

3.21 **forwarding port (FPt)**: [ITU-T G.800].

3.22 **hard re-routing**: [ITU-T G.8081].

3.23 **I-NNI**: [ITU-T G.8081], See also clause 8.

3.24 **interface**: [ITU-T G.8081].

3.25 **layer network**: [ITU-T G.805].

3.26 **link**: [ITU-T G.805].

3.27 **link connection**: [ITU-T G.805].

3.28 **management plane**: [ITU-T G.8081].

3.29 **multi-homed**: [ITU-T G.8081].

3.30 **name**: [ITU-T G.8081].

3.31 **permanent connection**: [ITU-T G.8081].

3.32 **policy**: [ITU-T G.8081].

3.33 **port controller**: [ITU-T G.8081].

3.34 **potential (resource) label range**: [ITU-T G.8081].

3.35 **potential SNPs**: [ITU-T G.8081].

3.36 **re-routing domain**: [ITU-T G.8081].

3.37 **restoration**: [ITU-T G.8081].

3.38 **route**: [ITU-T G.8081].

3.39 **routing area**: [ITU-T G.8081].

3.40 **routing control domain**: [ITU-T G.8081].

3.41 **routing level**: [ITU-T G.8081].

3.42 **service level agreement**: [ITU-T G.8081].

3.43 **soft permanent connection (SPC)**: [ITU-T G.8081].

3.44 **soft rerouting**: [ITU-T G.8081].

3.45 **subnetwork**: [ITU-T G.8081].

3.46 **subnetwork connection (SNC)**: [ITU-T G.8081].

3.47 **subnetwork point (SNP)**: [ITU-T G.8081].

3.48 **subnetwork point pool (SNPP)**: [ITU-T G.8081].

3.49 **subnetwork point pool link (SNPP link)**: [ITU-T G.8081].

3.50 **supplementary services**: [ITU-T G.8081].

3.51 **switched connection (SC)**: [ITU-T G.8081].

3.52 **termination connection point (TCP)**: [ITU-T G.8081].

3.53 **third party signalling**: [ITU-T G.8081].

3.54 **trail**: [ITU-T G.805].

3.55 **trail termination point (TTP)**: [ITU-T G.8081].

3.56 **transitional SNPP link**: [ITU-T G.8081].

3.57 **transport domain**: [ITU-T G.8081].

3.58 **transport plane**: [ITU-T G.8081].

3.59 **transport resource identifier**: [ITU-T G.8081].

3.60 **user-network interface for the control plane (UNI)**: [ITU-T G.8081], see also clause 8.

3.61 **virtual private network**: [ITU-T Y.1311].

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

AG	Access Group
AGC	Access Group Container
CC	Connection Controller
CCC	Calling party Call Controller
CI	Characteristic Information

CoS	Class of Service
CP	Connection Point
CPS	Connection Point Status
CTP	Connection Termination Point
DA	Discovery Agent
DCN	Data Communication Network
E-NNI	External Network-Network Interface (reference point)
FCAPS	Fault, Configuration, Accounting, Performance, Security
FwP	Forwarding Point
HOVC	Higher Order Virtual Container
id	identifier
I-NNI	Internal Network-Network Interface (reference point)
LOVC	Lower Order Virtual Container
LRM	Link Resource Manager
MI	Management Information
MO	Managed Object
NCC	Network Call Controller
PC	Protocol Controller
PDP	Policy Decision Point
PEP	Policy Enforcement Point
RC	Routing Controller
SCN	Signalling Control Network
SNC	Subnetwork Connection
SNP	Subnetwork Point
SNPP	Subnetwork Point Pool
SPC	Soft Permanent Connection
SC	Switched Connection
TAP	Termination and Adaptation Performer
TCP	Termination Connection Point
TTP	Trail Termination Point
UML	Unified Modelling Language
UNI	User-Network Interface (reference point)
VPN	Virtual Private Network

5 Overview

The purpose of the automatic switched optical network (ASON) control plane is to:

– facilitate fast and efficient configuration of connections within a transport layer network to support both switched and soft permanent connections;

– reconfigure or modify connections that support calls that have previously been set up;

– perform a restoration function.

A well-designed control plane architecture should give service providers control of their network, while providing fast and reliable call set-up. The control plane itself should be reliable, scalable, and efficient. It should be sufficiently generic to support different technologies, differing business needs and different distribution of functions by vendors (i.e., different packaging of the control plane components).

The ASON control plane is composed of different components that provide specific functions including that of route determination and signalling. The control plane components are described in terms that place no restrictions regarding how these functions are combined and packaged. Interactions among these components, and the information flow required for communication between components are achieved via interfaces.

This Recommendation deals with the control plane architectural components and the interaction between the control plane, management plane and transport plane. The management and transport planes are specified in other ITU-T Recommendations and are outside the scope of this Recommendation.

Figure 5.1 provides a high-level view of the interactions of the control, management and transport planes for the support of switched connections of a layer network. Also included in this figure is the DCN, which provides the communication paths to carry signalling and management information. The details of the DCN, management plane and the transport plane are outside the scope of this Recommendation. Functions pertaining to the control plane are described in this Recommendation. Management of the control plane is specified in [ITU-T G.7718].

Control plane deployment will occur within the context of commercial operator business practices and the multi-dimensional heterogeneity of transport networks. These business and operational considerations lead to the need for architectural support of, for example, strong abstraction barriers to protect commercial business operating practices, segmenting transport networks into domains according to managerial and/or policy considerations, and inherent transport network heterogeneity (including control and management). The domain notion embodied in the [ITU-T G.805] definition of administrative domain and the Internet administrative regions (e.g., autonomous systems) has been generalized in the control plane architecture to express differing administrative and/or managerial responsibilities, trust relationships, addressing schemes, infrastructure capabilities, survivability techniques, distributions of control functionality, etc. Domains are established by operator policies and have a range of membership criteria, as exemplified above.

The control plane supports connection services (see Annex A) through the automatic provisioning of end-to-end transport connections across one or more domains. This involves both a service and connection perspective:

– The service (call) perspective is to support the provisioning of end-to-end services while preserving the independent nature of the various businesses involved.

– The connection perspective is to automatically provision "path layer" connections (in support of a service) that span one or more domains.

Connection state information (e.g., fault and signal quality) is detected by the transport plane and provided to the control plane.

The control plane carries (distributes) link status (e.g., adjacency, available capacity and failure) information to support connection set-up/release and restoration.

Detailed fault management information or performance monitoring information is transported within the transport plane (via the overhead/OAM) and via the management plane (including the DCN).

The interconnection between and within domains is described in terms of reference points. As domains are established via operator policies, inter-domain reference points are service demarcation points for a single service layer (i.e., points where call control is provided). The exchange of information across these reference points is described by the multiple abstract interfaces between control components. A physical interface is provided by mapping one or more abstract component interfaces to a protocol. Multiple abstract interfaces may be multiplexed over a single physical interface. The reference point between a user and a provider domain is the UNI, which represents a user-provider service demarcation point. The reference point between domains is the E-NNI, which represents a service demarcation point supporting multi-domain connection establishment. The reference point within a domain is an I-NNI, which represents a connection point supporting intra-domain connection establishment. The information flows across these reference points are further described in clause 8.

The control plane may also be subdivided to allow the segregation of resources for example between virtual private networks (VPNs). If the resources are dedicated to independent domains, then no reference points are provided between these domains. The case where a portion of the resources are dynamically shared is for further study.

Separate descriptions are provided for the interactions between the:

– control plane and transport plane layer networks; and the

– management plane and transport plane resulting from the addition of the control plane for connection management and connection monitor configuration.

This Recommendation encompasses the control of transport layer network connections, including inter-layer interactions arising from requests for capacity in server layers.

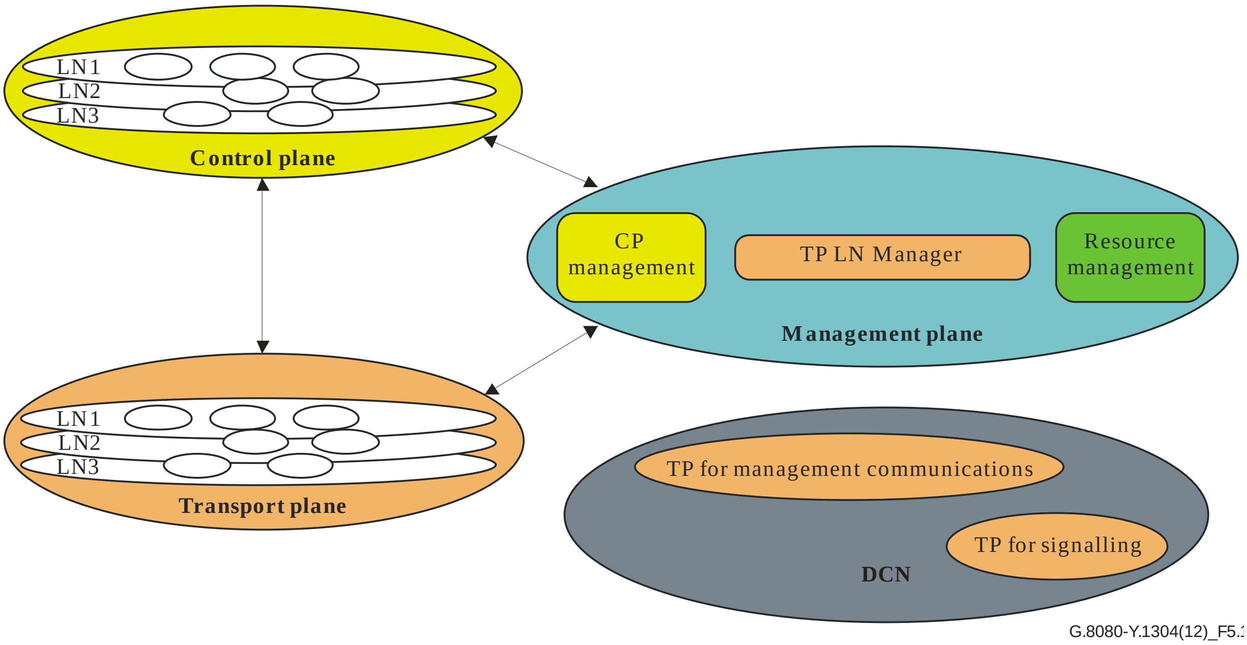


Figure 5.1 – Relationship between architectural components

5.1 Call and connection control

This Recommendation separates the treatment of call and connection control, as call control is only needed at domain boundaries (e.g., UNI, E-NNI). Thus, within a domain (i.e., I-NNI) it is only necessary to support procedures for connection control. Additionally, call control is provided at interlayer NCC boundaries. The functions performed by the call controllers at domain boundaries are defined by the policies associated by the interactions allowed between the domains. Policies are established by the operator. As such, an end-to-end call is considered to consist of multiple call segments, depending on whether the call traverses multiple domains. This allows for flexibility in the choice of signalling, routing and recovery paradigms in different domains.

It should be noted that the call is the representation of the service offered to the user of a network layer, while the connections are one of the means by which networks deliver said services. There may be other entities used in supporting calls, such as service specific processes.

5.1.1 Call control

Call control is a signalling association between one or more user applications and the network to control the set-up, release, modification and maintenance of sets of connections. Call control is used to maintain the association between parties and a call may embody any number of underlying connections, including zero, at any instant of time.

Call control may be realized by one of the following methods:

- separation of the call information into parameters carried by a single call/connection protocol;
- separation of the state machines for call control and connection control, whilst signalling information in a single call/connection protocol;
- separation of information and state machines by providing separate signalling protocols for call control and connection control.

Call control must provide coordination of connections (in a multi-connection call) and the coordination of parties (multi-party calls). To coordinate multiple connections, the following actions need to take place in the network:

- All connections must be routed so that they can be monitored by at least one coordinating (call control) entity.
- Call control associations must be completed before connections are set up. A call may exist without any connections (facilitating complex connection rearrangements).

A call can be considered to have three phases:

Establishment

During this phase, signalling messages are exchanged between users and the network to negotiate the call characteristics. The exchange of signalling messages between the calling party and the network is known as an outgoing call. The exchange of signalling messages between the network and the called party is referred to as an incoming call.

Active

During this phase, data can be exchanged on the associated connections and call parameters may also be modified (e.g., the addition of new parties in a point-to-multi-point call, where this type of call is supported).

Release

During this phase, signalling messages are exchanged between calling and called parties and the network to terminate the call. A call may be released by either the calling or called terminals or by proxy or network management.

5.1.2 Call admission control

Call admission control is a policy function invoked by an originating role in a network and may involve cooperation with the terminating role in the network. Note that a call being allowed to proceed only indicates that the call may proceed to request one or more connections. It does not imply that any of those connection requests will succeed. Call admission control may also be invoked at other network boundaries.

The originating call admission function is responsible for checking that a valid called user name and parameters have been provided. The service parameters are checked against a service level specification (a set of parameters and values agreed between a network operator and customer for a particular service indicating the '*scope*' of the service). If necessary, these parameters may need to be renegotiated with the originating user. The scope of this negotiation is determined by policies derived from the original service level specification, which itself is derived from the service level agreement (the service contract between a network operator and a customer that defines global responsibilities between them).

The terminating call admission function is responsible for checking that the called party is entitled to accept the call, based on the calling party and called party service contracts. For example, a caller address may be screened.

5.1.3 Connection control

Connection control is responsible for the overall control of individual connections. Connection control may also be considered to be associated with link control. The overall control of a connection is performed by the protocol undertaking the set-up and release procedures associated with a connection and the maintenance of the state of the connection.

5.1.4 Connection admission control

Connection admission control is essentially a process that determines if there are sufficient resources to admit a connection (or renegotiates resources during a call). This is usually performed on a link-by-link basis, based on local conditions and policy. For a circuit switched network, this may simply devolve to whether there are free resources available. In contrast, for packet switched networks such as ATM, where there are multiple quality of service parameters, connection admission control needs to ensure that admission of new connections is compatible with existing quality of service agreements for existing connections. Connection admission control may refuse the connection request.

5.1.5 Relationship between call state and connection state

The call state has dependency upon the state of the associated connections. This dependency is related to call type and policy. For example, where there is a single connection and it fails, the call may be immediately released, or alternatively, may be released after a period of time if no alternative connection can be obtained using mechanisms such as protection or restoration. Note that call and connection coincide at domain boundaries.

5.2 Interaction between control, transport and management planes

Figure 5.1 illustrates the general relationships between the control, management and transport planes. Each plane is autonomous, but some interaction will occur. The following provides further details on the interactions between the various planes.

5.2.1 Management – Transport interaction

The management plane interacts with transport resources by operating on a suitable information model, which presents a management view of the underlying resource. The objects of the information model are physically located with the transport resource, and interact with that resource via the management information (MI) interfaces of the layer-specific functional model. These interfaces should be collocated with the managed object and the control component.

5.2.2 Control – Transport interaction

Only two architectural components have a strong relationship to a physical transport resource.

At the lower limit of recursion, the connection controller (CC) provides a signalling interface to control a connection function. This component is physically located with the connection function and all further hardware details are hidden. However, given the limited information flow a new protocol may be useful to optimize this communication. The termination and adaptation performer (TAP) is physically located with the equipment that provides adaptation and termination functions, and provides a control plane view of link connections. The TAP hides the interaction with the hardware.

5.2.3 Management – Control interaction

Clause 7.1 states that each component has a set of special interfaces to allow for monitoring of the component operation, and dynamically setting policies and affecting internal behaviour. These interfaces are equivalent to the MI interface of the transport functional model, and allow the component to present a view to a management system and to be configured by a management system. This is discussed further in clause 7.1.

The management plane interacts with control components by operating on a suitable information model, which presents a management view of the underlying component. The objects of the information model are physically located with a control component, and interact with that component via the monitor and configuration interfaces of that component. These interfaces should be collocated with the managed object and the control component.

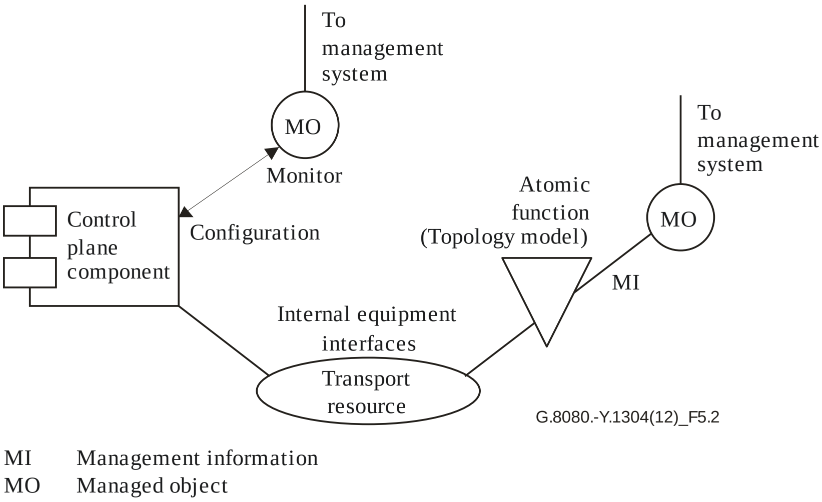


Figure 5.2 – Management/transport plane interactions with transport resources

The physical transport resources, which represent the physical reality of the equipment, are described in terms of ITU-T G.805 atomic functions. Managed objects (MO), which represent the external management view of the equipment, interact with the functional model specified in equipment recommendations via the MI reference points, which are also entirely within the equipment. Note that the managed object represents the management view regardless of the management protocol used. The information is independent of the protocol used.

From the control plane view, control plane components operate directly on the transport resources, so control plane operation appears autonomous to the management plane. Likewise, management plane operations appear autonomous to the control plane. This is exactly the same situation we have when multiple managers manage equipment. Each manager is unaware of each other's existence, and simply sees autonomous equipment behaviour. Although the information presented to the control plane is similar to that presented to management, it is not identical to the MI information. Control plane information overlaps the MI data because the control plane requires some but not all management information. For example, restoration is likely to be triggered by the same conditions that normally trigger protection actions.

Component-specific managed objects present a management view of control plane components via the monitor interfaces on the component. It is critical to realize that this is the view of the manageable aspects of the component, and not a view of the transport resource, which is obtained via the management view.

5.2.4 Resource management

Network resources may be partitioned between those under the authority of the management plane and those under the authority of the control plane. It shall not be possible for the control plane to modify resources that are under the authority of the management plane. This includes network resources not currently in use, but reserved for future use (e.g., by network planners). As such, resource management is performed by the management plane and is outside the scope of this Recommendation.

6 Transport resources and their organization

The functional architecture of the transport network describes the way that the transport resources are used to perform the basic transport functions in a manner that makes no reference to the control and management of those functions. For the purposes of control and management, each transport resource has a closely coupled agent that represents the role it has to play. These agents interact with other functions that are participating in the control and management through interfaces, and present information or execute operations as required. The transport resources are organized into routing areas and subnetworks for the purposes of control and management.

6.1 Transport entities

For the purpose of managing connections within a layer network, the underlying transport plane resources are represented by a number of entities in the control plane. Figure 6.1 illustrates the relationship between the transport resources described in [ITU-T G.805], the entities that represent these resources for the purposes of network management (as described in [ITU-T M.3100]) and the view of the transport resources as seen by the control plane.

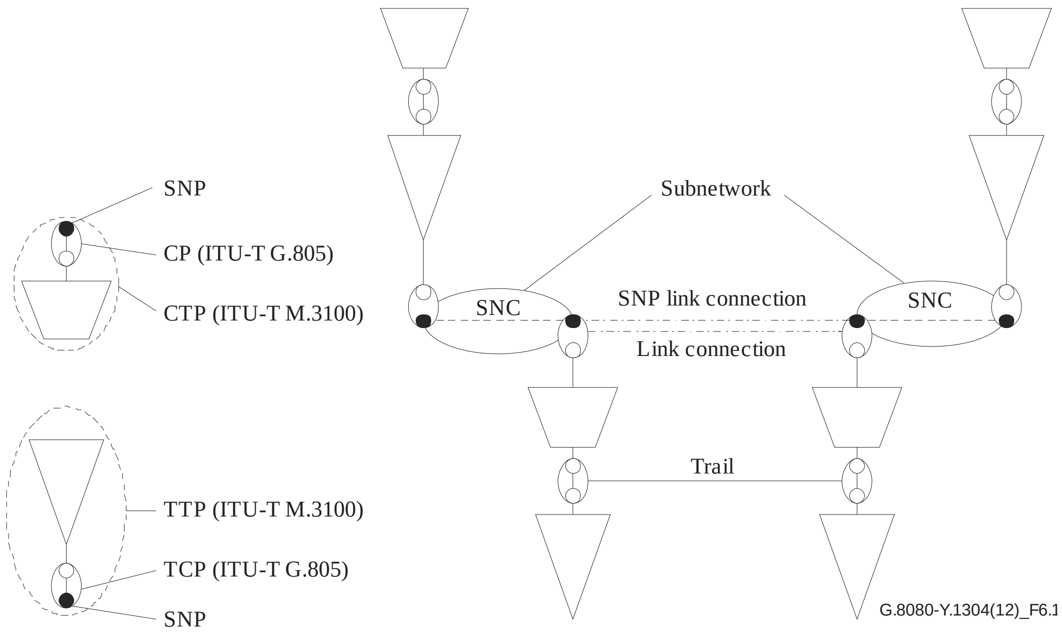


Figure 6.1 – Relationship between architectural entities in the transport plane, management plane and the control plane

An SNP has a number of relationships with other SNPs:

– A static relationship between two SNPs in different subnetworks. This is referred to as an SNP link connection.

– A dynamic relationship between two (or more in the case of broadcast connections) subnetwork points at the boundary of the same subnetwork. This is referred to as a subnetwork connection.

A subnetwork point may also be grouped with other SNPs for the purpose of routing. This is a subnetwork point pool (SNPP) and has a strong relationship with link ends (as defined in [b-ITU-T G.852.2]); however, this relationship is more flexible than the link end. An SNPP may be further subdivided into smaller pools. One use of this sub-structuring is to describe different degrees of route diversity. For example, all the SNPs in one subnetwork that have a relationship to a similar group on another subnetwork may be grouped into a single SNPP. This SNPP may be further sub-divided to represent diverse routes and further subdivided to represent, for example, individual wavelengths.

The association between SNPPs on different subnetworks is an SNPP link.

An SNPP link where each subnetwork is in a different layer is known as a transitional SNPP link. It may also be an SNPP link in which the subnetworks are in different sublayers of the same layer. They only occur across boundaries between layers or sublayers where [ITU-T G.800] transitional links can exist.

The SNP and SNP link connection states of interest to the control plane are described in clause 7.3.8 (Termination and adaptation performers) and clause 7.3.3 (Link resource manager) respectively.

Variable adaptation functions

A number of transport systems support variable adaptation, whereby a single server layer trail may dynamically support different clients. For example, different GFP mappings for packet clients or different multiplexing structures for SDH/OTN. The description below illustrates the application to the latter.

This situation is modelled by assigning SNPs for each CP in the various structures, and placing those SNPs in their respective layer subnetworks. When a particular SNP instance is allocated, this causes the relevant client specific process in the adaptation function to be activated and creates the associated CTP. SNPs in other layer networks that use the same resources become busy.

Figure 6.2 shows an example of an STM-1 trail that can support either a single VC-4 or three VC-3s.

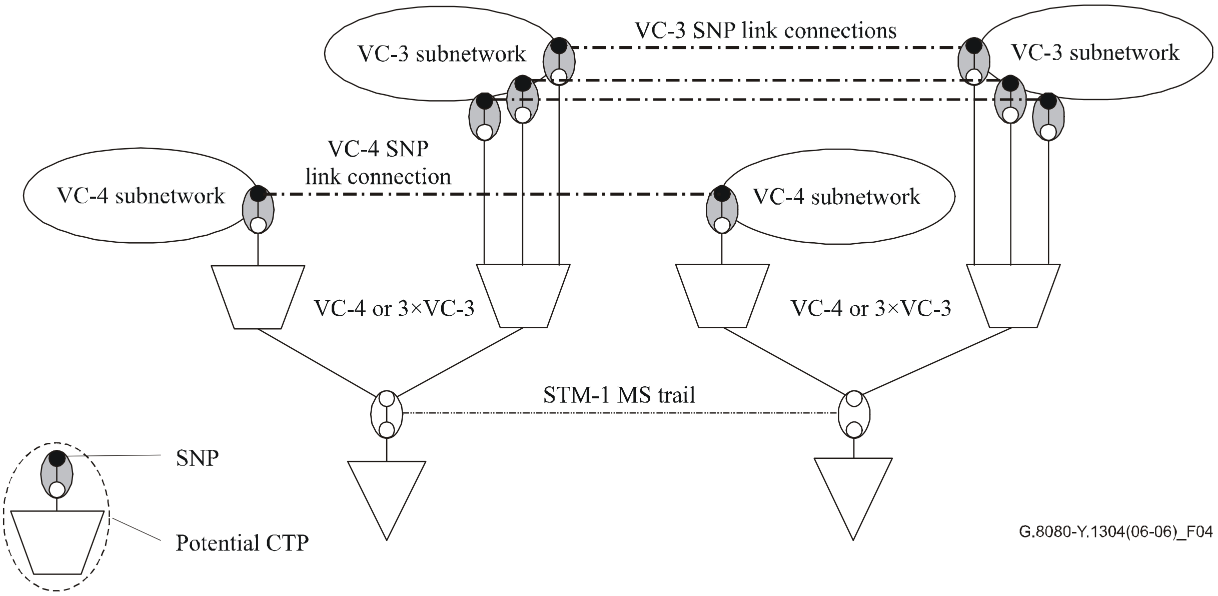


Figure 6.2 – Example of variable adaptation (STM-1 trail supporting both 3 × VC-3 or 1 × VC-4)

Link resources shared between VPNs

[ITU-T Y.1313] defines several basic service models through which layer one VPNs (L1VPNs) may be provided through the ASON architecture.

A VPN is a closed user group that can use a defined set of network resources. In the control plane, an SNPP can be public, that is, not associated with any VPN, or private, that is, associated with exactly one VPN. Connection routing in a VPN can only use the SNPPs associated with that VPN. In the transport plane, a CP can be assigned to an SNP in multiple SNPPs, public or private. Connectivity on a link that is shared between VPNs can be modelled by creating an SNP for each of the shared CPs in each VPN. When a CP is allocated to a particular SNP in one VPN, the SNPs representing the same resources in other VPNs become busy. Figure 6.3 shows an example of two VPNs, each with two SNPs in the control plane. In the transport plane, the first CP is assigned and allocated to the second SNP in VPN 2, the third CP is assigned and allocated to the second SNP in VPN 1, and the second CP is assigned to both the first SNP in VPN 1 and the first SNP in VPN 2. If the second CP is allocated to the first SNP in VPN 1, this SNP becomes available while the first SNP in VPN 2 becomes busy.

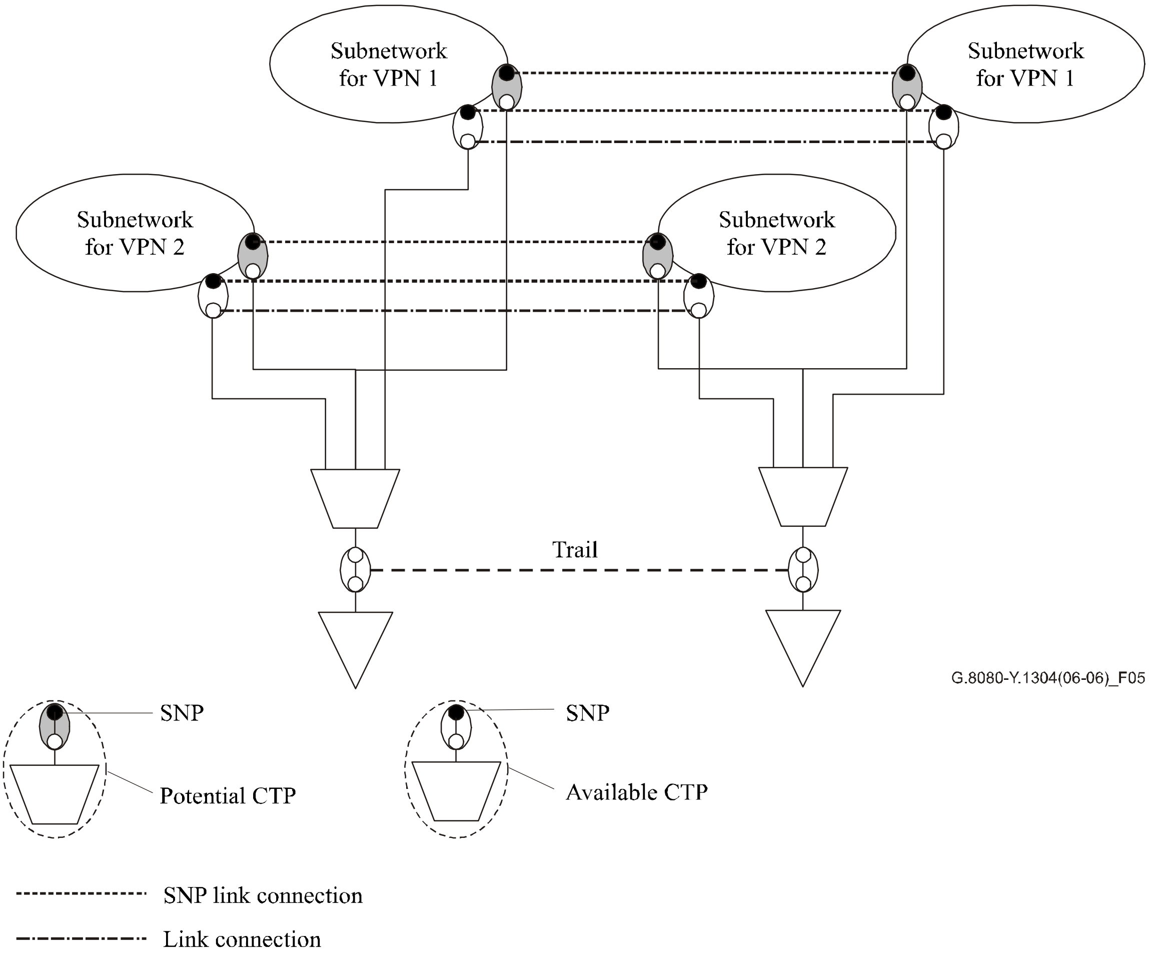


Figure 6.3 – Allocation of link resources between VPNs

In the case of circuit switching, the SNP is bound to a resource label, which provides a resource reservation and allocation. In the case of packet switching, the SNP is not directly bound to a resource label, and the resource label does not carry either any resource allocation. Therefore, in the case of packet switching, when a connection is established, an SNP is selected from a range of SNPs that is bound to a resource label. The connection request should include a resource reservation (CIR and EIR).

6.2 Routing areas

Within the context of this Recommendation, a routing area exists within a single layer network. A routing area is defined by a set of subnetworks, the SNPP links that interconnect them, and the SNPPs representing the ends of the SNPP links exiting that routing area. A routing area may contain smaller routing areas interconnected by SNPP links. The limit of subdivision results in a routing area that contains one subnetwork.

Note that the SNPP links fully contained within the routing area may be transitional links, interconnecting child RAs operating on different CI.

Routing areas and subnetworks are very closely related as both provide an identical function in partitioning a network. The critical distinction is that at the boundary, the link ends are visible from inside a routing area, whereas inside a subnetwork only connection points can be seen. Seen from the outside, subnetworks and RAs are identical, and the terms subnetwork and RA can be used almost synonymously. The distinction between the two is usually obvious from the context, though the term node is often used to denote either a subnetwork or RA. Also note that from the outside of both subnetworks and routing areas, it is not possible to see any internal details, and both subnetworks and routing areas appear as points in the network topology graph.

Where an SNPP link crosses the boundary of a routing area, all the routing areas sharing that common boundary have contained coincident SNPP links. This is illustrated in Figure 6.4.

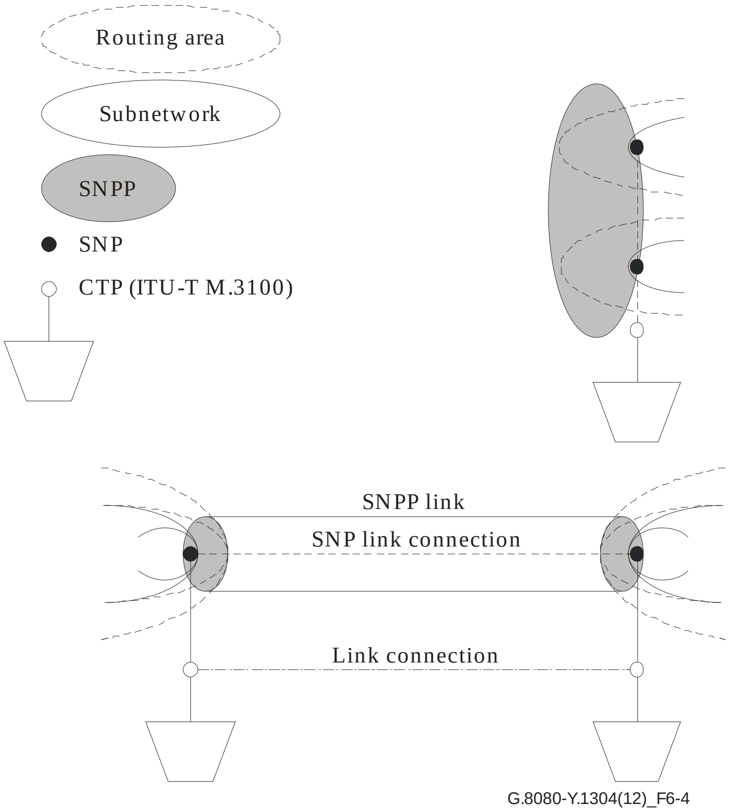


Figure 6.4 – Relationship between routing areas, subnetworks, SNPs and SNPP

6.2.1 Aggregation of links and routing areas

Figure 6.5 illustrates the relationships between routing areas, SNPPs, and SNPP links. Routing areas and SNPP links may be related hierarchically. In the example, routing area A is partitioned to create a lower level of routing areas, B, C, D, E, F, G and interconnecting SNPP links. This recursion can continue as many times as necessary. For example, routing area E is further partitioned to reveal routing areas H and I. In the example given, there is a single top level routing area. In creating a hierarchical routing area structure based upon "containment" (in which the lower level routing areas are completely contained within a single higher level routing area), only a subset of lower level routing areas, and a subset of their SNPP links are on the boundary of the higher level routing area. The internal structure of the lower level is visible to the higher level when viewed from inside of A, but not from outside of A. Consequently only the SNPP links at the boundary between a higher and lower level are visible to the higher level when viewed from outside of A. Hence the outermost SNPP links of B and C and F and G are visible from outside of A but not the internal SNPP links associated with D and E or those between B and D, C and D, C and E or between E and F or E and G. The same visibility applies between E and its subordinates H and I. This visibility of the boundary between levels is recursive. SNPP link hierarchies are therefore only created at the points where higher level routing areas are bounded by SNPP links in lower level routing areas.

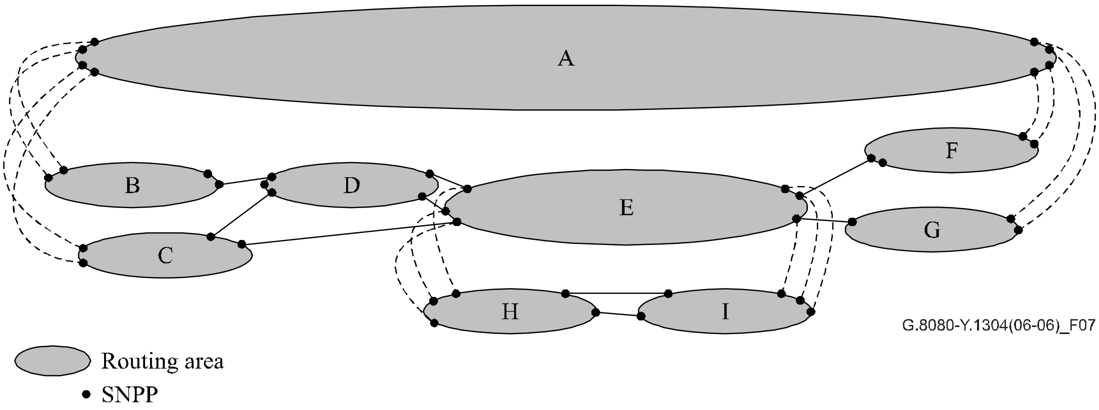


Figure 6.5 – Example of a routing area hierarchy and SNPP link relationships

Subnetwork points are allocated to an SNPP link at the lowest level of the routing hierarchy and can only be allocated to a single subnetwork point pool at that level. At the routing area hierarchy boundaries, the SNPP link pool at a lower level is fully contained by an SNPP link at a higher level. A higher level SNPP link pool may contain one or more lower level SNPP links. In any level of this hierarchy, an SNPP link is associated with only one routing area. As such routing areas do not overlap at any level of the hierarchy. SNPP links within a level of the routing area hierarchy that are not at the boundary of a higher level may be at the boundary with a lower level thereby creating an SNPP link hierarchy from that point (e.g., routing area E). This provides for the creation of a containment hierarchy for SNPP links.

A routing area may have an SNPP name space that is independent from those used in other routing areas. Note, an SNPP name is routable in the RA whose SNPP name space it belongs to.

6.2.2 Relationship to links and link aggregation

A number of SNP link connections within a routing area can be assigned to the same SNPP link if, and only if, they go between the same two subnetworks. This is illustrated in Figure 6.6. Four subnetworks, SNa, SNb, SNc and SNd and SNPP links 1, 2 and 3 are within a single routing area. SNP link connections A and B are in the SNPP link 1. SNP link connections B and C cannot be in the same SNPP link because they do not connect the same two subnetworks. Similar behaviour also applies to the grouping of SNPs between routing areas.

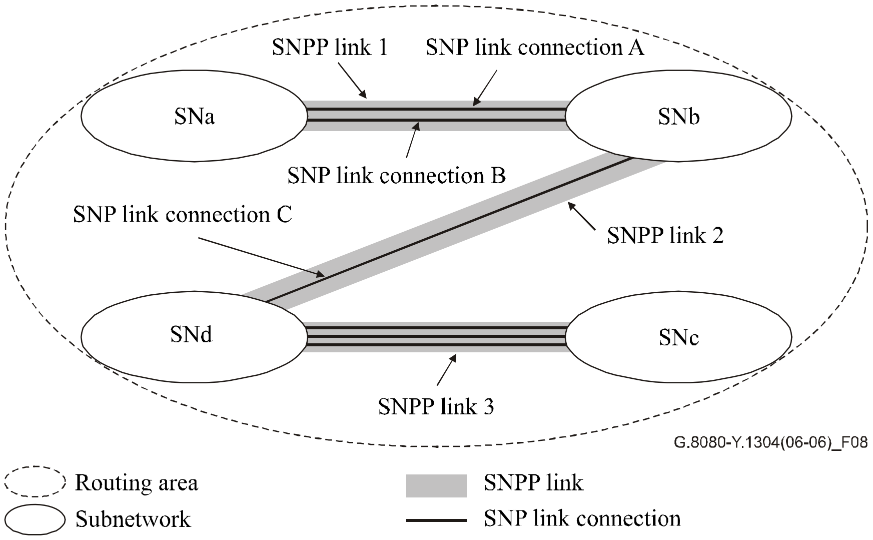


Figure 6.6 – SNPP link relationship to subnetworks

Figure 6.7 shows three routing areas, RA-1, RA-2 and RA-3 and SNPP links 1 and 2. SNP link connections A, B, and C cannot be in the same SNPP link because more than two routing areas are found in their endpoints. SNP link connections A & B are not equivalent to SNP link connection C for routing from Routing Area 3 (RA-3).

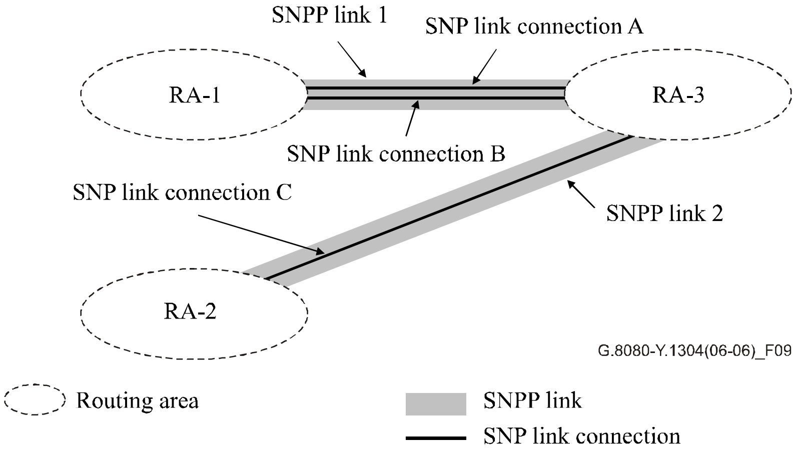


Figure 6.7 – SNPP link relationships to routing areas

SNP link connections between two routing areas, or subnetworks, can be grouped into one or more SNPP links. Grouping into multiple SNPP links may be required:

– if they are not equivalent for routing purposes with respect to the routing areas they are attached to, or to the containing routing area;

– if smaller groupings are required for administrative purposes.

There may be more than one routing scope to consider when organizing SNP link connections into SNPP links. In Figure 6.8, there are two SNP link connections between routing areas 1 and 3. If those two routing areas are at the top of the routing hierarchy (there is therefore no single top level routing area), then the routing scope of RA-1 and RA-3 is used to determine if the SNP link connections are equivalent for the purpose of routing.

The situation may however be as shown in Figure 6.8. Here RA-0 is a containing routing area. From RA-0's point of view, SNP link connections A & B could be in one (case a) or two (case b) SNPP links. An example of when one SNPP link suffices is if the routing paradigm for RA-0 is step-by-step. Path computation sees no distinction between SNP link connections A and B as a next step to get from say RA-1 to RA-2.

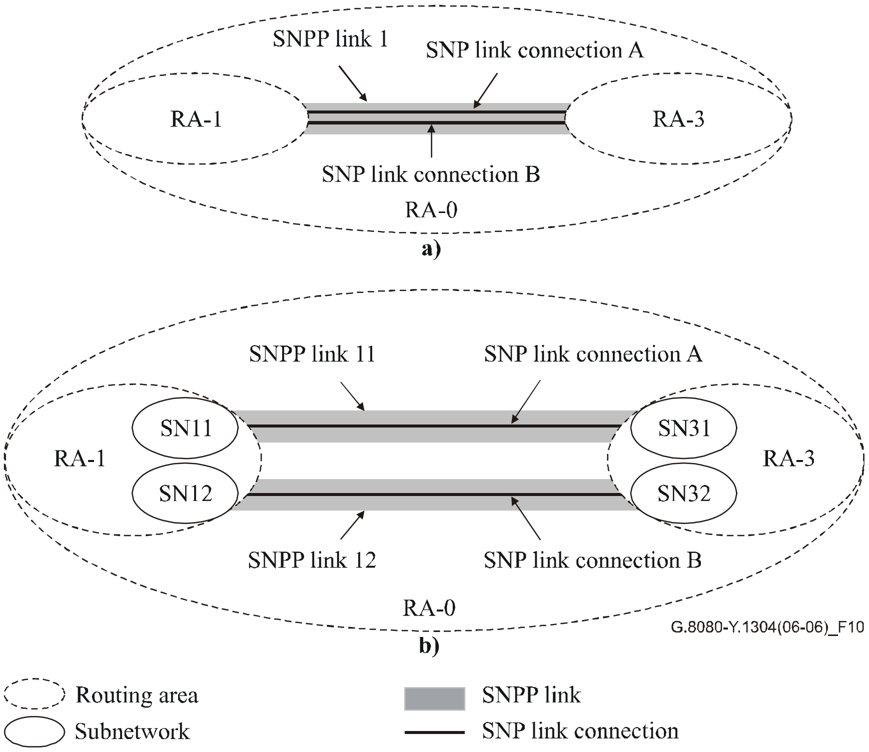


Figure 6.8 – Routing scope

From RA-1 and RA-3's point of view though, the SNP link connections may be quite distinct from a routing point of view as choosing SNP link connection A may be more desirable than SNP link connection B for cost, protection or another reason. In this case, placing each SNP link connection into its own SNPP link meets the requirement of "equivalent for the purpose of routing". Note that in Figure 6.8, SNPP link 11, link 12 and link 1 can all coexist.

Another reason for choosing SNPP link 11 (Figure 6.8-b) over SNPP link 12 could be because the cost of crossing RA-3 is different from SNPP link 11 than from SNPP link 12. This suggests that a mechanism to determine the relative cost of crossing RA-3 from link 11 and from link 12 would be useful. Such a mechanism could be used recursively to determine the relative cost of crossing RA-0. Note that this does not imply exposing the internal topology of any routing area outside of its scope. A query function could be invoked to return the cost of a particular route choice. The costs returned by such a query would be determined by policy applied to each routing area. A common policy should be used in all the routing areas, resulting in comparable costs. Such a query could also be generalized to apply routing constraints before calculating the cost.

Routing areas in different layers may be connected by transitional SNPP links. This enables multi-layer routing topology construction. Routing areas in different sublayers of the same layer may also be connected by transitional SNPP links.

6.3 Topology and discovery

The routing function understands topology in terms of SNPP links. Before SNPP links can be created, the underlying transport topology, i.e., the trail relationship between the access points, must be established. These relationships may be discovered (or confirmed against a network plan) using a number of different techniques; for example, use of a test signal or derived from a trail trace in the server layer. They may also be provided by a management system based on a network plan. The capability of the transport equipment to support flexible adaptation functions (and thus link connections for multiple client layer networks) may also be discovered or reported.

Link connections that are equivalent for routing purposes are then grouped into links. This grouping is based on parameters, such as link cost, delay, quality or diversity. Some of these parameters may be derived from the server layer but in general they will be provisioned by the management plane.

Separate links may be created (i.e., link connections that are equivalent for routing purposes may be placed in different links) to allow the division of resources between different ASON networks (e.g., different VPNs) or between resources controlled by ASON and the management plane.

The link information (e.g., the constituent link connections or resource label range with the available link bandwidth) is then used to configure the LRM instances (as described in clause 7.3.3) associated with the SNPP link. Additional characteristics of the link, based on parameters of the (potential) link connections, may also be provided. The LRMs at each end of the link must establish a control plane adjacency that corresponds to the SNPP link. The interface SNPP ids may be negotiated during adjacency discovery or may be provided as part of the LRM configuration. The link connections and CP names or resource labels (and link connections) are then mapped to interface SNP ids (and SNP link connection names). In the case where both ends of the link are within the same routing area the local and interface SNPP id and the local and interface SNP ids may be identical. Otherwise, at each end of the link the interface SNPP id is mapped to a local SNPP id and the interface SNP ids are mapped to local SNP ids. This is shown in Figure 6.9.

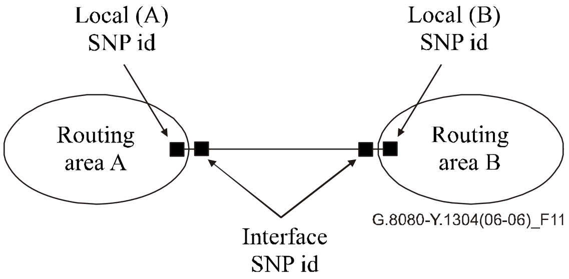


Figure 6.9 – Relationship between local and interface ids

The resulting SNP link connections may then be validated by a discovery process. The degree of validation required at this stage is dependent on the integrity of the link connection relationships initially provided by the transport plane or management plane and the integrity of the process used to map CPs to SNPs.

Validation may be derived from a trail trace in the server layer or by using a test signal and test connections. If test connections are used, the discovery process may set up and release these connections using either the management plane or the control plane. If the control plane is used, the link must be made temporarily available to routing and connection control, for test connections only.

Once the SNPP link validation is completed, the LRMs inform the RC component (see clause 7.3.2) of the SNPP link adjacency and the link characteristics, e.g., cost, performance, quality diversity, and bandwidth.

6.4 Domains

As introduced in clause 5, we have generalized the domain notion embodied in the ITU-T G.805 definition of administrative and management domains, along with the notion of Internet administrative regions, to express differing administrative and/or managerial responsibilities, trust relationships, addressing schemes infrastructure capabilities, survivability techniques, distributions of control functionality, etc. A domain thus represents a collection of entities that are grouped for a particular purpose.

A control domain is comprised of a collection of control plane components, and provides an architectural construct that encapsulates and hides the detail of a distributed implementation of a particular group of architectural component of one or more types. It allows for the description of a group of distributed components in such a way that the group can be represented by distribution interfaces on a single entity, the domain, that has identical characteristics to that of the interfaces of the original component distribution interfaces. The nature of the information exchanged between control domains captures the common semantics of the information exchanged between component distribution interfaces, while allowing for different representations inside the domain.

Generally a control domain is derived from a particular component type, or types, that interact for a particular purpose. For example, routing (control) domains are derived from routing controller components whilst a re-routing domain is derived from a set of connection controller and network call controller components that share responsibility for the re-routing/restoration of connections/calls that traverse that domain. In both examples the operation that occurs, routing or re-routing, is contained entirely within the domain. In this Recommendation, control domains are described in relation to components associated with a layer network.

As a domain is defined in terms of a purpose, it is evident that domains defined for one purpose need not coincide with domains defined for another purpose. Domains of the same type are restricted in that they may:

- fully contain other domains of the same type, but do not overlap
- border each other
- be isolated from each other.

An example of the relationships between components, domains and reference points is provided in Figure 6.10 which shows a domain, B, and its relationship to domains A, C and D. Each domain is derived from a component of type Z. The internal structure and interactions may be different in each domain, e.g., they may use different federation models.

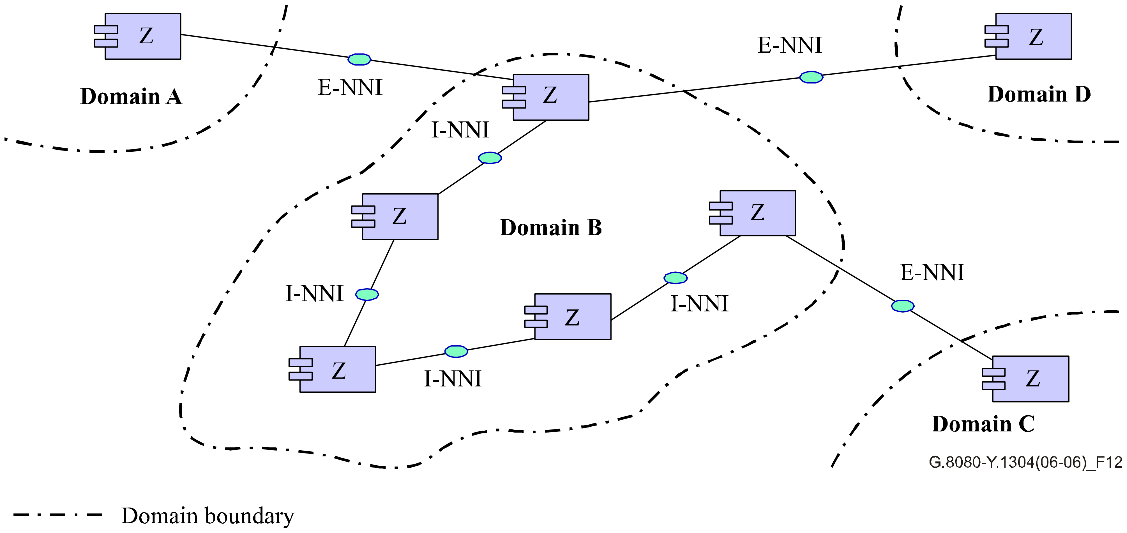


Figure 6.10 – Relationship between domains, protocol controllers and reference points

The same example is shown in Figure 6.11 with the relationships between components, domains and interfaces. The components interact via their protocol controllers, using protocol I on the I-PCs and protocol E on the E-PCs. It is also possible for the protocol used internal to A, for example, to be different to that used in B, and the protocol used between B and C to be different to that between A and B. The I-NNI interfaces are located between protocol controllers within domains whilst E-NNI interfaces are located on protocol controllers between domains.

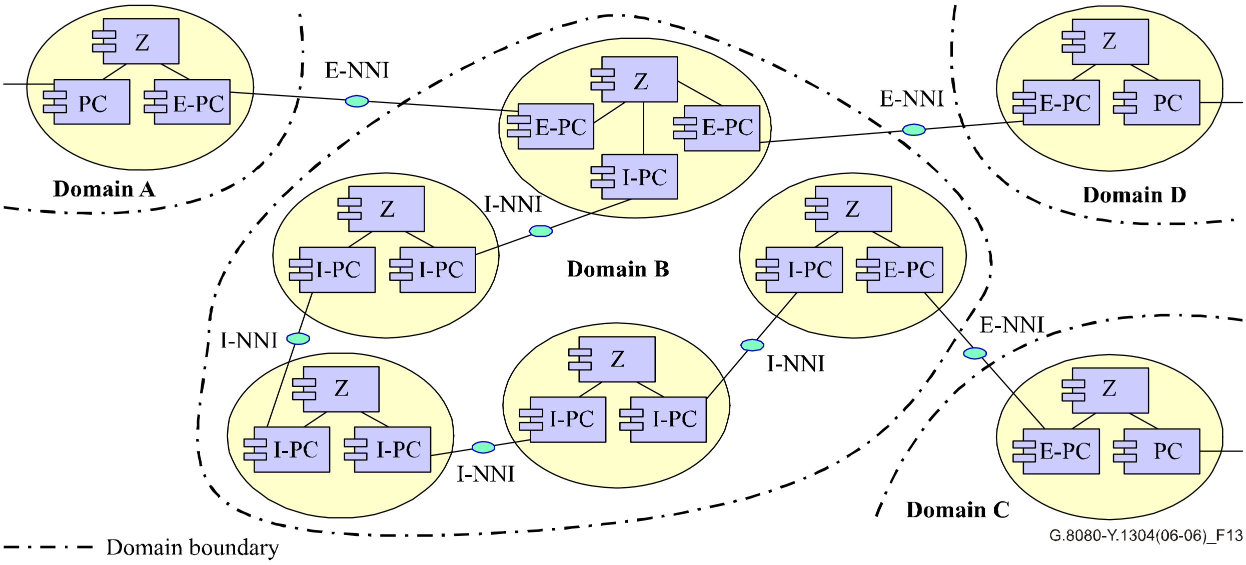


Figure 6.11 – Relationship between domains, protocol controllers and interfaces

6.4.1 Relationship between control domains and control plane resources

The components of a domain may, depending on purpose, reflect the underlying transport network resources. A routing control domain may, for example, contain components that represent one or more routing areas at one or more levels of aggregation, depending upon the routing method/protocol used throughout the domain.

6.4.2 Relationship between control domains, interfaces and reference points

I-NNI and E-NNI interfaces are always between protocol controllers. The protocols running between protocol controllers may or may not use SNPP links in the transport network under control and as such it is incorrect to show I-NNI and E-NNI interfaces on SNPP links.

I-NNI and E-NNI reference points are between components of the same type, where the component type is not a protocol controller, and represents primitive message flows (see clause 7).

In a diagram showing only domains and the relationships between them (and not revealing the internal structure of the domains), the information transfer is assumed to be over a reference point.

6.5 Multi-layer aspects

The description of the control plane can be divided into those aspects related to a single layer network, such as routing, creation and deletion of connections, etc., and those that relate to multiple layers. The relationship between the client and server layer networks is managed by means of the termination and adaptation performers (see clause 7.3.8). The topology and connectivity of the underlying server layer is not explicitly visible to the client layer, rather aspects of the server layer are encapsulated and presented to the client layer network. The server layer may be represented as a client layer SNPP link, or as client layer subnetworks interconnected by SNPP links. This abstraction of the server layer topology for the purpose of path computation in the client layer has similar properties to the hierarchical containment relationships described in clause 6.2.1. Thus the results of path computation in the client layer may not result in the optimum use of the server layer resources. If the resources available to the client layer network are insufficient to support a connection request, additional resources may be provided by activating or creating new connections in one or more server layer networks. Operator policies will govern the availability of underlying server layer resources to the client layer.

When the server layer is represented to the client layer as an SNPP link, this can be achieved by modifying SNPs from potential to available. If the server layer is represented as a set of SNPP links and subnetworks, then a new server layer connection must be requested directly by the control plane depending on operator policies. In the case where the capacity of the resulting server layer network connection is greater than the capacity required to support the client layer connection, a new SNPP link will be created as a side effect of this operation. The mechanism to add this new SNPP link to the topology of the client layer is not within the scope of this Recommendation.

Alternatively, the server layer topology may not be made visible to the client layer. In this case the decision to add new server layer network connections may be made by a planning process. The use of a planning system to invoke the configuration of the server layer network to provide additional resources for the client layer network may result in more optimum use of the server layer resources. Details of the planning process and its interaction with the client and the server control planes is outside the scope of this Recommendation. The decision to create/remove the server layer connections is a business decision that may be represented in operator policy.

As described above, there are two different ways to represent server layer resources in the client layer. The representation used is dependent on whether server layer resources have already been selected to provide connectivity to the client layer. A control plane instance for a layer network may use one or both of these approaches at a time. Other representations of server layer resources are for further study.

6.5.1 Representation as SNP link connections

A planning process may cause a pair of access points in the server layer to be connected, this creates a client layer SNPP link that contains multiple SNP link connections for the supported client layers. These SNP link connections may initially be either active or potential, as the underlying resource may or may not be allocated for the exclusive use of the client layer (see clause 6.1).

This process may be applied recursively in multiple client/server layer networks. Thus, the connection between the pair of access points may be supported by potential link connections. The connection of the access points in the server layer is not activated until a request to use one of the potential client layer SNP link connections is received by the control plane. At this time, the client layer request is "suspended" while the server layer connections are established. Once the server layer connections are in place, the client layer SNP link connections are made active, allowing the client layer signalling to resume. If a connection in the server layer cannot be established, the client layer SNP link connections cannot be made active, causing the client layer connection attempt to fail due to lack of resources.

When the SNP link connection is removed, the underlying resources that supported the connection are freed, allowing them to return to the potential state if allowed by policy. When all SNP link connections supported by a server trail return to the potential state, the server trail may be changed to potential if allowed by policy.

It should be noted that the SNPP link containing these SNP link connections can be represented using existing components and states as described in clause 6.1.

6.5.2 Representation as a set of SNPP links and subnetworks

A planning process may also allow the client layer control plane more flexibility in the selection of the server layer resources to satisfy a connection request. To accomplish this, some of the server layer resources are represented to the client layer as SNPP links and subnetworks with the appropriate client layer control plane components. This representation is possible if the server layer provides flexibility which allows client layer subnetworks to be interconnected. The server layer flexibility and its corresponding client layer flexibility are shown in Figure 6.12. This enables the routing controller to know that a set of client layer SNPs can be reachable through a common server layer. That is, in Figure 6.12, absence of the server layer representation would not allow paths to be computed between all of the client layer SNPs.

The adaptation and termination functions used to transition from the client layer to the server layer are represented as SNPP links in the client layer. In this figure, we used dashed elements to distinguish this representation of the server layer resources to the client layer from other client layer resources. Additionally, client and server layer SNPs are represented as small solid-line white circles, respectively; however, to the client layer routing controller, there is no distinction between the represented elements and the other client layer elements.

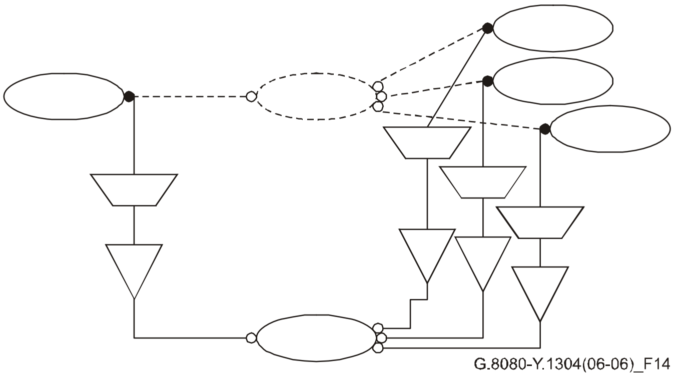


Figure 6.12 – Relationship between client and server architectural elements

As in the SNPP link representation, the SNP link connections connecting the client layer subnetwork to the server layer are potential, as the server layer resources have not been allocated for exclusive use to the client layer. When a connection request is received that utilizes a potential resource, a path through the server layer is calculated identifying the specific resources to be used. As a result of identifying these resources, client layer SNP link connections are created, and processing continues as above.

6.5.3 Multilayer routing topology

In addition to the single layer topology representation in clause 6.5.2, a routing topology representative of a multi-layer network may be constructed using transitional SNPP links that connect routing areas in different layers. Paths may be determined that traverse link and subnetwork connections in more than one layer and/or sublayer. Control plane components that are involved in configuring the resulting connection configure link connections, subnetwork connections, and transitional link connections.

The use of a transitional SNPP link implies that a sequence of adaptations between layers, or sequence of layer processors within a layer, is used. The transitional SNPP link enables a connected graph to be constructed that represents a multi-layer network for the purpose of routing.

Figure 6.13 illustrates the [ITU-T G.800] representation and the corresponding multilayer routing topologies.

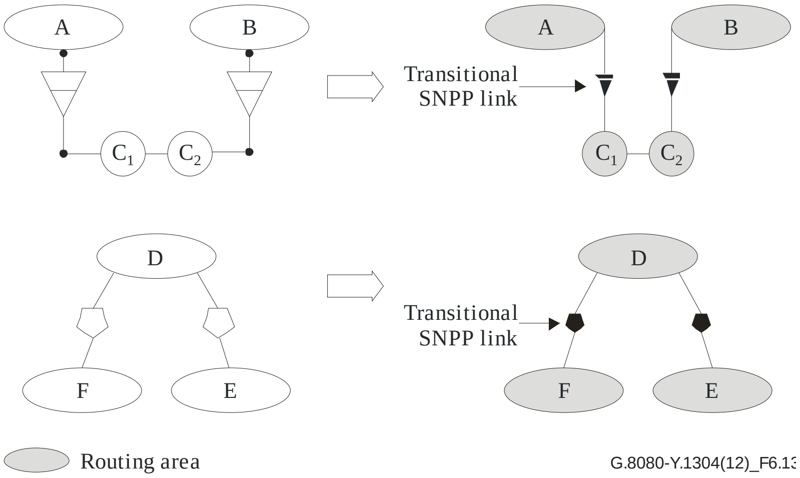


Figure 6.13 – Multilayer routing topology representations

A multilayer topology corresponding to the model in Figure 6.12 is illustrated in Figure 6.14 on the left. It differs from the model in Figure 6.12 in that the topology is multilayer (i.e., contains both client and server SNPPs).

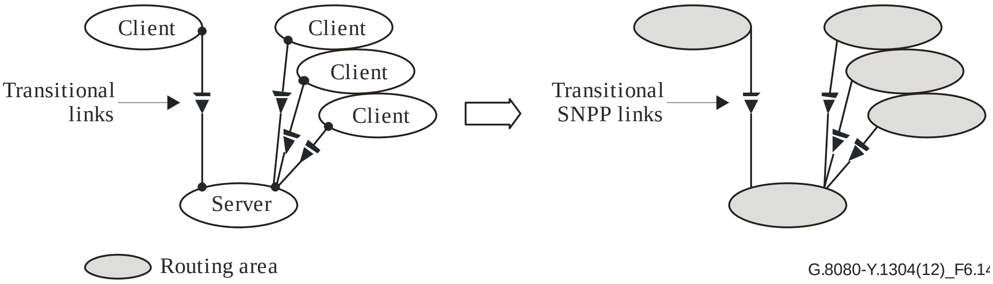


Figure 6.14 – Multilayer routing topology representation of Figure 6.12

A path computation begins by selecting two (sets of) points at the boundary of a routing area. The path computation may be accomplished using a set of subordinate path computations involving contained routing areas. The routing areas relevant to the path computation are selected by the path computation algorithm, and these routing areas may belong to one or more layers (i.e., they may be part of a multi-layer topology).

In a multilayer routing topology, when a transitional SNPP link is used between a routing area in a first layer to another routing area in an adjacent second layer and the first layer is traversed further in a route, it is expected that a corresponding transitional link is used to eventually return to the first layer in computed paths.

For segments of a path computed between sublayers, traversal of a single transitional SNPP link is allowed.

An additional example of a multilayer topology is shown in Appendix V.

NOTE – Use of transitional links for a 1:1 adaptation is described in this section. Usage for 1:n and m:1 is for further study.

6.6 Interlayer client support

In transport networks, network elements may support more than a single layer. For example at the edge of a multi-layer transport network, a network element may support client layer networks that are directly supported in the core of a multilayer transport network. For example, the edge network element may support the adaptation of service into a lower order VC whilst the core network elements may only provide flexibility at the high order VC layer; or the edge network element may adapt an Ethernet service into high order VCs. A general problem faced is how to transfer client characteristic information (CI) when a continuous/connected client layer network is not present between two client AGCs.

There are two solutions to this problem. Either, client layer topology may be created from server layer connections as described in clause 6.5 or, the client CI could be adapted, possibly multiple times, onto server layer connections. This would not be visible to the client routing controller.

Interfaces between network call controllers (NCCs, see clause 7.3.5.2) in different layer networks are used to apply ASON functions to the second solution. This interlayer interface enables an association between calls in a client/server layer relationship. This association can recurse to mirror a set of "stacked" adaptations. That is, the NCCs recurse with ITU-T G.805 layers. NCCs at different layers may still be instantiated differently from each other. For example an NCC could be distributed at a client layer and centralized at a server layer. A server layer CC creates the connection(s). The client CI is mapped to the server layer connection and this association is maintained by the client/server NCC relationship. In this situation, a client layer link connection is created as a result of the server layer connection and CI mapping, but the client layer CC is not involved in this. This recurses upward and creates a link connection at each of the affected client layers.

Appendix II illustrates this capability with an example.

The interlayer NCC relationship may occur at points other than where access group containers are attached to the client layer network. In Figure 6.12, a call traverses a client subnetwork first before being supported by a server layer subnetwork. In Figure 6.15, the call may also be supported by connections in a client layer subnetwork that is not contiguous with client layer subnetworks at the ingress or egress. Here, interlayer NCCs relationships are found between client subnetwork C2 and server subnetworks S1 and S2.

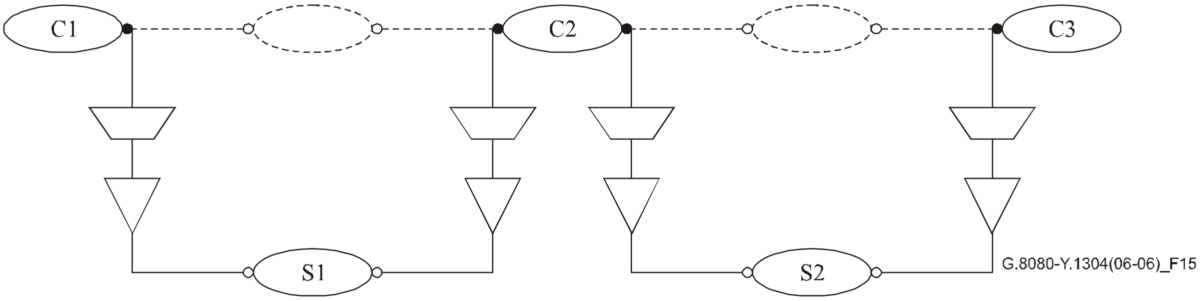


Figure 6.15 – Non-contiguous client layer subnetworks

6.7 Calls supported by calls at same layer

Similar to the arrangement of NCCs in an interlayer relationship, a call segment may be supported by a separate call at the same layer but not over an E-NNI. In this arrangement, an NCC to NCC call segment is supported by a complete call with calling/called party call controllers. Figure 6.16 illustrates this in a carrier's carrier business scenario. Here, a call between two clients associated with Carrier A is supported in two subnetworks that belong to Carrier A. Between those two subnetworks, the call is supported by a separate SPC within Carrier B. The connection returned by Carrier B is joined with the connections established in the Carrier A subnetworks.

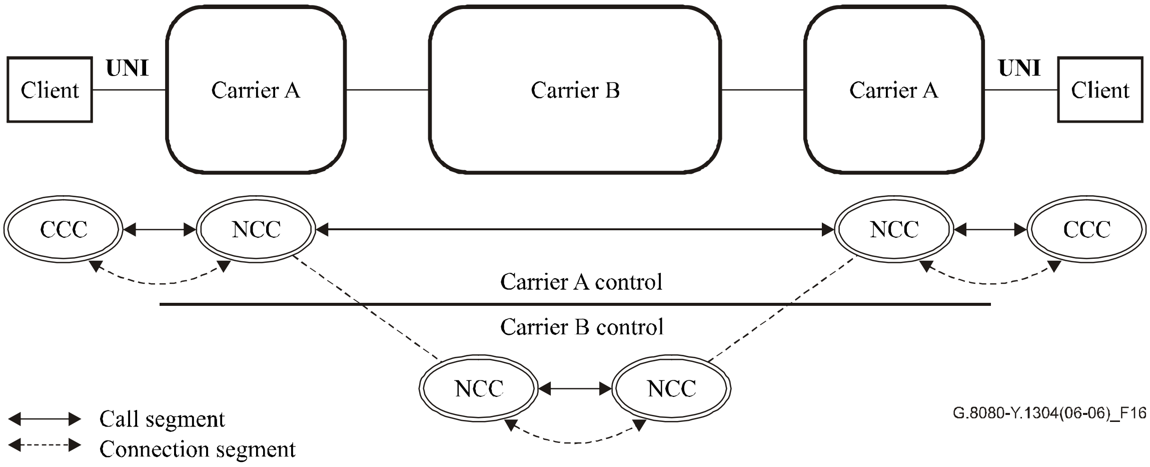


Figure 6.16 – Calls supported by calls in the same layer

6.8 Mapped server interlayer relationships

When a client layer CI is mapped to a server layer as described in clauses 6.5 and 6.6, several types of arrangements can exist. They include client/servers in 1:1 and 1:n relationships. The ratio refers to the number of connection points in each layer.

6.8.1 1:1 relationship

In the 1:1 relationship support for a client communication is supported by the server layer as a single trail. The NCC_{client}:NCC_{server} relationship is also 1:1. A specific example of this is shown in Appendix II where Ethernet CI is mapped to a single VC-3. Another example is a DS-3 into an STS-1.

As the NCC_{client} is trying to use the server layer, it must have knowledge of the relevant call parameters of the NCC_{server} including whether the client initiated the server layer call (server NCC coordination out interface) or whether the server layer already existed for client layer use (client NCC coordination out interface).

The NCC_{server} is not required to possess knowledge of the client layer call parameters, but should inform the NCC_{client} if there are changes in the server layer call.

6.8.2 1:n relationship

In a 1:n relationship, the client communication is supported by multiple connections in the server layer. This is supported either by the server NCC supporting multiple connections or multiple NCCs in the server layer supporting the one client layer NCC.

The example below in Figure 6.17 illustrates the latter case. The Ethernet call is mapped to a VC-4-2v VCAT call. The VCAT call is related to multiple server layer VC-4 calls. The Ethernet/GFP layer to VCAT relationship is 1:1.

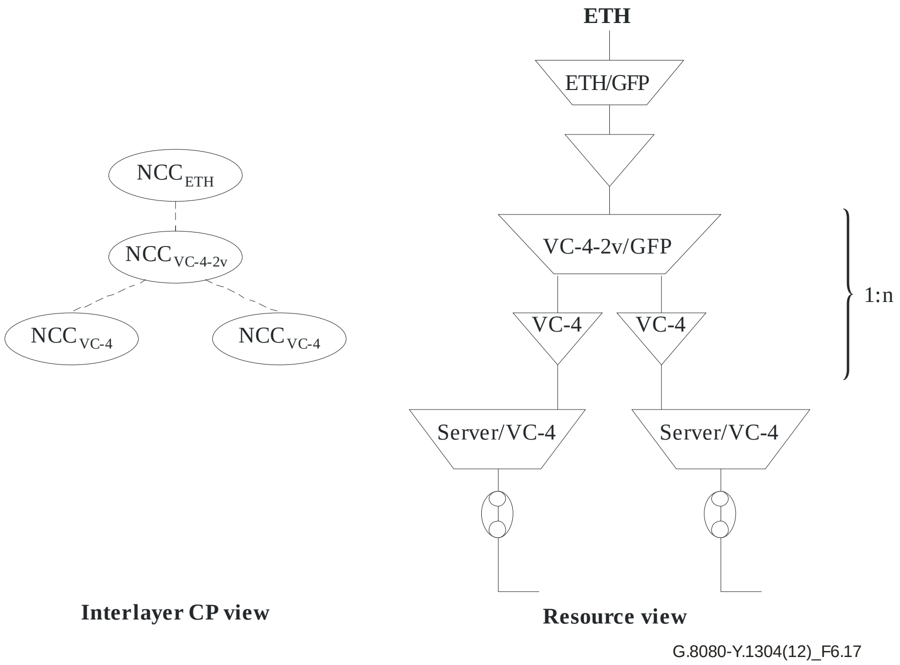


Figure 6.17 – Example 1:n mapped server

It is necessary for the client layer NCC (the VC-4-2v layer) to know about the call parameters of the server layer as it must ensure that there are a sufficient number of server layer calls as well as what their aggregate characteristics are. The server layer is not required to possess knowledge of the client layer call parameters, but should inform the client call if there are changes in the server layer call.

7 Control plane architecture

This clause describes a reference architecture for the control plane that supports the requirements in this Recommendation, identifying its key functional components and how they interact. This flexible reference architecture is intended to enable operators to support their internal business and managerial practices, as well as to bill for service usage. The control plane architecture should have the following characteristics:

– Support various transport infrastructures, such as those covered by [ITU-T G.805].

– Be applicable regardless of the particular choice of control protocol (i.e., employ a protocol neutral approach that is independent of the particular connection control protocols used).

– Be applicable regardless of how the control plane has been subdivided into domains and routing areas, and how the transport resources have been partitioned into subnetworks.

– Be applicable regardless of the implementation of connection control that may range from a fully distributed to a centralized control architecture.

This reference architecture describes the:

– functional components of the control plane, including abstract interfaces and primitives

– interactions between call controller components

– interactions among components during connection set-up

– functional component that transforms the abstract component interfaces into protocols on external interfaces.

Special components are defined in this Recommendation and are provided to allow for implementation flexibility. These components are protocol controllers and port controllers. The detail of the interfaces of these and other components are provided in other technology specific Recommendations.

Protocol controllers are provided to take the primitive interface supplied by one or more architectural components, and multiplex those interfaces into a single instance of a protocol. This is described in clause 7.4 and illustrated in Figure 7.28. In this way, a protocol controller absorbs variations among various protocol choices, and the architecture remains invariant. One or more protocol controllers are responsible for managing the information flows across a reference point.

Port controllers are provided to apply rules to system interfaces. Their purpose is to provide a secure environment for the architectural components to execute in, thereby isolating the architectural components from security considerations. In particular, they isolate the architecture from distribution decisions made involving security issues. This is described in clause 7.2.1 and Figure 7.2.

7.1 Notation

In this clause, we consider the component architectural notation based upon some simple building blocks from the vocabulary of the unified modelling language (UML).

Interface: An interface supports a collection of operations that specify a service of a component, and is specified independently from the components that use or provide that service. Operations specify the information passed in or out together with any applicable constraints. Interface definitions are presented in the form of a table, an example of which is presented in Table 1. Each interface has an interface name that identifies the role. Input interfaces represent services provided by the component; the basic input parameters are required for the specific role and basic return parameters are a result of the action on the input parameters. Output interfaces represent services used by the component; the basic output parameters define the information provided, the basic return parameters (if identified) are those required in response to the output parameters. Notification interfaces represent unsolicited output actions by the component, and are represented by an output interface with no return parameters. These three interface types are described separately in interface specifications.

Table 1 – Generic interface descriptions table format

Input interface	Basic input parameters	Basic return parameters
Interface name	Input parameters	Returned parameters

Output interface	Basic output parameters	Basic return parameters
Interface name	Output parameters	Returned parameters

Transaction semantics associated with a particular transaction are assumed to be handled transparently, and there is no need to explicitly mention separate parameters for this purpose in interface description.

Role: A role is the behaviour of an entity when it is participating in a particular context. Roles allow for the possibility that different entities participate at different times, and are denoted by annotating a relationship with the name of an interface.

Component: In this Recommendation, components are used to represent abstract entities, rather than instances of implementation code. They are used to construct scenarios to explain the operation of the architecture. This component is represented as a rectangle with tabs. This is illustrated in Figure 7.1.

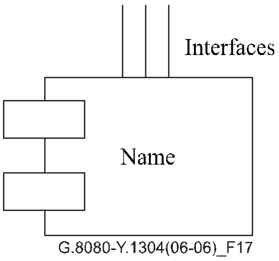


Figure 7.1 – Representation of a component

Generically, every component has a set of special interfaces to allow for monitoring of the component operation, and dynamically setting policies and affecting internal behaviour. These interfaces are not mandatory, and are provided on specific components only where necessary. Where appropriate, the use of the monitor interface is described in individual component descriptions. Components are not assumed to be statically distributed.

When interfaces on components are described, only the different interface types are specified. All components have the property of supporting multiple callers and multiple providers, and resolution of concurrent requests is not mentioned explicitly.

As components are used in an abstract way, this specification is extendable by the techniques of component sub-classing and composition.

7.2 Policy and federations

7.2.1 General model of policy

For the purposes of this policy model, systems represent collections of components, and a system boundary provides a point where policy may be applied. Policy is defined as the set of rules applied to interfaces at the system boundary, and implemented by port controller components. Policy ports are used to simplify the modelling of policies that are applied to multiple ports. System boundaries are nested to allow for correct modelling of shared policies applied to any scope (full system, any set of components, individual components, etc.). Note that the order of policy application is that which is specified by the nesting.

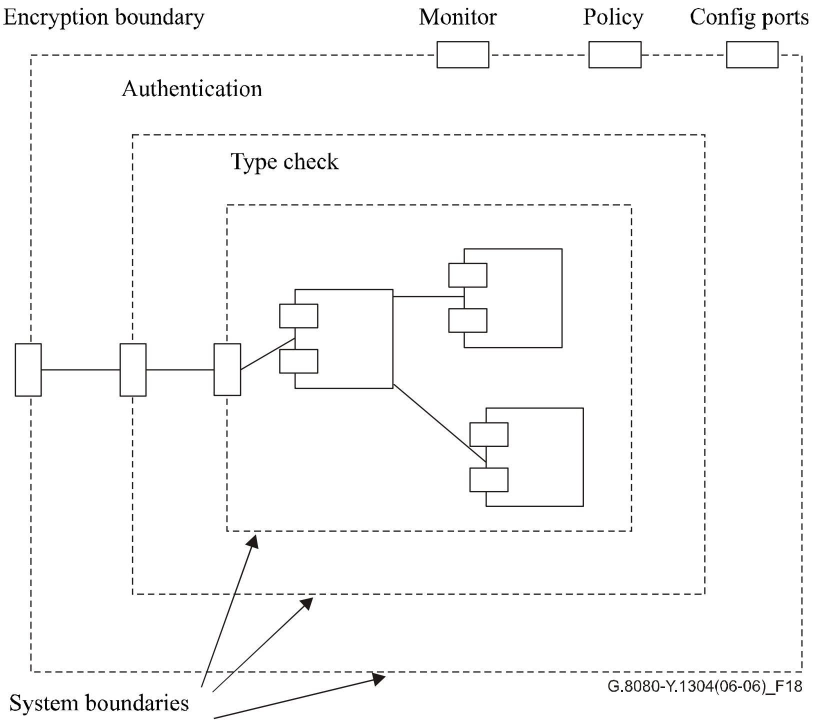


Figure 7.2 – Example of system boundaries related to policy control

In Figure 7.2, the dashed boxes represent system boundaries, while the closed rectangles on the boundary, called ports, represent port controller components.

The monitor, policy, and configuration ports may be available on every system (and component) without further architectural specification. The monitor port allows management information to pass through the boundary relating to performance degradations, trouble events, failures, etc., for components, subject to policy constraints. The policy port allows for the exchange of policy information relating to components. The configuration port allows for the exchange of configuration, provisioning and administration information relating to components (subject to policy constraints) that may dynamically adjust the internal behaviour of the system.

Figure 7.2 shows an example of how encryption, authentication and type checking may be implemented as a set of three nested port controllers, where the policy application order follows the nesting order. The components inside the authentication boundary do not specify encryption or authentication requirements, as these are properties of the component environment. Port controllers are defined for each independent aspect of port policy, and combined policy is achieved by composition of port controllers. This allows the creation of reusable components, which are distinguished by a descriptive prefix. Policy violations are reported via the monitoring port.

The policy port may be seen as a filter of incoming messages, where messages that are rejected have violated the policy. Policies may be dynamically changed via the system policy port, and in this way, dynamic behavioural changes may be described.

It is common to discuss how policy is applied at a reference point, but policy can only be applied to the individual interfaces crossing the reference point. A method of combining several interfaces into a single implementation interface is described later in clause 7.4, protocol controllers.

Other aspects of policy relate to variable behaviour of the components (such as schedules, access rights, etc.) and these aspects are specified and implemented by the components. Component behaviour may also be dynamically changed, and the ability to do this may be controlled by policy. This allows us to determine which aspects of system behaviour are specified where.

Policy, as other aspects of the system, may be distributed. An example of a suitable model for distribution could be the COPS protocol model of [b-IETF RFC 2753]. The policy enforcement point (PEP) (the point where the policy decisions are enforced) of that model corresponds to the port in this model. The policy decision point (PDP) is the point where policy decisions are made. This can be done within the port, though it may be distributed to a different system. This distribution decision depends on many factors that in turn depend on the actual policy. As an example, performance reasons may force the PDP to be within the port (encryption), while security reasons may force the PDP to be elsewhere (password lookup).

When the PEP and PDP are not collocated, cooperation is required.

7.2.2 General model of federation

The creation, maintenance, and deletion of connections across multiple domains is required. This is achieved by cooperation between controllers in different domains. For the purposes of this Recommendation, a federation is considered a community of domains that cooperate for the purposes of connection management, and is illustrated using the cooperation between connection controllers. (Connection controllers are described in clause 7.3.1.)

There are two types of federation:

– joint federation model

– cooperative model.

In the joint federation case one connection controller, the parent connection controller, has authority over connection controllers that reside in different domains. Where a connection is required that crosses multiple domains the highest-level connection controller (the parent) acts as the coordinator. This connection controller has knowledge of the highest-level connection controllers in each domain. The parent connection controller divides the responsibility for the network connection between the next level connection controllers, with each responsible for its part of the connection. This is illustrated in Figure 7.3. This model is recursive with a parent connection controller at one level being a child to a parent at a higher level.

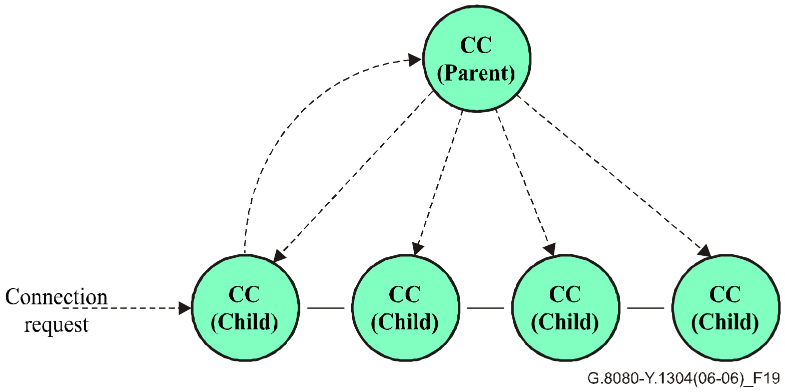


Figure 7.3 – Joint federation model

In the cooperative model, there is no concept of a parent connection controller. Instead, when a connection request is made, the originating connection controller contacts each of the connection controllers associated with domains of its own volition and there is no overall coordination. The simplest method of achieving this is for the originating connection controller to contact the next connection controller in the chain. This is illustrated in Figure 7.4, where each connection controller calculates what part of the connection it can provide and what the next connection controller will be. This continues until the connection is provided.

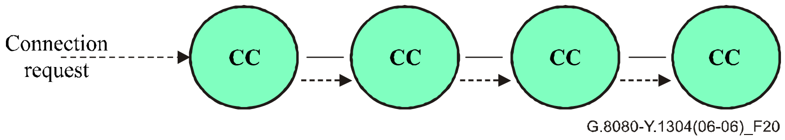


Figure 7.4 – Cooperative federation model

Federation between administrative domains is by means of the cooperative model. In this case, all administrative domains are expected to have the capability to federate with other administrative domains. Parent connection controllers within an administrative domain may federate with other parent connection controllers in other administrative domains by means of the cooperative model. An administrative domain may also be subdivided and the choice of federation model employed between domains within an administrative domain can be independent of what happens in another administrative domain. It is therefore possible to combine both federation models to construct large networks as illustrated in Figure 7.5. The principle described above can also be applied to federations of call controllers.

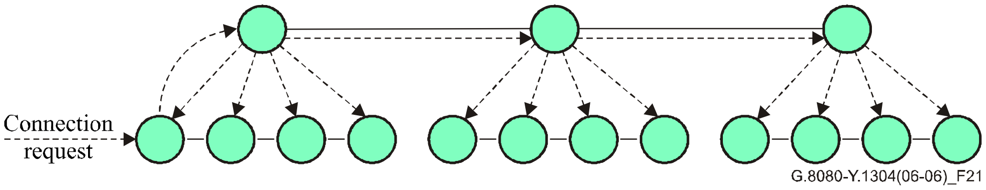


Figure 7.5 – Combined federation model

7.3 Architectural components

The components of the control plane architecture are described in this clause. Components can be combined in different ways, depending upon the required functionality. Appendix V illustrates examples of interactions of these components for use in connection set-up. Each component is described by a brief description of its primary function in this reference architecture. Component interfaces are provided next, and a more detailed description of operation is then given.

The connection controller, routing controller, calling/called party call controller, and network call controller are control plane components. These components are either public, in which case they use public SNPPs only, or private, in which case they use the SNPPs associated to a particular VPN. The VPN context of a control plane component is provided by the protocol controller associated with that component.

7.3.1 Connection controller (CC) component

The connection controller is responsible for coordination among the link resource manager, routing controller, and both peer and subordinate connection controllers for the purpose of the management and supervision of connection set-ups, releases and the modification of connection parameters for existing connections. This component services a single subnetwork, and provides the abstract interfaces to other control plane components given in Table 2. The connection controller component is illustrated in Figure 7.6.

NOTE – The route query interface does not apply for the CC interface at the UNI reference point.

In addition, the CC component provides a connection controller interface (CCI). This is an interface between a subnetwork in the transport plane and the control plane. It is used by control components to direct the creation, modification, and deletion of SNCs. Policy is not applied to the CCI.

Table 2 – Connection controller component interfaces

Input interface	Basic input parameters	Basic return parameters
Connection request in	A pair of local SNP names and optionally a route	A subnetwork connection
Peer coordination in	1) A pair of SNP names; or 2) SNP and SNPP; or 3) SNPP pair; or 4) route.	Confirmation signal

Output interface	Basic output parameters	Basic return parameters
Route query	Unresolved route fragment	Route
Link connection request	–	A link connection (an SNP pair)
Connection request out	A pair of local SNP names	A subnetwork connection
Peer coordination out	1) A pair of SNP names; or 2) SNP and SNPP; or 3) SNPP pair; or 4) route.	Confirmation signal
Remote topology status out	Topology information (link and/or subnetwork) including resource availability	–

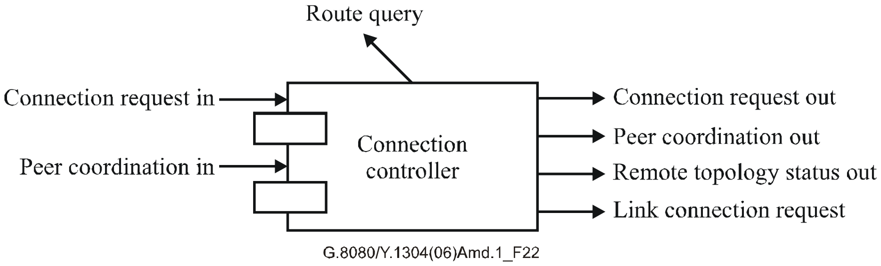


Figure 7.6 – Connection controller component

Remote topology status out: This interface is used to present topology status information learned by the connection controller. This status will include the ability to signal across a link.

Connection set-up operation

Connection set-up is performed in response to either a connection request, from an enclosing scope connection controller, or from a peer connection controller. In the case of hierarchical routing where the superior (i.e., parent) CC selects the source and destination SNPs, the connection request in/out interface is used. In all other cases, the peer coordination in/out interfaces are used. Component operation is the same in both cases.

The first unresolved portion of the route is resolved, via the route table query interface, into a set of links to be traversed, and this new set of links adds to the set. The connection controller inspects the new set of links to see which of these links are available for link connection allocation. Link connections are obtained and their links are removed from the link set. Next, corresponding subnetwork connections are requested from subordinate (i.e., child) connection controllers via the connection request out interface. Any unallocated route components are passed on to the next downstream peer connection controller. The actual sequence of operations depends on many factors, including the amount of routing information available and the access to particular link resource managers; however, the operation of the connection controller is invariant. Connection release is an analogous operation to connection set-up, except the operations are reversed.

7.3.2 Routing controller (RC) component

The role of the routing controller is to:

– respond to requests for path (route) information needed to set up connections. This information can range from end-to-end path details to a next hop. The route can be computed by one or more cooperating RCs;

– respond to requests for topology (SNPs and their abstractions) information for network management purposes.

As stated in clause 7.1, the routing controller component is an abstract entity that provides the routing function. It can be implemented as a single entity, or as a distributed set of entities that make up a cooperative federation.

Information contained in the route controller enables it to provide routes within the domain of its responsibility. This information includes both topology (SNPPs, SNP link connections) and SNP addresses (network addresses) that correspond to the end system addresses all at a given layer. Addressing information about other subnetworks at the same layer (peer subnets) may also be maintained. It may also maintain knowledge of SNP state to enable constraint based routing. Using this view, a possible route can be determined between two or more (sets of) SNPs taking into account some routing constraints. There are varying levels of routing detail that span the following:

– reachability (e.g., Distance Vector view – addresses and the next hops are maintained);

– topological view (e.g., Link State – addresses and topological position are maintained).

The routing controller has the interfaces provided in Table 3 and illustrated in Figure 7.7.

Table 3 – Routing controller interfaces

Input interface	Basic input parameters	Basic return parameters
Route query	Unresolved route element	Route
Local topology in	Local topology update	–
Network topology in	Network topology update	–
Remote topology in	Topology information (link and/or subnetwork) including resource availability	

Output interface	Basic output parameters	Basic return parameters
Route query	Unresolved route element	Route
Local topology out	Local topology update	–
Network topology out	Network topology update	–

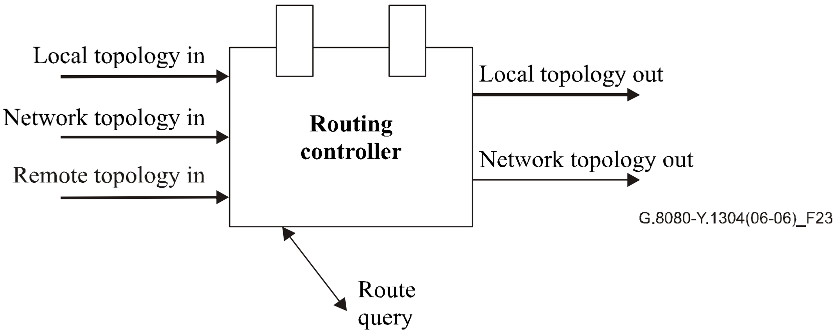


Figure 7.7 – Routing controller component

Route query interface: This interface accepts an unresolved route element and returns a route. An RC may query another RC (or RCs) in parent or child routing areas to assist in resolving the route. Examples of query results are:

1) Return an egress SNPP on this subnet that is on a path to a given destination SNPP.

2) Return a sequence of subnetworks that form a path between a given source/destination SNPP pair.

3) Return a sequence of subnetworks that form a path between two sets of SNPPs.

4) Return a sequence of SNPPs that form a path between a given source/destination SNPP pair.

5) Return a sequence of SNPPs that form a path between a given source/destination SNPP pair and includes one or more specific SNPPs.

6) Return a sequence of SNPPs that form a path between a given source/destination SNPP pair that is diverse from a given path.

The SNPPs returned must, either be all public or all associated to the same VPN.

Local topology interface: This interface is used to configure the routing tables with local topology information and local topology update information. This is the topology information that is within the domain of responsibility of the routing controller. Local topology information is identified to be either public or be associated to a particular VPN.

Network topology interface: This interface is used to configure the routing tables with network topology information and network topology update information. This is the reduced topology information (e.g., summarized topology) that is outside the domain of responsibility of the routing controller. Network topology information is identified to be either public or be associated to a particular VPN.

Remote topology In: This interface is used to accept topology information from a connection controller.

7.3.3 Link resource manager (LRMA and LRMZ) component

The LRM components are responsible for the management of an SNPP link; including the assignment and unassignment of SNP link connections (to a connection), managing resource reservation, configuration of policing and shaping functions (if required), providing topology and status information. LRM functions for circuit and packet switching are shown in Figure 7.8. Since an SNPP link can be either public or private, an LRM can also be either public or associated to exactly one VPN.

Layer network using circuit switching

The TAP supplies FwPt[1] and the corresponding resource labels to the LRM and associates these resource labels to SNP identifiers. When the TAP allocates an SNP identifier, the transport plane link connection is created; this provides an implicit reservation of the link resource. Tracking the assigned SNP identifiers allows the LRM to track link utilization. Since traffic loading is inherently constrained, policing and shaping functions are not required. In general, the same SNP identifier and resource label are used for both directions of a bidirectional connection.

Layer network using packet switching

The TAP supplies capacity and resources label to the LRM and associates these resource labels to SNP identifiers. The transport plane FwP and link connections are not created until an SNP identifier is assigned by the LRM. When a connection is requested, the LRM must record and track the requested bandwidth (CIR and EIR). The LRM (in cooperation with the TAP) should also configure the appropriate policing and shaping functions. The link information must include the admission control policy (e.g., amount of overbooking allowed for the CIR and EIR). The LRM advises the TAP when an SNP is assigned or unassigned. Different SNP identifiers and resource labels may be used for each direction of a bidirectional connection.

Required LRM functions for packet switching

- Bandwidth management function: the connection request must include bandwidth parameters (CIR and EIR). Due to the fact that the bandwidth is now the important link resource, SNPs and the corresponding labels identifying a flow are of less significance compared to circuit switching. In circuit switching, the SNP of a particular layer network has an implicit bandwidth, and the availability of an SNP also implies that the associated bandwidth is available. The LRMA shall provide a bandwidth management function that keeps track of the allocated bandwidth provided by the TAP and the assigned bandwidth that the currently existing connections have been granted.
- Connection admission control (CAC) function: When the LRMA receives a connection create request or connection modification request, the LRM's connection admission control function determines whether the request can be granted or whether it has to be rejected.
- SNP assignment function: If the CAC function result is positive, the connection create request is further processed and an SNP has to be selected and assigned to the connection. When a connection is deleted, the SNP is unassigned.

[1] FwPt – Forwarding Port from [ITU-T G.800]. This is the end of an unbound link connection. It is converted into a FwP (Forwarding Point) when a subnetwork connection is established. This does not change the associated resource label.

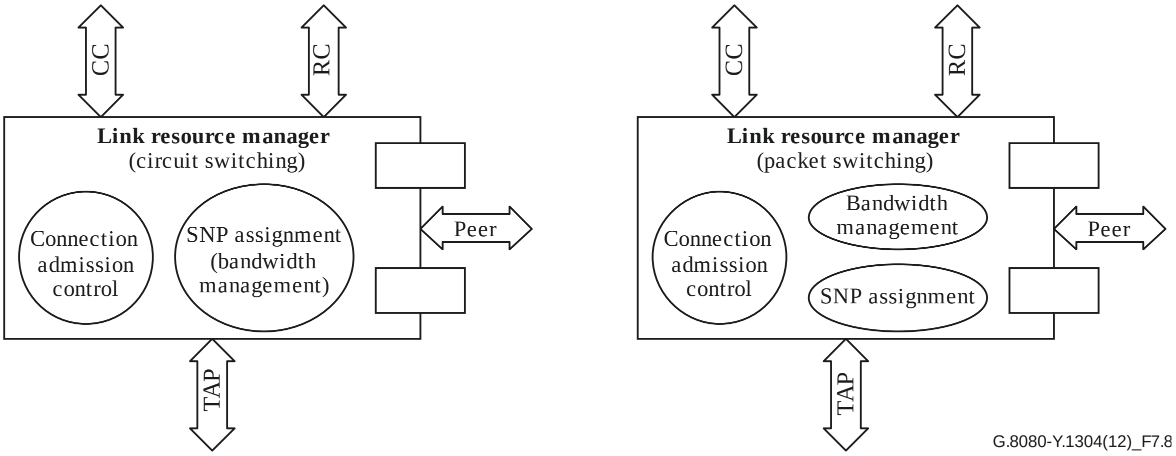


Figure 7.8 – Basic LRM functions for circuit switching and packet switching

Two LRM components are used – the LRMA and LRMZ. An SNPP link is managed by a pair of LRMA and LRMZ components one managing each end of the link. Requests to assign SNP link connections are only directed to the LRMA. If the required link resources are not available, then the LRM must either request additional capacity from the TAP or reject the connection request.

The two cases for SNPP link are illustrated in Figure 7.9.

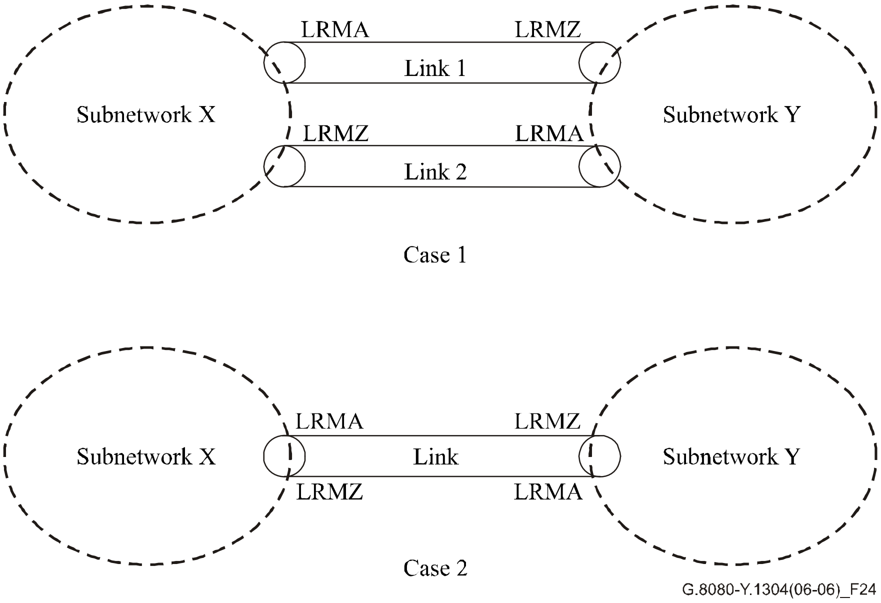


Figure 7.9 – SNPP link cases

In case 1, link 1 is dedicated to connection set-up requests originating from subnetwork X. Requests for SNP link connections from subnetwork X are directed to the adjacent LRMA for link 1, which can process the request without negotiation with the far end of the link. This LRMA can assign the SNP identifier and capacity (and hence the link connection) without negotiation with the LRMZ for link 1. Similarly, link 2 is dedicated to connection set-up requests originating from subnetwork Y. Requests for SNP link connections from subnetwork Y are directed to the adjacent LRMA for link 2. This LRMA can assign the SNP identifier without negotiation with the LRMZ for link 2. In this case, the same SNP identifier is used for both directions of transmission in a bidirectional connection. For a packet-switched network, the bandwidth assigned to a bidirectional connection may be asymmetric and must be tracked, by LRMA, independently for each direction. Also for packet-switched networks, the LRMA and LRMZ, in addition to assigning the SNP identifier, must communicate with the TAP to configure the policing and shaping functions.

In case 2, the link is shared between subnetworks X and Y for connection set-up. Requests for SNP link connections from subnetwork X are directed to the adjacent LRMA, since an LRMA component at the far end of the link can independently allocate SNP identifiers and link resources, the LRMA may need to negotiate an SNP identifier and capacity assignment with the LRMA at the far end (via the LRMZ at the far end). A similar process is required for request from subnetwork Y to its adjacent LRMA. Case 2 can be broken down into three sub-cases:

a) The same SNP identifier is used for both directions of a bidirectional connection.

b) The SNP identifiers are assigned independently for each direction at the source end of the link.

c) The SNP identifiers are assigned independently for each direction at the sink end of the link.

7.3.3.1 LRMA

The LRMA is responsible for the management of the A end of the SNPP link as described below.

The LRMA component interfaces are provided in Table 4 and illustrated in Figure 7.10.

Table 4 – LRMA component interfaces

Input interface	Basic input parameters	Basic return parameters
Connection request	Request id SNP Id (optional) CIR and EIR (packet switched only)	Request id SNP id pair or denied
Connection deletion	SNP Id; or request id	Confirm or denied
Configuration	Link information	–
Translation	Local id	Interface id
Connection modification	SNP Id; or request id New CIR and EIR (packet switched only)	Confirm or deny
SNP binding state	Busy, potential, allocated, shutting down	Resource released (in response to the shutting down state)
SNP operational state	Enabled, disabled	
Add SNP	List of SNP identifiers	confirm
Withdraw SNP	List of SNP identifiers	confirm
Output interface	**Basic output parameters**	**Basic return parameters**
Assign SNP (Case 1 only)	SNP id	confirm
SNP negotiation (Case 2 only)	Request id List of SNP ids CIR and EIR (packet switched only)	Request id SNP id
SNP release (unassign)	List of SNP id	Confirm

Table 4 – LRMA component interfaces

Topology	Link information	–
SNP id assignment/ unassignment (packet switched only)	SNP id CIR and EIR (to TAP)	Confirm or deny (TAP must bind the SNP to the resource label and configure the policing functions)
Capacity change request (packet switched only)	SNP id CIR and EIR	Link configuration

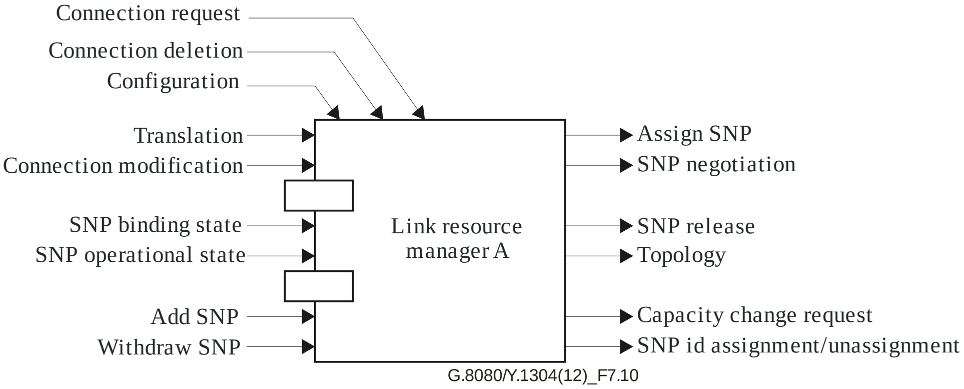


Figure 7.10 – Link Resource Manager A component

- **Functions**

– *Assignment of a link connection to a connection*

When a connection request is received, connection admission is invoked to decide if there is sufficient free resource to allow a new connection. Connection admission can also be decided based on prioritization or on other policy decisions. Connection admission policies are outside the scope of standardization.

For the purposes of the description below, the network configuration is shown in Figure 7.9, and the connection request originates from subnetwork X. The designations x and y are added to the LRM components to clarify the location and role.

If there are insufficient local resources, the request is rejected or the local LRMAx may request the TAP to convert an SNP with a binding state of potential and the associated potential capacity to allocated (if the policy allows such requests).

If sufficient local resources are available, the connection request is allowed to process as described in the cases below. Note that, for circuit-switched networks, the local LRMAx can determine if sufficient capacity is available since the bandwidth is always symmetric and is implicit in the availability of an SNP. For packet-switched networks, since the bandwidth may be asymmetric and must be tracked explicitly, it is possible that there are insufficient far-end resources available to support a connection, in which case the connection request will be rejected.

– Case 1 circuit-switched layer network: Since the SNP identifiers (and the corresponding link connections) are only assigned from one end of the SNPP link, the LRMAx can select the SNP identifier without interaction with the LRMZy at the far end of the link. The LRMAx passes the SNP identifier to the connection controller.

– Case 1 packet-switched layer networks: Since the link resources are only assigned from one end of the link, the LRMAx can perform the admission control and bandwidth reservation process without interaction with LRMZy at the far end of the link. LRMAx selects the SNP identifier, configures the policing and shaping functions and informs LRMZy. LRMZy must assign the SNP id and communicate with the TAP to configure the policing and shaping functions. LRMAx tracks the capacity assigned to connections for both directions of transmission on the link. The LRMAx passes the SNP identifier to the connection controller.

- Case 2a circuit-switched layer networks: Since the SNP identifiers (and the corresponding link connections) may be used by the LRMA at either end of the SNPP link, the LRMAx passes a list of usable SNP ids to the LRMZy. The LRMZy (in cooperation with its local LRMAy) selects one of the SNPs and returns the id to the originating LRMAx. The originating LRMAx passes the SNP identifier to the connection controller.
- Case 2a packet-switched layer networks: LRMAx adds the requested capacity for the A to Z direction of transmission on the link to its local copy of the link capacity assignment. Since the resources are assigned independently by the LRMA at either end of the SNPP link and SNPs are assigned from a common pool, the LRMAx passes a list of the useable SNP ids and the bandwidth parameters for the Z to A direction of transmission to the LRMZy at the far end of the link. The LRMZy passes this information to the local LRMAy which confirms that the link capacity is available, adds this to its local copy of the assigned link capacity and selects and assigns an SNP identifier and communicates with the TAP to configure the policing and shaping functions. If the available resources are insufficient to support the connection, the request is rejected; or the LRMAy may request additional resources from the TAP. This information is returned to the originating LRMAx. If the request has been accepted by the remote LRMAy, the local LRMAx assigns the SNP, communicates with the TAP to configure the policing and shaping functions. The LRMAx passes the SNP identifier to the connection controller. If the request is denied by the remote LRMAy, the local LRMAx rejects the connection request and removes any local reservations.
- Case 2b packet-switched layer networks: LRMAx adds the requested capacity for the A to Z direction of transmission on the link to its local copy of the link capacity assignment, and selects an SNP identifier for the A to Z direction of transmission. Since the resources are assigned independently by the LRMA at either end of the SNPP link, the LRMAx passes the selected SNP (for the A to Z) and the bandwidth parameters for the Z to A direction of transmission to the LRMZy at the far end of the link. The LRMZy passes the bandwidth requirements to the local LRMAy which confirms that the link capacity is available, adds this to its local copy of the assigned link capacity, and selects and assigns an SNP identifier (from its local pool for the A to Z direction of transmission) and communicates with the TAP to configure the policing and shaping functions. If the available resources are insufficient to support the connection, the request is rejected; or the LRMAy may request additional resources from the TAP. This information is returned to the local LRMZy which then assigns the SNP provided by the remote LRMAx and passes the information to the remote (originating) LRMAx. If the request has been accepted by the remote LRMAy, the local LRMAx assigns the SNP (for the A to Z direction of transmission), provides the Z to A SNP to the local LRMZx and communicates with the TAP to configure the policing and shaping functions. The LRMAx passes the SNP identifiers to the connection controller. If the request is denied by the remote LRMAy, the local LRMAx rejects the connection request and removes any local reservations.
- Case 2c packet-switched layer networks: LRMAx adds the requested capacity for the A to Z direction of transmission on the link to its local copy of the link capacity assignment and selects an SNP identifier for the Z to A direction of transmission. Since the resources are assigned independently by the LRMA at either end of the SNPP link, the LRMAx passes the selected SNP and the bandwidth parameters for the Z to A direction of transmission to the LRMZy at the far end of the link. The LRMZy passes the bandwidth requirements and SNP to the local LRMAy which confirms that the link capacity is available, adds this to its local copy of the assigned link capacity and selects an SNP identifier (from its local pool

for the Z to A direction of transmission), assigns the SNP identifier provided by the remote LRMAx and communicates with the TAP to configure the policing and shaping functions. If the available resources are insufficient to support the connection, the request is rejected; or the LRMAy may request additional resources from the TAP. This information is returned to the local LRMZy which then assigns the SNP provided by the local LRMAy and passes the information to the remote (originating) LRMAx. If the request has been accepted by the remote LRMAy, the local LRMAx assigns the SNP (for the A to Z direction of transmission), provides the Z to A SNP to the local LRMZx, communicates with the TAP to configure the policing and shaping functions. The LRMAx passes the SNP identifiers to the connection controller. If the request is denied by the remote LRMAy, the local LRMAx rejects the connection request and removes any local reservations.

– *Deletion of a connection*

Case 1: When a request to delete a connection is received, the corresponding SNP is marked as unassigned and the corresponding resources are removed from the assigned link capacity. The associated LRMZy is informed so that it can release the SNP identifier.

Case 2: When a request to delete a connection is received, LRMAx marks the corresponding SNP identifier as unassigned and the corresponding resources are removed from the assigned link capacity. Both the local LRMZx and the LRMZy at the far end of the link are informed. The LRMZ releases the SNP identifier. The remote LRMZy passes the request to its local LRMAy which marks the SNP identifier as unassigned and removes the resource reservation.

– *Interface to local id translation*

If required, the LRM provides the translation of an interface id to a local id. This is used, for example, if the ends of the SNPP link are in different routing areas.

- **Topology**

This function provides the link topology using the interface SNPP ids; the allocated SNP ids; assigned SNP ids; allocated capacity (packet switched only); assigned capacity (packet switched only).

It also provides link characteristics, e.g., link cost, diversity and quality. Some characteristics, for example link cost, may vary with link utilization. The process used to modify link characteristics is controlled by a local policy.

7.3.3.2 LRMZ

The LRMZ is responsible for the management of the Z end of the SNPP link as described below.

The LRMZ component interfaces are provided in Table 5 and illustrated in Figure 7.11.

Table 5 – LRMZ component interfaces

Input interface	Basic input parameters	Basic return parameters
SNP assignment	SNP id CIR and EIR (packet switched only)	Confirmation
SNP negotiation In (Case 2 only)	Request id List of SNP ids CIR and EIR (packet switched only)	Request id SNP id or denied
SNP unassignment	SNP id	Confirmation
Configuration	Link information	–
Translation	Local id	Interface id
Output interface	**Basic output parameters**	**Basic return parameters**
Topology	Link information	–
SNP id assignment/unassignment	(Packet switched only) SNP id	

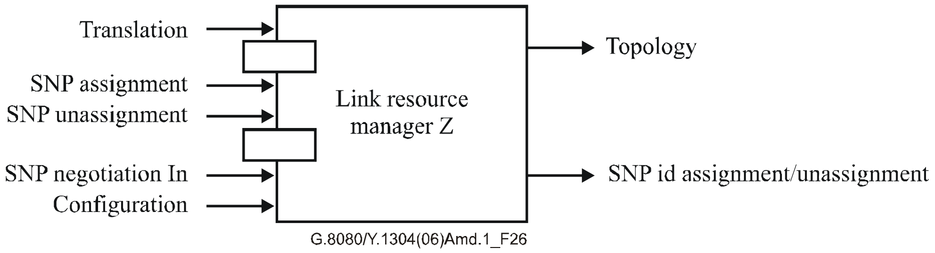


Figure 7.11 – Link resource manager Z component

- **Functions**

– *Assignment of SNP identifiers (case 1 only)*

When the remote LRMAx requests LRMZy to assign an SNP, LRMZy implements the request and, in the case of packet-switched networks, it also informs the TAP which configures the shaping and policing function (if required).

– *Negotiation and assignment of SNP (only used for case 2)*

- Case 2a circuit-switched networks: When a list of usable SNP ids is received from the remote LRMAx, one is selected (by the local LRMAy) and returned.
- Case 2a packet-switched networks: When a list of usable SNP ids and connection bandwidth parameters are received, the local LRMAy is informed. If sufficient capacity is available, the local LRMAy selects and returns an SNP identifier to the LRMZy. The LRMZy assigns this SNP identifier and informs the originating LRMAx. If the local LRMAy determines that the available link capacity is not sufficient, the request is denied.
- Case 2b packet-switched networks: When an SNP id and connection bandwidth parameters are received, the local LRMAy is informed. If sufficient capacity is available, the local LRMAy selects and returns an SNP identifier to the LRMZy. The LRMZy assigns the SNP identifier provided by the remote LRMAx and returns the SNP id provided by the local LRMAy to the remote (originating) LRMAx. The originating LRMAx provides this SNP id to its local LRMZx so that it can be assigned. If the local LRMAy determines that the available link capacity is not sufficient, the request is denied.

- Case 2c packet-switched networks: When an SNP id and connection bandwidth parameters are received, the local LRMAy is informed. If sufficient capacity is available, the local LRMAy selects and returns an SNP identifier to the LRMZy. The LRMZy assigns this SNP identifier and returns it to the remote (originating) LRMAy. The originating LRMA then provides the SNP id to the local LRMZx so that it can be assigned. If the local LRMAy determines that the available link capacity is not sufficient, the request is denied.

– *Unassignment of SNP identifiers in case 1*

When the associated LRMAx indicates that an SNP has been unassigned, the corresponding SNP identifier in LRMZy is marked as available.

– *Unassignment of SNP identifier (only used for case 2)*

When the associated LRMAx indicates that an SNP has been unassigned, the SNP is marked as available. The local LRMAy is also informed.

– *Interface to local id translation (case 1 only)*

If required, the LRM provides the translation of an interface id to a local id. This is used, for example, if the ends of the SNPP link are in different routing areas.

Topology (case 1 only)

This function provides the link topology using the interface SNPP ids; allocated SNP ids; assigned SNP ids; allocated capacity (packet switched only); assigned capacity (packet switched only).

7.3.4 Traffic policing (TP) component

This component is a subclass of policy port, whose role is to check that the incoming user connection is sending traffic according to the parameters agreed upon. Where a connection violates the agreed parameters, then the TP may instigate measures to correct the situation.

NOTE – This is not needed for a continuous bit rate transport layer network, and is not further expanded in this Recommendation. Likewise the TP policy interface will not be elaborated in this Recommendation.

7.3.5 Call controller components

Calls are controlled by means of call controllers. There are two types of call controller components:

– A calling/called party call controller: This is associated with an end of a call and may be co-located with end systems or located remotely and acts as a proxy on behalf of end systems. This controller acts in one, or both, of two roles, one to support the calling party and the other to support the called party.

– A network call controller: A network call controller provides three roles, one for support of the calling party, another to support the called party and a third to support calls across domain boundaries.

A calling party call controller interacts with a called party call controller by means of one or more intermediate network call controllers.

Note, the call control is only necessary at inter-domain signalling associations. In a call, signalling and connection routing progress to complete the connection, realizing the call. As such, when domain boundaries are crossed, NCCs are used. Call state is recorded in NCCs.

Further, NCCs are not hierarchical; the NCC that exists at the edge of a set of nested domains must implement the policy for all of those domains. Separate NCCs do not exist for each of the nested domains.

7.3.5.1 Calling/called party call controller

The role of this component is:

- generation of outgoing call requests
- acceptance or rejection of incoming call requests
- generation of call termination requests
- processing of incoming call termination requests
- call state management.

This component has the interfaces provided in Table 6. The calling/called party call controller component is illustrated in Figure 7.12.

Table 6 – Calling/called party call controller component interfaces

Input interface	Basic input parameters	Basic return parameters
Call accept	Transport resource identifier, VPN transport resource identifier or call name	Confirmation or rejection of call request
Call release in	Transport resource identifier or VPN transport resource identifier	Confirmation of call release
Call modification accept	Call name, parameters to change	Confirmation or rejection of call modification

Output interface	Basic output parameters	Basic return parameters
Call request	Transport resource identifier or VPN transport resource identifier; route (optional, for VPN only)	Confirmation or rejection of call request
Call release out	Transport resource identifier or VPN transport resource identifier	Confirmation of call release
Call modification request	Call name, parameters to change	Confirmation or rejection of call modification

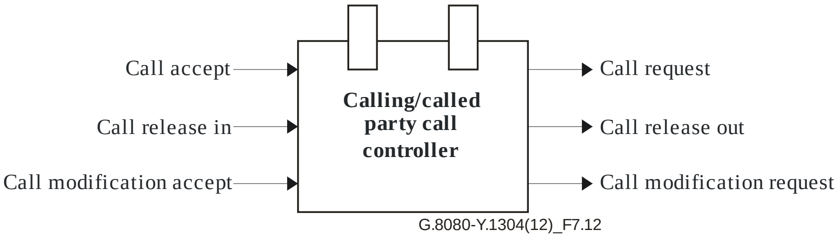


Figure 7.12 – Calling/called party call controller component

Call request: This interface is used to place requests for set-up, maintenance and release of a call. This interface also accepts a confirmation or rejection of a call request.

Call accept: This interface is used to accept incoming call requests. It also confirms or rejects the incoming call request.

Call release: This interface is used to place, receive and confirm release requests.

Call modification request: This interface is used to place requests to modify an existing call. It also receives the confirmation or rejection of the request.

Call modification accept: This interface is used to accept incoming requests to modify an existing call. It also confirms or rejects the request.

Note that the same calling/called party call controller may play the role of originator or terminator in different transactions.

7.3.5.2 Network call controller

Network call controllers are instantiated at domain boundaries (i.e., at E-NNI reference points or UNI reference points, where the call parameters need to be examined, e.g., different administrations, different recovery domains, etc.).

Call controllers that are adjacent (in the context of a call) form a call segment.

The role of this component is:

- processing of incoming call requests;
- generation of outgoing call requests;
- generation of call termination requests;
- processing of call termination requests;
- translation from VPN call source and destination identifiers to transport resource identifiers;
- call admission control based on validation of call parameters, user rights and access to network resource policy;
- state management of client calls and itself;
- adaptation management of transport resources via the TAP component.

This component has the interfaces provided in Table 7 and illustrated in Figure 7.13.

Table 7 – Network call controller component interfaces

Input interface	Basic input parameters	Basic return parameters
Call request accept	UNI transport resource identifier or UNI transport resource identifier alias	Confirmation or rejection of call request
Network call coordination in	UNI transport resource identifier or UNI transport resource identifier alias	Confirmation or rejection
Call release in	UNI transport resource identifier or UNI transport resource identifier alias	Confirmation of call release
Client NCC coordination in	Optional client call parameters, optional client layer identification, transport resource identifiers	A pair of SNPs in the client layer.
Server NCC coordination in	A pair of SNPs	Confirmation or rejection of use
Call modification accept	Call name, parameters to change	Confirmation or rejection of call modification

Table 7 – Network call controller component interfaces

Output interface	Basic output parameters	Basic return parameters
Call indication	UNI transport resource identifier or UNI transport resource identifier alias	Confirmation of rejection of call request
Connection request out	UNI transport resource identifier or UNI transport resource identifier alias	A pair of SNPs
Network call coordination out	UNI transport resource identifier or UNI transport resource identifier alias	Confirmation or rejection of call request
Directory request	UNI transport resource identifier or UNI transport resource identifier alias	Local name
Policy out	Call parameters	Accept or rejection of call
Call release out	UNI transport resource identifier or UNI transport resource identifier alias	Confirmation of call release
Client NCC coordination out	A pair of SNPs in the client layer.	Confirmation or rejection of use
Server NCC coordination out	Optional call parameters, Layer identification, Transport resource identifiers	A pair of SNPs
Call modification request	Call name, parameters to change	Confirmation or rejection of call modification

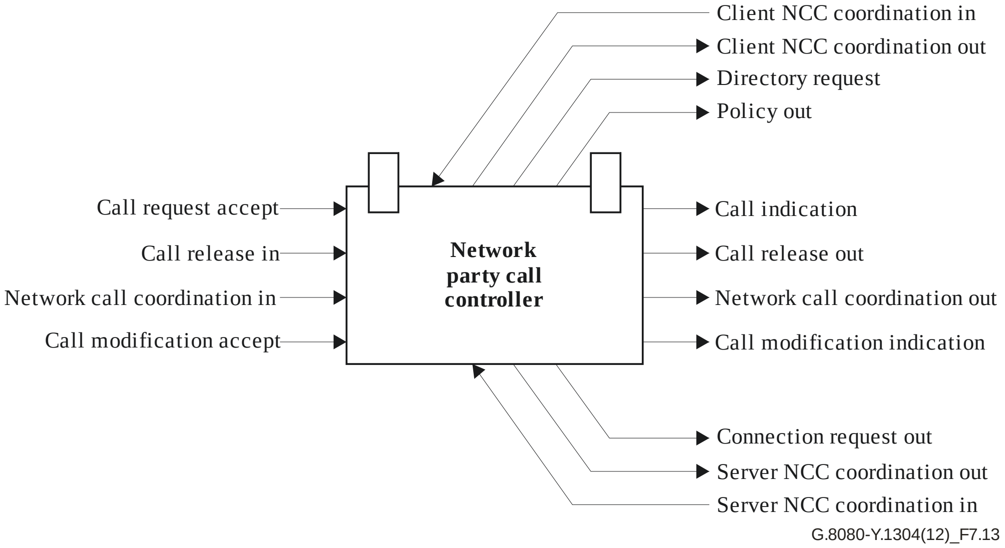

Client NCC coordination in
Client NCC coordination out
Directory request
Policy out
Call request accept
Call release in
Network call coordination in
Call modification accept
Network party call controller
Call indication
Call release out
Network call coordination out
Call modification indication
Connection request out
Server NCC coordination out
Server NCC coordination in
G.8080-Y.1304(12)_F7.13

Figure 7.13 – Network call controller component

Call request accept: This interface is used to accept a call source and destination identifier pair. This interface also confirms or rejects the incoming call set-up request.

Connection request out: This interface is used to place a connection set-up request to a connection controller as a pair of SNPs.

Directory request: This interface is used to get an SNPP name from a UNI transport resource identifier or alias. For aliases, it is a matter of policy which SNPP is returned if multiple SNPPs are represented.

Network call coordination: This interface is used for network level call coordination.

Call release in/out: These interfaces are used to place, receive and confirm release requests.

Policy out: This interface provides policy checking.

Client NCC coordination In: This interface is used to accept a request from a client layer NCC for a pair of SNPs. The NCC is provided with source and destination identifiers in its layer in order for it to provide a network connection for use by the client layer. SNPs in the client layer that are supported by an adaptation to the network connection are returned. This interface is also used by the client to release or modify the use of the SNP pair. The NCC returns the result of the action.

Client NCC coordination out: This interface is used to present a pair of SNPs to a client layer that are supported by an adaptation to a network connection. The client NCC indicates whether or not it accepts this resource. This interface is also used by the server to release or present a modified SNP pair. The client NCC returns the result of the action.

Server NCC coordination out: This interface is used to request a pair of SNPs (input and output) that can be used by the call to transfer characteristic information. It is identical to the return parameters of the connection request out interface except that a network connection in this layer is not assumed to be created. This interface is also used to release or request modification of the use of the SNP pair provided by the server layer. The server NCC returns the result of the action.

Server NCC coordination in: This interface is used to accept a pair of SNPs (input and output) presented from a server layer NCC. It may be accepted or rejected. This interface is also used by the server to release or present a modified SNP pair. The NCC returns the result of the action.

Call modification accept: This interface is used to accept a call modification request. This interface also confirms or rejects the incoming call modification request.

Call modification indication: This interface is used to continue a call modification request to another NCC. It also receives confirmation or rejection of the request.

The role of call admission control in the calling party network call controller is to check that a valid called user name and service parameters have been provided. The service parameters are checked against a service level specification. If necessary, these parameters may need to be renegotiated with the calling party call controller. The scope of this negotiation is determined by policies derived from the original service level specification, which itself is derived from the service level agreement.

The role of call admission control in the called party network call controller, if present, is to check that the called party is entitled to accept the call, based on the calling party and called party service contracts. For example, a caller address may be screened, and the call may be rejected.

The directory request interface of the network call controller is used to access a directory function that is used to transform identifiers between or within name spaces. An identifier is supplied as input to the directory function which returns one or more identifiers. How the directory is maintained or configured is outside the scope of this Recommendation and will be described in other Recommendations. Examples of mappings that could be provided by the directory function are:

– UNI transport resource identifier to SNPP identifier. A call controller requires an SNPP in order to make a request to a connection controller.

– UNI transport resource identifier alias to UNI transport resource identifier. An application of this is to be able to identify resources associated with a multi-homed AGC with one identifier which is a UNI transport resource identifier alias.

– SNPP identifier to UNI transport resource identifier. To use the server NCC coordination out and client NCC coordination in interfaces, UNI transport resource identifiers are needed that are in the server layer. This is used to make the server layer call. Before the call is made, the client layer does not have the UNI transport resource identifiers of the server layer, only the SNPPs in the client layer on the edge of the server layer subnetwork. A mapping between the client layer SNPPs and the server layer UNI transport resource identifiers provides the means of invoking the server layer call.

– SNPP alias to SNPP. This is used for coordination between routing levels.

7.3.5.3 Call controller interactions

The interaction between call controller components is dependent upon both the type of call and the type of connection, as described below.

Switched connections: The calling party call controller (associated with an end terminal) interacts with the network call controller to form an incoming call and the network call controller interacts with the called party call controller (associated with an end terminal) to form an outgoing call. The network call controller interacts with the connection controllers to provide the call. An example of this interaction is illustrated in Figure 7.14. It should be noted that the calling/called party call controllers have no direct interaction with the connection controller associated with the corresponding network call controller.

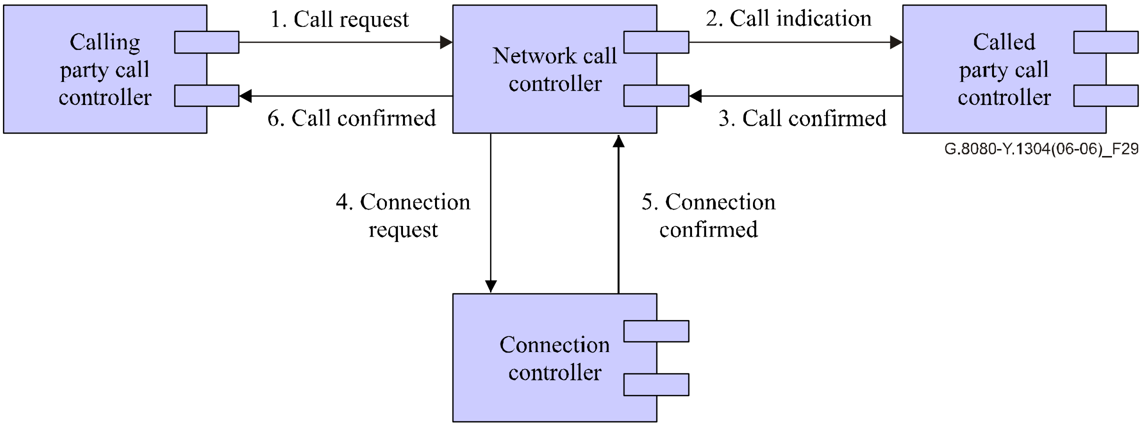


Figure 7.14 – Called/calling party call controller interaction for switched connections: Example 1

Figure 7.14 shows the situation whereby the called party call controller accepts the call, prior to the ingress network call controller requesting the connection. It is also valid to define the interaction such that the connection set-up follows the call, as is illustrated in Figure 7.15.

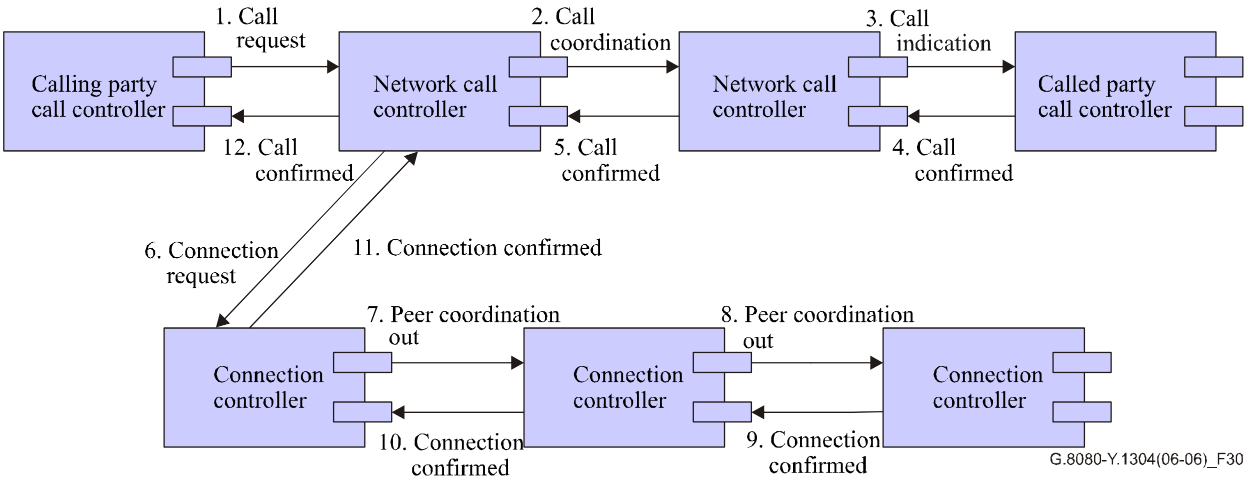


Figure 7.15 – Called/calling party call controller interaction for switched connections: Example 2

Soft permanent connections: The network management system is considered to contain the calling/called party controllers. The management system issues a command to configure the calling party call controller that initiates the network call controllers on the control plane when the call configuration commands are sent to the control plane. The response to a call configuration command from the control plane is considered as a call set-up confirmation by the management plane. This represents a null call with no service. The protocols between the network management plane and the control plane are a command and command response interface. Figure 7.16 illustrates the call controller interactions for soft permanent connections.

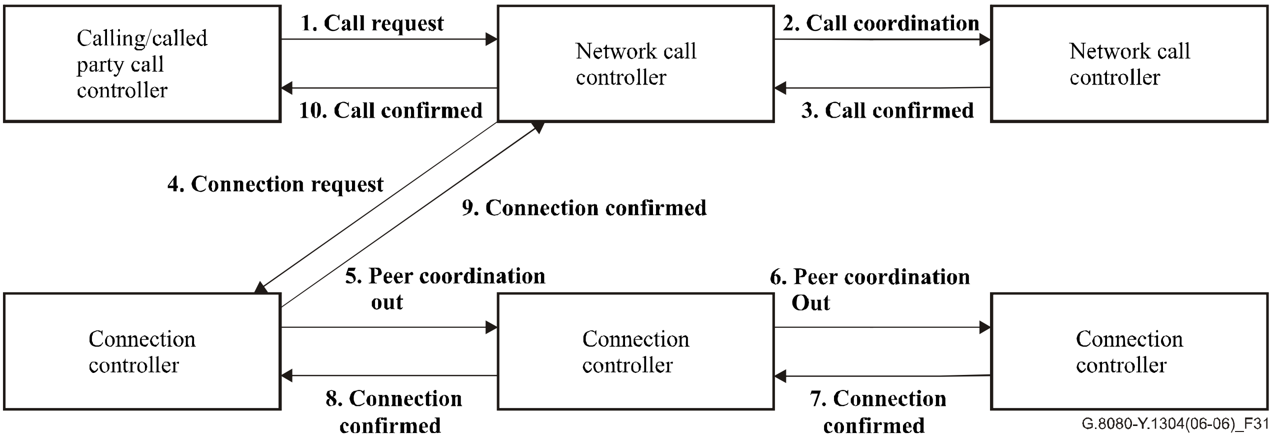
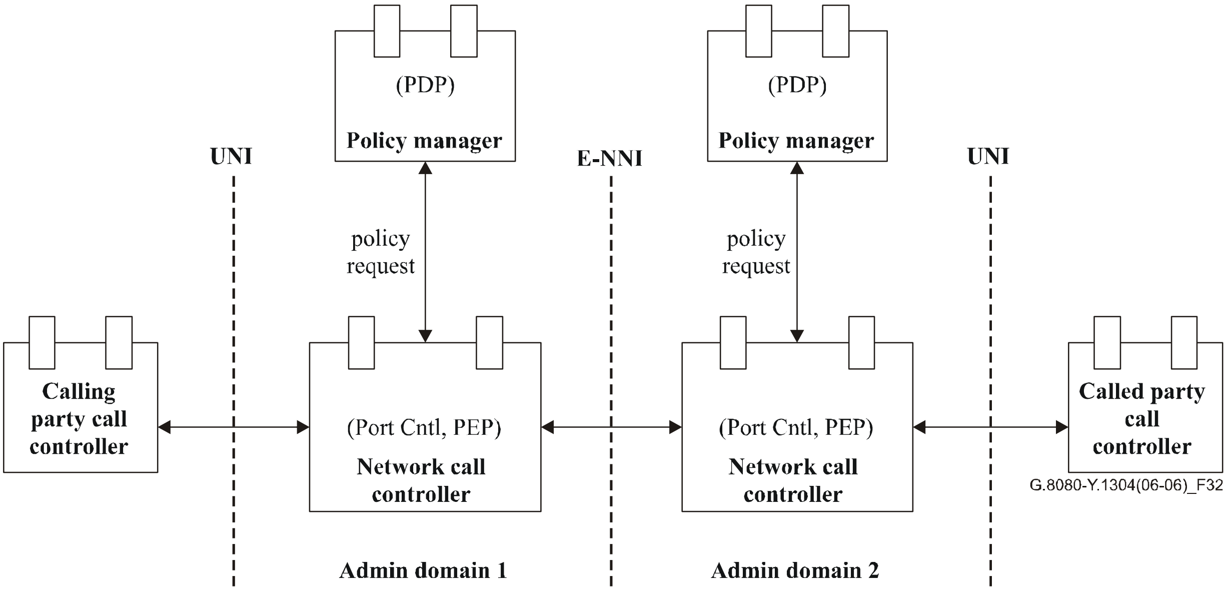


Figure 7.16 – Call controller interactions for soft permanent connections

Proxy call: The calling/called party call controller interacts with the network call controller by means of a call protocol, but is not coincident with the user.

Figure 7.17 indicates an example of the interactions necessary to support call admission control policy between network call controllers.



Port Cntl	Port Controller
PDP	Policy Decision Point
PEP	Policy Enforcement Point

Figure 7.17 – Example of call admission control policy interactions

Layered calls: Two NCCs in different layers may cooperate to allow support of client CI in a server layer. Use of these interlayer interfaces is governed by operator policy. This may be initiated either to or from a server layer depending on what layer the operation is initiated from. From an NCC, the request to a server layer NCC returns the same result as the "Connection Request Out" interface. The difference is that an association with a server NCC is made. This action either results in the use or creation of, a server layer call segment that will support the client NCC. If the server layer needs to create a call as a result of the use of the "Server NCC Coordination Out" or "Client NCC Coordination In" interfaces, the source and destination identifiers are used as call parameters. An identical action to the "Call Request Accept" interface behaviour is then performed if connection establishment at that server layer is determined to be the correct action. The server layer NCC could alternately use its "Server NCC Coordination Out" interface to make a (layer recursive) request for an SNP pair from another layer NCC that is a server to it.

An NCC could also initiate an action to a client layer whereby it presents a pair of SNPs that can be used by the client layer for transferring client CI. The "Client NCC Coordination Out" or "Server NCC Coordination In" interfaces are used for this purpose. When this interface is used, the SNP pair presented is able to transfer client CI and no call action at the server layer is initiated. This is used for an operation where a server layer has already established a call and this is presented to the client layer at a later point in time. The client layer may accept or reject the use of the offered SNP pair.

7.3.5.4 Call modification

The service provided by a call can be modified by actions initiated by a CCC or network management application acting on an NCC at the UNI. The degree of modification is set by operator policy and the policy may or may not be shared with the end user (e.g., informing the user of what bandwidth increments are allowed). The extent to which a call can be modified is subject to the following rules:

- The CI associated with the call at the UNI is not modifiable.
- The link connection end-points associated with the call at the UNI-N are not modifiable. They may be added/removed however when connections are added/removed from a call.

Actions can either be modification of a call segment where the NCCs remain fixed, or the creation/deletion of call segments within an overall call where NCCs are created/deleted.

Examples of what may be modified at the UNI include bandwidth (e.g., rate of Ethernet call) and number of CCCs involved (e.g., multi-party call).

Examples of what may occur within the network as a result of UNI call modification requests include:

- changing the number of server layer connections associated with a VCAT call that supports an Ethernet call.
- In response to a request to increase the availability of a call, adding an additional connection to create a 1+1 configuration.

7.3.5.5 Call failure handling

For a new call request, if the network is unable to establish all the connections required to satisfy the call request, any connections or partial connections that have been established will be torn down (deleted) and the call request will be rejected.

For call modifications, if the network is unable to add the connections requested, then the call modification is considered to have failed. Any connections or partial connections will be removed and no changes will be made to the existing call.

7.3.6 Directory service

The directory service component is responsible for identifier resolution and coordination among peer directory service components. The role of this component is to provide mappings between identifier spaces for other components.

NOTE – All interfaces as Table 8 below are not intended to be used in one instance of this component. Only the directory request interface might be required for basic usage, but distributed implementations might use more interfaces.

Directory service functions can be implemented in both distributed and centralized applications. In a centralized application, peer coordination interfaces of the DS component might be unused.

Table 8 – Directory service component interfaces

Input interface	Basic input parameters	Basic return parameters
Directory request in	1) UNI/E-NNI transport resource identifier; or 2) UNI/E-NNI transport resource identifier alias; or 3) SNPP identifier; or 4) SNPP alias;	1) SNPP identifier; or 2) UNI/E-NNI transport resource identifier; or 3) UNI/E-NNI transport resource identifier; or 4) SNPP identifier.
Peer coordination in	1) <UNI/E-NNI transport resource identifier, SNPP identifier> 2) <UNI/E-NNI transport resource identifier alias, UNI/E-NNI transport resource identifier> 3) <SNPP identifier, UNI/E-NNI transport resource identifier> 4) <SNPP alias, SNPP identifier> 5) <SNPP identifier, SNPP alias>	
Directory information in	1) <UNI/E-NNI transport resource identifier, SNPP identifier> 2) <UNI/E-NNI transport resource identifier alias, UNI/E-NNI transport resource identifier> 3) <SNPP identifier, UNI/E-NNI transport resource identifier> 4) <SNPP alias, SNPP identifier > 5) <SNPP identifier, SNPP alias> 6) list of transport resource identifiers	

Table 8 – Directory service component interfaces

Output interface	Basic output parameters	Basic return parameters
Directory request out	1) UNI/E-NNI transport resource identifier; or 2) UNI/E-NNI transport resource identifier alias; or 3) SNPP identifier; or 4) SNPP alias;	1) SNPP identifier; or 2) UNI/E-NNI transport resource identifier; or 3) UNI/E-NNI transport resource identifier; or 4) SNPP identifier.
Peer coordination out	1) <UNI/E-NNI transport resource identifier, SNPP identifier> 2) <UNI/E-NNI transport resource identifier alias, UNI/E-NNI transport resource identifier> 3) <SNPP identifier, UNI/E-NNI transport resource identifier> 4) <SNPP alias, SNPP identifier> 5) <SNPP identifier, SNPP alias>	
Directory information out	1) <UNI/E-NNI transport resource identifier, SNPP identifier> 2) <UNI/E-NNI transport resource identifier alias, UNI/E-NNI transport resource identifier> 3) <SNPP identifier, UNI/E-NNI transport resource identifier> 4) <SNPP alias, SNPP identifier> 5) <SNPP identifier, SNPP alias> 6) list of Transport Resource Identifiers	

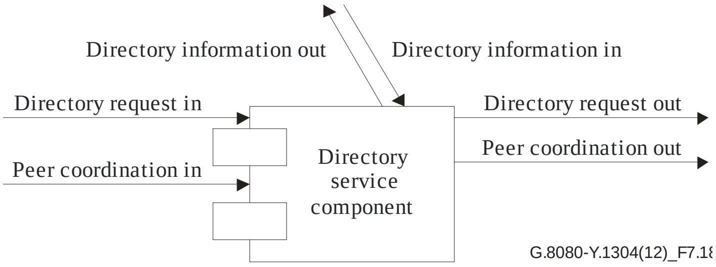


Figure 7.18 – Directory service component

Directory request in/out

This interface is used to get an SNPP identifier from a UNI/E-NNI transport resource identifier or alias. And this interface is also used to get UNI/E-NNI transport resource identifier from a UNI transport resource identifier alias or SNPP identifier. Directory request should be bidirectional. CC could initialize a directory request and send to/receive from a DS component when it needs to decode DS.

Peer coordination in

This interface is used to get directory information from a peer directory service component.

Peer coordination out

This interface is used to transmit directory information to a peer directory service component.

Directory information in/out

This interface is used to receive/send directory information from other components which could include management plane applications, and equipment management functions (EMFs) on subnetworks (e.g., NEs). The list of TRIs may be used by the DS component to create mappings to SNPPs and return in response to requests.

7.3.7 Discovery agent (DA)

The federation of discovery agents operates in the transport plane name space, and provides for separation between that space and the control plane names. The federation has knowledge of connection points (CPs) and termination connection points (TCPs) in the network, while a local DA has knowledge of only those points assigned to it. Discovery coordination involves accepting potential hints about pre-existing CPs and link connections. The DA holds the CP-CP link connections to enable SNP-SNP link connections to be bound to them later. The resolution interfaces assist in discovery by providing name translation from global TCP handles to the address of the DA responsible for the point, together with the local name of the TCP. Note that hints come from cooperation with other components, or from external provisioning systems.

Discovery agents have no private equipment interfaces, and can be located on any suitable platform.

Table 9 – Discovery agent (DA) component interface

Input interface	Basic input parameters	Basic return parameters
Coordination in		
Hints in	CP pairs	
Resolution request	TCP name	

Output interface	Basic output parameters	Basic return parameters
Coordination out		
CP link connection	CP pair	
Resolution result		DA DCN address, TCP index

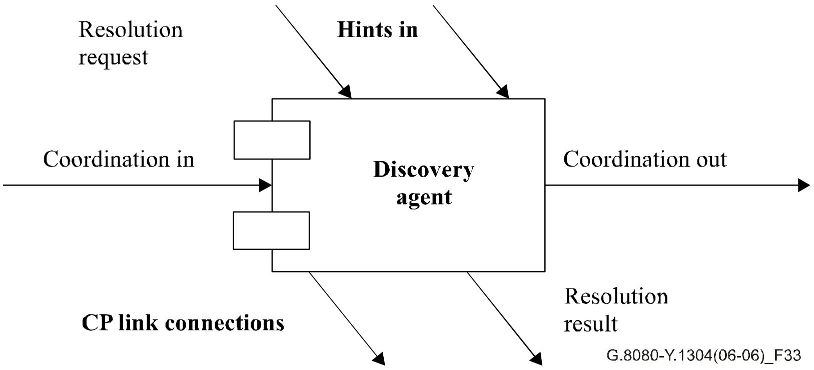


Figure 7.19 – Discovery Agent component

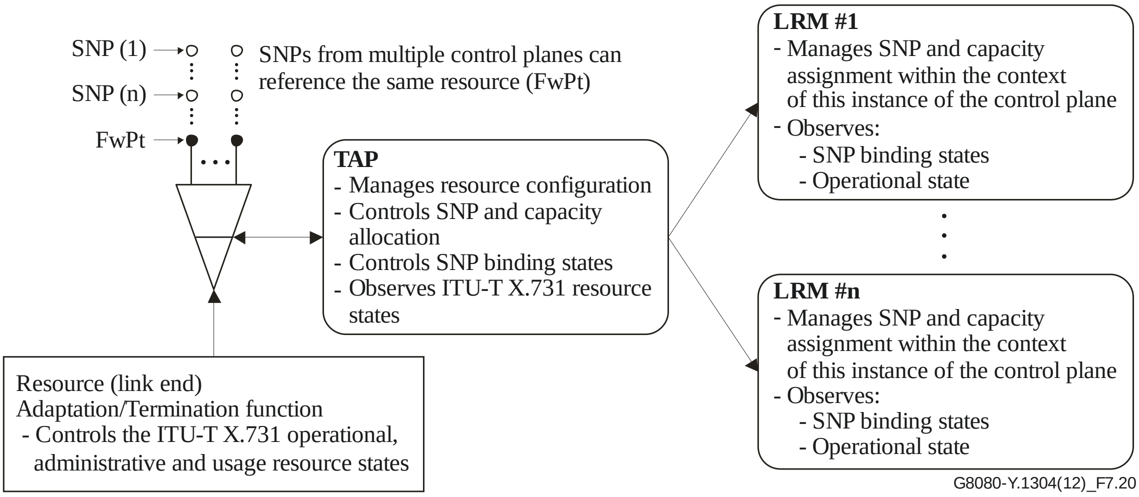
The applicability of this component to packet-switched networks is for further study.

7.3.8 Termination and adaptation performers

The TAP is collocated with the adaptation and termination function. It provides the control plane (the LRM) with a view of the status and utilization of the resource supporting a link, and hides any hardware and technology with specific details of the adaptation and termination control.

7.3.8.1 TAP resource model

Only those resources that will be utilized by a control plane are made visible to the TAP. Before a resource is permanently withdrawn from the control plane, all SNPs referencing the resource must be deleted. The relationship between TAP and other components is shown in Figure 7.20.


SNP (1)
SNP (n)
FwPt
SNPs from multiple control planes can reference the same resource (FwPt)
TAP
- Manages resource configuration
- Controls SNP and capacity allocation
- Controls SNP binding states
- Observes ITU-T X.731 resource states
LRM #1
- Manages SNP and capacity assignment within the context of this instance of the control plane
- Observes:
- SNP binding states
- Operational state
LRM #n
- Manages SNP and capacity assignment within the context of this instance of the control plane
- Observes:
- SNP binding states
- Operational state
Resource (link end)
Adaptation/Termination function
- Controls the ITU-T X.731 operational, administrative and usage resource states
G8080-Y.1304(12)_F7.20

FwPt (forwarding port) – The (unbound) end of a link connection. It is converted into a FwPt (forwarding point) when it is bound to a subnetwork connection. The binding does not change the resource label.

Figure 7.20 – Relationship between TAP, LRM, and transport plane

The termination and adaptation performer (TAP) operates at two different times and provides two different functions.

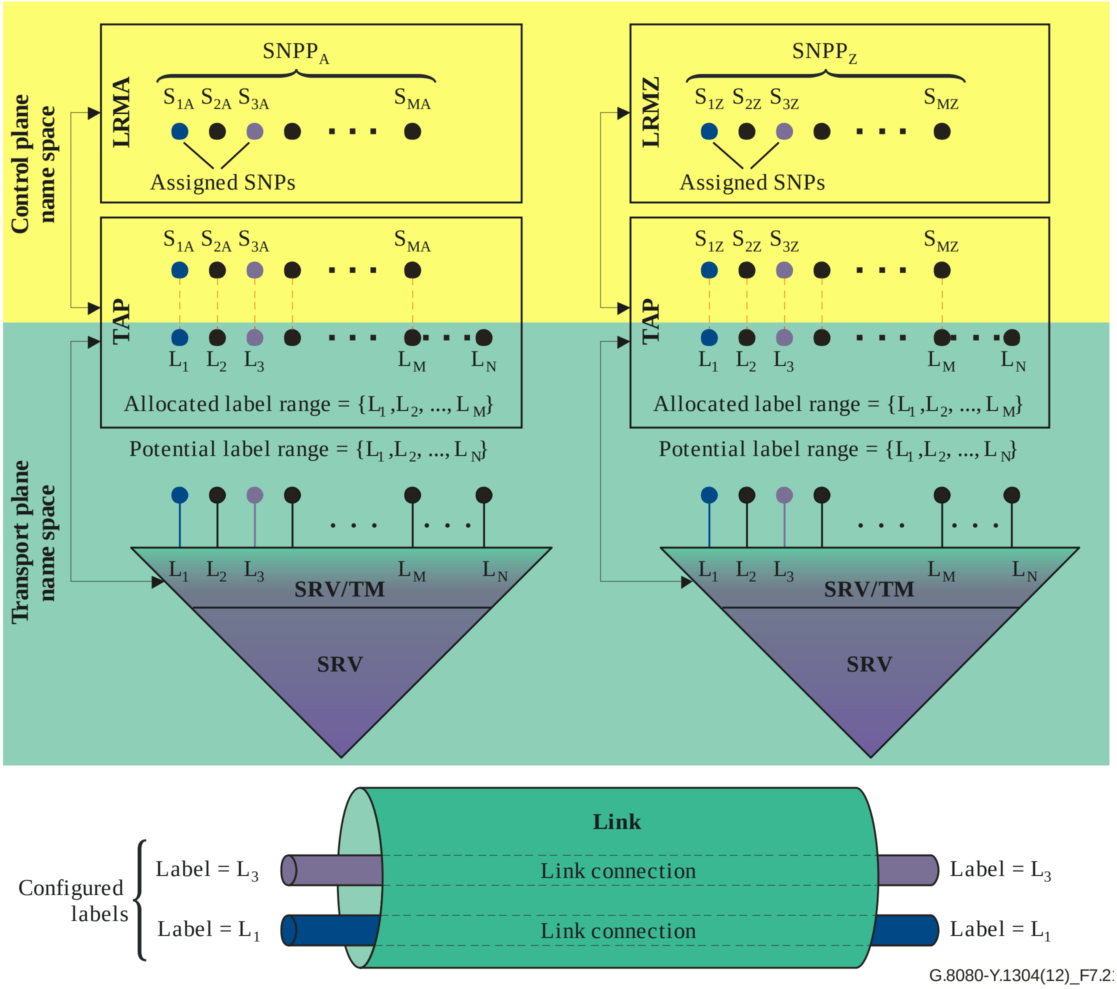
When a resource is assigned to a control plane, the TAP is configured with a list of the resource identifiers, the capacity of the link resource, together with the capacity reservation policy. For a circuit-switched network, only the resource labels are configured since they carry an implicit resource capacity and reservation policy. The link resources may be shared between multiple control planes (e.g., different layer networks or different layer 1 VPNs). For each LRM that is within the scope of the TAP (i.e., references resources controlled by the TAP), the TAP is configured with the permitted bindings between the resource labels and SNPs. The TAP controls the allocation of SNP identifiers and capacity to each LRM. In the case of packet-switched networks, the TAP provides the potential and allocated capacity (EIR and CIR) to each LRM, together with the capacity assignment policy. The LRM can only assign SNP identifiers or resource capacity that has been allocated by the TAP.

The TAP makes resources visible to an LRM by binding a resource label to an SNP identifier. The existence of the SNP identifier is independent of the configuration of the resources.

In the case of a circuit-switched network, when the TAP allocates the resource to an LRM (i.e., sets the SNP binding state to allocated), it also configures those resources and creates the transport plane FwPt and link connection. This configuration action is independent of the assignment of those resources to a connection. An LRM may request the TAP to modify the list of allocated SNP identifiers (i.e., change the binding state of the SNPs to allocated from potential or from allocated to potential).

In the case of a packet-switched network, when the TAP allocates capacity and resource labels to an LRM (i.e., sets the SNP binding state to allocated), it only performs configuration required to allow those resources to be activated. When the LRM assigns an SNP to a connection, the TAP creates the FwP (i.e., it activates the binding between the SNP and the resource label), it also configures the shaping and policing functions, if required. The LRM is responsible for the assignment of the allocated capacity within the constraints of the capacity reservation policy provided by the TAP. An LRM may request the TAP to modify the allocated capacity or the list of allocated SNP identifiers (i.e., change capacity or the binding state of the SNPs to allocated from potential or from allocated to potential).

Figure 7.21 below depicts the relationships between the potential and allocated resource identifiers, or resource labels, and SNPs. Moreover, it shows that the assigned SNPs are associated with labels from the allocated label range that are configured labels, i.e., a link connection exists for those SNPs.



Control plane name space
Transport plane name space
LRMA
SNPP$_A$
S_{1A} S_{2A} S_{3A} S_{MA}
Assigned SNPs
TAP
S_{1A} S_{2A} S_{3A} S_{MA}
L_1 L_2 L_3 L_M L_N
Allocated label range = {L_1 ,L_2, ..., L_M}
Potential label range = {L_1 ,L_2, ..., L_N}
L_1 L_2 L_3 L_M L_N
SRV/TM
SRV
LRMZ
SNPP$_Z$
S_{1Z} S_{2Z} S_{3Z} S_{MZ}
Assigned SNPs
TAP
S_{1Z} S_{2Z} S_{3Z} S_{MZ}
L_1 L_2 L_3 L_M L_N
Allocated label range = {L_1 ,L_2, ..., L_M}
Potential label range = {L_1 ,L_2, ..., L_N}
L_1 L_2 L_3 L_M L_N
SRV/TM
SRV
Link
Configured labels
Label = L_3
Link connection
Label = L_3
Label = L_1
Link connection
Label = L_1
G.8080-Y.1304(12)_F7.2:

Figure 7.21 – Control plane and transport plane link resource model

The various types of resource labels are:

– **Potential (resource) label range**

The "potential label range" is the full label range of resource labels in the transport plane name space that an adaptation function supports. In packet switching layers, this range can be much larger than the allocated label range. Example: the 20-bit MPLS label provides 2^{20} = 1'048'576 possible label values including reserved labels (label values 0..15) for specific purposes.

- **Configured (resource) label**

  A "configured label" is a label that has been configured in the transport plane in support of a connection. If a label is configured, a forwarding table entry exists on the receiving end of the link such that packets can be forwarded to an outgoing link if a packet is received with a label value that is equal to the configured label. If a label is configured, a packet flow can be distinguished from other flows and can be forwarded based on the label value. This is equivalent to the existence of a link connection. This means that a link connection is created whenever a label has been configured consistently on either end of a link. The deletion of the configuration entry in the transport plane also deletes the link connection.

- **Allocated (resource) label range**

  The "allocated label range" is the set of labels that can be used by the adaptation function of a particular link to carry user traffic. It is a subset of the potential label range. The allocated label range must not include reserved label values. When a system uses a per platform (system) label space, each interface is typically configured with an (allocated) label range that does not overlap with the label ranges of the other interfaces, and a specific label value is selected from this label range in response to, e.g., a control plane connection request.

  The allocated labels are entities that can be referenced in the transport plane name space. Each allocated label is associated with one or multiple SNP IDs that exist in the control plane name space (1:n relationship). In the simplest case, there is exactly one SNP ID per allocated label (1:1 relationship between allocated label and SNP ID). TAP holds the binding information between SNPs and an allocated label.

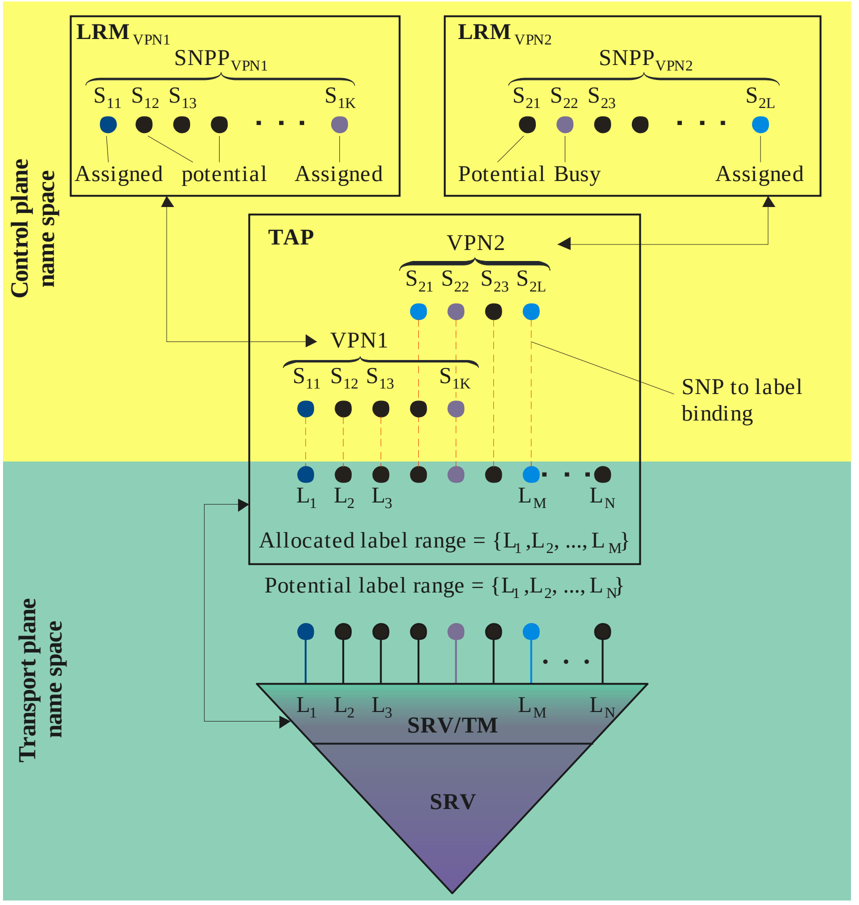
- **Potential SNPs**

  Potential SNPs are those SNPs that are associated with a label. In general, multiple SNPs can be associated with a single label.

- **Assigned SNPs**

  Assigned SNPs are those SNPs out of the set of potential SNPs that have been assigned to a particular connection. This means that the associated label is a configured label.

  Figure 7.22 depicts how VPNs can be modelled. In the provided example, the allocated label range is subdivided into three sub-sets, a sub-set that can exclusively be used by VPN1, a sub-set that can exclusively be used by VPN2, and a sub-set that is shared between VPN1 and VPN2. Each label of the sub-sets for exclusive use has a single SNP associated, whereas each label of the shared sub-set is associated with two SNPs, one SNP within the scope of LRM_{VPN1} and another one SNP within the scope of LRM_{VPN2}. When an SNP is assigned, e.g., by LRM_{VPN1} that corresponds to a label from the shared label sub-set, the SNP in LRM_{VPN2} becomes busy.



G.8080-Y.1304(12)_F7.22

Figure 7.22 – Control plane organizational model for VPNs

7.3.8.2 TAP states

The TAP holds the SNP binding states and the capacity allocation to each LRM, and provides a specific (coordinated) view to each LRM. As described in Table 10, the SNP binding states that the TAP provides to the LRM are constrained by the administrative state of the resources.

The transport plane resources are aware, from the ITU-T X.731 usage state (idle/busy), if the resources have been allocated to the TAP. The resources have no visibility of any allocation that the TAP makes to LRMs. Therefore, the resources should use the shutting down state to withdraw resources from the TAP.

The TAP uses the SNP binding states to allocate resources to LRMs. The TAP has no visibility of the assignment of those resources to connections. Therefore, the TAP should use the SNP binding state of shutting down to remove resources from the LRM.

Table 10 – SNP binding states

State	Description
Busy	Permitted binding, the resource label and capacity being referenced by the SNP is currently allocated to another control plane or the management plane.
Potential	Permitted binding, currently the resource label and capacity being referenced by the SNP is not allocated to any control plane or the management plane.
Allocated	Permitted binding and the resource label and capacity being referenced by the SNP has been configured for and allocated to this LRM.
Shutting down	TAP notification that the resource label and capacity being referenced by the SNP must be returned within an explicit timeframe e.g.: – immediately (interrupt the current call) – quickly (re-route call before dropping) – next maintenance window – when call is dropped.

When an SNP identifier is in the allocated state, the TAP must correctly configure the resources (e.g., variable adaptation) and set the state of any other SNPs referencing the same resource to busy.

When SNP identifiers are bound to their corresponding FwP, the TAP is responsible for holding the SNP-FwP binding. A local TAP cooperates with a remote TAP via the LRM to coordinate any variable adaptation or other coordination required when forming the FwP link connections.

If an LRM wishes to use capacity or an SNP with a binding state of "potential" to satisfy a connection request then during connection set-up, a pair of TAPs cooperate via the LRM to coordinate any adaptation set-up, or link resource allocation, required by the link connection.

When the TAP modifies the resource capacity that is allocated to an LRM, it also makes a corresponding adjustment to the potential resource capacity.

The TAP provides SNP state information to the LRM and accepts resource state status from the (transport plane) adaptation and termination functions to ensure that the management plane indications are consistent. Management plane consistency includes ensuring that the alarm state of the link connection is consistent, so that spurious alarms are neither generated nor reported.

There are three ITU-T X.731 states for transport resources:

- Operational: This state reflects the combined status of the trail supporting the link and adaptation function. It is controlled by the underlying resources and is observed by TAP.
- Administrative: This state reflects the permission to use the resource which is managed by a management interface to the TAP.
- Usage: This state reflects whether the resource is actively in use. As TAP allocates and de-allocates resources to the control plane, it adjusts the usage state accordingly.

Permitted combinations of the resource states and the SNP binding state for each SNP are described in Table 11 below:

Table 11 – Resource and SNP binding states

ITU-T X.731 resource states			SNP binding states	
Operational	**Administrative**	**Usage**	**LRM x**	**All other LRMs (Note 2)**
enabled/disabled (Note 1)	unlocked	Busy	Potential	Potential
enabled/disabled (Note 1)	unlocked	Busy	Allocated (Note 3)	Busy
enabled/disabled (Note 1)	shutting down (Note 4)	Busy	Shutting down	Busy
enabled/disabled	Locked (Note 5)	Idle	Busy (Note 6)	Busy (Note 6)

NOTE 1 – When an LRM observes that the operational state of a link is disabled, it may notify the routing controller component, it may also notify the connection controllers for the connections that are impacted. The call controller manages the recovery of any connections that are using a failed link.

NOTE 2 – If an LRM does not contain an SNP that references the same resource, then the binding state is not present.

NOTE 3 – The LRM assigns allocated SNPs and resource capacity to a connection. These assignments are not visible to the TAP.

NOTE 4 – If the resource administrative state is changed from unlocked to shutting down, then the TAP must change the binding state of any allocated SNPs that are referencing that resource to shutting down.

NOTE 5 – If the resource administrative state is set to locked, then the TAP must set the SNP binding state to busy.

NOTE 6 – This combination occurs when the resource is allocated to the management plane or when the resource is being withdrawn from the control plane. The management plane will operate directly on the transport plane resources. Changes to the ITU-T X.731 states will not be visible to the TAP during this time.

7.3.8.3 Adding/removing resources from a control plane

The resource administrative and usage states may be used to control the addition or withdrawal of a resource from the control plane. This is illustrated in the administrative state transition diagram in Figure 7.23.

This is visible to TAP and is the ITU-T X.731 resource state.

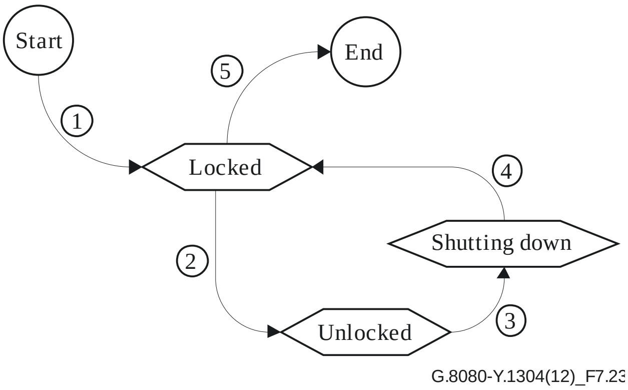


Figure 7.23 – Resource administrative state

Transition descriptions:

Transition	Description	Interface
1	Resource is made visible to the TAP	MI, usage update
2	TAP is permitted to use the resource The usage state is set to busy	Resource state
3	The resource is being withdrawn	Resource state
4	TAP has set the binding state of all SNPs that reference the resource to busy	Usage update
5	Resource is withdrawn from the control plane: If the withdrawal is permanent, then the TAP is instructed via the MI to delete all SNPs that referenced the resource.	Resource state

The MI, or management interface, for TAP is any of the management interfaces assumed for any ASON component as shown in Figure 7.2.

7.3.8.4 SNPx binding state transitions per LRM

Figure 7.24 shows the SNP binding state held by an LRM. This is the view that the TAP provides to each LRM based on the ITU-T X.731 state of the resources. Operations on the TAP affect each LRMs' SNP binding state view.

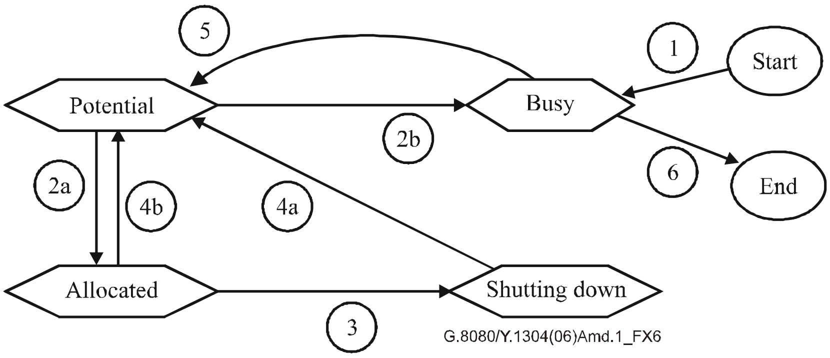


Figure 7.24 – SNP binding state in LRM

Transition descriptions:

Transition	Description	Interface
1	TAP adds SNP in the scope of the LRM	Add SNP
2a	TAP allocates resource to an LRM	SNP binding state; SNP operational state
2b	TAP sets the SNP binding state to busy when: a) the TAP has allocated the resource to another LRM; or b) the administrative state of the resource has been set to shutting down.	SNP binding state
3	TAP requests return of a resource	SNP binding state
4a	LRM is no longer using the resource TAP modifies states to potential	Release SNP

Transition	Description	Interface
4b	LRM is no longer using the resource TAP modifies states to potential	Release SNP
5	TAP moves resource to potential since it is: a) no longer allocated; or; b) the administrative state has been set to unlocked.	SNP binding state
6	SNP is removed from the scope of the LRM.	Withdraw SNP

7.3.8.5 TAP component interfaces

Table 12 – Termination and adaptation performer (TAP) component interface

Input interface	Basic input parameters	Basic return parameters
Operational state	Enabled, disabled	Confirm
Administrative state	Locked, unlocked; shutting down	Confirm for locked, Unlocked. User quit for shutting down
SNP id assigned/unassigned (packet switched only)	SNP id (From LRM) CIR and EIR	
Capacity change request	List of SNP ids CIR and EIR (packet switched only)	Link configuration

Output interface	Basic output parameters	Basic return parameters
Control	Hardware specific	Hardware specific
(Link) Configuration	List of SNP ids CIR and EIR, capacity assignment policy (packet switched only)	confirm
Capacity change (packet switched only)	CIR EIR	Confirm
SNPx binding state	Busy, Potential, Allocated, Shutting down	Resource released (in response to the shutting down state)
SNPx operational state	Enabled, disabled	Confirm
Add SNP	List of SNP identifiers	Confirm
Withdraw SNP	List of SNP identifiers	Confirm
Usage update	New user, User quit	Usage state (idle, busy)

Operational state: This interface accepts resource state information from transport plane adaptation and termination functions.

Administrative state: This interface accepts administrative state from the MI.

SNP id assigned/unassigned: This interface receives notification of SNP binding actions from LRM.

Capacity change request: This interface receives requests from LRM to change the capacity of packet resources associated with its assigned SNPs.

Control: This hardware specific interface allows the TAP to communicate with the resources that it controls.

Configuration: This interface allows the TAP to provide the link end configuration information to an LRM.

Capacity change: This interface is used by the TAP to advise the LRM if the capacity of the link has been modified. This interface is only used for packet switching.

SNPx binding state: SNP binding state is sent to an LRM.

SNPx operational state: SNP operational state is sent to an LRM.

Add SNP: This interface is used to inform an LRM of a new SNP.

Withdraw SNP: This interface is used to inform an LRM of the removal of an SNP.

Usage update: This interface provides resource state usage information to transport plane adaptation and termination functions.

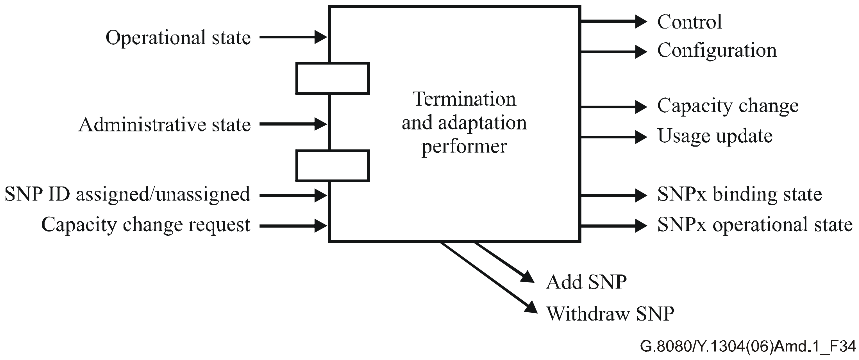


Figure 7.25 – Termination and adaptation performer component

7.3.9 Link discovery process

The generic process of discovery is split into two separate and distinct times and name spaces. The first part takes place entirely in the transport plane name space (CPs and CTPs).

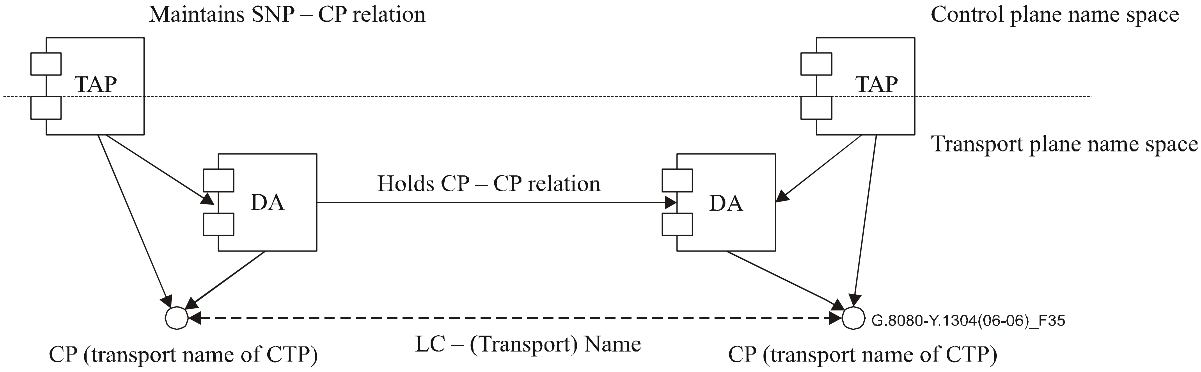

Maintains SNP – CP relation
Control plane name space
TAP
TAP
Transport plane name space
DA
Holds CP – CP relation
DA
G.8080-Y.1304(06-06)_F35
CP (transport name of CTP)
LC – (Transport) Name
CP (transport name of CTP)

Figure 7.26 – Discovery of transport link connections (LC)

The DA operates entirely within the transport name space and is responsible for holding the transport name of the link connection (associated with each CP). This information may be obtained by using transport mechanisms invisible to the control plane name space, by holding previously obtained relation information or by provisioning. The DA assists in an underlying automatic discovery process by cooperatively resolving transport CP names among all the DAs in the network, thus enabling the DAs (or other components) responsible for each end of the transport link connection to communicate about that link connection.

A CP can be assigned to a set of VPNs, including the empty set and the singleton set. This set of VPNs can be represented by an ownership tag. The DA verifies that ownership tag attached to each CP of a link connection is the same.

The second part takes place entirely within the control plane name space (SNPs).

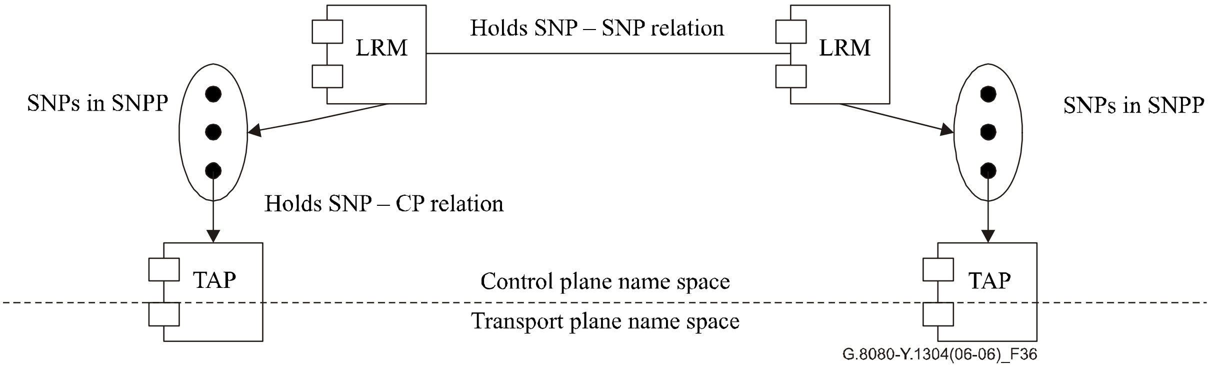


Figure 7.27 – Population of control plane link connections

The link resource manager (LRM) holds the SNP-SNP binding information necessary for the control plane name of the link connection, while the TAP holds the relation between the control plane name (SNP) and the transport plane name (CP) of a resource. This separation allows control plane names to be completely separate from transport plane names, and completely independent of the method used to populate the DAs with those transport names.

In order to assign an SNP-SNP link connection to an SNPP link, it is only necessary for the transport name for the link connection to exist. Thus it is possible to assign link connections to the control plane without the link connection being physically connected. This assignment procedure may be verified by the LRMs exchanging the transport link connection name (i.e., CP-CP name or TCP-TCP name) that corresponds to the SNP.

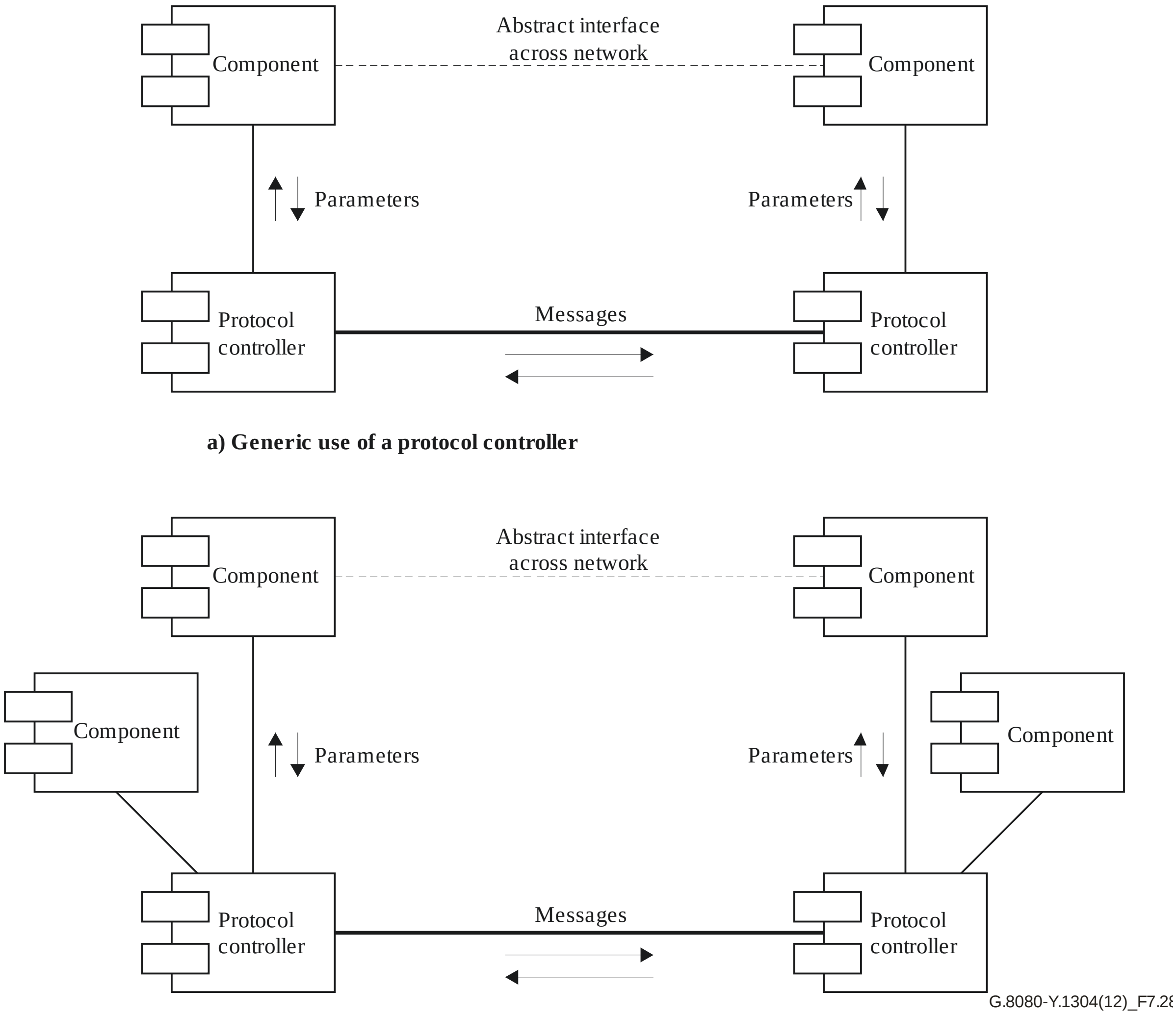
Note that the fully qualified SNPP link name is a control plane name reflecting the structure of transport plane resources.

7.4 Protocol controller (PC) components

The protocol controller provides the function of mapping the parameters of the abstract interfaces of the control components into messages that are carried by a protocol to support interconnection via an interface. Protocol controllers are a subclass of port controllers, and provide all the functions associated with those components. In particular, they report protocol violations to their monitoring ports. They may also perform the role of multiplexing several abstract interfaces into a single protocol instance as shown in Figure 7.28. The details of an individual protocol controller are in the realm of protocol design, though some examples are given in this Recommendation.

The role of a transport protocol controller is to provide authenticated, secure, and reliable transfer of control primitives across the network by means of a defined interface. This permits transactions to be tracked and to ensure expected responses are received, or that an exception is reported to the originator. When security functions are present, the protocol controller will report security violations via its monitoring port.

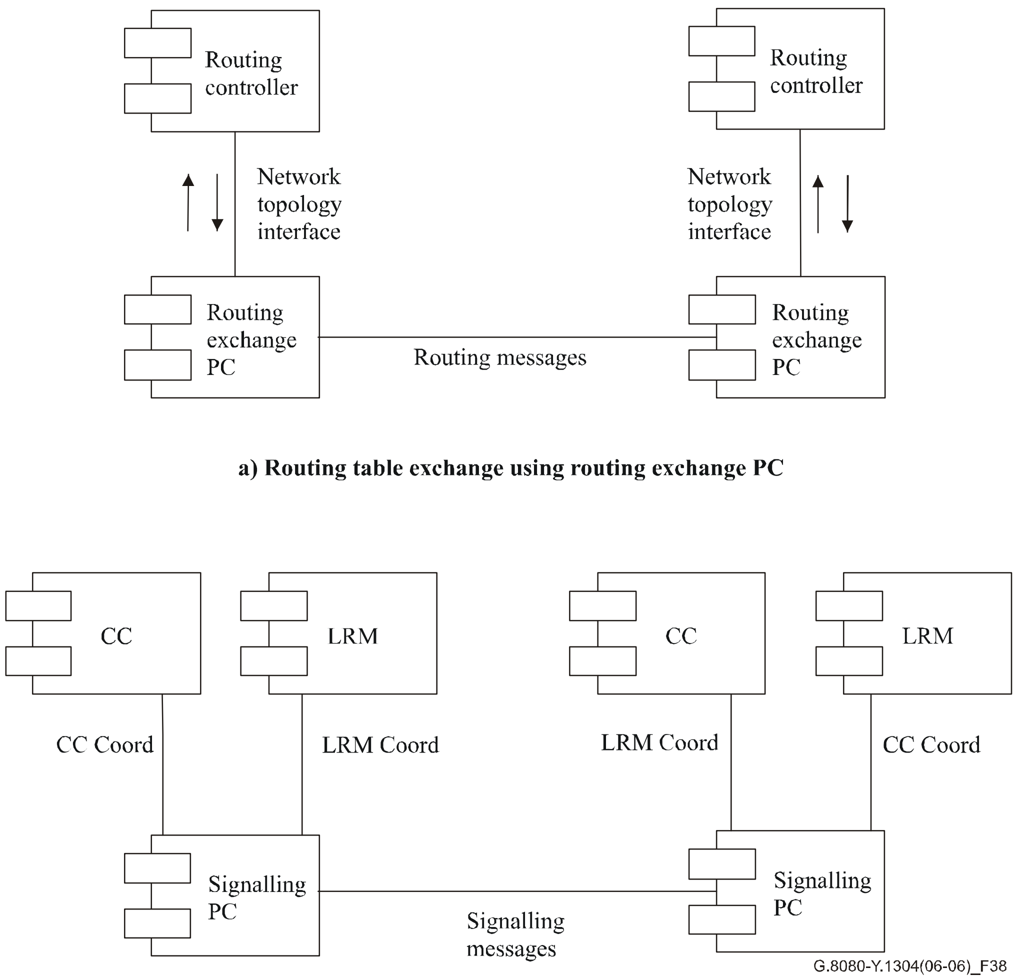
Signalling primitives are passed between the connection controller and the protocol controller, which is semantically transparent to the messaging primitives as this results in external protocol messages and vice versa. Signalling messages are passed between the two protocol controllers. This is illustrated in Figure 7.29.



a) Generic use of a protocol controller

b) Generic multiplexing of different primitive streams into a single protocol

Figure 7.28 – Protocol controller



a) Routing table exchange using routing exchange PC

b) Multiplexing of LRM and CC coordination using signalling PC

Figure 7.29 – Examples of protocol controller use

Examples of protocol controller use is the transfer of the following information:

- Route table update messages via a routing exchange protocol controller (shown in Figure 7.29-a).
- Link resource manager coordination messages (where appropriate as in available bit rate connections) via a link resource manager protocol controller.
- Connection control coordination messages via a connection controller protocol controller, (shown in Figure 7.29-b). Note that the LRM and CC coordination interfaces may be multiplexed over the same protocol controller.

For the route query interface between the connection controller and routing controller, authenticated and secure information is transferred with the appropriate domain scope policy.

8 Reference points

This Recommendation defines various logical interfaces (i.e., reference points) within a typical transport network where signalling/routing information is exchanged. Reference points may be supported by multiple interfaces. These reference points are the UNI, the I-NNI and the E-NNI. It is important to recognize that there will be multiple domains within the ASON and that the UNI and E-NNI in particular will be used for inter-domain control signalling. The following clauses describe the specific functionalities that need to be carried across the various reference points (UNI, I-NNI and E-NNI) and how they differ.

Policy may be applied at the interfaces that support a reference point. The policies applied are dependent on the reference point and functions supported. For example, at the UNI, I-NNI and E-NNI reference points, policy may be applied to call and connection control. In addition, for the I-NNI and E-NNI reference points, policy may be applied to routing.

A reference point represents a collection of services, provided via interfaces on one or more pairs of components. The component interface is independent of the reference point, hence the same interface may be involved with more than one reference point. From the viewpoint of the reference point the components supporting the interface are not visible, hence the interface specification can be treated independently of the component.

The information flows that carry services across the reference point are terminated (or sourced) by components, and multiple flows need not be terminated at the same physical location. These may traverse different sequences of reference points as illustrated in Figure 8.1.

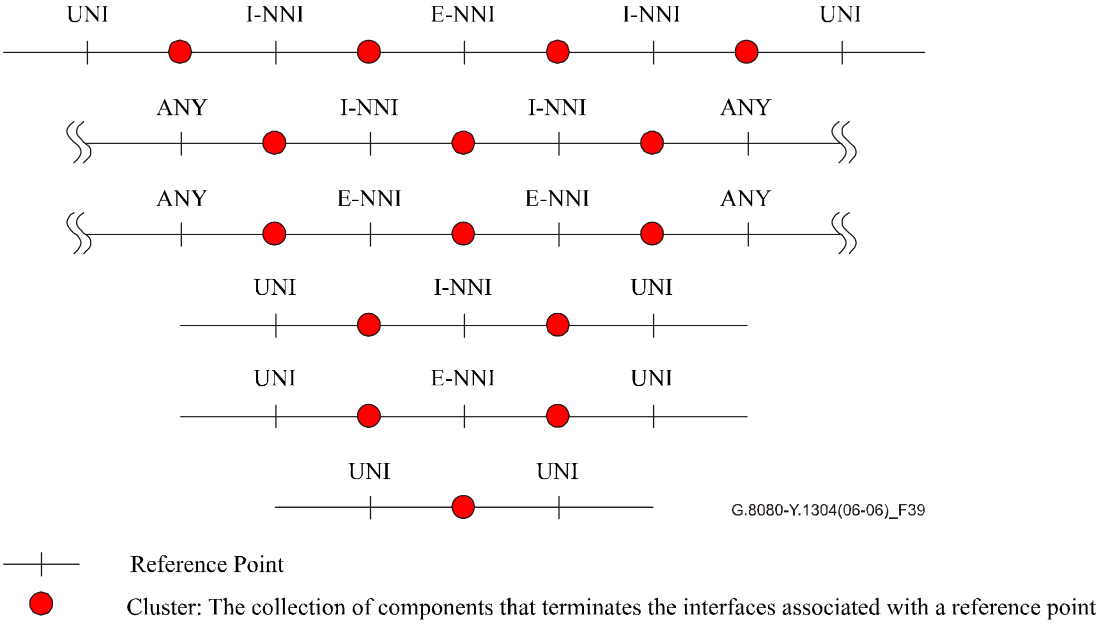


Figure 8.1 – Reference points

8.1 UNI

Information flows expected across the UNI reference point support the following functions:

– call control

– resource discovery

– connection control

– connection selection.

Note, there is no routing function associated with the UNI reference point.

Additional functions such as security and authentication of calls, or enhanced directory services, may be added to this basic set of functions.

The use of the UNI reference point in L1 VPNs is for further study.

8.2 I-NNI

Information flows expected across the I-NNI reference point support the following functions:

– resource discovery

– connection control

– connection selection

– connection routing.

8.3 E-NNI

Information flows expected across the E-NNI reference point support the following functions:

– call control

– resource discovery

– connection control

– connection selection

– connection routing.

Additional functions such as security and authentication of calls, or enhanced directory services may be added to this basic set of functions.

When the E-NNI reference point exists between a VPN customer domain and a VPN in a service provider domain, supplementary services may be supported (see [ITU-T Y.1312]). Examples are:

– VPN user authentication and authorization

– VPN user policy management, including connectivity restrictions

– transparent transfer of control information between VPN users

– VPN participation in the customer routing control domain.

Support for such services is outside the scope of this Recommendation.

8.4 User architecture

The user side will be referred to as the UNI-C (for "client"), and the network side will be referred to as the UNI-N (for "network").

The ITU-T G.8080/Y.1304 UNI transport resource identifier (see clause 10) defines one or more globally unique names to each SNPP link that is part of a UNI. These names are used to identify call destinations. Given that a UNI may contain multiple SNPP links, as in the case of multi-homing, a UNI may therefore have multiple globally unique names for its bearer resources. Note that these names are not user names.

When there are multiple SNPP links that are part of the same UNI, those addresses can be used to discriminate between which SNPP link to use. Factors such as diversity or cost could be used by callers to select the appropriate SNPP link. SNPP links between a common AGC and a network may be in the same UNI if, on the network side, they are within the scope of a common network call controller component.

UNI transport resource identifiers can be used to differentiate between UNIs to a user. When there are multiple UNIs, each has distinct UNI transport resource identifiers and they do not share a common address.

The following describes the UNI-C architecture:

1) There exists a transport entity called an access group container (AGC) that can terminate multiple SNPP links. This entity can contain a set of ITU-T G.805 access groups.

2) An AGC is a single layer entity that contains access groups, LRMs, and TAPs. It is similar to ITU-T G.805 subnetworks except that it is not recursively defined, may or may not be a matrix (it does not have to be specified), and has no defined subnetwork connections. Multiple AGCs from different layers may be coincident in the same equipment.

3) Control plane functions associated with a UNI-C in an AGC are call control (calling/called party call controller), and resource discovery (LRM). Limited connection control and connection selection is present to interact with the connection controller on the UNI-N side. This is because the connection control on the UNI-N has a routing interface whereas connection control on the UNI-C tracks connection acceptance/release from the UNI-N side.

4) Applications that use one or more trails on an AGC are known as "<application name> connection users". They interact directly with ITU-T G.805 access points by presenting and receiving adapted information. For each connection user there may be an "<application name> connection requestor". These entities interact with UNI-Cs to request/release connections. A single connection requestor could obtain connections from one or more UNI-Cs for a related connection user.

5) A user is considered to be multi-homed when there are two or more SNPP links connecting the AGC to the network. There is also a service agreement between the user and the network such that the network offers reliability, diversity, or other service characteristic between connections on different multi-homed SNPP links.

8.5 Inter-layer NCC interactions

8.5.1 NCC to NCC calls

A call may exist between a pair of NCCs in the absence of CCCs. This call is within the same layer. To allow such calls to be requested, a transport resource identifier is assigned to a set of SNPs that reference resources that may be used to support a call. This is analogous to the UNI transport resource identifier associated with transport resources at the UNI.

The NCC to NCC call may be used in at least two cases. First, as a call that is invoked in an interlayer call; and second as another type of boundary between domains at the same layer.

When used in an interlayer call, a client NCC is used to initiate the call between another pair of NCCs in a server layer. As the client layer NCC invokes the server layer call, a domain boundary is crossed. This domain boundary is instituted to provide a policy control point, as well as provide separation of the SNPP as well as transport resource identifiers used in the client and server layers. The transport resource identifiers used for the call request are in the server layer. This is illustrated in Figure 8.2.

A server layer connection from an NCC to NCC call used to support a mapped client CI, has an association with an adaptation. Such a call/connection may exist before the adaptation is actually used.

8.5.2 Name space interactions

UNI transport resource identifiers are defined to be globally unique. The additional transport resource identifiers defined are not required to be part of the UNI transport resource identifier space. Transport resource identifiers associated with NCC to NCC calls may be from separate identifier spaces.

Within a single layer network, there may be independent SNPP identifiers spaces. Connections can be created across these different SNPP identifier spaces due to the fact that RCs understand the mapping of SNPP identifiers between routing levels.

When two SNPP identifier spaces are not mapped via routing, it is allowed to map one SNPP identifier space into the transport resource identifiers associated with the second SNPP identifier space. One purpose for this would be for a business boundary that has no routing exchange. An interlayer boundary is such a case.

This mapping is accessed as an address resolution function that takes as input, an SNPP identifier and returns a transport resource identifier associated with an SNPP in another identifier space. The existing address resolution in this Recommendation is accessed via the directory request output interface of the NCC component. The additional address resolution function would also be accessed by the same output interface. For an interlayer call, a client NCC at the edge of a subnetwork that represents server layer flexibility has two SNPPs in the client layer that it requires a connection for. This is obtained from a route query. The interlayer address resolution function is used by the client layer NCC to obtain the server layer transport resource corresponding to those two client layer SNPPs. Using the two transport resource identifiers, a call is made to the server layer.

In Figure 8.2 below, SNPP-X in the client layer is mapped to transport resource identifier B and SNPP-Y is mapped to transport resource identifier C. An interlayer call can be invoked using the returned transport resource identifier.

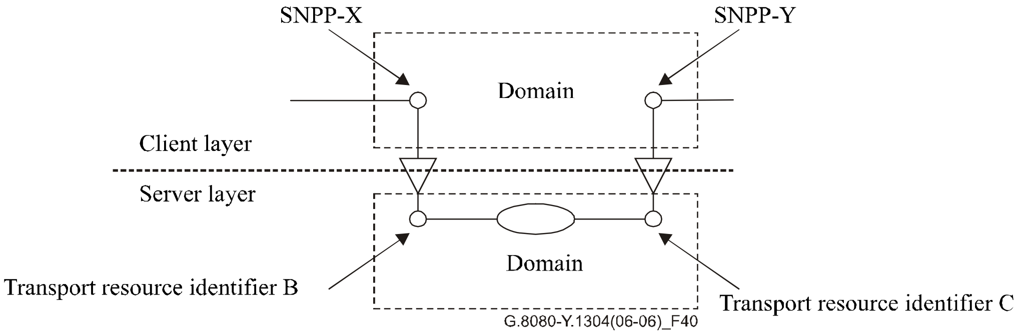


Figure 8.2 – Name space interactions

9 Control plane use of signalling control networks (SCNs)

With the multiple layers capabilities described in clauses 6.5-6.8, the existence of multiple SCNs (DCNs) creates an additional control plane component communication scenario.

It is possible that multiple disjoint SCNs exist between two signalling or routing controllers that are adjacent in their layer. This is illustrated in Figure 9.1. This might happen if two service providers are using a third party in the middle of the network to provide the server layer service. In Figure 9.1, A_{client} and D_{client} are on separate SCNs and do not have any direct connectivity. Messages have to go through an intermediate SCN in order to reach D_{client} from A_{client}. The SCNs may form an even more complex topology, each SCN containing a set of policies regarding which messages are allowed to be carried over its SCN.

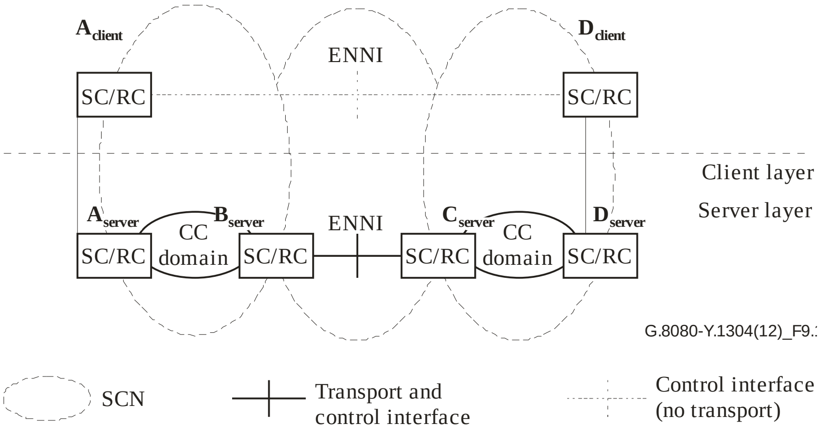


Figure 9.1 – Example of disjoint SCNs in the client layer

10 Identifiers

10.1 Name spaces

There are three separate transport names spaces in the ASON naming syntax:

1) a routing area name space

2) a subnetwork name space

3) a link context name space.

The first two spaces follow the transport subnetwork structure and need not be related. Taken together, they define the topological point where an SNPP is located. The link context name space specifies within the SNPP where the SNP is. It can be used to reflect sub-SNPP structure, and different types of link names.

An SNPP name is a concatenation of:

- One or more nested routing area names.
- An optional subnetwork name within the lowest routing area level. This can only exist if the containing RA names are present.
- One or more nested resource context names.

Using this design, the SNPP name can recurse with routing areas down to the lowest subnetwork and link sub-partitions (SNPP sub-pools). This scheme allows SNPs to be identified at any routing level.

SNP name: An SNP is given an address used for link connection assignment and, in some cases, routing. The SNP name is derived from the SNPP name concatenated with a locally significant SNP index.

An SNPP alias is an alternate SNPP name for the same SNPP link that may be generated from another SNPP name space.

This Recommendation does not specify formats or values for identifiers in instances of ASON name spaces. It is possible for implementations to assign the same value to identifiers in different name spaces (e.g., SNPP and TRI) in order to simplify configuration.

NOTE – The SNPP alias may be generated from the same or different SNPP name space. If present in a routing area, it is available to the RC that is associated with the RA.

10.2 Names and addresses

Names and addresses provide the necessary means of identification for control plane components to cooperatively control resources in a transport network. As defined in [ITU-T G.8081], addresses are location dependent and names are not. It is important to note that a given identifier may serve as a name in one context and as an address in another context. For example, consider the routing area A shown in Figure 6.9. Inside routing area A, the local SNP ID is an address while the interface SNP ID is a name. However, outside the routing area, the interface SNP ID may be an address in the context of some other routing area (not shown in Figure 6.9).

Names and addresses are needed for various entities in the ASON control plane, as described below:

E-NNI transport resource

The E-NNI SNPP link may be assigned a name for the network call controllers to specify E-NNIs. These names must be globally unique and are assigned by the ASON network. Multiple names may be assigned to the SNPP link. An alias may exist for a set of E-NNI transport resource identifiers, for example when a call must traverse multiple domains and the user can specify a transit domain, but not a specific E-NNI.

When the E-NNI reference point exists between a VPN customer domain and a VPN in a service provider domain, the E-NNI transport resource identifier can be unique among all other E-NNI SNPP links assigned to the VPN and not necessarily globally unique. It can be assigned by the VPN customer or by the ASON network.

UNI transport resource: The UNI SNPP link requires a name for the calling party call controller and network call controller to specify destinations. These names must be globally unique and are assigned by the ASON network. Multiple names may be assigned to the SNPP link. UNI transport resource identifiers may be in a 1:N or N:1 relationship with SNPP links. This enables a calling/called party to associate different applications with specific addresses over a common link. An alias may exist for a set of UNI transport resource identifiers.

Routing controller protocol controller: The routing controller protocol controller (RCPC) requires a DCN address for exchanging routing protocol messages with peer RCPCs. The routing controller protocol controller also must have a name that identifies it to peers for the purpose of maintaining routing protocol relationships.

Routing controller: The routing controller requires a name that identifies it as the source for topology information that it generates and shares with other RCs.

Network call controller protocol controller: The network call controller requires a DCN address for exchanging call signalling messages. The network call controller protocol controller also must have a name that identifies it to peers for the purpose of maintaining call signalling relationships.

Connection controller protocol controller: This requires a DCN address for exchanging connection signalling messages. These addresses are unique within the scope of an administrative domain. The connection controller protocol controller may have a name that identifies it to peers for the purpose of maintaining connection signalling relationships.

Calling/called party call controller protocol controller: This requires a DCN address for exchanging call signalling messages. The calling/called party call controller protocol controller also must have a name that identifies it to peers for the purpose of maintaining call signalling relationships.

Subnetwork: This is given an address representing the collection of all SNPs on that subnetwork, which is used for connection routing. The address is unique within the scope of an administrative domain.

Routing area: This is given an address representing the collection of all SNPPs on that routing area, which is used for connection routing.

10.3 Relationships between identifiers

Transport resources may be acted upon by both control plane components and management plane components (see clause 5.2). As a result, transport resources will be referenced by context specific identifiers. To allow the applications in these different contexts to exchange information about the common resource, a mapping is required between transport resource identifiers used in a management plane context and those used in the control plane context.

Within the set of identifiers used by the control plane for transport plane resources, mappings are also required. For example, mapping a transport resource identifier to one or more SNPPs.

Details of any mappings between identifiers are outside of the scope of this Recommendation and are described in the appropriate G.771x.x series of ITU-T Recommendations.

11 Connection availability enhancement techniques

This clause describes the strategies that can be used to maintain the integrity of an existing call in the event of failures within the transport network.

[ITU-T G.805] describes transport network availability enhancement techniques. The terms "Protection" (replacement of a failed resource with a pre-assigned standby) and "Restoration" (replacement of a failed resource by re-routing using spare capacity) are used to classify these techniques. In general, protection actions complete in the tens of millisecond range, while restoration actions normally complete in times ranging from hundreds of milliseconds to up to a few seconds.

The ASON control plane provides a network operator with the ability to offer a user calls with a selectable class of service (CoS) (e.g., availability, duration of interruptions, errored seconds, etc.). Protection and restoration are mechanisms (used by the network) to support the CoS requested by the user. The selection of the survivability mechanism (protection, restoration or none) for a particular connection that supports a call will be based on: the policy of the network operator, the topology of the network and the capability of the equipment deployed. Different survivability mechanisms may be used on the connections that are concatenated to provide a call. If a call transits the network of more than one operator then each network should be responsible for the survivability of the transit connections. Connection requests at the UNI or E-NNI will contain only the requested CoS, not an explicit protection or restoration type.

The protection or restoration of a connection may be invoked or temporarily disabled by a command from the management plane. These commands may be used to allow scheduled maintenance activities to be performed. They may also be used to override the automatic operations under some exceptional failure conditions.

The protection or restoration mechanism should:

- be independent of, and support any, client type (e.g., IP, ATM, SDH, Ethernet);
- provide scalability to accommodate a catastrophic failure in a server layer, such as a fibre cable cut, which impacts a large number client layer connections that need to be restored simultaneously and rapidly.
- utilize a robust and efficient signalling mechanism, which remains functional even after a failure in the transport or signalling network;
- not rely on functions which are non-time critical to initiate protection or restoration actions. Therefore, consideration should be given to protection or restoration schemes that do not depend on fault localization.

The description of how protection and restoration capabilities are used by the transport, control and management planes of an ASON enabled network is for further study.

11.1 Protection

Protection is a mechanism for enhancing availability of a connection through the use of additional, assigned capacity. Once capacity is assigned for protection purposes there is no re-routing and the SNPs allocated at intermediate points to support the protection capacity do not change as a result of a protection event. The control plane, specifically the connection control component, is responsible for the creation of a connection. This includes creating both a working connection and a protection connection, or providing connection specific configuration information for a protection scheme. For transport plane protection, the configuration of protection is made under the direction of the management plane. For control plane protection, the configuration of protection is under the direction of the control plane rather than the management plane.

Control plane protection occurs between the source connection controller and the destination connection controller of a control plane protection domain, where the source and destination are defined in relation to the connection. The operation of the protection mechanism is coordinated between the source and destination. In the event of a failure, the protection does not involve re-routing or additional connection set-up at intermediate connection controllers, only the source and destination connection controllers are involved. This represents the main difference between protection and restoration.

11.2 Restoration

The restoration of a call is the replacement of a failed connection by re-routing the call using spare capacity. In contrast to protection, some, or all, of the SNPs used to support the connection may be changed during a restoration event. Control plane restoration occurs in relation to re-routing domains. A re-routing domain is a group of call and connection controllers that share control of domain-based re-routing. The components at the edges of the re-routing domains coordinate domain-based re-routing operations for all calls/connections that traverse the re-routing domain. A re-routing domain must be entirely contained within a routing control domain or area. A routing control domain may fully contain several re-routing domains. The network resources associated with a re-routing domain must therefore be contained entirely within a routing area. Where a call/connection is re-routed inside a re-routing domain, the domain-based re-routing operation takes place between the edges of the re-routing domain and is entirely contained within it.

The activation of a re-routing service is negotiated as part of the initial call establishment phase. For a single domain, an intra-domain re-routing service is negotiated between the source (connection and call controllers) and destination (connection and call controller) components within the re-routing domain. Requests for an intra-domain re-routing service do not cross the domain boundary.

Where multiple re-routing domains are involved, the edge components of each re-routing domain negotiate the activation of the re-routing services across the re-routing domain for each call. Once the call has been established, each of the re-routing domains in the path of the call have knowledge as to which re-routing services are activated for the call. As for the case of a single re-routing domain, once the call has been established the re-routing services cannot be renegotiated. This negotiation also allows the components associated with both the calling and called parties to request a re-routing service. In this case, the service is referred to as an inter-domain service because the requests are passed across re-routing domain boundaries. Although a re-routing service can be requested on an end-to-end basis, the service is performed on a per re-routing domain basis (that is between the source and destination components within each re-routing domain traversed by the call).

During the negotiation of the re-routing services, the edge components of a re-routing domain exchange their re-routing capabilities and the request for a re-routing service can only be supported if the service is available in both the source and destination at the edge of the re-routing domain.

A hard re-routing service offers a failure recovery mechanism for calls and is always in response to a failure event. When a link or a network element fails in a re-routing domain, the call is cleared to the edges of the re-routing domain. For a hard re-routing service that has been activated for that call, the source blocks the call release and attempts to create an alternative connection segment to the destination at the edge of the re-routing domain. This alternative connection is the re-routing connection. The destination at the edge of the re-routing domain also blocks the release of the call and waits for the source at the edge of the re-routing domain to create the re-routing connection. In hard re-routing, the original connection segment is released prior to the creation of an alternative connection segment. This is known as break-before-make. An example of hard re-routing is provided in Figure 11.1. In this example, the routing control domain is associated with a single routing area and a single re-routing domain. The call is re-routed between the source and destination nodes and the components associated with them.

Soft re-routing service is a mechanism for the re-routing of a connection for administrative purposes (e.g., path optimization, network maintenance, and planned engineering works). When a re-routing operation is triggered (generally via a request from the management plane) and sent to the location of the re-routing components, the re-routing components establish a re-routing connection to the location of the rendezvous components. Once the re-routing connection is created, the re-routing components use the re-routing connection and delete the initial connection. This is known as make-before-break.

During a soft re-routing procedure, a failure may occur on the initial connection. In this case, the hard re-routing operation pre-empts the soft re-routing operation and the source and destination components within the re-routing domain proceed according to the hard re-routing process.

If revertive behaviour is required (i.e., the call must be restored to the original connections when the failure has been repaired), network call controllers must not release the original (failed) connections. The network call controllers must continue monitoring the original connections, and when the failure is repaired, the call is restored to the original connections.

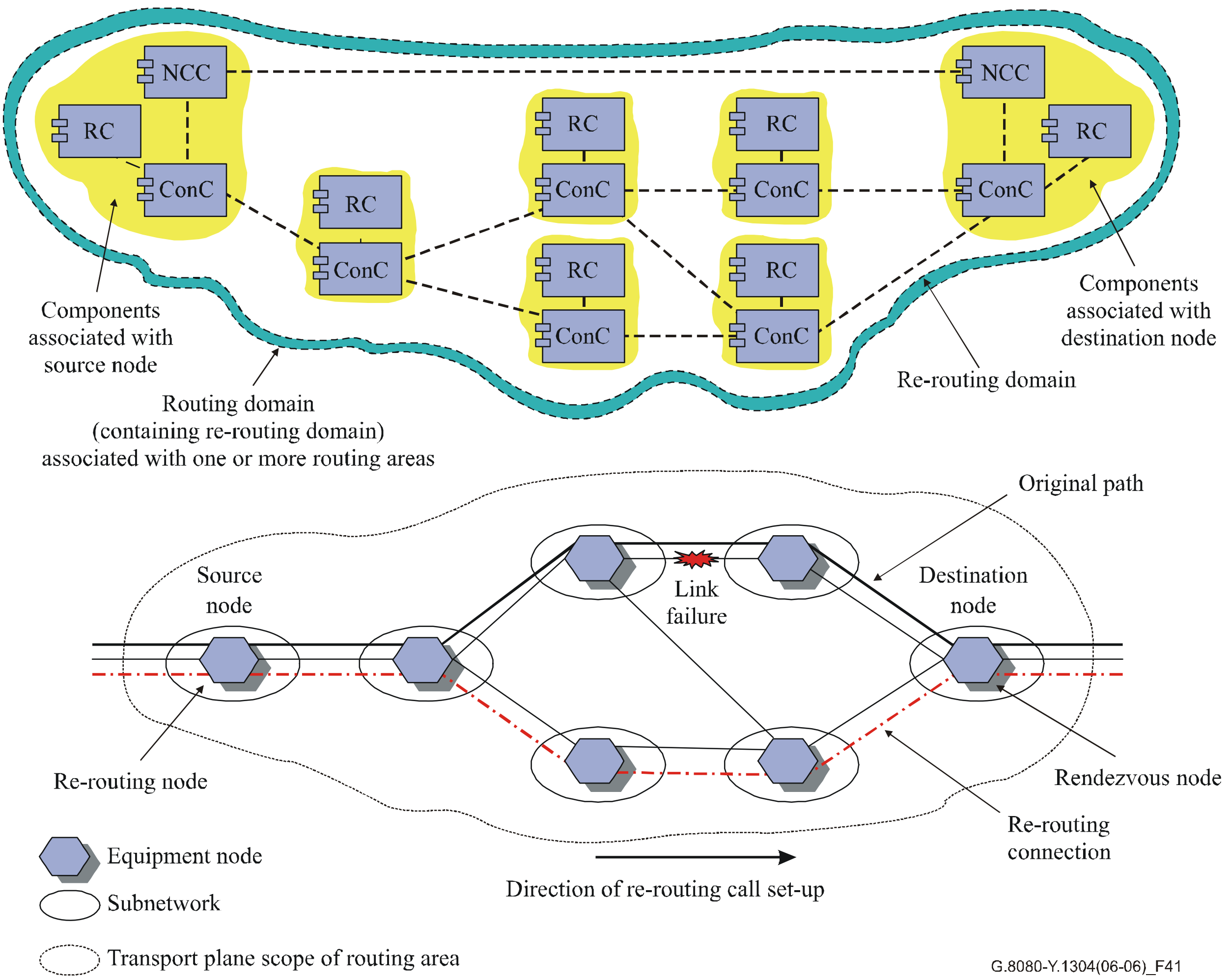

NCC
RC
ConC
RC
ConC
RC
ConC
RC
ConC
RC
ConC
RC
ConC
NCC
RC
ConC
Components associated with source node
Routing domain (containing re-routing domain) associated with one or more routing areas
Components associated with destination node
Re-routing domain
Original path
Source node
Link failure
Destination node
Re-routing node
Rendezvous node
Re-routing connection
Equipment node
Subnetwork
Transport plane scope of routing area
Direction of re-routing call set-up
G.8080-Y.1304(06-06)_F41

Figure 11.1 – Example of hard re-routing

11.2.1 Re-routing in response to failure

11.2.1.1 Intra-domain failures

Any failures within a re-routing domain should result in a re-routing (restoration) action within that domain such that any downstream domains only observe a momentary incoming signal failure (or previous section fail). The connections supporting the call must continue to use the same source (ingress) and destination (egress) gateways nodes in the re-routing domain.

11.2.1.2 Inter-domain failures

Two failure cases must be considered, failure of a link between two gateway network elements in different re-routing domains and failure of inter-domain gateway network elements.

11.2.1.3 Link failure between adjacent gateway network elements

When a failure occurs outside of the re-routing domains (e.g., the link between gateway network elements in different re-routing domains A and B in Figure 11.2-a) no re-routing operation can be performed. In this case, alternative protection mechanisms may be employed between the domains.

Figure 11.2-b shows the example with two links between domain A and domain B. The path selection function at the A (originating) end of the call must select a link between domains with the appropriate level of protection. The simplest method of providing protection in this scenario is via a protection mechanism that is pre-established (e.g., in a server layer network. Such a scheme is transparent to the connections that run over the top of it). If the protected link fails, the link protection scheme will initiate the protection operation. In this case, the call is still routed over the same ingress and egress gateway network elements of the adjacent domains and the failure recovery is confined to the inter-domain link.

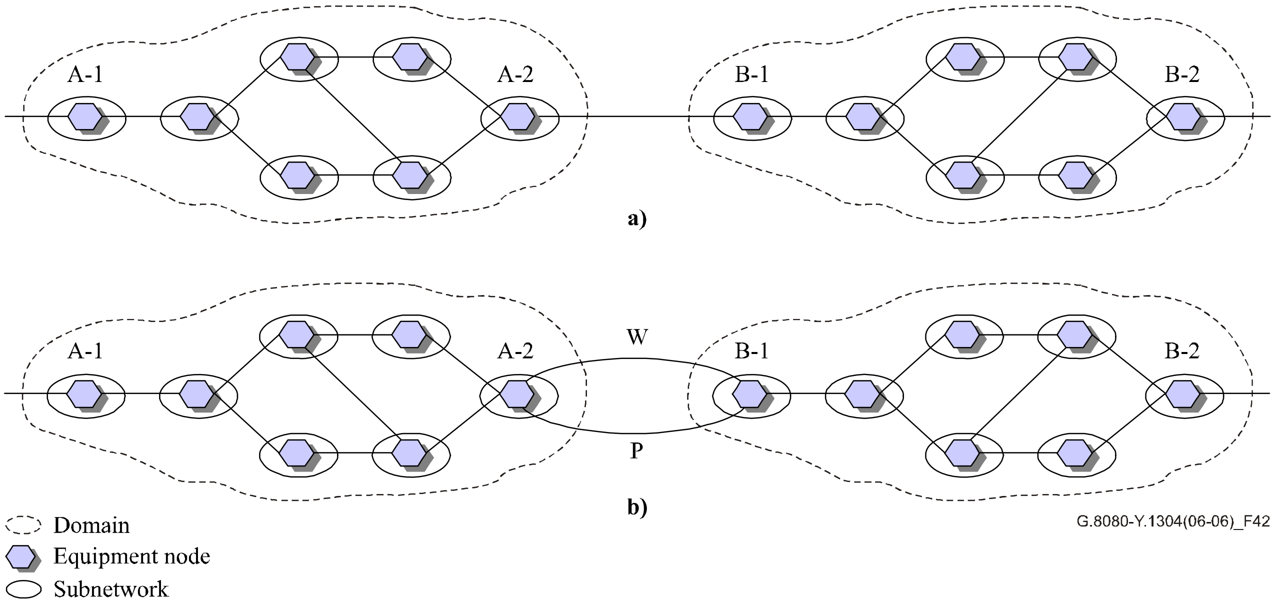


Figure 11.2 – Link failure scenarios

11.2.1.4 Gateway network element failure

This case is shown in Figure 11.3. To recover a call when B-1 fails a different gateway node, B-3, must be used for domain B. In general this will also require the use of a different gateway in domain A, in this case A-3. In response to the failure of gateway NE B-1 (detected by gateway NE A-2) the source node in domain A, A-1, must issue a request for a new connection to support the call. The indication to this node must indicate that re-routing within domain A between A-1 and A-2 is to be avoided, and that a new route and path to B-2 is required. This can be considered as re-routing in a larger domain, C, which occurs only if re-routing in A or B cannot recover the connection.

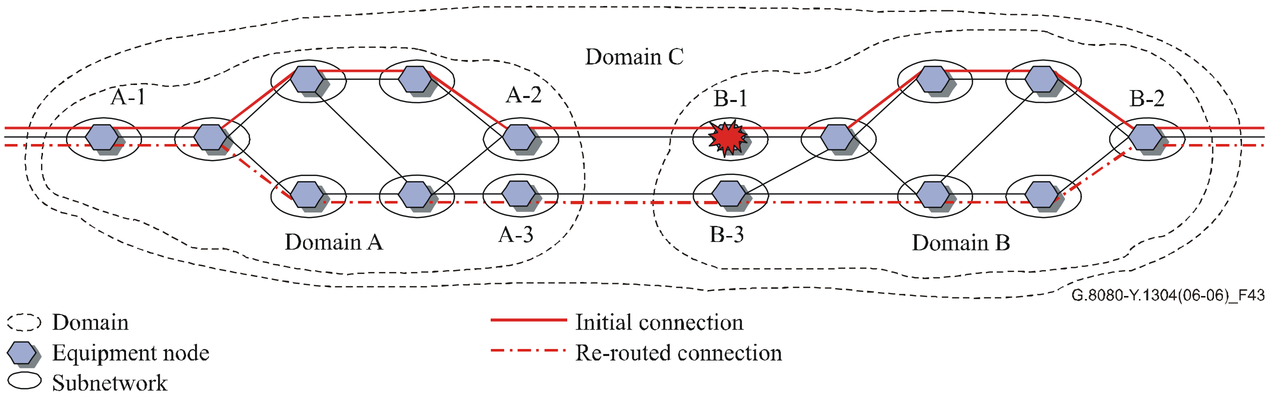


Figure 11.3 – Re-routing in event of a gateway network element failure

11.3 Nested routing domains

CP protection and restoration domains are types of routing control domains. As such, they inherit the containment property of control domains. Coordination between the actions taken by two domains in a containment relationship to a resource failure is a matter of policy.

12 Resilience

Resilience refers to the ability of the control plane to continue operating under failure conditions. Operation of the control plane depends upon elements of the data communication network (DCN), the transport plane, the management plane and the internal components of the control plane itself (refer to Figure 5.1). Additional information is provided in Appendix I.

12.1 Principles of control and transport plane interactions

The following principles are used for control and transport plane interactions when communications become available between the two planes.

1) The control plane relies on the transport plane for information about transport plane resources.

2) Consistency between the control plane view and the corresponding transport network element is established first (vertical consistency).

3) Once local consistency is established, horizontal consistency is attempted. Here, control plane components synchronize with their adjacent components. This is used to re-establish a consistent view of routing, call, and connection state.

Another principle of control and transport plane interaction is that:

4) existing connections in the transport plane are not altered if the control plane fails and/or recovers. Control plane components are therefore dependent on SNC state.

For resiliency, the transport plane resource and SNC state information should be maintained in non-volatile store. Further, some information about the control plane use of the SNC should be stored. This includes whether the SNC was created by connection management and how it was used. For example, which end of the SNC is towards the head end of the whole connection. At a given node, the control plane must ensure it has resource and SNC state information that is consistent with the resource and SNC state information maintained by the transport NE. If not, the control components responsible for that node must:

- advertise zero bandwidth available to adjacent nodes to ensure there will be no network requests to route a new connection through that node;
- not perform any connection changes (e.g., releases).

SNC state is the most important information to recover first because it is the basis of connections that provide service to end users. This follows the principle above. During recovery, the control plane reconstructs the call and connection state corresponding to existing connections. For example, routing will need to disseminate correct SNP information after it is synchronized by the local control plane components (LRM).

The control plane re-establishment of information consistency with the transport NE should occur in the following sequence:

- the link resource manager synchronizes with the transport NE state information
- the connection controller then synchronizes with the link resource manager
- the network call controller then synchronizes with the connection controller.

Following the re-establishment of local state consistency, the control plane must then ensure SNC state information consistency with adjacent nodes, as discussed in principle 3 above, prior to participating in control plane connection set-up or release requests.

12.2 Principles of protocol controller communication

When communication between protocol controllers is disrupted, existing calls and their connections are not altered. The management plane may be notified if the failure persists and requires operator intervention (for example, to release a call).

A failure of the DCN may affect one or more protocol controller to protocol controller communication sessions. The protocol controller associated with each signalling channel must detect and alarm a signalling channel failure.

When a protocol controller to protocol controller communication session recovers, state re-synchronization between the protocol controllers should be performed.

Failure of a protocol controller is handled similar to a failure of a protocol controller to protocol controller session.

12.3 Control and management plane interactions

If management plane functions become unavailable, various control functions may be impaired. When management plane functions become available, the control plane components may need to report to the management plane actions that they took while the management plane was unavailable (e.g., call records).

Annex A

Connection services

(This annex forms an integral part of this Recommendation.)

The control of connectivity is essential to the operation of a transport network. The transport network itself can be described as a set of layer networks, each acting as a connecting function whereby associations are created and removed between the inputs and outputs of the function. These associations are referred to as connections. Three types of connection establishment are defined:

1) **Permanent connection**: This form of connection is established by provisioning every network element along the path with the required information to establish an end-to-end connection. Provisioning is provided either by means of management systems or by manual intervention. Where a network management system is used, access to a database model of the network is normally required first, to establish the most suitable route, and then to send commands to the network elements that support the connection. This type of connection is referred to as a hard permanent connection. See Figure A.1. Note, permanent connections are not described by the ITU-T G.8080/Y.1304 architecture.

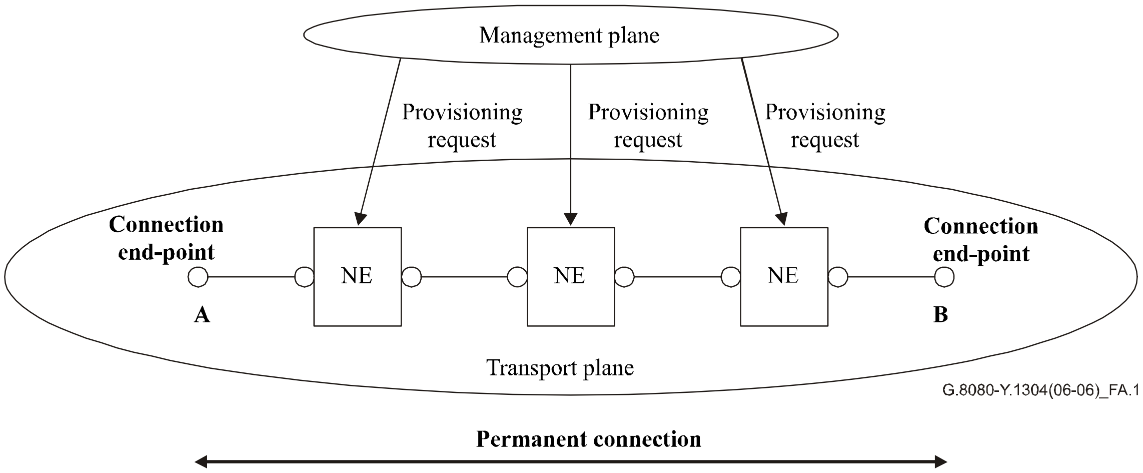


Figure A.1 – Example of end-to-end transport connection set-up using provisioning via management plane

2) **Switched connection (SC)**: This form of connection is established on demand by the communicating end-points within the control plane using a dynamic protocol message exchange in the form of signalling messages. These messages flow across either the I-NNI or E-NNI within the control plane. This type of connection is referred to as a switched connection. Such connections require network naming and addressing schemes and control plane protocols. See Figure A.2.

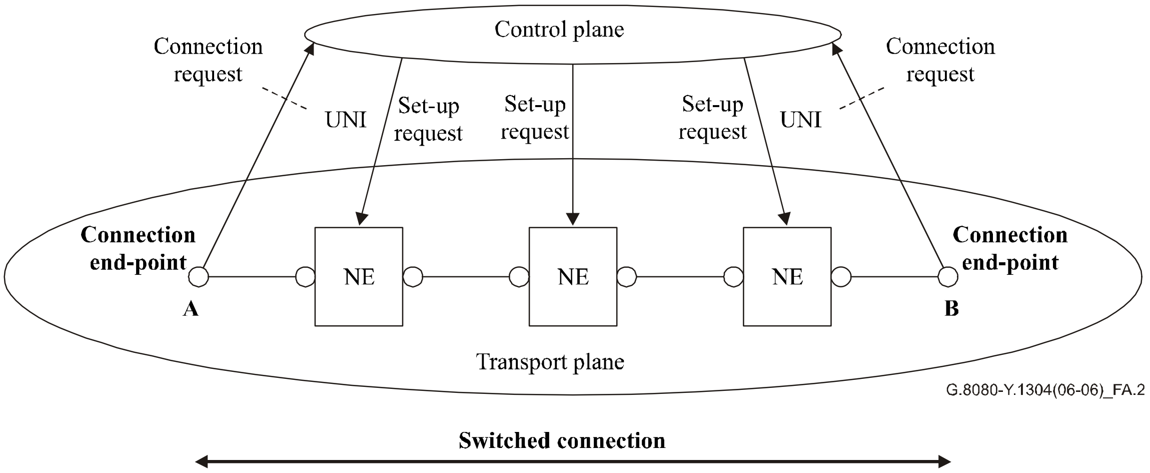


Figure A.2 – Example of end-to-end transport connection set-up using control plane signalling (a switched connection from A to B)

3) **Soft permanent connection (SPC)**: This form of connection establishment exists whereby a network provides a permanent connection at the edge of the network and utilizes a switched connection within the network to provide end-to-end connections between the permanent connections at the network edges. Connections are established via network generated signalling and routing protocols. The establishment of such connections is dependent upon the definition of an NNI. Provisioning is therefore only required on the edge connections. There is no defined UNI. This type of network connection is known as a soft permanent connection (SPC). From the perspective of the end-points a soft permanent connection appears no different than a provisioned, management controlled, permanent connection. See Figure A.3.

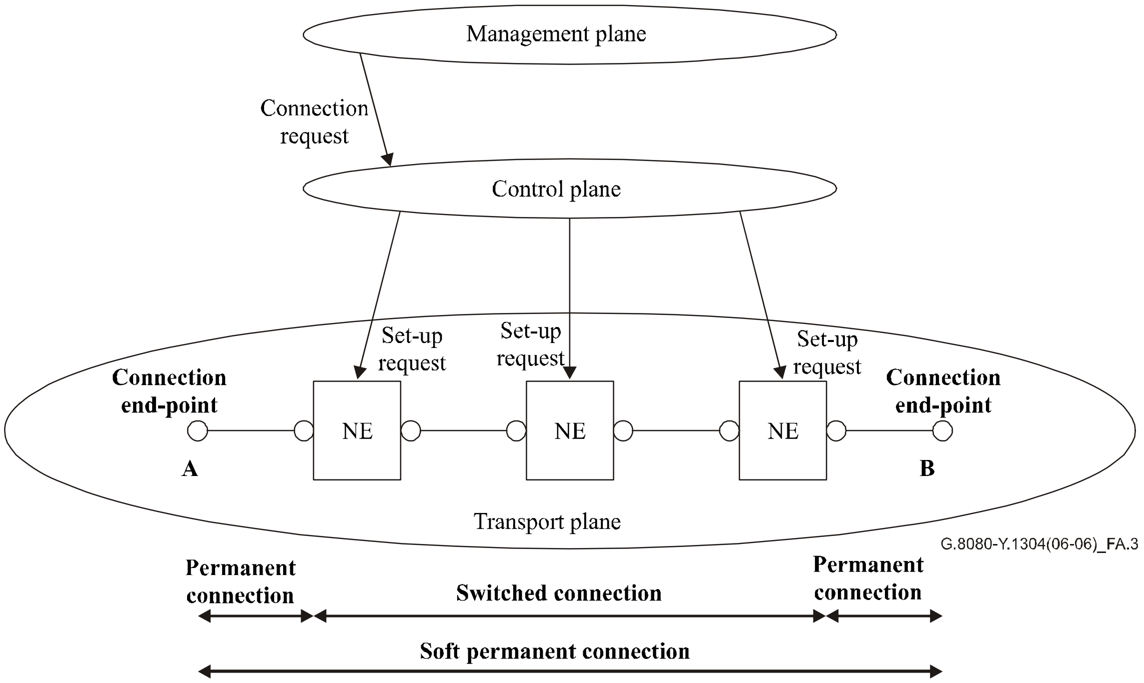


Figure A.3 – Example of end-to-end transport connection set-up as soft permanent connection (SPC)

The most significant difference between the three methods above is the party that sets up the connection. In the case of provisioning, connection set-up is the responsibility of the network operator, whilst in the signalled case, connection set-up may also be the responsibility of the end user. Additionally, third party signalling should be supported across a UNI.

NOTE 1 – The type of connection may have impact on future billing systems.

The control plane shall support either a switched connection (SC) or soft permanent connection (SPC) of the basic connection capability in the transport network. These connection capability types are defined below:

- unidirectional point-to-point connection
- bidirectional point-to-point connection
- unidirectional point-to-multipoint connection.

NOTE 2 – A further connection type can be considered, namely an asymmetric connection. This may be constructed either as two unidirectional point-to-point connections, having different properties in each direction, or as a special case of bidirectional connection.

The function of a UNI is to pass signalling messages directly to the network control plane entity. Alternatively, where a network operator already has extensive management systems in place that provide planning assignment and auto-configuration, these signalling messages may be passed directly to service management and network management system agents to effect connection set-up. Such an application will allow near real time automated service provision from the existing management platforms.

Appendix I

Resilience relationships

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

Resilience refers to the ability of the control plane to continue operating under failure conditions. Operation of the control plane depends upon elements of the data communication network (DCN), the transport plane, the management plane and the internal components of the control plane itself (refer to Figure 5.1). The following clauses identify the control plane dependencies on those areas. The desired degree of control plane resiliency can then be engineered by providing appropriate redundancy for the dependent functions.

I.1 Control plane – DCN relationships

The control plane relies on the DCN for the transfer of signalling messages over some or all of the following interfaces (refer to Figure I.1): UNI, NNI, NMI. The impact of a signalling channel failure on the operation of the control plane will be examined for each of the protocol controllers associated with each interface.

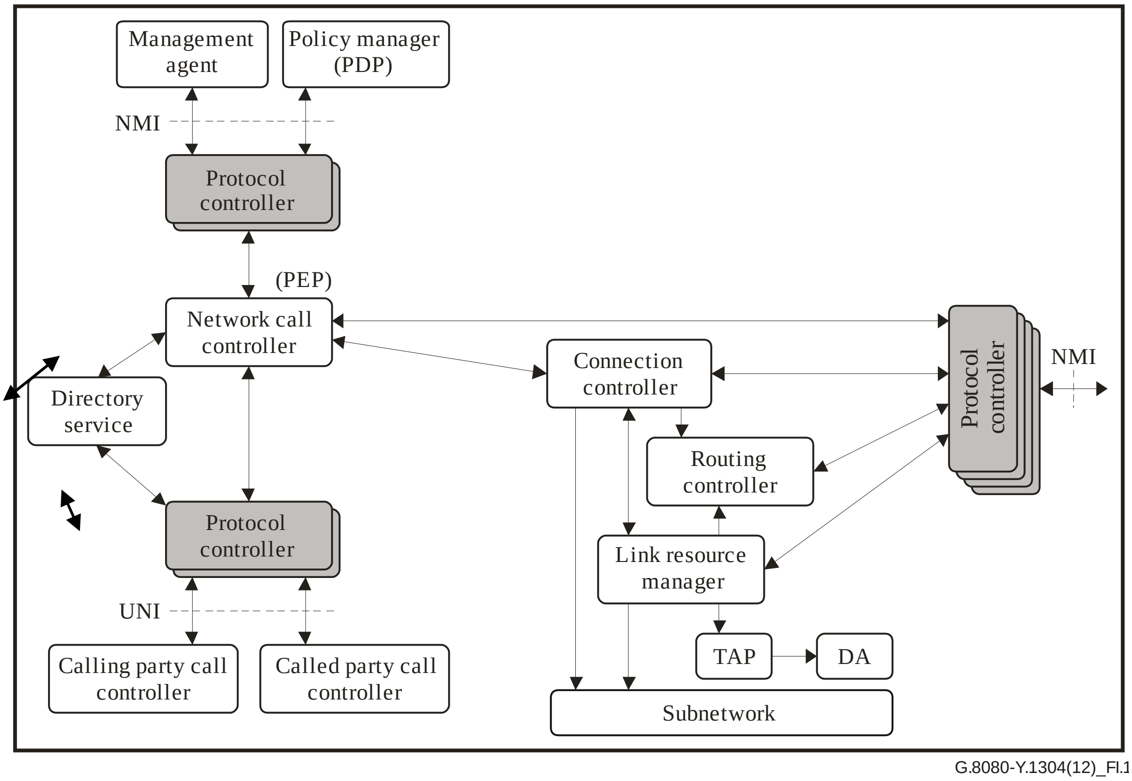


Figure I.1 – Control plane components (an interpretation)

I.1.1 UNI

There are potentially two separate protocol controllers handling the signalling sessions over the UNI: one for the calling party call controller link and one for the called party call controller link.

I.1.1.1 Failure case

A failure of the signalling session supporting the UNI for the calling party call controller link will result in the loss of the call request/call release control flows.

A failure of the signalling session supporting the UNI for the called party call controller link will result in the loss of the call request/call indication control flows.

A failure of either of the UNI-related signalling session impacts the network call controller function.

In all cases above, existing calls and their connections are not altered. The management plane may be notified if the failure persists and requires operator intervention (for example, to release a call).

I.1.1.2 Recovery case

When the signalling channel recovers, state re-synchronization between the client call controllers and the network call controller, and the connection controllers over the UNI, should be performed.

I.1.2 NNI

There are potentially four separate protocol controllers handling the signalling sessions over the NNI: one for the network call controller link, one for the connection controller link, one for the routing controller link and one for the link resource manager link.

I.1.2.1 Failure case

A failure of the signalling session supporting the NNI for the network call controller link will result in the loss of the network call controller coordination control flows. Call set-up or release will not be possible, but there is no impact on connection set-up or release.

A failure of the signalling session supporting the NNI for the connection controller link will result in the loss of the connection controller coordination and connection request/call release control flows. Connection set-up or release will not be possible. Further, if call control is piggybacked on connection control, no call set-up or release will be possible either.

A failure of the signalling session supporting the NNI for the routing controller link will result in the loss of the network/local topology control flows.

A failure of the signalling session supporting the NNI for the link resource manager link will result in the loss of the SNP negotiation/release control flows.

A failure of the link resource manager signalling session impacts the routing controller function and the connection controller function. A failure of the routing controller signalling session impacts the connection controller function. A failure of the connection controller signalling session impacts the network call controller function.

In all cases above, existing calls and their connections are not altered. The management plane may be notified if the failure persists and requires operator intervention (for example, to release a call).

Note that a failure of the DCN may affect one or more or all of the above signalling sessions simultaneously. The protocol controller associated with each signalling channel must detect and alarm a signalling channel failure.

I.1.2.2 Recovery case

Upon restoral of a previously failed signalling channel, the corresponding protocol controller must ensure all messaging resumes in sequence. Components are responsible for re-establishing state information after protocol controller recovery.

I.2 Control plane – Transport plane relationships

This clause considers only those transport plane failures that affect the ability of the control plane to perform its functions, for example when an LRM cannot be informed. Transport plane failures, such as port failures, are not within the scope of this Recommendation as it is expected that the control plane is informed of this situation. Information consistency between the two planes is treated in clause 12.1.

I.2.1 Transport plane information – Query

The control plane will query the transport plane under the following scenarios:

– when a connection controller signalling session activates, or re-activates (for example, following the recovery of a data link or transport NE);

– control plane queries about the transport resources;

– as part of transport resource information synchronization (for example, when the control plane recovers following a failure).

I.2.2 Transport plane information – Event driven

The transport plane will inform the control plane on an event basis under the following scenarios:

– failure of a transport resource

– addition/removal of a transport resource.

I.2.2.1 Transport plane protection

Transport plane protection actions, which are successful, are largely transparent to the control plane. The transport plane is only required to notify the control plane of changes in the availability of transport resources.

Transport plane protection attempts, which are unsuccessful, appear to the control plane as connection failures, and may trigger control plane restoration actions, if such functionality is provided. Given that the control plane supports restoration functionality, the following relationships exist.

The routing controller must be informed of the failure of a transport plane link or node and update the network/local topology database accordingly. The routing controller may inform the local connection controller of the faults.

I.2.3 Transport plane dependency on control plane

If the control plane fails, new connection requests that require the use of the failed control plane components cannot be processed. Note, however that the management plane could be used as a fallback to respond to new connection requests. Established connections must not be affected by a control plane failure.

I.3 Control plane – Management plane relationships

The control plane may obtain directory and policy information from the management plane during the call admission control validation process. Failure of the directory or policy servers could result in the failure of connection set-up requests.

Examples of this are:

– At the network call controller (at the calling or called party end), call requests may need to be validated by policy checking.

– When connection controllers request a path from the routing controller, a policy server may need to be consulted.

Call release actions can take place in the control plane if the management plane is not available. A record of these actions must be maintained by the control plane so that when the management plane becomes available, a log can be sent to the management plane or the control plane can be queried for this information.

I.3.1 NMI

All control components have monitor, policy and configuration ports which provide the management view of the control plane components (see clause 7.2.1).

There are potentially two separate protocol controllers/signalling sessions involving management information flows: one for the policy manager session and one for a transport management session. Other protocol controllers may be introduced in the future for other management functions.

I.3.1.1 Failure case

A failure of the signalling session supporting the policy manager link will result in the loss of the policy out control flows.

A failure of the transport management signalling session will result in the loss of FCAPS (fault, configuration, accounting, performance, security) information exchange.

A failure of the policy session impacts the network call controller function. For example, the potential failure of new connection set-up requests during the call admission control validation process requires policy manager access.

I.3.1.2 Recovery case

When management signalling communication is recovered, information stored in the control plane that should be sent to management plane is sent (e.g., call records). Information pending from the management plane to the control plane should be sent (e.g., revised policy or configuration).

I.4 Intra-control plane relationships

The impact of control plane component failures on the operation of the control plane overall will be examined per the component relationship illustrated in Figure I.1. To achieve continuous operation of the control plane under a component failure, the ability to detect a component failure and switch to a redundant component, without loss of messages and state information, is required.

If control plane components are not redundant, then when a failed component recovers, it must re-establish a sufficient view of the transport plane resources in order to be operational.

It is assumed that the communications between components other than protocol controllers (i.e., non-PC communications) is highly reliable. Such communications is likely internal to a control plane node and is implementation specific, thus it is outside the scope of this Recommendation.

I.4.1 Network call controller

The failure of a network call controller will result in the loss of new call set-up requests and existing call release requests.

I.4.2 Connection controller

The failure of a connection controller will result in the loss of new connection set-up requests and existing connection release requests. As call control signalling is often implemented via the connection controller and its protocol controller, a failure of the connection controller may impact the network call controller function (e.g., may not be able to release existing calls).

I.4.3 Routing controller

The failure of a routing controller will result in the loss of new connection set-up requests and loss of topology database synchronization. As the connection controller depends on the routing controller for path selection, a failure of the routing controller impacts the connection controller. Management plane queries for routing information will also be impacted by a routing controller failure.

I.4.4 Link resource manager

The failure of a link resource manager will result in the loss of new connection set-up requests and existing connection release requests, and loss of SNP database synchronization. As the routing controller depends on the link resource manager for transport resource information, the routing controller function is impacted by a link resource manager failure.

I.4.5 Protocol controllers

The failure of any of the protocol controllers has the same effect as the failure of the corresponding DCN signalling sessions as identified above. The failure of an entire control plane node must be detected by the neighbouring nodes NNI protocol controllers.

I.4.6 Intra-control plane information consistency

As discussed in clause 12.1, at a given node, control plane component resource and SNC state information consistency with the local transport NE resource and state information must be established first. Then control plane components must ensure SNC state information consistency with its adjacent control plane components. Any connection differences must be resolved such that no connection fragments remain or misconnections occur. Following the control plane information consistency cross-check, the control plane components are permitted to participate in control plane connection set-up or release requests.

Appendix II

Example of layered call control

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

Figure II.1 illustrates the mapped server case with the inter-layer call model for two Ethernet clients. They attach to a common VC-3 network that does not support Ethernet switching. Suppose that a 40 Mbit/s call is requested over a Gigabit Ethernet UNI. To carry Ethernet CI, a VC-3 connection is created. The decision by the NCC_{MAC} to make a call to the corresponding $\text{NCC}_{\text{VC-3}}$ is driven by operator policy. Both layers are shown with only the VC-3 layer having a network connection. Once the VC-3 connection is established, the ETH FPP link connection between the two NCC_{MAC} comes into existence.

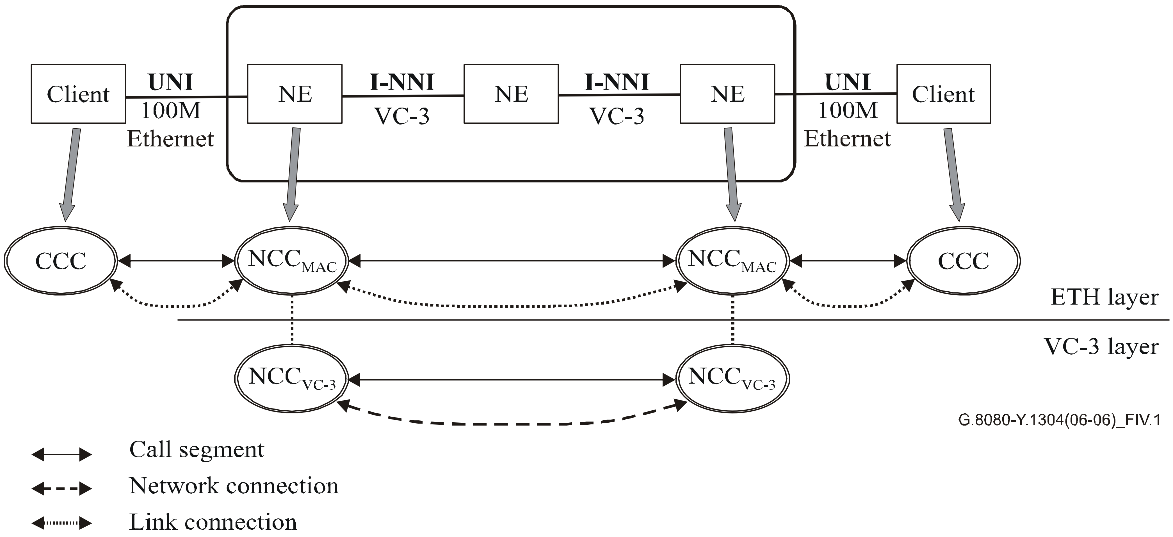


Figure II.1 – Ethernet over VC-3 example

In the sequence of events, the establishment of calls at different server layers may be independent in time. For example, the incoming Ethernet call could trigger the VC-3. Alternately the VC-3 connection may already exist and then be associated to an incoming MAC call. The association of the VC-3 connection to the requested Ethernet call is also driven by operator policy.

There are numerous other examples of interlayer calls such as fibre channel over SDH/OTN.

Appendix III

Component interactions for connection set-up

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

Clause 7.1 states that controller components are abstract entities that may be implemented as a single entity or as a distributed set of entities making up a cooperative federation. However, for clarity of illustration, the examples in this appendix show potential implementation approaches in which the components shown are not abstract entities but rather specific instances of implementation code. Specifically:

– network call controllers are shown as a distributed cooperative federation

– routing controllers are shown in a distributed cooperative federation

– connection controllers are shown as a single entity for a matrix

– LRM are shown as a single entity handling all link ends for a matrix.

In some examples, a shaded box is used to show the boundaries of the distributed cooperative federation that make up an abstract entity.

In order to control a connection, it is necessary for a number of components to interact.

Three basic forms of algorithm for dynamic path control can be distinguished: hierarchical, source routing and step-by-step routing as shown in the following figures. The different forms of path control result in a different distribution of components between nodes and relationships between these connection controllers. In case an RC does not have sufficient routing information to provide a route for a connection request, it may communicate with other RCs to resolve the route using the route query interface as described in clause 7.3.2.

III.1 Hierarchical routing

In the case of hierarchical routing, as illustrated in Figure III.1, a node contains a routing controller, connection controllers and link resource managers for a single level in a routing area hierarchy. The decomposition of routing areas follows the decomposition of a layer network into a hierarchy of subnetworks (in line with the concepts described in [ITU-T G.805]. Connection controllers are related to one another in a hierarchical manner. Each routing area has its own dynamic connection control that has knowledge of the topology of its routing area but has no knowledge of the topology of routing areas above or below itself in the hierarchy, or other routing areas at the same level in the hierarchy.

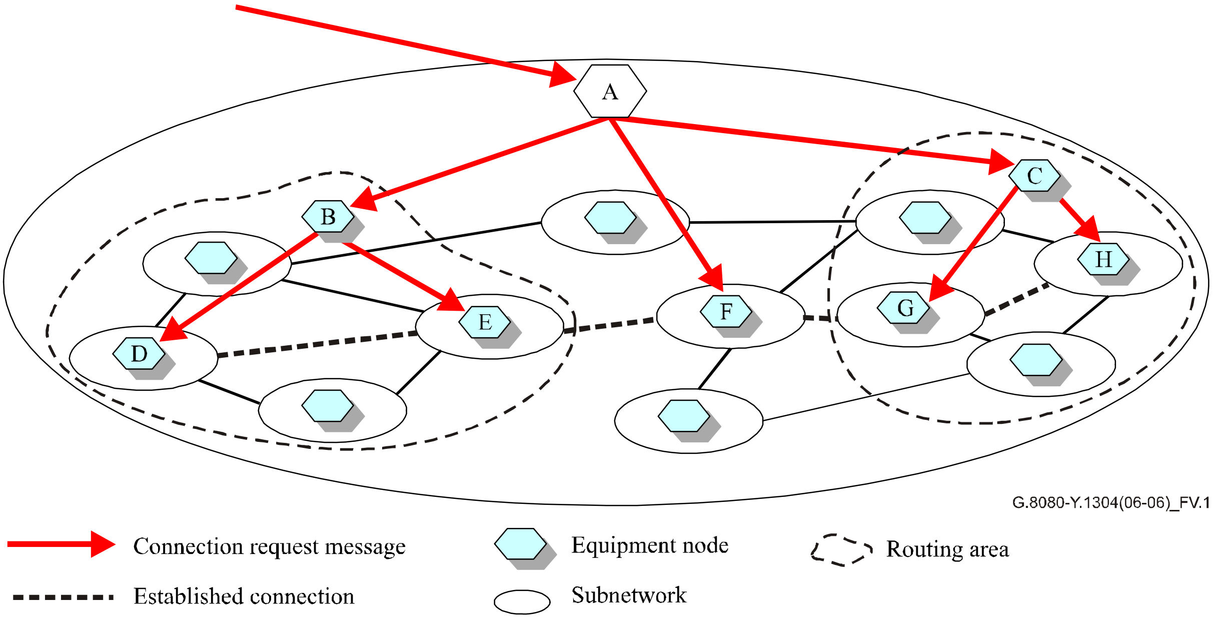


Figure III.1 – Hierarchical signalling flow

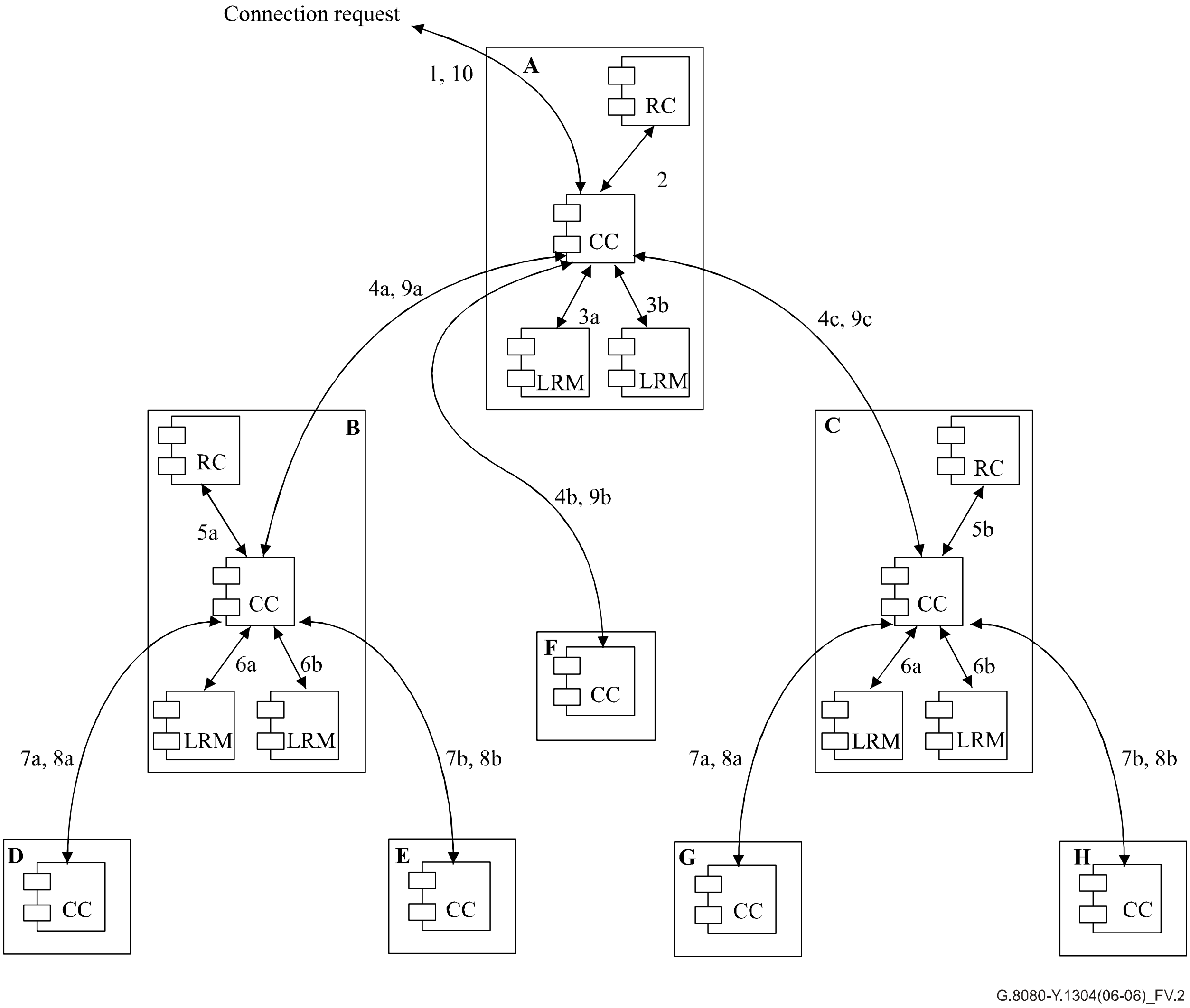


Figure III.2 – Hierarchical routing interactions

In Figure III.2, the detailed sequence of operations involved in setting up a connection using hierarchic routing is described. The steps involved are listed below:

1) A connection request arrives at the connection controller (CC) from the connection request in interface, specified as a pair of SNPs at the edge of the top level routing area.

2) The routing controller (RC) is queried (using the Z end SNP over the route table query interface) and returns the set of links and subnetworks involved.

3) Link connections are obtained (in any order, i.e., 3a, or 3b in Figure III.2) from the link resource managers (LRM) over the link connection request interface.

4) Having obtained link connections (specified as SNP pairs), subnetwork connections can be requested from the child routing areas, by passing a pair of SNPs over the connection request in interface and confirming subnetwork connections to the CC via the connection request out interface. Again, the order of these operations is not fixed, the only requirement being that link connections are obtained before subnetwork connections can be created. The initial process now repeats recursively.

5) The child routing controllers now resolve a route between the SNPs specified.

6) Link connections are obtained (in any order) from the link resource managers (LRM) over the link connection request interface.

7) As a final step, the lowest level switches, which do not contain any routing or link allocation components at all, provide the necessary subnetwork connections.

8) The remaining steps indicate the flow of confirmations that the connection has been set up, culminating in step 10, where the confirmation is returned to the original user.

III.2 Source and step-by-step routing

While similar to hierarchical routing, for source routing, the connection control process is now implemented by a federation of distributed connection and routing controllers. The significant difference is that connection controllers invoke a different sequence of path computation functions between routing levels for hierarchical vs source routing. The signal flow for source (and step-by-step) routing is illustrated in Figure III.3.

In order to reduce the amount of network topology, each controller only needs to have available that portion of the topology that applies to its own routing area.

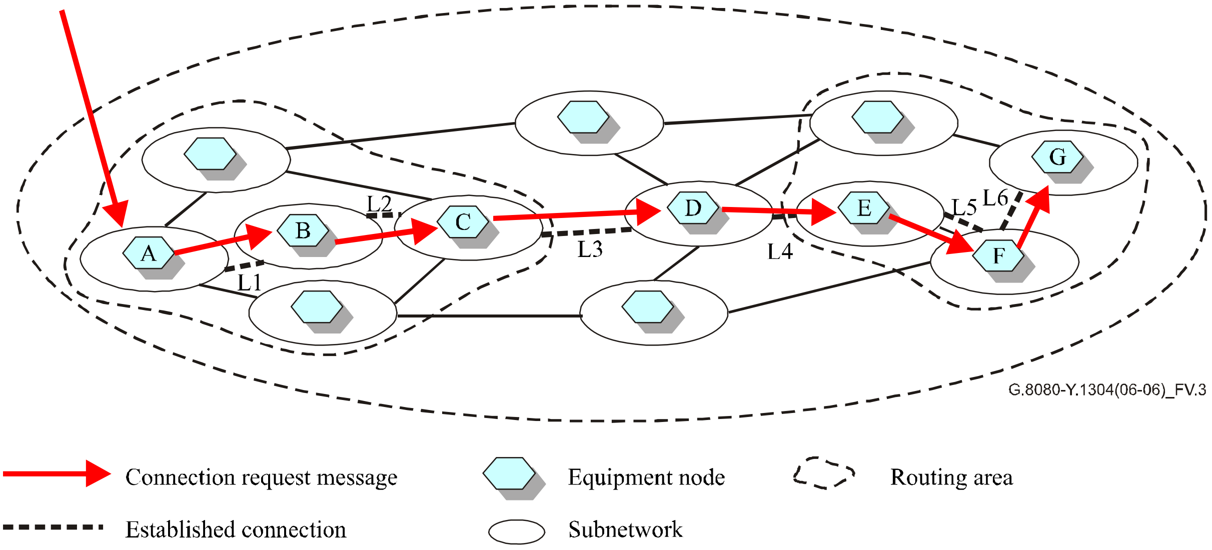


Figure III.3 – Source and step-by-step signalling flow

III.2.1 Source routing

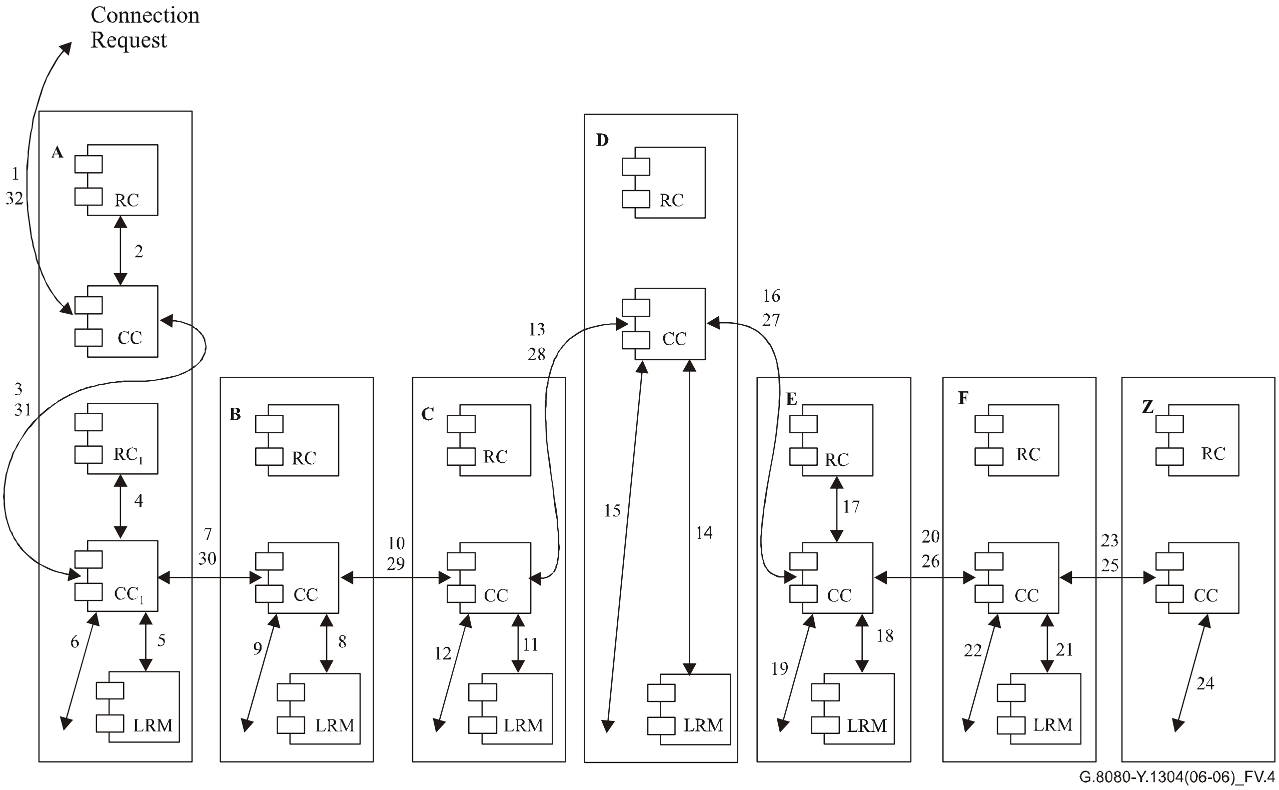


Figure III.4 – Source routing interactions

In the following steps, we describe the sequence of interactions shown in Figure III.4. The following notation is used: X_A represents the component at the highest level in node A, X_{An} represents the component that is at the next nth highest level in node A.

1) A connection request arrives at the connection controller (CC_A) from the connection request in interface, specified as a pair of names (A and Z) at the edge of the subnetwork.

2) The routing controller (RC_A) is queried (using the Z end SNP over the route table query interface) and returns route (A, L3, L4, Z).

3) As CC_A does not have access to the necessary link resource manager (LRM_C), the request (A, L3, L4, Z) is passed on to a peer CC_{A1} (over the connection request out/in interface), which controls routing through this routing area.

4) CC_{A1} queries RC_{A1} (over the route query interface) for L3 and obtains a list of additional links, L1 and L2.

5) Link L1 is local to this node, and a link connection for L1 is obtained from LRM_A over the link connection request interface.

6) The SNC is made across the local switch (controller not shown).

7) The request, now containing the remainder of the route (L2, L3, L4 and Z), is forwarded to the next peer CC_B (over the peer coordination out/in interface).

8) LRM $_B$ controls L2, so a link connection is obtained from this link over the link connection request interface.

9) The SNC is made across the local switch (controller not shown).

10) The request, now containing the remainder of the route (L3, L4 and Z), is forwarded to the next peer CC_C (over the peer coordination out/in interface).

11) LRM$_C$ controls L3, so a link connection is obtained from this link over the link connection request interface.

12) The SNC is made across the local switch (controller not shown).

13) The request, now containing the remainder of the route (L4, Z), is forwarded to the next peer CC$_D$ (over the peer coordination out/in interface).

14) LRM$_D$ controls L4, so a link connection is obtained from this link over the link connection request interface.

15) The SNC is made across the local switch (controller not shown).

16) The request, now containing the remainder of the route (Z), is forwarded to the next peer CC$_E$ (over the peer coordination out/in interface).

17) CC$_E$ queries RC$_E$ (over the route table query interface) for Z and obtains links L5 and L6.

The process of connecting across the next routing area (i.e., steps 18 to 24 in Figure III.4) is identical to that already described. Events 25 to 32 describe the flow of confirmation signals to the connection originator.

III.2.2 Step-by-step routing

In this form of routing there is further reduction of routing information in the nodes, and this places restrictions upon the way in which routing is determined across the subnetwork. Figure III.5 applies to the network diagram of Figure III.3.

The process of step-by-step routing is identical to that described for source routing, with the following variation: routing controller RC$_{A1}$ can only supply link L1, and does not supply link L2 as well. CC$_B$ must then query RC$_B$ (via the route table query interface) for L2 in order to obtain L2. A similar process of obtaining one link at a time is followed when connecting across the second routing area.

1） A connection request arrives at the connection controller (CC$_A$) from the connection request in interface, specified as a pair of names (A and Z) at the edge of the subnetwork.

2） The routing controller (RC$_A$) is queried (using the Z end SNP over the route table query interface) and returns the egress link, L3.

3） As CC$_A$ does not have access to the necessary link resource manager (LRM$_C$), the request (A, L3, Z) is passed on to a peer CC$_{A1}$ (over the connection request out/in interface), which controls routing through this routing area.

4） CC$_{A1}$ queries RC$_{A1}$ (over the route query interface) for L3 and obtains L1.

5） Link L1 is local to this node, and a link connection for L1 is obtained from LRM$_A$ over the link connection request interface.

6） The SNC is made across the local switch (controller not shown).

7） The request, now containing the route (L3 and Z), is forwarded to the next peer CC$_B$ (over the peer coordination out/in interface).

8） CC$_{B1}$ queries RC$_{B1}$ (over the route query interface) for L3 and obtains L2.

9） LRM$_B$ controls L2, so a link connection is obtained from this link over the link connection request interface.

10） The SNC is made across the local switch (controller not shown).

11） The request, now containing the remainder of the route (L3 and Z), is forwarded to the next peer CCC (over the peer coordination out/in interface).

12） LRM$_{C}$ controls L3, so a link connection is obtained from this link over the link connection request interface.

13） The SNC is made across the local switch (controller not shown).

14） The request, now containing the remainder of the route (Z), is forwarded to the next peer CC$_{D}$ (over the peer coordination out/in interface).

15） CC$_{D}$ queries RC$_{D}$ (over the route query interface) for Z and obtains link L4.

16） LRM$_{D}$ controls L4, so a link connection is obtained from this link over the link connection request interface.

17） The SNC is made across the local switch (controller not shown).

18） The request, now containing the remainder of the route (Z), is forwarded to the next peer CC$_{E}$ (over the peer coordination out/in interface).

19） CC$_{E}$ queries RC$_{E}$ (over the route query interface) for Z and obtains link L5.

20） LRM$_{E}$ controls L5, so a link connection is obtained from this link over the link connection request interface.

21） The SNC is made across the local switch (controller not shown).

22） The request, now containing the remainder of the route (Z), is forwarded to the next peer CC$_{F}$ (over the peer coordination out/in interface).

23） CC$_{F}$ queries RC$_{F}$ (over the route query interface) for Z and obtains link L6.

24） LRM$_{F}$ controls L6, so a link connection is obtained from this link over the link connection request interface.

25） The SNC is made across the local switch (controller not shown).

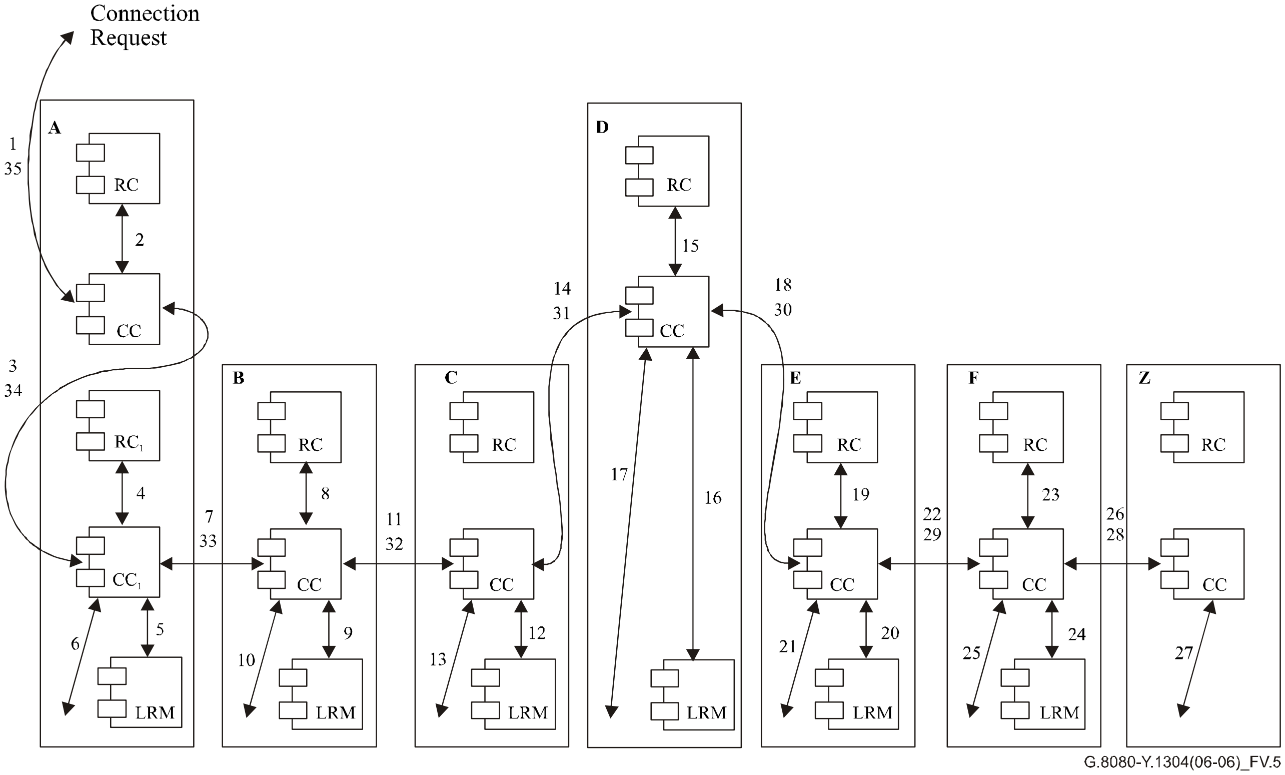


Figure III.5 – Step-by-step routing

III.2.3 Combination of source and step-by-step routing

Figure III.6 illustrates an example where source and step-by-step routing can be used, but at different routing levels. In this example, the low level routing is source routing, while the high level routing is step-by-step.

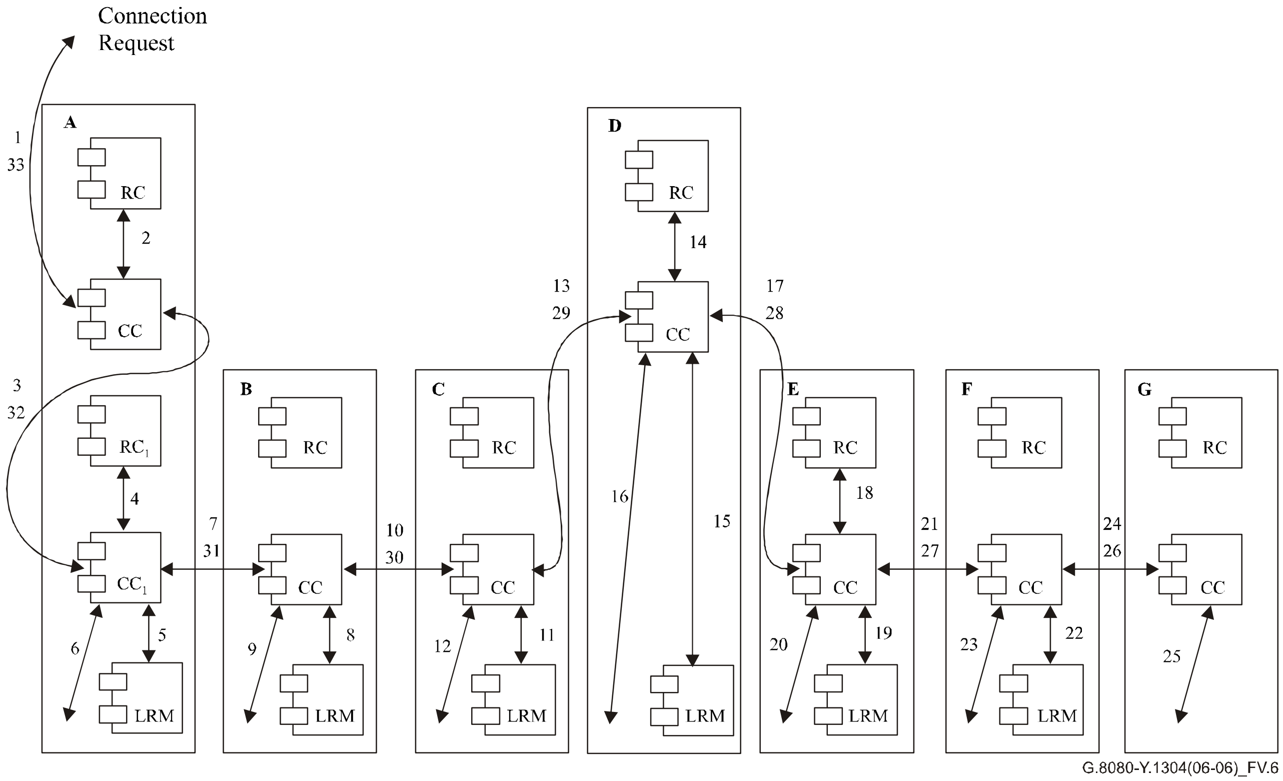


Figure III.6 – Combined source and step-by-step routing

1) A connection request arrives at the connection controller (CC$_A$) from the connection request in interface, specified as a pair of names (A and Z) at the edge of the subnetwork.

2) The routing controller (RC$_A$) is queried (using the Z end SNP over the route table query interface) and returns the egress link, L3.

3) As CC$_A$ does not have access to the necessary link resource manager (LRM$_C$), the request (A, L3, Z) is passed on to a peer CC$_{A1}$ (over the connection request out/in interface), which controls routing through this routing area.

4) CC$_{A1}$ queries RC$_{A1}$ (over the route table query interface) for L3 and obtains a list of additional links, L1 and L2.

5) Link L1 is local to this node, and a link connection for L1 is obtained from LRM$_A$ over the link connection request interface.

6) The SNC is made across the local switch (controller not shown).

7) The request, now containing the remainder of the route (L2, L3 and Z), is forwarded to the next peer CC$_B$ (over the peer coordination out/in interface).

8) LRM$_B$ controls L2, so a link connection is obtained from this link over the link connection request interface.

9) The SNC is made across the local switch (controller not shown).

10) The request, now containing the remainder of the route (L3 and Z), is forwarded to the next peer CC$_C$ (over the peer coordination out/in interface).

11) LRM$_C$ controls L3, so a link connection is obtained from this link over the link connection request interface.

12) The SNC is made across the local switch (controller not shown).

13) The request, now containing the remainder of the route (Z), is forwarded to the next peer CC$_D$ (over the peer coordination out/in interface).

14) CC$_D$ queries RC$_D$ (over the route table query interface) for Z and obtains link L4.

15) LRM$_D$ controls L4, so a link connection is obtained from this link over the link connection request interface.

16) The SNC is made across the local switch (controller not shown).

17) The request, now containing the remainder of the route (Z), is forwarded to the next peer CC$_E$ (over the peer coordination out/in interface).

18) CC$_E$ queries RC$_E$ (over the route table query interface) for Z and obtains links L5 and L6.

III.3 Connection protection

When the control plane is used to provide protection, a protection connection is set up to protect the working connection before the happening of a failure. After a working connection failure is detected, only the source and destination connection controllers are involved to complete the protection switching operation from the original working connection to protection connection.

Figure III.7 shows an example of connection protection using source based routing and distributed signalling. Here the protection signalling flow is shown after a link failure is detected. The relationship of the working and protection is assumed to be 1:1. That is, CI is not transferred on both working and protection at the same time. Instead, when the working path is interrupted by a link failure, the control plane is used to switch user CI to the protection path.

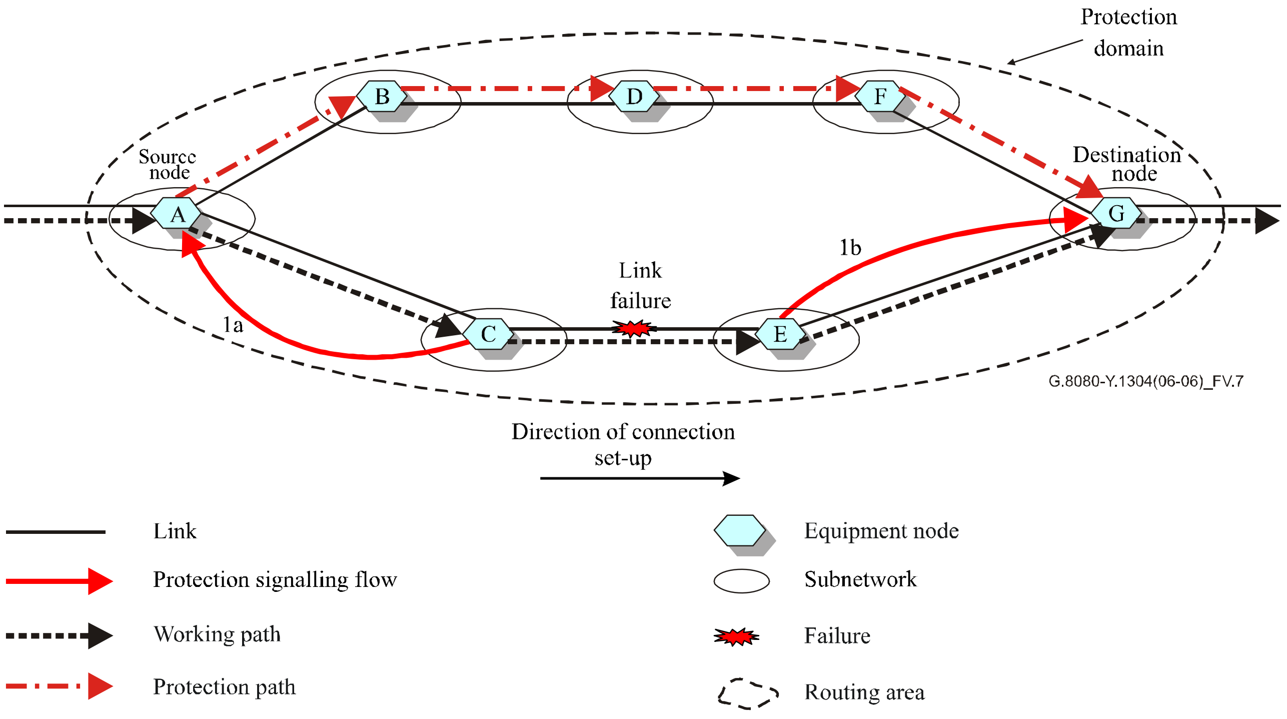


Figure III.7 – Protection signalling flow

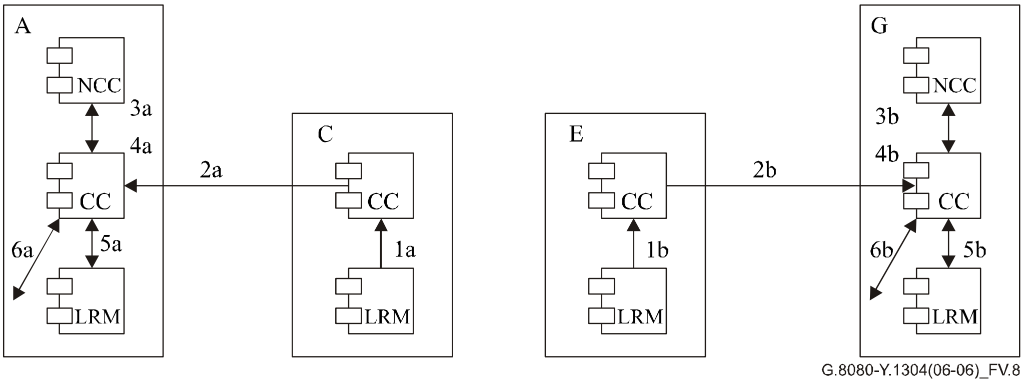


Figure III.8 – Protection interactions

In Figure III.8, the detailed sequence of operations involved in protection is described. The steps involved are listed below:

1) A bidirectional link failure notification generated by the link resource managers (LRM) arrives at the connection controller (CC) containing the failure link information. This occurs in node E and node C.

2) The link failure notification is forwarded to CC_A from CC_C and to CC_G from CC_E.

3) At both CC_A and CC_G, the NCCs are alerted to the failure of the working path.

4) The NCCs initiate the protection switching request to their CCs that cause the SNC to be made across the local switch from working connection to protection connection.

III.4 Restoration – Hard re-routing – Intra-domain – Hierarchical method

In hard re-routing, which is known as break-before-make, the original connection segment is released prior to the creation of an alternative connection segment.

Figure III.9 shows the signalling flow of a hard re-routing scenario with hierarchical connection control after an intra-domain link failure is detected. In the step of re-routing connection creation, hierarchical algorithm is adopted.

In Figure III.10, the detailed sequence of operations involved in Figure III.9 is described. The steps involved are listed below:

1) An intra-domain link failure notification generated by the link resource managers (LRM) arrives at the connection controller (CC), containing the crankback routing information which specifies the failure link. This may occur in node J or node H or both according to which node detects the link failure.

2) The intra-domain link failure notification is forwarded to CC_B.

3) Link connections are released (in any order, i.e., 3a, or 3b in Figure III.10) by LRM.

4) The SNCs are released by the lowest level switches.

5) The connection release confirmations are returned to CC_B.

6) The routing controller (RC_B) is queried with crankback routing information and returns the set of links excluding the failure link and subnetworks involved.

7-9) Steps 7 to 9 describe the flow of connection set-up using hierarchical algorithm which is identical to that described in clause III.1, Hierarchical routing.

10) If failed to set up the connection in re-routing domain A, the crankback routing information is forwarded to upper level re-routing domain C.

11) The remaining link connections are released by the LRM.

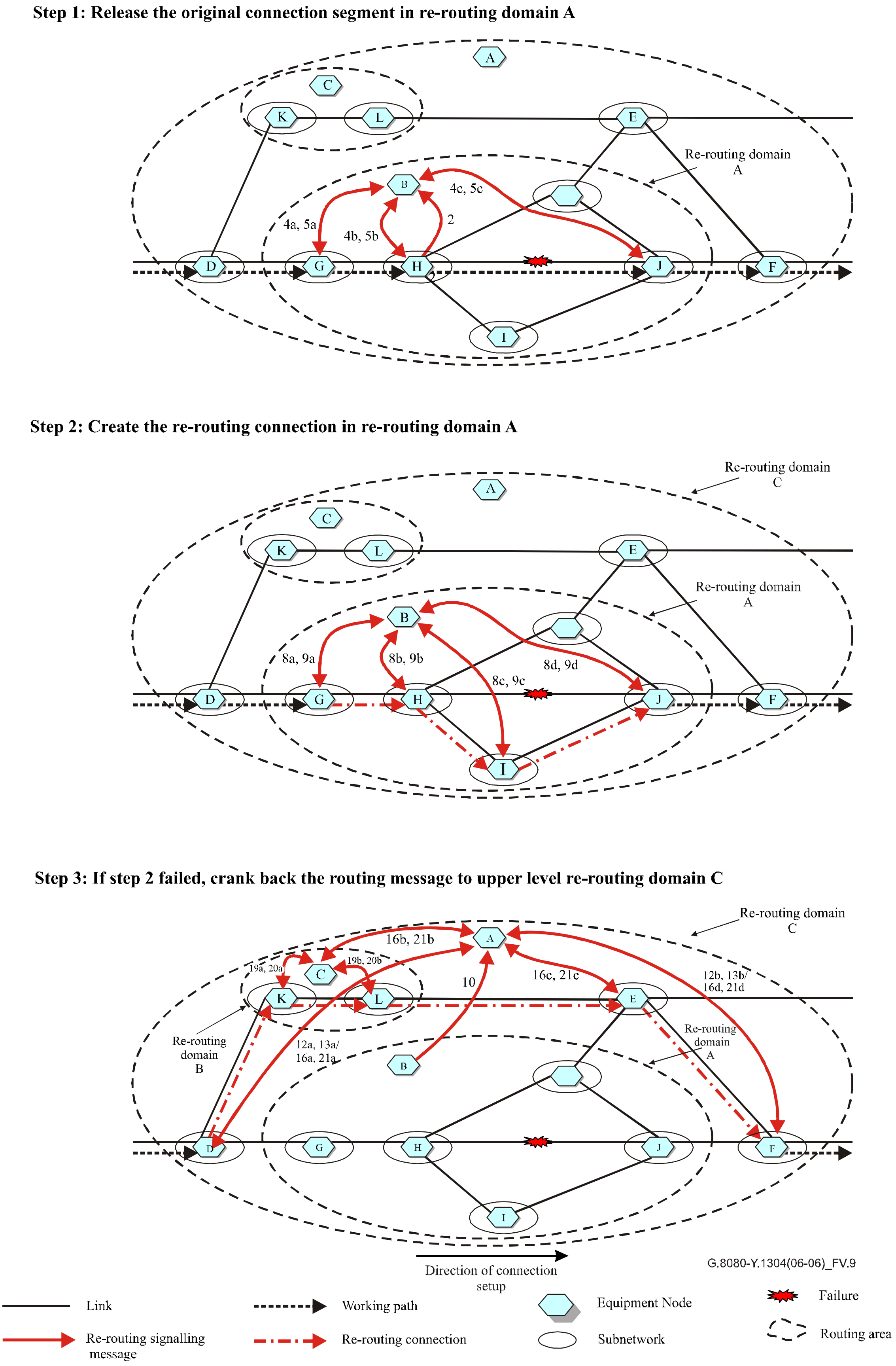
12) The SNCs are released by the lowest level switches. This requires release at nodes G and J via CC_B and then CC_G and CC_J.

13) The connection release confirmations are returned to CC_A. This includes release from CC_B.

14) RC_A is queried with crankback routing information and returns the set of links excluding the failure link and subnetworks involved.

15-21) Steps 15 to 21 describe the flow of connection set-up using hierarchical algorithm which is identical to that described in clause III.1, Hierarchical routing.

22) If failed to set up the connection in re-routing domain C, the crankback routing information is forwarded to upper level re-routing domain.



Step 1: Release the original connection segment in re-routing domain A

Step 2: Create the re-routing connection in re-routing domain A

Step 3: If step 2 failed, crank back the routing message to upper level re-routing domain C

Figure III.9 – Signalling flow of hard re-routing using a hierarchical algorithm after an intra-domain link failure

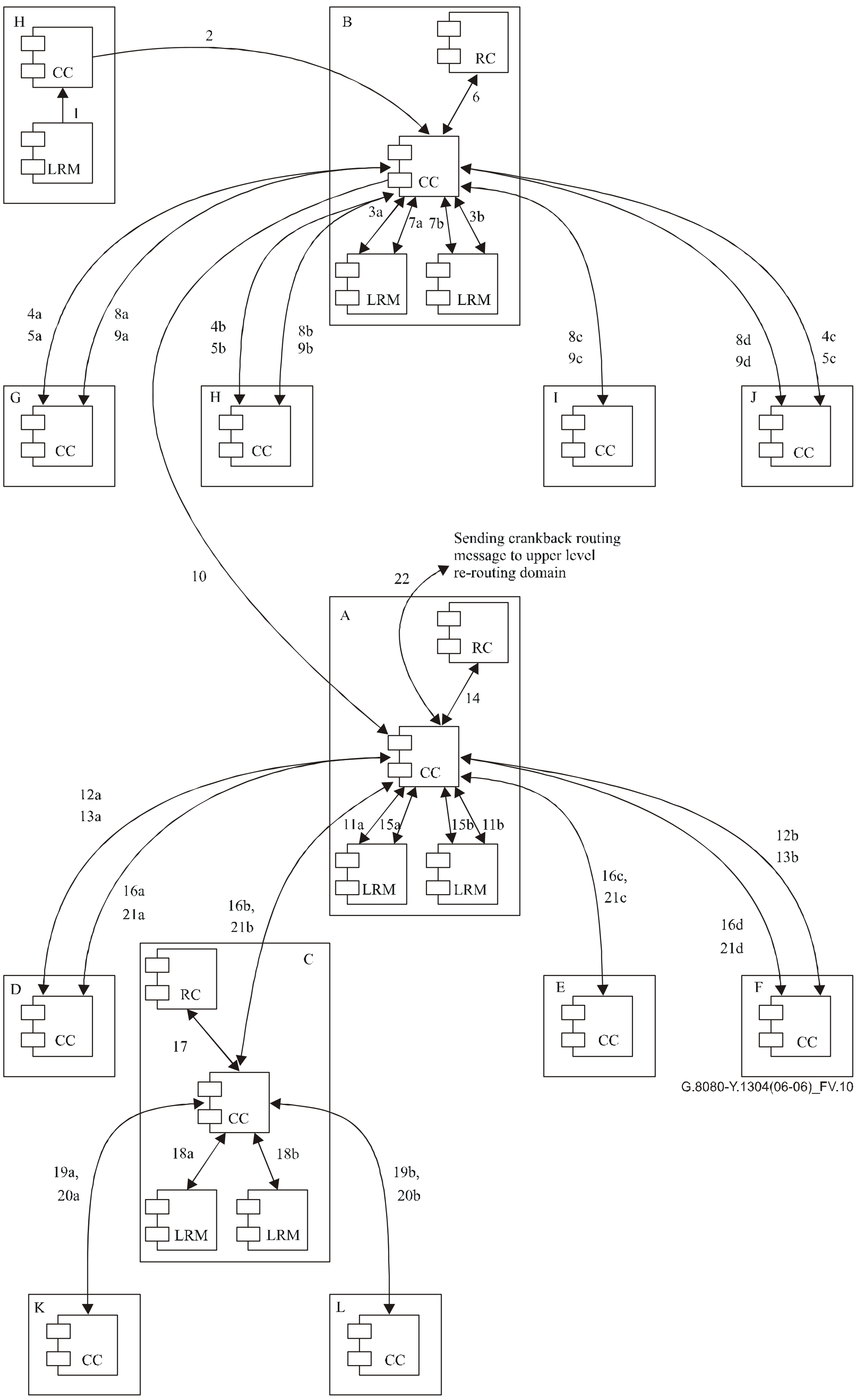


Figure III.10 – Component interactions of hard re-routing using a hierarchical algorithm after an intra-domain link failure

III.5 Restoration – Soft re-routing – Intra-domain – Source method

Soft re-routing service is a mechanism for the re-routing of a call for administrative purposes. When a re-routing operation is triggered (generally via a request from the management plane) and sent to the location of the re-routing components, the re-routing components establish a re-routing connection that traverses (or does not traverse) the appointed set of components according to the administrative purposes. In soft re-routing, which is known as make-before-break, the initial connection is deleted after the creation of a re-routing connection.

Figure III.11 shows the signalling flow of a soft re-routing scenario with source (or step-by-step) routing connection control after receiving a management plane request to re-route a connection excluding a certain intra-domain link.

In Figure III.12, the detailed sequence of operations using source routing involved in Figure III.11 is described. The steps involved are listed below:

1) A management plane request arrives at the connection controller (CC_G), containing constraints that the re-routing connection must comply with. For example, an explicit route of the re-routing connection. In this example, there is an exclusion constraint that specifies that link L1 is not to be used in the re-routing connection.

2a) Routing controller (RC_G) receives a re-routing connection set-up request initiated by CC_G containing the pair of SNPs at the edge of the re-routing domain A and the exclusion constraint.

2b) RC_G returns the set of links excluding the link L1.

3-15) Steps 3 to 15 describe the flow of connection set-up using a source routing algorithm which is identical to that described in clause III.2.3, Source and step-by-step routing. The new connection is joined to the original one coming into domain A at G and J.

If the connection is set up successfully in re-routing domain A, then step 16a is followed, otherwise step 16b is followed.

16a) The link connection of the original path is released by LRM_G in step 16a which consists of steps 16a1 and 16a2.

17) The SNC is released across the local switch.

18) The connection release request, containing the original connection information, is forwarded to CC_H.

19) The link connection of the original path is released by LRM_H.

20) The SNC is released across the local switch.

21) The connection release request, containing the original connection information, is forwarded to CC_J.

22) The SNC is released across the local switch.

23) The connection release confirmation is returned to the source CC_G and the re-routing process completes.

16b) The crankback routing information is forwarded to CC_D in upper level re-routing domain C.

17a) RC_D receives a re-routing connection set-up request initiated by CC_D containing the pair of SNPs at the edge of the re-routing domain C and the exclusion constraint to avoid domain A.

17b) RC_D returns the set of links excluding domain A.

18-39) Steps 18 to 39 describe the flow of connection set-up using a source routing algorithm which is identical to that described in clause III.2.3, Source and step-by-step routing.

40) The link connection of the original path is released by LRM_D in step 40 which consists of steps 40a and 40b.

41) The SNC is released across the local switch.

42) The connection release request, containing the original connection information, is forwarded to CC_G.

43) The link connection of the original path is released by LRM_G.

44) The SNC is released across the local switch.

45) The connection release request, containing the original connection information, is forwarded to CC_H.

46) The link connection of the original path is released by LRM_H.

47) The SNC is released across the local switch.

48) The connection release request, containing the original connection information, is forwarded to CC_J.

49) The link connection of the original path is released by LRM_J.

50) The SNC is released across the local switch.

51) The connection release request, containing the original connection information, is forwarded to CC_F.

52) The SNC is released across the local switch.

53) The connection release confirmation is returned to the source CC_D.

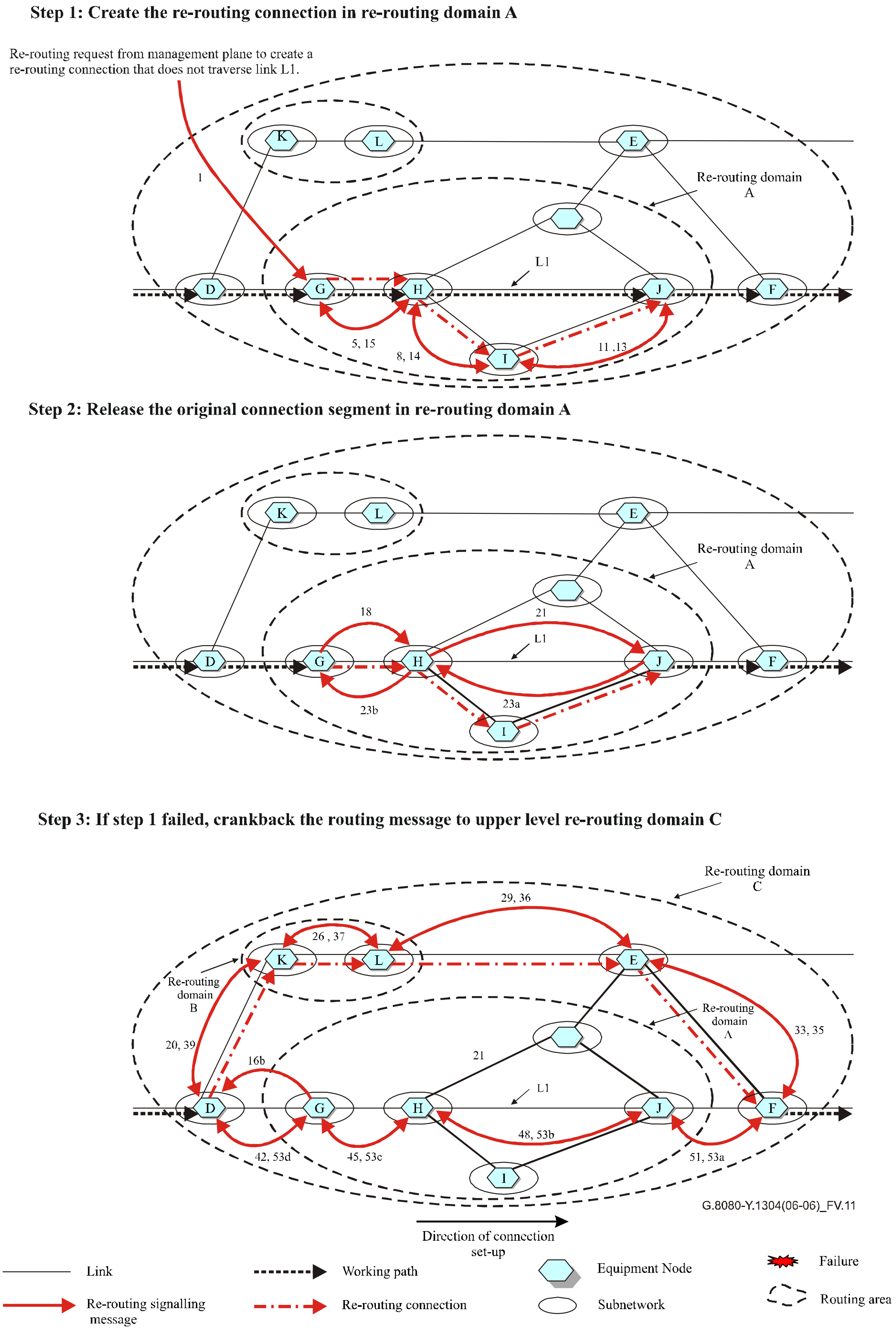

Step 1: Create the re-routing connection in re-routing domain A
Re-routing request from management plane to create a re-routing connection that does not traverse link L1.
1
K
L
E
Re-routing domain A
L1
D
G
H
J
F
I
5, 15
8, 14
11 .13
Step 2: Release the original connection segment in re-routing domain A
K
L
E
Re-routing domain A
18
21
L1
D
G
H
J
F
23b
23a
I
Step 3: If step 1 failed, crankback the routing message to upper level re-routing domain C
Re-routing domain C
29, 36
26 , 37
K
L
E
Re-routing domain B
20, 39
16b
21
Re-routing domain A
33, 35
L1
D
G
H
J
F
48, 53b
42, 53d
45, 53c
51, 53a
I
G.8080-Y.1304(06-06)_FV.11
Direction of connection set-up
Link
Working path
Equipment Node
Failure
Re-routing signalling message
Re-routing connection
Subnetwork
Routing area

Figure III.11 – Signalling flow of soft re-routing using a source (or step-by-step) routing algorithm excluding an intra-domain link

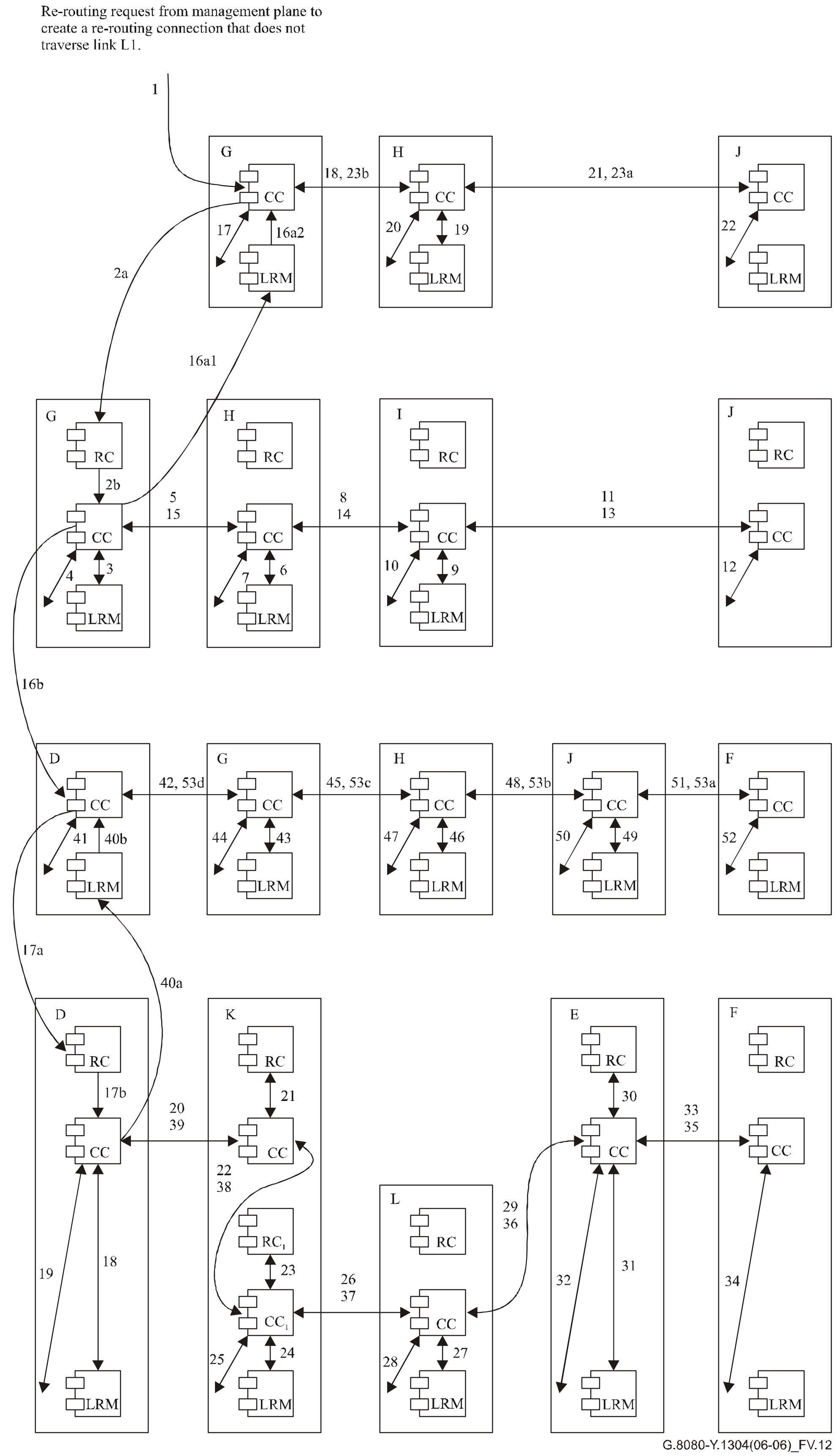


Figure III.12 – Component interactions of soft re-routing using a source routing algorithm excluding an intra-domain link

III.6 Restoration – Revertive re-routing – Intra-domain – Source method

In revertive behaviour re-routing, the original connection must not be released and is monitored by the network call controllers. When the failure is repaired, the call is restored to the original connection.

Figure III.13 shows the signalling flow of a revertive behaviour re-routing scenario with source (or step-by-step) routing connection control after an intra-domain link failure is detected.

In Figure III.14, the detailed sequence of operations using source routing involved in Figure III.13 is described. The steps involved are listed below:

1) An intra-domain link failure notification generated by the link resource managers (LRM) arrives at the connection controller (CC), containing the crankback routing information which specifies the failure link. This may occur in node J or node H or both according to which node detects the link failure.

2) The intra-domain link failure notification is forwarded to CC_G. No SCN changes are made.

3) The routing controller (RC_G) is queried with crankback routing information and returns the set of links excluding the failure link and subnetworks involved.

4-16) Steps 4 to 16 describe the flow of connection set-up using a source routing algorithm which is identical to that described in clause III.2.3, Source and step-by-step routing.

17) If failed to set up the connection in re-routing domain A, the crankback routing information is forwarded to upper level re-routing domain C.

18) RC_D is queried with crankback routing information and returns the set of links excluding domain A.

19-40) Steps 19 to 40 describe the flow of connection set-up using a source routing algorithm which is identical to that described in clause III.2.3, Source and step-by-step routing.

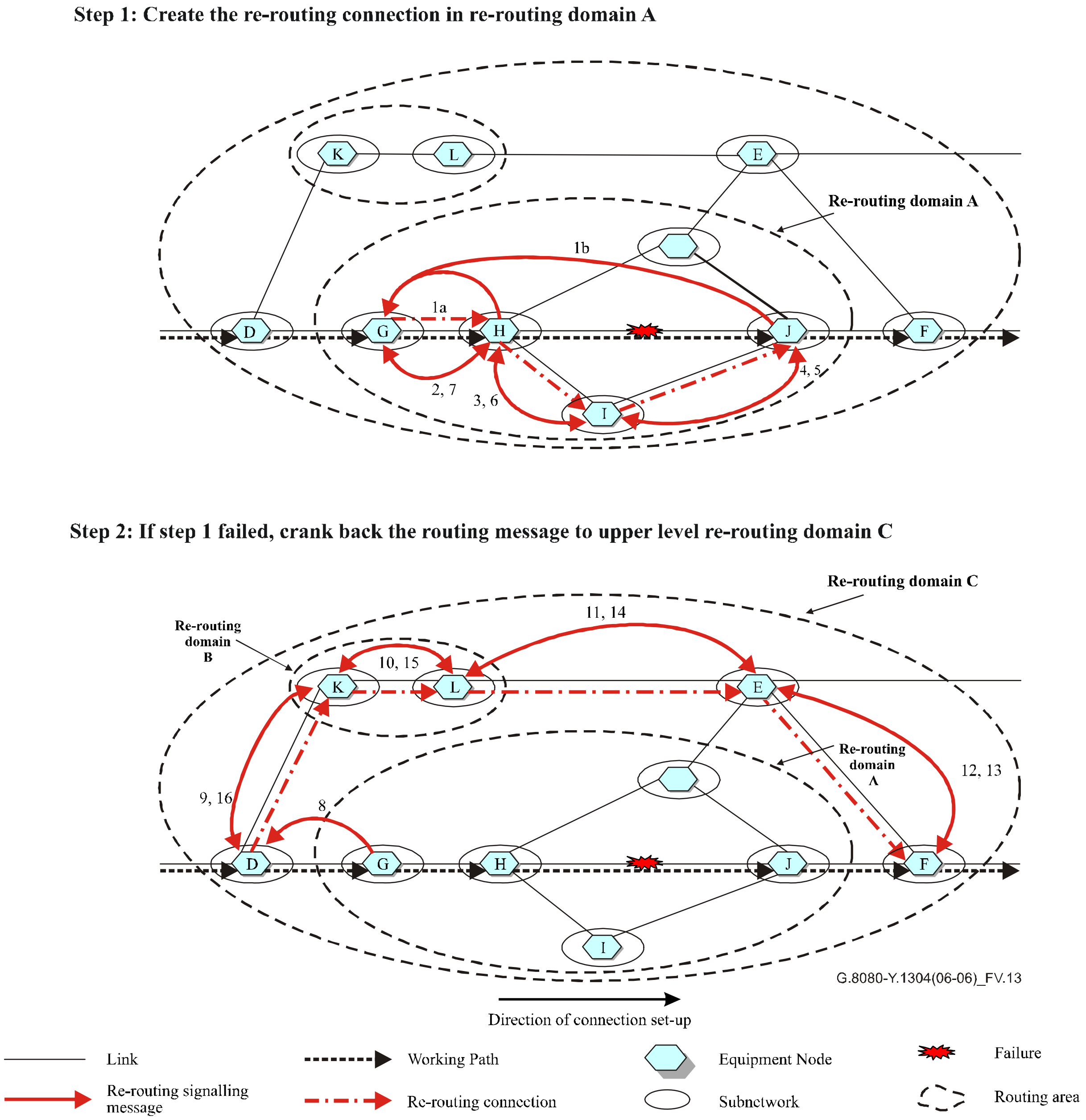

Step 1: Create the re-routing connection in re-routing domain A
K
L
E
Re-routing domain A
1b
1a
D
G
H
J
F
2, 7
3, 6
4, 5
I
Step 2: If step 1 failed, crank back the routing message to upper level re-routing domain C
Re-routing domain C
11, 14
Re-routing domain B
10, 15
K
L
E
Re-routing domain A
12, 13
9, 16
8
D
G
H
J
F
I
G.8080-Y.1304(06-06)_FV.13
Direction of connection set-up
Link
Working Path
Equipment Node
Failure
Re-routing signalling message
Re-routing connection
Subnetwork
Routing area

Figure III.13 – Signalling flow of revertive behaviour re-routing using a source (or step-by-step) routing algorithm after an intra-domain link failure

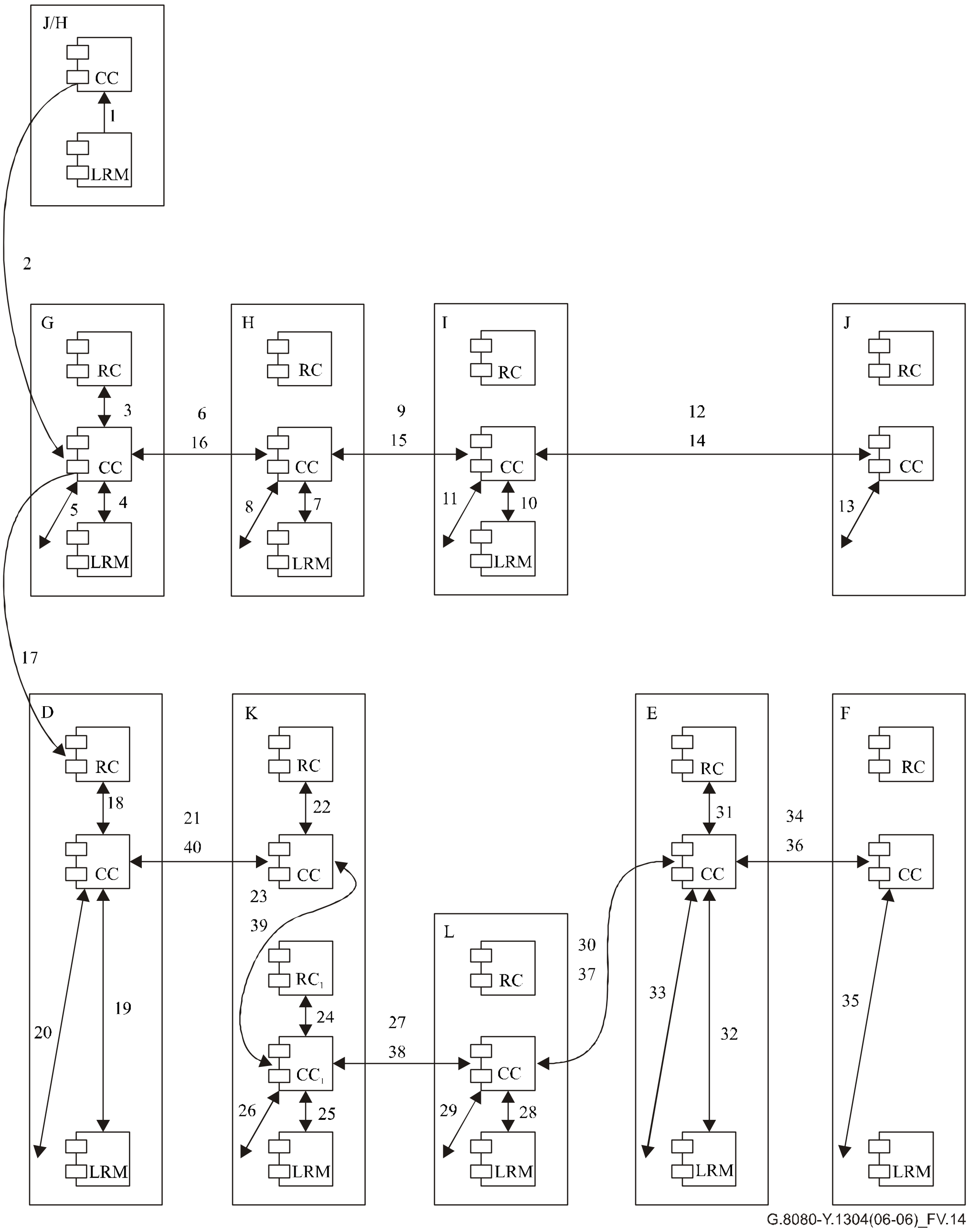


Figure III.14 – Component interactions of revertive behaviour re-routing using a source routing algorithm after an intra-domain link failure

III.7 Source routing using a routing query interface

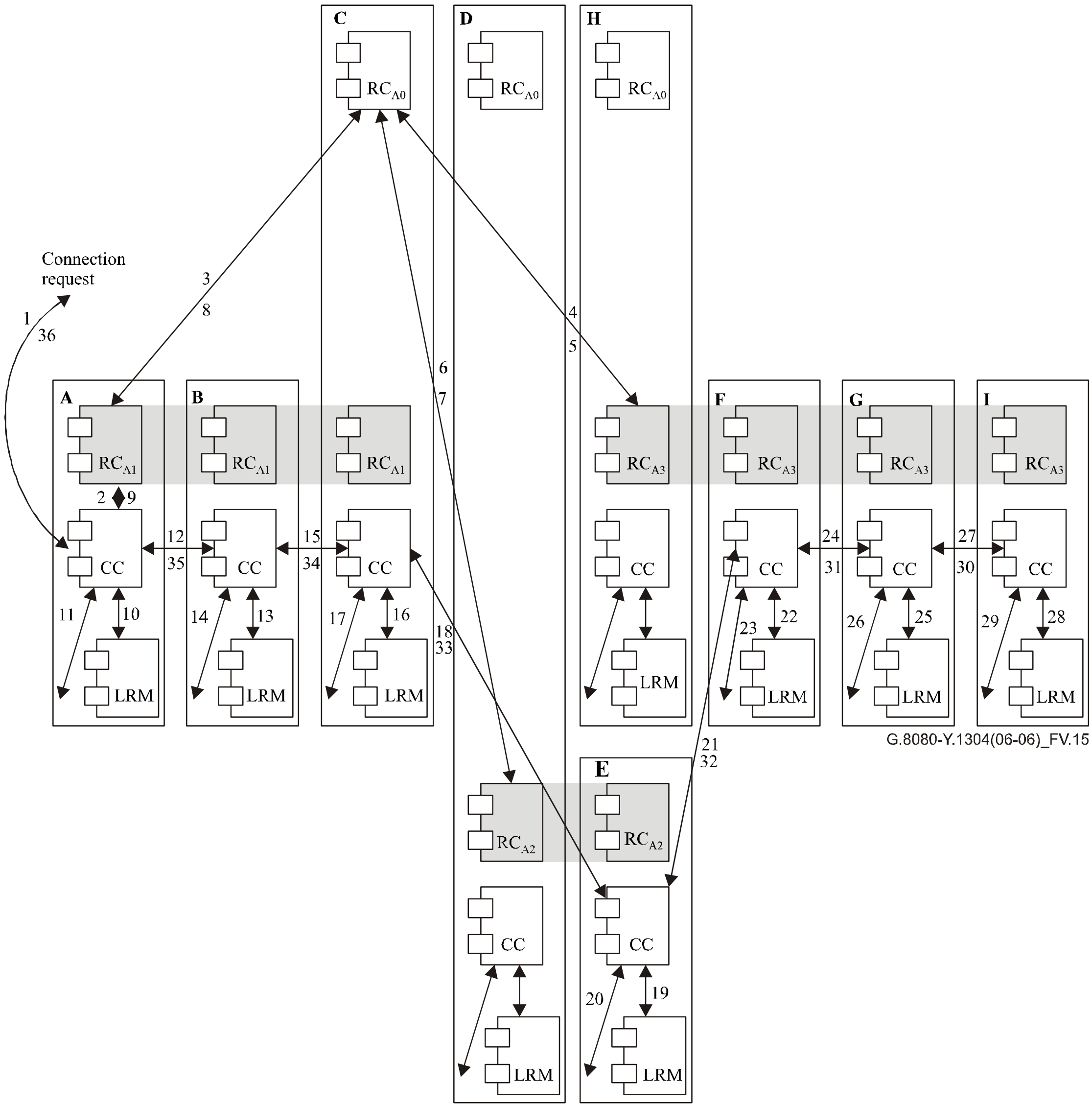


Figure III.15 – Component interactions for source routing using a routing query interface

Figure III.15 illustrates the detailed sequence of operations involved in setting up a connection using source routing assisted by RC-RC route query. The notation $RC_{A1,}$ RC_{A2}, etc., represent the routing controller in areas A1, A2, etc. The actual communication components may be facilitated by other intermediate components – for example, the communication from RC_{A0} on node C to RC_{A2} on node D may be performed by transferring the message through RC_{A0} on node D.

The steps involved are listed below.

1) A connection request arrives at the connection controller (CC_A) from the connection_request_in interface, specified as a pair of names (A and Z) at the edge of the subnetwork.

2) The routing controller RC_{A1} on node A is queried (using the Z end SNP over the route query interface).

3) The routing controller RC_{A1} on node A recognizes that the destination address is not visible within area A1 so it sends a route query to RC_{A0} on node C for assistance over the route query interface. While RC_{A1} on node C has the same routing information as RC_{A1} on node A as they are in a common routing area, RC_{A0} on node C has visibility to the destination making the computation of a path possible.

4) In the process of computing a path to the destination, RC_{A0} on node C recognizes that to reach the destination it needs to reach area A3. However, since there are multiple paths between area A1 and area A3, it needs the assistance of RC_{A2} and RC_{A3} to determine the best path. Thus a query is sent by RC_{A0} on node C to RC_{A3} on node H to determine which link from A2 to A3 should be used.

5) RC_{A3} on node H computes the possible paths from the links entering area A3 from area A2 to the destination within area A3. From this, it can determine the costs of using either of the paths, and returns this information to RC_{A0} on node C.

6) As with RC_{A3} on node H, RC_{A0} on node C sends a query to RC_{A2} on node D to determine the paths between the egress links that egress area A2 and enter area A3 and the ingress links that enter area A2 from area A1.

7) RC_{A2} on node D computes the possible paths across area A2, and returns this information to RC_{A0} on node C.

8) RC_{A0} on node C provides to RC_{A1} on node A the list of paths developed from the edge of area A1 to the destination in area A3 and includes the aggregate cost for each path developed.

9) RC_{A1} on node A now has the necessary information to compute a path across area A1 utilizing the cost information provided by RC_{A0} on node C to determine the lowest cost end-to-end path. For the remainder of this example, we assume the path chosen is from A, via L1 to B, via L2 to C, via L3 to E, via L4 to F, via L5 to G, and via L6 to I. It then sends the response back to CC on node A, which starts the process to form the end-to-end connection request using route (A, L1, L2, L3, L4, L5, L6 and Z).

10) L1 is local to node A, and a link connection for L1 is obtained from LRM_A over the link connection request interface.

11) The appropriate SNC is established on the local switch (controller not shown).

12) The connection request (L2, L3, L4, L5, L6 and Z) is then forwarded to the next CC on node B (over the peer coordination_out/in interface).

13) LRM_B controls L2, so a link connection is obtained from this link over the link connection_request interface.

14) The appropriate SNC is established on the local switch (controller not shown).

15) The connection request (L3, L4, L5, L6 and Z) is then forwarded to the next CC on node C (over the peer coordination_out/in interface).

16) LRM_C controls L3, so a link connection is obtained from this link over the link connection_request interface.

17) The appropriate SNC is established on the local switch (controller not shown).

18) The connection request (L4, L5, L6 and Z) is then forwarded to the next CC on node E (over the peer coordination_out/in interface).

19) LRM_E controls L4, so a link connection is obtained from this link over the link connection_request interface.

20) The appropriate SNC is established on the local switch (controller not shown).

21) The connection request (L5, L6 and Z) is then forwarded to the next CC on node F (over the peer coordination_out/in interface).

22) LRM_F controls L5, so a link connection is obtained from this link over the link connection_request interface.

23) The appropriate SNC is established on the local switch (controller not shown).

24) The connection request (L6 and Z) is then forwarded to the next peer CC on node G (over the peer coordination_out/in interface).

25) LRM_G controls L6, so a link connection is obtained from this link over the link connection_request interface.

26) The appropriate SNC is established on the local switch (controller not shown).

27) The connection request (Z) is then forwarded to the next CC on node I.

28) LRM_I controls the egress link to the destination node, so a link connection is obtained from this link over the link connection request interface.

29) The appropriate SNC is established on the local switch (controller not shown).

30) The CC on node I then sends a confirmation back to the CC on node G. The exchange of responses then repeats between pairs of CCs all the way going back to the connection originator CC on node A.

Appendix IV

Combining protection and restoration domains

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

The coordination between CP protection and restoration is performed by the control plane. CP protection and restoration domains are routing domains since they require path computation to create connections in a domain. Thus the relationship between protection and restoration domains is a containment relationship. It is possible for a protection domain and restoration domain to share the same routing controller scope.

When a fault (or repair) occurs, the smallest containing protection/restoration domain should act first, followed by successively larger protection/restoration domains should they be necessary. When the containing domain cannot fix the connection, it must notify the failure to the larger domains. Then the larger domain may use the re-routing or protection action to fix the failure.

What has been mentioned above is the general principle for protection/restoration domain interaction. For example, if the smallest domain where an error occurs is a protection domain, a protection action is applied first. Only when it cannot fix the connection (the second error), the connection failure is signalled to the immediate containing domain.

When the smallest domain where an error occurs is a restoration domain, a re-routing action is applied first. If restoration fails, the connection failure is signalled to the larger domain. The larger domain is either a restoration domain or protection domain. Subsequent restoration/protection actions are taken in the ever increasingly larger containment domains as needed.

Following is an example for this general principle. In a 1+1 SNCP, after a working connection failure is detected, the source and destination controllers are involved to complete the protection switching operation from the original working connection to protection connection. During the protection operation, there is no re-routing operation involved. If the protection connection fails, restoration becomes active because the containing domain is a re-routing domain. The control plane is in charge of the coordination between protection and restoration.

Figure IV.1 shows an example of the coordination between a re-routing domain and a protection domain. The re-routing domain contains the protection domain. After a working connection failure is detected (SNC [A, C, E, F]), the transport plane is involved to complete the protection switching operation from the original working connection to protection connection (SNC [A, B, D, F]). When the protection connection (SNC [A, B, D, F]) fails, connection controllers in the control plane activate the re-routing mechanism, and restoration is performed in a re-routing domain resulting in connection [G, H, I, J, K].

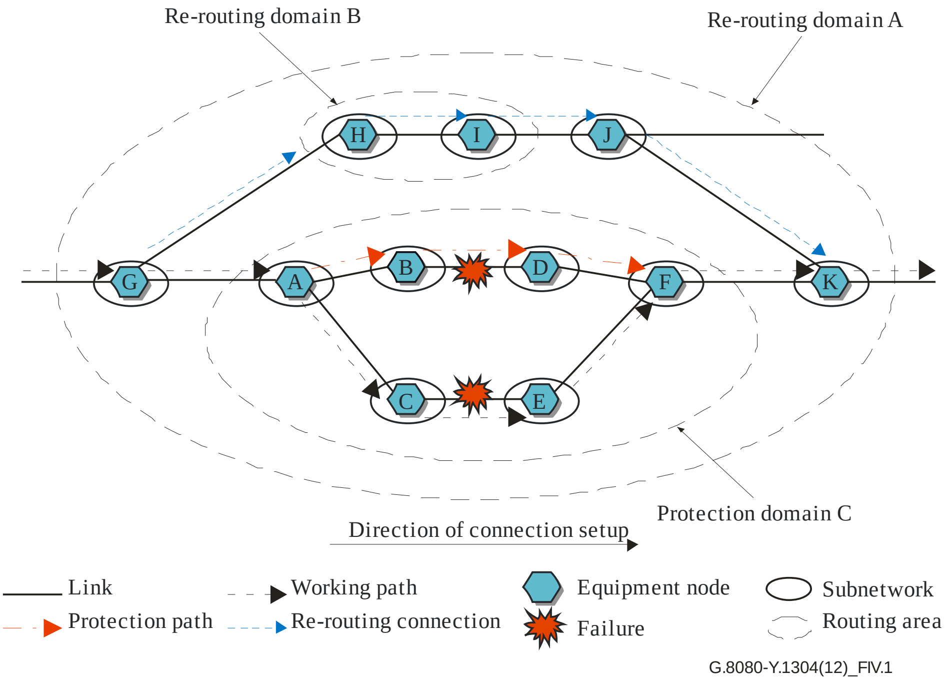
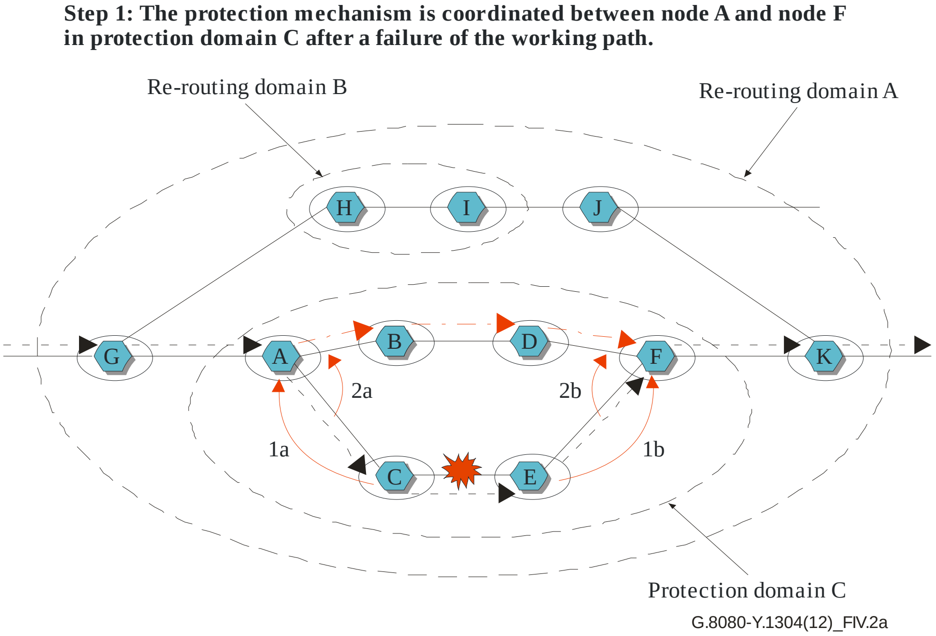
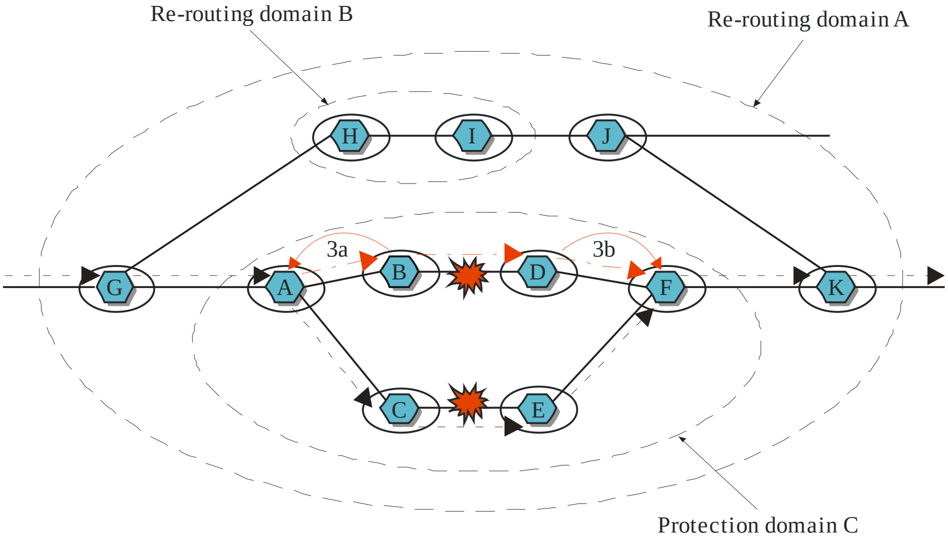


Figure IV.1 – Nested protection and restoration domains

Figure IV.2 shows the signalling flow of a re-routing scenario with source (or step-by-step) routing connection control after protection domain failure events corresponding to Figure IV.1.

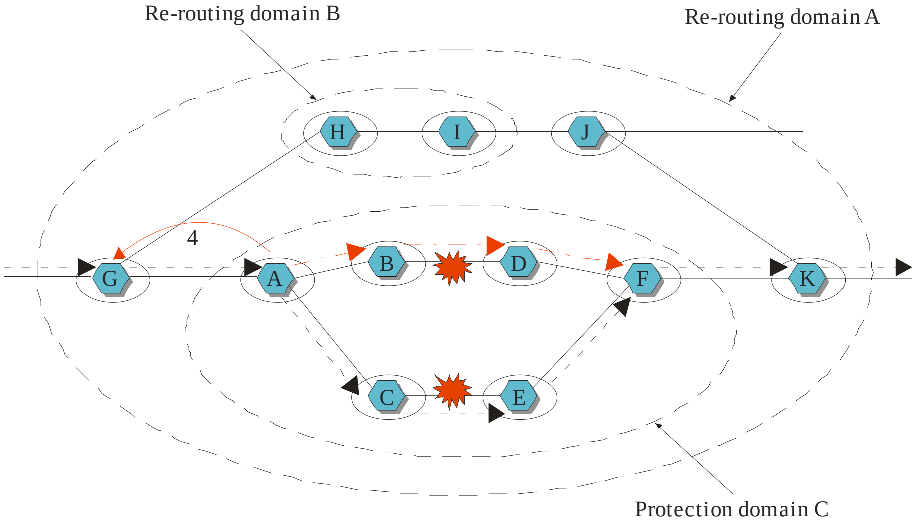

Step 1: The protection mechanism is coordinated between node A and node F
in protection domain C after a failure of the working path.
Re-routing domain B
Re-routing domain A
H
I
J
G
A
B
D
F
K
C
E
2a
2b
1a
1b
Protection domain C
G.8080-Y.1304(12)_FIV.2a

Step 2: Both node A and node F receive a failure notification after the protection path fails again .



G.8080-Y.1304(12)_FIV.2b

Step 3: If the protection domain cannot repair the connection, the connection failure is signalled to the containing restoration domain



G.8080-Y.1304(12)_FIV.2c

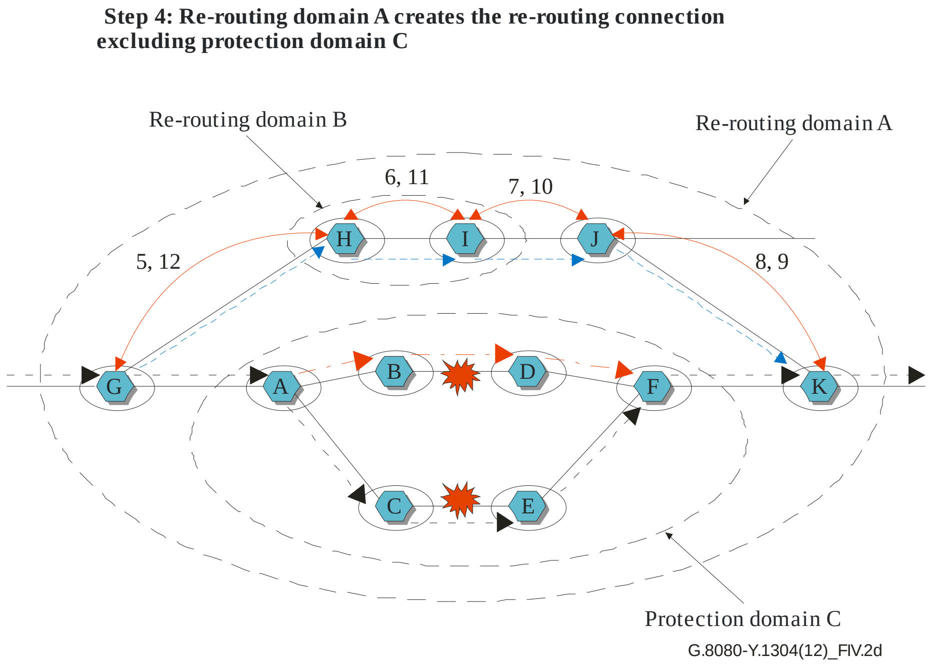


Figure IV.2 – Signalling flow of combination of protection and restoration

In Figure IV.3, the detailed sequence of operations using the source routing method is described after the protection path fails. This corresponds to steps 2-4 of Figure IV.2. The steps involved are listed below:

1) A bidirectional link failure notification generated by the link resource managers (LRM) arrives at the connection controller (CC) containing the failure link information. This occurs in node B or node D.

2) The link failure notification is forwarded to CC$_A$ from CC$_B$ and to CC$_F$ from CC$_D$.

3) At both CC$_A$ and CC$_F$, the NCCs are alerted to the failure of the protection path.

4) NCC$_A$ identifies that there is not any additional assigned capacity to protect the working path in the protection domain.

5) The failure is propagated outside of protection domain C to re-routing domain A. protection domain failure notification is generated by the connection controller (CC), containing the crankback routing information which specifies the failure domain. The protection failure notification is forwarded to CC$_G$ from CC$_{A.}$

6) RC$_G$ is queried for re-routing connection with crankback routing information and returns the set of links excluding protection domain C.

7)-29) Steps 8 to 29 describe the flow of connection set-up using a source routing algorithm which is identical to that described in clause III.2.3 Source and step-by-step routing.

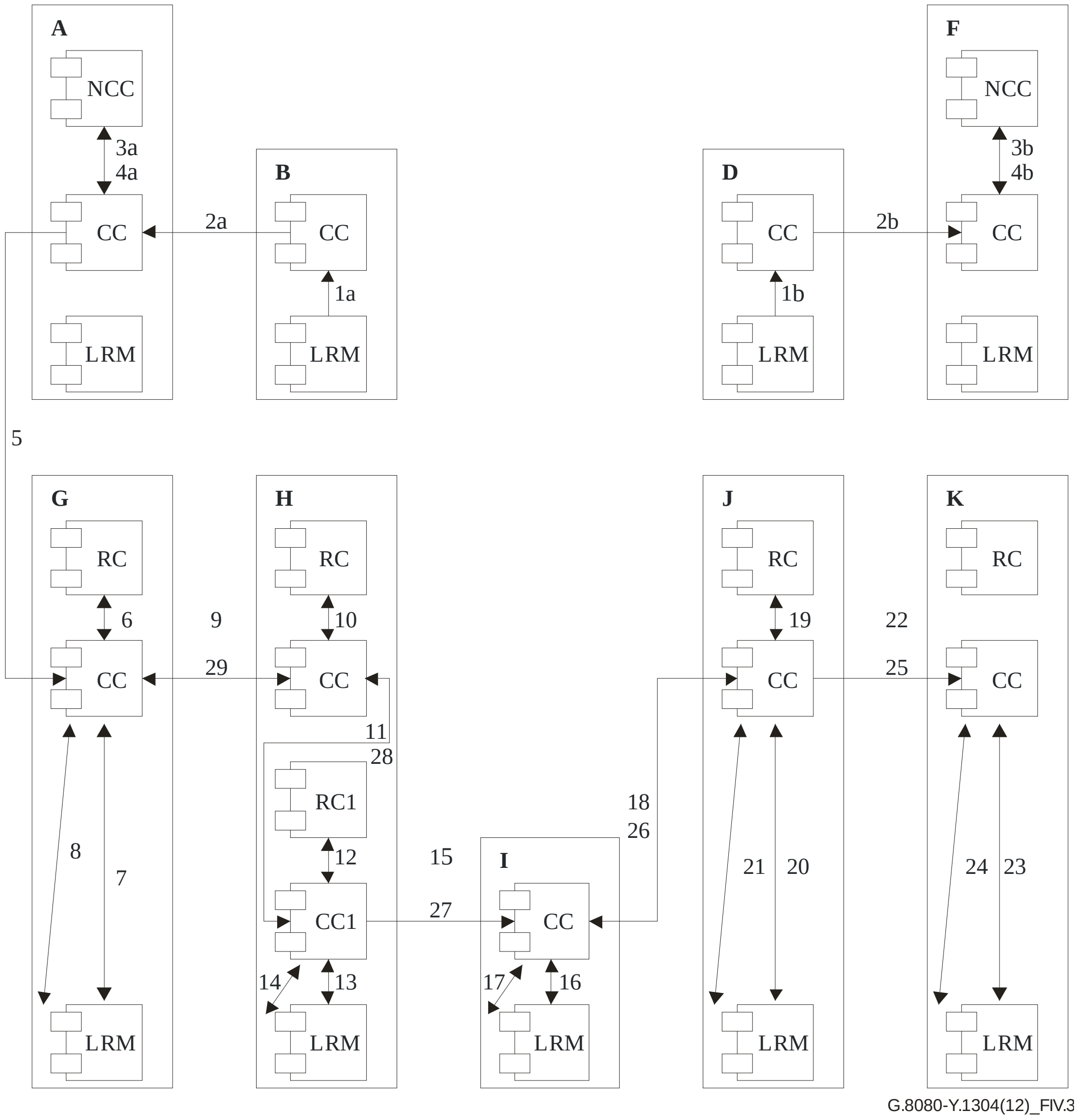


Figure IV.3 – Component interactions of combination of protection and restoration

Appendix V

Example of explicit multi-layer routing topology

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

In some situations connection routing can benefit from a topology view that includes explicit detail from multiple layer networks. One example of this is an ODU layer network that contains within it subnetworks supported by transparent optical (OCh) layer networks.

If the OCh server layer network topology is projected into the ODU client layer it is not possible to associate different routing attributes or constraints specific to the transparent optical subnetworks with that topology. Figure V.1 shows an example of an explicit multilayer topology for such a network. In this example the transitions from the ODU layer to the OCh layer are shown using transitional SNPP links with an adaptation/termination icon. This indicates the transition from ODUk to OTUk to OCh occurs between the routing areas connected by these links. The structure of the OCh layer topology is two rings interconnected via 3R regenerators. The presence of the regenerators is shown using transitional SNPP links with singleton layer processor icons (diamond shape). This indicates the presence of a client layer (OTU) dependent function between the ends of these links.

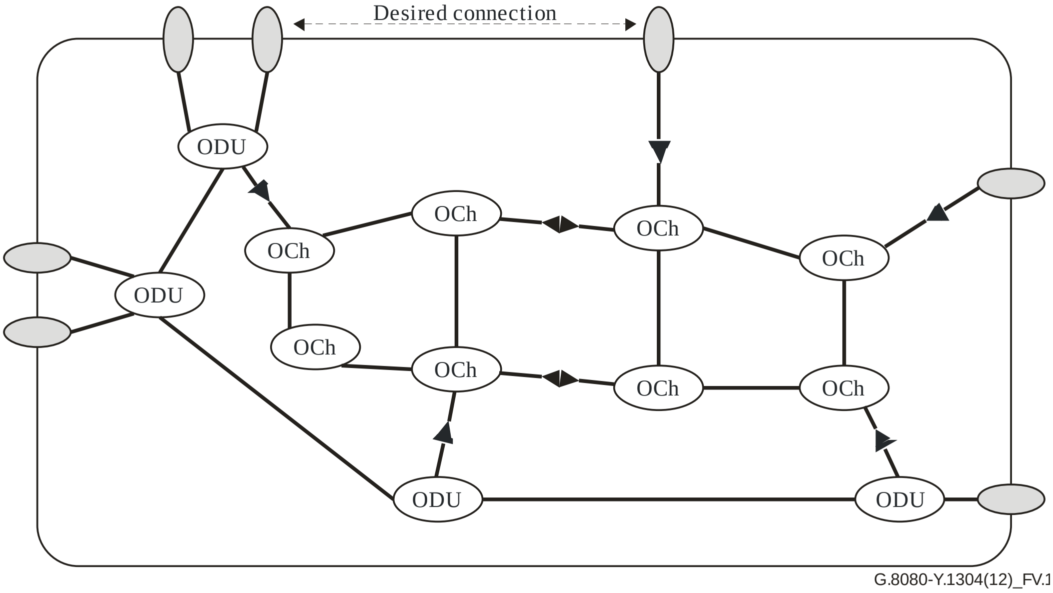


Figure V.1 – Explicit multi-layer topology example

Attributes can be associated with the transitional SNPP links and OCh layer SNPP links that are specific to OCh layer network routing (e.g., modulation type, FEC type, wavelength, etc.). These OCh specific attributes would not be associated with the ODU SNPP links (i.e., those between a pair of ODU routing areas).

This topology may be used to calculate ODUk paths that cross both ODU and OCh routing areas and meet both ODU path constraints and, where necessary, OCh path constraints. This facilitates more optimal route selection across the entire network. The use of an explicit multilayer topology in this case is particularly straightforward since the relationship between ODUk, OTUk, and OCh is 1:1:1. Therefore potential concerns about allocating more server-layer resources than are required by the client layer path do not arise.

Bibliography

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

[b-ITU-T X.731] Recommendation ITU-T X.731 (1992) | ISO/IEC 10164-2:1993, *Information technology – Open Systems Interconnection – Systems management: State management function*.

[b-IETF RFC 2753] IETF RFC 2753 (2000), *A Framework for Policy-based Admission Control*.

[b-UML] Unified Modelling Language (UML) (1999), *OMG UML Specification v. 1.3: OMG document ad/99-06-08*.

ITU-T Y-SERIES RECOMMENDATIONS

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

GLOBAL INFORMATION INFRASTRUCTURE	
General	Y.100–Y.199
Services, applications and middleware	Y.200–Y.299
Network aspects	Y.300–Y.399
Interfaces and protocols	Y.400–Y.499
Numbering, addressing and naming	Y.500–Y.599
Operation, administration and maintenance	Y.600–Y.699
Security	Y.700–Y.799
Performances	Y.800–Y.899
INTERNET PROTOCOL ASPECTS	
General	Y.1000–Y.1099
Services and applications	Y.1100–Y.1199
Architecture, access, network capabilities and resource management	Y.1200–Y.1299
Transport	**Y.1300–Y.1399**
Interworking	Y.1400–Y.1499
Quality of service and network performance	Y.1500–Y.1599
Signalling	Y.1600–Y.1699
Operation, administration and maintenance	Y.1700–Y.1799
Charging	Y.1800–Y.1899
IPTV over NGN	Y.1900–Y.1999
NEXT GENERATION NETWORKS	
Frameworks and functional architecture models	Y.2000–Y.2099
Quality of Service and performance	Y.2100–Y.2199
Service aspects: Service capabilities and service architecture	Y.2200–Y.2249
Service aspects: Interoperability of services and networks in NGN	Y.2250–Y.2299
Numbering, naming and addressing	Y.2300–Y.2399
Network management	Y.2400–Y.2499
Network control architectures and protocols	Y.2500–Y.2599
Smart ubiquitous networks	Y.2600–Y.2699
Security	Y.2700–Y.2799
Generalized mobility	Y.2800–Y.2899
Carrier grade open environment	Y.2900–Y.2999
Future networks	Y.3000–Y.3099

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

SERIES OF ITU-T RECOMMENDATIONS

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

Printed in Switzerland
Geneva, 2012